

**PROCESAMIENTO DIGITAL DE SEÑAL ELECTROCARDIOGRÁFICA CON
TECNOLOGIA DSP ORIENTADO AL ANALISIS DE VARIABILIDAD DE LA
FRECUENCIA CARDIACA**

AUTOR:

Javier Enrique González Barajas

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICO – MECANICAS
MAESTRÍA EN INGENIERÍAS. ÁREA ELÉCTRICA Y ELECTRÓNICA**

Bucaramanga 2005

**PROCESAMIENTO DIGITAL DE SEÑAL ELECTROCARDIOGRÁFICA CON
TECNOLOGIA DSP ORIENTADO AL ANALISIS DE VARIABILIDAD DE LA
FRECUENCIA CARDIACA**

AUTOR:

Ing. Javier Enrique González B.

Maestría en Ingenierías. Área Eléctrica y Electrónica

DIRECTOR:

Ing. MSC Jaime Barrero Pérez

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECANICAS
MAESTRÍA EN INGENIERÍAS. ÁREA ELÉCTRICA Y ELECTRÓNICA**

Bucaramanga 2005

TÍTULO : PROCESAMIENTO DIGITAL DE SEÑAL ELECTROCARDIOGRÁFICA CON TECNOLOGÍA DSP ORIENTADO AL ANÁLISIS DE VARIABILIDAD DE LA FRECUENCIA CARDIACA *

AUTORES: Jaime Barrero Pérez. Ing. Electricista. Msc. **
Javier González Barajas. Ing. Electrónico. Msc.

PALABRAS CLAVES: Variabilidad frecuencia cardiaca, onda R, análisis estadístico, análisis en frecuencia.

La variabilidad de la frecuencia cardiaca, es la fluctuación alrededor del ritmo cardiaco, la cual es un reflejo del funcionamiento del sistema de control cardiorrespiratorio y es el tema central de este proyecto de investigación donde se quiere ha desarrollado una herramienta que permite evaluar el funcionamiento de los sistemas simpático y parasimpático del sistema nervioso autónomo, muy utilizada en el seguimiento de pacientes diabéticos y de pos infarto, dando información sobre el balance del sistema cerebro vascular.

Las variables manejadas en el análisis de la variabilidad de la frecuencia cardiaca en estudios de corto tiempo se extraen de los complejos cardiacos, tomando principalmente la cuantificación de los tiempos R-R.

La medida de la variabilidad de la frecuencia cardiaca (VFC) es de fácil obtención, de carácter no invasiva y confiable. Esta puede ser caracterizada de dos maneras: por el cálculo de índices basados en operaciones estadísticas sobre intervalos R-R (análisis en el dominio del tiempo) o por el análisis espectral de un arreglo de intervalos R-R (análisis en el dominio de la frecuencia). La implementación de ambos métodos requiere calcular con precisión el intervalo de tiempo entre dos ondas R. El análisis puede ser caracterizado sobre segmentos cortos de ECG o en grabaciones de ECG durante 24 horas (estudios Holter).

Durante la toma de electrocardiogramas, pueden existir diferentes fuentes causantes de interferencias que pueden alterar la señal durante la adquisición de los datos para realizar estudios de la VFC, por lo cual, la aplicación de la teoría de filtros adaptativos implementados en este trabajo para el procesamiento de señales electrocardiográficas en tiempo real harán más confiable la medición de la variabilidad de la frecuencia cardiaca.

Los filtros adaptativos son aquellos que son realizados con base en algoritmos que varían su función de transferencia dinámicamente. El filtrado adaptativo se ha utilizado cuando se necesita la realización de simulaciones o modelado de un sistema cuyas características son variables en el tiempo. Las variaciones en los coeficientes de la función de transferencia del filtro son definidas por un criterio de optimización y son realizados de acuerdo con un algoritmo de adaptación. El algoritmo y los coeficientes son escogidos de acuerdo con la aplicación.

* Trabajo de Investigación

** Facultad de Ing. Físico- mecánicas. Maestría en Ingenierías. Área Eléctrica y Electrónica

TITLE: DIGITAL PROCESSING OF ELECTROCARDIOGRAPHIC SIGNAL WITH TECHNOLOGY DSP ORIENTED TO THE ANALYSIS OF VARIABILITY OF THE CARDIAC FREQUENCY *

AUTHORS: Jaime Barrero Pérez. Ing. Electricista. Msc. **

Javier González Barajas. Ing. Electrónico. Msc.

Juan Camilo Guzmán. MD.

KEY WORDS: Variability cardiac frequency, wave R, statistical analysis, frequency analysis.

The variability of the cardiac frequency, is the fluctuation around the heart rate, which is a reflection of the operation of the cardio-respiratory control system and it is the main subject of this project of investigation where it wanted has developed a tool that allows to evaluate the operation of the sympathetic and parasympathetic systems of the independent nervous system, very used in the pursuit of diabetic and pos infarct patients, giving information about the balance of the vascular brain system.

The variables handled in the analysis of the variability of the cardiac frequency in short time studies are extracted of the complexes cardiac, taking mainly the quantification from times R-R.

The measurement of the variability of the cardiac frequency (VFC) is of easy obtaining, of noninvasive and reliable character. This can be characterized of two ways: by the calculation of indices based on statistical operations on intervals R-R (analysis in the dominion of the time) or by the spectral analysis of an arrangement of intervals R-R (analysis in the dominion of the frequency). The implementation

of both methods requires to accurately calculate the time interval between two waves R. The analysis can be characterized on short segments of ECG or in recordings of ECG during 24 hours (Holter studies).

During the taking of electrocardiograms, different sources can exist causes from interferences that can alter the signal during the acquisition of the data to make studies of the VFC, thus, the application of the theory of implemented adaptive filters in this work for the electrocardiographic signal processing in real time will make the measurement more reliable of the variability of the cardiac frequency.

The adaptive filters are those that are made with base in algorithms that vary their transference function dynamically. The adaptive filtrate has been used when is needed the accomplishment of simulations or modeled a system whose characteristics are variable in the time. The variations in the coefficients of the transference function of the filter are defined by an optimization criterion and are made in agreement with an adaptation algorithm. The algorithm and the coefficients are chosen in accordance with the application.

* Work Research

** Department of Phisical – Mecanical Engineers. Master of Engineer. Area of Electrical and Electronics.

INDICE

INTRODUCCIÓN	1
1. PROCESAMIENTO DIGITAL DE LA SEÑAL ELECTROCARDIOGRÁFICA	2
1.1 LA ELECTROCARDIOGRAFÍA.....	2
1.2 Interferencias En La Adquisición De Señales De Ecg.....	5
1.3 Adquisición Y Digitalización.....	7
1.4 Filtrado Digital Del Ecg.....	11
1.4.1 Filtros suavizadores (<i>smoothing</i>)	11
1.4.2 Filtro de muesca (<i>Notch</i>).....	13
1.4.3 Filtros derivadores	17
2. FILTRADO ADAPTATIVO	23
2.1 Principios básicos del filtrado adaptativo.....	23
2.2 Filtro digital notch adaptativo diseñado para remoción del ruido de la línea de potencia de señales biomédicas.	26
2.2.1 Diseño de un filtro FIR notch adaptativo de segundo orden.	26
2.2.2 Sistema de Cancelación de ruido.....	29
2.2.3 Sistema para estimación de la señal de referencia $V_2(n)$	34
2.2.4 Filtrado por cancelación para eliminación del corrimiento de la línea de base.	36
3. EL ANÁLISIS DE LA VARIABILIDAD DE LA FRECUENCIA CARDIACA 42	
3.1 MEDIDAS DE CORTO TIEMPO DE LA VARIABILIDAD DE LA FRECUENCIA CARDIACA.....	43
3.1.2 Detección de la onda R.	44
3.1.2 Medidas en el dominio del tiempo de la VFC.	53
3.1.3 Medidas en el dominio de la frecuencia de la VFC.	55
4. DISEÑO Y ENSAYO DE ALGORITMOS.....	65
4.1 Adquisición y Digitalización de la Señal Electrocardiográfica.....	65
4.2 Filtrado adaptativo para remover el corrimiento de la línea de base.	68
4.3 Filtrado adaptativo para reducir la interferencia de la señal de 60 Hz.....	72
4.4 Detección de la onda R del complejo cardíaco.	90
4.5 Análisis en el dominio del tiempo de los intervalos RR.....	98
4.5 Análisis en el dominio de la frecuencia de los intervalos RR.	99
4.6 Cálculo de índices del dominio de la frecuencia.....	102
5. RESULTADOS FINALES Y CONCLUSIONES	104
5.1 Pruebas al sistema de remoción del desplazamiento de la línea de base.	105
5.2 Pruebas al sistema de remoción de interferencia de 60 Hz.....	108
5.3 Pruebas realizadas a los algoritmos de análisis de la V.F.C.	115
6. CONCLUSIONES.....	120
REFERENCIAS BIBLIOGRÁFICAS	1201
APENDICE I.....	123
APENDICE I I	171

LISTA DE FIGURAS

FIGURA 1. Derivaciones bipolares.	3
FIGURA 2. Derivaciones unipolares.	3
FIGURA 3. Papel electrocardiográfico.	4
FIGURA 4. Interferencias en ECG.	6
Figura 5. Respuesta en frecuencia del filtro FIR	9
Con frecuencia de corte $\pi/4$.	9
Figura 6. Señal senosoidal diezmada.	10
Figura 7. Respuesta en frecuencia del filtro promediado	12
móvil de Hanning.	12
Figura 8. Señal filtrada con Filtro Hanning.	13
Figura 9. Respuesta en frecuencia Típica del filtro <i>Notch</i> (orden 3).	14
Figura 10. Respuesta al impulso del	15
Filtro <i>notch</i> de orden 50.	15
Figura 11. Respuesta en el dominio de la frecuencia	15
del filtro <i>Notch</i> de orden 50.	15
Figura 12. Efecto Filtrado <i>Notch</i> orden 3.	16
Figura 13. Efecto Filtrado <i>Notch</i> orden 50.	16
Figura 14. Magnitud y fase del derivador	17
de dos puntos.	17
Figura 15. Filtrado con diferenciador de dos puntos.	18
Figura 16. Respuesta en frecuencia	19
del filtro diferencia central de tres puntos.	19
Figura 17. Filtrado con diferenciador central	20
de tres puntos.	20
Figura 18. Respuesta en frecuencia para filtro de segunda derivada.	21
Figura 19. Filtrado basado en la segunda derivada.	22
Figura 20. Estructura general de	23
Un filtro adaptativo.	23
Figura 21 . Filtro Adaptativo FIR.	24
Figura 22. Diagrama de bloques de un filtro FIR notch (Clásico y	
adaptativo.	28
Figura 23. Sistema adaptativo de cancelación de ruido.	30
Figura 24. Algoritmo del sistema de cancelación de ruido para	31
estimar la señal de error.	31
Figura 25. Espectro de la señal de entrada.	32
Figura 26. Señales senosoidales con componente de 60 Hz.	32
Figura 27. Resultados obtenidos con filtrado el sistema de cancelación y	
con un filtro <i>nocth</i> FIR.	33
Figura 28. Corrimiento de la Línea de base en un ECG.	36
Figura 29. Filtro por cancelación para eliminación del corrimiento	37
de la línea de base del ECG.	37
Figura 30. Señal ECG con corrimiento de línea de base	39
Y cálculo de los valores medios.	39
Figura 31. Función estimada de la línea de base.	39
Figura 32. Resultado obtenido de la eliminación de la línea de base.	40
Figura 33. Comparación entre filtrado por cancelación y filtro FIR	41
pasa altos.	41
Figura 34. Filtrado pasa banda y Primera y segunda derivada aplicada al	
ECG.	45

Figura 35. Suma ponderada y aplicación de umbral al ECG.....	45
Figura 36. Etapas de filtrado para detección.....	46
del complejo QRS.	46
Figura 37. Respuesta en frecuencia del filtro pasa banda.	47
Figura 38. Señal de entrada y de salida del filtro Pasa bandas. (Dominio del tiempo).	48
Figura 39. Señal de entrada y de salida del filtro Pasa bandas. (Dominio de la frecuencia).	48
Figura 40. Respuesta en frecuencia del Filtro derivador.	49
Figura 41. Salida del filtro derivador. (Dominio del tiempo).	49
Figura 42. Salida del filtro derivador. (Dominio de la frecuencia).	50
Figura 43. Elevación al cuadrado de la señal.	50
Figura 44. Señal ECG después del filtrado por integrador de ventana móvil.	51
Figura 45. Detección de la onda R por medio de un nivel de umbral.	52
Figura 46. Componentes Espectrales la señal de ritmo cardiaco.	56
Figura 47. Comparación de las componentes espectrales en un sujeto normal y pacientes con neuropatías. (condición normal y cambio postural).....	57
Figura 48. Señal obtenida por interpolación lineal de los intervalos RR. ...	58
Figura 49. Análisis espectral de la señal obtenida por interpolación lineal.	58
Figura 50. Señal con muestreo uniforme y no uniforme.	62
Figura 51. Espectro en potencia con FFT de la señal uniformemente muestreada.	62
Figura 52. Espectro en potencia por el método de Lomb. Señal uniformemente muestreada.	63
Figura 53. Señal obtenida por interpolación y espectro en frecuencia por método de la FFT.	63
Figura 54. Espectro en potencia por el método de Lomb. Señal no uniformemente muestreada.	64
Figura 55. Estimación de PSD de intervalos RR por Medio del método de Lomb.	64
Figura 56. Etapa de acondicionamiento.	65
Figura 57. Algoritmo de Digitalización y diezmado.	66
Figura 58. Señal adquirida y diezmada Con el DSK 5402.	67
Figura 59. Filtrado adaptativo por cancelación para remoción del corrimiento de la línea de base.	68
Figura 60. Algoritmo de Filtrado adaptativo para la remoción de corrimiento de la línea de base.	68
Figura 61. Señal senosoidal con corrimiento de línea de base.	69
Figura 62. Espectro de la señal senosoidal con corrimiento de la línea de base.	69
Figura 63. Función estimada de la línea de base.	70

Figura 64. Espectro de la función estimada de la línea de base.....	70
Figura 65. Señal senosoidal filtrada.....	71
Figura 66. Espectro de la señal filtrada.....	71
Figura 67. Respuesta en frecuencia Filtro FIR notch.....	72
Figura 68. Respuesta en frecuencia de la función estimadora.....	73
Figura 69. Señal original, señal contaminada y resultado del e(n).....	74
Obtenido con el sistema por cancelación.....	74
Figura 70. Comparación de resultados entre el sistema por cancelación y por medio de un filtro FIR notch convencional de orden 50.....	75
Figura 71. Espectro de señales obtenidas con la función estimada por ...	76
Análisis de Fourier.....	76
Figura 72. Espectro de señales obtenidas con la función estimada por ...	76
Figura 73. Valor de frecuencia central.....	77
que depende de una función rampa.....	77
Figura 74. Magnitud del espectro en.....	78
Potencia para una señal con frecuencia fija y variable.....	78
Figura 75. Magnitud del espectro en potencia Para una señal contaminada con ruido de 60 Hz con frecuencia variable.....	78
Figura 76. Resultado obtenido con sistema por cancelación.....	79
Figura 77. Resultado obtenido con el modelo senosoidal en la frecuencia de la interferencia.....	80
Figura 78. Resultado comparativo entre el sistema adaptativo y convencional.....	81
Figura 79. Espectro señal ECG contaminada con interferencia de 60 Hz..... De frecuencia fija.....	82
Figura 80. Salida Del Sistema Por Cancelación (Domino De La Frecuencia).....	82
Figura 81. señal filtrada con filtro adaptativo y por filtro notch convencional orden 50. (Dominio del tiempo).....	83
Figura 82. Salida sistema por cancelación y por filtro notch convencional orden 50. (Dominio de la frecuencia).....	84
Figura 83. Señal ECG contaminada con interferencia de frecuencia variable y resultado obtenido con el sistema por cancelación y con filtro notch convencional de orden 50. (Dominio del tiempo).....	84
Figura 84. Señal ECG contaminada con interferencia de frecuencia variable y resultado obtenido con el sistema por cancelación y con filtro notch convencional de orden 50. (Dominio de la frecuencia).....	85
Figura 85. Algoritmo de filtrado para remoción de la interferencia de 60 Hz.....	86
Figura 86. Respuesta en frecuencia del filtro notch con valor inicial..... del coeficiente a_1	87
Figura 87. Resultado obtenido de filtrar señal ECG con interferencia con de 0.25 de la señal de entrada.(Dominio del tiempo).....	87
Figura 88. Resultado obtenido de filtrar señal ECG con interferencia con de 0.25 de la señal de entrada.(Dominio de la frecuencia).....	88
Figura 89. Respuesta en frecuencia del filtro notch con valor final del coeficiente a_1	89
Figura 90. Señal ECG de entrada.(Detección de onda R).....	90
Figura 91. Proceso de filtrado pasabanda. (Detección de onda R).....	91
Figura 92. Filtrado con derivador. (Detección de onda R).....	92

Figura 93. Elevación al cuadrado de la señal ECG. (Detección de onda R)	93
Figura 94. Integrador de Ventana móvil. (Detección de onda R)	94
Figura 95. Aplicación del umbral. (Detección de onda R)	95
Figura 96. Tiempos RR. (Detección de onda R)	95
Figura 97. Señal ECG sin interferencia e intervalos RR	96
Calculados.	96
Figura 98. Señal ECG con interferencia e intervalos RR	96
Calculados.	96
Figura 99. Detección de intervalos RR en señal ECG con	97
corrimiento en su línea de base.	97
Figura 100. Arreglo de datos tiempos RR.	98
Figura 101. Resultado obtenido por la plataforma DSK5402 para análisis	99
en el dominio del tiempo.	99
Figura 102. Tiempos RR.	100
Figura 103. Análisis en el dominio de la Frecuencia de los tiempos RR.	100
Método de Lomb.	100
Figura 104. Análisis en el dominio de la Frecuencia de los tiempos RR.	101
Método de la FFT.	101
Figura 105. Resultado del análisis espectral de	101
tiempos RR por medio del DSP.	101
Figura 106. Índices del análisis en el dominio de la	102
Frecuencia de los tiempos RR.	102
Figura 107. Resultado del los índices calculados del análisis espectral de	103
tiempos RR por medio del DSP.	103
Figura 108. Algoritmo final.	106
Figura 109. Proceso de remoción de interferencia	107
provocada por una señal senosoidal ($F= 0.2$ Hz).	107
Figura 110. Proceso de remoción de interferencia	108
provocada por una señal senosoidal ($F= 0.3$ Hz).	108
Figura 111. Señal ECG + interferencia de 60 Hz.	109
Simulador Lion Heart.	109
Figura 112. ECG + Interferencia de 59 y 59.5 Hz.	110
Figura 113. ECG + Interferencia de 60.1 y 61.0 Hz.	111
Figura 114. Señal ECG + 60 Hz con amplitud 0.25.	112
Figura 115. Señal ECG + 60 Hz con amplitud 0.5.	113
Figura 116. Señal ECG + 60 Hz con amplitud 1.	114
Figura 117. Pruebas de análisis de VFC en dos sujetos en reposo.	115
Figura 118. Comparación en Pruebas de análisis de VFC en dos sujetos	116
en reposo (Lomb y FFT).	116
Figura 119. Prueba realizada en mesa basculante.	116
Figura 120. Comparación de espectros con lomb y FFT	117
con datos tomados en mesa basculante.	117
Figura 121. Interfaz gráfica para visualización de datos importados de la	118
plataforma DSK5402.	118

TABLAS

TABLA 1. Medidas en el dominio del tiempo de la VFC.	53
TABLA 2. Índices del dominio del tiempo. (Sujeto normal y diabético tipo I)	54
TABLA 3. Índices del dominio del tiempo. (Sujeto normal y diabético con y sin NACV)	54
TABLA 4. Medidas en el dominio de la Frecuencia en estudios de corto tiempo de la VFC.	56

INTRODUCCIÓN

El estudio e implementación de algoritmos para análisis de la variabilidad de la frecuencia cardíaca como herramienta para el estudio de las enfermedades del sistema nervioso autónomo ha sido el tema central en este trabajo de investigación, en el cual se ha tratado las principales técnicas de tratamiento digital de señales para ser implementadas en un procesador digital que dejan las bases para el diseño de equipos autónomos para el estudio del electrocardiograma y los continuos cambios de su frecuencia instantánea.

En el primer capítulo se ha recopilado los conceptos básicos de electrocardiografía y de los requerimientos básicos para su adquisición. Teniendo en cuenta de que uno de los mayores problemas encontrados en el registro del electrocardiograma es la aparición de las interferencias, se han referenciado las técnicas convencionales para el tratamiento de las distorsiones causadas por la señal de 60 Hz de las fuentes de servicio eléctrico y el corrimiento de la línea de base.

Por ser el filtrado adaptativo una de las técnicas que actualmente se utiliza para la remoción de interferencias del electrocardiograma, se ha dedicado el segundo capítulo al estudio del diseño e implementación de estos filtros para el tratamiento del corrimiento de la línea de base y de la interferencia causada por la señal de 60 Hz del sistema eléctrico.

El tema de la variabilidad de la frecuencia cardíaca es tratado en el tercer capítulo, en el cual se describen los diferentes tipos de análisis matemáticos utilizados para el estudio de los intervalos RR en los dominios del tiempo y de la frecuencia. En este capítulo se destaca la comparación de dos métodos para el análisis de la frecuencia cardíaca de los tiempos RR: el método tradicional utilizando la interpolación y FFT contra el método de Lomb de análisis espectral de datos desigualmente muestreados en el tiempo.

El cuarto capítulo contiene los diferentes algoritmos implementados en la plataforma DSK5402 del fabricante *Texas Instruments*. Este capítulo ilustra los diferentes resultados previos obtenidos durante la simulación y ensayo de cada algoritmo por medio de la herramienta matemática *Matlab* y la comparación con los resultados obtenidos de los algoritmos implementados en el DSP.

Los resultados obtenidos durante las pruebas realizadas con señales electrocardiográficas provenientes de equipos de simulaciones y de sujetos voluntarios han sido registrados en el capítulo número cinco.

En el transcurso de este informe final se han referenciado una serie de archivos que contienen las líneas de código que corresponden a rutinas elaboradas en la herramienta *Matlab* para ilustrar las diferentes simulaciones de los algoritmos implementados en el DSP. Estos códigos se pueden detallar en el apéndice 1. Los códigos elaborados en lenguaje C que corresponden a los algoritmos implementados en la plataforma DSK5402 que son mencionados en el capítulo cuatro son detallados en el apéndice 2.

1. PROCESAMIENTO DIGITAL DE LA SEÑAL ELECTROCARDIOGRÁFICA

1.1 LA ELECTROCARDIOGRAFÍA.

La electrocardiografía es el estudio y registro de la actividad eléctrica del corazón con el fin de realizar un diagnóstico de su estado.

Dentro del corazón existe una gran cantidad de fibras musculares que están interconectadas unas con otras y constituyen todo un sistema especializado de conducción donde se origina el impulso eléctrico que activa el corazón por lo cual es el marcapaso principal y está formado por el nodo sinusal ubicado en la pared de la aurícula derecha, en la unión entre la aurícula y la vena cava superior. También se encuentra el nodo auriculoventricular (AV) localizado en el subendocardio hacia el lado derecho del tabique intraauricular por encima del anillo de la válvula tricúspide (cerca de su hojuela septal) y próximo a la desembocadura del seno coronario.

Para poder estudiar el comportamiento del corazón es aplicada la técnica del electrocardiograma en la cual se toma una gráfica que resulta de registrar la actividad eléctrica del corazón, utilizando un aparato que está compuesto de un mecanismo de transporte de papel, un circuito amplificador y un sistema que conmuta con las diferentes derivaciones que forman el trazado electrocardiográfico, un cable con terminales que se conectan a brazos y piernas y un quinto alambre que permite tomar las derivaciones del precordio.

Un trazado normal de un electrocardiograma se compone de 12 derivaciones: tres derivaciones bipolares, tres derivaciones unipolares y seis derivaciones precordiales. Una derivación es bipolar cuando se registran la diferencia de potenciales entre dos puntos del cuerpo y las unipolares registran la diferencia de potenciales entre un punto del cuerpo y el centro eléctrico del corazón.

Las derivaciones bipolares se forman con la derivación I conectada a ambos brazos; la derivación II conectada al brazo derecho y la pierna izquierda, y la derivación III, conecta el brazo izquierdo y la pierna izquierda. El electrodo conectado a la pierna derecha es simplemente una toma a tierra para efectos de seguridad. Según Einthoven los electrodos están colocados en los vértices de un triángulo equilátero en cuyo centro se encuentra ubicado el corazón (triángulo de Einthoven). Sobre esta hipótesis se basan todos los cálculos de la electrocardiografía.(ver figura 1)

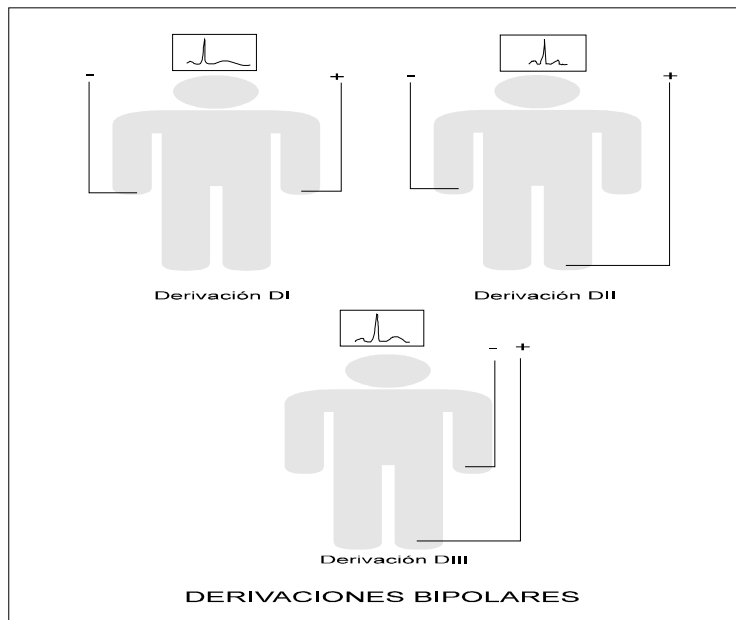


FIGURA 1. Derivaciones bipolares.

Las derivaciones unipolares fueron propuestas por Wilson en 1934, donde se plantea un método de medir, no la diferencia de potencial entre dos electrodos, sino la potencia de fuerza en una extremidad. Mediante conexiones, junta el terminal negativo de los alambres de las tres extremidades en un solo electrodo recibiendo el nombre de terminal central. (ver figura 2)

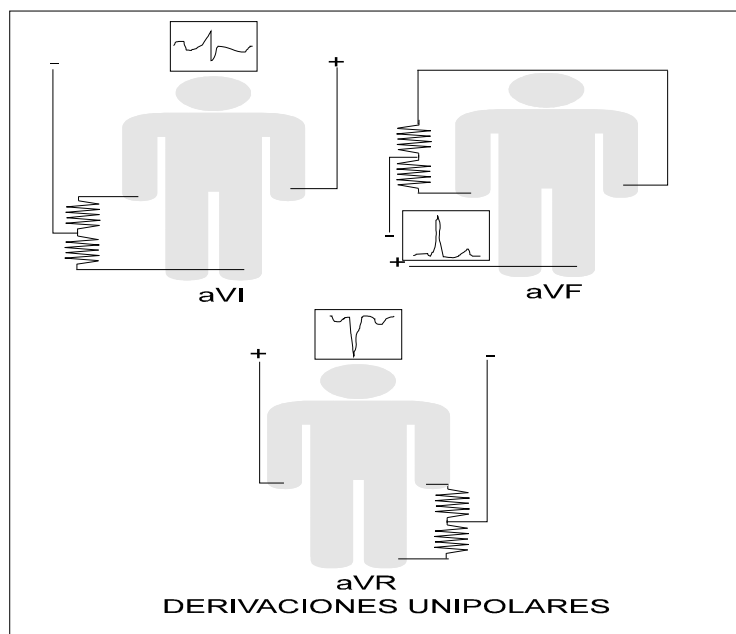


FIGURA 2. Derivaciones unipolares.

En el electrocardiograma se adquieren un conjunto de señales que están relacionadas con el comportamiento de cada sección del sistema eléctrico del corazón.

La primera onda que aparece en electrocardiograma es la onda P la cual indica la activación auricular y dura de 60 a 110 milisegundos.

La activación del nodo sinusal se produce antes del principio de la onda P pero no se registra en el ECG de superficie.

La activación del nodo auriculoventricular no produce deflexión en el electrocardiograma de superficie correspondiendo el segmento PR(PQ), que es isoeléctrico en gran parte de este lapso.

Una vez el impulso alcanza el Haz de His, la velocidad de conducción se hace mayor. Se ha estimado que el tiempo que tarda el impulso desde esta estructura hasta el comienzo del QRS va entre 35 y 55 milisegundos.

La secuencia de la despolarización ventricular es determinada en forma fundamental por la propagación del impulso eléctrico a través de un sistema de conducción especializado, en donde el miocardio ventricular es activado de endocardio a epicardio a través de las ramificaciones terminales de la red de Purkinje lo cual permite una activación de ambos ventrículos casi sincrónica. A diferencia de la activación auricular que se hace en forma longitudinal, la activación del miocardio ventricular se hace en forma transversal.

El tiempo desde el inicio hasta la terminación de la activación del miocardio ventricular corresponde al intervalo del complejo QRS, el cual en promedio, dura entre 60 y 100 milisegundos. Tiene la misma duración que la activación auricular aunque la masa ventricular es mucho mayor que la masa auricular. Esto se debe a la presencia del sistema de conducción especializado en los ventrículos el cual no aparece en las aurículas definido claramente.

En la figura 3 se pueden apreciar las ondas que componen el electrocardiograma en su respectivo orden y su representación en un papel electrocardiográfico. En el papel electrocardiográfico se ubica el tiempo en el eje horizontal y la amplitud en milivoltios en el eje vertical. [1 ,2, 3, 4 y 5]

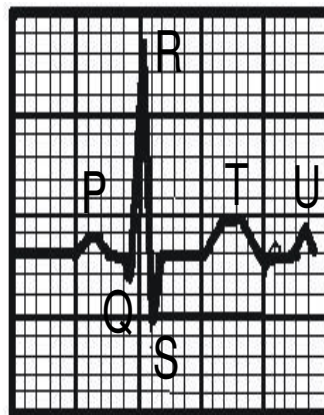


FIGURA 3. Papel electrocardiográfico.

1.2 INTERFERENCIAS EN LA ADQUISICIÓN DE SEÑALES DE ECG

Actualmente los sistemas de detección de ECG estándar (tanto clínico como de monitoreo) deben estar en capacidad de eliminar o reducir las perturbaciones dadas por fuentes de ruido externos o internos. La presencia de dichas perturbaciones conlleva a distorsiones de la señal ECG y por consiguiente, a inconsistencias a la hora de realizar un diagnóstico.

Uno de los mayores problemas encontrados en el registro de ECG es la aparición frecuente de la interferencia de 60 Hz y sus armónicos generados por la fuente de alimentación. Las fuentes de esta interferencia son los equipos eléctricos que operan cerca de los dispositivos electromédico, incluyendo luces, cables de A.C., etc.[6]

La pérdida de contacto entre los electrodos y la piel es otra causa de interferencia que puede alterar la señal electrocardiográfica. En estas condiciones se puede producir un desplazamiento transitorio de la línea de base causado por el cambio en la impedancia electrodo-piel debido al movimiento del electrodo. [7]

Las contracciones musculares o electromiografía (EMG) causan interferencias a niveles de mV. La línea de base del EMG está usualmente en el rango de los μ V. Las señales resultantes de la contracción muscular pueden ser asumidas como transcientes abruptos de ruido Gaussiano de valor medio igual a cero y banda limitada.[8][9]

En la siguiente figura se ilustran unos ejemplos de interferencias en electrocardiograma. (Ver figura 4)

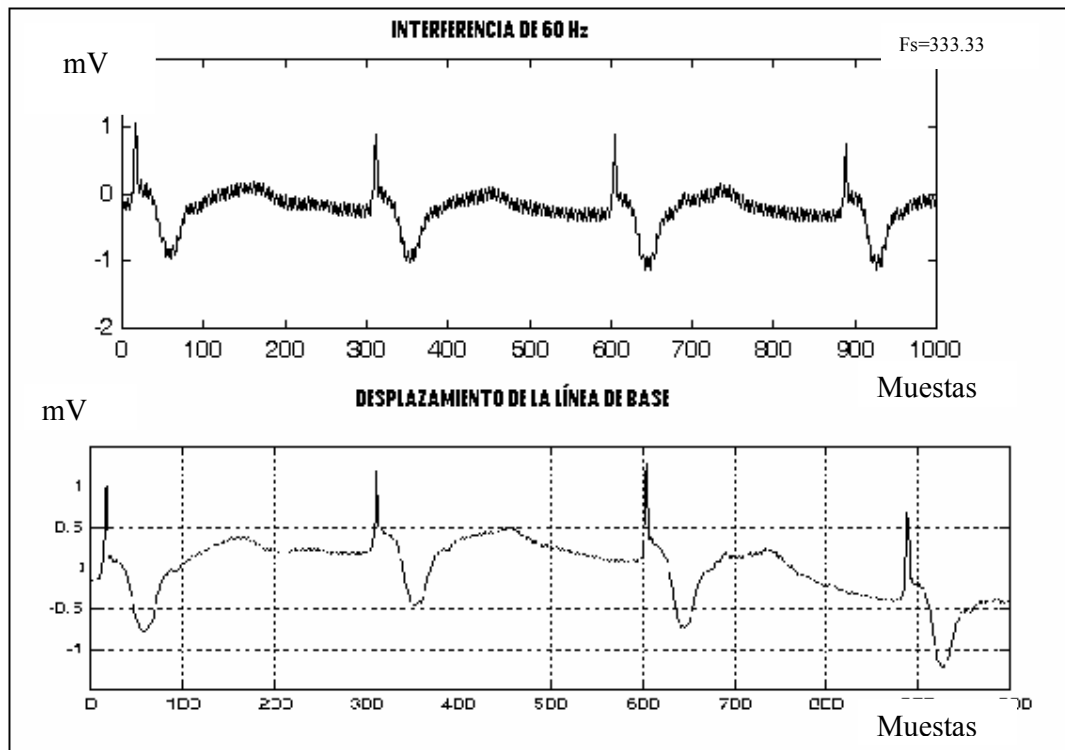


FIGURA 4. Interferencias en ECG.

1.3 ADQUISICIÓN Y DIGITALIZACIÓN.

Los sistemas de adquisición de datos permiten tomar muestras de señales provenientes de algún fenómeno para después reconstruirlas con el fin de realizar un posterior procesamiento dentro de un sistema digital con ayuda de transductores que toman los potenciales electrofisiológicos de una señal biológica, pasando a una etapa de acondicionamiento adquiriendo la señal de salida del transductor para filtrarla, amplificar y aislarla para entregarla a la etapa de conversión analógica a digital. [10]

Por lo general la etapa de acondicionamiento posee una primera sección de preamplificación compuesta de un amplificador diferencial con alta impedancia de entrada, un sistema de protección contra descargas de altos voltajes producidos por los desfibriladores y un filtro pasa banda (0.05 Hz – 100Hz). Por lo general el preamplificador está basado en un amplificador de instrumentación con baja ganancia (aproximadamente 60 v/v) y está conectado al paciente a través de un par de electrodos. Como etapa siguiente al preamplificador, se conecta a una etapa de amplificación con una ganancia ya superior (100 v/v), con la cual se logra tener una señal electrocardiográfica robusta y con los niveles de voltaje adecuados para ser aislada y digitalizada. [11 y 12]

Las señales, como las registradas en un electrocardiograma, pueden describir una variedad muy amplia de fenómenos físicos y aunque se pueden representar de muchas maneras, la información dentro de una señal está contenida en un patrón de variaciones. Bajo ciertas condiciones esta señal se pueden representar y reconstruir completamente, partiendo de sus valores instantáneos, o muestras igualmente espaciadas en el tiempo permitiendo de forma más flexible el procesamiento de las señales, lo cual es conocido como el teorema de muestreo de Nyquist. [13] Cuando la señal que se desea digitalizar es muestreada con una frecuencia por debajo de la establecida por el criterio de Nyquist se presenta el fenómeno conocido como *aliasing* (alias) que causa una incorrecta reconstrucción de la señal adquirida. [14 y 15] Para el caso de las señales electrocardiográficas se recomienda una mínima frecuencia de muestreo entre 250 y 500 HZ, con una resolución de 12 bits, pero actualmente se ha optado por emplear la electrocardiografía de alta resolución o también llamada de alta frecuencia que es de gran utilidad a la hora de realizar estudios que permiten detectar despolarizaciones cardiacas de baja amplitud y de alta frecuencia que no son apreciables en el electrocardiograma común de doce derivaciones y con bajas tasas de muestreo. [16 y 17]

En muchos casos, después de tener una señal digitalizada, es necesario reducir el número de muestras adquiridas por medio de un cambio en la tasa de muestreo como método para ahorrar espacio en memoria. En este trabajo de investigación se ha implementado la técnica del diezmado. Este es un proceso de conversión de tasa de muestreo en el dominio digital el cual se puede ver como una operación de filtrado lineal. La señal de entrada $x(n)$ se caracteriza por poseer una tasa de muestro de $F_x=1/T_x$ y la señal de salida $y(m)$ mediante $F_y=1/T_y$, donde T_x y T_y son los intervalos de muestreo correspondientes. El objetivo de la conversión de tasas de muestreo es obtener otra secuencia $y(m)$ directamente de $x(n)$. El proceso de reducción de la tasa de muestro por un factor D (submuestreo por D) se denomina diezmado, en

donde la señal resultante puede ser una versión de la señal de entrada $x(n)$ con un valor en su frecuencia de muestreo de $F_x/2D$.

Para evitar el *aliasing* es necesario reducir el ancho de banda de $x(n)$ a $F_{\max}=F_x/2D$ o, equivalentemente a $\omega_{\max} = \pi/D$ y después se debe submuestrear por D . En este trabajo de investigación, se utiliza una frecuencia de muestreo equivalente a 2KHz y al factor de submuestreo se le ha asignado un valor de 4, por lo cual requiere de un filtro pasa bajos con frecuencia de corte $\pi/4$.

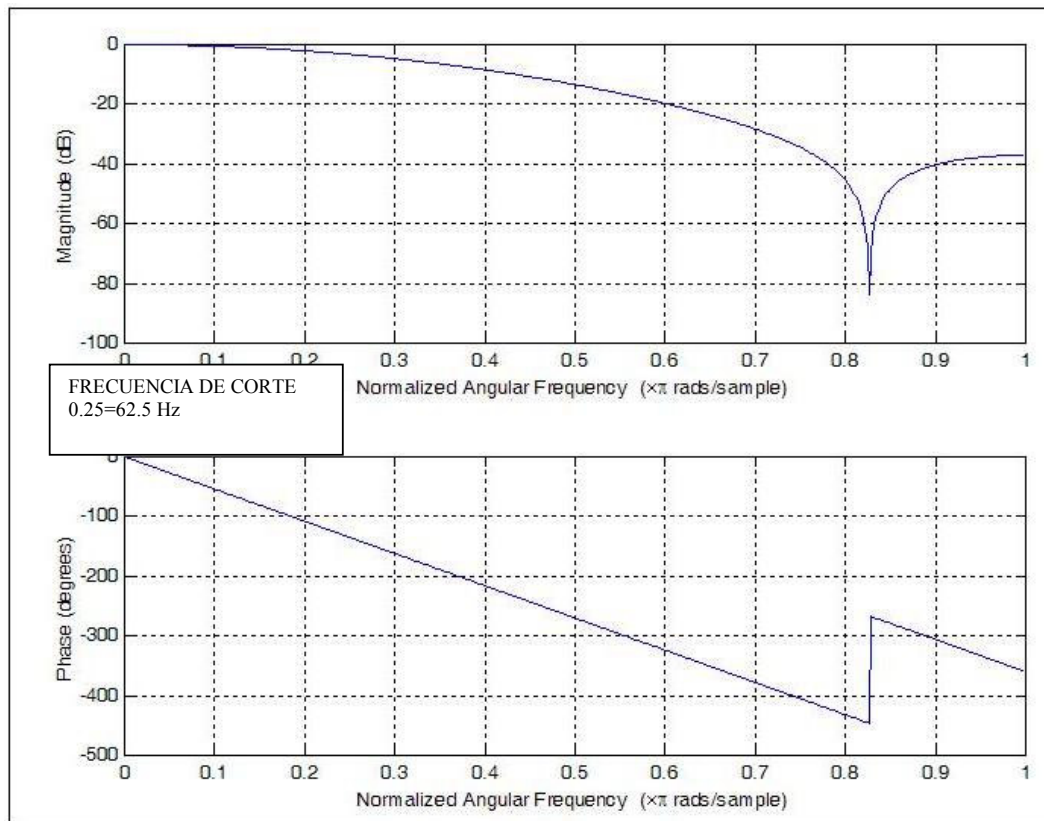
Por medio del asistente matemático Matlab[®] se pueden obtener los coeficientes para un filtro FIR pasa bajos con frecuencia de corte $\pi/4$, utilizando el comando *FIR1(N,Wn)*. Donde N toma el valor del orden del filtro y Wn equivale al valor de la frecuencia de corte. Como resultado se obtienen el arreglo de datos B que contiene los coeficientes del filtro.

El valor de Wn debe estar en el rango entre 0 y 1, en el cual 1 representa la mitad de la frecuencia de muestreo.

Asignando al valor N el orden de 6 y como frecuencia de corte $Wn = 0.25$ (62.5 Hz) se obtienen los siguientes coeficientes para el filtro FIR pasa bajos:

```
B = [ 0.00848592006158    0.06975525008428    0.24503094177825
      0.35345577615177    0.24503094177825    0.06975525008428
      0.00848592006158]
```

En la figura 5 se ilustra la respuesta en frecuencia del filtro FIR pasa bajos con frecuencia de corte 62.5 Hz (0.25).



**Figura 5. Respuesta en frecuencia del filtro FIR
Con frecuencia de corte $\pi/4$.**

En el programa *diezmado.mat* se encuentra un ejemplo del proceso de diezmado donde se tiene una señal senoidal de 10 Hz y con frecuencia de muestreo $F_s=2\text{KHz}$. En la figura 6 se puede apreciar el resultado de realizar el proceso de diezmado a la señal tomada.

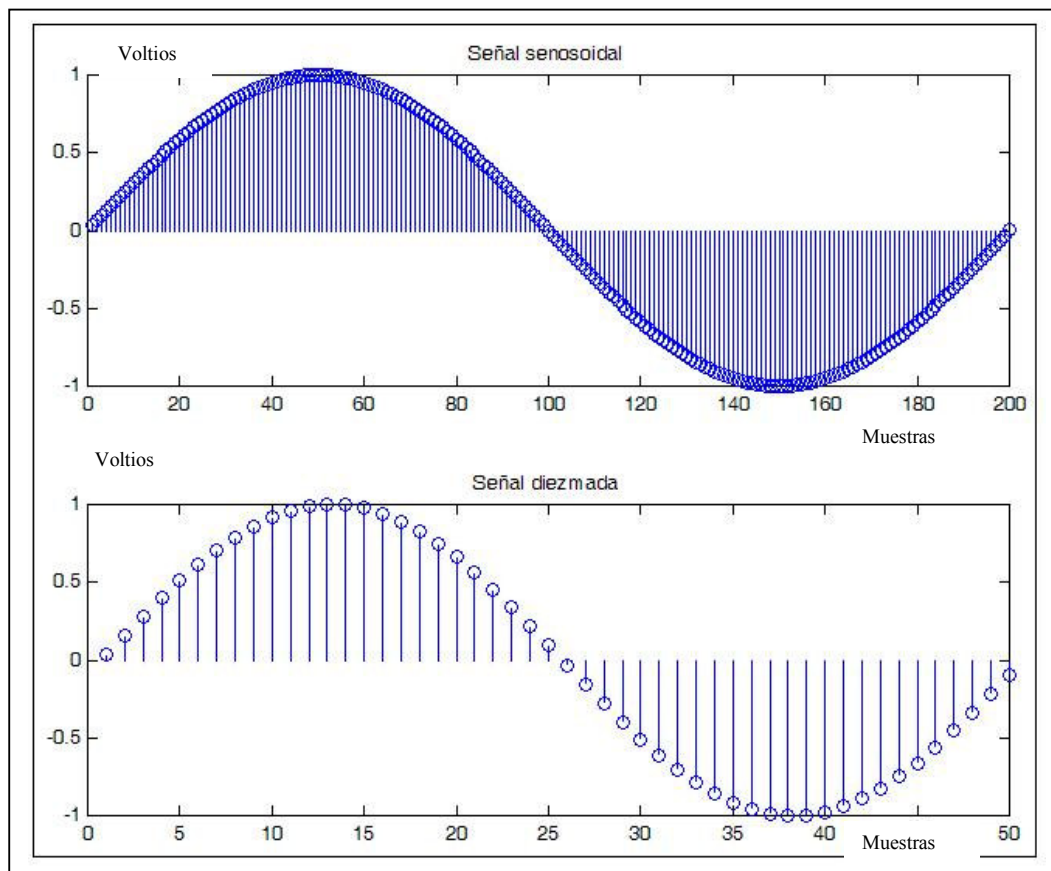


Figura 6. Señal senosoidal diezmada.

1.4 FILTRADO DIGITAL DEL ECG.

El uso de filtros digitales ha sido de gran impacto en aplicaciones médicas. La función de los filtros digitales es la misma de los filtros analógicos, con la diferencia de que su implementación está basada en circuitos lógicos o computadores personales y sus operaciones son realizadas sobre una secuencia de números obtenidos de las muestras tomadas de una señal continua. La precisión de los filtros digitales depende principalmente de los errores de redondeo, el cual es directamente determinado por el número de bits que el diseñador escoge para la representación de las variables en el filtro.[19 y 18]

1.4.1 Filtros suavizadores (*smoothing*)

Uno de los tareas de procesamiento mas comunes aplicados al ECG es el suavizado (*smoothing*) de datos para reducción de ruido de alta frecuencia. Un método para reducir este ruido consiste en realizar un promediado de una cantidad de datos consecutivos. Como ejemplo de un filtro suavizador se puede mencionar el filtro de promediado móvil de Hanning. La función de transferencia del filtro y su versión en el dominio z se pueden ver en la ecuación 1 y 2, respectivamente.

$$y(n) = \frac{1}{4} [x(n) + 2x(n-1) + x(n)]$$

Ecuación 1.

$$H(Z) = \frac{1}{4} [1 + 2Z^{-1} + Z^{-2}]$$

Ecuación 2.

En la figura 7 se ilustra la respuesta en frecuencia del filtro de promediado móvil de Hanning utilizando la función `FREQZ` del software Matlab.(ver programa *prommovil.mat*)

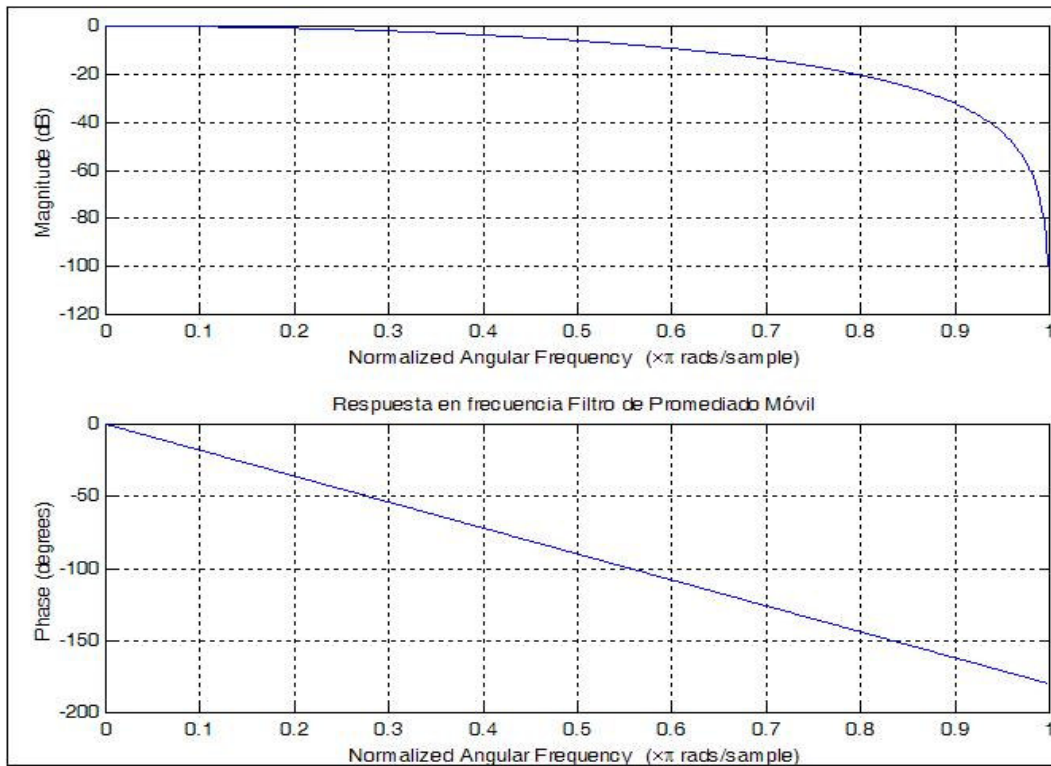


Figura 7. Respuesta en frecuencia del filtro promediado móvil de Hanning.

En la figura 8 se pueden ver un ejemplo de los resultados que se pueden obtener utilizando el filtro de suavizado Hanning aplicado a una señal compuesta por suma de senosoidales de baja frecuencia contaminada con ruido de 110 Hz

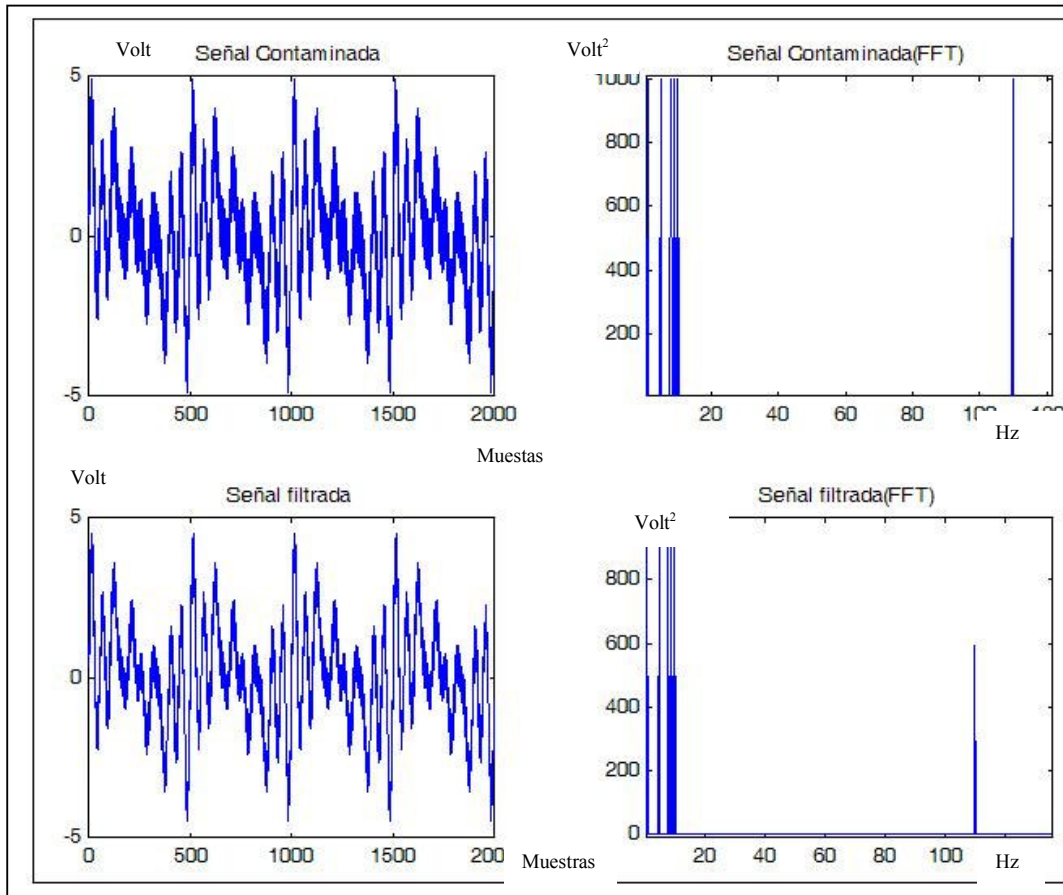


Figura 8. Señal filtrada con Filtro Hanning.

1.4.2 Filtro de muesca (*Notch*)

Un problema común en el procesamiento de señales biomédicas es la remoción del ruido una frecuencia particular o de un rango específico (por ejemplo 60 Hz). Como un simple método para remover completamente una frecuencia específica de una señal se puede poner un cero en el círculo unitario localizado en la correspondiente frecuencia. Por ejemplo, con una frecuencia de muestreo de 500 muestras por segundo, un cero ubicado a $3\pi/25$ remueve el ruido de la línea de 60 Hz. La ecuación en diferencias se muestra en la ecuación 3.

$$y(n) = \left[x(n) - 2 \cos\left(\frac{60}{500} * 2\pi\right) x(n-1) + x(n-2) \right]$$

Ecuación 3.

En la figura 9 se puede ver la respuesta en frecuencia del filtro notch de tercer orden planteado en la ecuación 3 y en la figuras 10 y 11 de un filtro notch de orden 50.

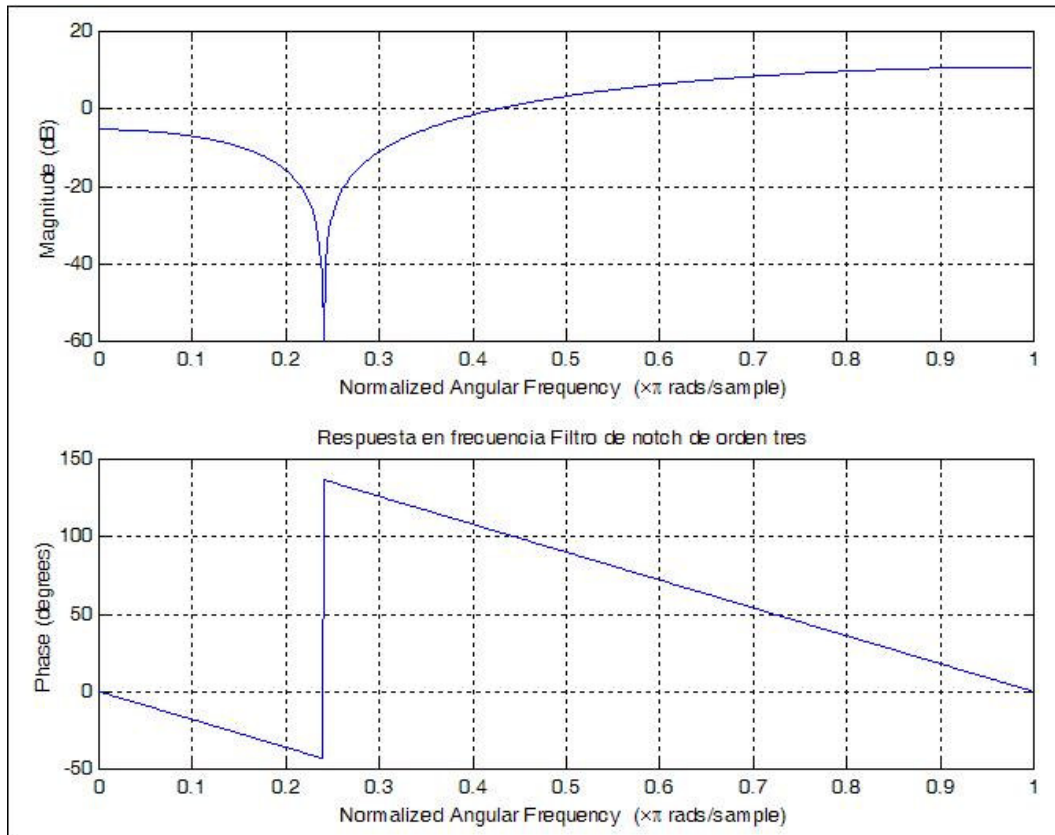


Figura 9. Respuesta en frecuencia Típica del filtro *Notch* (orden 3)

Para una señal ECG tomada de una base de datos, muestreada a una frecuencia de 333.333 Hz y con una interferencia aditiva de una señal senoidal con frecuencia de 60 HZ , se puede diseñar un filtro notch de orden 50 para eliminar esta componente con ayuda del comando `FILTER1` de Matlab (ver programa *filtronotch.mat*).

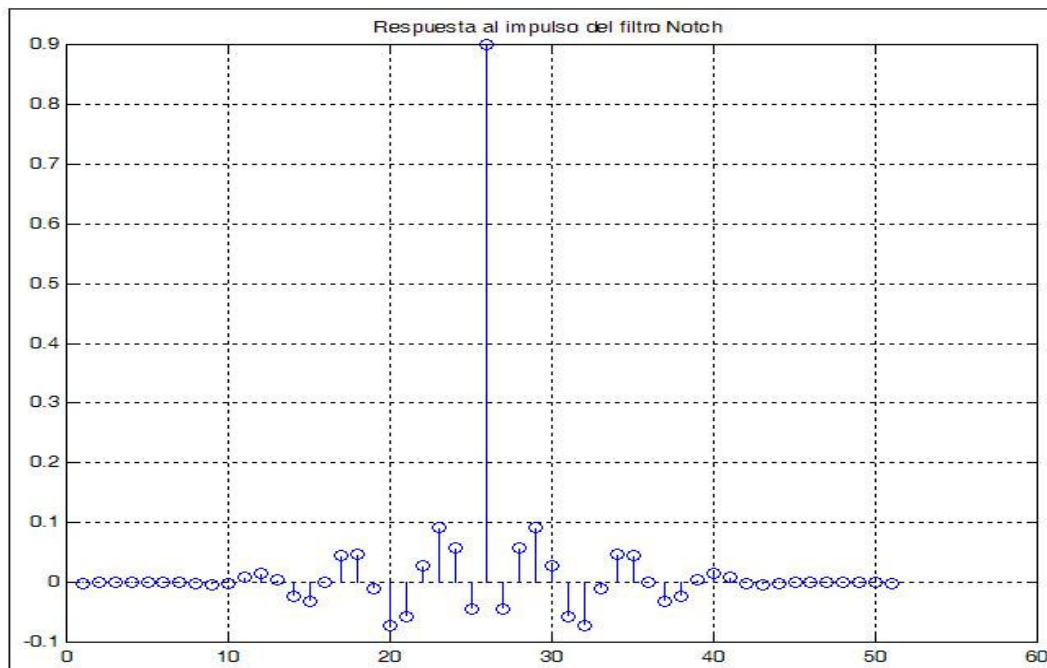


Figura 10. Respuesta al impulso del Filtro *notch* de orden 50.

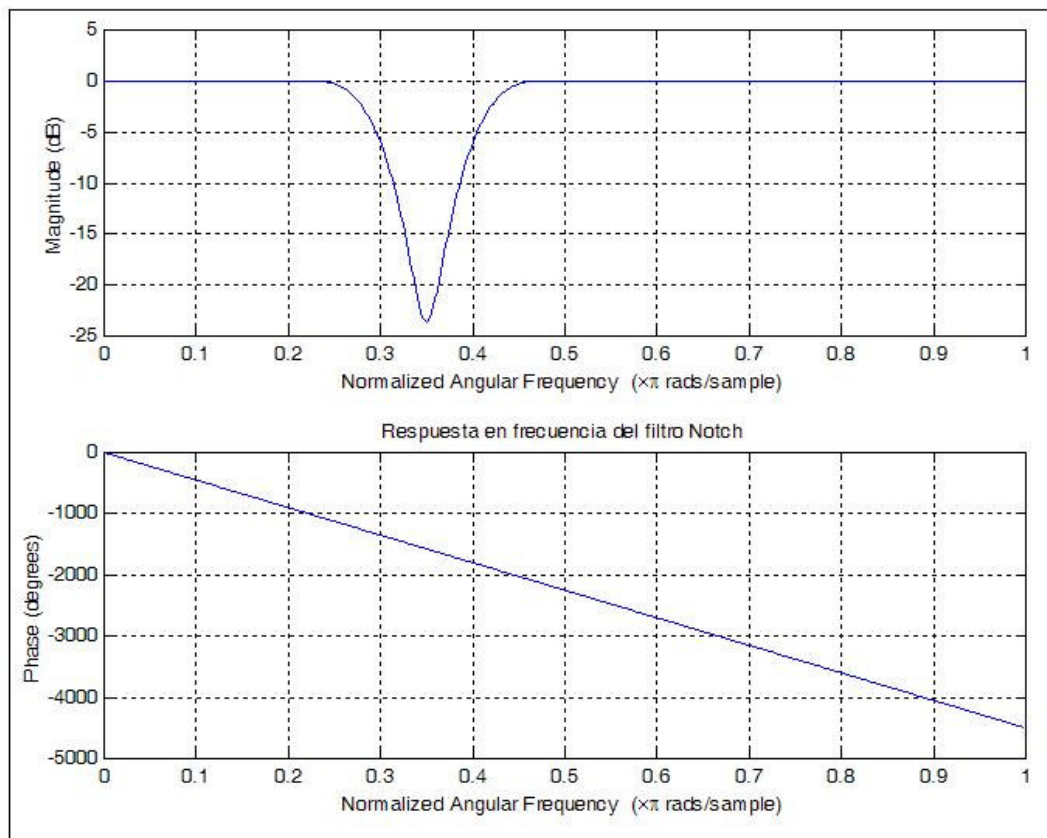


Figura 11. Respuesta en el dominio de la frecuencia del filtro *Notch* de orden 50.

Por lo general un filtro notch puede afectar las frecuencias vecinas al valor del que se desea eliminar. Este problema es tratado en el transcurso de este trabajo de investigación dando como una eficiente solución la implementación de otro tipos de filtrado. En la figura 12 y 13 contienen el espectro de una señal compuesta por la sumatoria de funciones senosoidales con frecuencias de valor cercano a 60Hz y se aprecia el efecto del filtro notch al intentar remover la señal con frecuencia de 60 Hz. Se puede observar gráficamente el efecto de un filtro notch (orden 3 y 50) sobre valores de frecuencias cercanos al que se pretende eliminar. El programa *efectofiltronotch.mat* contiene una demostración práctica de este efecto. [20]

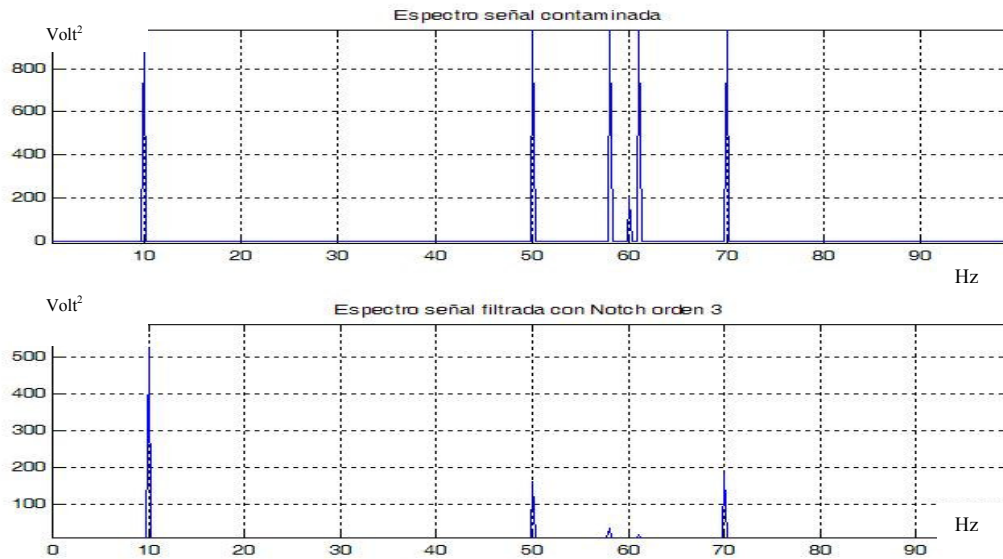


Figura 12. Efecto Filtrado *Notch* orden 3.

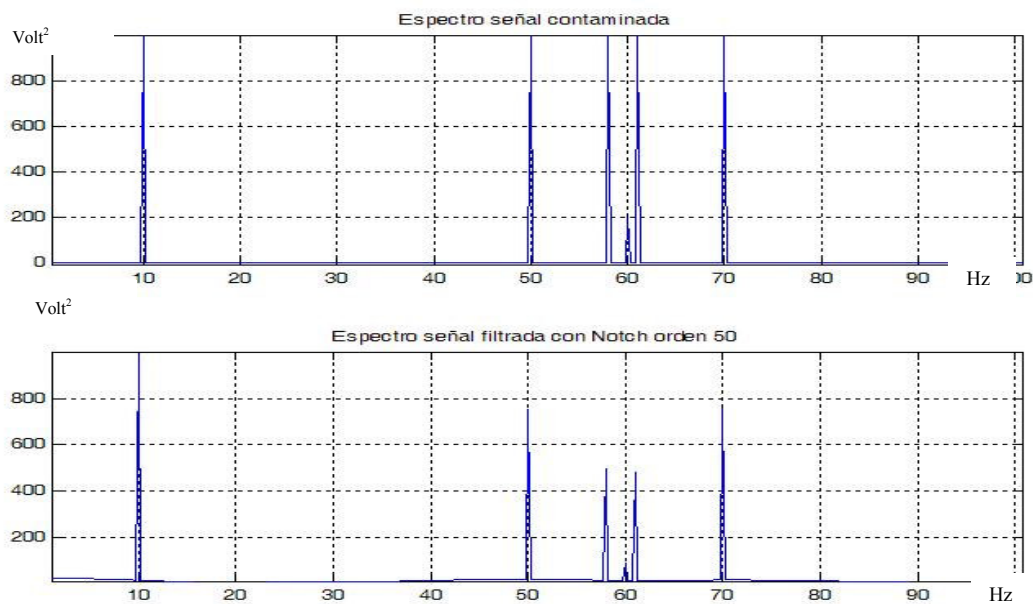


Figura 13. Efecto Filtrado *Notch* orden 50.

1.4.3 Filtros derivadores

La respuesta de una función derivadora, utilizada comúnmente para remover el corrimiento de la línea de base del electrocardiograma, se incrementa linealmente con la frecuencia. Estos métodos discutidos a continuación, pueden ofrecer, en vez de complejidad de cálculos, una aproximación a una verdadera derivación. Como ilustración práctica se recomienda observar el archivo *derivadores.mat* donde se encuentran los códigos para la implementación de estos filtros.

1.4.3.1 Diferencia de dos puntos

El algoritmo de la diferencia de dos puntos es el más sencillo de los algoritmos de derivación. Coloca un cero en $Z = 1$ sobre el círculo unitario. La magnitud de su respuesta se aproxima mucho a una respuesta ideal, pero desde que no vaya a cero en $f_s/2$ (ver figura 14). Su ecuación de diferencia y su función de transferencia se describen en la ecuación 4.

$$y(n) = \frac{1}{T} [x(n) - x(n-1)]$$
$$H(Z) = \frac{1}{T} [1 - Z^{-1}]$$

Ecuación 4.

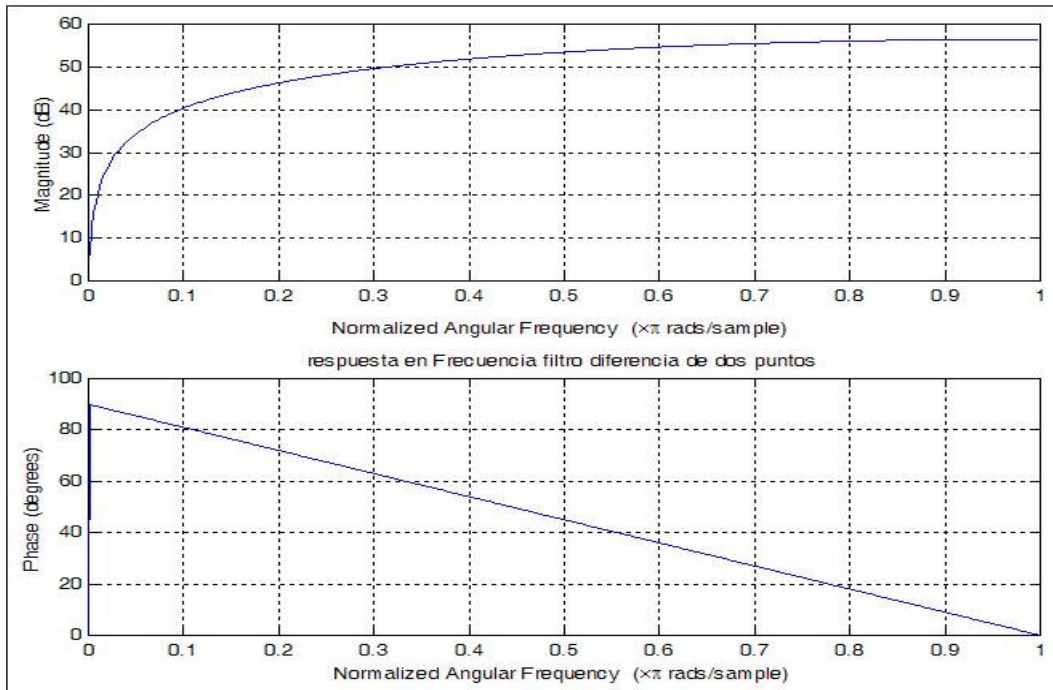


Figura 14. Magnitud y fase del derivador de dos puntos.

En la figura 15 se ilustra como ejemplo el resultado que se puede obtener de aplicar este filtro a una señal compuesta por una suma de senosoidales

contaminadas con una interferencia de baja frecuencia y el efecto sobre componentes de frecuencias cercanas a la interferencia. Se puede apreciar que las componentes espectrales de la señal con valores cercanos a la frecuencia de la interferencia son afectadas considerablemente causando serias distorsiones a la señal procesada.

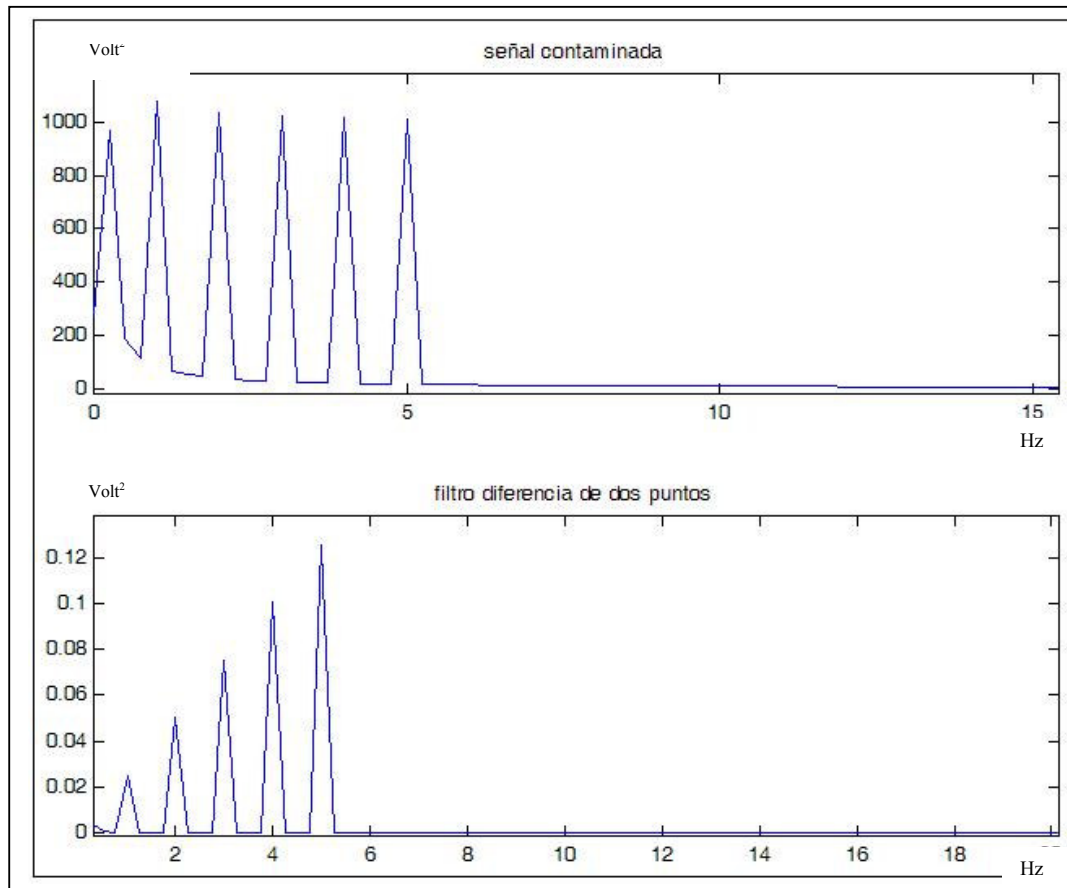


Figura 15. Filtrado con diferenciador de dos puntos.

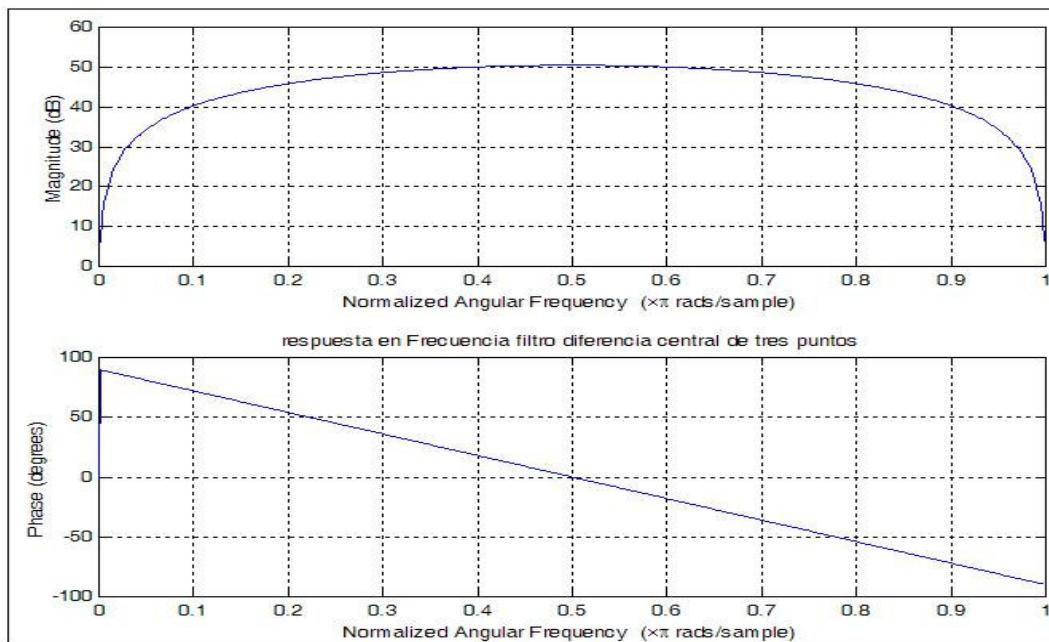
1.4.3.2 Diferencia central de tres puntos.

El algoritmo del diferenciador central de tres puntos coloca ceros en $Z=1$ y $Z=-1$, así que la aproximación de la derivada es pobre por encima de $f_s/10$ (ver figura 16). Por lo tanto, puesto que su respuesta va a cero en $f_s/2$, el filtro tiene un poco de efecto suavizador. Su ecuación de diferencias y su función de transferencia se muestra en la ecuación 5.

$$y(n) = \frac{1}{2T} [x(nT) - x(nT - 2T)] \quad \text{Hz}$$

$$H(Z) = \frac{1}{2T} [1 - Z^{-2}]$$

Ecuación 5.



**Figura 16. Respuesta en frecuencia
del filtro diferencia central de tres puntos**

En la figura 17 se puede apreciar nuevamente el resultado que se obtiene al tener como señal de entrada del filtro diferenciador central de tres puntos, a una señal compuesta por una suma de senosoidales contaminadas con una interferencia de baja frecuencia y el efecto sobre componentes cercanas, las cuales son alteradas.

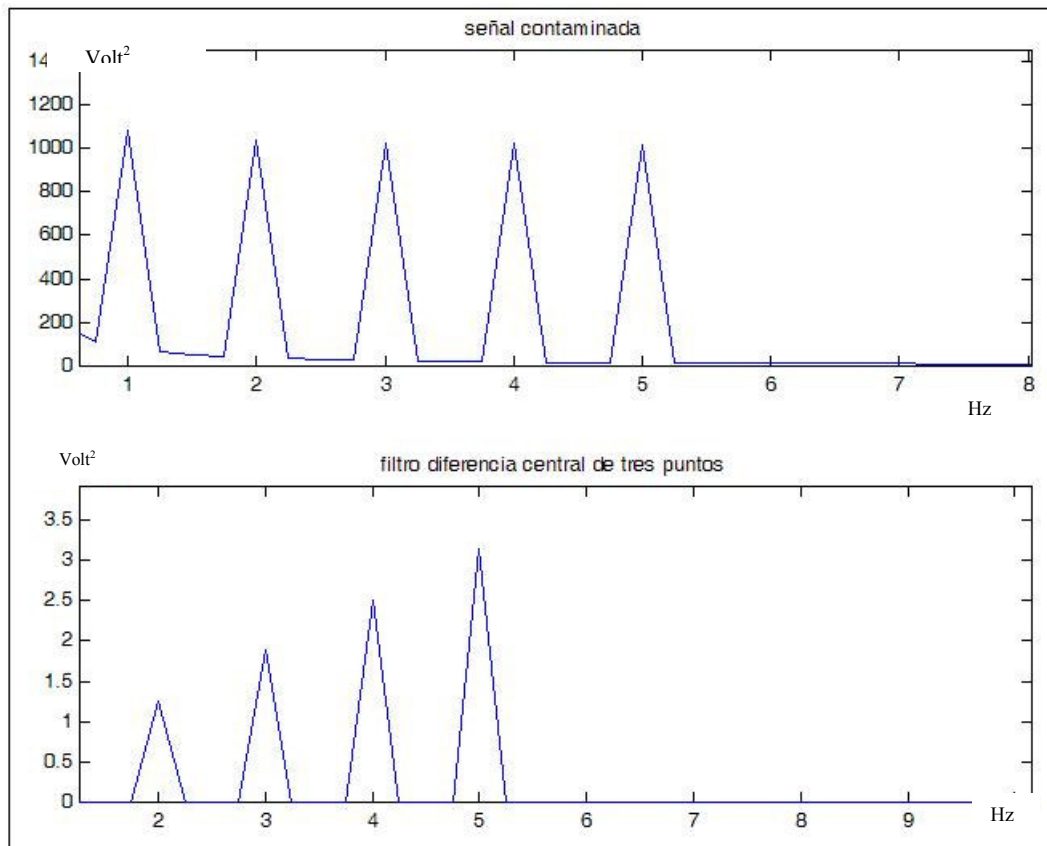


Figura 17. Filtrado con diferenciador central de tres puntos.

1.4.3.3 Segunda derivada.

En la ecuación 6 se puede apreciar la ecuación en diferencia y la función de transferencia con la cual se puede hacer un algoritmo que permite hacer una aproximación de la segunda derivada.

$$y(n) = x(nT) - 2x(nT - 2) + x(nT - 4)$$

$$H(Z) = (1 - 2Z^{-2} - Z^4)$$

Ecuación 6.

En la figura 18 se puede ver la respuesta en frecuencia del filtro de segunda derivada.

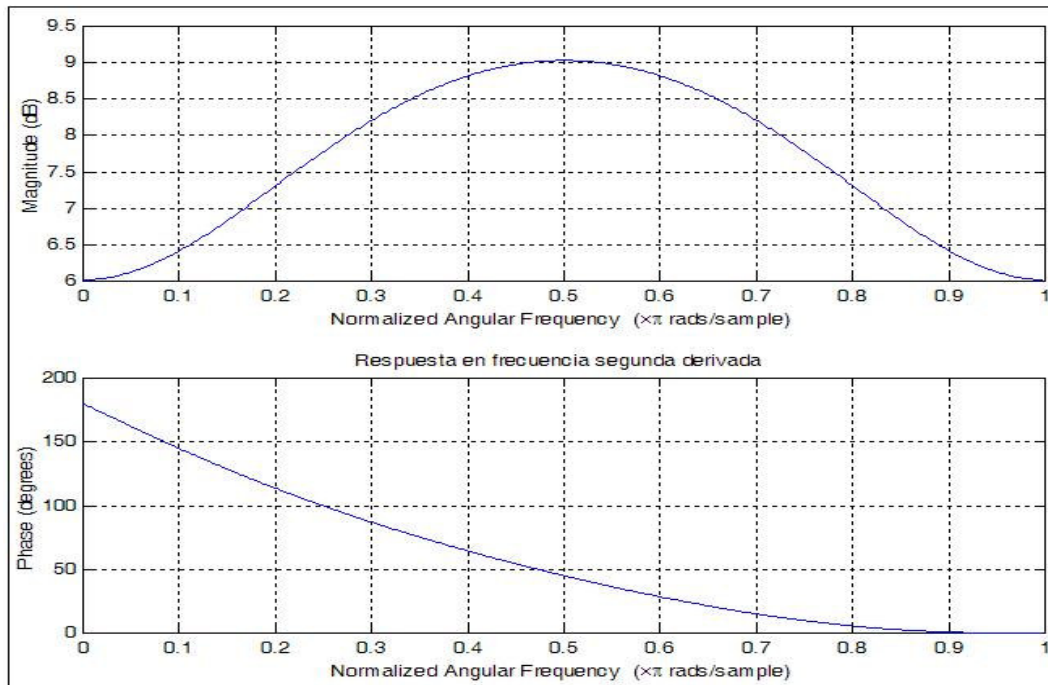


Figura 18. Respuesta en frecuencia para filtro de segunda derivada.

En la figura 19 se ilustra como ejemplo el efecto que provoca el filtro de segunda derivada a una señal compuesta por una suma de senosoidales contaminadas con una interferencia de baja frecuencia.[20]

Teniendo en cuenta los resultados observados del efecto de los filtros derivadores sobre las componentes cercanas a la frecuencia de la interferencia, se propone en este proyecto de investigación plantear una estrategia para eliminarlas sin causar distorsiones en la señal electrocardiográfica. Esta Estrategia está basada en el diseño e implementación de un sistema de filtrado adaptativo que permite la remoción del corrimiento de la línea de base sin causar distorsiones a la señal bajo estudio.

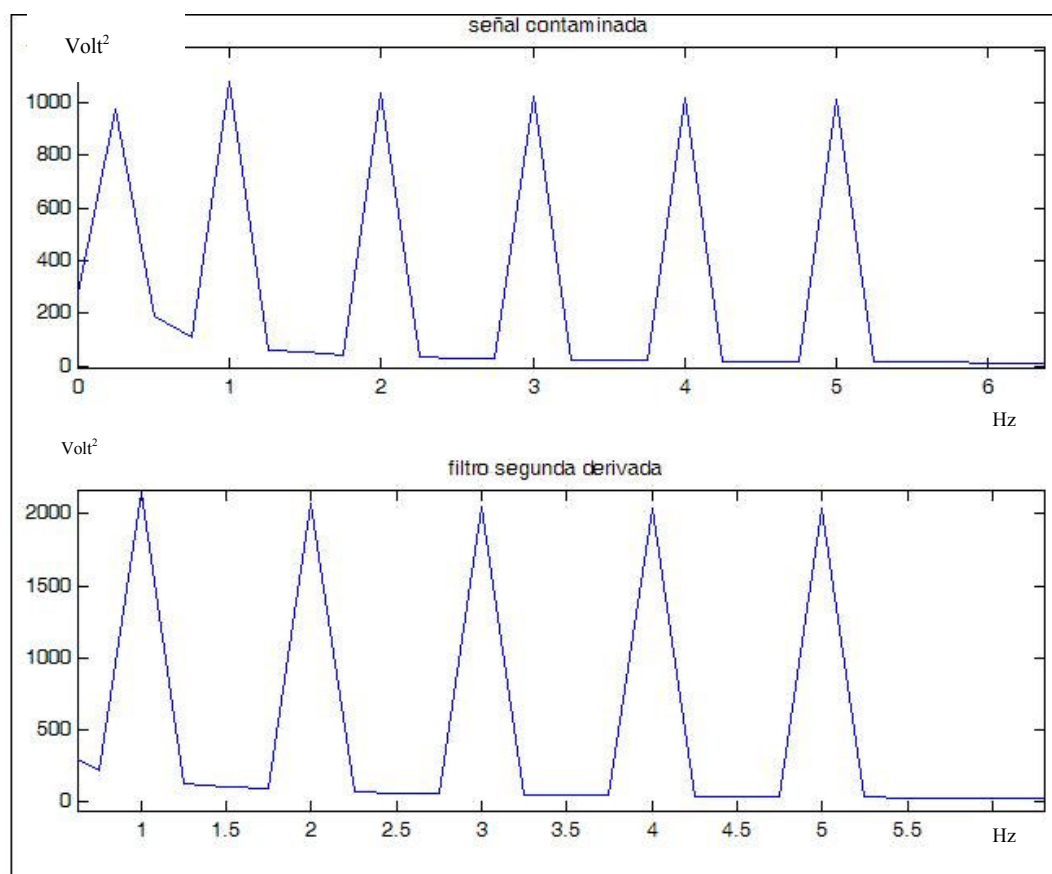


Figura 19. Filtrado basado en la segunda derivada.

2. FILTRADO ADAPTATIVO

2.1 PRINCIPIOS BÁSICOS DEL FILTRADO ADAPTATIVO

El filtrado adaptativo es utilizado cuando se desea eliminar una interferencia dentro de una señal y para lograrlo es necesario utilizar un sistema cuyos coeficientes deben cambiar con el tiempo.

La estructura básica de un filtro adaptativo aparece ilustrada en la figura 20. La señal de entrada $x(n)$ es filtrada para producir una salida $y(n)$ que es observada por un bloque que evalúa la señal de la salida con respecto a una señal deseada $d(n)$ y cualquier diferencia entre las dos constituye una señal de error $e(n)$ que es luego llevada a otra etapa donde es usada para decidir como modificar los parámetros del filtro con el objetivo de eliminar el error. Si las características de la señal de de entrada cambia con el tiempo, este lazo de ajuste debe recalcular los parámetros del filtro.

Los bloques funcionales en la figura 20 son muy generalizados y pueden ser escogidos de diferentes formas para resolver problemas prácticos. El filtro puede ser implementado de manera analógica o digital y con la posibilidad de poseer una línea de demora, ceros y polos u otras estructuras.

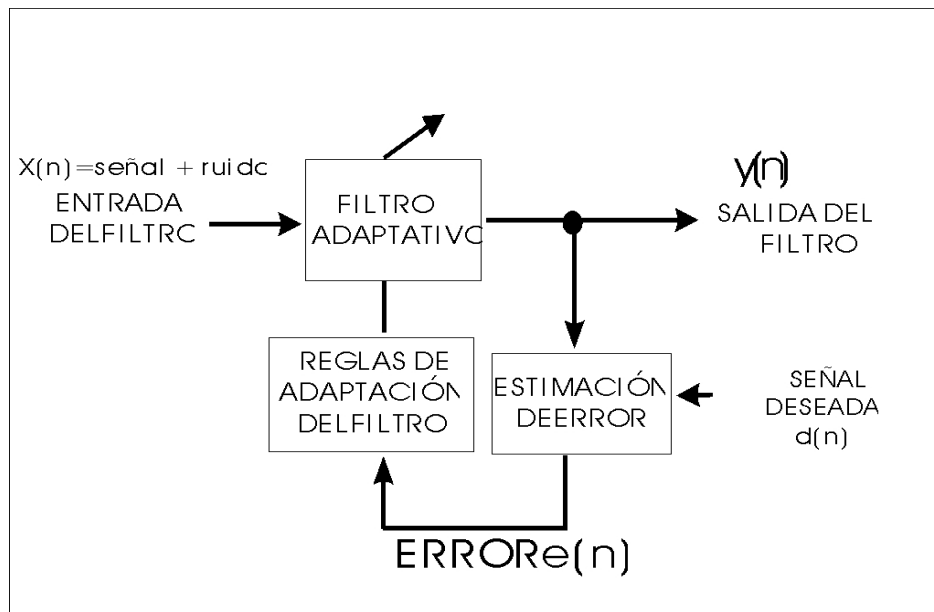


Figura 20. Estructura general de Un filtro adaptativo.

En la figura 21 se tiene un filtro discreto en el tiempo con respuesta finita al impulso (FIR), basado en una línea de demora y un conjunto de N coeficientes

w_0 hasta w_{N-1} de ganancia ajustable. La salida del filtro $y(n)$ se compone de la suma de las entradas retrasadas y escaladas dados por la ecuación 7.

$$y(n) = \sum_{i=0}^{N-1} w_i x(n-i)$$

Ecuación 7.

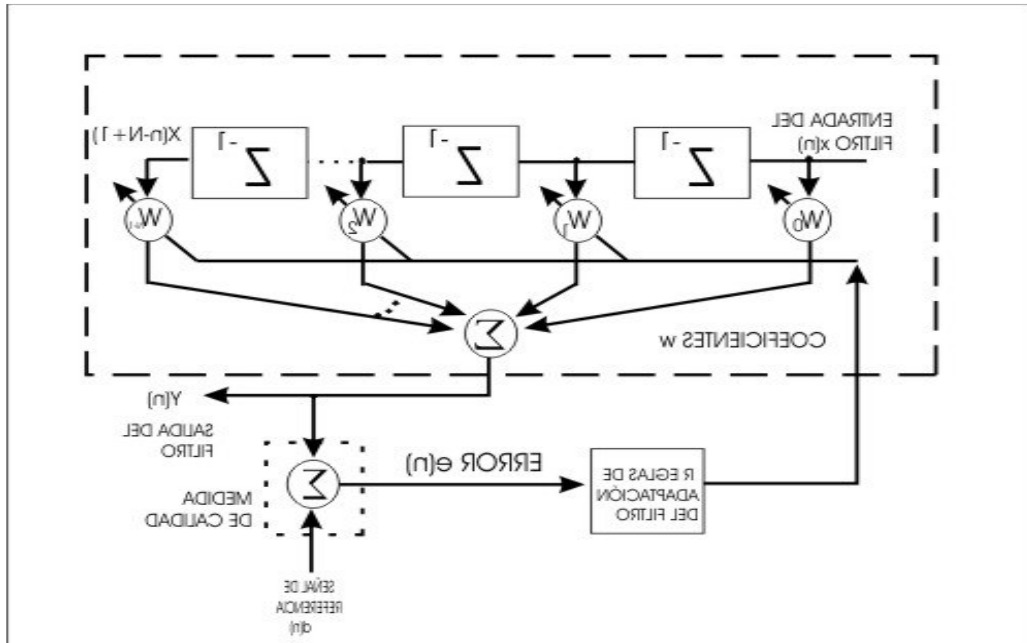


Figura 21 . Filtro Adaptativo FIR.

Por conveniencia, $y(n)$ también puede ser escrito como el producto punto (o interno) de dos vectores. (Ver ecuación 8)

$$y(n) = X^t(n)W$$

Ecuación 8.

Donde :

$$W = [w_0 \ w_1 \ w_2 \ ... \ w_{N-1}]^t$$

$$X(n) = [x(n) \ x(n-1) \ x(n-2) \ \ x(n-N+1)]^t$$

Esta estructura de filtro digital es una de las más simples. Este filtro es estable (para cada W limitado) y la respuesta al impulso es dada directamente por los elementos del vector W . [21]

Para el diseño de un filtro adaptativo FIR se debe asumir que un conjunto A de N coeficientes W_i ($0 \leq i \leq N-1$) se debe determinar tal que para un conjunto $x(n)$

dado, la desviación o error $e(n)$ entre el conjunto $y(n)$ y el conjunto $d(n)$ sea mínimo.

Usando el criterio de mínimos cuadrados sobre el dominio de los índices $(0,1,\dots,N_0-1)$ con $N_0 > N$, la función de error $E(A)$ es minimizada y se puede expresar como se ve en la ecuación 9.

$$E(A) = \frac{1}{N_0} \sum_{n=0}^{N_0-1} [d(n) - y(n)]^2$$

Ecuación 9.

De la definición de $E(A)$ y de $y(n)$, se puede minimizar el error haciendo que:

$$\frac{\partial E}{\partial W_i} = 0 \text{ para } 0 \leq i \leq N-1 \text{ (ver ecuación 10)}$$

$$\frac{\partial E}{\partial W_i} = -\frac{2}{N_0} \sum_{n=0}^{N_0-1} x(n-i)[d(n) - y(n)]; \quad 0 \leq i \leq N-1$$

Ecuación 10.

Por medio del algoritmo del filtro adaptativo LMS (*Least-mean-square*), basado en la minimización del error cuadrático medio expresado en la ecuaciones 9 y 10, se puede dar la estrategia planteada en la ecuación 11 para la actualización de los coeficientes W del filtro.

$$w_i(n+1) = w_i(n) + \mu e(n) X(n)$$

$$e(n) = d(n) - W(n)X(n)$$

Ecuación 11.

Donde $W(n) = [w_0(n) \ w_1(n) \ \dots \ w_{L-1}(n)]$ es el vector de coeficientes, $X(n) = [x(n) \ x(n-1) \ \dots \ x(n-L+1)]$ es el vector de la señal de entrada, $d(n)$ es la señal de referencia, $e(n)$ es la señal de error y μ es el tamaño del paso utilizado en las iteraciones. El filtro adaptativo LMS es muy popular porque es relativamente fácil de implementar en software debido a su simplicidad computacional y el uso eficiente de la memoria. Además, se destaca por su robusto funcionamiento en la presencia de errores numéricos causados por la aritmética finita. Por último, su comportamiento ha sido analizado al punto donde un usuario puede fácilmente estructurar el sistema con el adecuado desempeño con solo un limitado conocimiento de las señales de entrada y la señal deseada.[22]

2.2 FILTRO DIGITAL NOTCH ADAPTATIVO DISEÑADO PARA REMOCIÓN DEL RUIDO DE LA LÍNEA DE POTENCIA DE SEÑALES BIOMÉDICAS.

La conveniente eliminación del ruido de la línea de potencia contenido en las señales analizadas por equipos biomédicos ha sido el centro de atención de investigadores en los últimos años. En las primeras aplicaciones, esta interferencia ha sido removida por medio de filtros rechaza banda (*notch*) con los coeficientes apropiados. Aunque estos métodos pueden ser aceptables para algunas aplicaciones, no se considera una solución universal para el monitoreo de la electrofisiología. Un filtro rechaza banda (*notch*) no es adecuado así la frecuencia de la línea de poder sea estable (cabe mencionar que no siempre posee el valor exacto de los 60 Hz). Esta variación de frecuencia, comúnmente, es asumida como Gausiana. Esto sugiere que la distribución puede ser o no centrada en los 60 Hz y la desviación de la frecuencia no siempre se encuentra en una sola dirección.

Para obtener una mejor respuesta de un filtro notch que tenga un óptimo ancho de banda y con la capacidad de responder a los continuos cambios en el valor de la frecuencia central del ruido de 60 Hz, se propone en este proyecto de investigación un proceso de filtrado notch basado en un filtro FIR adaptativo de segundo orden que solo requiere la adaptación de solo un coeficiente. Con este sistema se espera reducir los requerimientos de cálculos en tiempo real y consumo de memoria durante el proceso de adaptación.

2.2.1 Diseño de un filtro FIR notch adaptativo de segundo orden.

El método de colocación de ceros es el de mayor ventaja para el diseño un filtro FIR, los ceros del filtro son colocados sobre el círculo unitario a una posición equivalente a la frecuencia rechazada ω_0 . Para una señal muestreada a una frecuencia ω_s , los ceros son determinados por la ecuación 12:

$$z_{1,2} = \cos(\beta_0) \pm j\sin(\beta_0)$$

Ecuación 12.

Donde el ángulo β_0 es dado por la ecuación 13:

$$\beta_0 = \left[\frac{\omega_0}{\omega_s} \right] 2\pi = \omega_0 \Delta T$$

Ecuación 13.

La función de transferencia del filtro FIR de segundo orden es escrita como se ve en la ecuación 14:

$$H(Z) = (z - z_1)(z - z_2)$$

Ecuación 14.

Por comparación con la forma canónica de un filtro FIR de segundo orden, los coeficientes pueden ser identificados según la ecuación 15.

$$w_0 = 1, w_1 = -2\cos(\beta_0), w_2 = 1.$$

Ecuación 15.

La formula no recursiva en el dominio del tiempo queda expresada en la ecuación 16.

$$y(n) = w_0x(n) + w_1x(n-1) + w_2x(n-2)$$

Ecuación 16.

Como la frecuencia ω se aproxima a la frecuencia del ruido ω_0 , la magnitud de la función de transferencia del filtro evaluada en este punto puede ser aproximada por la expresión contenida en la ecuación 17.

$$|H(Z)| \approx 2\sqrt{2}|\sin(\beta_0)|[1 - \cos(\beta_0 - \omega\Delta T)]^{\frac{1}{2}}$$

Ecuación 17.

El ancho de banda y el factor de calidad pueden ser calculados por las ecuaciones 18 y 19

$$BW = \frac{1}{\sqrt{2}|\sin(\beta_0)|} \text{ rad}$$

Ecuación 18.

$$Q = \frac{\text{frecuencia central}}{\text{ancho de banda}} = \omega_0 \sqrt{2}|\sin(\beta_0)|$$

Ecuación 19

Si el ruido no está centrado alrededor de la frecuencia estimada, se hace necesario realizar una variación en la ecuación 15 remplazando ω_0 por $\omega_0 \pm \Delta\omega$. Donde $\Delta\omega$ representa la variación de la frecuencia central de la interferencia. (Ver ecuación 20).

$$\beta = \left[\frac{\omega_0 \pm \Delta\omega}{\omega_s} \right] 2\pi = \left[\frac{\omega_0}{\omega_s} \right] 2\pi \pm \left[\frac{\Delta\omega}{\omega_s} \right] 2\pi = \beta_0 \pm \Delta\beta$$

Ecuación 20

Los coeficientes del filtro se calculan según la ecuación 21

$$w_0 = 1, w_1 = -2\cos(\beta_0 \pm \Delta\beta), w_2 = 1.$$

Ecuación 21

El coeficiente a_1 es el único que es relacionado directamente con las variaciones de frecuencia rechazada. En la figura 22 se puede ver el esquema de un filtro digital FIR notch de segundo orden y un filtro FIR notch de segundo orden adaptativo, donde se ilustra la presencia de un algoritmo de adaptación que actualiza el valor del coeficiente w_1 según las expresiones contenidas en la ecuación 11 descrita en la sección 2.1.

En el programa *filtronotchcerosfir.mat* se encuentra un ejemplo práctico del diseño de un filtro FIR notch de segundo orden aplicando el método de colocación de ceros con un valor de $BW=1,039$ rad y $Q=3,649635$. En este programa se puede ver que el filtro notch puede afectar las componentes cercanas a la frecuencia de 60 Hz.

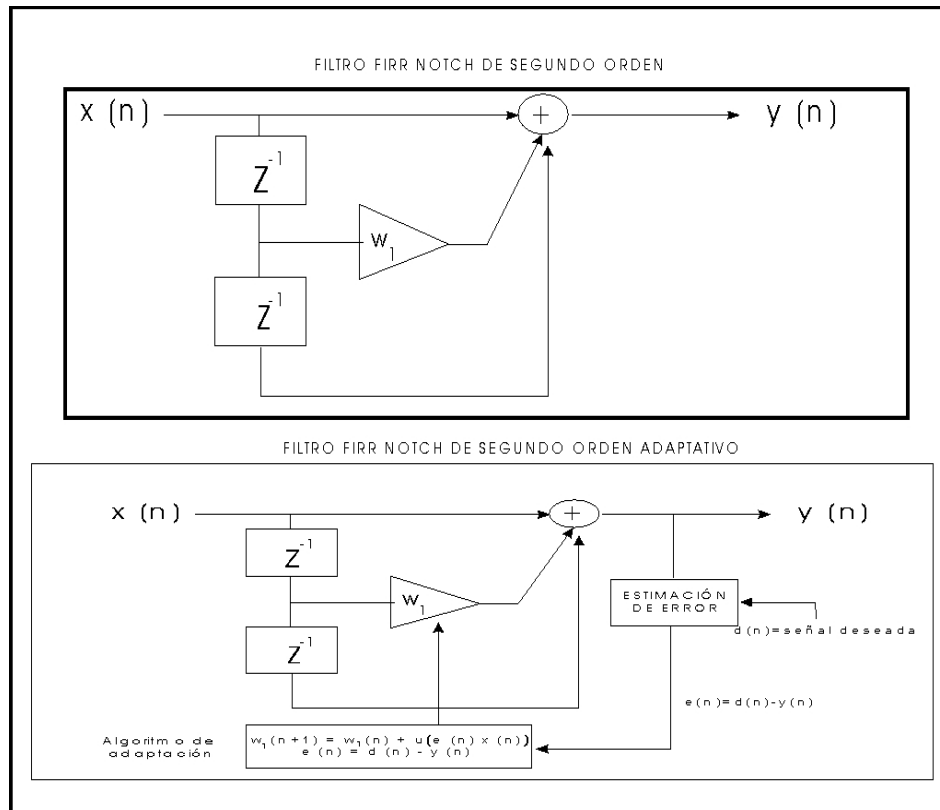


Figura 22. Diagrama de bloques de un filtro FIR notch (Clásico y adaptativo).

2.2.2 Sistema de Cancelación de ruido.

En muchos casos de diseños de filtros adaptativos, como los utilizados para el tratamiento de la señal electrocardiográfica, la señal deseada $d(n)$ no es conocida, por lo cual es necesario utilizar una estrategia en esta investigación que consiste en la implementación de un sistema adaptativo de cancelación de ruido, con base a una señal de referencia que es una función matemática que estima la clase de interferencia que afecta la señal.

En la figura 23 se puede ver el diagrama de bloques del sistema de cancelación, en el cual se asume como entrada al sistema la señal $X(n)$ con un ruido aditivo $V1(n)$ (ver ecuación 22). La función estimadora $G(z)$ tiene como labor tomar un señal de referencia $V2(n)$ y filtrarla para luego restarla el resultado a la señal $X(n)$ (ecuación 23 y 24).

Como señal de referencia $V2(n)$ se han propuesto diferentes modelos basados en una función senoidal con una frecuencia central de 60 Hz. La estimación de la señal $V2(n)$ se discutirá en la sección 2.2.3 .

En la ecuación 25 se aprecia que la señal $e(n)$ se puede expresar en función de la señal de entrada compuesta por las señales $x(n)$ y $V1(n)$. Esto lleva a la conclusión de que entre mas se aproxime la señal $z(n)$ a la interferencia $V1(n)$ se obtendrá a la salida del sistema por cancelación una señal $e(n)$ muy cercana a la señal $x(n)$

La función estimadora $G(z)$ (ver ecuación 26), se deriva del diseño del filtro notch descrito en la sección anterior, en la cual se encuentra el coeficiente w_1 , calculado anteriormente.

El valor del coeficiente W_1 de la función $G(z)$ cambia su valor a través del tiempo por medio de un algoritmo de adaptación en función de la señal de error $e(n)$ y del coeficiente de adaptación μ (ver ecuación 11).[23]

$$\text{Señal de entrada} = x(n) + V1(n)$$

Ecuación 22

$$Z(n) = V2'(n) * G(n)$$

Ecuación 23

$$e(n) = \text{señal de entrada} - Z(n)$$

Ecuación 24

$$e(n) = x(n) + V1(n) - Z(n)$$

Ecuación 25

$$G(z) = -W_1 z^{-1} - 1 z^2$$

Ecuación 26

Un ejemplo práctico para estudiar el funcionamiento del sistema adaptativo de cancelación se ha implementado en el programa *fircancelacion.mat*, en el cual

se toma una señal con interferencia de 60 Hz con una frecuencia de muestreo $f_s=500$ Hz compuesta por la suma de funciones senosoidales de frecuencias 10, 50, 58, 61 y 70 Hz . En este programa se demuestra lo conveniente de tener una señal de referencia V2 muy similar a la interferencia V1.

En la figura 24 se puede apreciar el algoritmo planteado en el programa *fircancelacion.mat* para la estimación de la señal de error por medio del sistema de cancelación de ruido. La figura 25 ilustra el espectro en frecuencia de la señal de entrada. En la figura 26 se ilustra la magnitud del espectro en potencia de la señal de entrada contaminada con la interferencia de 60 Hz y en la figura 27 el resultado obtenido a la salida del sistema que cumple con la relación planteada en la ecuación 26 donde la interferencia es de valor muy cercano a la señal V2(n).

En la figura 27 también se realiza la comparación con el resultado obtenido por el filtro notch FIR convencional con orden 50, en el cual se aprecia la eliminación de la componente de 60 Hz pero afectando la magnitud de las otras componentes cercanas a 60 Hz.

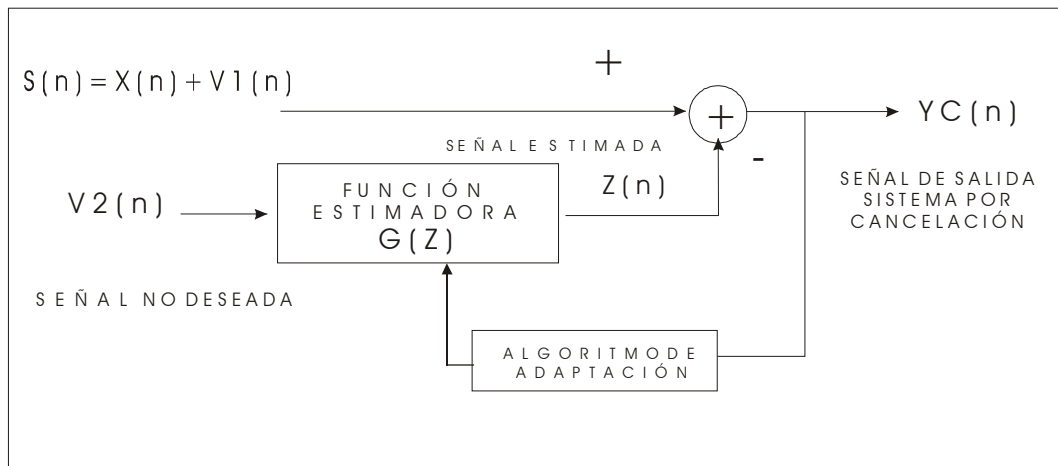


Figura 23. Sistema adaptativo de cancelación de ruido.

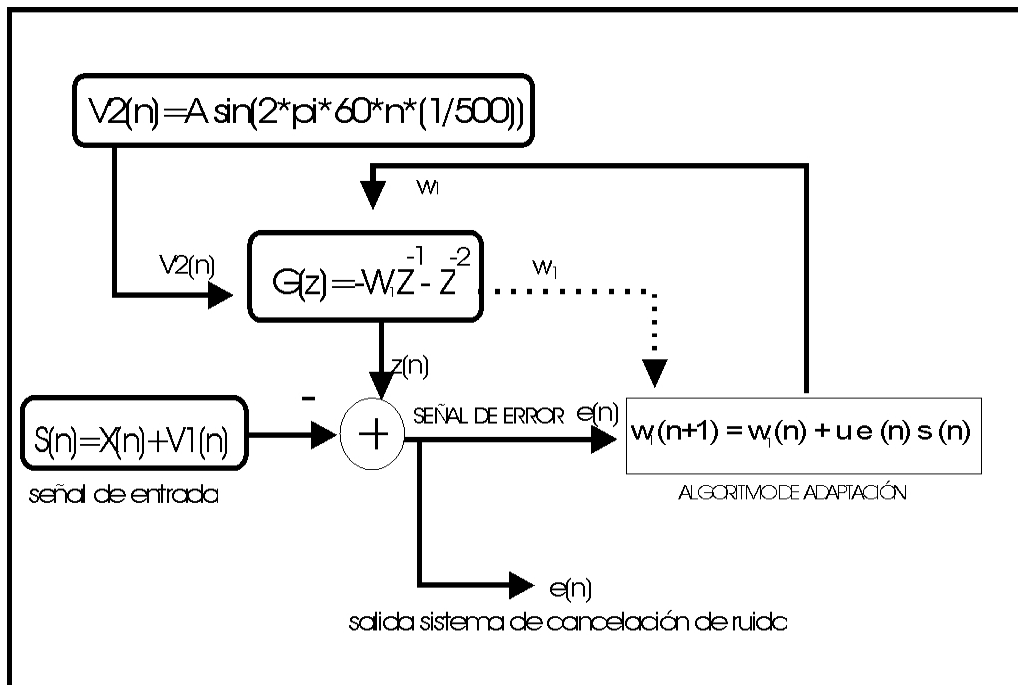


Figura 24. Algoritmo del sistema de cancelación de ruido para estimar la señal de error.

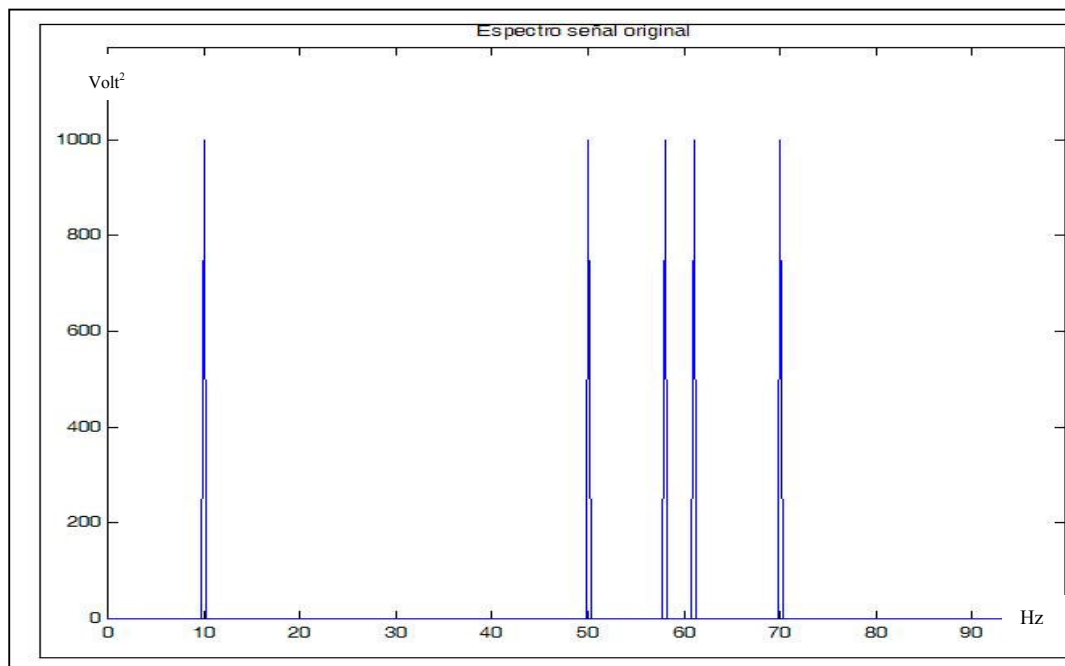


Figura 25. Espectro de la señal de entrada.

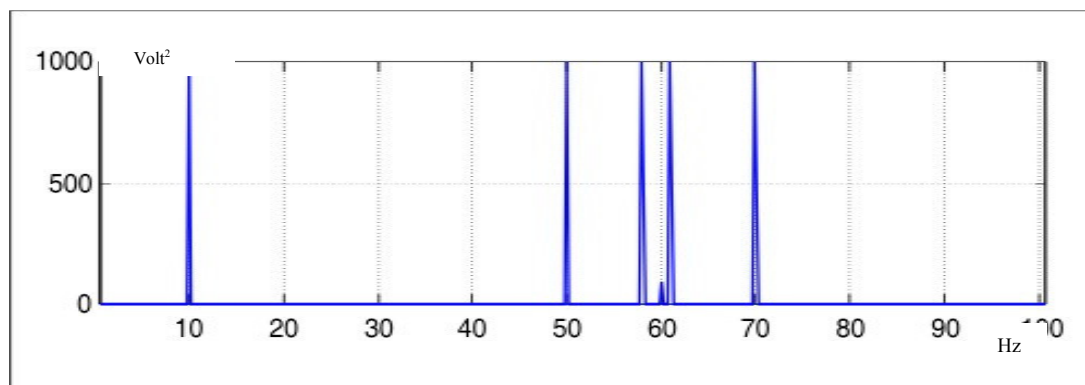


Figura 26. Señales senosoidales con componente de 60 Hz.

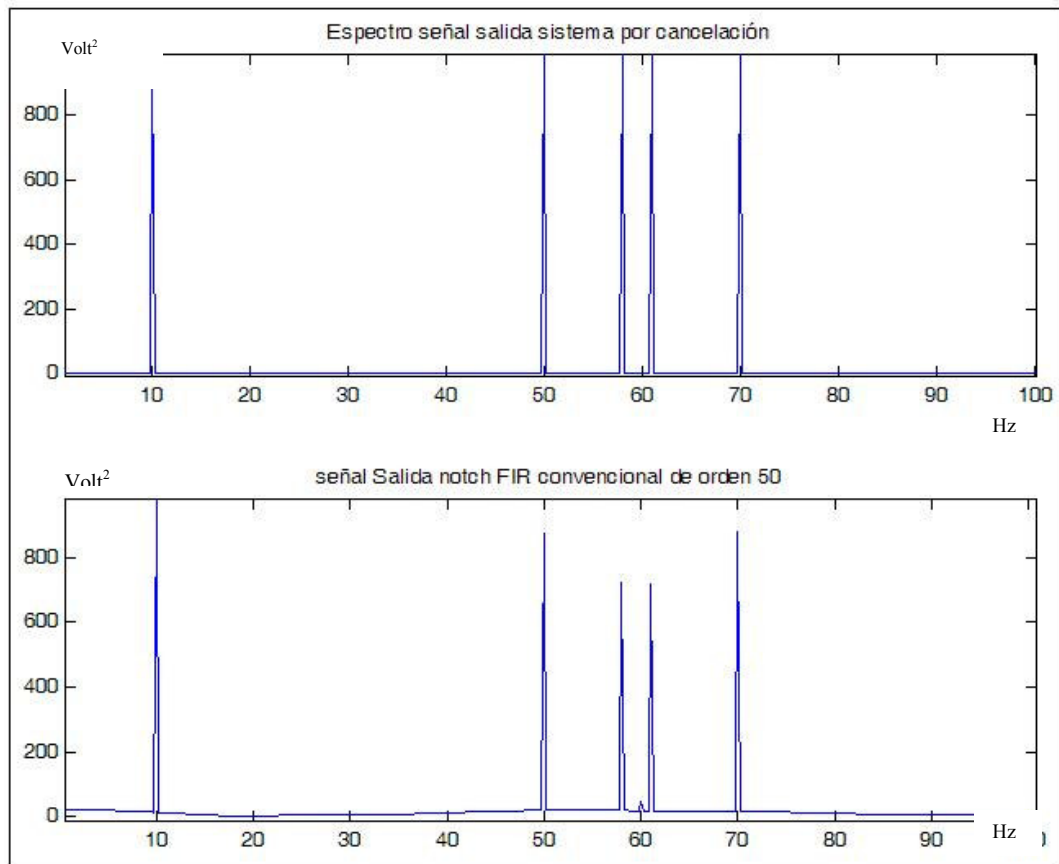


Figura 27. Resultados obtenidos con filtrado el sistema de cancelación y con un filtro *nocth* FIR.

2.2.3 Sistema para estimación de la señal de referencia V2(n).

Como método propuesto en este proyecto de investigación para crear la función estimada V2, se realiza la estimación espectral de la componente que se desea eliminar el cual ofrece la ventaja de tener un valor aproximado de la magnitud de la interferencia que afecta a la señal electrocardiográfica.

Por medio del espectro de potencia de una señal es posible poder calcular el coeficiente de Fourier para una determinada componente en frecuencia gracias a la relación existente con los valores extraídos de la Transformada Discreta de Fourier (DFT).

En la ecuación 27 se proporciona la expresión utilizada en este proyecto de investigación para obtener los coeficientes de la serie de Fourier y por medio de la ecuación 28 se puede calcular la DFT. Teniendo en cuenta un número de muestras N=2000 y una frecuencia de muestreo de 500 Hz se obtiene una resolución de 20.25 Hz.

$$a_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-jk(2\pi/N)n}$$

Ecuación 27.

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\Omega n}$$

Ecuación 28.

Con base en las ecuaciones 27 y 28 se puede ver que los coeficientes a_k son proporcionales a muestras igualmente espaciadas de la función utilizada para calcular la DFT, donde Ω se utiliza para denotar el espaciado de las muestras $2\pi/N$. Por lo cual se puede utilizar la ecuación 29 para el cálculo de los coeficientes de Fourier.

$$a_k = \frac{1}{N} X(k\Omega)$$

Ecuación 29.

Según la teoría de las series de Fourier se puede hacer la suposición de que una señal se puede representar como suma de un conjunto de señales exponenciales complejas y por medio de los coeficientes de Fourier se puede determinar el aporte a la señal de cada componente en frecuencia.

Como método alternativo de estimar la amplitud de la señal de referencia V2 es posible calcular el coeficiente de Fourier para la componente de 60 Hz el cual dará una aproximación de la amplitud y la fase de la interferencia aditiva que contamina a la señal bajo estudio.

En la ecuación 30 se detalla la expresión para el cálculo del coeficiente para la componente de 60 Hz. Donde f_s es la frecuencia de muestreo, k es el coeficiente a calcular y N es el número de muestras tomadas de la señal.

$$a_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cos(2\pi kn / N) + j \frac{1}{N} \sum_{n=0}^{N-1} x[n] \sin(2\pi kn / N)$$

$$60 = \frac{f_s}{N} k$$

Ecuación 30.

Por medio de la ecuación 31 se puede construir la función v2.

$$V2 = A * \cos(2\pi 60n / f_s + \theta)$$

$$A = 2\sqrt{(\text{real}(a_k))^2 + (\text{img}(a_k))^2}$$

$$\theta = \text{angle}(a_k)$$

Ecuación 31.

2.2.4 Filtrado por cancelación para eliminación del corrimiento de la línea de base.

Además de las interferencias ocasionadas por las líneas de alimentación de energía eléctrica, también existen señales que distorsionan el ECG que son el producto de movimientos musculares o del mal contacto de los electrodos sobre la piel.

Este tipo de interferencias pueden ocasionar lo que se conoce muy comúnmente como corrimiento de la línea de base (*drift baseline* o *wander baseline*). Este problema puede ser inducido por los cambios en la respiración y movimientos del cuerpo, especialmente durante registros capturados en pruebas que involucren ejercicio físico.

Las componentes en el dominio de la frecuencia del corrimiento de la línea de base se encuentran por debajo de los 0.5 Hz, pero en caso de las pruebas de esfuerzo se puede presentar un límite mayor. También pueden ser del mismo rango que las componentes de baja frecuencia del ECG. Por lo tanto, la remoción de dichas componentes pueden alterar severamente la información clínica del registro electrocardiográfico, dando como consecuencia la no recomendación de filtros pasa altos, analógicos o digitales, con frecuencias de corte superiores a los 0.5 Hz. En la figura 28 se puede observar una señal electrocardiográfica con el típico corrimiento de la línea de base.

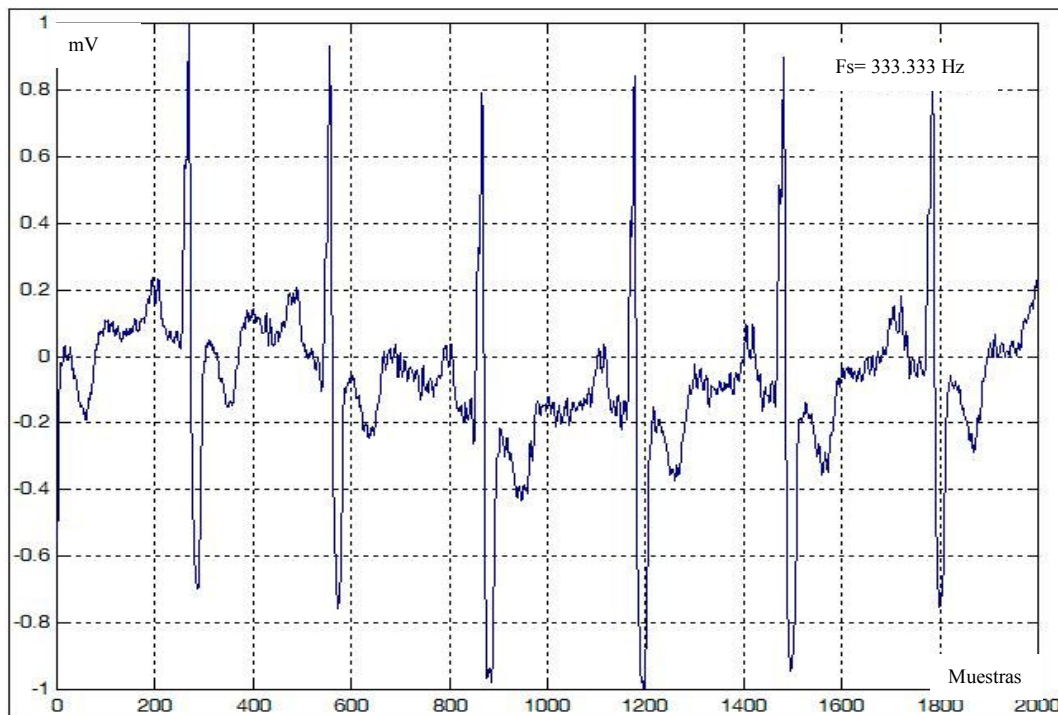


Figura 28. Corrimiento de la Línea de base en un ECG.

Como una buena estrategia para el tratamiento de este tipo de interferencia se propone estimar una función que modele el desplazamiento de la línea de

base. Esta función se puede crear por medio de la selección de muestras en diferentes puntos de la señal electrocardiográfica y por medio de técnicas de interpolación se puede obtener la curva estimada.[24]

En este trabajo de investigación como tratamiento del corrimiento de la línea de base se propone utilizar un sistema basado en el esquema del filtro por cancelación descrito en la sección 2.2.3 en el cual se crea una función $V2(n)$ estimada por medio de polinomios de interpolación utilizando los valores medios calculados de la señal de entrada $S(n)$. (ver figura 27)

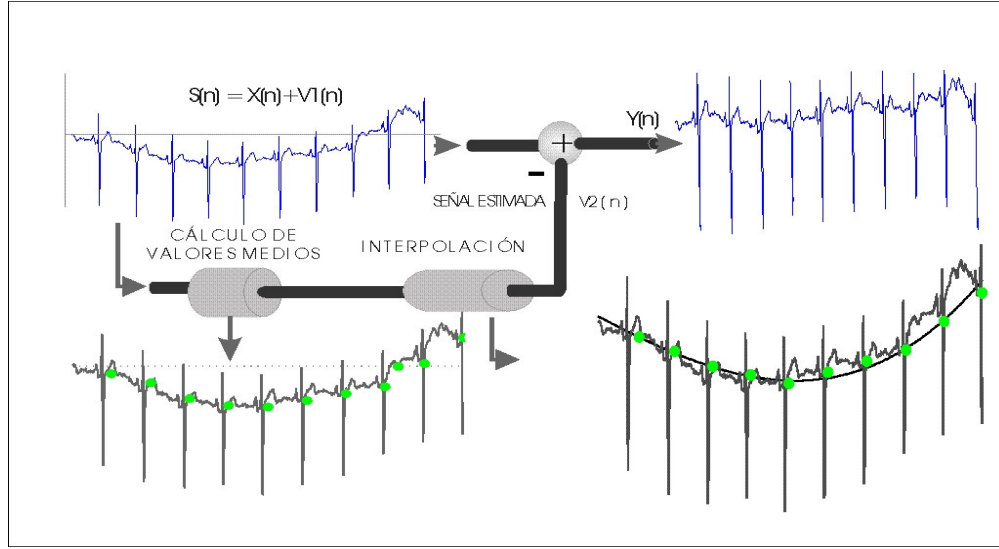


Figura 29. Filtro por cancelación para eliminación del corrimiento de la línea de base del ECG.

Como un método para la estimación de la función del desplazamiento de la línea de base por medio de las muestras recolectadas de la señal electrocardiográfica se ha escogido la regresión polinomial en el cual se crea una función basada en un polinomio de m -ésimo grado . (ver ecuación 32).

$$y = a_0 + a_1x + a_2x^2 + + a_mx_i^m$$

Ecuación 32.

Tomando la suma de los cuadrados de los residuos contenida en la ecuación 33 y su derivada con respecto a cada uno de los coeficientes (ecuación 34) se puede obtener un sistema de ecuaciones (ecuación 35).

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2 - - a_mx_i^m)^2$$

Ecuación 33.

$$\begin{aligned}
\frac{\partial S_r}{\partial a_0} &= -2 \sum (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m) \\
\frac{\partial S_r}{\partial a_1} &= -2 \sum x_i (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m) \\
\frac{\partial S_r}{\partial a_2} &= -2 \sum x_i^2 (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m) \\
&\vdots \\
&\vdots \\
&\vdots \\
\frac{\partial S_r}{\partial a_m} &= -2 \sum x_i^m (y_i - a_0 - a_1 x_i - a_2 x_i^2 - \dots - a_m x_i^m)
\end{aligned}$$

Ecuación 34.

$$\begin{array}{ccccccc}
a_0 n + & a_1 \sum x_i + & a_2 \sum x_i^2 + & \dots + & a_m \sum x_i^m & = & \sum y_i \\
a_0 \sum x_i + & a_1 \sum x_i^2 + & a_2 \sum x_i^3 + & \dots + & a_m \sum x_i^{m+1} & = & \sum x_i y_i \\
a_0 \sum x_i^2 + & a_1 \sum x_i^3 + & a_2 \sum x_i^4 + & \dots + & a_m \sum x_i^{m+2} & = & \sum x_i^2 y_i \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_0 \sum x_i^m + & a_1 \sum x_i^{m+1} + & a_2 \sum x_i^{m+2} + & \dots + & a_m \sum x_i^{2m} & = & \sum x_i^{2m} y_i
\end{array}$$

Ecuación 35.

Con la ecuación 35, el problema se reduce a determinar un polinomio de grado m con mínimos cuadrados resolviendo un sistema de $m+1$ ecuaciones lineales simultáneas.[25]

A través de las siguientes líneas de código en Matlab® en el programa *lineadabase.mat* se realiza la simulación del proceso de eliminación del corrimiento de la línea de base de una señal electrocardiográfica.

En este trabajo de investigación se utiliza el cálculo de valores medios de la señal como método para la selección de las muestras con las cuales se construye el sistema de ecuaciones lineales que permite obtener los coeficientes del polinomio que serán utilizados para la estimación de la curva.

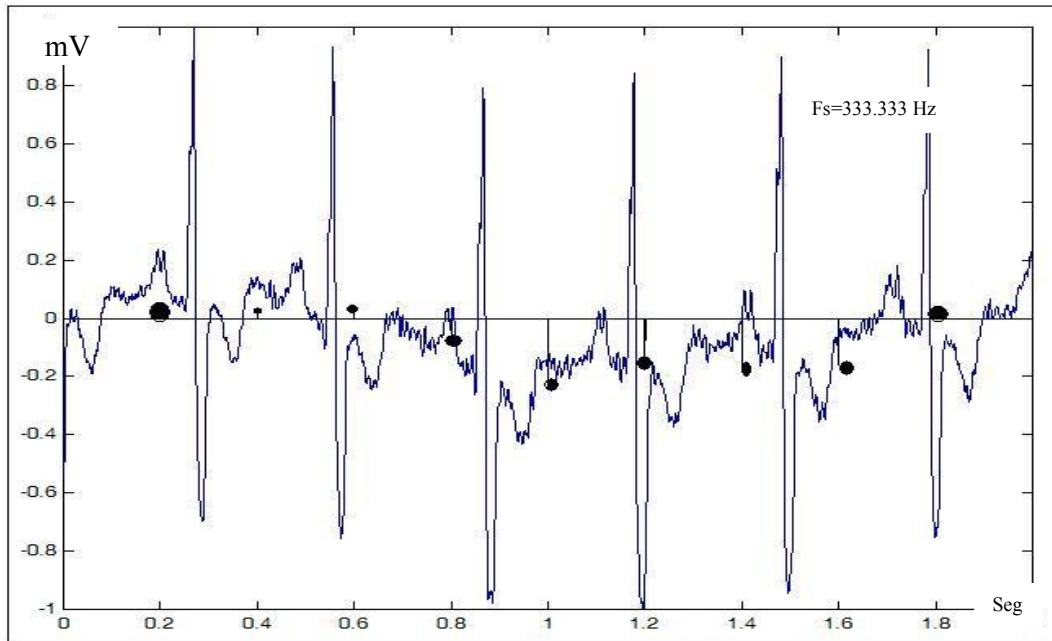
En el programa *lineadabase.mat* se calculan los valores medios de la señal en ventanas con una longitud similar a la duración de un complejo cardiaco. En este ejemplo se sugiere tomar alrededor de 200 muestras para obtener un ciclo de la señal.

Tomando una ventana de 2000 muestras de señal electrocardiográfica con una tasa de muestreo de 500 Hz se obtienen 10 valores medios y se puede realizar el proceso de con un vector de interpolación conformado por un arreglo de 2000 datos ($\Delta t = 0.001$ seg).

En la figura 30 se ilustra el resultado obtenido de tomar una señal electrocardiográfica almacenada en una base de datos con un corrimiento de línea de base y los diferentes valores medios calculados.

La función estimada de la línea de base se puede apreciar en la figura 31 y el resultado final del proceso de eliminación de la interferencia en la figura 32.

En la figura 33 se ilustra la comparación entre el resultado obtenido por el sistema de filtrado por cancelación y un filtro FIR pasa altos.



**Figura 30. Señal ECG con corrimiento de línea de base
Y cálculo de los valores medios.**

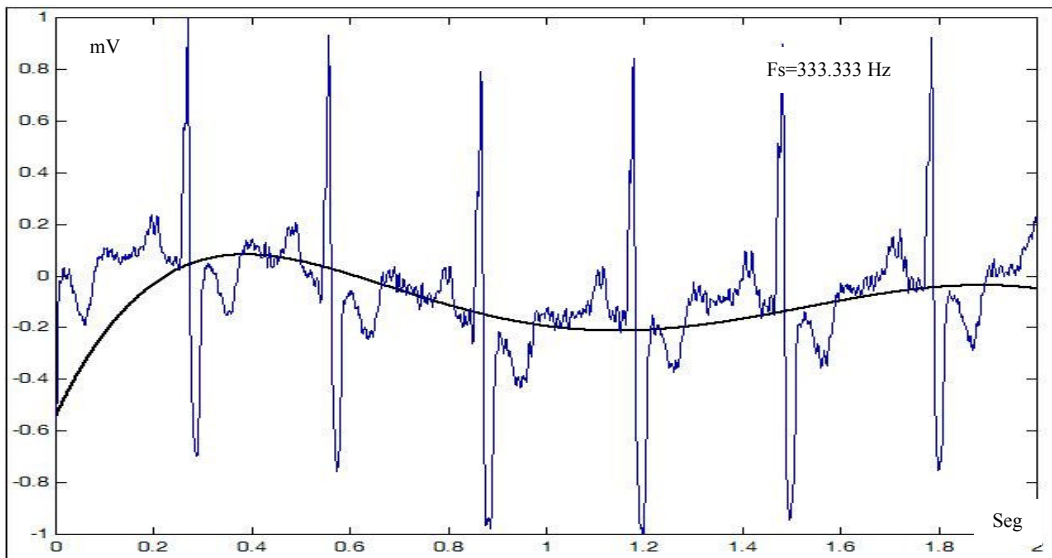


Figura 31. Función estimada de la línea de base.

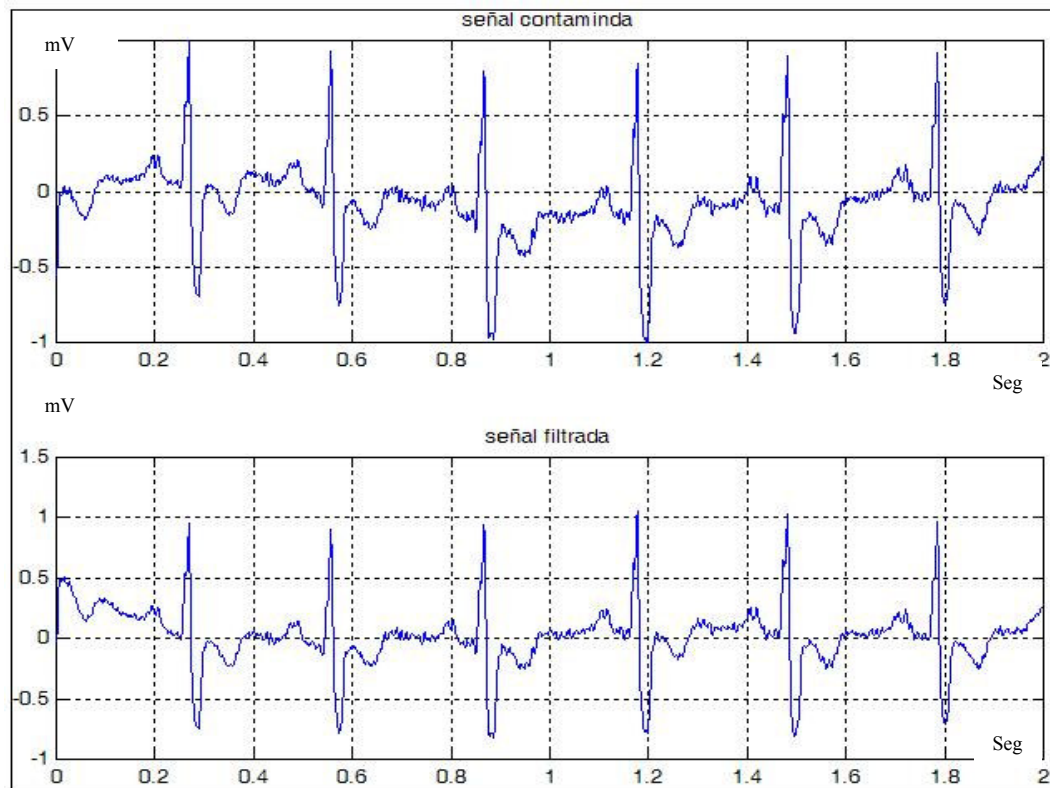


Figura 32. Resultado obtenido de la eliminación de la línea de base.

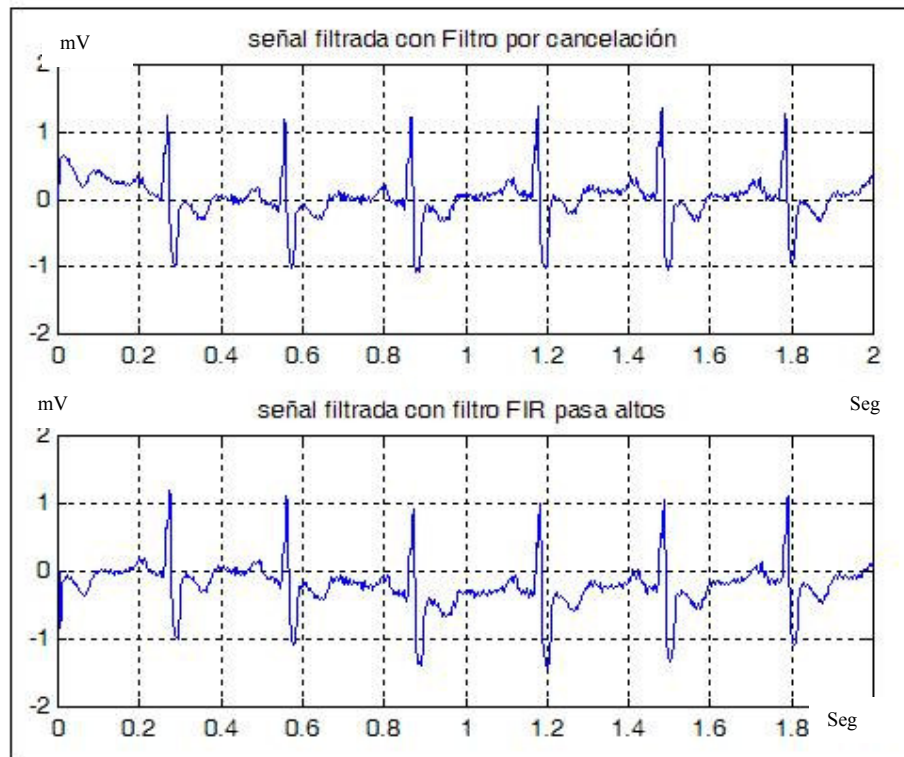


Figura 33. Comparación entre filtrado por cancelación y filtro FIR pasa altos.

3. EL ANÁLISIS DE LA VARIABILIDAD DE LA FRECUENCIA CARDIACA

El término sistema nervioso autónomo es atribuido en el inicio del siglo XX a JN Langhelly para describir aquellos nervios que están relacionados predominantemente con la regulación de las funciones corporales.

Los nervios autonómicos comprenden las terminales simpáticas y las parasimpáticas. La influencia del sistema parasimpático llega al corazón a través del nervio vagal el cual provee de gran inervación al nodo sino-atrial, las vías del conducto atrioventricular y el miocardio. Los efectos de estas divisiones son complementarios, mientras la actividad de los nervios simpáticos excitan el corazón, comprime los vasos sanguíneos, los nervios parasimpáticos inducen las respuestas opuestas. Los continuos cambios en el balance del sistema simpático – parasimpático hacen que el ritmo sinusal exhiba fluctuaciones las cuales reciben el nombre de variabilidad de la frecuencia cardiaca (VFC), la cual es un espejo del sistema de control cardiorrespiratorio y es una herramienta para el seguimiento de pacientes diabéticos y de pos infarto, dando información sobre el balance del sistema simpático – parasimpático y el riesgo de muertes por anomalías cardíacas en estas personas. La medición de la VFC está basada en el análisis de los intervalos de latidos cardiacos adyacentes o de valores instantáneos de la frecuencia cardiaca para cada latido sucedido en grabaciones continuas de electrocardiogramas (más precisamente intervalos entre puntos sucesivos de picos R del complejo QRS). La VFC no es una medida del ritmo del nodo sino atrial, porque el análisis no está basado en las mediciones de los intervalos PP (despolarizaciones atriales) pero si de los intervalos RR (despolarizaciones ventriculares).

En los últimos años las muertes repentinas por causas del corazón han sido consideradas como problemas de salud pública en países occidentales. Recientemente se han acumulado evidencias de la relación del sistema nervioso autónomo con estas muertes cardiacas repentinas. Una reducción de la VFC ha sido reportada en numerosas enfermedades cardiológicas y no cardiológicas. Una baja VFC después de infarto en el miocardio puede reflejar un decrecimiento en la actividad vagal sobre el corazón, lo cual conduce a una prevalencia del mecanismo parasimpático y a una inestabilidad eléctrica del corazón. En pacientes con un reciente trasplante de corazón se ha observado una gran reducción en la VFC .[26]

3.1 MEDIDAS DE CORTO TIEMPO DE LA VARIABILIDAD DE LA FRECUENCIA CARDIACA.

Los análisis de la VFC de corto tiempo están basados usualmente en grabaciones electrocardiográficas de 5 minutos ofreciendo la gran ventaja de que se pueden fácilmente caracterizar en condiciones controladas; en un laboratorio con temperatura constante, libre de ruido y de otros factores de estímulos externos, los cuales pueden influenciar sobre el estado cardiovascular autonómico.

La señal de la frecuencia cardiaca latido a latido después de la detección del complejo QRS debe poseer unos prerrequisitos antes de la realización de una análisis en el dominio espectral; por lo cual la señal debe ser lo suficientemente larga, aleatoria, estacionaria y libre de arritmias e interferencias.

“Suficientemente larga” significa que, asumiendo un estado estacionario, la consistencia de la estimación del espectro en potencia mejora con un incremento en la duración de la señal.

El término “aleatorio” es definido como unas secuencias que no pueden ser determinadas o definidas por una expresión o regla matemática única. El término “estacionario” significa que sus índices estadísticos no cambian con el tiempo.

La frecuencia de muestreo y la resolución de la amplitud del registro del electrocardiograma para el análisis de la VFC deben ser escogidas apropiadamente. La frecuencia de muestreo determina la precisión del reconocimiento del complejo QRS y, por lo tanto, determina la exactitud de la medición de la duración de los intervalos RR sucesivos. Como también, una baja frecuencia de muestreo, puede introducir errores en la detección de la onda R. El valor mínimo de la frecuencia de muestreo está en el rango de los 250 y 500 Hz, pero en equipos digitales para análisis tipo Holter, las frecuencias de muestreo presentan valores mayores de 1024 Hz que pueden mejorar la exactitud de las medidas de la duración de los intervalos RR. Con respecto a la resolución de los dispositivos para conversión análogo a digital utilizados para la digitalización del electrocardiograma, se recomienda usar una resolución de 12 bits. [27]y [28]

3.1.2 Detección de la onda R.

La detección de la onda R es un proceso fundamental en el análisis de la variabilidad de la frecuencia cardíaca debido a que los índices calculados (dominio del tiempo y frecuencia) están basados en la medición del tiempo entre cada onda R.

La diferenciación forma la base de muchos algoritmos de detección del complejo QRS la cual es la base fundamental en el cálculo de los tiempos RR. Esto es básicamente un filtrado pasa altos donde el filtro diferenciador hace resaltar las altas frecuencias que son características del complejo QRS atenuando las bajas frecuencias que son propias de las ondas P y T.

Un algoritmo basado en la primera y segunda derivada originalmente desarrollado por Balda en 1977 fue modificado por Ahlstrom y Tompkins para ser usado en análisis de alta velocidad de grabaciones de ECG. Friesen implementó en 1990 el algoritmo como parte de un estudio para comparar sensibilidad al ruido en diferentes tipos de algoritmos de detección del complejo QRS.

El algoritmo se puede representar a través de las siguientes ecuaciones en diferencias:

Primera derivada:

$$y_0[n] = x[n] - x[n-2]$$

Segunda derivada:

$$y_1[n] = x[n] - 2x[n-2] + x[n-4]$$

Suma de primera y segunda derivada:

$$y_2[n] = 1.3y_0[n] + 1.1y_1[n]$$

Umbral:

$$y_2[n] > 1$$

Por medio de las líneas de código contenidas en el programa *deteccionondaR.mat* se puede simular en Matlab el algoritmo anteriormente planteado. En las figuras 34 y 35 se puede apreciar el resultado obtenido al aplicar el algoritmo a una señal ECG proveniente de una base de datos.

En este proyecto de investigación se ha optado por la implementación del algoritmo para la detección del complejo QRS desarrollado por Pan y Tompkins en 1985 el cual fue descrito por Hamilton y Tompkins en 1986. Este reconoce el complejo QRS basado en análisis de pendiente, amplitud y anchura. Además cuenta con la implementación de un umbral cuyo valor cambia dependiendo de la amplitud de la señal.

En la figura 36 se ilustra el diagrama de bloques para el desarrollo del algoritmo que contiene una etapa de filtrado pasa banda conformado por dos filtros en

cascada (pasa altos y pasa bajos integrador). Seguidamente se encuentran los procesos de diferenciación, elevación al cuadrado y promediado en el tiempo de la señal.

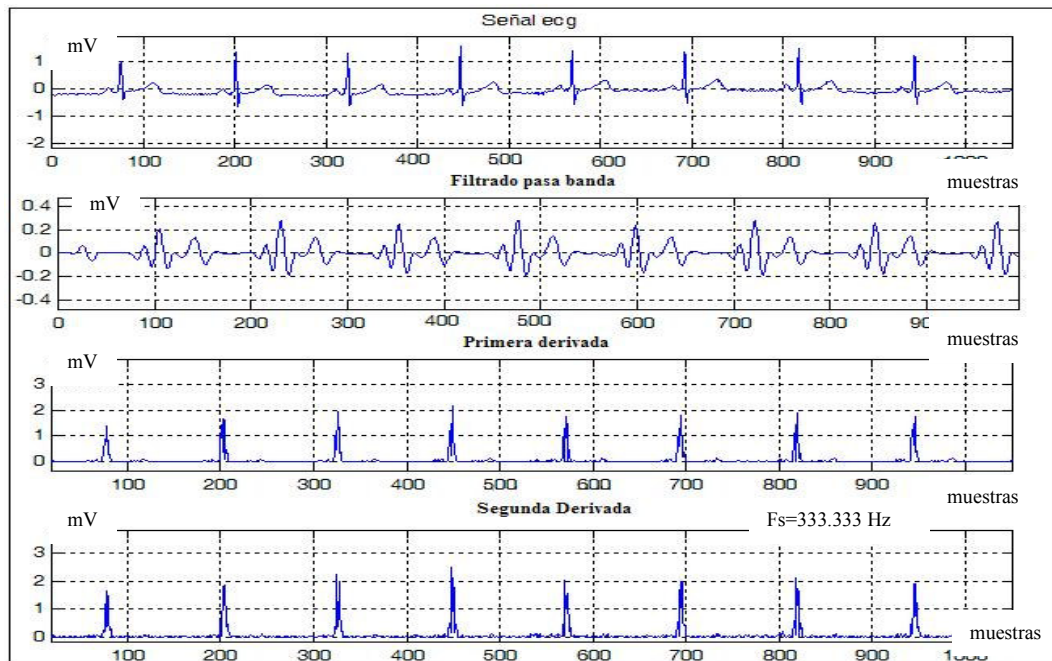


Figura 34. Filtrado pasa banda y Primera y segunda derivada aplicada al ECG.

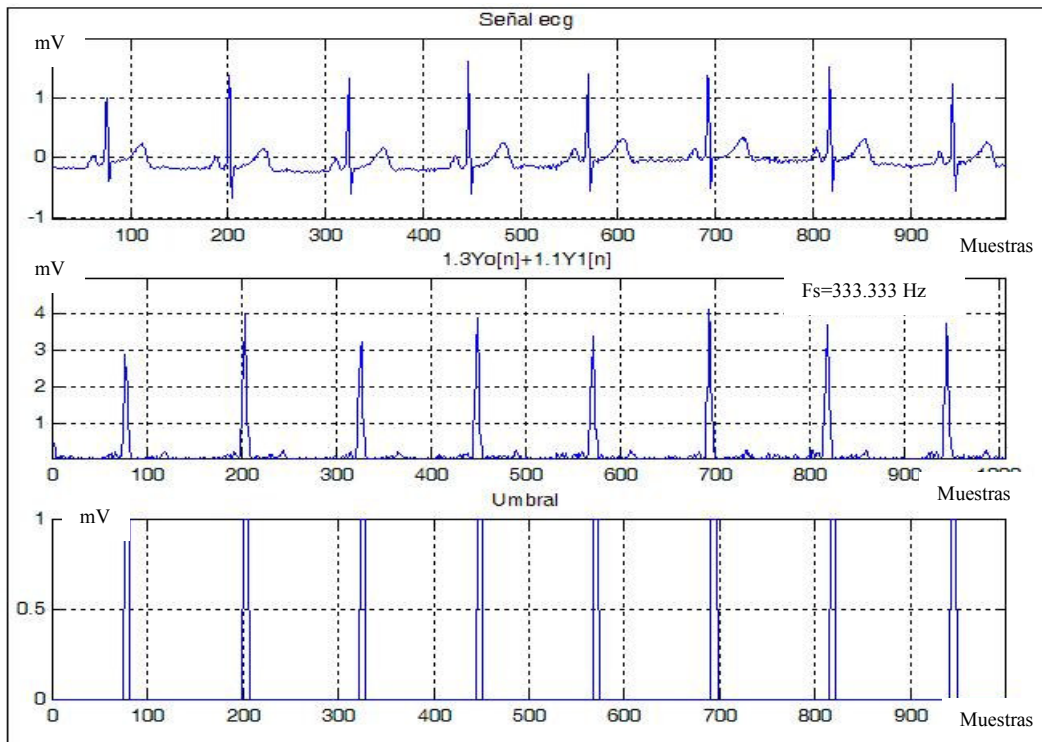


Figura 35. Suma ponderada y aplicación de umbral al ECG.

La función del filtrado pasa banda es aislar la energía predominante en el complejo QRS centrada en la frecuencia de 10 Hz, atenuando las características de las ondas *P* y *t* y el ruido electromiográfico.

Como siguiente paso se tiene la diferenciación, como una técnica estándar para encontrar altas pendientes que normalmente distinguen el complejo QRS de otras ondas del ECG.

Seguidamente se calcula el cuadrado de cada muestra de la señal. En esta etapa todos los valores toman valores positivos y son acentuadas las componentes de alta frecuencia.

El cuadrado de la señal pasa a través de un integrador de ventana móvil, el cual suma el área bajo la forma de onda sobre un intervalo de 150 ms, avanzando una muestra del intervalo.

Un umbral adaptativo es aplicado a la onda que ha pasado por el filtro pasa banda y el integrador móvil. El umbral es actualizado por medio de la estimación de los niveles picos de la señal.

Por último es necesario la implementación de un algoritmo que calcule la duración entre cada complejo QRS que es equivalente a los tiempos RR.

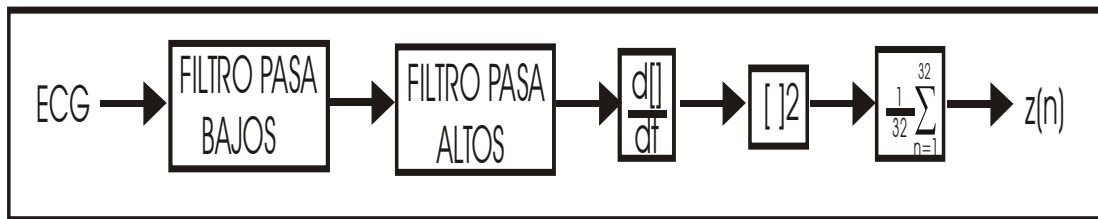


Figura 36. Etapas de filtrado para detección del complejo QRS.

El filtro pasabanda propuesto en el esquema de Tompkins trata de reducir el ruido en la señal ECG resaltando el espectro del complejo QRS que se encuentra entre los 5 y 15 Hz.

Por medio de líneas de comando de Matlab se puede diseñar un filtro FIR de estas especificaciones para tratar una señal ECG extraída de una base de datos con período de muestreo de 0.008 seg.

En la figura 37 se encuentra la respuesta en frecuencia del filtro FIR.

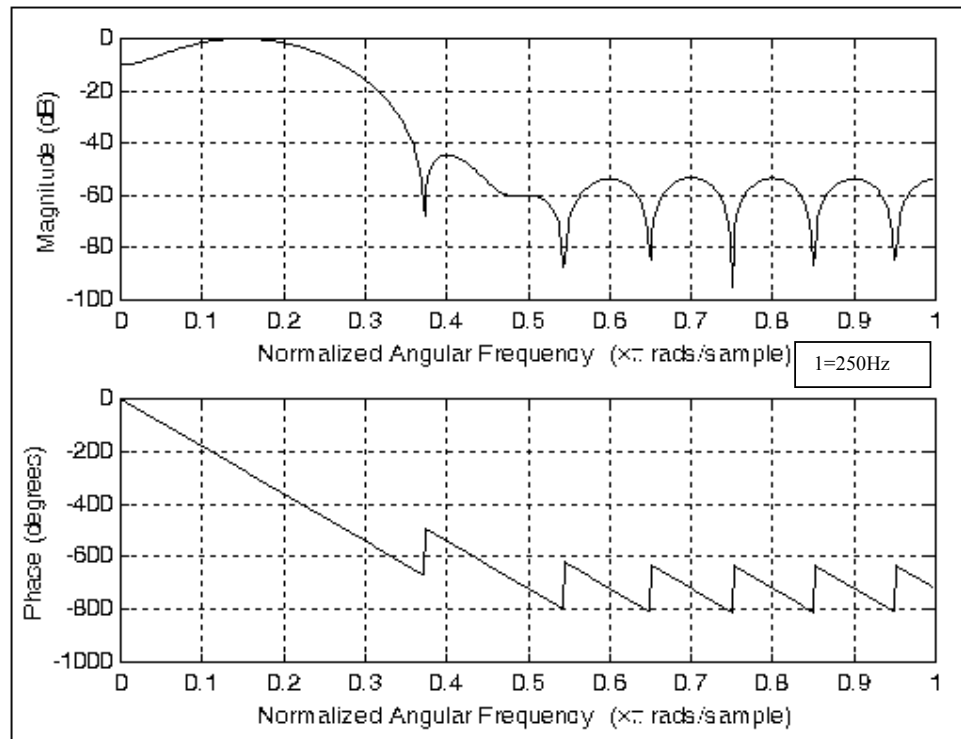


Figura 37. Respuesta en frecuencia del filtro pasa banda.

La figura 38 contiene el resultado obtenido de tomar una señal ECG y someterla al proceso del filtrado pasa banda. En la figura 39 se puede diferenciar los respectivos espectros de frecuencia de la señal de entrada y de salida.

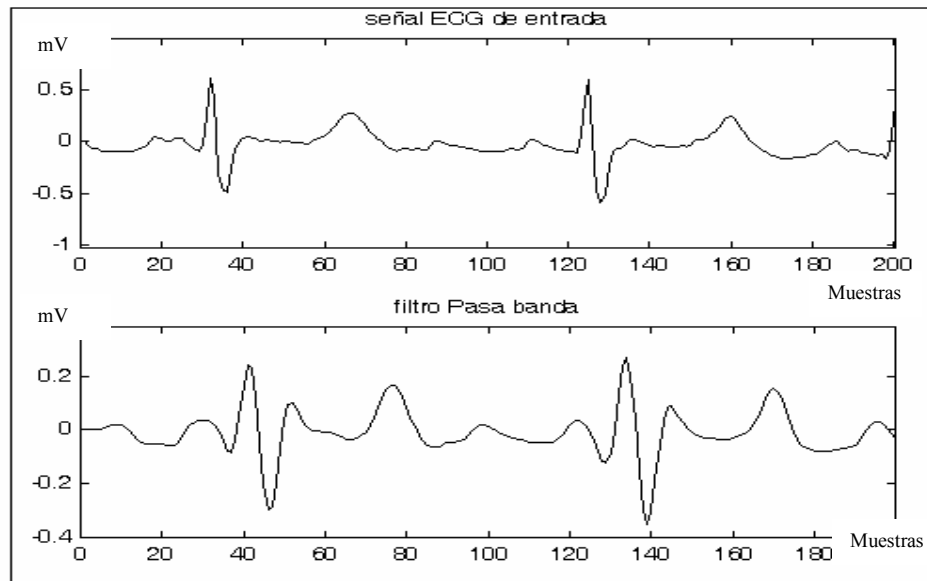


Figura 38. Señal de entrada y de salida del filtro Pasa bandas. (Dominio del tiempo).

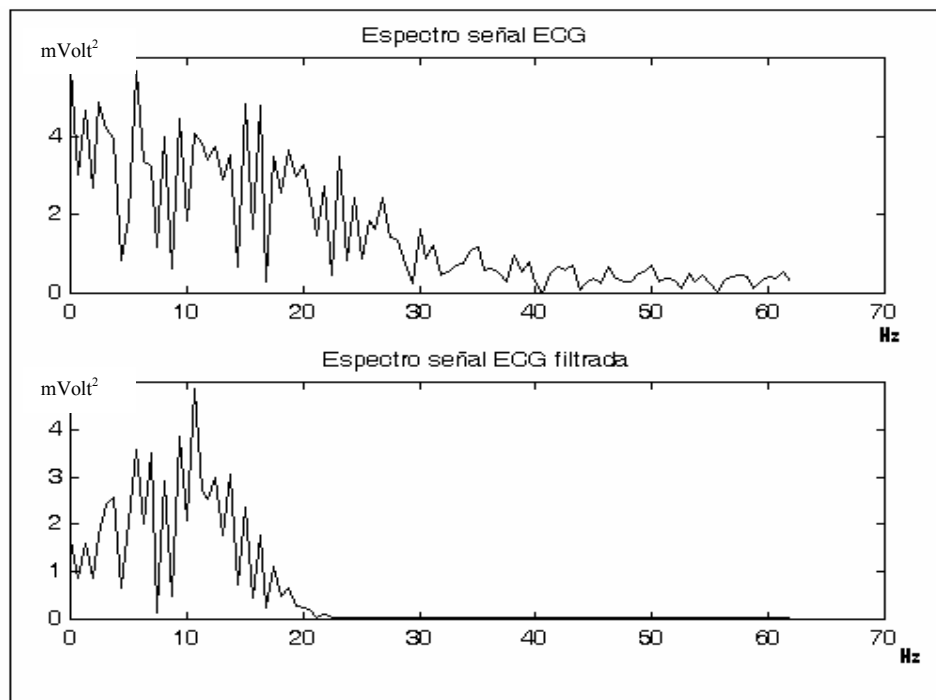


Figura 39. Señal de entrada y de salida del filtro Pasa bandas. (Dominio de la frecuencia).

Después del proceso de filtrado, la señal resultante pasa a un proceso de diferenciación donde se obtiene información de la pendiente del complejo QRS. La respuesta en frecuencia del filtro derivador se puede detallar en la figura 40. El resultado obtenido a la salida del filtro derivador en el dominio del tiempo y de la frecuencia se pueden observar respectivamente en las figuras 41 y 42.

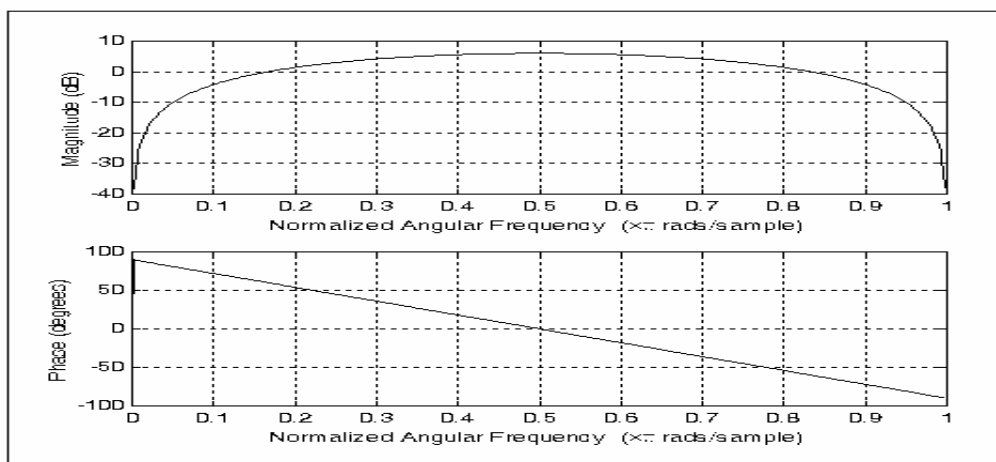


Figura 40. Respuesta en frecuencia del Filtro derivador.

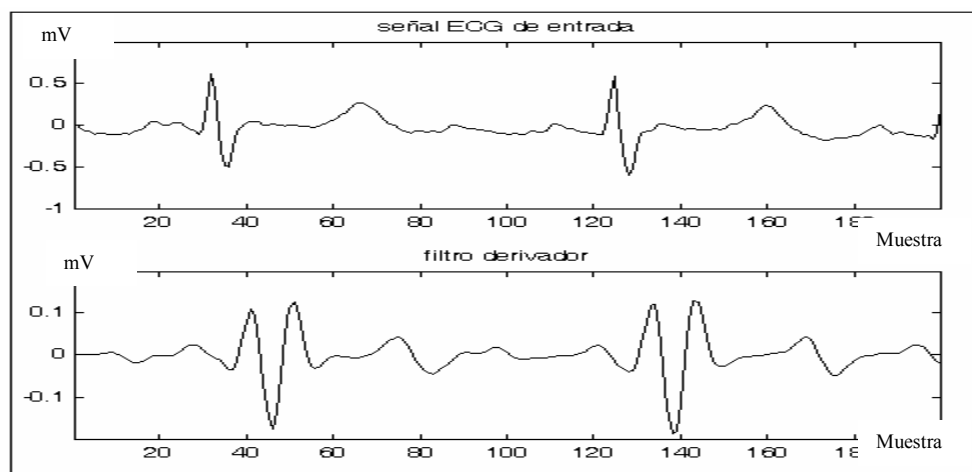
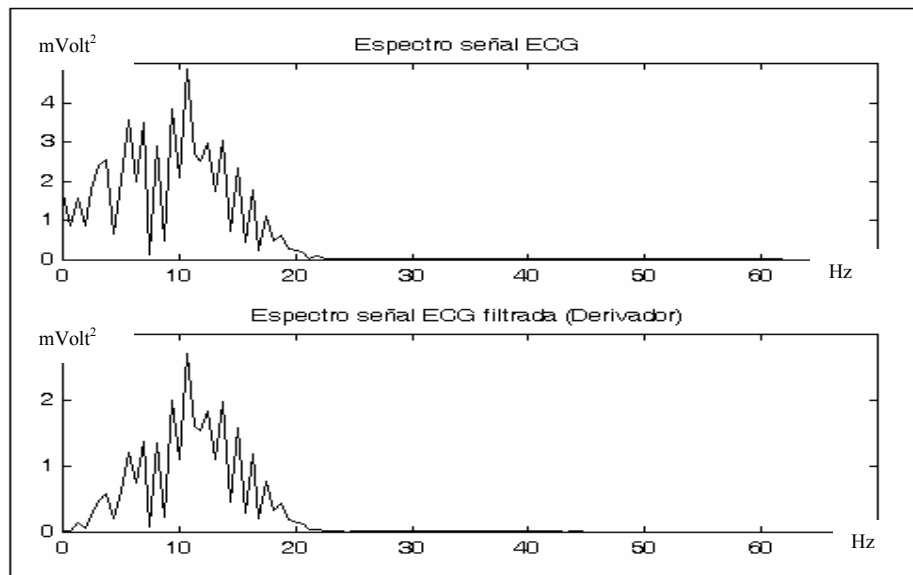


Figura 41. Salida del filtro derivador. (Dominio del tiempo).



**Figura 42. Salida del filtro derivador.
(Dominio de la frecuencia).**

Como paso siguiente al proceso de derivación se tiene la elevación al cuadrado de los datos. En esta operación se toman todos los datos obtenidos para el proceso anterior para ser elevados al cuadrado obteniendo una señal positiva y amplificada. Esto resalta altas componentes en frecuencia de la señal. La figura 43 contiene el resultado de la elevación al cuadrado.

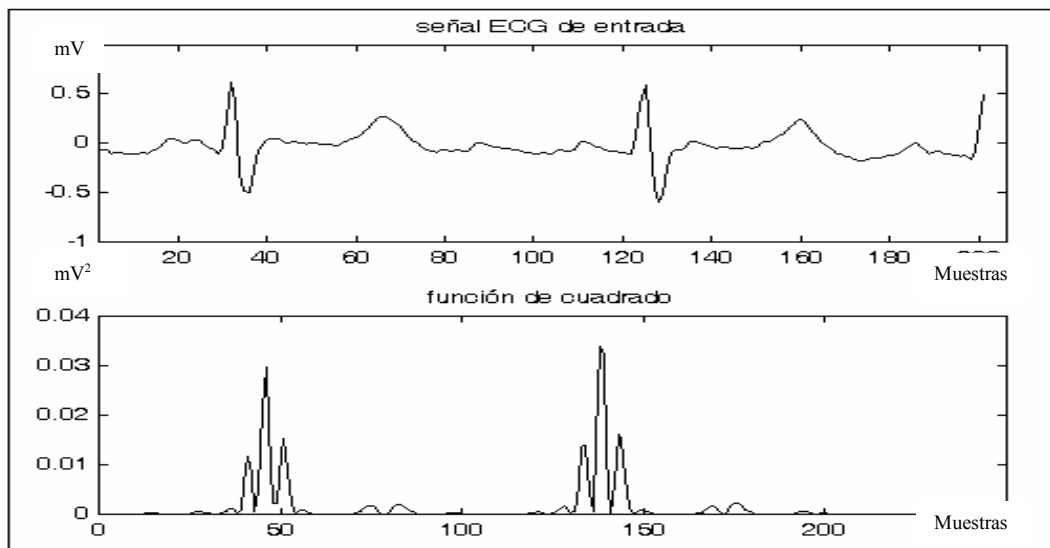


Figura 43. Elevación al cuadrado de la señal.

Con el fin de obtener más información de la onda R se implementa un filtro integrador de ventana móvil. El número de muestras en el ancho de la ventana es un parámetro que debe ser escogido cuidadosamente. El ancho de la ventana debe ser aproximadamente teniendo en cuenta la duración del complejo QRS. Teniendo en cuenta que la frecuencia de muestreo de la señal electrocardiográfica adquirida por la plataforma DSK5402 es de 500 Hz, se ha escogido un ancho de ventana de 20 muestras (0.04 seg). En la figura 44 se lustra el resultado obtenido del filtro integrador de ventana móvil.

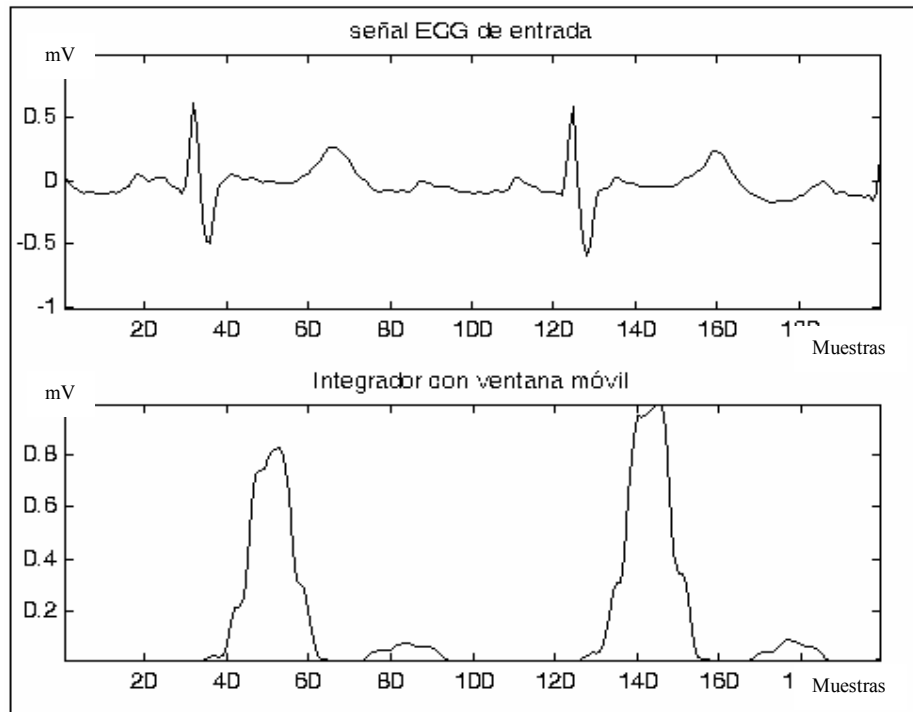


Figura 44. Señal ECG después del filtrado por integrador de ventana móvil.

Los autores de este procedimiento utilizado para la detección del complejo QRS (Pan y Tompkins) recomiendan tener un conjunto de umbrales para la obtención de los picos de la señal. Los valores picos de la señal son definidos como todos aquellos que pertenecen al complejo QRS, mientras los picos de ruido son los correspondientes a las ondas T, interferencias por movimiento muscular, etc.

Dos conjuntos de umbrales son aplicados a la forma de onda obtenida a la salida de la ventana integradora:

$SPKI = 0.125 \text{ PEAKI} + 0.875 \text{ SPKI}$ Si PEAKI es un pico de la señal.
 $NPKI = 0.125 \text{ PEAKI} + 0.875 \text{ NPKI}$ Si PEAKI es un pico de ruido.

$UMBRAL11 = NPKI + 0.25 (SPKI - NPKI)$
 $UMBRAL12 = 0.5 \text{ THRESHOLD11}$

Todas las variables mencionadas en el conjunto de umbrales se refieren a la forma de onda filtrada con la función integradora y se describen a continuación:

PEAKI : es el pico total.

SPKI: es la estimación del pico de la señal.

NPKI: es la estimación del pico del ruido.

UMBRAL11: es el primer umbral aplicado.

UMBRAL12: es el segundo umbral aplicado.

Un pico es determinado cuando la señal cambia de dirección en un intervalo de tiempo. De esta manera, SPKI es el pico que el algoritmo supone que pertenece al complejo QRS, mientras que NPKI es cualquier pico que no está relacionado con la señal de interés. Como se puede ver en las expresiones los umbrales son calculados de valores previos según la señal ECG de una persona en particular.

Cuando un nuevo pico es detectado, este debe ser catalogado como un pico de ruido o un pico de señal. Si el nivel del pico excede el valor de UMBRAL11 durante el primer análisis de la señal, este se considera como un pico del complejo QRS.

Por medio de las líneas de programas contenidas en el archivo *detecRumbral.mat* se ha implementado el algoritmo para establecer el nivel de umbral que permita la detección de la onda R.

La figura 45 contiene el resultado final obtenido por medio de las etapas de filtrado y el establecimiento de un nivel umbral aplicado a un registro de señal electrocardiográfica obtenida de una base de datos.

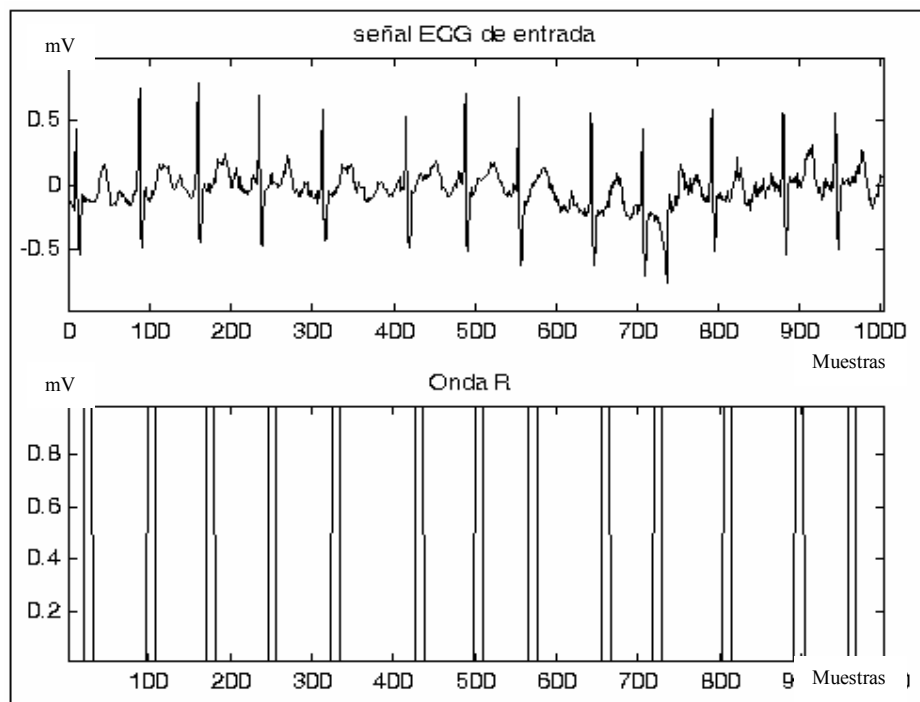


Figura 45. Detección de la onda R por medio de un nivel de umbral.

3.1.2 Medidas en el dominio del tiempo de la VFC.

Durante este proyecto de investigación fueron tomados los cálculos estadísticos más simples y comunes en el estudio de la VFC en el dominio del tiempo que son derivados de mediciones directas de los intervalos RR y están resumidos en la tabla 1.

VARIABLE	UNIDAD	DESCRIPCIÓN	FORMULA
SDNN	ms	Desviación estandar de los intervalos RR.	$\sigma_{RR} = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (RR(i) - \bar{RR})^2}$
RMSSD	ms	Raíz cuadrada del promedio de la suma de las diferencias entre intervalos RR adyacentes.	$RMS_{diff} = \sqrt{\frac{1}{N} \sum_{i=1}^N (diffRR(i))^2}$
NN50	n	Número de pares de intervalos RR adyacentes cuya diferencias es de más de 50 ms.	$(RR(i) - RR(i+1)) > 50 \text{ ms}$
pNN50	%	Porcentaje del conteo de NN50 de todos los intervalos RR: (conteo de Nn50/conteo total de los intervalos RR)x100.	$\% (RR(i) - RR(i+1)) > 50 \text{ ms}$

Tabla 1. Medidas en el dominio del tiempo de la VFC.

Tomado de Clinical Guide to Cardiac Autonomic Tests. Malik Marek. Kluwer Academic Publishers. 1998.

En algunos países se han realizado estudios en los cuales se compararon índices derivados de los intervalos RR en exámenes realizados con personas normales y pacientes con enfermedades como el caso de los diabéticos tipo I con la presencia de neuropatía autonómica cardiovascular. En la tabla 2 es un gran ejemplo en la que se pueden contrastar los resultados obtenidos en estudios de este estilo y se puede apreciar que en los pacientes con diabetes tipo I el valor medio de la frecuencia cardiaca es mayor que en los sujetos sanos.[29]

Parámetros de la VFC	Diabéticos tipo I	Sujetos normales
<i>En tiempo</i>	(n = 110)	(n = 100)
Media RR (ms)	763,0 ± 157,7	889,9 ± 127,6
Desviación estándar (ms)	45,9 ± 32,7	68,7 ± 39,9
Coefficiente de variación (%)	8,1 ± 5,7	8,9 ± 5,3
RR máximo (ms)	1 123,6 ± 1 497,1	1 235,0 ± 441,9
RR mínimo (ms)	632,8 ± 138,2	714,2 ± 144,1

**Tabla 2. Índices del dominio del tiempo.
(sujeto normal y diabético tipo I)**

Tomado de: Variabilidad de la Frecuencia cardíaca en reposo para detectar neuropatía autonómica cardiovascular en diabéticos tipo I.

Revista Cubana End. 1999;10(1):25-37

En la tabla 3 se hace la comparación entre sujetos sanos y pacientes diabéticos con neuropatía autonómica cardiovascular (NACV) y sin ella. La tabla 3 ilustra que el valor de la Media RR en los pacientes diabéticos con NACV presentaron una mayor frecuencia cardíaca que en los pacientes sin NACV.

Parámetros	Con NACV	Sin NACV	Sujetos normales
	(n = 58)	(n = 52)	(n = 100)
Media RR (ms)	766,4 ± 131,2	771,9 ± 152,9	882,1 ± 148,9
Desviación estándar (ms)	39,6 ± 36,1	54,0 ± 29,1	68,1 ± 40,2
Coefficiente de variación (%)	5,0 ± 4,3	7,4 ± 3,9	8,0 ± 5,3
<i>Intervalos RR</i>			
. RR max (ms)	969,5 ± 502,1	1 242,6 ± 426,6	1 230,0 ± 442,6
. RR min (ms)	620,0 ± 147,1	644,8 ± 132,3	713,9 ± 143,4

**Tabla 3. Índices del dominio del tiempo.
(sujeto normal y diabético con y sin NACV)**

Tomado de: Variabilidad de la Frecuencia cardíaca en reposo para detectar neuropatía autonómica cardiovascular en diabéticos tipo I.

Revista Cubana End. 1999;10(1):25-37

Los autores del estudio del cual fueron tomados los datos de las tablas 2 y 3 consideraron que el estudio de la VFC a corto plazo puede ser aplicado como método de detección temprana de la NACV en los diabéticos tipo I.

Estos datos ofrecen un ejemplo de comparación de las medidas en el dominio del tiempo de la VFC entre sujetos normales y pacientes con alguna enfermedad.

3.1.3 Medidas en el dominio de la frecuencia de la VFC.

El análisis espectral de potencia descompone la señal de ritmo cardíaco en sus componentes en frecuencia y los cuantifica en términos de su intensidad relativa. Esto provee estimaciones de la función de densidad del espectro en potencia (PSD) de la señal de ritmo cardíaco.

En la tabla 4 se ilustran las principales medidas en el dominio de la frecuencia en estudios de corto tiempo de la VFC. En la figura 46 se puede apreciar que cada componente en frecuencia que se derivan de la función de densidad espectral de la señal de ritmo cardíaco tienen sus propias interpretaciones.

Para las mediciones de corto tiempo, es necesario poseer grabaciones de señal electrocardiográfica de por lo menos 10 tiempos la longitud de onda del menor valor de frecuencia de la componente espectral que se encuentre bajo investigación. Acorde a esto, el estudio de las componentes de alta frecuencia HF de la VFC (0.15 Hz) requiere obtener una grabación de aproximadamente 1 minuto. Para el estudio de componentes de baja frecuencia LF (0.004 Hz) es necesario tener una grabación de 4 minutos y una duración de 1 hora para el estudio de componentes de muy baja frecuencia VLF (0.003 Hz).

Por lo general la magnitud de cada componente en frecuencia es expresada en potencia absoluta (integral de la curva de la densidad espectral de potencia entre los límites de frecuencia de cada componente), pero también es común el uso de unidades normalizadas que expresan la potencia centrada en la frecuencia de interés (HF o LF) dividido por la potencia total menos la potencia de la VLF. Además relación entre la potencia de la componente LF y la potencia de la componente HF (radio LF/HF) ha sido introducido para describir el balance del sistema simpático-vagal. Por lo tanto, aunque las unidades normalizadas y la relación LF/HF no son medidas precisas, pueden representar la potencia relativa de las componentes de frecuencia (LF y HF).

En el caso de neuropatías asociadas con la diabetes mellitus caracterizada por la alteración de las pequeñas fibras nerviosas suele aparecer una reducción en los parámetros en el dominio del tiempo de la VFC. En pacientes diabéticos sin evidencia de neuropatías autonómicas se han reportado la reducción de la magnitud de potencia de las componentes LF y HF durante condiciones controladas. De cualquier modo, cuando la relación LF/HF y las componentes LF y HF en unidades normalizadas, no hubo diferencias significativas en comparación con sujetos normales. De esta manera, la manifestación inicial de

esta neuropatía es probable que implique problemas en ambos tramos del sistema nervioso autónomo. (Ver figura 47)[30 ,31 y 32]

Potencia VLF	ms ²	<0.04Hz	Potencia de las componentes de muy baja frecuencia
Potencia LF	ms ²	0.04 - 0.15 Hz	Potencia de las componentes de baja frecuencia
LFnu	n.u.		Potencia LF en unidades normalizadas. Potencia LF/(Potencia total - potencia VLF) x100.
Potencia HF	ms ²	0.15 - 0.4 Hz	Potencia de las componentes de alta frecuencia.
Hfnu	n.u.		Potencia HF en unidades normalizadas. Potencia HF/(Potencia total - potencia VLF) x100.
LF/HF			Relación de LF a HF. Potencia LF/Potencia HF.

Tabla 4. Medidas en el dominio de la Frecuencia en estudios de corto tiempo de la VFC.

Tomado de Clinical Guide to Cardiac Autonomic Tests. Malik Marek. Kluwer Academic Publishers. 1998.

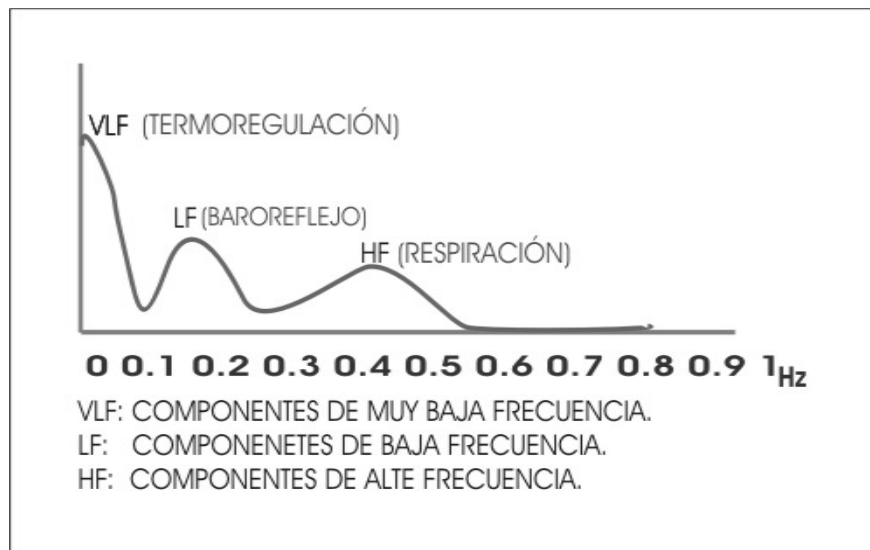


Figura 46. Componentes Espectrales la señal de ritmo cardíaco.

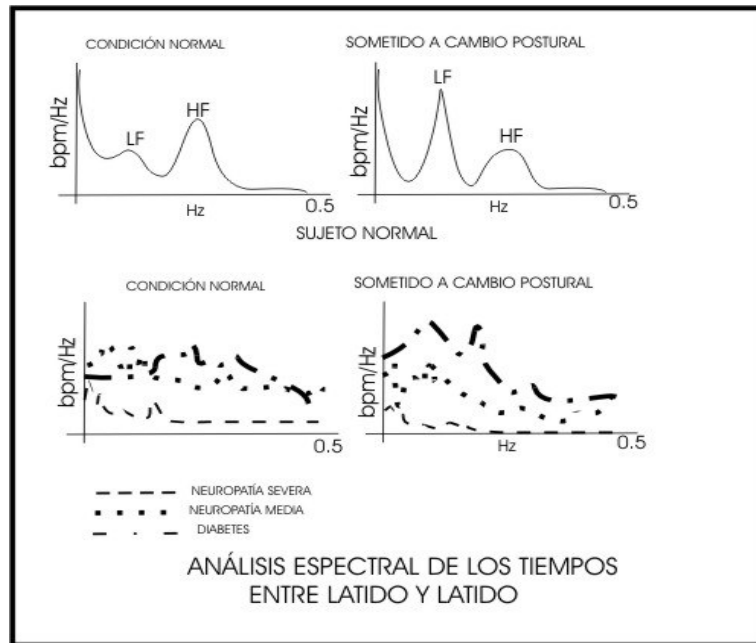


Figura 47. Comparación de las componentes espectrales en un sujeto normal y pacientes con neuropatías. (condición normal y cambio postural)

Teniendo en cuenta que La VFC es analizada con base en las series de ritmo cardiaco que están formados por valores instantáneos de tiempo t_n que están en función de los intervalos R-R detectados de la señal electrocardiográfica y que la ocurrencia de estos intervalos no se presentan a periodos iguales, se obtendrá una serie de tiempo que consiste en una señal con datos desigualmente muestreados. Debido a esto, se han propuesto diferentes métodos para el análisis en el dominio de la frecuencia de esta serie de tiempo. Tradicionalmente se ha solucionado este problema por medio de la interpolación lineal de la serie datos y posteriormente se realiza un remuestreo a intervalos iguales de tiempo obteniendo una nueva señal de ritmo cardiaco con una tasa de muestreo igual para todos los datos. Esto permite calcular los respectivos valores en el dominio de la frecuencia a través de la Transformada Discreta de Fourier (DFT).[33 y 34]

En la figura 48 se ilustra un conjunto de intervalos RR de un sujeto normal tomado de una base de datos y la señal obtenida por medio de interpolación y remuestreo. La figura 49 contiene la PSD estimada por medio de la FFT. Para fines más prácticos es recomendable analizar el programa *psdinterpolación.mat* que contiene el código que toma una serie de tiempo compuesta por intervalos RR y realiza el proceso de interpolación lineal y de análisis espectral.

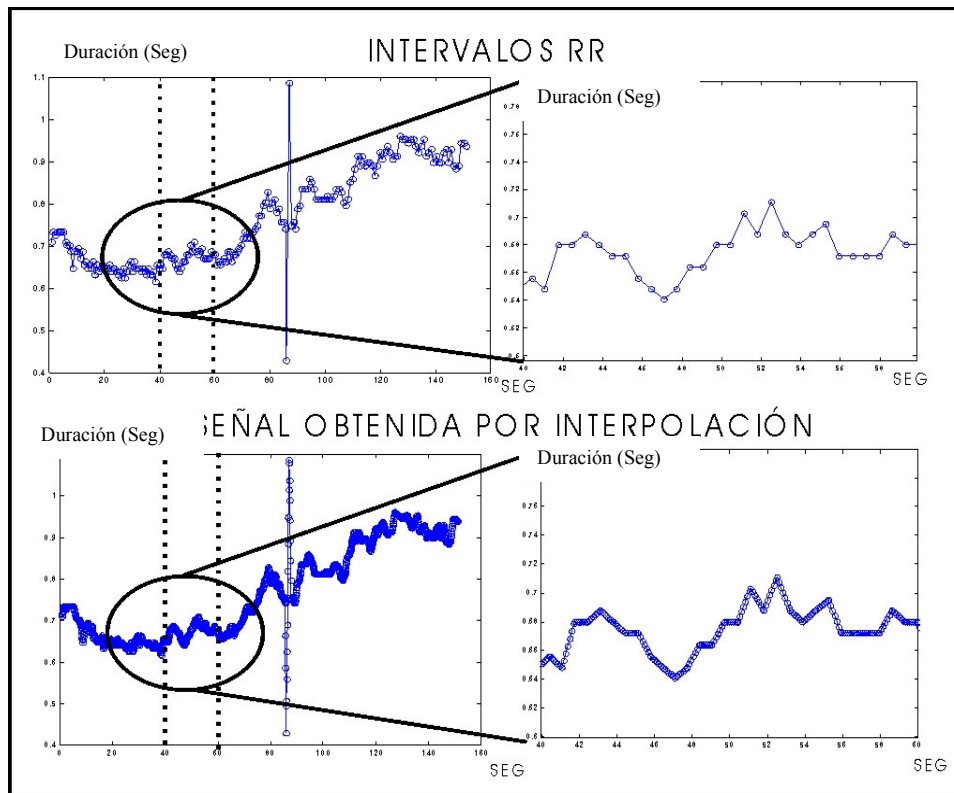


Figura 48. Señal obtenida por interpolación lineal de los intervalos RR.

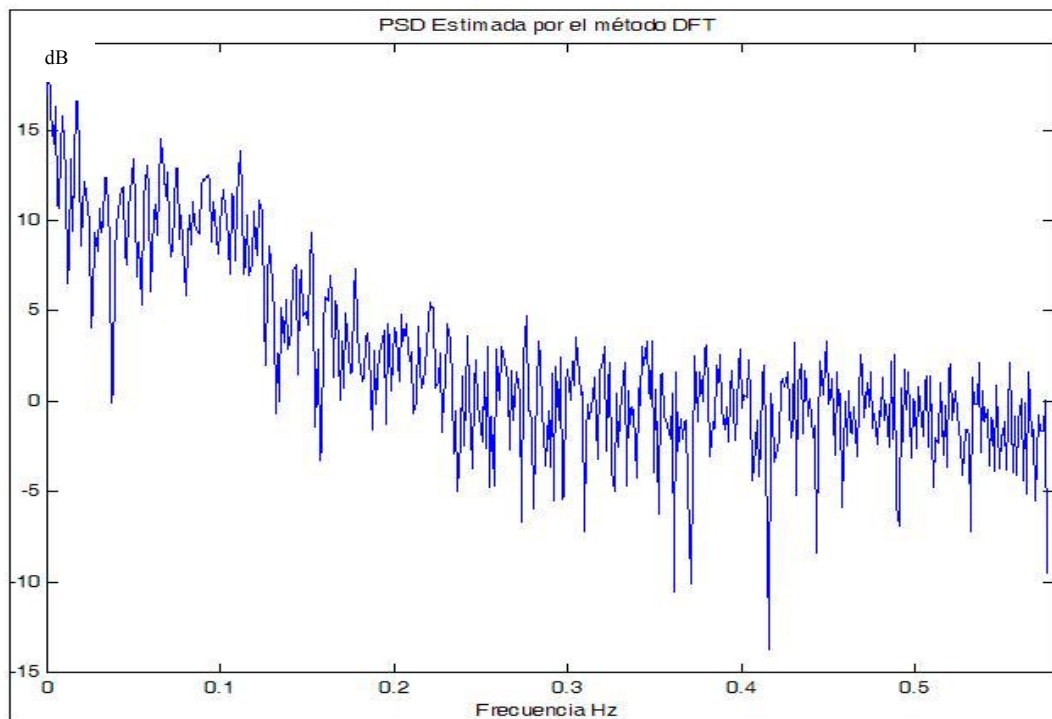


Figura 49. Análisis espectral de la señal obtenida por interpolación lineal.

El proceso de remuestreo utilizado en la estimación de la PSD de la señal de ritmo cardíaco por medio de la DFT puede introducir desviaciones en la estimación de la densidad espectral de potencia durante análisis en el dominio de la frecuencia de la señal de ritmo cardíaco. Los latidos ectópicos introducen un ruido de banda ancha en el espectro, la eliminación de estos latidos y la posterior interpolación introduce nuevas alteraciones de la densidad espectral estimada. Esta problemática ha sido recientemente abordada usando un método de estimación de densidad espectral que considera directamente las muestras no uniformes, evitando el efecto pasa-bajos introducido por el remuestreo. La técnica esta basada en el método original de Lomb de estimación de la densidad espectral de potencia (PSD) sin remuestreo, donde se ha comprobado los resultados de este análisis con señales reales de ritmo cardíaco. Este análisis permite obtener una frecuencia límite a partir de la cual el espectro puede no estar libre de solapamiento. [35]y [36]

La estimación de la PSD por medio del método de Lomb está basado en la teoría general de transformación de señales en el cual se proyecta una señal $x(t)$ sobre una base ortonormal $b(t)$. Este método toma los valores desigualmente muestreados de $x(t)$ considerando que la función base es el núcleo de Fourier $b_i(t)=e^{j2\pi f_i t}$.

El método de Lomb para la estimación de la PSD está basado en la minimización del cuadrado de las diferencias entre la proyección de la señal sobre las funciones bases y la señal bajo estudio.

Teniendo que la señal bajo estudio $x(t)$ debe ser continua y la base ortonormal $b(t)$ se obtienen los coeficientes $c(i)$ que representan a $x(t)$ en el dominio de la transformación. (ver ecuación 36).

$$c(i) = \int_{-\infty}^{+\infty} x(t)b_i(t)dt$$

Ecuación 36.

El error cuadrático $e(c_i)$ entre la señal $x(t)$ y su representación a través de los coeficientes $c(i)$ se define en la ecuación 37.

$$e(c_i) = \int_{-\infty}^{+\infty} (x(t) - c(i)b_i(t))^2 dt$$

Ecuación 37.

En el caso de señales compuestas por datos igualmente muestreados se debe trabajar con la contraparte discreta de la ecuación 36 que conduce al análisis de Fourier. Para asumir este problema, Lomb propone una estimación del espectro de Fourier de una señal desigualmente muestreada con el ajuste del modelo propuesto en la ecuación 38, donde el error cuadrático medio ε_n es minimizado con los apropiados parámetros a y b .

$$x(t_n) + \varepsilon_n = a \cos(2\pi f_i t_n) + b \sin(2\pi f_i t_n)$$

Ecuación 38.

Fácilmente se puede ver que la anterior expresión es una particularización de la forma general para señales reales. (Ver ecuación 39)

$$x(t_n) + \varepsilon_n = c(i) e^{j2\pi f_i t_n}$$

Ecuación 39.

La minimización de la varianza de ε_n (error cuadrático medio) lleva también a la minimización de la expresión contenida en la ecuación 40 de la cual se puede tener los valores para cada $c(i)$. (ver ecuación 41)

$$\sum_{n=1}^N |x(t_n) - c(i) b_i(t_n)|^2$$

Ecuación 40.

$$c(i) = \frac{1}{k} \sum_{n=1}^N x(t_n) b_i^*(t_n) \quad k = \sum_{n=1}^N |b_i(t_n)|^2$$

Ecuación 41.

Este resultado obtenido se puede referir como un método generalizado de Lomb para una transformación de señales con datos desigualmente muestreados. La potencia de la señal evaluada en el índice i de la transformación ($P_x(i)$) se puede ver en la ecuación 42.

$$P_x(i) = c(i) \sum_{n=1}^N x^*(t_n) b_i(t_n) = k |c(i)|^2 = \left| \hat{c}(i) \right|^2 \quad \text{con } c(i) = \hat{c}(i) \sqrt{k}$$

Ecuación 42.

En el caso de la transformada de Fourier se tiene que $k=N$ y $P_x(i)=P_x(f)=c^2(f)$. En el su trabajo original, Lomb introduce una demora a la base (seno y coseno en vez de exponenciales), lo cual hace más eficiente el algoritmo de estimación, obteniéndose el periodograma de Lomb normalizado (ecuación 43).

$$P_x(f) = \frac{1}{2\sigma^2} \left\{ \frac{\left[\sum_{n=1}^N (x(t_n) - \bar{x}) \cos(2\pi f(t_n - \tau)) \right]^2}{\sum_{n=1}^N \cos^2(2\pi f(t_n - \tau))} + \frac{\left[\sum_{n=1}^N (x(t_n) - \bar{x}) \sin(2\pi f(t_n - \tau)) \right]^2}{\sum_{n=1}^N \sin^2(2\pi f(t_n - \tau))} \right\}$$

$$\text{teniendo en cuenta que: } \tan(4\pi\tau) = \frac{\sum_n \sin(4\pi\tau_n)}{\sum_n \cos(4\pi\tau_n)}$$

y que σ^2 es la varianza de $x(t_n)$.

Ecuación 43.

Para el desarrollo de una aplicación más eficiente se puede implementar el algoritmo rápido de lomb. Teniendo en cuenta las ecuaciones 44 y 45 se pueden obtener las expresiones contenidas en las ecuaciones 46 y 47 que permiten el cálculo más eficiente del periodograma de Lomb.[37 y 38]

$$S_h \equiv \sum_{n=1}^N \left(x(t_n - \bar{x}) \sin(2\pi f t_n) \right) \text{ y } C_h \equiv \sum_{n=1}^N \left(x(t_n - \bar{x}) \cos(2\pi f t_n) \right)$$

Ecuación 44.

$$S_2 \equiv \sum_{n=1}^N (\sin(4\pi f t_n)) \text{ y } C_2 \equiv \sum_{n=1}^N (\cos(4\pi f t_n))$$

Ecuación 45.

$$\sum_{n=1}^N \left(x(t_n - \bar{x}) \cos(2\pi f (t_n - \tau)) \right) = C_h \cos(2\pi f \tau) + S_h \sin(2\pi f \tau)$$

$$\sum_{n=1}^N \left(x(t_n - \bar{x}) \sin(2\pi f (t_n - \tau)) \right) = S_h \cos(2\pi f \tau) - C_h \sin(2\pi f \tau)$$

Ecuación 46.

$$\sum_{n=1}^N (\cos^2(2\pi f (t_n - \tau))) = \frac{N}{2} + \frac{1}{2} C_2 \cos(4\pi f \tau) + \frac{1}{2} S_2 \sin(4\pi f \tau)$$

$$\sum_{n=1}^N (\sin^2(2\pi f (t_n - \tau))) = \frac{N}{2} - \frac{1}{2} C_2 \cos(4\pi f \tau) - \frac{1}{2} S_2 \sin(4\pi f \tau)$$

Ecuación 47

En el programa *psdlombsenos.mat* se toma una señal compuesta por suma de tres señales senosoidales con frecuencias de 4, 6 y 8 Hz. El programa ilustra el resultado obtenido al calcular el espectro en potencia de la señal por medio del método de la FFT y el método de Lomb. También se puede observar la misma señal pero con muestras tomadas a intervalos desiguales de tiempo con su respectivo espectro en potencia calculado por el método de Lomb. El programa *psdlombsenos.mat* muestra la manera de cómo se realiza el análisis en el dominio de la frecuencia de la señal con muestras tomadas de forma irregular sin la necesidad de usar métodos de interpolación y remuestreo, ofreciendo los mismos resultados que los obtenidos con el método de la FFT aplicado a la señal original con tasa de muestreo periódica.

En la figura 50 se ilustran las dos versiones de la señal manejadas en el programa *psdlombsenos.mat* (muestreo uniforme y no uniforme). En la figura 51 y 52 se puede apreciar el espectro en potencia de la señal muestreada uniformemente calculado por medio de la FFT y por el método de Lomb respectivamente. La figura 53 contiene la señal obtenida por medio de

interpolación y remuestreo a partir de las muestras desigualmente tomadas de la señal original y su espectro en frecuencia calculado por medio de la FFT.

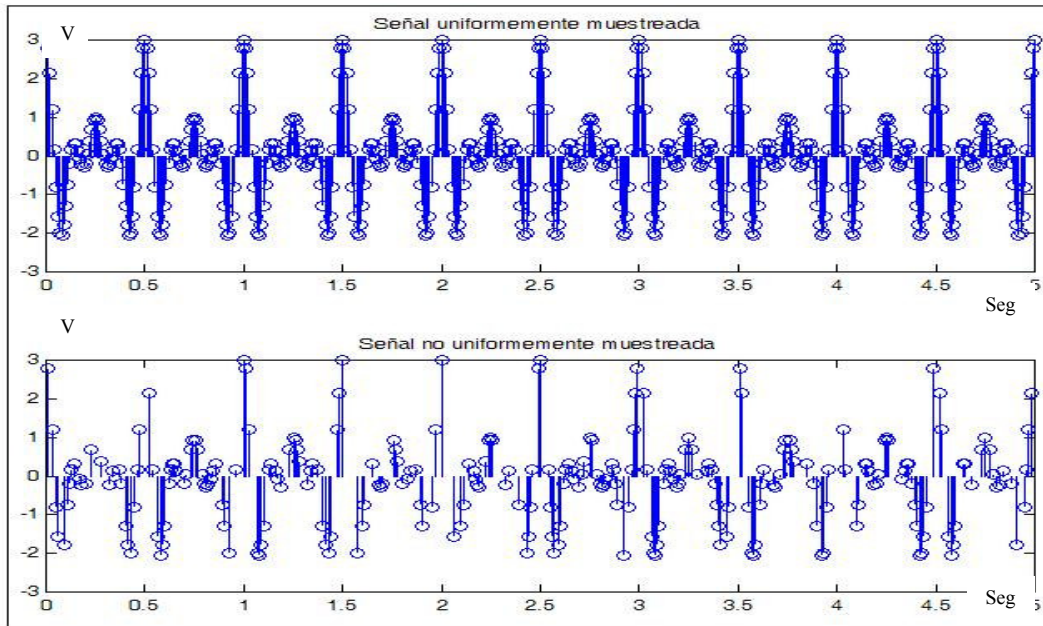


Figura 50. Señal con muestreo uniforme y no uniforme.

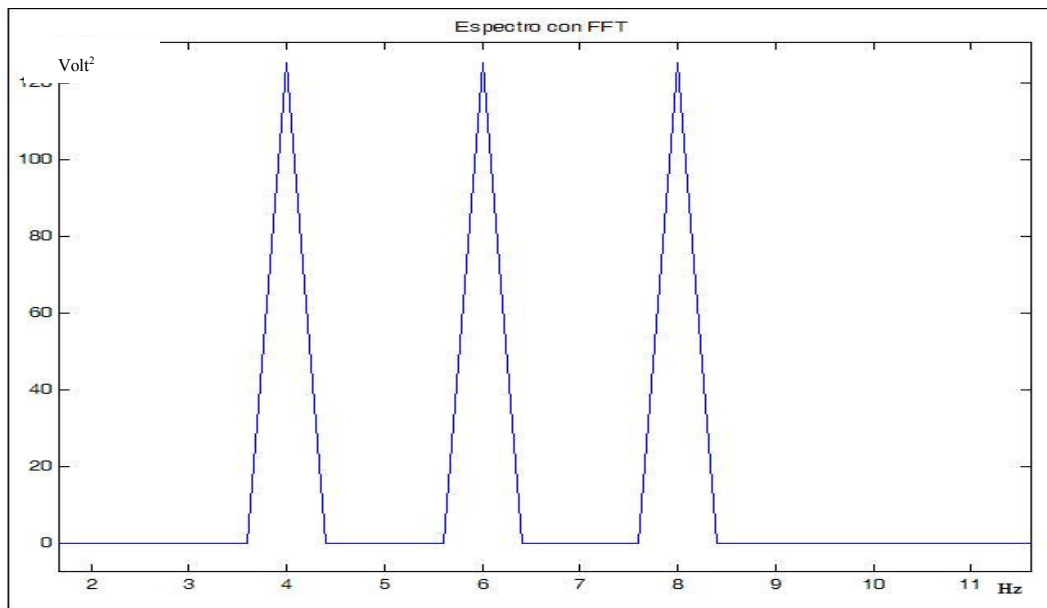


Figura 51. Espectro en potencia con FFT de la señal uniformemente muestreada.

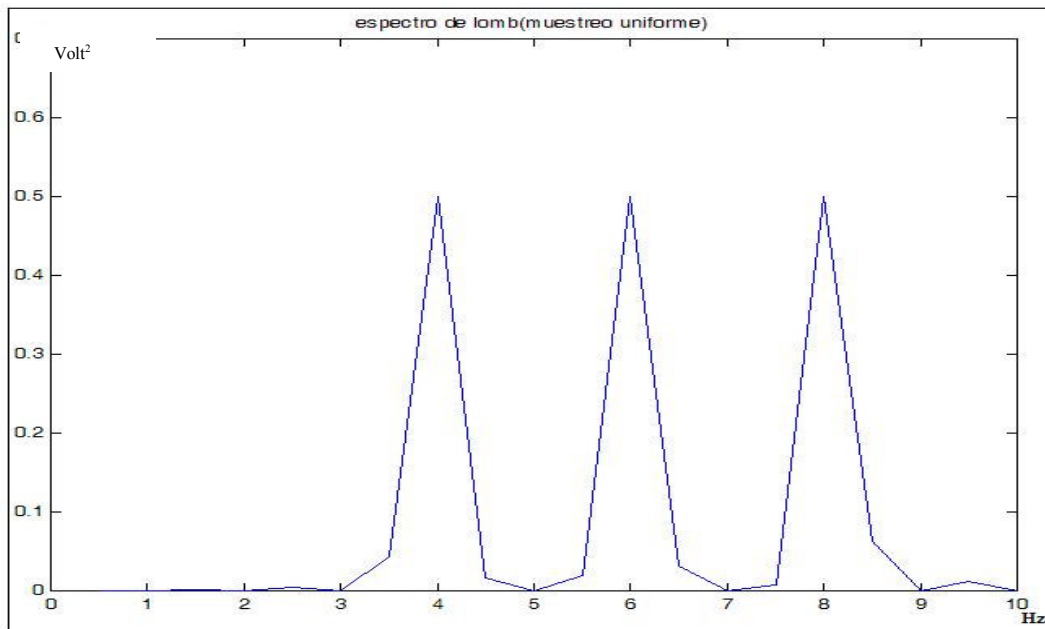


Figura 52. Espectro en potencia por el método de Lomb. Señal uniformemente muestreada.

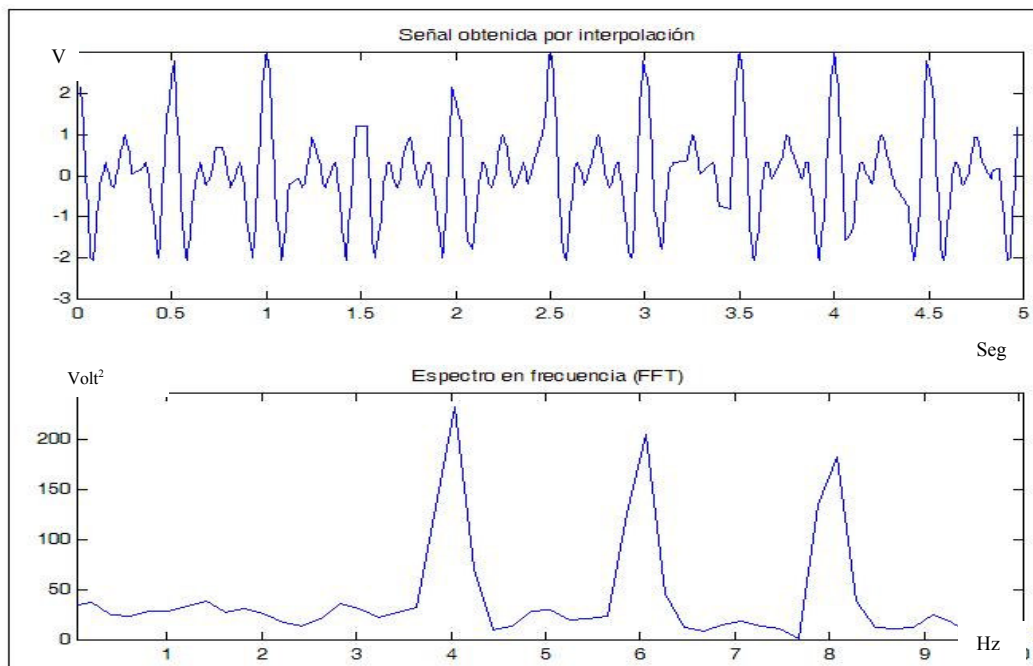


Figura 53. Señal obtenida por interpolación y espectro en frecuencia por método de la FFT.

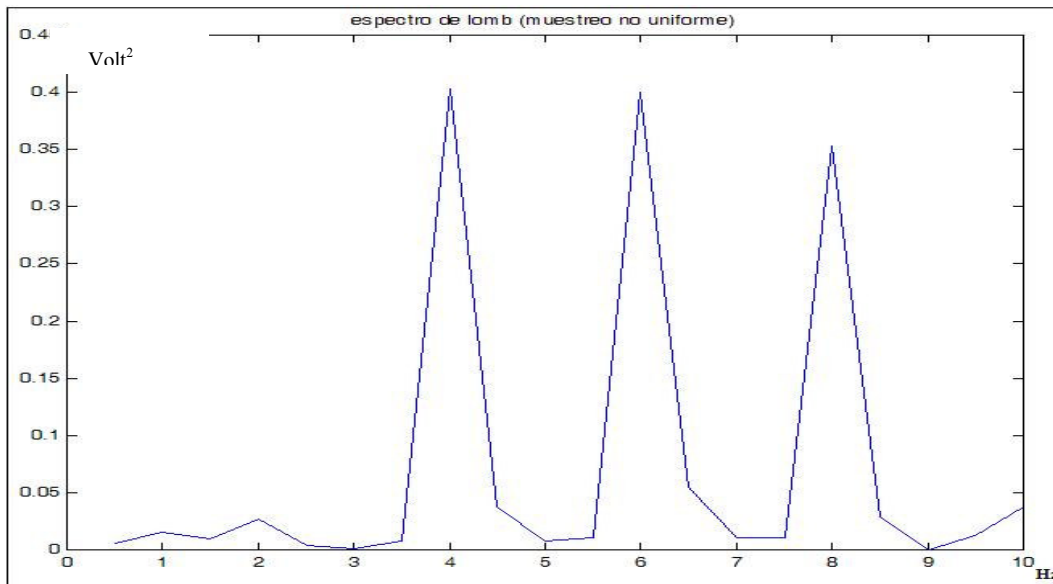


Figura 54. Espectro en potencia por el método de Lomb. Señal no uniformemente muestreada.

En la figura 55 se puede apreciar el resultado obtenido dentro del programa *psdinterpolacion.mat* al realizar la estimación de la PSD de los intervalos RR por medio del método de Lomb. La figura 55 ilustra el efecto pasa bajos introducido en la estimación de la PSD por medio del método de remuestreo y FFT ocasionando una atenuación en las frecuencias altas (0.15 Hz – 0.4 Hz) en relación con las bajas frecuencias.

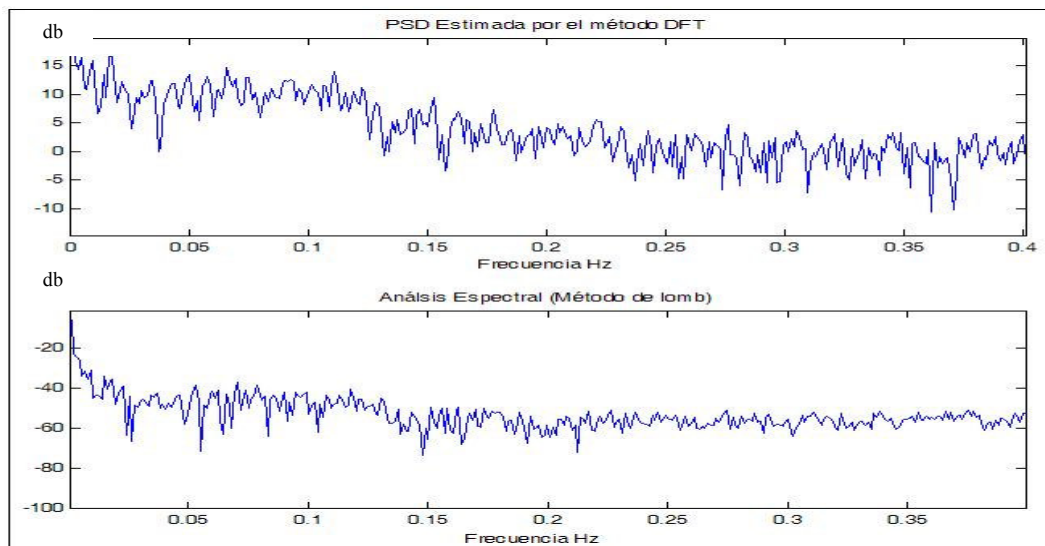


Figura 55. Estimación de PSD de intervalos RR por Medio del método de Lomb.

4. DISEÑO Y ENSAYO DE ALGORITMOS.

En este capítulo se explica el diseño y ensayo de cada uno de los algoritmos implementados en lenguaje C para su respectiva evaluación en la herramienta de desarrollo DSK 5402 del fabricante Texas Instruments® para procesadores DSP de referencia TMS320CV5402.

Estos algoritmos fueron previamente simulados con ayuda del asistente matemático Matlab® utilizando la teoría para el procesamiento digital de la señal electrocardiográfica planteada en los capítulos 1, 2 y 3.

En este trabajo de investigación el procesamiento digital de la señal electrocardiográfica se ha llevado a cabo a través de varias etapas:

1. Adquisición y digitalización de la señal electrocardiográfica.
2. Filtrado adaptativo para remover el corrimiento de la línea de base.
3. Filtrado adaptativo para reducir la interferencia de la señal de 60 Hz.
4. Detección de la onda R del complejo cardiaco.
5. Análisis en el dominio del tiempo de los intervalos RR.
6. Análisis en el dominio de la frecuencia de los intervalos RR.

4.1 ADQUISICIÓN Y DIGITALIZACIÓN DE LA SEÑAL ELECTROCARDIOGRÁFICA.

Teniendo en cuenta las recomendaciones descritas en la sección 1.3 se ha procedido a implementar una etapa de acondicionamiento que permite tener una señal electrocardiográfica con los valores de amplitud suficiente para ser manejada por la plataforma DSK5402. La salida del circuito de acondicionamiento se encuentra conectada directamente a la entrada del circuito para conversión análogo a digital (CODEC) de la plataforma DSK5402 anulando los efectos del filtro pasa altas para evitar la distorsión de la señal electrocardiográfica. (Ver figura 56)

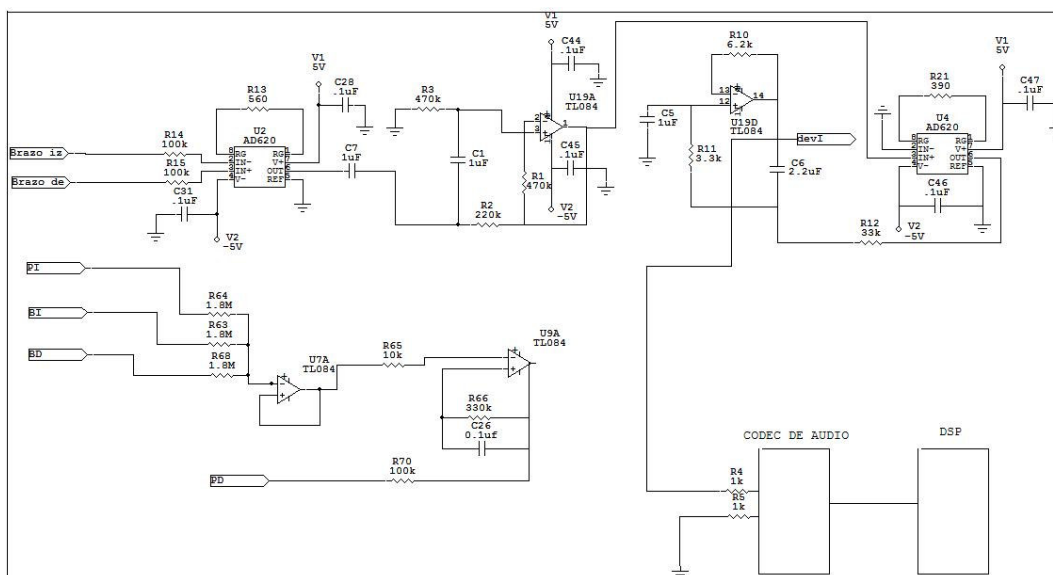


Figura 56. Etapa de acondicionamiento.

Según el algoritmo planteado en el programa diezclado.mat se ha implementado en la plataforma DSK 5402 el proceso de digitalización de la señal adquiriendo 8000 muestras a una frecuencia de muestreo de 2KHz y diezclado ($n=4$) el cual entrega a la salida una señal de 2000 muestras con una frecuencia de muestreo de 500Hz (ver figura 57). Con esta cantidad de muestras se logrará adquirir 4 segundos de señal electrocardiográfica, suficiente para poder visualizar cuatro ciclos cardiacos de un paciente con una frecuencia cardiaca de 60 latidos por minuto.

El código de este algoritmo se encuentra contenido en el archivo *decimación.c*. Para probar el funcionamiento del algoritmo de adquisición, se ha realizado una prueba en la cual se ha adquirido a través del CODEC del DSK5402 una señal senoidal de 10 HZ con una tasa de muestreo con valor de 2Khz y de la realización del proceso de diezclado. (ver figura 58)

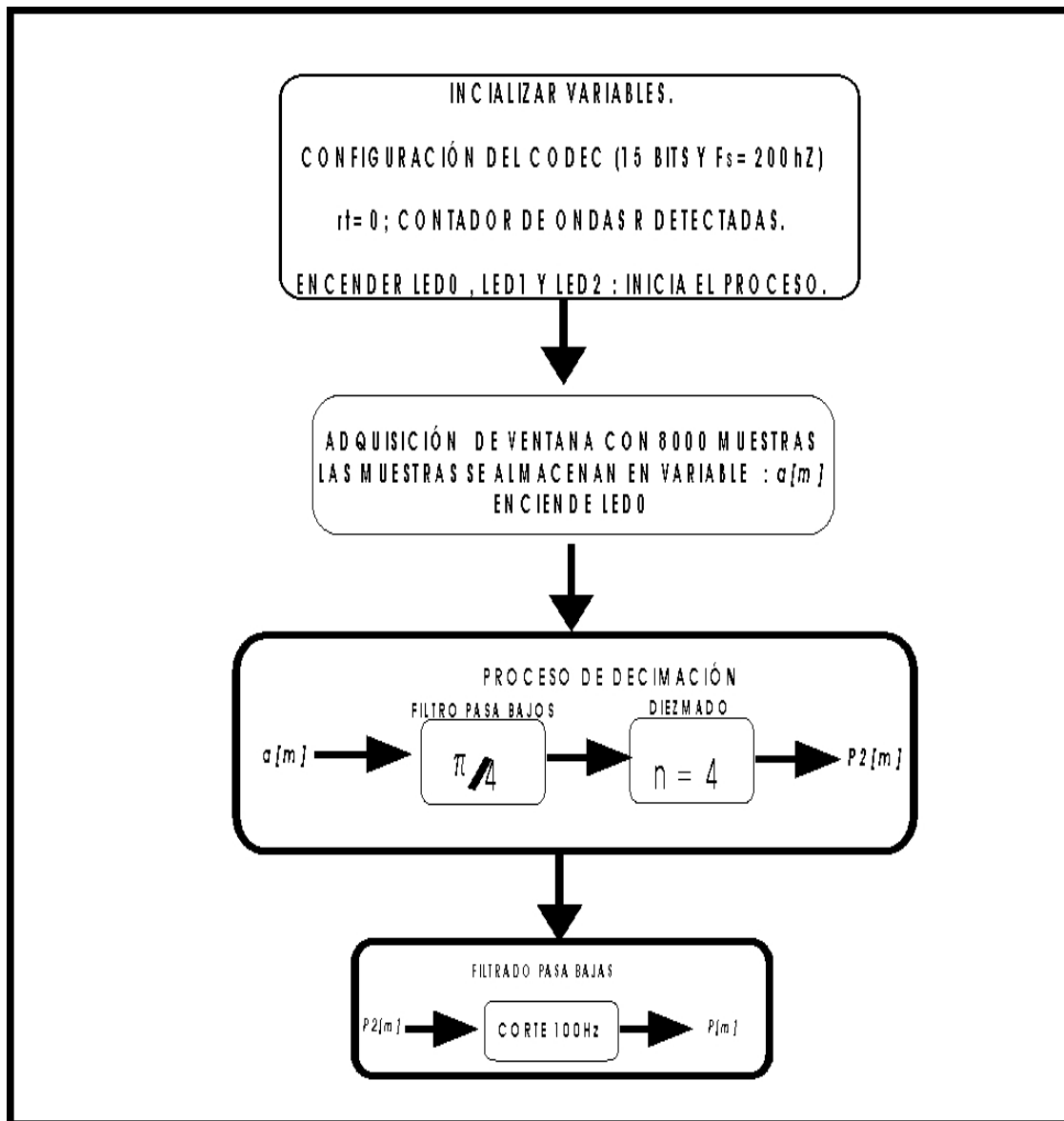
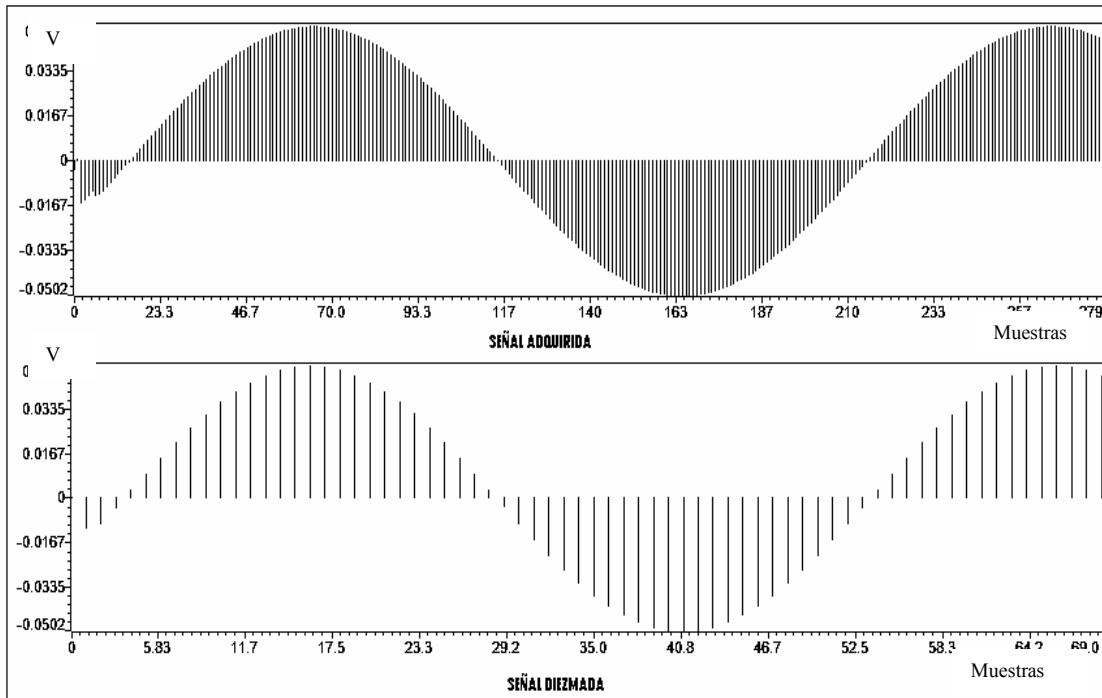


Figura 57. Algoritmo de Digitalización y diezclado.



**Figura 58. Señal adquirida y diezmada
Con el DSK 5402.**

4.2 FILTRADO ADAPTATIVO PARA REMOVER EL CORRIMIENTO DE LA LÍNEA DE BASE.

Por medio de las líneas de código en Matlab® contenidas en el programa *lineadabase.mat* (mencionado en el ítem 2.2.4) se ha utilizado la estrategia de filtrado adaptativo basado en un sistema de cancelación ilustrado en la figura 59 para ser implementado en lenguaje C un algoritmo de filtrado adaptativo basado en un filtro por cancelación para la remoción del corrimiento de línea de base en una señal electrocardiográfica. (Ver figura 60)

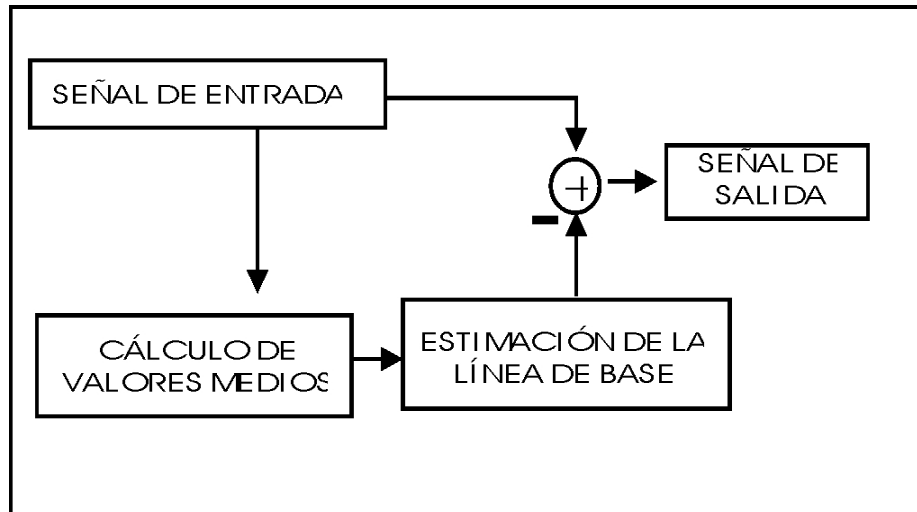


Figura 59. Filtrado adaptativo por cancelación para remoción del corrimiento de la línea de base.

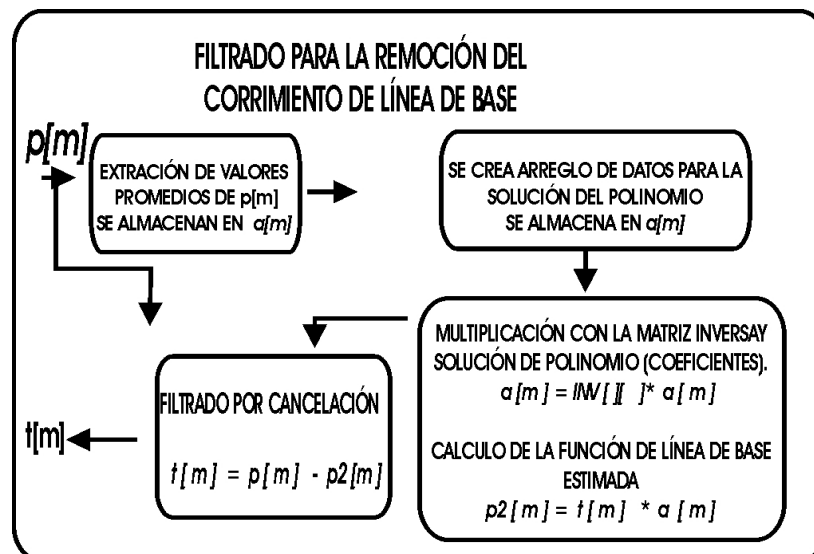


Figura 60. Algoritmo de Filtrado adaptativo para la remoción de corrimiento de la línea de base.

Las líneas de código en lenguaje C del filtro adaptativo para la remoción del corrimiento de línea de base se puede detallar en el programa *filtro línea de base.c* donde se pueden ver los diferentes pasos que se ejecuta el algoritmo ilustrado en la figura 61. En el programa se tiene en cuenta que la frecuencia de muestro de la señal que se desea procesar es de 500 Hz con un tamaño de 2000 muestras y se ha seleccionado una ventana de 200 muestras para el cálculo de los valores medios (se toma un valor medio cada 0.4 segundos de la señal).

En la figura 62 se encuentra una muestra una señal senosoidal con frecuencia de 39 Hz sumada a una señal de muy baja frecuencia (0.1 Hz) y su espectro en frecuencia se ilustra en la figura 50. La función estimada de la línea de base y su espectro en frecuencia se pueden observar en las figura 63 y 64 respectivamente. El resultado final del proceso de remoción del corrimiento de línea de base se encuentra ilustrado en las figuras 65 y 66 .

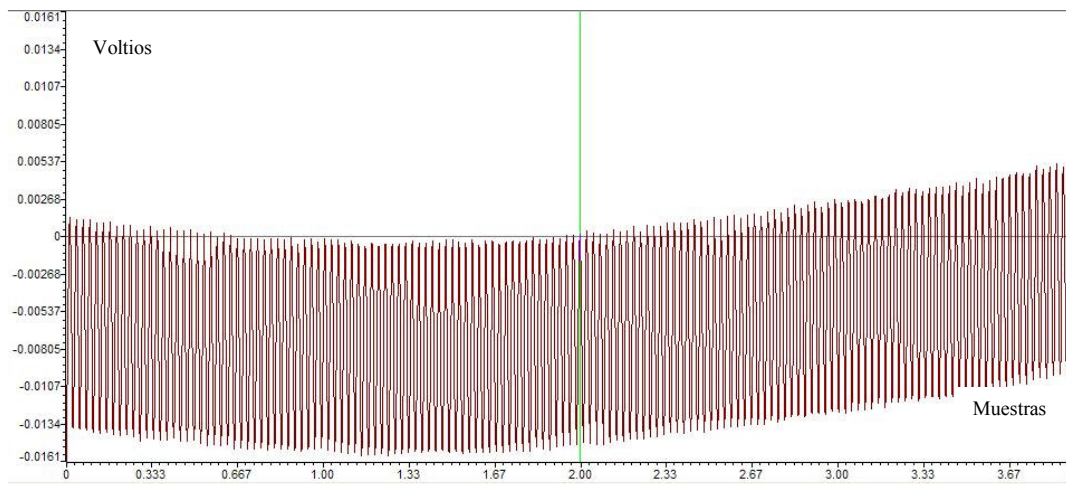


Figura 61. Señal senosoidal con corrimiento de línea de base.

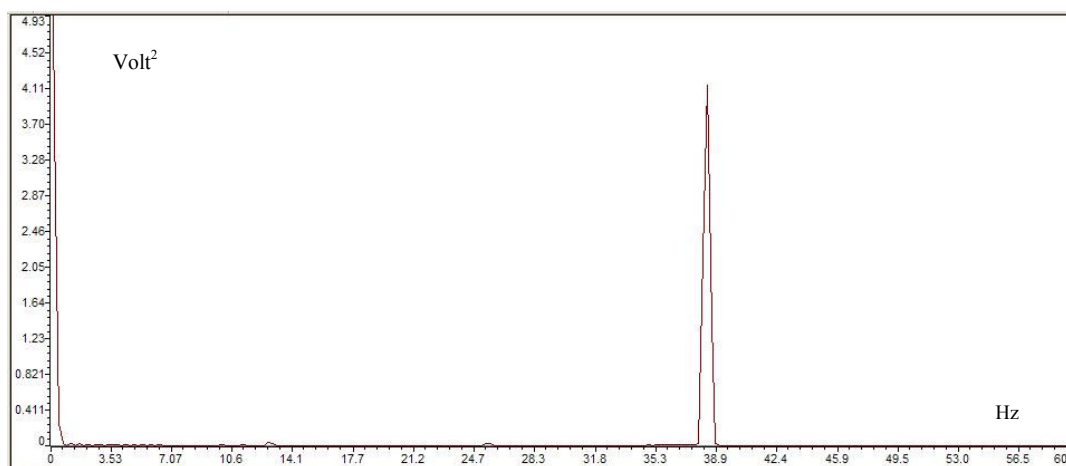


Figura 62. Espectro de la señal senosoidal con corrimiento de la línea de base.

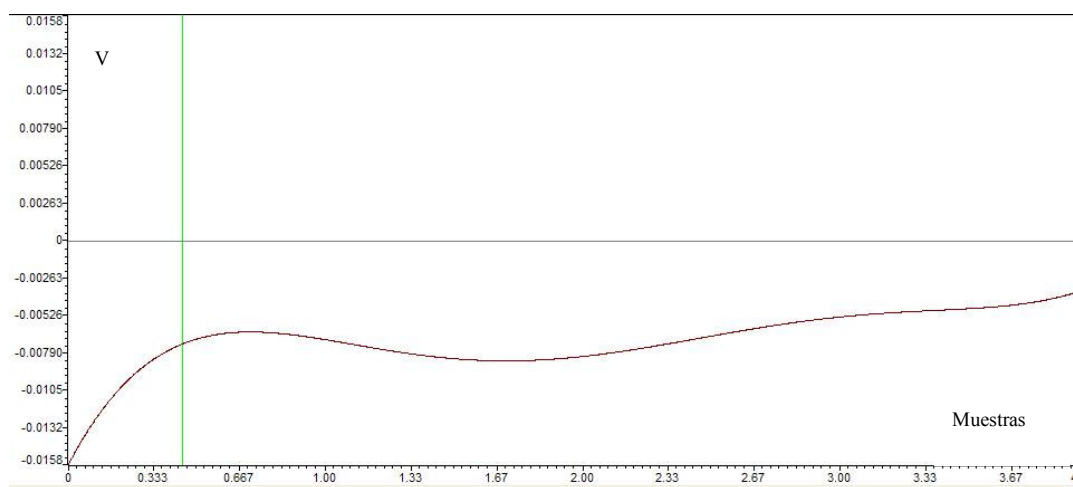


Figura 63. Función estimada de la línea de base.

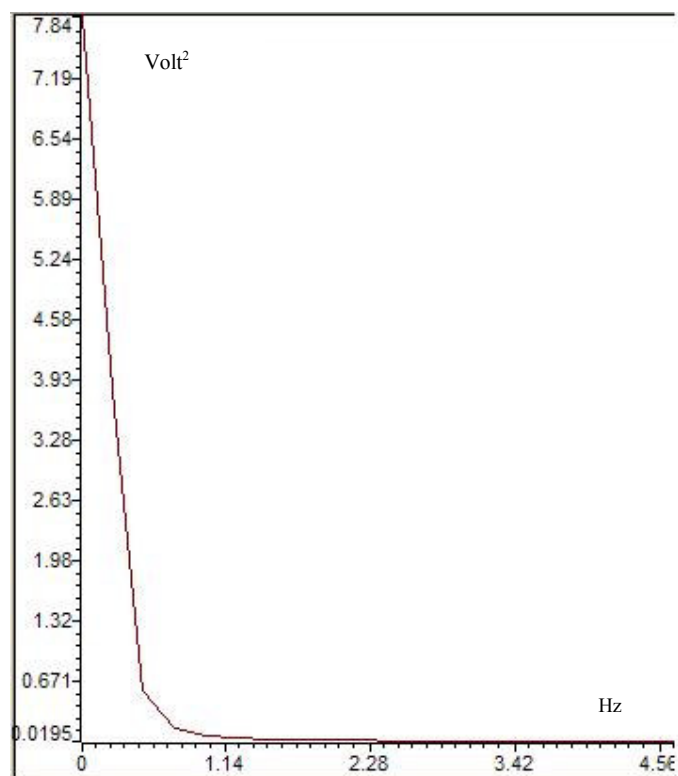


Figura 64. Espectro de la función estimada de la línea de base.

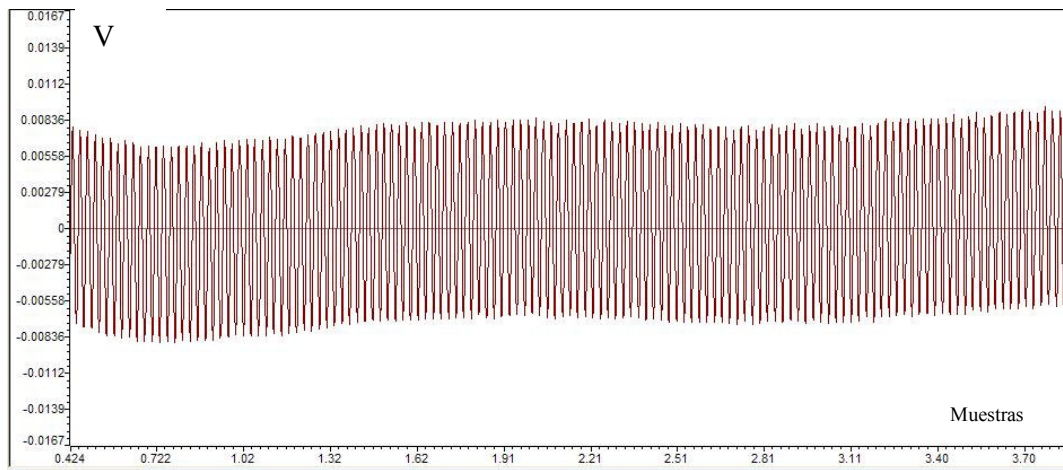


Figura 65. Señal senosoidal filtrada.

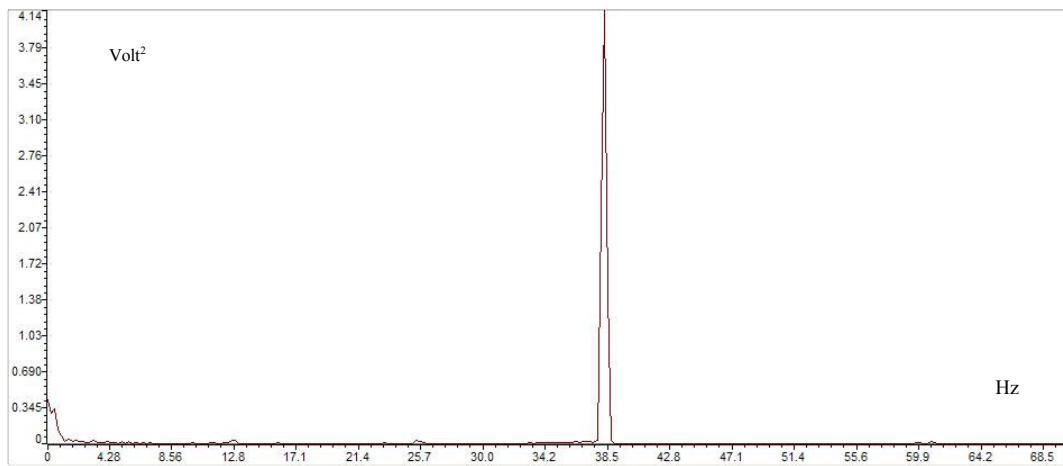


Figura 66. Espectro de la señal filtrada.

4.3 FILTRADO ADAPTATIVO PARA REDUCIR LA INTERFERENCIA DE LA SEÑAL DE 60 HZ.

Según la teoría descrita en la sección 2.2 se ha diseñado un filtro FIR notch de segundo orden adaptativo para la remoción de la interferencia de 60 Hz presente en la señal electrocardiográfica adquirida.

Según la ecuación 13 detallada en la sección 2.2.1, se pueden calcular el valor de los coeficientes del filtro FIR *notch* teniendo en cuenta que la frecuencia de muestreo de la señal es de 500 Hz:

$$\beta_0 = \left[\frac{\omega_0}{\omega_s} \right] 2\pi = \left[\frac{2\pi 60}{2\pi 500} \right] 2\pi = 0.753982236$$

$$w_0 = 1,$$

$$w_1 = -2\cos(\beta_0) = -2\cos(0.753982236) = -1.457937256.$$

$$w_2 = 1.$$

Para el diseño de este filtro solo es necesario el cálculo de un solo coeficiente (w_1) dando el valor de 1 a w_0 y w_2 . El consumo de memoria del sistema será mínimo ya que solo requiere un espacio para la variable w_1 que es el coeficiente que será adaptado.

Según la ecuación 15 la fórmula no recursiva del filtro FIR notch de segundo orden es:

$$y(n) = x(n) - 1.457937256x(n-1) + x(n-2)$$

En la figura 67 se puede observar la respuesta en frecuencia del filtro FIR notch.

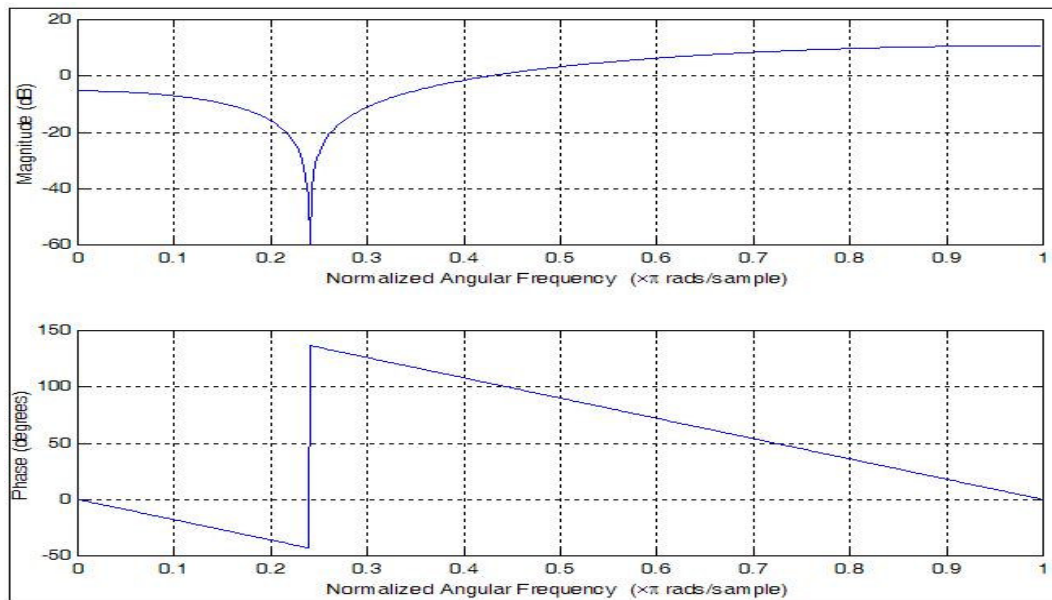


Figura 67. Respuesta en frecuencia Filtro FIR notch.

Con base en el filtro notch diseñado anteriormente, se procede a implementar el sistema para el tratamiento del ruido de 60 Hz y el sistema por cancelación de ruido y evaluar su desempeño para que posteriormente sea implementado el filtro FIR notch adaptativo.

Por medio de las siguientes líneas de código en Matlab® contenidas en el programa *fircanreferencia.mat* se ha simulado preliminarmente el proceso del sistema por cancelación para estimar la señal de error utilizando una señal compuesta por la suma de diferentes señales senosoidales con una interferencia aditiva de 60 Hz con el fin de evaluar el verdadero desempeño del sistema. Posteriormente a esta prueba se procedió a implementar en la plataforma de desarrollo DSK5402 el filtro adaptativo FIR notch de tercer orden utilizando el filtro por cancelación.

La respuesta en el dominio de la frecuencia de $G(z)$ se puede ver en la figura 68. Los espectros en potencia de la señal original y con la interferencia de 60 Hz acompañados del espectro resultante del proceso del sistema por cancelación se pueden apreciar en la figura 69.

Como parte de este ensayo se ha diseñado un filtro FIR notch convencional de orden 50 y ha sido aplicado con intención de remover la interferencia. La comparación entre al resultado obtenido entre las dos técnicas de filtrado se puede observar en la figura 70.

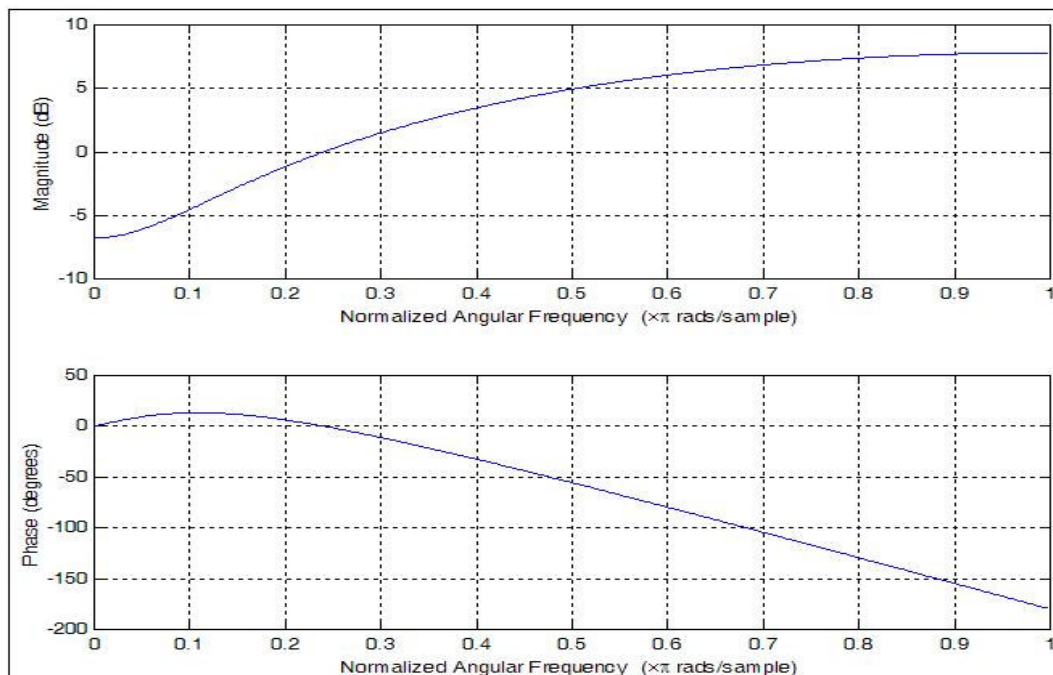


Figura 68. Respuesta en frecuencia de la función estimadora.

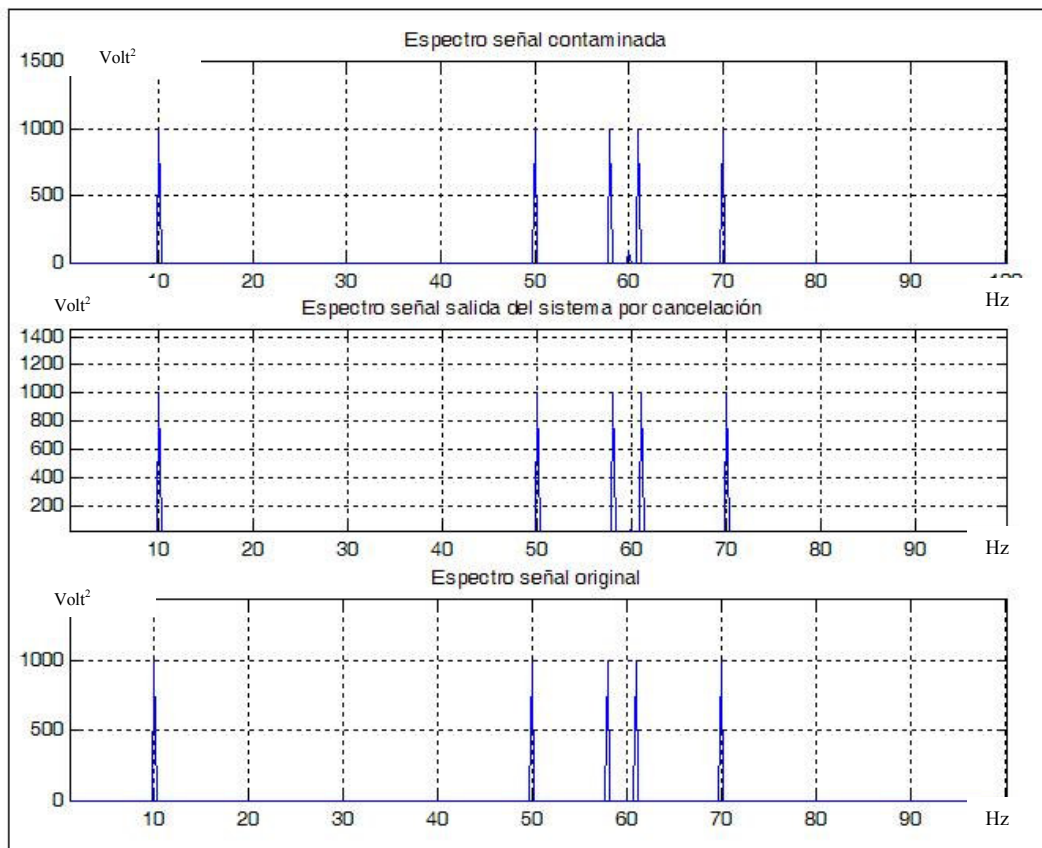


Figura 69. Señal original, señal contaminada y resultado del $e(n)$ Obtenido con el sistema por cancelación.

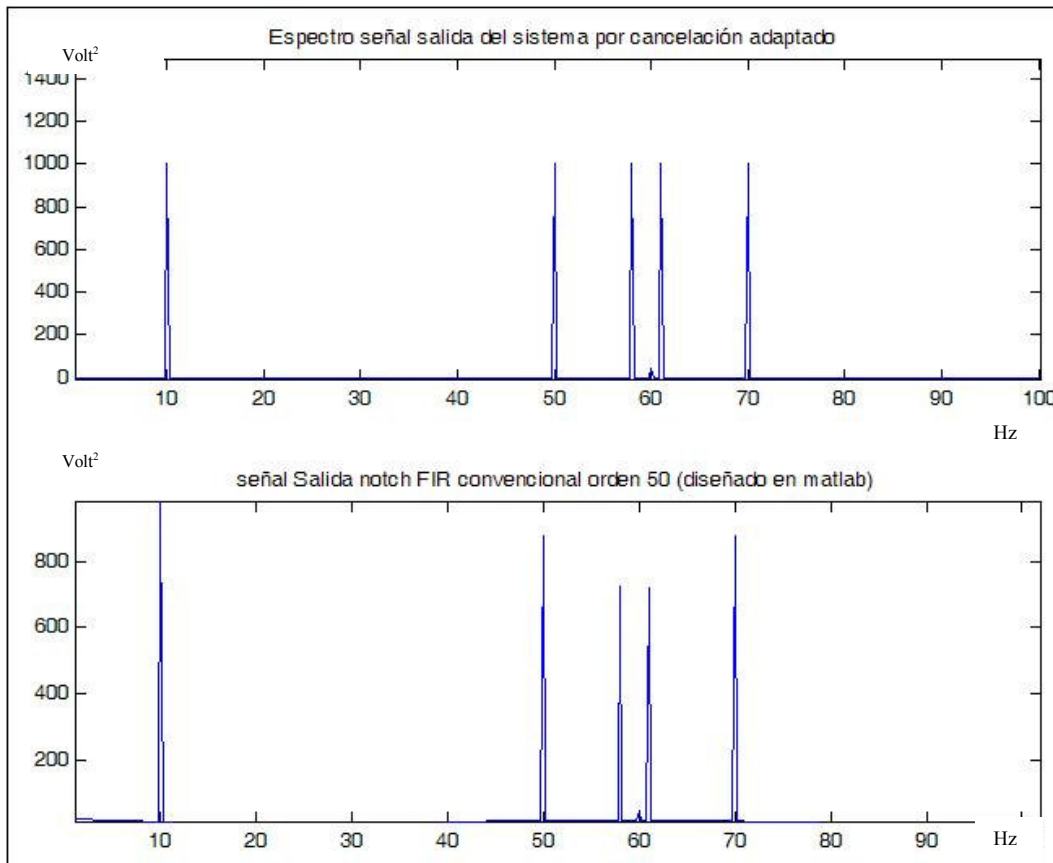


Figura 70. Comparación de resultados entre el sistema por cancelación y por medio de un filtro FIR notch convencional de orden 50.

Tomando la misma señal (con frecuencia de muestreo $f_s = 500$ Hz) compuesta por la suma de funciones senosoidales de frecuencias 10, 50, 58, 61 y 70 Hz, con una interferencia aditiva ($V_1(n)$) de 60 Hz se puede implementar el algoritmo de filtrado adaptativo utilizando las relaciones expresadas en las ecuaciones planteadas en la sección 2.2.3 para la estimación de la función V_2 . En el siguiente código del programa *fircanrefestimada.mat*, se ha implementado en Matlab el método de estimación de la función V_2 basado en el cálculo de los coeficientes de Fourier.

En la figura 71 se puede observar el espectro en frecuencia de la señal contaminada junto con la señal original y el resultado obtenido con el filtrado adaptativo.

En la figura 72 se observa el resultado obtenido por el sistema cuando se realiza una prueba con la señal de muestra sin interferencia ($V_1(n)=0$).

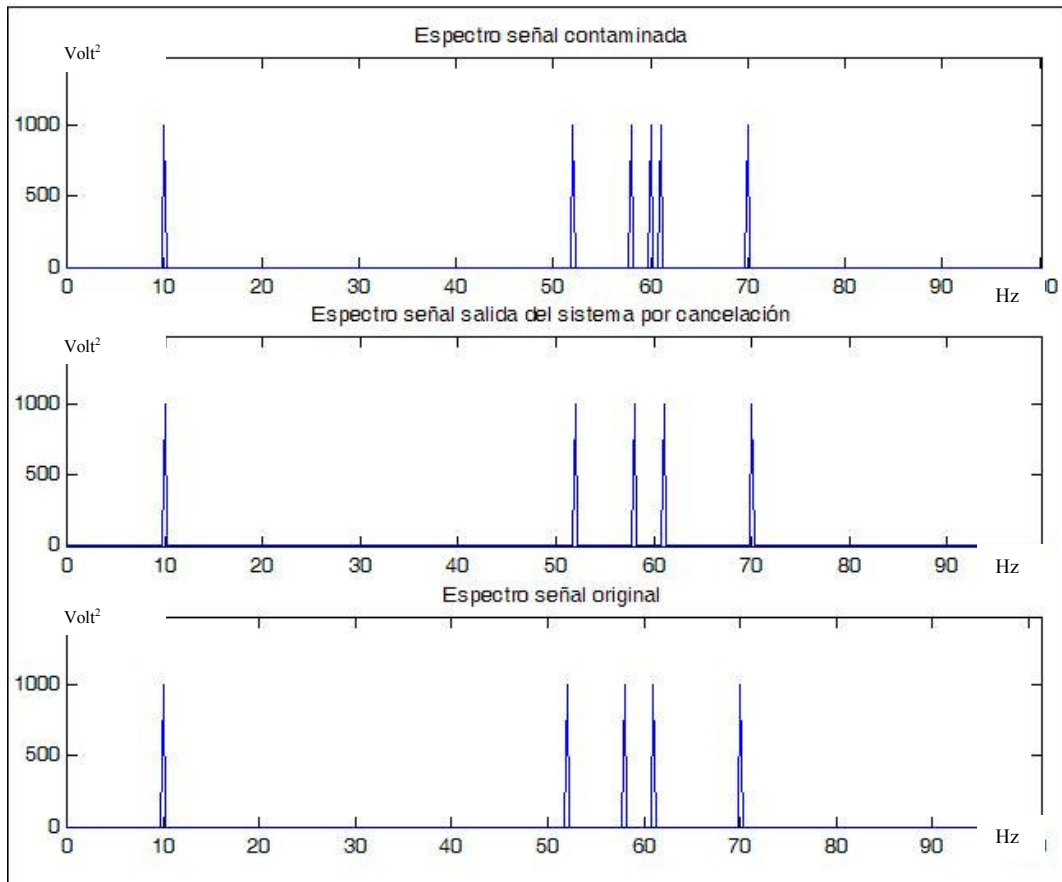


Figura 71. Espectro de señales obtenidas con la función estimada por Análisis de Fourier.

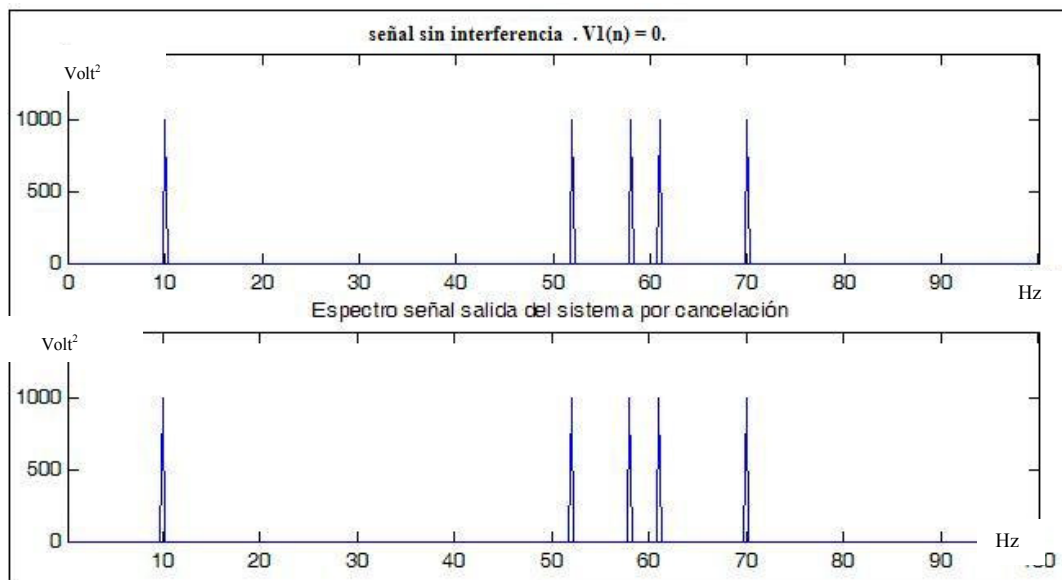


Figura 72. Espectro de señales obtenidas con la función estimada por Análisis de Fourier con V1(n)=0.

En casos reales se puede asumir que el valor de la frecuencia de la señal de interferencia puede tener variaciones en el transcurso del tiempo. En el caso de la señal de las líneas de potencia se ha considerado comúnmente que esta variación de frecuencia puede ser asumida como Gausiana.[23] Esto sugiere que la distribución puede ser o no centrada en los 60 Hz y la desviación de la frecuencia no siempre se encuentra en una sola dirección.

Una manera sencilla de simular variaciones en el valor de la frecuencia de una señal de 60 Hz es a través de una función rampa que permita que el una variación de ± 0.1 Hz. Esta simulación se encuentra contenida en el programa *simulacionrampa.mat*. (Ver figura 73)

En la figura 74 se puede apreciar la diferencia entre el espectro en potencia de una señal de 60 Hz con frecuencia fija y la misma señal pero con una variación en su valor de frecuencia que depende de una función rampa.

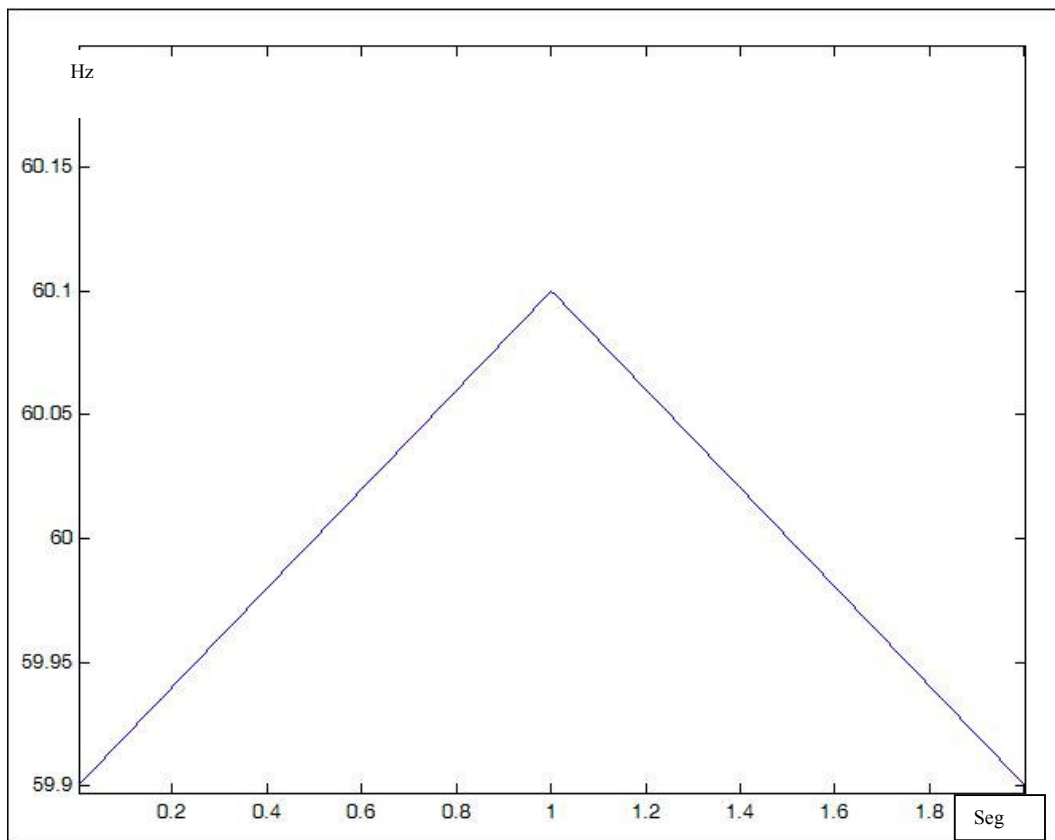


Figura 73. Valor de frecuencia central que depende de una función rampa.

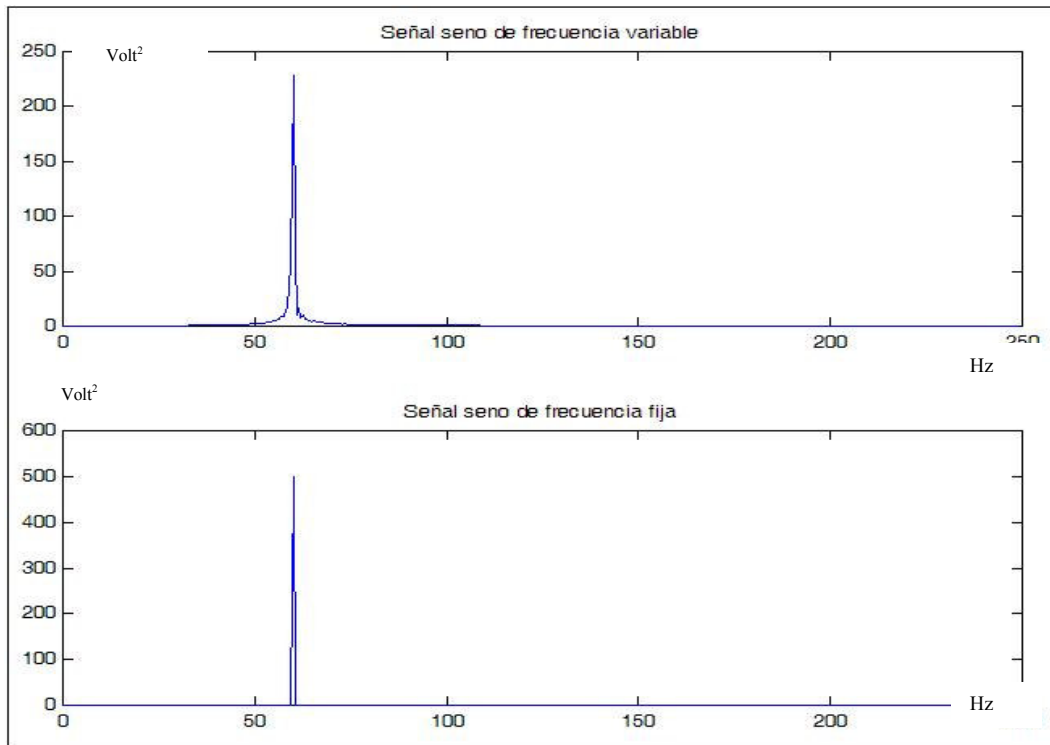


Figura 74. Magnitud del espectro en Potencia para una señal con frecuencia fija y variable.

Utilizando la señal compuesta por funciones senos de frecuencias 10, 50, 58, 61 y 70 Hz se puede simular el efecto de una interferencia con frecuencia central variable (ver figura 75).

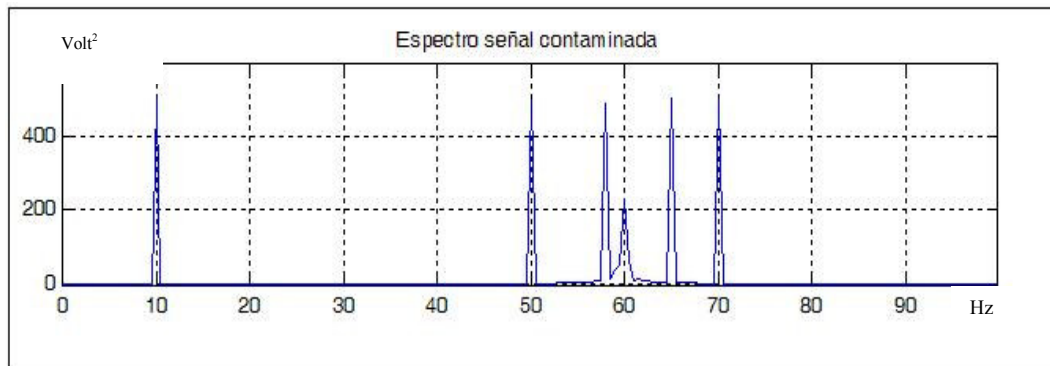


Figura 75. Magnitud del espectro en potencia Para una señal contaminada con ruido de 60 Hz con frecuencia variable.

Como estrategia para mejorar la respuesta del sistema de cancelación ante variaciones del valor de la frecuencia central se puede proponer un nuevo diseño de la función estimada V_2 que contemple cambios de frecuencia.

En la figura 76 se ilustra el resultado de utilizar el sistema por cancelación en presencia de una interferencia aditiva con cambios de frecuencia.

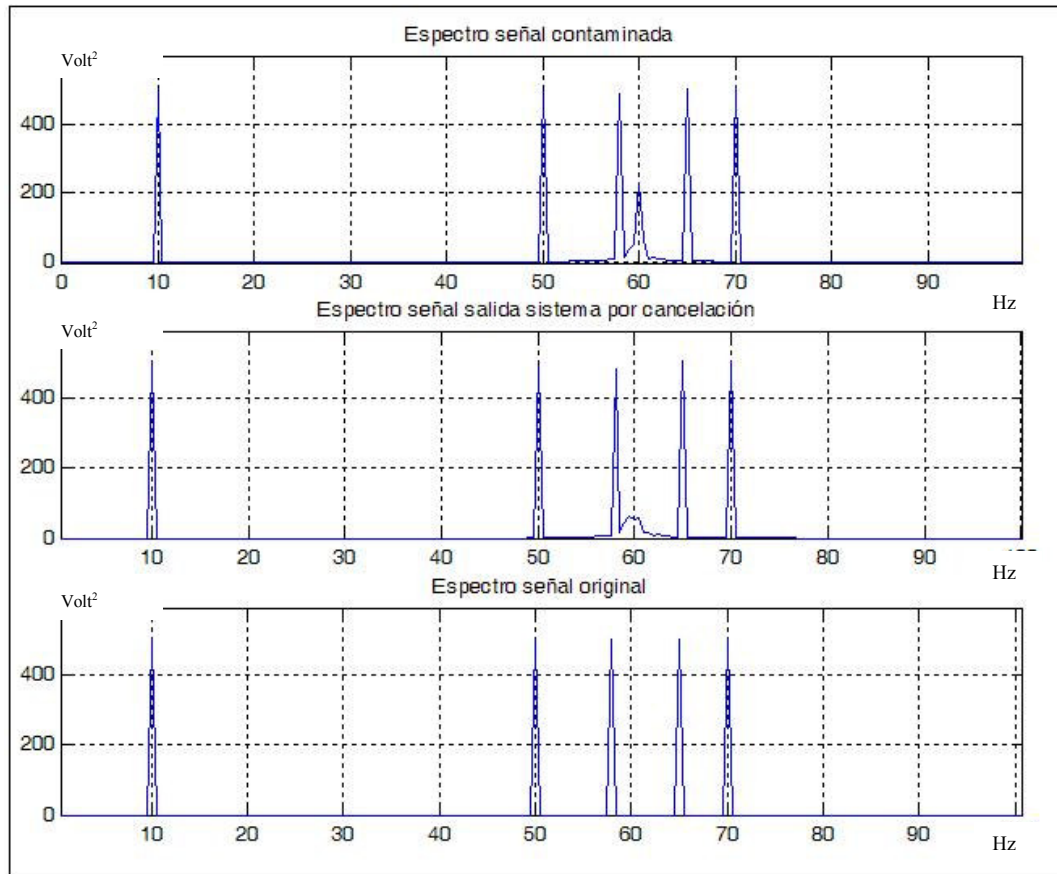


Figura 76. Resultado obtenido con sistema por cancelación.

En el programa *simulacionseno.mat* se ha simulado el efecto del algoritmo de filtrado utilizando una variación sinusoidal en su valor frecuencia de la interferencia.

En la figura 77 se puede apreciar el resultado obtenido con este nuevo diseño de la función estimada V_2 . La figura 78 contiene una comparación entre el filtrado adaptativo por cancelación y el filtrado notch convencional de orden 50.

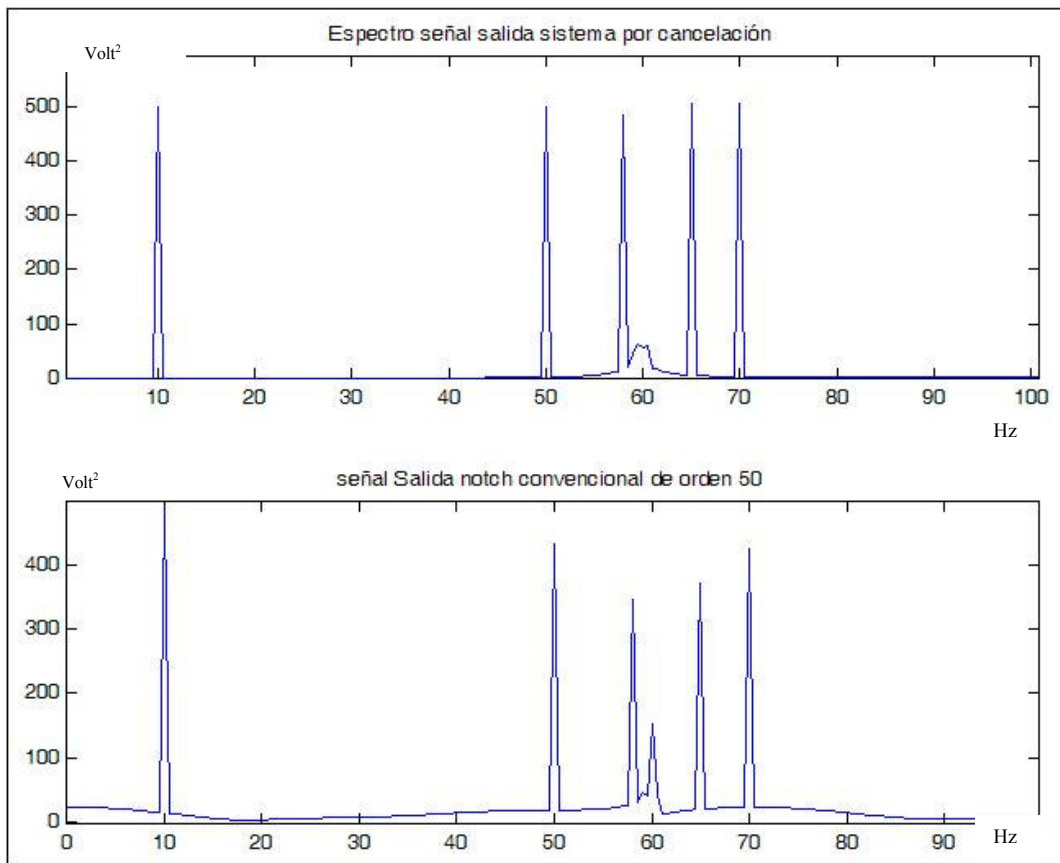


Figura 77. Resultado obtenido con el modelo senosoidal en la frecuencia de la interferencia.

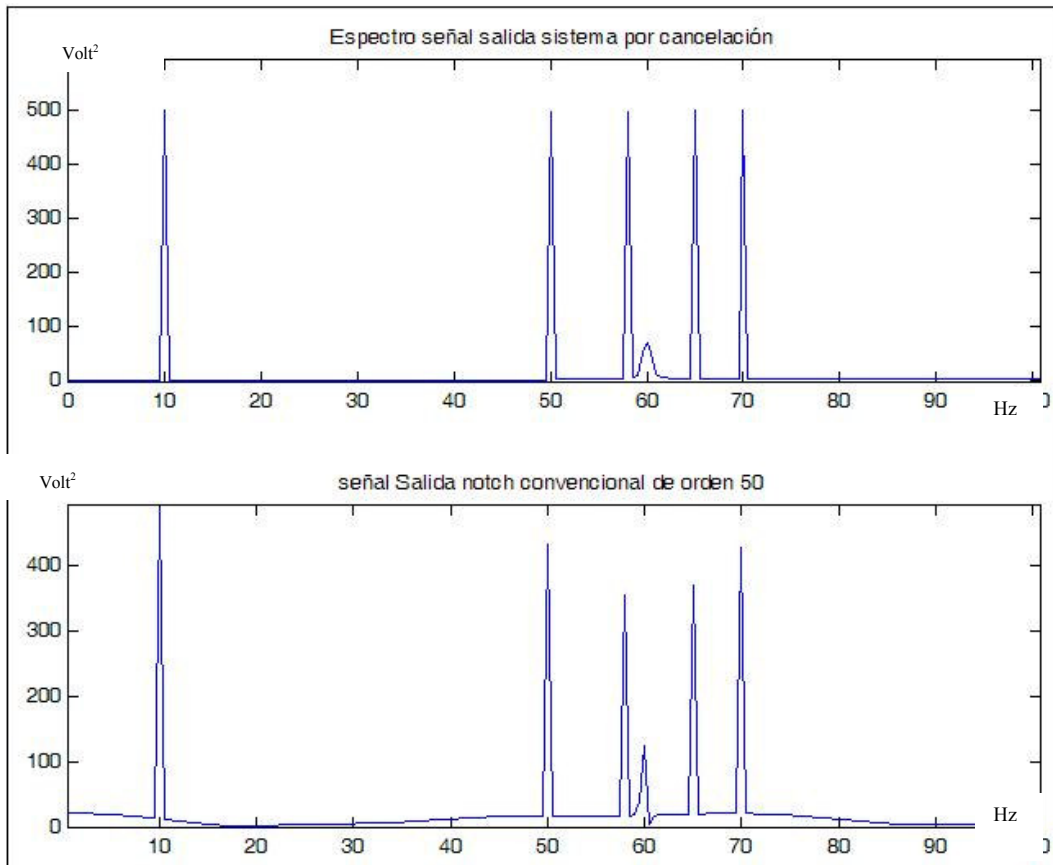
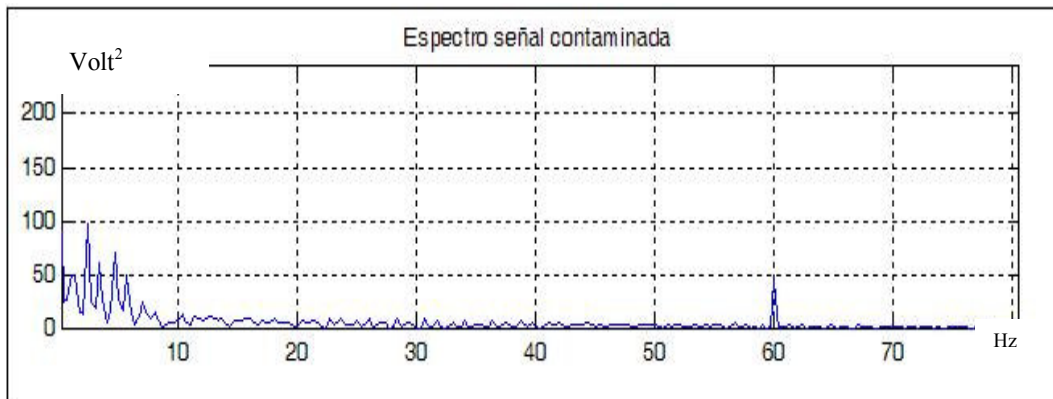


Figura 78. Resultado comparativo entre el sistema adaptativo y convencional.

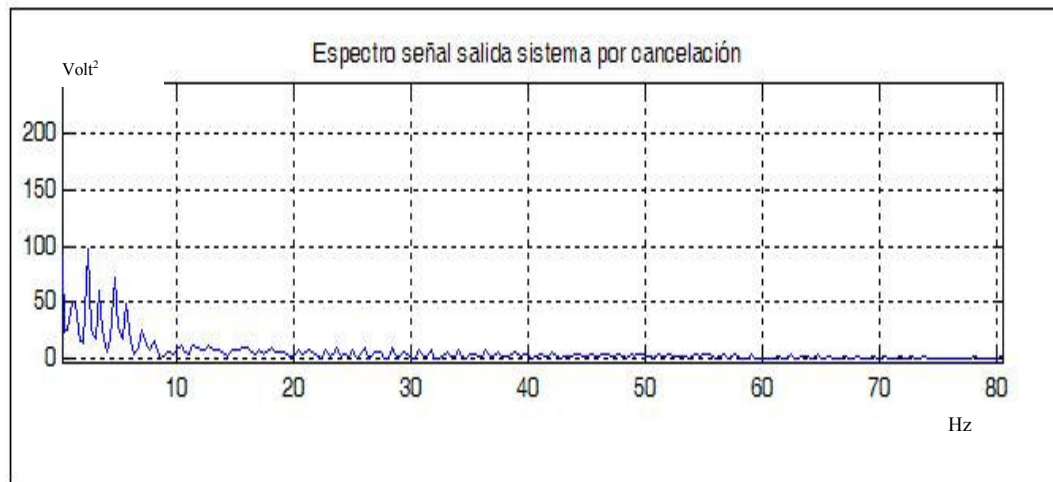
Con el fin de evaluar el desempeño del algoritmo de filtrado adaptativo ante la presencia de una interferencia de 60 Hz con frecuencia fija, se ha elaborado el programa *evalfrecfija.mat* donde se toma la señal de entrada $X(n)$ que corresponde a la señal ECG tomada de una base de datos con frecuencia de muestreo de 333.333 Hz y la somete a la operación de resta junto con la señal V2 procesada a través de la función estimadora $G(Z)$. Esta operación de resta dará como resultado la señal que es utilizada en el algoritmo de adaptación del coeficiente $a1$ junto con la constante $\mu \ll 1$. [23]

En la figuras 79, 80, 81 y 82 se pueden apreciar los resultados.

Por medio del programa *evalfrecseno.mat* se realiza la simulación de contaminación de una señal cardiaca con ruido de 60 Hz con frecuencia central que varía según una función seno. Se pueden visualizar los resultados obtenidos en las figuras 83 y 84, donde se ilustra, respectivamente en el dominio del tiempo y de la frecuencia.



**Figura 79. Espectro señal ECG contaminada con interferencia de 60 Hz
De frecuencia fija.**



**Figura 80. Salida Del Sistema Por Cancelación (Domino De La
Frecuencia).**

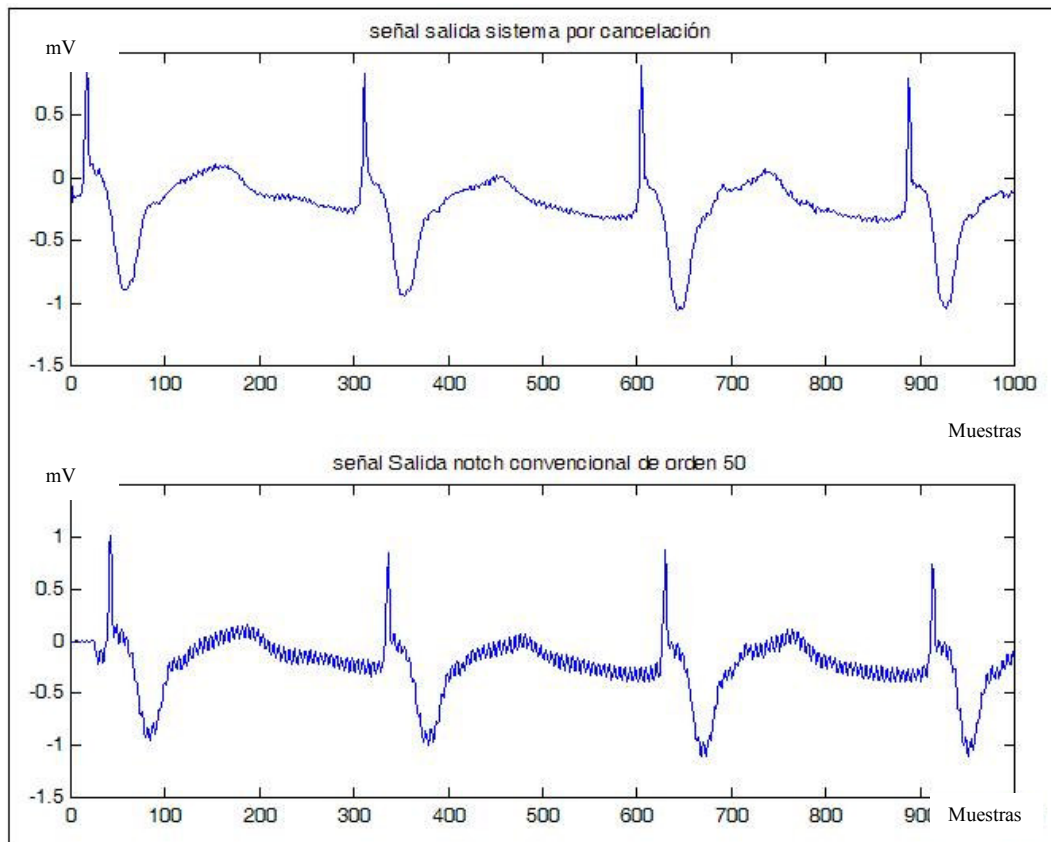


Figura 81. señal filtrada con filtro adaptativo y por filtro notch convencional orden 50. (Dominio del tiempo)

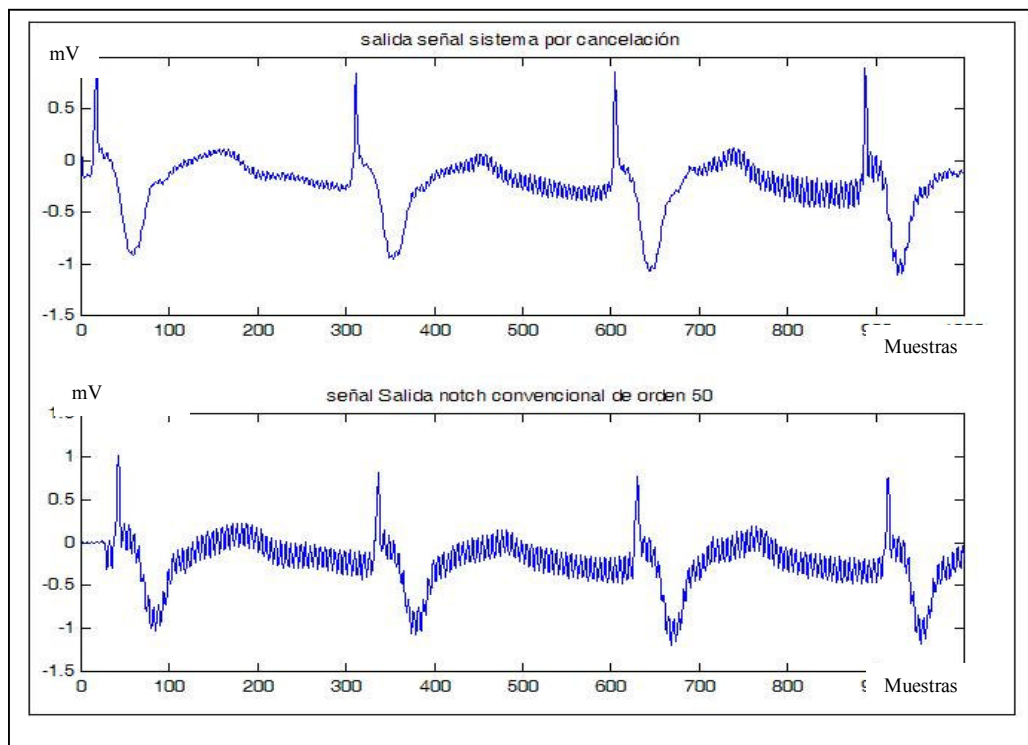


Figura 82. Salida sistema por cancelación y por filtro notch convencional orden 50. (Dominio de la frecuencia)

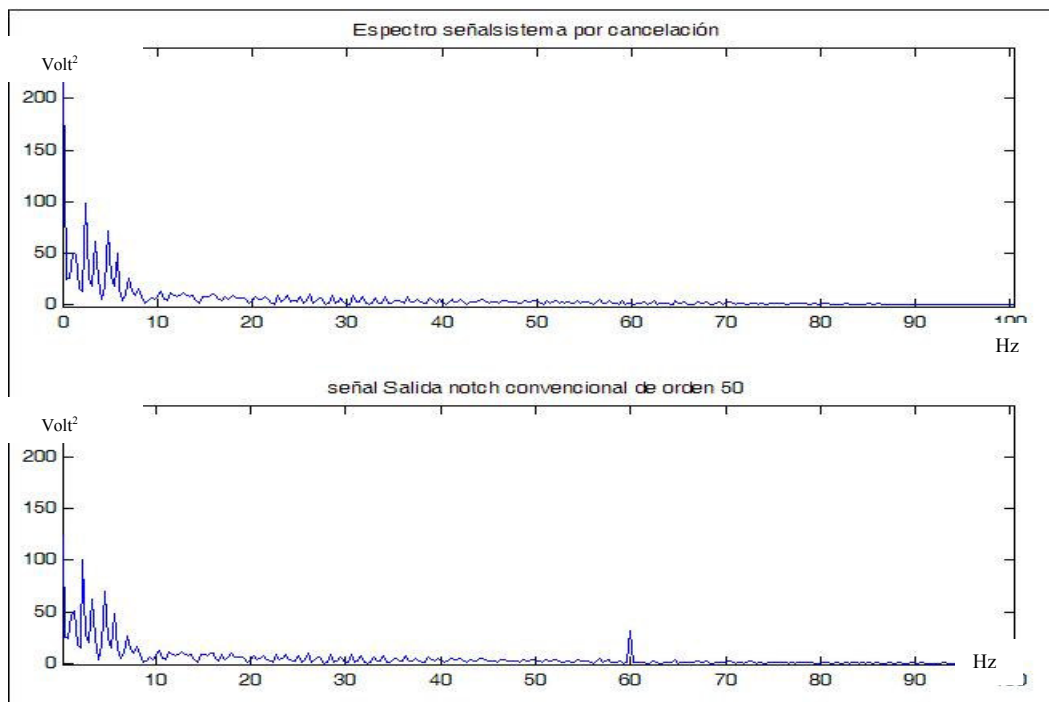


Figura 83. Señal ECG contaminada con interferencia de frecuencia variable y resultado obtenido con el sistema por cancelación y con filtro notch convencional de orden 50. (Dominio del tiempo).

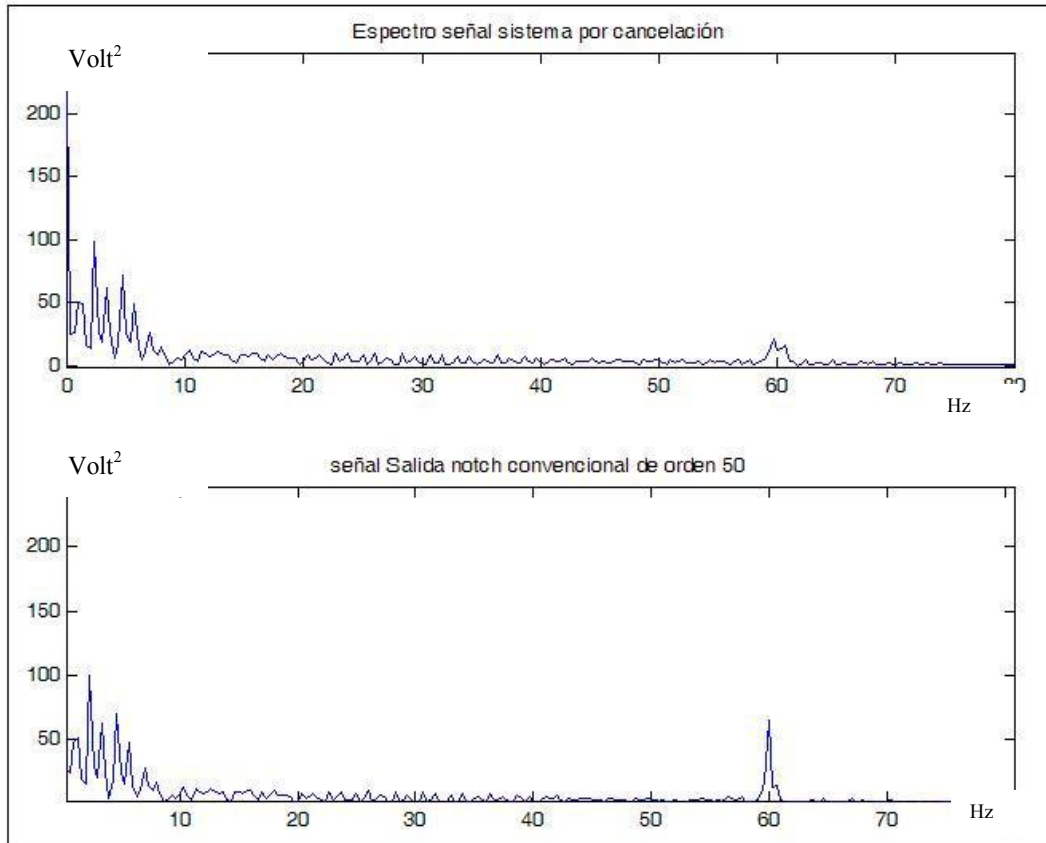


Figura 84. Señal ECG contaminada con interferencia de frecuencia variable y resultado obtenido con el sistema por cancelación y con filtro notch convencional de orden 50. (Dominio de la frecuencia).

Para la implementación en la plataforma DSK5402 del filtrado para la remoción de la interferencia de 60 Hz se ha tomado como base el sistema de cancelación propuesto en la figura 23 para diseñar la estructura del filtro notch adaptativo de tercer orden ilustrado en la figura 22.(ver figura 85)

En el archivo *notch.c* se encuentra contenido en líneas de lenguaje C la implementación del algoritmo de filtrado adaptativo para la remoción de la interferencia de 60 Hz. Este algoritmo toma una ventana de 2000 muestras de señal muestreada a 500 Hz. Esto permitirá a la plataforma DSK5402 adquirir ventanas de 4 segundos de señal de señal electrocardiográfica y realizar el proceso de adaptación del coeficiente y filtrado cada cuatro ciclos cardiacos en un sujeto con frecuencia de 60 latidos por segundos. Como constante de adaptación μ se usó un valor muy pequeño ($\mu=0.001$) ,[23] el cual es adaptado con un número de iteraciones igual al número de muestras adquirida en cada ventana.

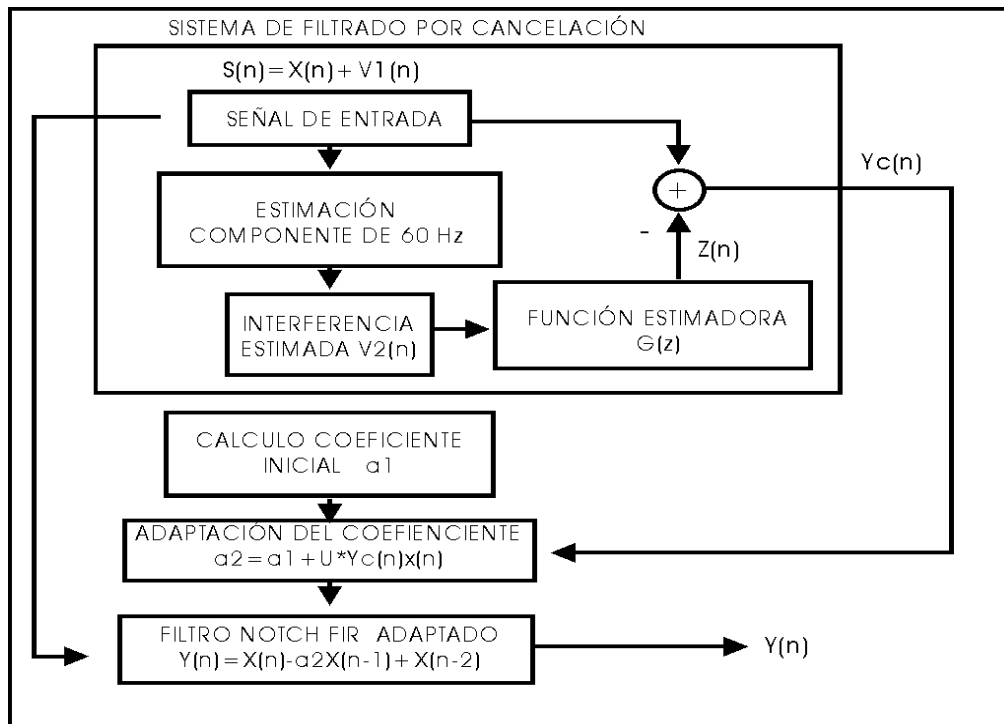


Figura 85. Algoritmo de filtrado para remoción de la interferencia de 60 Hz.

En la figura 86 se ilustra la respuesta en frecuencia con el coeficiente inicial a_1 ($c=-1.457937254$) del filtro notch. Introduciendo una señal electrocardiográfica proveniente de un simulador de parámetros contaminada con interferencia de 60 Hz con valor de 25% de la amplitud de la señal electrocardiográfica se obtiene el resultado en la señal de salida contemplado en las figuras 87 y 88. La función de transferencia con el coeficiente adaptado que toma un nuevo valor ($c=-1.456463$) se observa en la figura 89.

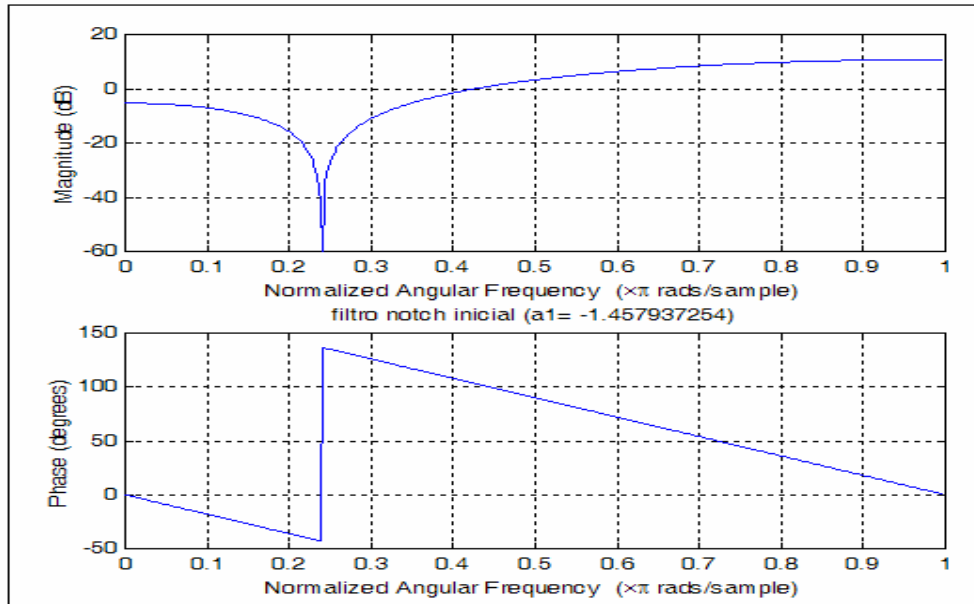


Figura 86. Respuesta en frecuencia del filtro notch con valor inicial del coeficiente a_1 .

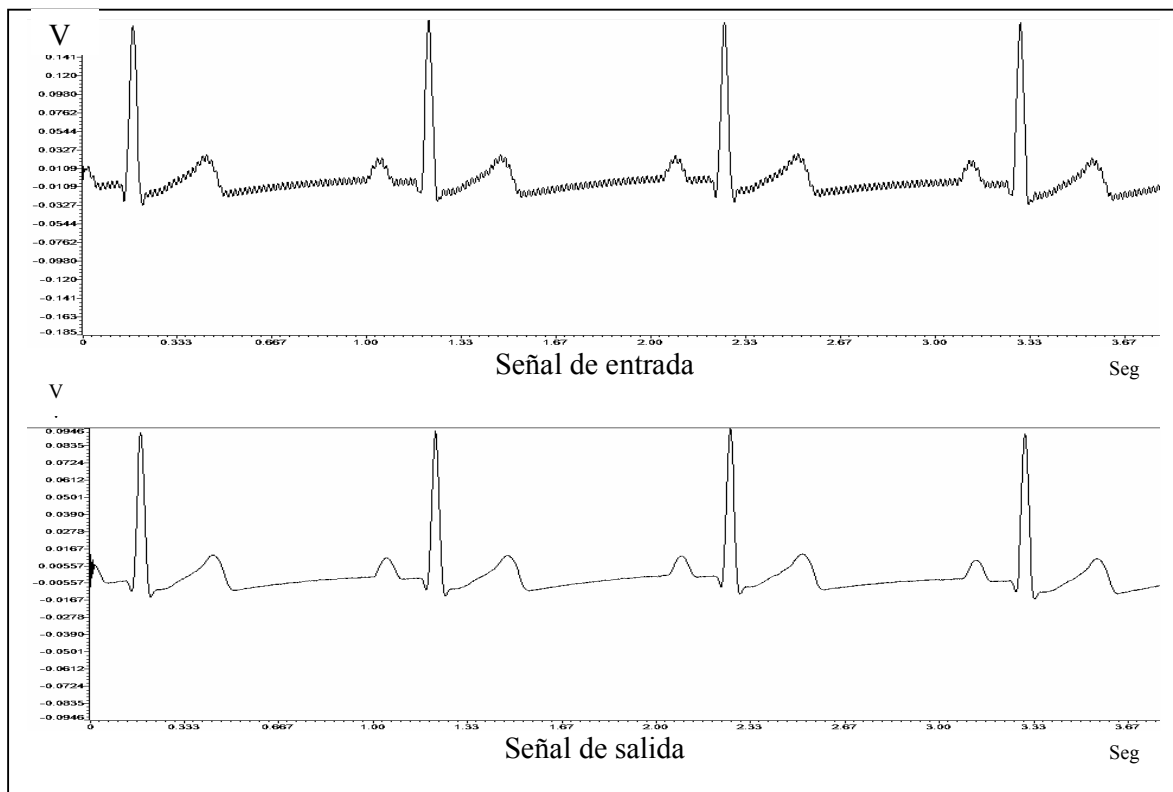


Figura 87. Resultado obtenido de filtrar señal ECG con interferencia con de 0.25 de la señal de entrada. (Dominio del tiempo)

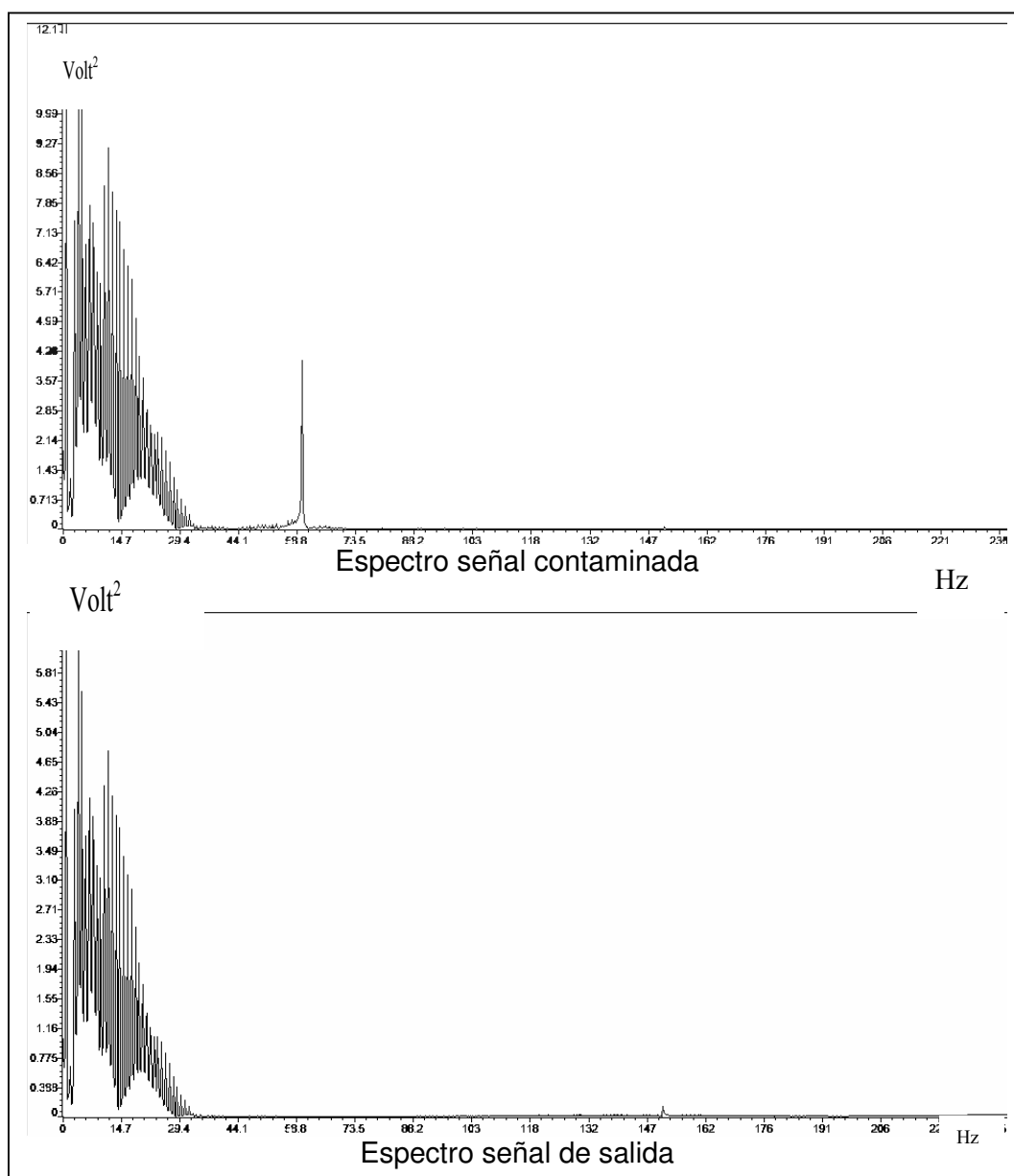


Figura 88. Resultado obtenido de filtrar señal ECG con interferencia con de 0.25 de la señal de entrada.(Dominio de la frecuencia)

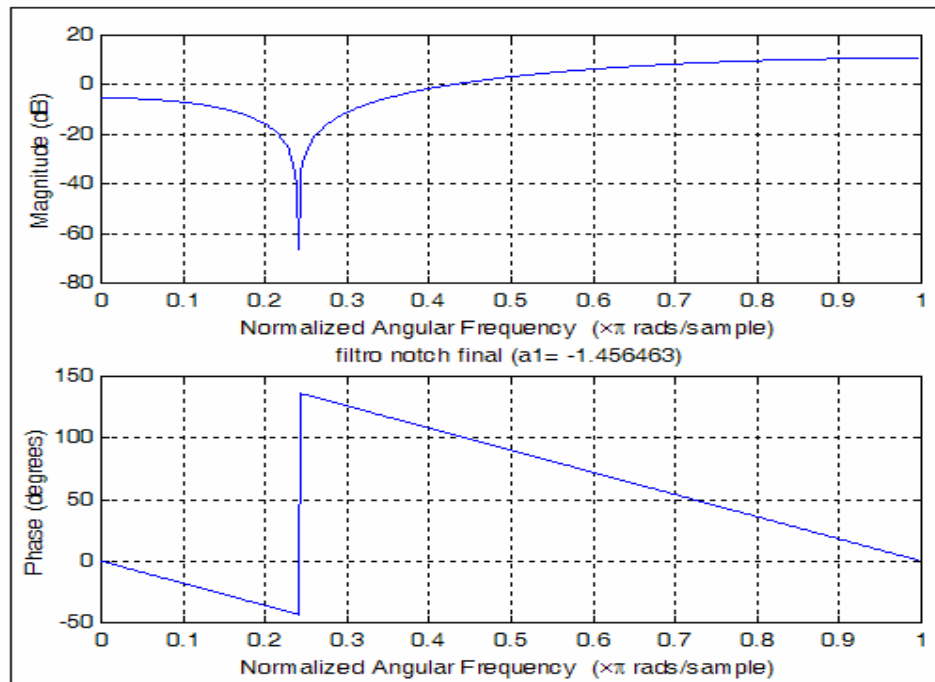


Figura 89. Respuesta en frecuencia del filtro notch con valor final del coeficiente a_1 .

4.4 DETECCIÓN DE LA ONDA R DEL COMPLEJO CARDIACO.

Por medio del lenguaje C se ha implementado en la plataforma DSK5402 un algoritmo para la detección de la onda R del complejo cardiaco según las recomendaciones detalladas en la figura 36.

El programa implementado se encuentra contenido en el archivo *detección qrs.c* el cual permite a la plataforma DSK5402 tomar una señal cardiaca con una frecuencia de muestreo de 500 Hz y realizar el cálculo de los tiempos RR con una precisión de 2 milisegundos.

Realizando pruebas con señales provenientes de un simulador de señales electrofisiológicas (ver figura 90) que fueron tomadas por medio de la plataforma DSK5402 se pudieron obtener los resultados contenidos en las gráficas 91, 92, 93, 94, 95 y 96, en las cuales se pueden observar la salida obtenida de cada etapa del proceso para la detección de la onda R y la cuantificación de los intervalos RR.

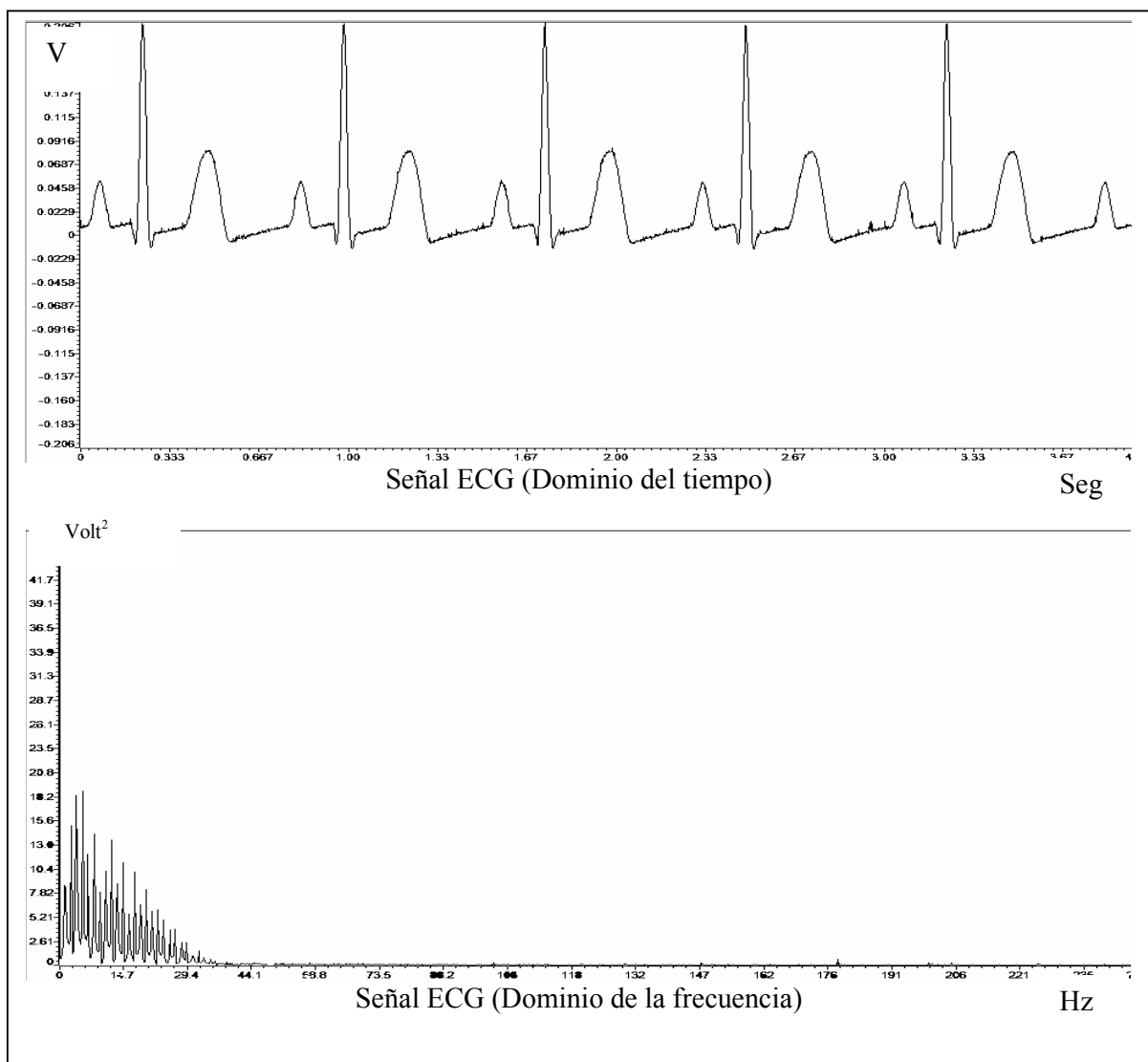


Figura 90. Señal ECG de entrada.(Detección de onda R)

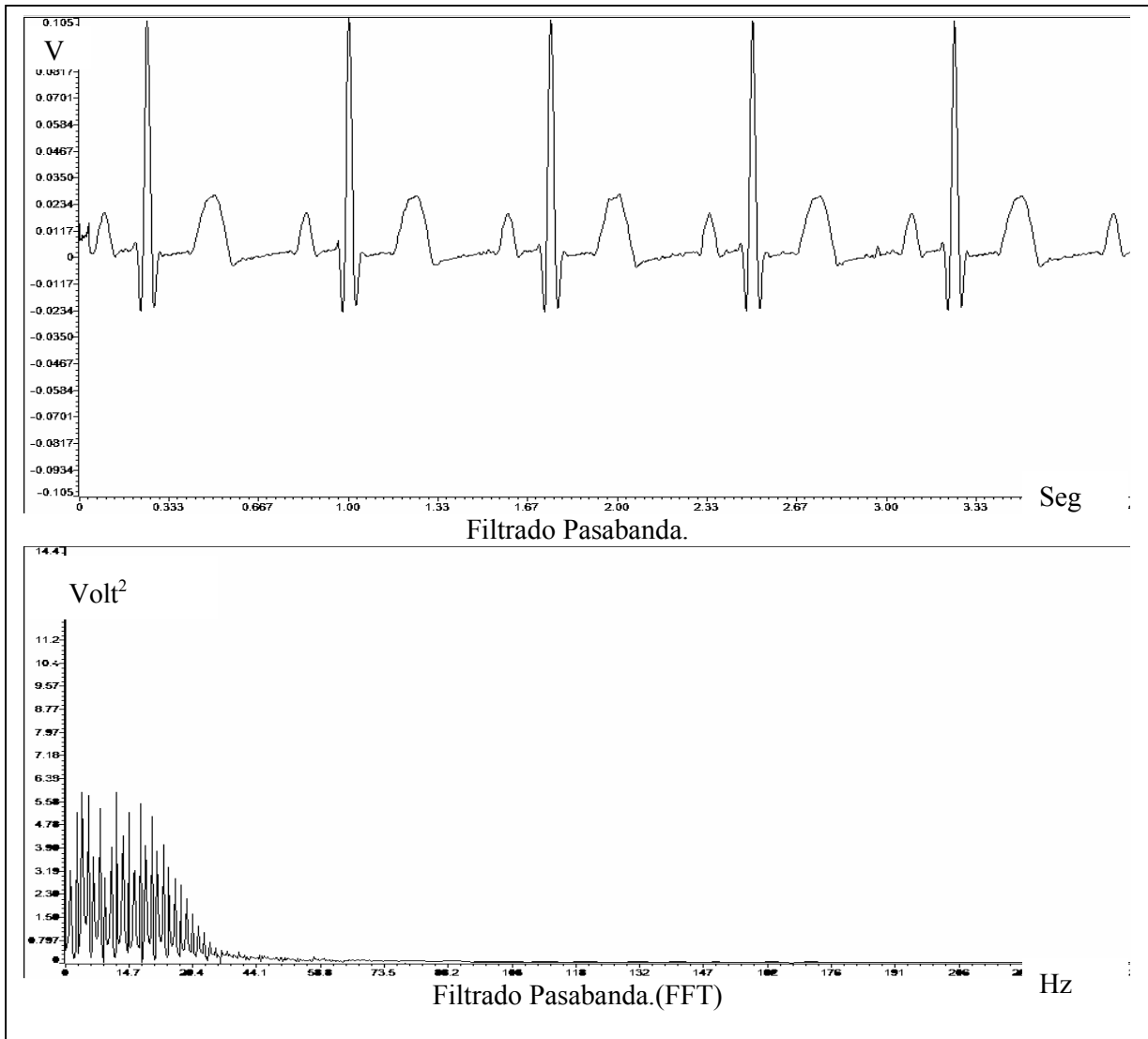


Figura 91. Proceso de filtrado pasabanda. (Detección de onda R)

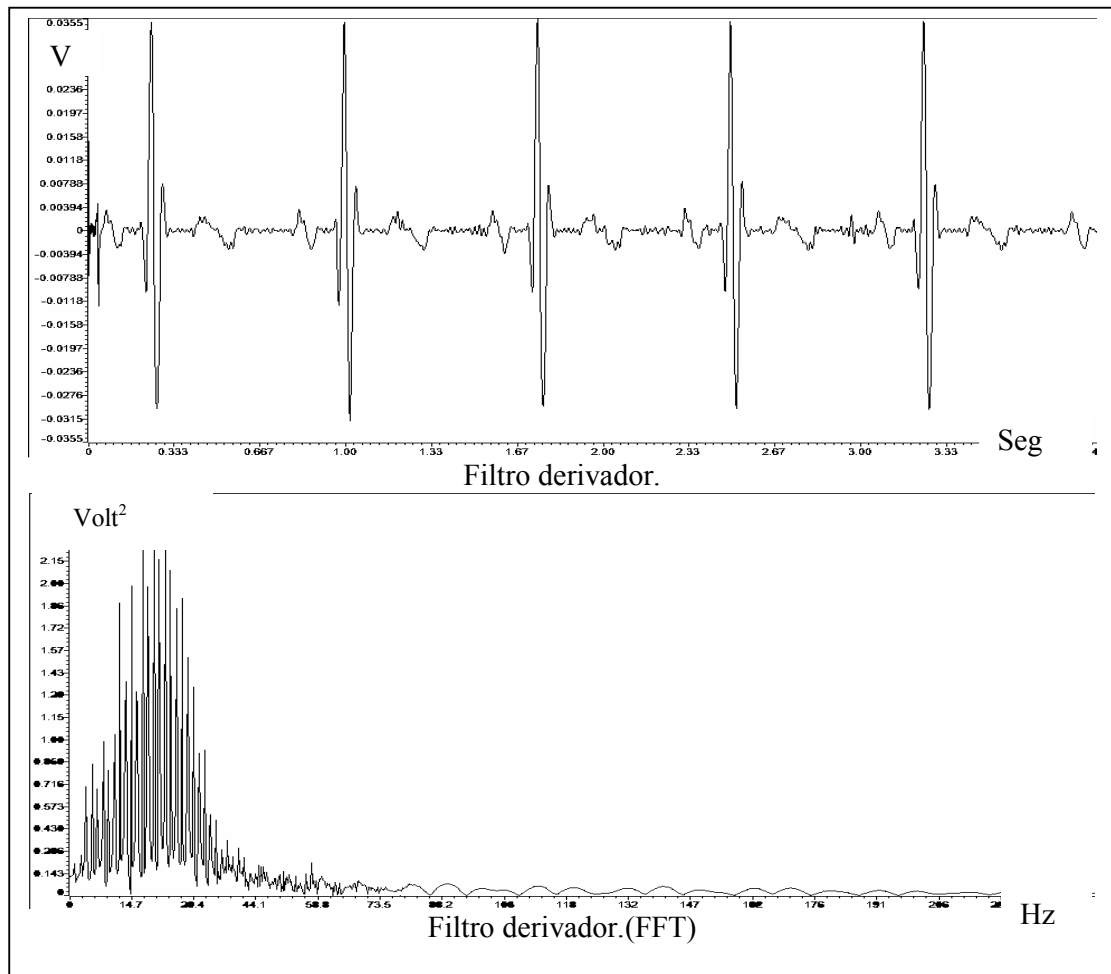


Figura 92. Filtrado con derivador. (Detección de onda R)

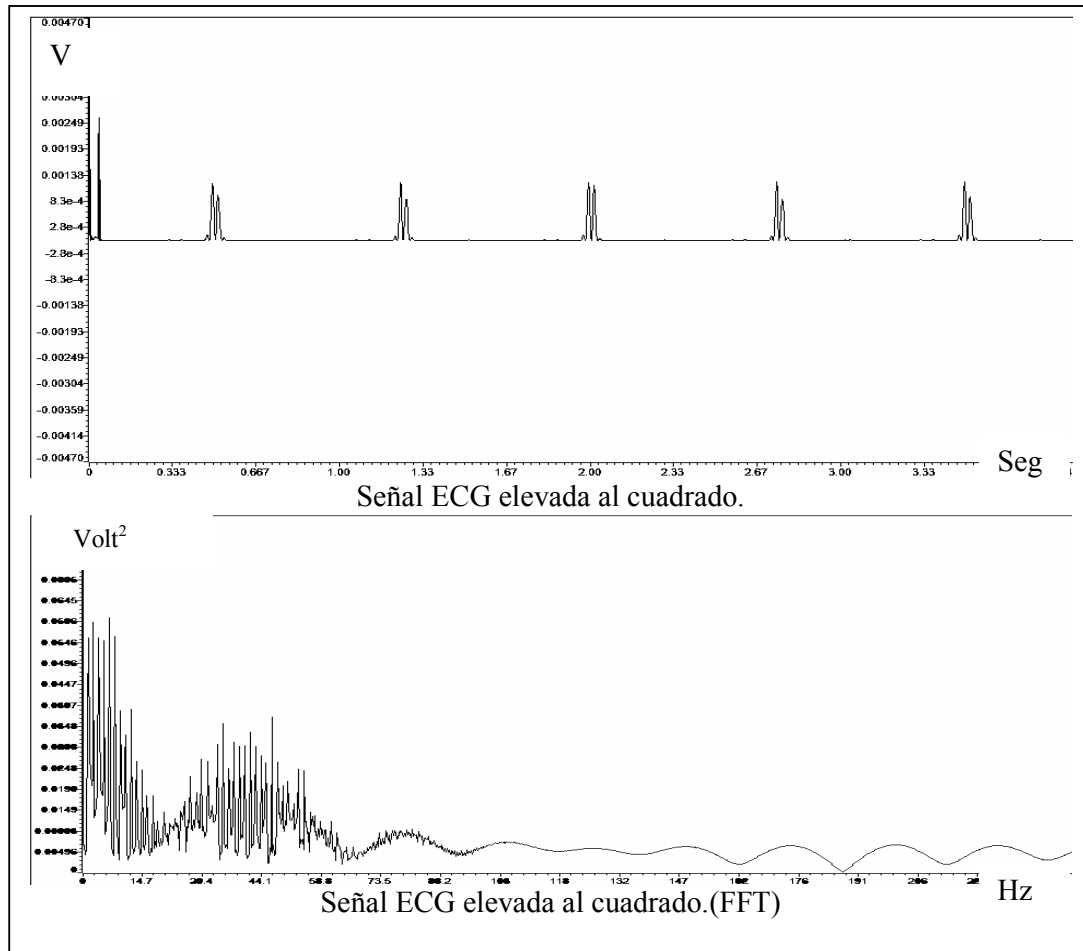


Figura 93. Elevación al cuadrado de la señal ECG. (Detección de onda R)

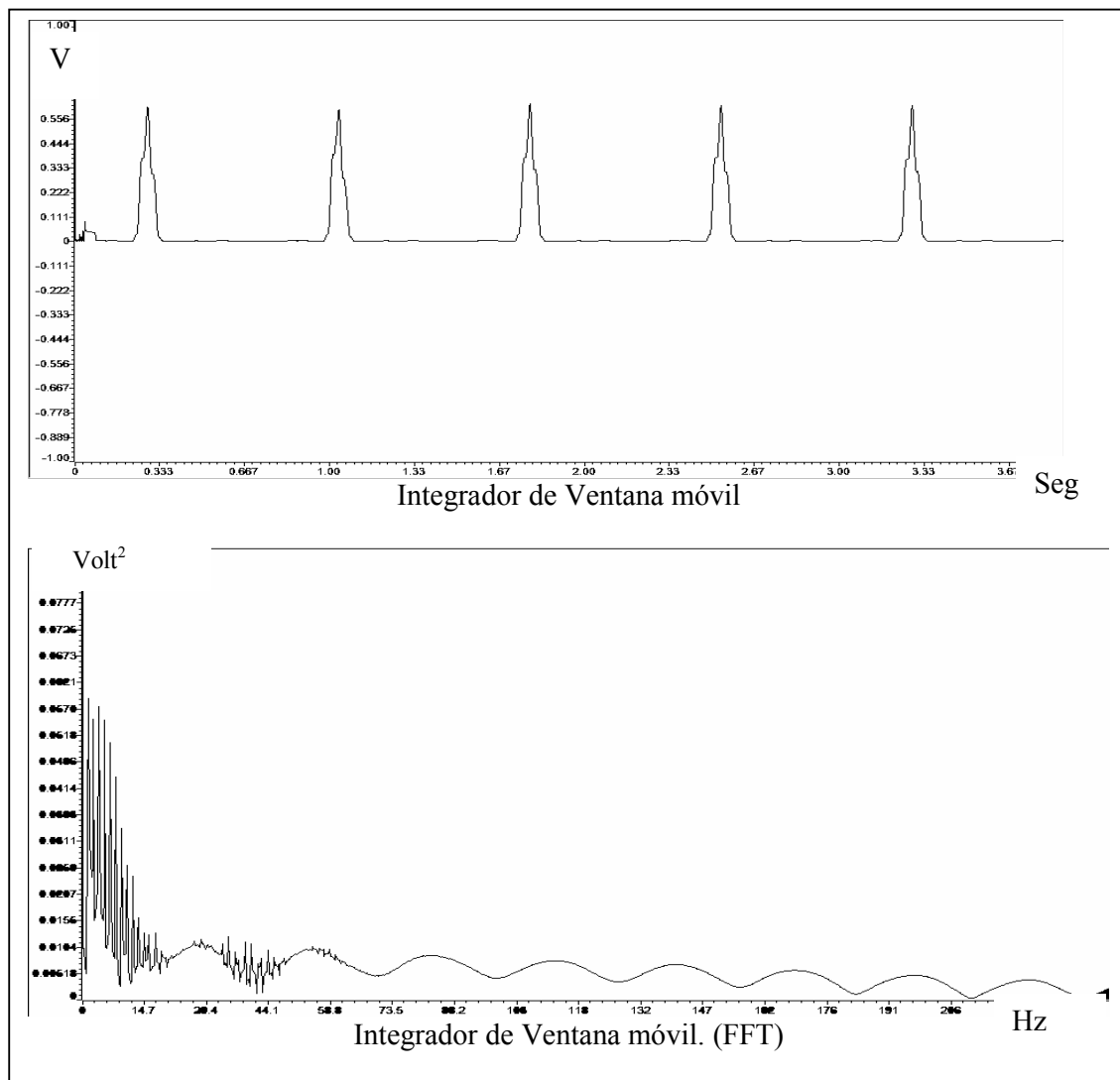


Figura 94. Integrador de Ventana móvil. (Detección de onda R)

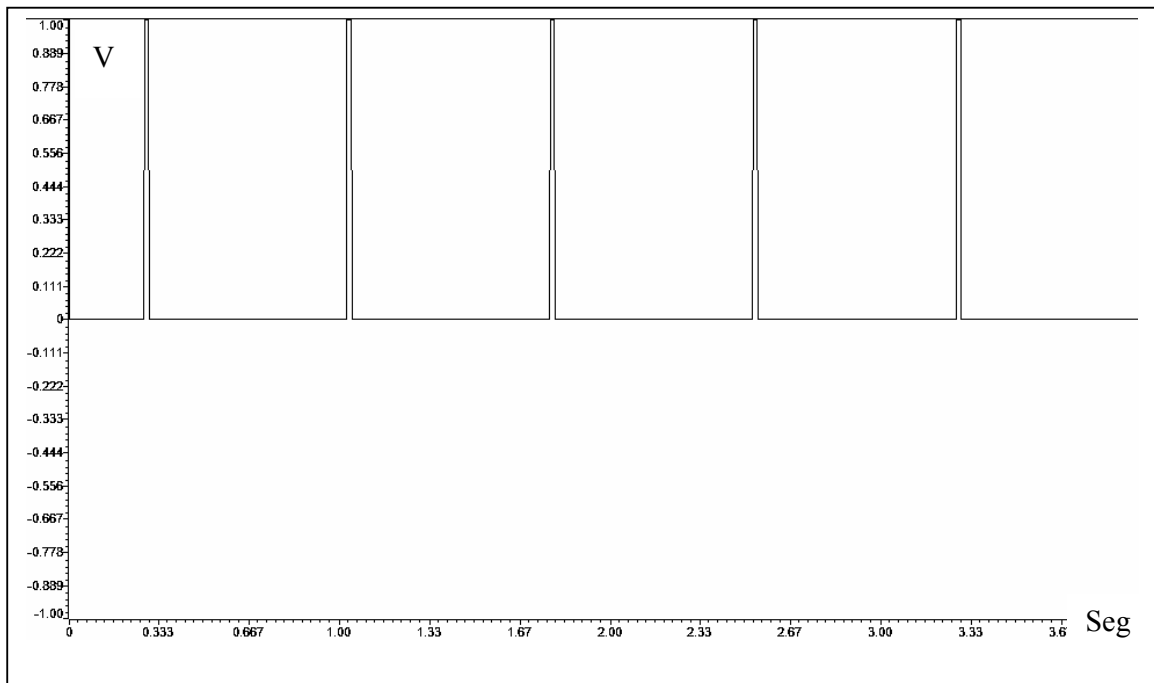


Figura 95. Aplicación del umbral. (Detección de onda R)

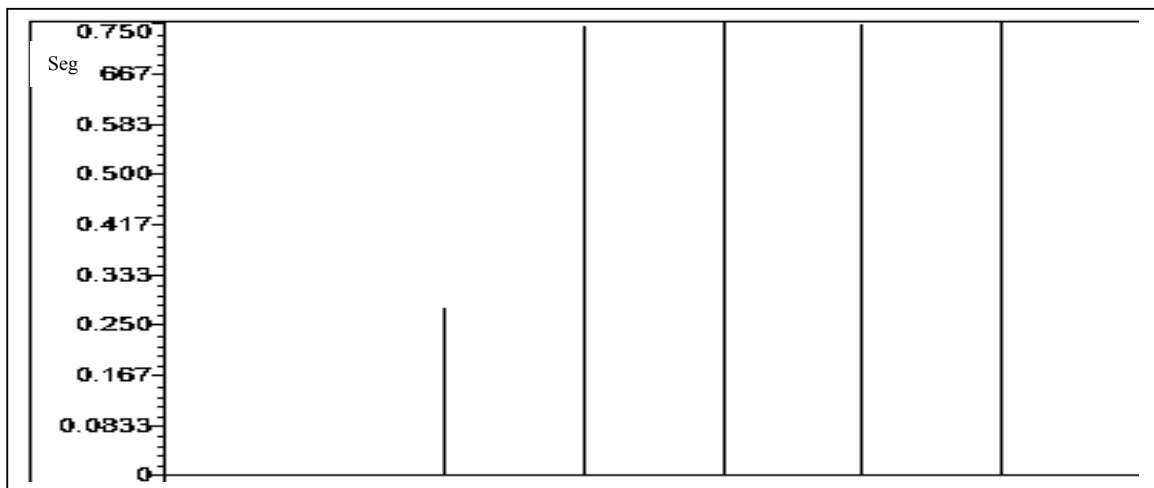


Figura 96. Tiempos RR. (Detección de onda R)

Como demostración de la necesidad e la remoción del corrimiento de la línea de base se ha elaborado una simulación en matlab donde se compara el resultado obtenido de calcular los intervalos RR a una señal ECG con línea de base normal y con un corrimiento en su línea de base. (ver programa *ondaRlineabase.mat*).

En la figura 97 se ilustra una señal ECG y los respectivos intervalos calculados y en la figura 98 contiene la misma señal ECG con corrimiento de su línea de base y los correspondientes intervalos calculados.

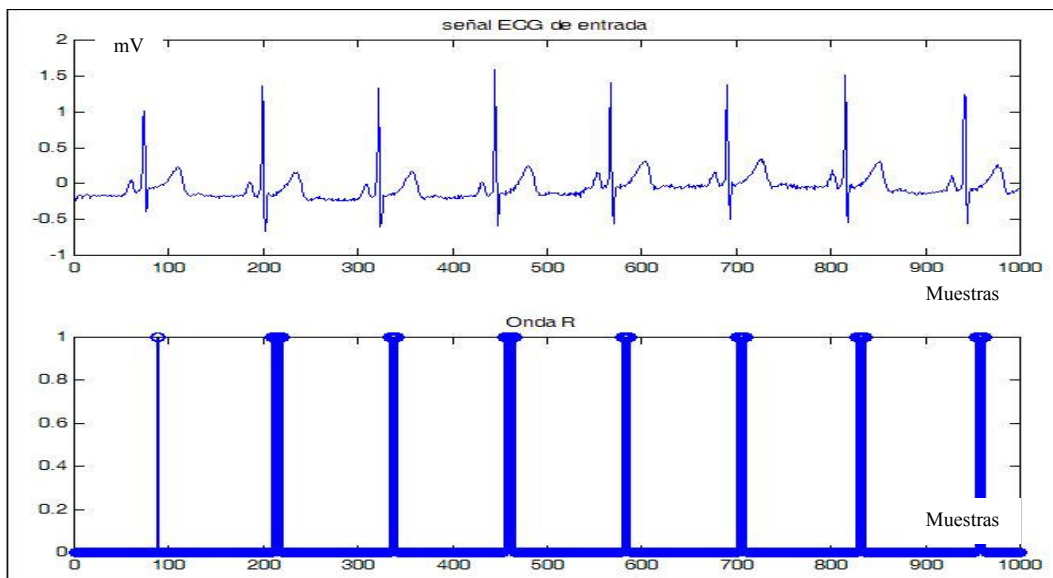


Figura 97. Señal ECG sin interferencia e intervalos RR Calculados.

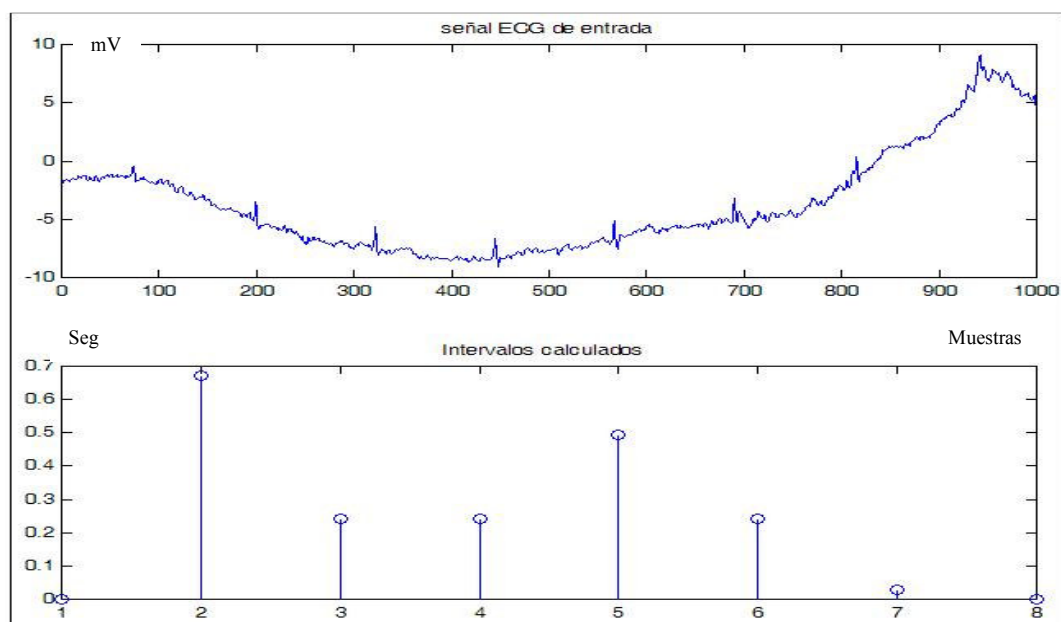


Figura 98. Señal ECG con interferencia e intervalos RR Calculados.

En la figura99 se ilustra una prueba realizada al algoritmo para la detección de la onda R con una señal cardiaca con corrimiento en su línea de base. En la parte superior de la figura 97 se puede apreciar que el trazo electrocardiográfico adquirido por la plataforma DSK5402 posee siete complejos y el la parte inferior se ilustra que solo fueron detectados seis intervalos. Esta prueba confirma la necesidad de poseer un sistema de remoción del corrimiento de la línea de base para evitar posibles errores en el cálculo de los intervalos RR.

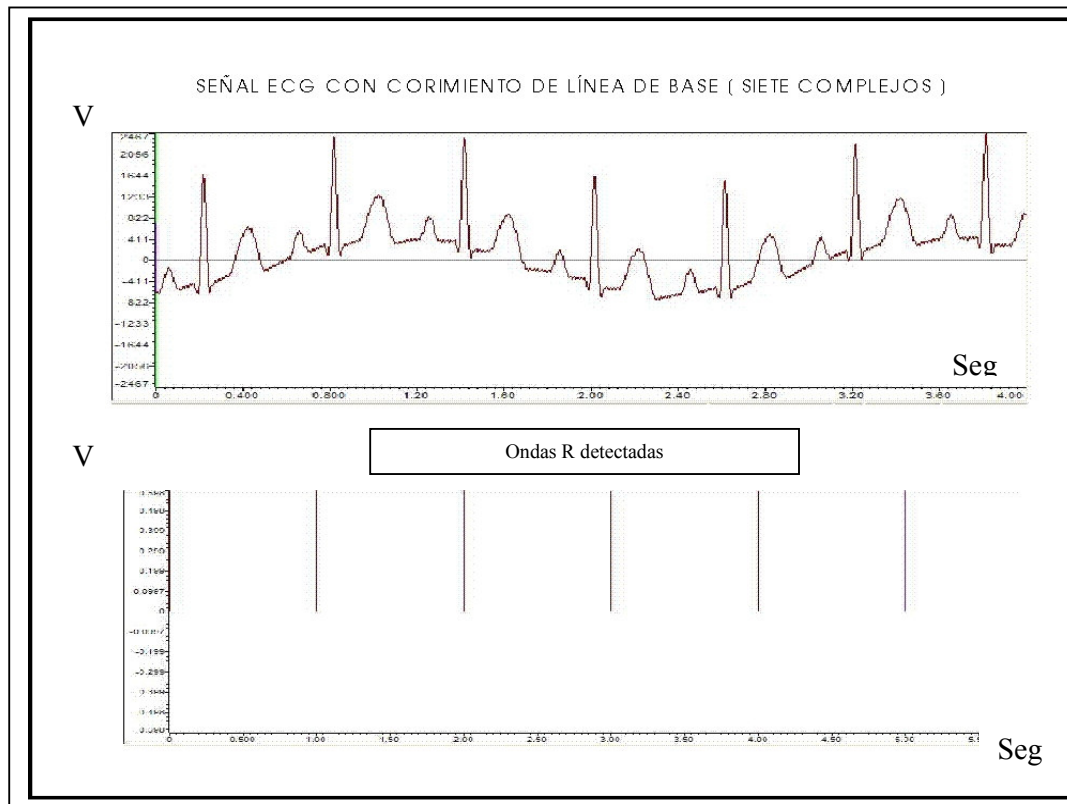


Figura 99. Detección de intervalos RR en señal ECG con corrimiento en su línea de base.

4.5 ANÁLISIS EN EL DOMINIO DEL TIEMPO DE LOS INTERVALOS RR.

En este trabajo de investigación se seleccionaron los índices más comunes utilizados en el análisis de los tiempos RR en el dominio del tiempo en periodos de corto tiempo resumidos en la tabla 1.

En el programa *dominiotiempo.mat* se ha realizado una simulación donde se calculan los índices mencionados sobre un arreglo de datos que contienen tiempos RR tomados de una base de datos según las formulas ilustradas en la tabla 1.

En la figura 100 se puede observar el arreglo de datos que contienen los tiempos RR utilizados en la simulación y el resultado obtenido de cada índice calculado.

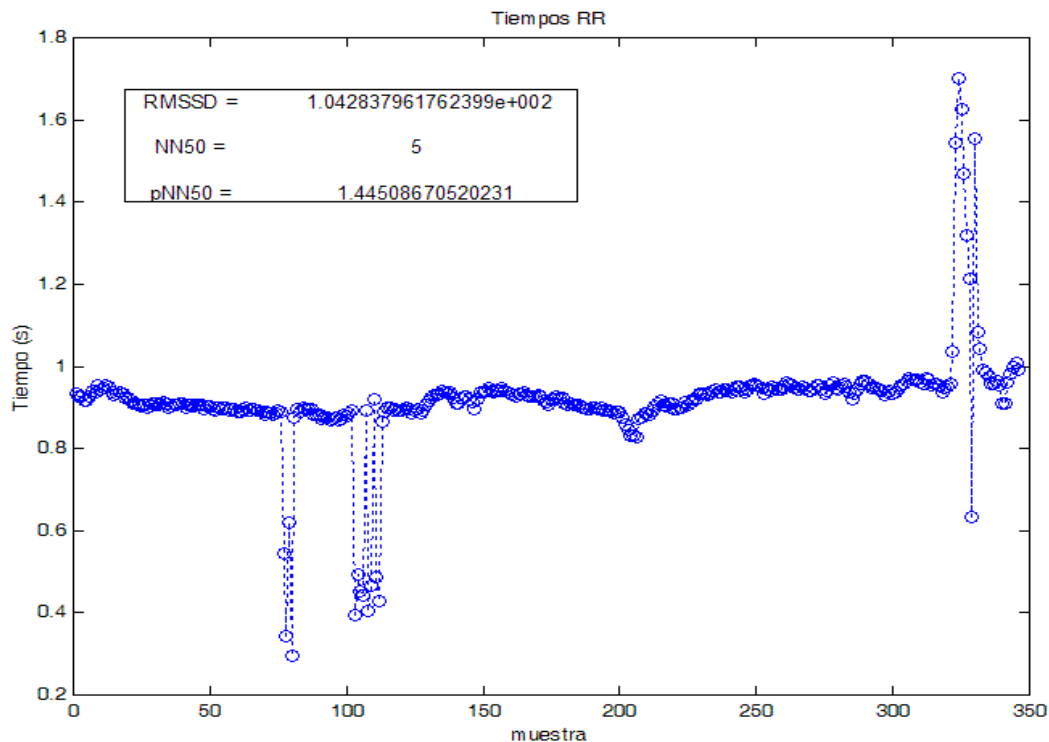


Figura 100. Arreglo de datos tiempos RR.

Por medio de las siguiente líneas de código en lenguaje C se ha implementado un algoritmo contenido en el archivo *índices dominio del tiempo.c* para el cálculo de los índices utilizados en el análisis en el dominio del tiempo de los intervalos RR para un arreglo de datos contenido en el archivo "tiempos.h" que contiene un número de 346 tiempos de intervalos RR.

En la figura 101 se observa el mismo resultado obtenido con la plataforma DSK5402.

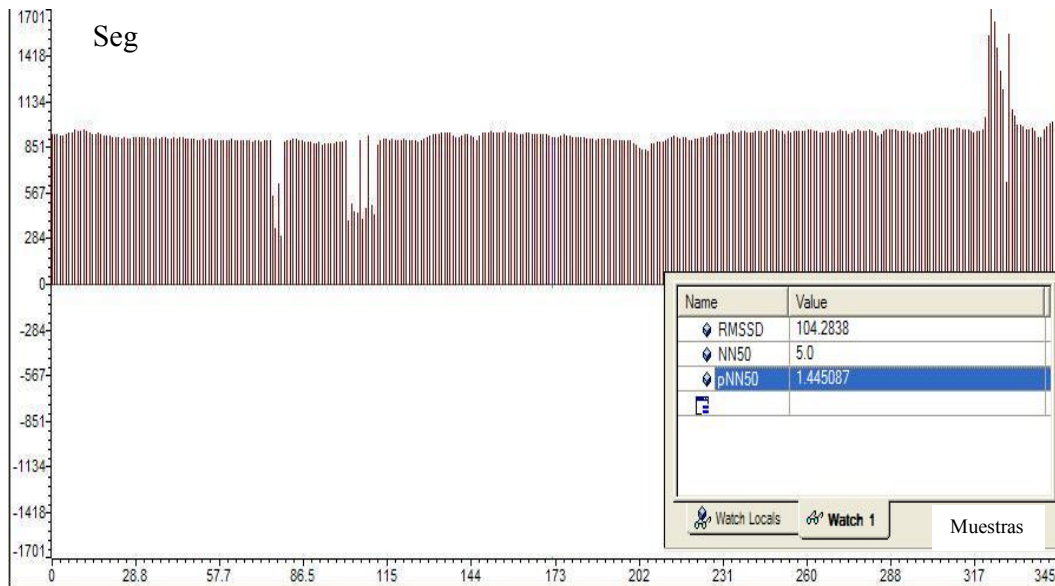


Figura 101. Resultado obtenido por la plataforma DSK5402 para análisis en el dominio del tiempo.

4.5 ANÁLISIS EN EL DOMINIO DE LA FRECUENCIA DE LOS INTERVALOS RR.

Para realizar el análisis en el dominio de la frecuencia de la serie de tiempo correspondiente a los intervalos RR extraídos de la señal cardíaca, se ha optado en este proyecto de investigación por usar el método del análisis espectral basado en la técnica de Lomb para datos desigualmente muestreados en el tiempo.

En el programa *PSDtiemposRR.mat* se encuentra contenido el código que realiza el análisis en el dominio de la frecuencia de un arreglo de datos conformado por tiempos RR (t_n). En el código se toma la serie de datos para crear otro arreglo que contiene los tiempos en que fue tomada cada muestra (t_n). Seguidamente se someten los datos x_n al proceso de enventanado por medio de la función de Hanning teniendo en cuenta que los datos no fueron tomados en un periodo regular de muestreo, por lo cual la función es evaluada en cada instante de tiempo t_n .

Por medio del algoritmo planteado en el programa *PSDtiemposRR.mat* se obtiene el análisis en el dominio de la frecuencia de los intervalos RR sin la necesidad de realizar el proceso de interpolación y remuestreo que convencionalmente se recomienda hacer para este tipo de análisis basado en

la FFT. También se realiza una comparación entre el resultado obtenido con el método de Lomb y el método tradicional por medio de la FFT. En la figura 102 se encuentra visualizado el arreglo de datos que contiene los intervalos RR. En la figura 103 se encuentra el análisis en el dominio de la frecuencia realizado por medio del método de la FFT y en la figura 104 se puede apreciar el resultado obtenido a través de la técnica de Lomb.

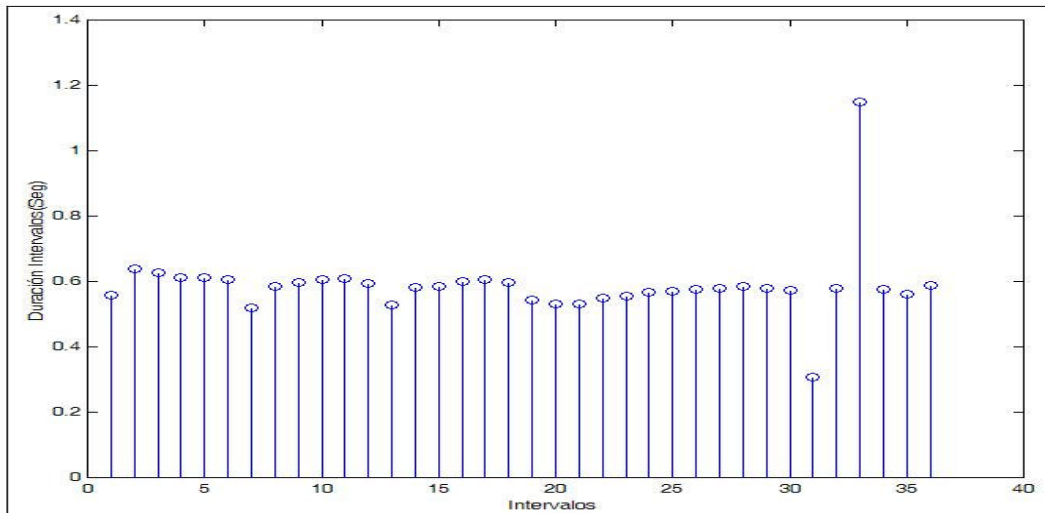
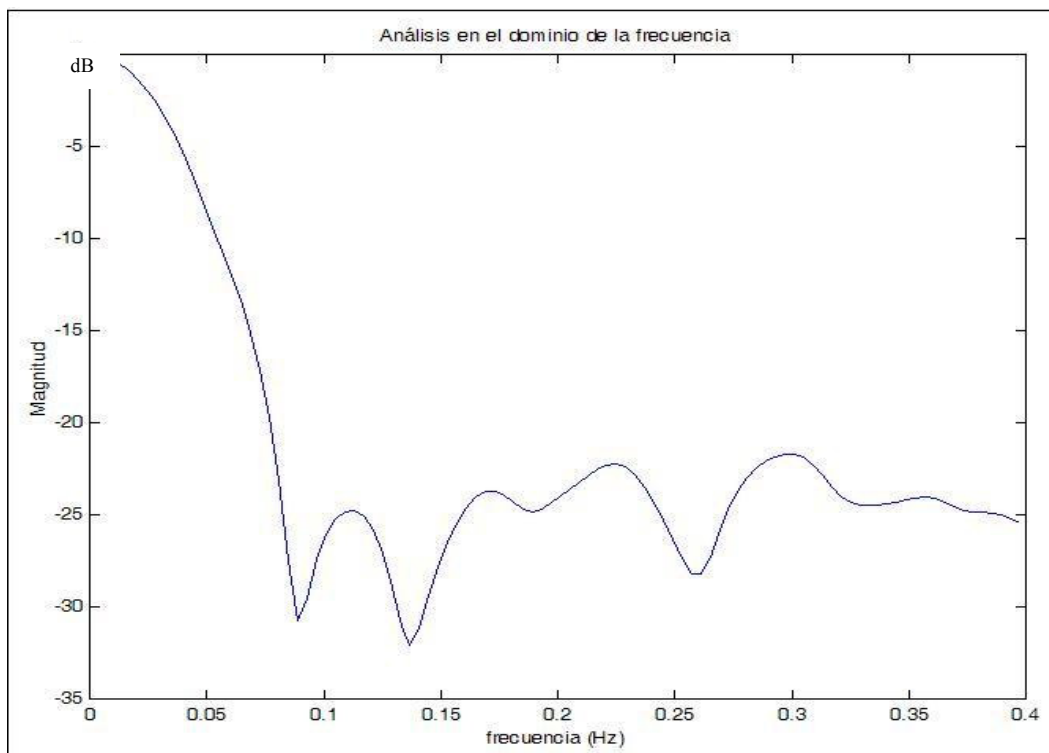
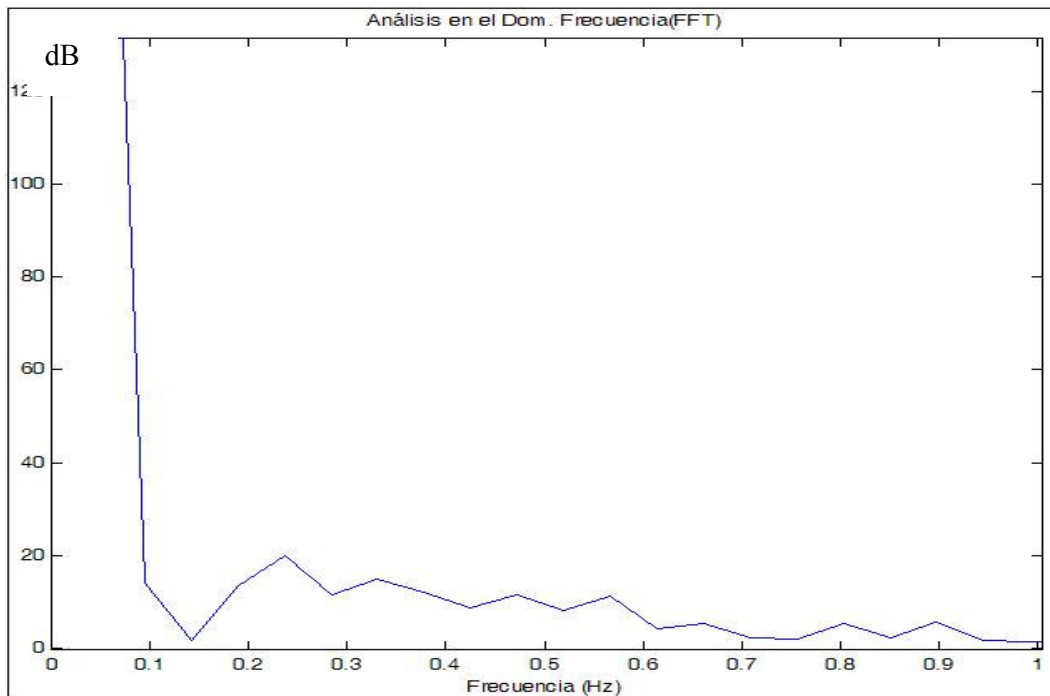


Figura 102. Tiempos RR.



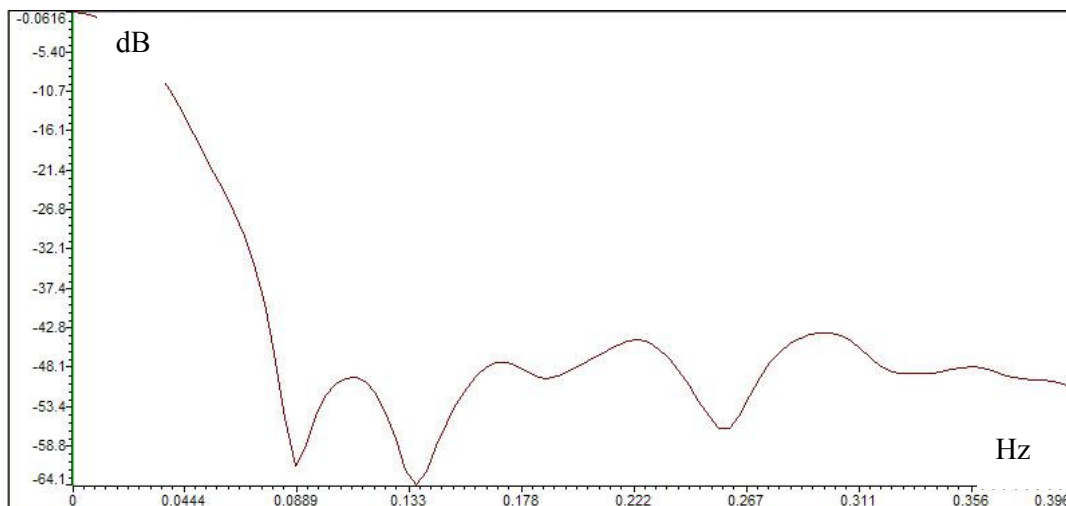
**Figura 103. Análisis en el dominio de la Frecuencia de los tiempos RR.
Método de Lomb.**



**Figura 104. Análisis en el dominio de la Frecuencia de los tiempos RR.
Método de la FFT.**

El código en lenguaje C implementado en el DSP para el análisis en el dominio de la frecuencia de los intervalos RR por medio del método de lomb está contenido en el archivo *lomb.c*.

En la figura 105 se ilustra el resultado obtenido por medio del DSP.



**Figura 105. Resultado del análisis espectral de
tiempos RR por medio del DSP.**

4.6 CÁLCULO DE ÍNDICES DEL DOMINIO DE LA FRECUENCIA.

El valor de los índices se obtiene por medio del cálculo del área bajo la curva obtenida en el análisis. En la sección 3.1.2 fueron mencionados los principales índices calculados en el análisis en el dominio de la frecuencia de la VFC. En la tabla 4 se puede apreciar estas medidas y su respectivo rango de frecuencias. en el dominio de la frecuencia tomando como límites de para cada región los respectivos rangos de frecuencias ilustrados en la tabla 2. Cómo método para calcular el área bajo la curva se utilizó la técnica de integración numérica basado en la regla de Simpson 1/3.[Chapra-Canale.]

El código contenido en el programa *dominofrecuencia.mat* permite ver los algoritmos utilizados en el cálculo del área de cada región comprendida entre los respectivos rangos establecidos para las diferentes medidas tomadas en el análisis en el dominio de la frecuencia de los tiempos rr. Este programa toma el arreglo de datos utilizado en *PSDtiemposRR.mat* y calcula los índices del análisis en el dominio de la frecuencia.

En la figura 106 ilustra el análisis en el dominio de la frecuencia de la figura 105 y el resultado de los respectivos índices.

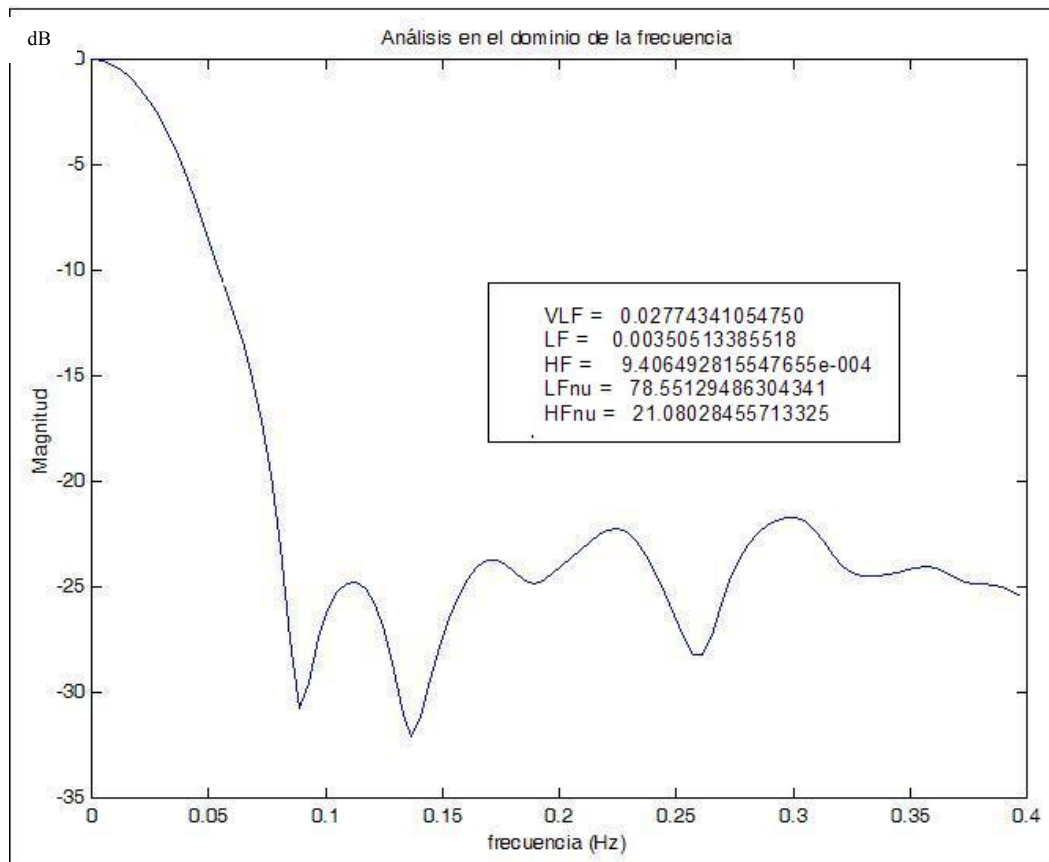


Figura 106. Índices del análisis en el dominio de la Frecuencia de los tiempos RR.

El código en lenguaje C implementado para el cálculo de los índices del análisis en el dominio de la frecuencia de los tiempos RR se encuentra almacenado en el programa *indices dominio frecuencia.c*.

En la figura 107 se puede apreciar el resultado obtenido por la plataforma de desarrollo utilizando el mismo arreglo de datos del programa *dominotiempo.mat*

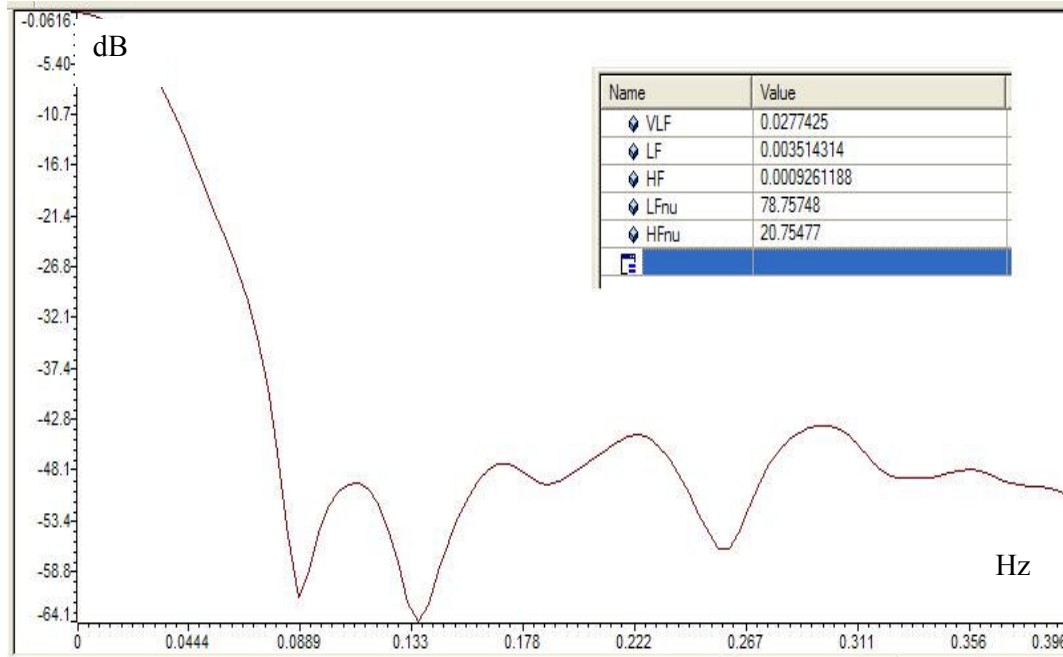


Figura 107. Resultado del los índices calculados del análisis espectral de tiempos RR por medio del DSP.

5. RESULTADOS FINALES

Culminada la etapa de diseño y simulación de algoritmos en este trabajo de investigación se procedió a implementar en la plataforma DSK5402 un algoritmo en código C que contiene cada uno de los procesos aplicados a la señal electrocardiográfica y que son necesarios para la realización del análisis de la variabilidad de la frecuencia cardiaca. En este capítulo se encuentran ilustradas las diferentes pruebas realizadas a cada etapa del algoritmo final.

En la figura 108 se puede detallar los diferentes procesos que realiza la plataforma DSK5402 para el tratamiento de la señal electrocardiográfica. El algoritmo ilustrado en la figura 108 indica que en cada iteración se analiza una ventana de señal electrocardiográfica, la cual tiene una duración de 4 segundos determinado por el algoritmo de adquisición y digitalización planteado en la sección 4.1. En cada iteración la señal electrocardiográfica es sometida un bloque de filtrado y de detección y cálculos de los intervalos RR.

Al terminar el número de iteraciones el sistema realiza los análisis en el dominio del tiempo y de la frecuencia a los intervalos RR recolectados.

5.1 PRUEBAS AL SISTEMA DE REMOCIÓN DEL DESPLAZAMIENTO DE LA LÍNEA DE BASE.

Para evaluar el desempeño del sistema de remoción del corrimiento de línea de base se utilizó un generador de señales convencional conectado a la salida del circuito amplificador de señal electrocardiográfica a través de un sumador implementado con un amplificador operacional. Este circuito se conecta directamente a la entrada del circuito conversor análogo a digital (CODEC) del DSK5402.

Por medio de la plataforma DSK5402 fueron adquiridas señales senosoidales de muy baja frecuencia (menores a 0.3 Hz) sumadas a señales electrocardiográficas para provocar desplazamientos en la línea de base.

En la figura 109 se puede apreciar dos señales electrocardiográficas con corrimiento en sus respectivas líneas de base provocadas por una señal senoidal con un valor de frecuencia igual a 0.2 Hz. Además la figura 109 contiene las diferentes funciones de línea de base estimada y el resultado obtenido al aplicar el filtrado por cancelación a cada una de la señales.

La figura 110 contiene el mismo proceso de filtrado aplicado a una interferencia provocada por la adición de una señal senoidal con un valor de frecuencia igual a 0.3 Hz.

En las figuras 109 y 110 se puede verificar la efectividad del sistema de filtrado adaptativo el cual estima matemáticamente la interferencia que es utilizada para la remoción del corrimiento de la línea de base. En los resultados obtenidos se puede observar que la interferencia es eliminada sin causar ninguna distorsión de la señal electrocardiográfica.

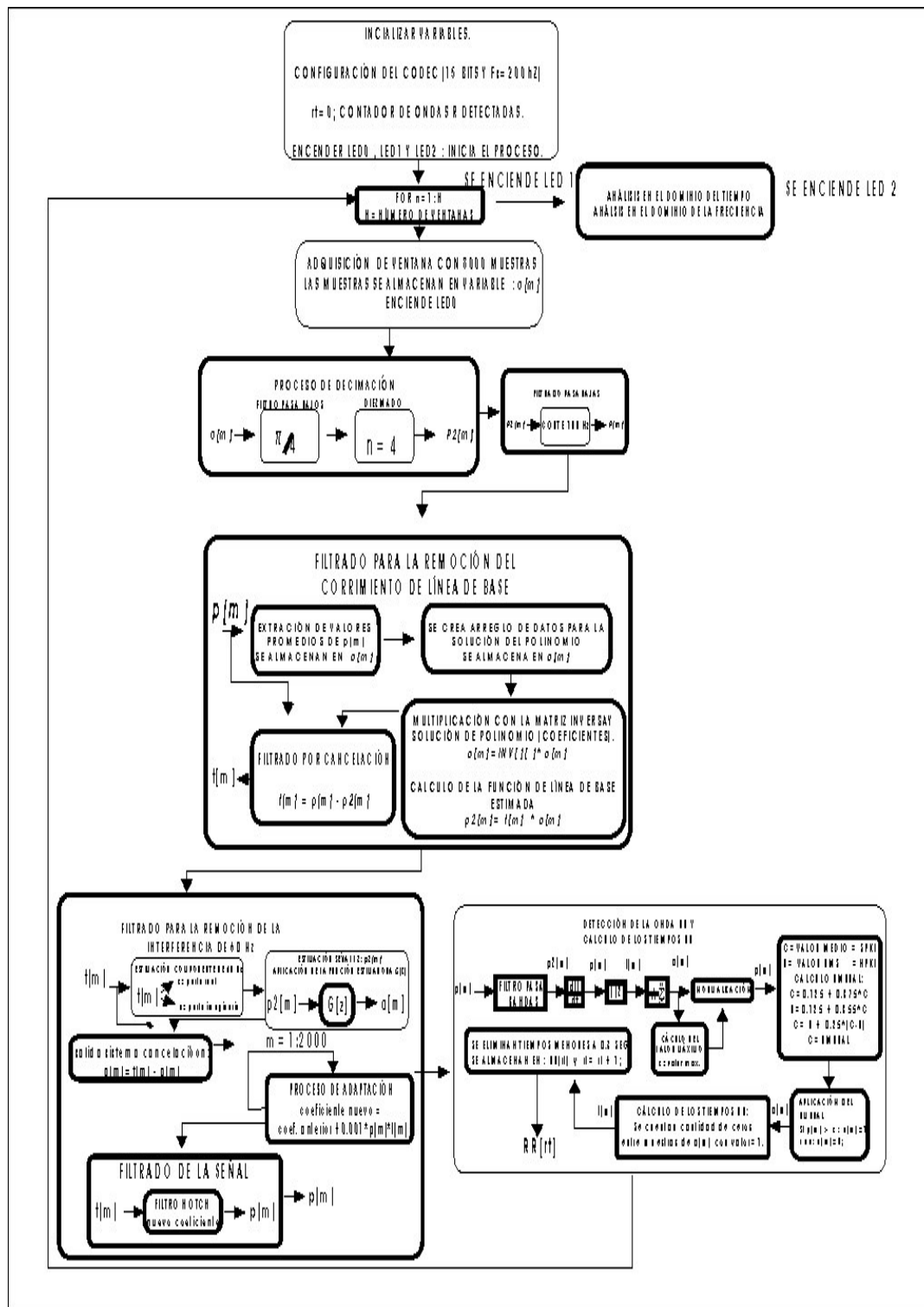


Figura 108. Algoritmo final.

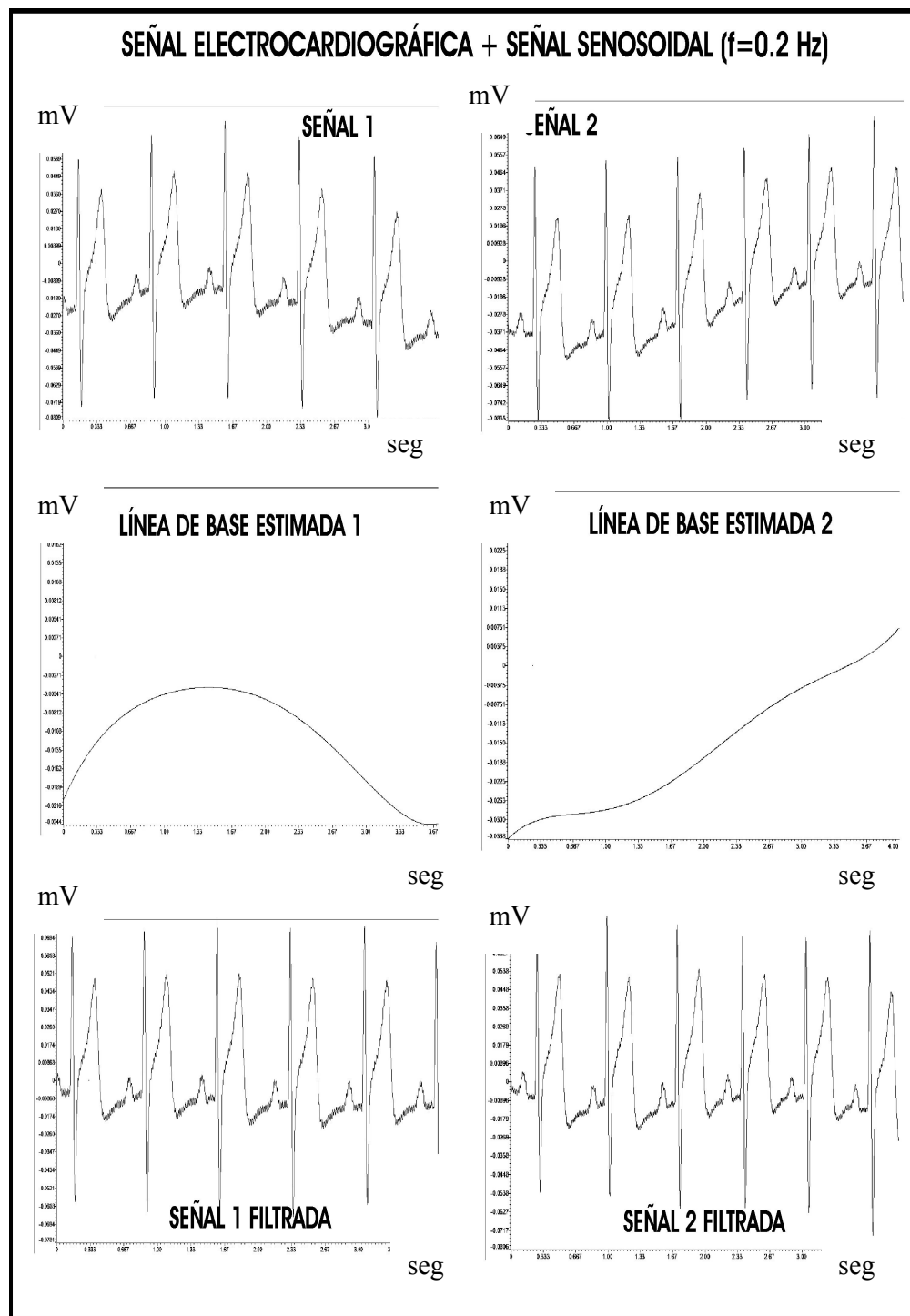


Figura 109. Proceso de remoción de interferencia provocada por una señal senosoidal ($F= 0.2$ Hz).

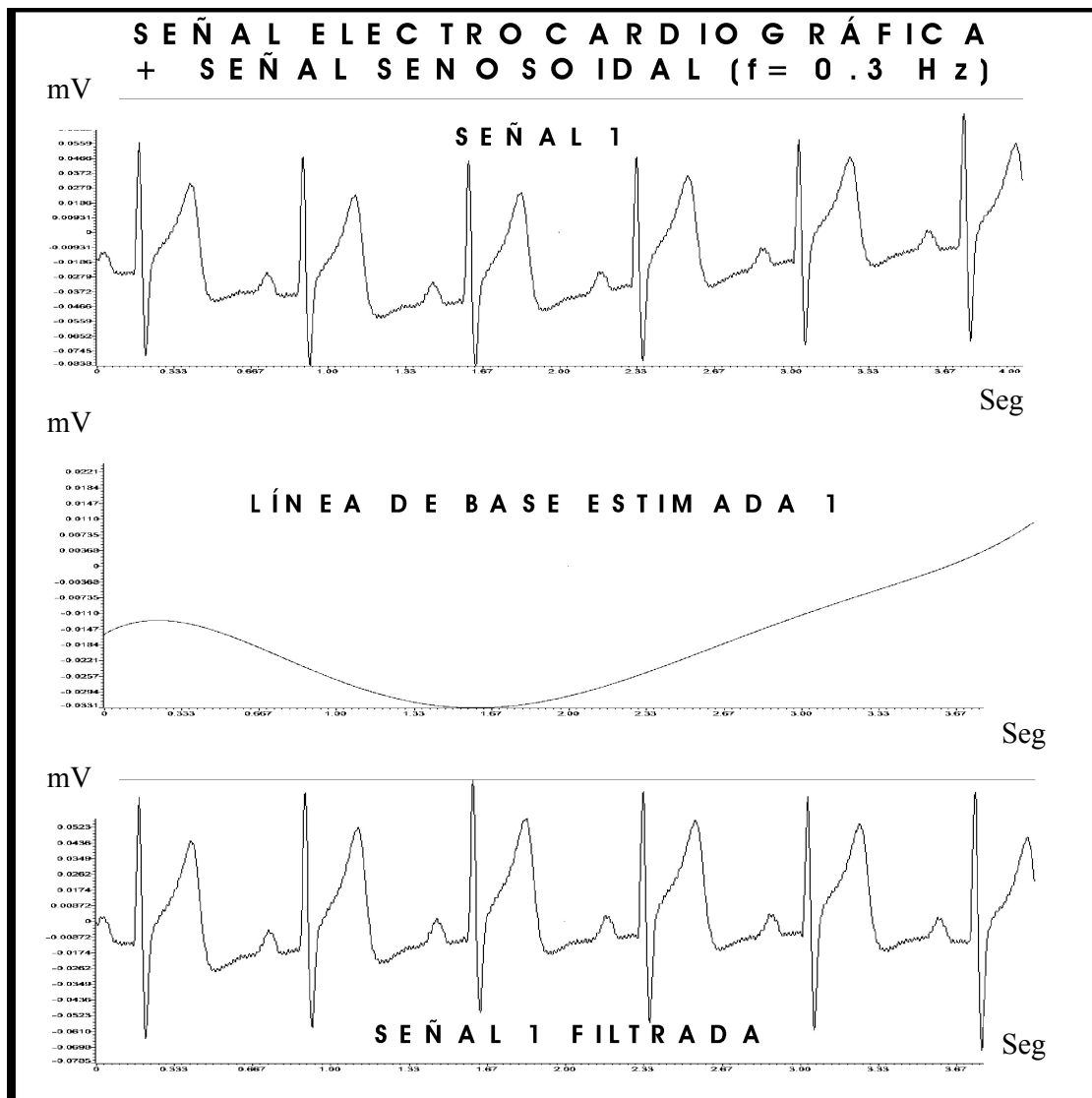


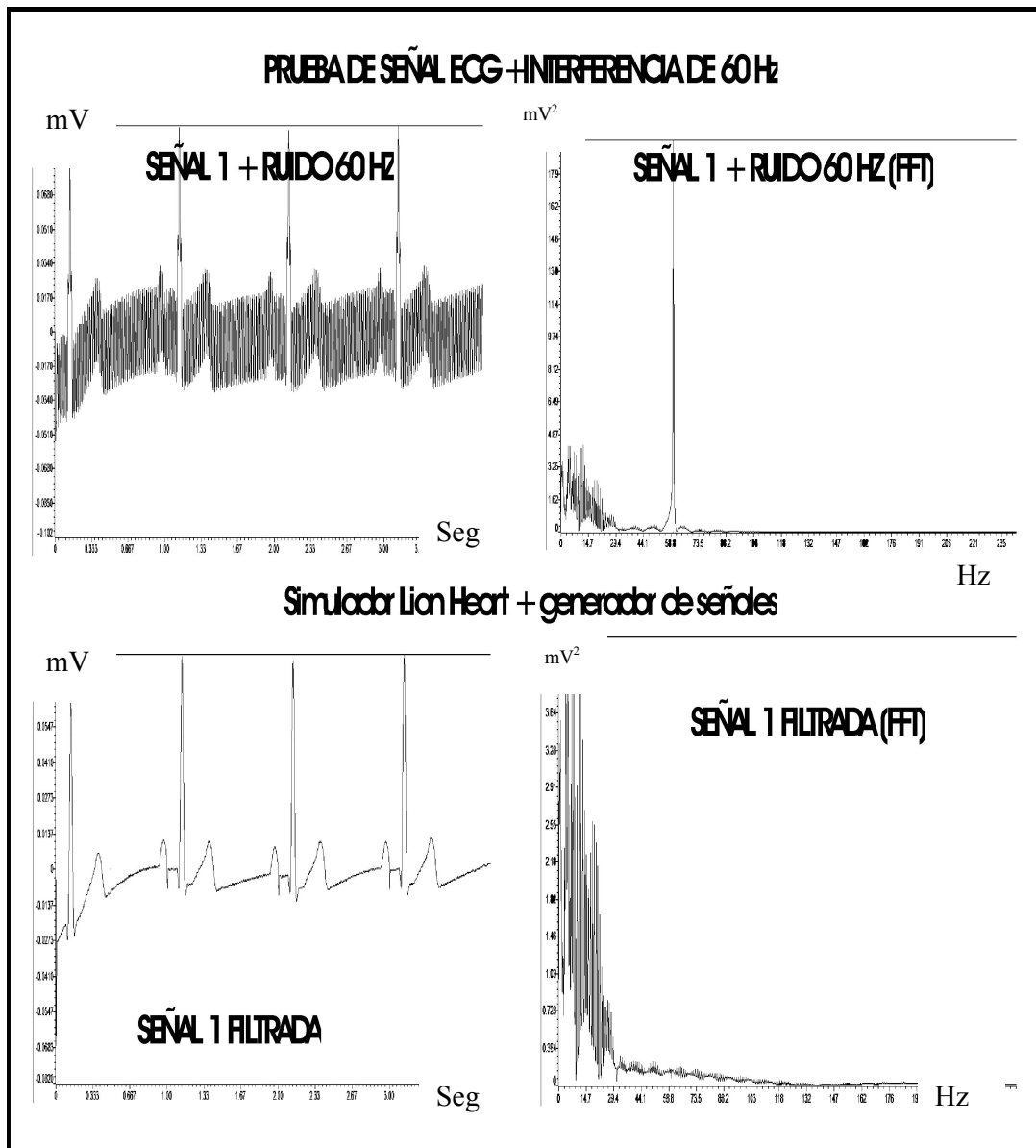
Figura 110. Proceso de remoción de interferencia provocada por una señal senoidal ($F=0.3$ Hz).

5.2 PRUEBAS AL SISTEMA DE REMOCIÓN DE INTERFERENCIA DE 60 HZ.

En la metodología para evaluar el sistema de remoción de interferencia de 60 Hz fueron utilizados dos tipos de equipos simuladores de señales electrofisiológicas: Lion Heart y Medsin 300B.

Con el equipo Lion Heart fueron realizadas pruebas con ruido aditivo provenientes de un circuito sumador y un generador de señales senoidales.

En la figura 110 se pudo apreciar el resultado obtenido de sumar a una señal electrocardiográfica proveniente del equipo Lion Heart con ruido aditivo de 60 Hz.



**Figura 111. Señal ECG + interferencia de 60 Hz.
Simulador Lion Heart.**

Con ayuda del simulador Lion Heart se realizaron pruebas variando la frecuencia de la señal de interferencia en un rango entre 59 y 61 Hz para evaluar el comportamiento del sistema de filtrado ante estas variaciones.

En la figuras 112 y 113 se pueden apreciar los resultados de las pruebas realizadas al sistema para la remoción de la interferencia de 60 Hz con variaciones en el valor de la frecuencia central. Los resultados obtenidos indican que el sistema puede remover la interferencia en un escenario donde el valor de la frecuencia no sea constante en el tiempo.

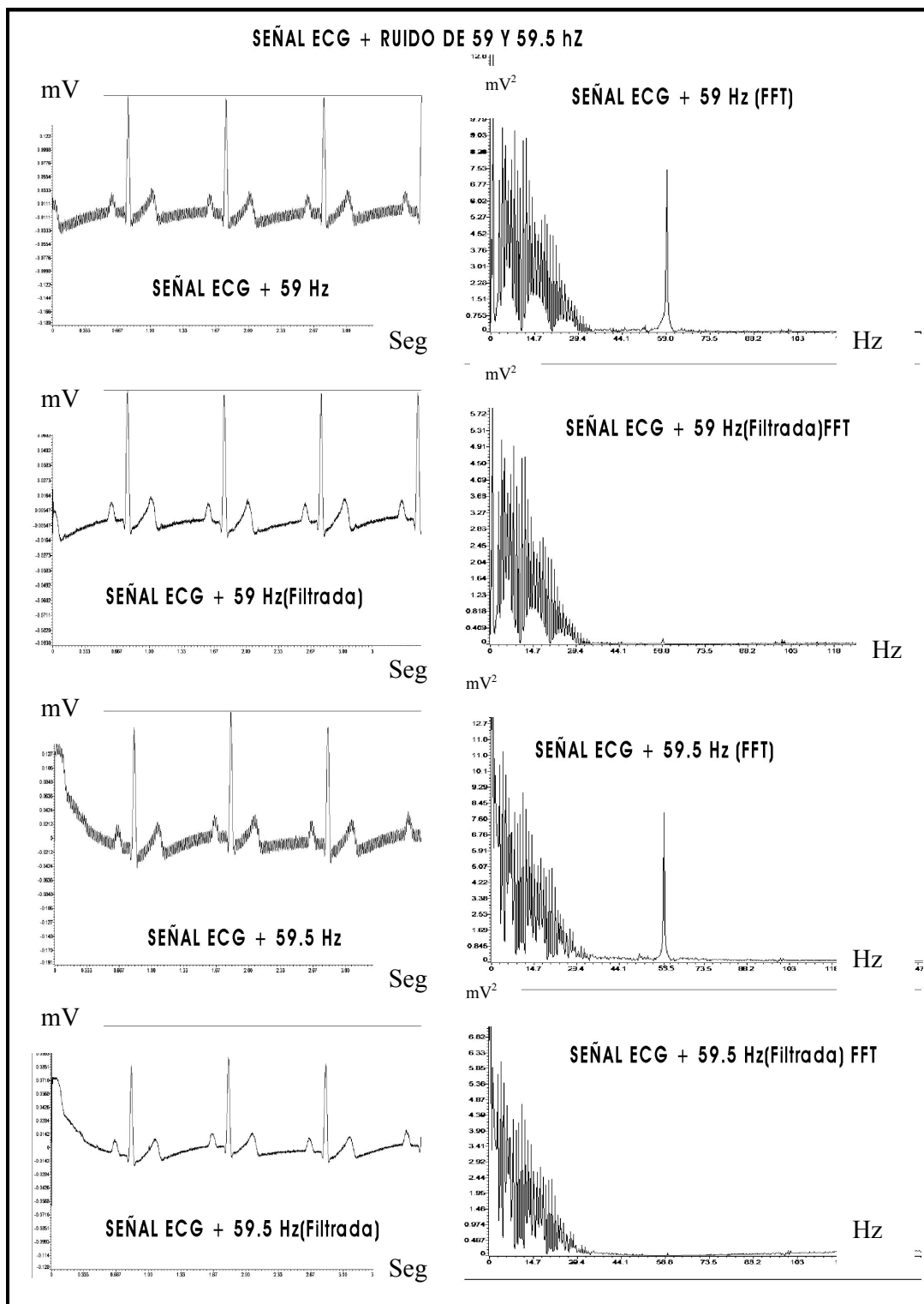


Figura 112. ECG + Interferencia de 59 y 59.5 Hz.

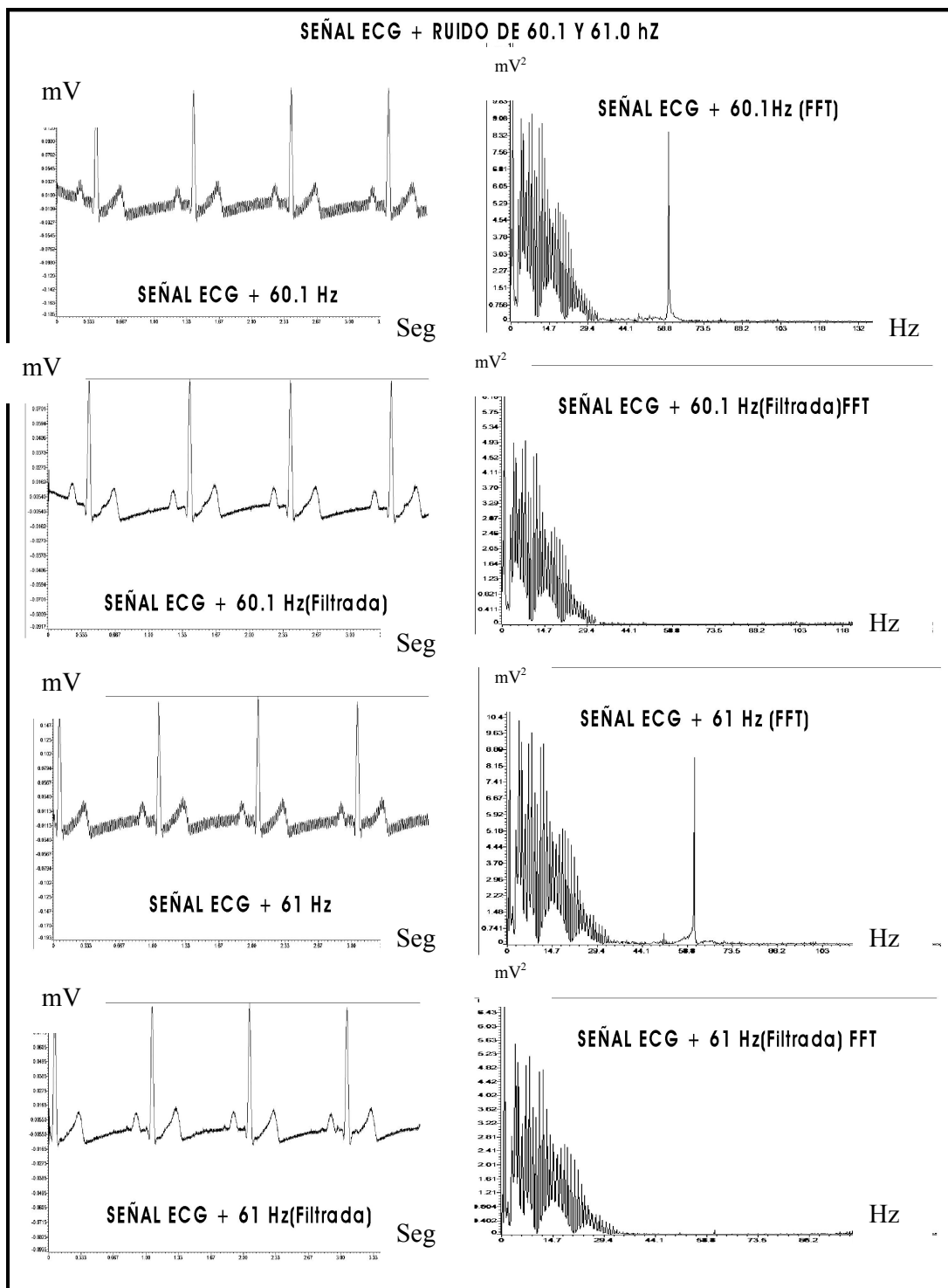


Figura 113. ECG + Interferencia de 60.1 y 61.0 Hz.

En las figuras 114,115 y 116 se pueden apreciar los resultados obtenidos de realizar pruebas con el simulador medsin300B con diferentes valores de amplitud de la interferencia de 60 Hz con relación de 0.25 , 0.5 y 1 con respecto a la amplitud de la señal electrocardiográfica. En las figuras se puede detallar como la respuesta en frecuencia del filtro notch dependiendo de la señal de entrada y de la magnitud de la interferencia que la afecta.

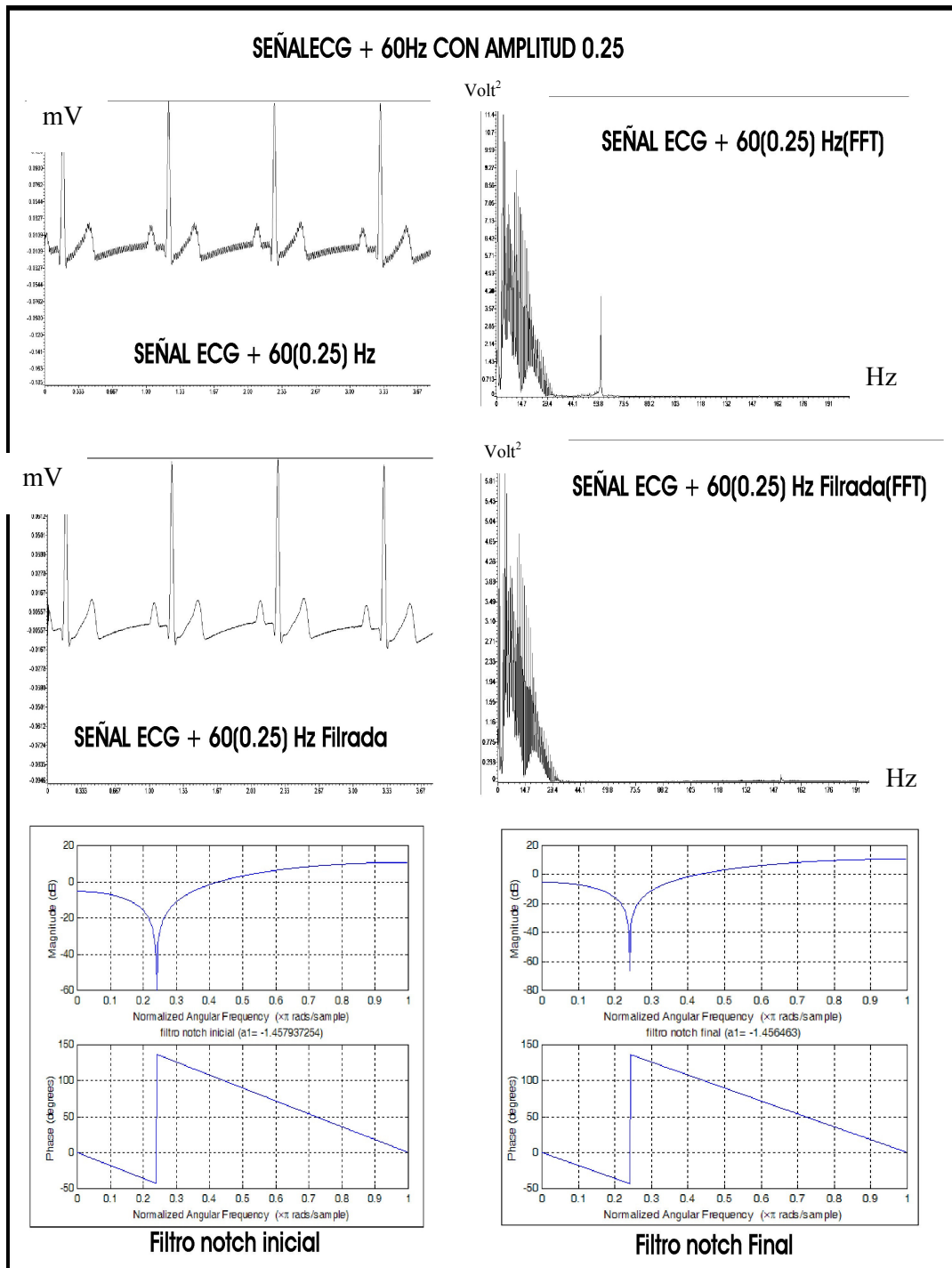


Figura 114. Señal ECG + 60 Hz con amplitud 0.25.

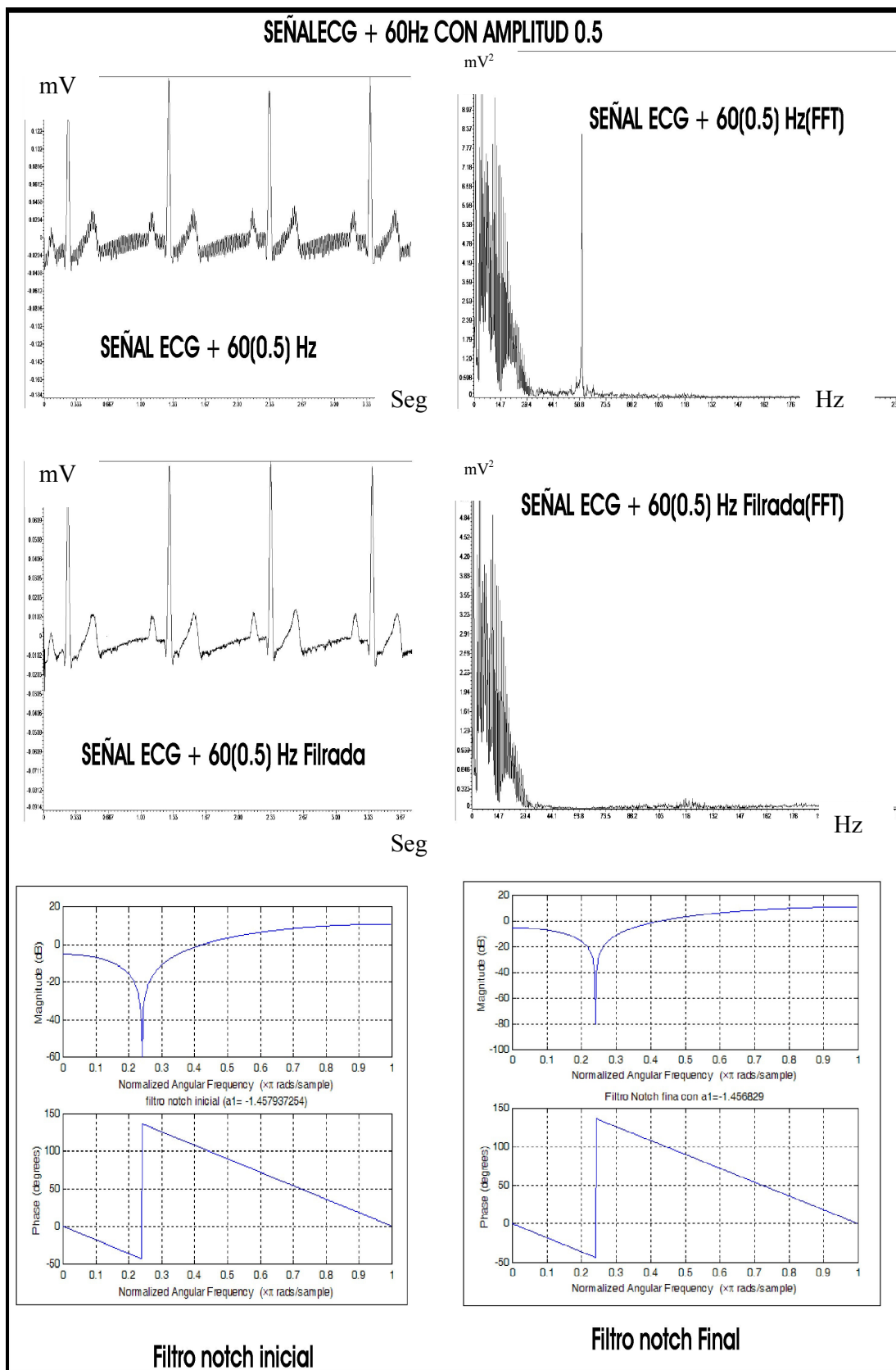


Figura 115. Señal ECG + 60 Hz con amplitud 0.5.

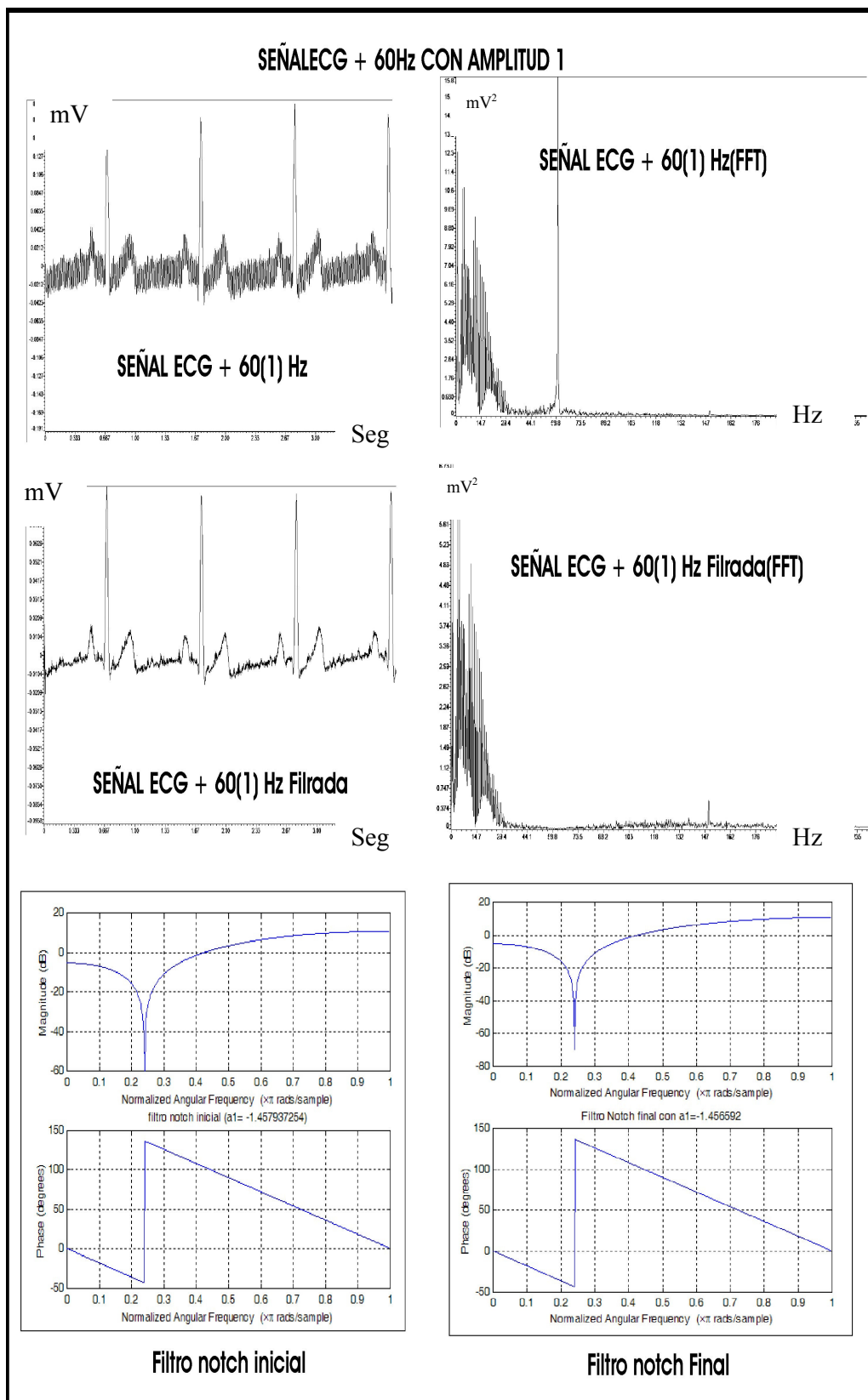


Figura 116. Señal ECG + 60 Hz con amplitud 1.

5.1 PRUEBAS REALIZADAS A LOS ALGORITMOS DE ANÁLISIS DE LA V.F.C.

Por medio del módulo de desarrollo DSK5402 se realizaron pruebas preliminares adquiriendo señales electrocardiográficas a sujetos normales sentados en una silla en un estado de reposo en instalaciones de la Fundación Cardiovascular de Colombia. Los datos adquiridos fueron exportados a Matlab para poder visualizar los resultados ilustrados en la figura 117 y hacer comparaciones con los resultados obtenidos por medio del método de remuestreo y FFT en el análisis en el dominio de la frecuencia. En la gráfica 118 se puede apreciar una comparación entre los resultados obtenidos del análisis espectral con el método de Lomb (figura 117) y el resultado obtenido por medio del método de remuestreo y FFT.

La figura 119 contiene el resultado obtenido de tomar señal electrocardiográfica de un sujeto sobre una mesa basculante. En la figura 119 se puede apreciar que los datos fueron adquiridos en dos posiciones diferentes de la mesa (0° y 60°) y en la figura 120 se comparados resultados obtenidos por el método de lomb y de la FFT.

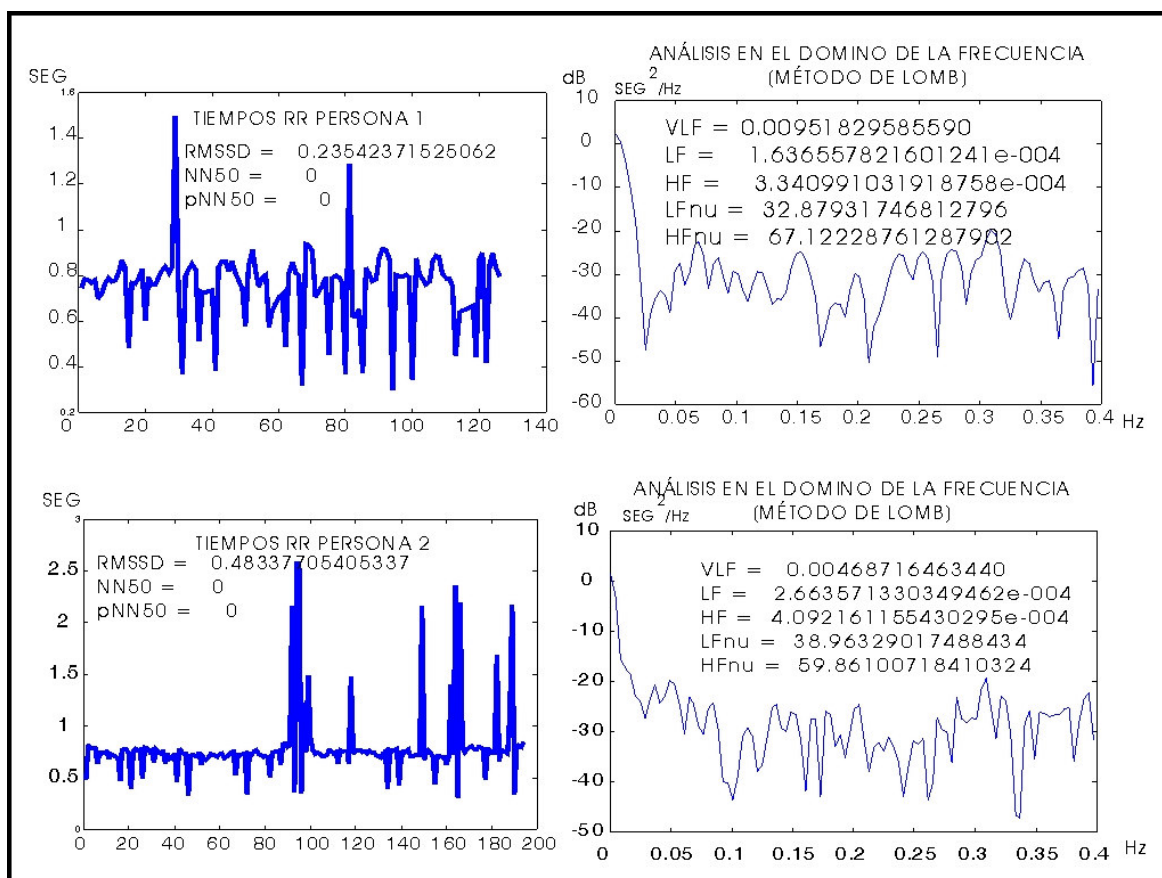


Figura 117. Pruebas de análisis de VFC en dos sujetos en reposo.

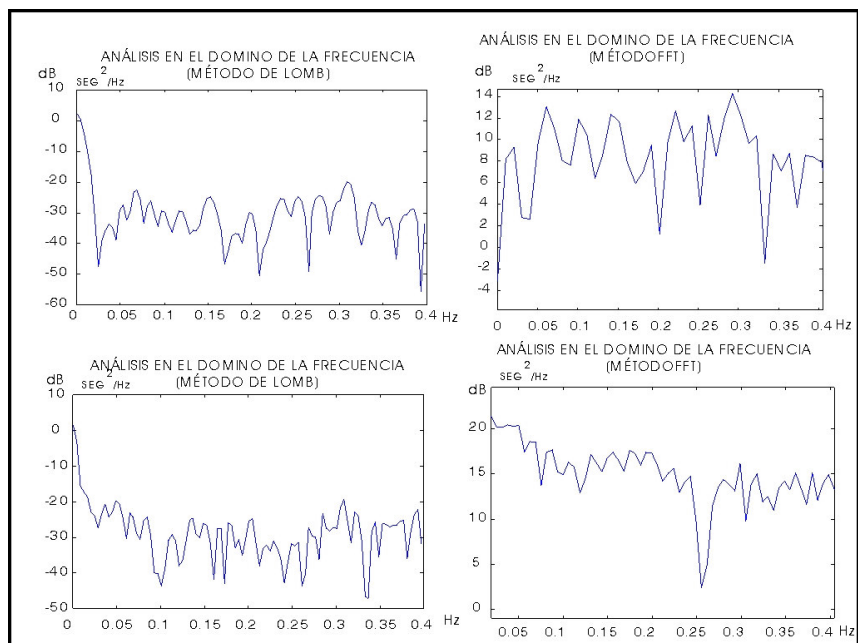


Figura 118. Comparación en Pruebas de análisis de VFC en dos sujetos en reposo (Lomb y FFT).

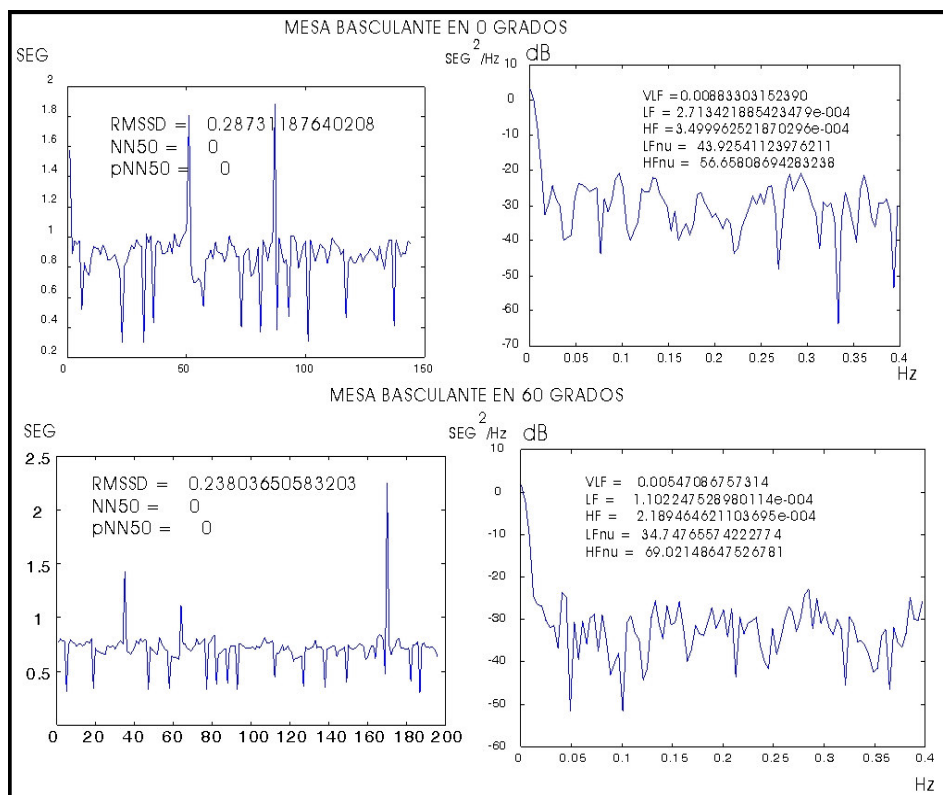


Figura 119. Prueba realizada en mesa basculante.

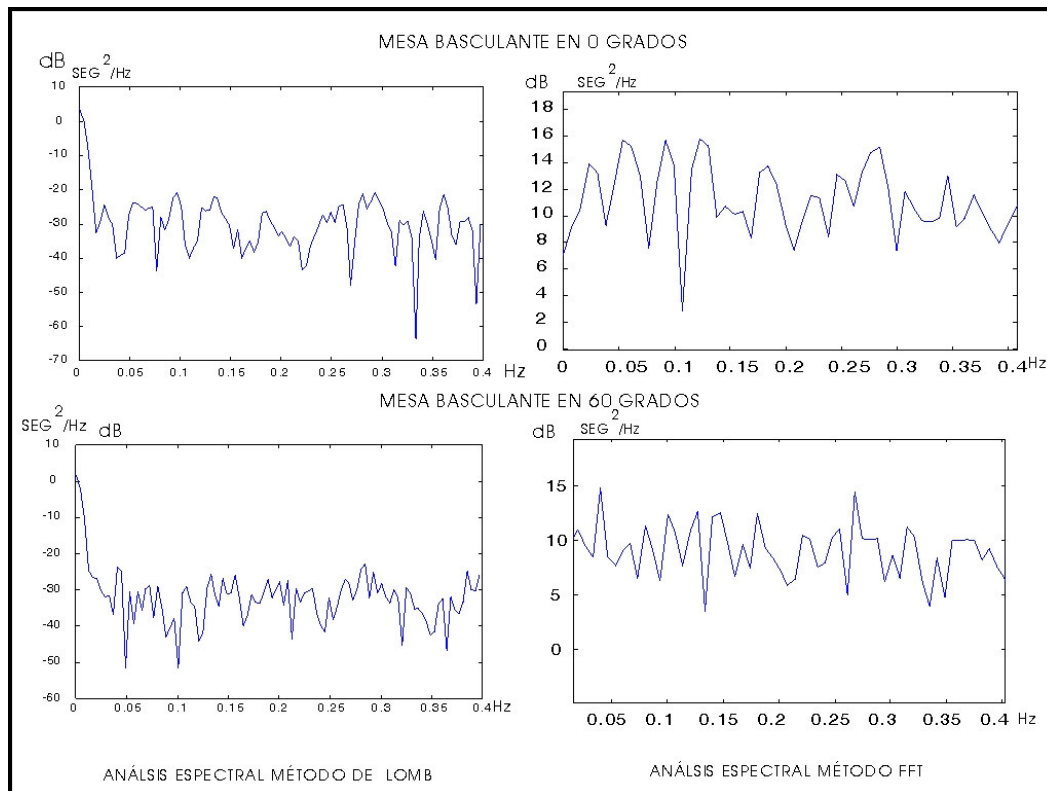


Figura 120. Comparación de espectros con lomb y FFT con datos tomados en mesa basculante.

Las comparaciones realizadas entre los resultados obtenidos por medio del método de la FFT y de Lomb en el análisis espectral de los tiempos RR dejan ver que hay diferencias entre cada técnica.

Estas diferencias pueden ser causadas por el efecto del proceso de interpolación y remuestreo utilizado en el análisis espectral por medio de la FFT.[35 y 36]

Con los resultados obtenidos se deja la opción al cuerpo médico para realizar futuros estudios clínicos donde se puedan apreciar las ventajas y desventajas durante el diagnóstico realizado con base al análisis espectral de la señal de ritmo cardiaco basado en la técnica de Lomb, la cual puede evitar las distorsiones causadas por el remuestreo.

Para comodidad del cuerpo médico se ha elaborado una interfaz gráfica en labView que toma los datos importados del software de desarrollo de la plataforma DSK5405, que ofrece la comodidad de archivar en disco duro los datos de las variables utilizadas en el programa elaborado para el DSP . (Ver figura 121)

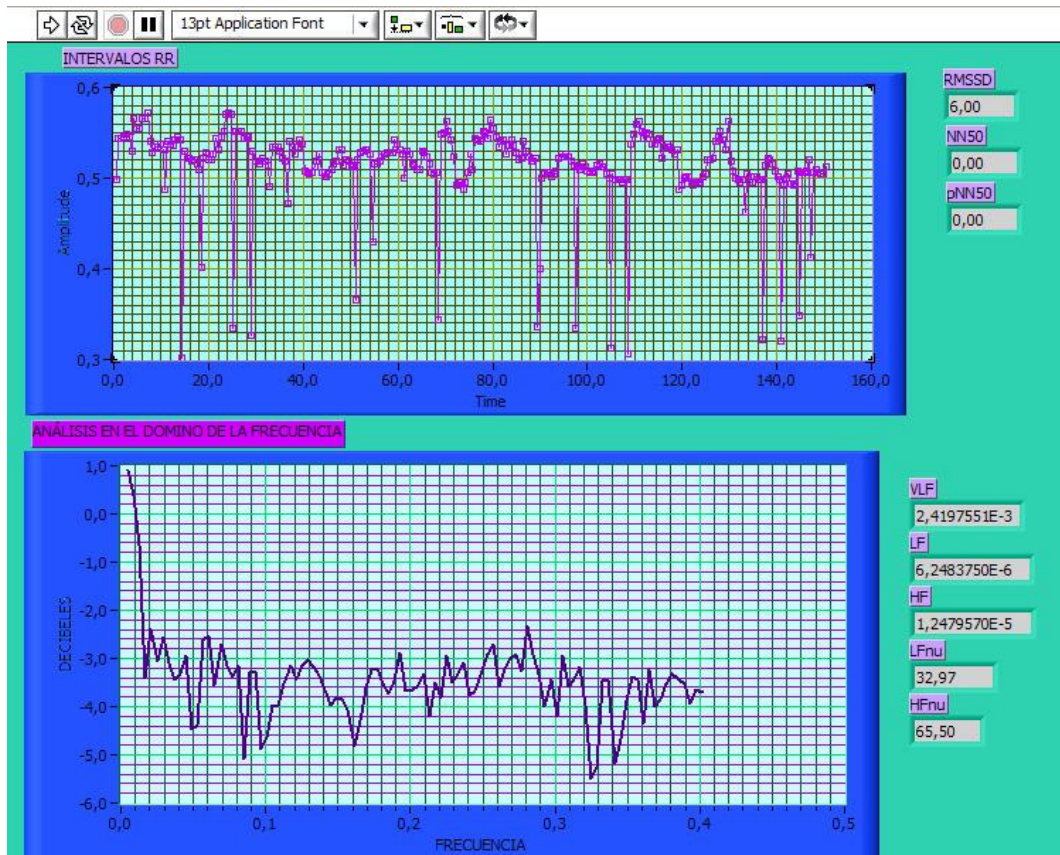


Figura 121. Interfaz gráfica para visualización de datos importados de la plataforma DSK5402.

6. CONCLUSIONES

Con la elaboración del trabajo de investigación: “Procesamiento Digital de Señal Electrocardiográfica con Tecnología DSP Orientado al Análisis de la Variabilidad de la Frecuencia Cardíaca” se ha profundizado en el estudio de las técnicas para la eliminación de interferencias y análisis de una señal de origen biológico. Este proyecto deja una serie de algoritmos en lenguaje C que ofrecen la facilidad de ser implementados en plataformas basadas en procesadores digitales de señales que ofrecen una gran ventaja para el estudio de señales de origen biológico gracias a su capacidad para la realización de operaciones matemáticas.

Dentro de este trabajo se ha realizado el diseño e implementación, en una plataforma basada en procesadores digitales de señales (DSP), de dos tipos de filtrado adaptativo que permiten la remoción del corrimiento de la línea de base de la señal electrocardiográfica y la disminución de las distorsiones causadas por la interferencia de la señal de 60 Hz.

Esta implementación se realizó con base a un estudio de las posibles distorsiones que pueden ocasionar las técnicas convencionales de filtrado digital en el procesamiento de señales biológicas, obteniendo una serie de algoritmos desarrollados con resultados satisfactorios. En este trabajo de investigación se pudieron obtener estructuras de filtrado que se adaptan mejorando su respuesta en el dominio de la frecuencia según las características de la señal de entrada y de la interferencia.

Dentro de los resultados obtenidos cabe mencionar el desarrollo de los algoritmos para la detección y cuantificación de los intervalos RR del complejo cardíaco que contribuyen a la implementación de un sistema para el análisis de la variabilidad de la frecuencia cardíaca basado en DSP.

El algoritmo desarrollado para la detección y cuantificación de los intervalos RR pueden ser considerado de bajo costo computacional si es comparado con otros que utilizan técnicas para el reconocimiento de la onda R basados en transformadas en el dominio tiempo-frecuencia, debido a que solo necesita el cálculo de una serie de ecuaciones en diferencia.

En este proyecto de investigación se realiza una implementación de algoritmos para el análisis espectral de los intervalos RR el cual deja la puerta abierta para la realización de futuras investigaciones en el estudio de esta serie de tiempo como herramienta para el seguimiento de enfermedades del sistema nervioso autónomo.

Como principal recomendación se sugiere la continuación de proyectos de investigación en el tema de la variabilidad de la frecuencia cardíaca apoyados en estudios clínicos que permitan caracterizar el método de Lomb para el análisis de la señal de ritmo cardíaco y mostrar las ventajas pudiese tener para el diagnóstico de enfermedades del sistema nervioso autónomo.

REFERENCIAS BIBLIOGRÁFICAS

- [1] GUYTON, Arthur. Tratado de Fisiología Médica. México 1997. P 112 – 127.
- [2] VELEZ, Hernán. BORRERO, Jaime. Cardiología. Corporación para Investigaciones Biológicas. Medellín. Colombia. 1985.
- [3] RUEDA, Oscar L. Curso de Electrocardiografía. Grupo de Electrocardiografía. Dpto. de ciencias Básicas – Fisiología. Facultad de Salud. Universidad Industrial de Santander. 2001.
- [4] Robert, E. Kleiger MD, Facc, Phyllis K. Stein, PhD. Time Domain Measurements of Heart Rate Variability. Ambulatory Electrocardiography. Cardiology Clinics. Washington University Medical Center. St. Luis. Missouri. USA. 10:p487-p489.
- [5] Dubin, Dale. Rapid Interpretation or EKG's. Cover Inc. Hong King. 2000. p1-p30.
- [6] WILCHES, Mauricio. Bioingeniería. Editorial Universidad de Antioquia. Colombia 1991. 609:p505-p504.
- [7] BERTRÁN N, MADAN G, PETER N. A Baseline Tracking Algorithm for Drift Reduction in Electrocardiography. IEEE Engineering in Medicine y Biology Society 10th Annual International Conference. CH2566-8/88/0000—1230. 1988 IEEE.
- [8] Bruce, Eugene. Biomedical Signal Procesing and Signal Modeling. John Wiley & ons, Inc. USA. 2001. p280- p282.
- [9] CASTELLANOS, G., PUERTA L, RÍOS A. Reducción en Tiempo Real de Interferencia de 60 Hz y de la Línea de Base en la Adquisición de Señales de ECG. Universidad Nacional de Colombia. Grupo Control y Procesamiento Digital de Señales. Manizales. Colombia 2000.p19-p25.
- [10] IOTECH. Signal Conditioning and PC-based Data Acquisition Handbook. Iotech. 1998.USA. P5 – P7.
- [11] SARMIENTO Luis O, Retamoso Alfonso. Interfase Para la Adquisición y Procesamiento de Señales Cardiacas, Diseño y Construcción. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. UIS. Colombia. 1998. <http://colibri.uis.edu.co/biblioteca>.
- [12] Pinto A. Miguel, González B. Javier. Cardiotacómetro Digital, Diseño y Construcción. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. UIS. Colombia. 2002. <http://colibri.uis.edu.co/biblioteca>.
- [13] OPPEHEIM Alan, WILLSKY Alan. Señales y Sistemas. Prentice- Hall. Méjico 1994.860: P549.
- [14] HEWLET PACKARD, The Funamentals of Signal Analysis. Hewlett Packard, Aplication Note number 243. P21
- [15] STREMLER, Ferrel. Sistemas de Comunicación. Alfa omega. Colombia. 1989. 690:P51,P125,P127.
- [16] FABER, Thomas. Impact of Electrocardiogram Recording Format on QT Interval Measuement an QT Dispersion Assessment. Department ofCardiological Sciences, St George's Hospital medical School, London 2000.P2.
- [17] Iturralde Torres, Pedro. El Centenario de la Electrocardiógrafo de Einthoven Parte II. Archivos de Cardiología de México. Instituto Nacional de

- Cardiología "Ignacio Chávez ". Vol. 71 Número2/abril- junio 2001:167-171. México DF 2000.
- [18] CARL, Joseph. Introduction to Biomedical Equipment Technology. Prentice Hall. USA 1998. 702: P125-231.
- [19] SEDRA, Adel. Circuitos Microelectrónicos. Oxford University Press. Mejiro 1998. p918
- [20] Tompkins, Wills J. Biomédical Digital Signal Processing. Prentice Hall. 1993 USA. 367: p100-116.
- [21] Treichler J, Johnson JR, Larimore, M. Theory and Design of Adaptive Filters. Topics in Digital Signal Processing. John Wiley & Sons. USA 1987. 343: p15-32.
- [22] Douglas, SC. Introduction to Adaptive Filters. Digital Signal Processing Handbook. Ed. Vijay Madisetti and Douglas B. Williams. Boca Raton. CRC Press LLC, 1999. Chapter 18 and 19.
- [23] Mohammed F, Barr R. Adaptive Digital Notch Filter Design on the Circle for the Removal of Powerline Noise From Biomedical Signals. IEEE Transactions on Biomedical Engineering. Vol 41, No 6 June 1994.
- [24] Ezenwa B, Gupta M, Nikforuk P, Prasad K. A Baseline Algorithm for Drift Reduction In Electrocardiography. IEEE Engineering in Medicine & Biology Society 10th Annual International Conference. Signal Processing and Analysis: Poster Session II . CH2566-8/88/000—1230. 1988 IEEE.
- [25] Chapra, Canale. Métodos Numéricos para Ingenieros.
- [26] CONNY M.A. Van Ravenswaaij – Arts, M.D. Luis A.A. Kollée, MD. Heart Rate Variability. American College of Physicians. USA. 1993. 118: p436-p437.
- [27] ROBERT, E. Kleiger MD, FACC, PHYLLIS K. Stein, PhD. Time Domain Measurements of Heart Rate Variability. Ambulatory Electrocardiography. Cardiology Clinics. Washington University Medical Center. St. Louis. Missouri. USA. 10:p487-p489.
- [28] MALICK, Malick. Clinical Guide To Cardiac Autonomic Test. Short- Term Measurement of Heart Rate Variability. Klumer Academic Publishers. London. 1998. Chapter six. p156 – p 160.
- [29] Romero JC, Pereira O, Licea M, Fagot O. Variabilidad de la frecuencia cardíaca en reposo para detectar neuropatía autonómica cardiovascular en diabéticos tipo I. Rev Cubana End 1999;10(1):25-37.
- [30] ZSOLT, Ori, MD. Heart Rate Variability. Frequency Domain Analysis. Ambulatory Electrocardiography. Northwestern University Medical School, Chicago. USA August 1992. 10: p499 – p501.
- [31] MALICK, Malick. Clinical Guide To Cardiac Autonomic Test. Short- Term Measurement of Heart Rate Variability. Klumer Academic Publishers. London. 1998. Chapter six. p161 – p 166.
- [32] MALICK, Malick. Clinical Guide To Cardiac Autonomic Test. Autonomic Responses to Blockades and Provocations. Klumer Academic Publishers. London. 1998. Chapter five. p135 – p 137.
- [33] Mendoza A, Ochoa J, Cardona J. OBTENCION Y ANALISIS ESPECTRAL DE LA VARIABILIDAD DE LA FRECUENCIA CARDIACA DE UNA SENAL ELECTROCARDIOGRAFICA UTILIZANDO LA TRANSFORMADA DE Fourier. Universidad Industrial de Santander. Bucaramanga 1997. <http://colibri.uis.edu.co>.

- [34] Lerma C, Infante O, José M. Sistema para el Análisis de la Variabilidad de la Frecuencia Cardíaca. Grupo de Biología Teórica. Instituto de Investigaciones Biomédicas, UNAM.
- [35] G.B. Moody. Spectral analysis of heart rate without resampling. In computers in cardiology. IEEE computer society press, 1993, p. IN Press.
- [36] W. H. Press and G.B. Rybicki. Fast Algorithm for Spectral Analysis of Unevenly Sampled Data. Astrophysical J. , vol 338, pp 277 – 280, 1989.
- [37] Laguna P, Moody G, Mark R. Power Spectral Density of Unevenly Sampled Data by Least-Square Analysis: performance and Application to Heart Rate Signals. IEEE Transactions on Biomedical Engineering, Vol 45, No 6, June 1998. p698-p715.
- [38] Press W, Teukolsky S, Vetterling W. Numerical Recipes in C. The Art of Scientific Computing. Cambridge University Press. Second Edition. Pp572-574. 1992. USA.

APENDICE I

1. Programa diezmado.mat

```
%Programa que simula el proceso de diezmado con n=4.
clc
close all
clear all
Fs=2000;
T=1/Fs;
%se simula una señal de entrada con 2000 muestras
%Con frecuencia de muestreo Fs=2000Hz.
for i=1:2000
    a(i)=sin(2*pi*10*T*i);%a=señal de entrada
end
stem(a(1:200))
title('Señal senosoidal')
B=fir1(6,0.25);%Coeficientes del filtro
p = filter(B,[1],a);%P=señal filtrada
z=0;
% proceso de decimación
for m=1:500
    z=4*m;
    p2(m)=p(z);
end
figure
stem(p2(1:50))
title('Señal diezmada')
```

2. Programa prommovil.mat

```
%Programa que simula el efecto de filtrado por medio de un
%Filtro de Promediado Móvil de Hanning
% Se utiliza como ejemplo una señal ECG con FS=333.333Hz.
%ec=señal electrocardiográfica. (ver archivo original)
B=[0.25 0.5 0.25];
T=1/333.333;
n=1:1000;
r=sin(2*pi*150*n*T);
ecr=ec+r';
plot(ecr)
figure
FREQZ(B,1)
title('Respuesta en frecuencia Filtro de Promediado Móvil')
y = filter(B,1,ecr);
figure
subplot(2,1,1)
plot(ecr)
title('Señal Contaminada')
subplot(2,1,2)
plot(y)
title('Señal filtrada')
```

3. Programa filtronotch.mat

```

%Filtro que simula el diseño de un filtro notch de orden 50
%para eliminar una interferencia de 60 hez en una señal
%ecg con FS= 333.333 Hz.
%ec=señal electrocardiográfica (ver archivo original)
T=1/333.333;
w=[0.3 0.4];
N=50;
B = FIR1(N,w,'stop');
stem(B)
title('Respuesta al impulso del filtro Notch')
figure
FREQZ(B,[1])
title('Respuesta en frecuencia del filtro Notch')
for i=1:1000
    r(i)=0.2*sin(2*pi*60*i*T);
    ecr(i)=ec(i)+r(i);
end
figure
subplot(2,1,1)
plot(ec)
title('señal ecg')
subplot(2,1,2)
plot(ecr)
title('señal ecg con ruido de 60 Hz')
y=filter(B,[1],ecr);
figure
subplot(2,1,1)
plot(ecr)
title('SEÑAL CONTAMINADA')
subplot(2,1,2)
plot(y)
title('SEÑAL FILTRADA')

```

4. Proframa efectofiltronotch.mat

```

%Programa que simula el efecto de un filtro notch de
orden 50 en componentes cercanas a la frecuencia que se
desea eliminar (60 Hz).
%señal de muestra:
n=1:2000;
fs=500;
s1=sin(2*pi*10*n*(1/fs));
s2=sin(2*pi*50*n*(1/fs));
s3=sin(2*pi*58*n*(1/fs));
s4=sin(2*pi*61*n*(1/fs));
s5=sin(2*pi*70*n*(1/fs));
s=s1+s2+s3+s4+s5;

```

```

%Interferencia

```

```

v1=0.5*sin(2*pi*60*n*(1/fs));
x=s+v1;
w=[0.23 0.25];
N=50;
B = FIR1(N,w,'stop');
stem(B)
title('Respuesta al impulso del filtro Notch')
figure
FREQZ(B,[1])
title('Respuesta en frecuencia del filtro Notch de orden
50')
for k=1:1000
    w(k)=fs*(k-1)/2000;
end
ecfn=filter(B,[1],x);
figure
subplot(2,1,1)
plot(s)
title('señal de entrada')
subplot(2,1,2)
plot(x)
title('señal de contaminada')
figure
subplot(2,1,1)
xf=abs(fft(x));
plot(w,xf(1:1000))
grid on
title('Espectro señal contaminada')
subplot(2,1,2)
scf=abs(fft(ecfn));
plot(w,scf(1:1000))
grid on
title('Espectro señal filtrada con Notch')

```

5. Programa derivadores.mat

```

%Simulación del efecto de filtros derivadores sobre una
señal ecg con fs=333.333Hz
%ec = señal electrocardiográfica (ver archivo original)
T=1/333.333;
%derivada de dos puntos
B=[333.333 -333.333];
freqz(B,[1])
title('respuesta en Frecuencia filtro diferencia de dos
puntos')
y=filter(B,[1],ec);
figure
subplot(2,1,1)
plot(ec)
title('señal ecg')

```

```

subplot(2,1,2)
plot(y)
title('filtro diferencia de dos puntos')
%diferencia central de tres puntos
B1=[166.6665  0 -166.6665];
figure
freqz(B1,[1])
title('respuesta en Frecuencia filtro diferencia central de
tres puntos')
y2=filter(B1,[1],ec);
figure
subplot(2,1,1)
plot(ec)
title('señal ecg')
subplot(2,1,2)
plot(y2)
title('filtro diferencia central de tres puntos')
% aproximación de polinomios por mínimos cuadrados
B2=[ 2*33.3333  33.3333 0 - 33.3333 -2* 33.3333];
figure
freqz(B2,[1])
title('respuesta en Frecuencia aproximación de polinomios
por Mínimos Cuadrados')
y3=filter(B1,[1],ec);
figure
subplot(2,1,1)
plot(ec)
title('señal ecg')
subplot(2,1,2)
plot(y3)
title('filtro aproximación por mínimos cuadrados de
polinomios ')
%segunda derivada
B3=[1 -2 -1];
figure
freqz(B3,[1])
y4=filter(B3,[1],ec);
title('Respuesta en frecuencia segunda derivada ')

figure
subplot(2,1,1)
plot(ec)
title('señal ecg')
subplot(2,1,2)
plot(y4)
title('filtro segunda derivada ')

```

6. Programa filtronotchcerosfir.mat

```
%Programa que simula del diseño de un filtro FIR notch
%de segundo orden aplicando el método de colocación de
ceros.
%ec= señal electrocardiográfica (ver archivo original)

T=1/333.333;
%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*333.333;
n=1:1000;
%interferencia simulada
v1=0.2*sin(2*pi*60*n*T);
%contaminación de la señal
x=ec+v1';
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a0=1;
a2=1;
G=[1 a1 1];
freqz(G,[1])
z=filter(G,1,x);
for k=1:500
    w(k)=333.333*(k-1)/1000;
end
xfft=abs(fft(x));
zfft=abs(fft(z));
figure
subplot(2,2,1)
plot(x)
title('señal contaminada con ruido con f=60 Hz')
subplot(2,2,2)
plot(z)
title('señal filtrada')
subplot(2,2,3)
plot(w,xfft(1:500))
title('Espectro señal ecg contaminada con ruido de f = 60
Hz')
subplot(2,2,4)
plot(w,zfft(1:500))
title('Espectro señal filtrada')
```

7. Programa filtronotchcerosiir.mat

```
%Este programa ejecuta un filtro notch IIR de segundo orden
notch con adaptación de un cero y un polo.
%ec= señal electrocardiográfica (ver archivo original)
n=1:1000;
ec=ec';
%Interferencia
v1=0.09*sin(2*pi*60*n*(1/333.333));
x=ec+v1;
%Calculo de coeficientes iniciales
%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*333.333;
%Coeficientes Numerador
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a=a1;
a0=1;
a2=1;
%coeficientes denominador
alfa=0.2;
b0=1;
b1=-2*alfa*cos(b);
b2=alfa*alfa;
%función de transferencia del notch
N1=[a0 a1 a2];
D1=[b0 b1 b2];
%polos del notch
pli=0.5*(-b1+sqrt(b1*b1-4*b2));
p2i=0.5*(-b1-sqrt(b1*b1-4*b2));
figure
freqz(N1,D1)
title('Respuesta en frecuencia del notch')
e=filter(N1,D1,x);
figure
subplot(2,1,1)
plot(ec)
title('señal de entrada')
subplot(2,1,2)
plot(x)
title('señal de contaminada')
for k=1:500
    w(k)=333.333*(k-1)/1000;
end
figure
subplot(2,1,1)
plot(x)
title('señal contaminada')
subplot(2,1,2)
```

```

plot(e)
title('señal Salida')
figure
subplot(2,1,1)
xf=10*abs(fft(x));
plot(w,xf(1:500))
title('Espectro señal contaminada')
subplot(2,1,2)
ef=10*abs(fft(e));
plot(w,ef(1:500))
title('Espectro señal salida')

```

8. Programa fircancelacion.mat

```

%funcionamiento del sistema de cancelación cancelación para
eliminar ruido simulando la prescencia
%ruido aditivo que equivale a una función seno con
frecuencia igual a 60 Hz en una señal
%compuesta por una suma de senosoidales.
%señal de muestra:
n=1:2000;
fs=500;
s1=sin(2*pi*10*n*(1/fs));
s2=sin(2*pi*50*n*(1/fs));
s3=sin(2*pi*58*n*(1/fs));
s4=sin(2*pi*61*n*(1/fs));
s5=sin(2*pi*70*n*(1/fs));
s=s1+s2+s3+s4+s5;
%Interferencia
v1=0.09*sin(2*pi*60*n*(1/fs));
x=s+v1;
%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*fs;
%Calculo de coeicientes iniciales
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a=a1;
a0=1;
a2=1;
B=[1 a1 1];
%función de transferencia del notch inicial
figure
freqz(B,[1])
title('Respuesta en frecuencia del notch FIR')

```



```

figure
subplot(2,1,1)
plot(s)
title('señal de entrada')
subplot(2,1,2)
plot(x)
title('señal de contaminada')
%Señal de reerencia V2
v2=0.09*sin(2*pi*60*n*(1/fs));
% Proceso de adaptación
%G(Z)=go[0 -a1 -1] Función de Transferencia del Estimador
%z=Señal esimada
    v(1)=a1;
for j=1:1000
    B1=[0 -a1 -1];
    z=filter(B1,[1],v2);
    e=x-z;
    c=a1;
    for i=2:1000
        q=c+0.00000001*e(i)*x(i-1);
        c=q ;
    end
    v(j)=c;
a1=c;
end
for k=1:1000
    w(k)=fs*(k-1)/2000;
end
figure
subplot(2,1,1)
plot(x)
title('señal contaminada')
subplot(2,1,2)
plot(e)
title('señal Salida')
figure
subplot(3,1,1)
xf=abs(fft(x));
plot(w,xf(1:1000))
grid on
title('Espectro señal contaminada')
subplot(3,1,2)
en=abs(fft(e));
plot(w,en(1:1000))
grid on
title(' Espectro señal salida del sistema por cancelación
')
subplot(3,1,3)
scf=abs(fft(s));
plot(w,scf(1:1000))

```

```

grid on
title('Espectro señal original')
wc=[0.23 0.25];
N=50;
B1 = FIR1(N,wc,'stop');
ecfn=filter(B1,[1],x);
figure
subplot(2,1,1)
plot(e)
title('señal salida notch adaptado')
subplot(2,1,2)
plot(ecfn)
title('señal Salida notch')
figure
subplot(2,1,1)
plot(w,en(1:1000))
title('Espectro señal salida sistema por cancelación ')
subplot(2,1,2)
d=abs(fft(ecfn));
plot(w,d(1:1000))
title('señal Salida notch FIR convencional de orden 50')

```

9. Programa lineadabase.mat

```

%Programa que realiza la simulación del proceso de
%eliminación del corrimiento de la línea de base de una
%señal electrocardiográfica.
%ec= señal electrocardiográfica (ver archivo original)
fs=333.333;
T= 0.001;
n=1:2000;
%P=señal de entrada
p=ec+6;
vma(1:2000)=0;
%Promedio de datos de la señal ECG
z=0;
for r=1:10
    c=0;
    for i=1:200
        c=p(i+z)+c;
        xi=c;
    end
    a(r)=(1/200)*c;
    z=z+200;
    vma(z)=a(r);
end
figure
plot(n,p,'y',n,vma,'g')
grid on
%creación de la base de tiempo.

```

```

c=0;
b=0;
for i=1:2000
    t(i)=c+T;
    c=t(i);
end
for k=1:10
    b=b+200;
    tb(k)=t(b);
end
%arreglo de datos. Se multiplican con la matriz inversa
%para dar solución al sistema de ecuaciones.
n=11;
m=5;
tb=[0 tb ];
a=[p(1) a];
s(1)=n;
for r=1:(2*m)
    c=0;
    for r1=1:n
        c=c+tb(r1)^r;
    end
    s(r+1)=c;
end

for i=1:m+1
    c=0;
    l=i-1;
    for j=1:n
        c=c+(a(j)*tb(j)^l);
    end
    d(i)=c;
end

%Cálculo de la matriz inversa
a=0;
for i=1:m+1
    for j=1:m+1
        coef(i,j)=s(j+a);
    end
    a=a+1;
end
INV=inv(coef);
%multiplicación y cálculo de la solución del sistema de
ecuaciones.
%Se obtienen los coeficientes del polinomio
a=INV*d';
%Calculo de la función estimada de la línea de base
utilizando los coeficientes encontrados
for i=1:2000;
    c=0;

```

```

        for l=1:m+1
            g=l-1;
            c=c+t(i)^(g)*a(l);
        end
        p2(i)=c;
    end
    figure
    plot(t,vma,'b',t,p2,'g',t,p,'r')
    %RESTA: Señal contaminada - Función estimada.
    T=p-p2';
    figure
    subplot(2,1,1)
    plot(t,p)
    title('señal contaminda')
    grid on
    subplot(2,1,2)
    plot(t,T)
    title('señal filtrada')
    grid on
    %Filtro fir pasa altos
    q=333.333/2;
    wn=0.05/q;
    A=fir1(10,wn,'high');
    yf=filter(A,[1],p);
    figure
    subplot(2,1,1)
    plot(t,T)
    title('señal filtrada con Filtro por cancelación')
    grid on
    subplot(2,1,2)
    plot(t,yf)
    title('señal filtrada con filtro FIR pasa altos')
    grid on

```

10. Programa psdinterpolación.mat

```

%Programa que toma un conjunto de intervalos RR (nr001.mat)
de un sujeto normal tomados de una base de datos y la señal
obtenida por medio de interpolación y remuestreo.
load nr001
RR=nr001(1:1000);
N=length(RR);
c=0;
for i =1:N
    t(i)=c+RR(i);
    c=t(i);
end

```

```

plot(t,RR,'o-')
title('Tiempos RR')
xlabel('Ocurrencia de intervalos (Seg)')
ylabel('Duración (Seg)')
mini=min(RR);
Tp=mini/4;
TN=max(t);
k=0;
L=0;
%interpolación lineal.
for i =2:N
    m=(RR(i)-RR(i-1))/(t(i)-t(i-1));
    b=RR(i)-m*t(i);
    t1=t(i-1):Tp:t(i);
    n=length(t1);
    for s=1:n
        RRI(s+L)=m*t1(s)+b;
        ts(s+L)=t1(s);
    end
    L=L+n;
end

figure
plot(ts,RRI,'o-')
title('Tiempos RR (señal obtenida por interpolación)')
xlabel('Ocurrencia de intervalos (Seg)')
ylabel('Duración (Seg)')
NR=length(RRI);
RRI=RRI-mean(RRI);
V=hamming(NR);
RIV=V.*RRI';
RIF=abs(fft(RIV));
F=1/Tp;
NR=NR-1;
for i=1:500
    wf(i)=(i-1)*F/NR;
end
RIF=10*log10(RIF);
figure
plot(wf,RIF(1:500))
title('PSD Estimada por el método DFT')
xlabel('Frecuencia Hz')
%Espectro obtenido por medio del método de lomb.
dw=wf(2);
w=0.0000001:dw:1;
xm=(1/N)*sum(RR);
var=(1/(N-1))*sum((RR-xm).^2);
N=length(RR);
N1=length(w);
xm=0;
s=0;

```

```

for i=1:N
    xm= xm + RR(i);
    W2= 0.54+0.46*cos(2*pi*t(i)/t(N));
    RR(i)=RR(i)*W2;
    s=s+W2;
end
xm=xm/N ;
NOR=N/(s*s);
for r=1:N1
    c=0;
    b=0;
    Sh=0;
    Ch=0;
    S2=0;
    C2=0;
    for i=1:N
        Sh=Sh+( (RR(i)-xm)*sin( 2*pi*w(r)*(t(i)) ) );
        Ch=Ch+( (RR(i)-xm)*cos(2*pi*w(r)*(t(i))) );
        S2=S2+sin(2*pi*w(r)*(t(i)));
        C2=C2+cos(2*pi*w(r)*(t(i))) ;
    end
    tao=(atan(S2/C2))/(4*pi*w(r));
    A=Ch*cos(2*pi*w(r)*tao)+Sh*sin(2*pi*w(r)*tao);
    B=Sh*cos(2*pi*w(r)*tao)-Ch*sin(2*pi*w(r)*tao);
    A2=(N/2)+0.5*C2*cos(4*pi*w(r)*tao)+0.5*S2*sin(4*pi*w(r)*tao);
    B2=(N/2)-0.5*C2*cos(4*pi*w(r)*tao)-0.5*S2*sin(4*pi*w(r)*tao);
    P(r)=NOR*((A^2)/A2)+((B^2)/B2);
end
Pl=10*log10(P);
figure
plot(w,Pl)
title('Análisis Espectral (Método de lomb)')
figure
subplot(2,1,1)
plot(wf,RIF(1:500))
title('PSD Estimada por el método DFT')
xlabel('Frecuencia Hz')
subplot(2,1,2)
plot(w,Pl)
title('Análisis Espectral (Método de lomb)')

```

11. Programa psdlombsenos.mat

```
%Programa que toma una señal compuesta por suma de tres
señales senosoidales con frecuencias de 4, 6 y 8 Hz.
% El programa ilustra el resultado obtenido al calcular el
espectro en potencia de la señal
% por medio del método de la FFT y el método de Lomb.
N=500;
T=0.01;
n=1:N;
f1=4;
f2=6;
f3=8;
x=cos(2*pi*f1*T*n)+cos(2*pi*f2*T*n)+cos(2*pi*f3*T*n);
c=0;
for i=1:N
    t(i)=c+T;
    c=t(i);
end
c=round(rand(N,1));
x1=x.*c';
k=0;
for i=1:N
    if x1(i)==0
        x1(i)=x1(i);
    else
        k=k+1;
        xr(k)=x1(i);
        tr(k)=t(i);
    end
end
subplot(2,1,1)
stem(t,x)
title('Señal uniformemente muestreada')
subplot(2,1,2)
stem(tr,xr)
title('Señal no uniformemente muestreada')
%espectro señal uniformemente muestreada con FFT
W=hanning(N);
xw=x.*W';
xfft=abs(fft(xw));
xfft=xfft(1:(N/2));
Fs=1/T;
for h=1:N/2
    wk(h)=Fs*(h-1)/(N);
end
figure
plot(wk,xfft)
title('Espectro con FFT')
```

```

%espectro señal uniformemente muestreada con Lomb
dw=0.5;
w=0.0001:dw:10;
xm=(1/N)*sum(x);
var=(1/(N-1))*sum((x-xm).^2);
N=length(x);
N1=length(w);
xm=0;
s=0;
for i=1:N
    xm= xm + x(i);
    W2= 0.54+0.46*cos(2*pi*t(i)/t(N));
    x(i)=x(i)*W2;
    s=s+W2;
end
xm=xm/N
NOR=N/(s*s)

for r=1:N1
    c=0;
    b=0;
    Sh=0;
    Ch=0;
    S2=0;
    C2=0;
    for i=1:N
        Sh=Sh+( (x(i)-xm)*sin( 2*pi*w(r)*(t(i)) ) );
        Ch=Ch+( (x(i)-xm)*cos(2*pi*w(r)*(t(i)))));
        S2=S2+sin(2*pi*w(r)*(t(i)));
        C2=C2+cos(2*pi*w(r)*(t(i))) ;
    end
    tao=(atan(S2/C2))/(4*pi*w(r));

    A=Ch*cos(2*pi*w(r)*tao)+Sh*sin(2*pi*w(r)*tao);
    B=Sh*cos(2*pi*w(r)*tao)-Ch*sin(2*pi*w(r)*tao);

    A2=(N/2)+0.5*C2*cos(4*pi*w(r)*tao)+0.5*S2*sin(4*pi*w(r)*tao
);
    B2=(N/2)-0.5*C2*cos(4*pi*w(r)*tao)-
0.5*S2*sin(4*pi*w(r)*tao);
    P(r)=NOR*((A^2)/A2)+((B^2)/B2));
end
figure
plot(w,P)
title('espectro de lomb(muestreo uniforme)')

```



```

%espectro señal no uniformemente muestreada(Metodo de Lomb)
x=xr;
t=tr;
dw=0.5;
w=0.0001:dw:10;
xm=(1/N)*sum(x);
var=(1/(N-1))*sum((x-xm).^2);
N=length(x);
N1=length(w);
xm=0;
s=0;
for i=1:N
    xm= xm + x(i);
    W2= 0.54+0.46*cos(2*pi*t(i)/t(N));
    x(i)=x(i)*W2;
    s=s+W2;
end
xm=xm/N ;
NOR=N/(s*s);

for r=1:N1
    c=0;
    b=0;
    Sh=0;
    Ch=0;
    S2=0;
    C2=0;
    for i=1:N
        Sh=Sh+( (x(i)-xm)*sin( 2*pi*w(r)*(t(i)) ) );
        Ch=Ch+( (x(i)-xm)*cos(2*pi*w(r)*(t(i)))));
        S2=S2+sin(2*pi*w(r)*(t(i)));
        C2=C2+cos(2*pi*w(r)*(t(i))) ;
    end
    tao=(atan(S2/C2))/(4*pi*w(r));
    A=Ch*cos(2*pi*w(r)*tao)+Sh*sin(2*pi*w(r)*tao);
    B=Sh*cos(2*pi*w(r)*tao)-Ch*sin(2*pi*w(r)*tao);

    A2=(N/2)+0.5*C2*cos(4*pi*w(r)*tao)+0.5*S2*sin(4*pi*w(r)*tao);
    B2=(N/2)-0.5*C2*cos(4*pi*w(r)*tao)-0.5*S2*sin(4*pi*w(r)*tao);
    P1(r)=NOR*((A^2)/A2)+((B^2)/B2);
End
figure
plot(w,P1)
title('espectro de lomb (muestreo no uniforme)')
%espectro señal no uniformemente muestreada(remuestreo y FFT)

```

```

mini=min(tr);
Tp=mini/4
TN=max(tr);
N=length(xr);
k=0;
L=0;
%interpolación lineal.
for i =2:N
    m=(xr(i)-xr(i-1))/(tr(i)-tr(i-1));
    b=xr(i)-m*tr(i);
    t1=tr(i-1):Tp:tr(i);
    n=length(t1);
    for s=1:n
        XRI(s+L)=m*t1(s)+b;
        ts(s+L)=t1(s);
    end
    L=L+n-1;
end
figure
plot(ts,XRI)
title('señal obtenida por interpolación')
xlabel('Tiempo (Seg)')
ylabel('amplitud')
NR=length(XRI);
V=hamming(NR);
XIv=V.*XRI';
XIF=abs(fft(XIv));
F=1/Tp;
for i=1:NR
    w(i)=(i-1)*F/NR;
end
figure
plot(w,XIF)
title('PSD Estimada por el método DFT')
xlabel('Frecuencia Hz')
figure
subplot(2,1,1)
plot(ts,XRI)
title('Señal obtenida por interpolación')
subplot(2,1,2)
plot(w,XIF)
title('Espectro en frecuencia (FFT)')

```

11. Programa fircanreferencia.mat

```
%programa que simula el desempeño del sistema por
cancelación para estimar la señal de error e(n)
% en una señal compuesta por la suma de diferentes señales
senosoidales con una interferencia aditiva
%de 60 Hz con el fin de evaluar el verdadero desempeño de
este tipo filtro y comprobar %
%que solo remueve la componente de 60 Hz.
%señal de muestra:
n=1:2000;
fs=500;
s1=sin(2*pi*10*n*(1/fs));
s2=sin(2*pi*50*n*(1/fs));
s3=sin(2*pi*58*n*(1/fs));
s4=sin(2*pi*61*n*(1/fs));
s5=sin(2*pi*70*n*(1/fs));
s=s1+s2+s3+s4+s5;
%Interferencia
v1=0.09*sin(2*pi*60*n*(1/fs));
x=s+v1;
%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*fs;
%Calculo de coeicientes iniciales
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a=a1;
a0=1;
a2=1;
B=[1 a1 1];
%función de transferencia del notch inicial
figure
freqz(B,[1])
title('Respuesta en frecuencia del notch inicial')
figure
subplot(2,1,1)
plot(s)
title('señal de entrada')
subplot(2,1,2)
plot(x)
title('señal de contaminada')
%Señal de reerencia V2
v2=0.05*sin(2*pi*60*n*(1/fs));
% Proceso de adaptación
%G(Z)=go[0 -a1 -1] Función de Transferencia del Estimador
%z=Señal estimada
v(1)=a1;
for j=1:1000
    B1=[0 -a1 -1];
    z=filter(B1,[1],v2);
```

```

    e=x-z;
    c=a1;
    for i=2:1000
        q=c+0.00000001*e(i)*x(i-1);
        c=q ;
    end
    v(j)=c;
    a1=c;
end
for k=1:1000
    w(k)=fs*(k-1)/2000;
end
figure
subplot(2,1,1)
plot(x)
title('señal contaminada')
subplot(2,1,2)
plot(e)
title('señal Salida')
figure
subplot(3,1,1)
xf=abs(fft(x));
plot(w,xf(1:1000))
grid on
title('Espectro señal contaminada')
subplot(3,1,2)
en=abs(fft(e));
plot(w,en(1:1000))
grid on
title('Espectro señal salida del sistema por cancelación ')
subplot(3,1,3)
scf=abs(fft(s));
plot(w,scf(1:1000))
grid on
title('Espectro señal original')
%Filtro notch de orden 50 diseñado por matlab
B=fir1(50,[0.23 0.25],'stop')
ecfn=filter(B,[1],x);
figure
subplot(2,1,1)
plot(e)
title('señal salida sistema por cancelación')
subplot(2,1,2)
plot(ecfn)
title('señal Salida notch')
figure
subplot(2,1,1)
plot(w,en(1:1000))
title('Espectro señal salida del sistema por cancelación
adaptado)

```

```

subplot(2,1,2)
d=abs(fft(ecfn));
plot(w,d(1:1000))
title('señal Salida notch FIR convencional orden 50
(diseñado en matlab)')

```

12. Programa fircanrefestimada.mat

```

%Programa que implementa el método de estimación de la
%función V2 basado en el cálculo de los coeficientes de
%Fourier. señal de muestra:
n=1:2000;
fs=500;
s1=sin(2*pi*10*n*(1/fs));
s2=sin(2*pi*58*n*(1/fs));
s3=sin(2*pi*52*n*(1/fs));
s4=sin(2*pi*61*n*(1/fs));
s5=sin(2*pi*70*n*(1/fs));
s=s1+s2+s3+s4+s5;
%Interferencia
v1=sin(2*pi*60*n*(1/fs)+0.45);
%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*fs;
x=s+v1;
%Calculo de coeficientes iniciales
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a=a1;
a0=1;
a2=1;
%función de transferencia del notch inicial
N1=[a0 a1 a2];
figure
freqz(N1,[1])
title('Respuesta en frecuencia del notch inicial')
figure
subplot(2,1,1)
plot(s)
title('señal de entrada')
subplot(2,1,2)
plot(x)
title('señal de contaminada')
%estimación de la amplitud de la interferencia de 60 Hz
c=0;
%cálculo de la parte real del coeficiente de la señal x[n]
for j=1:2000
    real=c+x(j)*cos(2*pi*240*j/2000);
    c=real;
end

```

```

%cálculo de la parte imaginaria del coeficiente de la señal
x[n]
d=0;
for q=1:2000
    img=d+x(q)*sin(2*pi*240*q/2000);
    d=img;
end
fase=atan(img/real);
fase=(pi/2)-fase;
A=(sqrt(real*real+img*img))*2/2000;
%Señal de referencia V2
v2=A*sin(2*pi*60*n*(1/fs)+fase);
% Proceso de adaptación
%G(Z)=go[0 -a1 -1] Función de Transferencia del Estimador
%z=Señal esimada
V(1)=a1;
G=[0 -a1 -1];
z=filter(G,[1],v2);
e=x-z;
c=a1;
for i=2:2000
    q=c+0.00000001*e(i)*x(i-1);
    c=q ;
    V(i)=c;
end

alf=c;
G=[0 -alf -1];
z=filter(G,[1],v2);
ef=x-z;
for k=1:1000
    w(k)=fs*(k-1)/2000;
end
figure
subplot(2,1,1)
plot(x)
title('señal contaminada')
subplot(2,1,2)
plot(ef)
title('señal Salida')
figure
subplot(3,1,1)
xf=abs(fft(x));
plot(w,xf(1:1000))
grid on
title('Espectro señal contaminada')
subplot(3,1,2)
efn=abs(fft(ef));
plot(w,efn(1:1000))
grid on
title('Espectro señal salida del sistema por cancelación')

```

```

subplot(3,1,3)
scf=abs(fft(s));
plot(w,scf(1:1000))
grid on
title('Espectro señal original')
wc=[0.23 0.25];
N1=50;
B1 = FIR1(N1,wc,'stop');
ecfn=filter(B1,[1],x);
figure
subplot(2,1,1)
plot(ef)
title('señal salida del sistema por cancelación')
subplot(2,1,2)
plot(ecfn)
title('señal Salida notch')
d=abs(fft(ef));
figure
subplot(2,1,1)
plot(w,d(1:1000))
title('Espectro señal salida del sistema por cancelación')
subplot(2,1,2)
d2=abs(fft(ecfn));
plot(w,d2(1:1000))
title('señal Salida notchs')

```

13. programa simulacionrampa.mat

```

%programa que simula variaciones en el valor de la
%frecuencia de una señal de 60 Hz a través de una función
%rampa que permita que el una variación de  $\pm 0.1$  Hz
n=1:1000;
n1=1:500;
T=1/500;
c=0;
for i=1:1000
    t(i)=c+T;
    c=t(i);
end
y=0.2*n1*T+59.9;
R(1:500)=y;
R(501:1000)=y;
plot(t,R)
for n=1:1000
    v1(n)=0.5*sin(2*pi*R(n)*n*T);
    v1n(n)=sin(2*pi*60*n*T);
end
figure
plot(t,v1)
v1f=abs(fft(v1));
v1nf=abs(fft(v1n));

```

```

for k=1:500
    w(k)=500*(k-1)/1000;
end
figure
subplot(2,1,1)
plot(w,v1f(1:500))
title('Señal seno de frecuencia variable')
subplot(2,1,2)
plot(w,v1nf(1:500))
title('Señal seno de frecuencia fija')
%señal de muestra:
n=1:1000;
fs=500;
s1=sin(2*pi*10*n*(1/fs));
s2=sin(2*pi*50*n*(1/fs));
s3=sin(2*pi*58*n*(1/fs));
s4=sin(2*pi*65*n*(1/fs));
s5=sin(2*pi*70*n*(1/fs));
s=s1+s2+s3+s4+s5;
%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*fs;
x=s+v1;
%Calculo de coeficientes iniciales
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a=a1;
a0=1;
a2=1;
%función de transferencia del notch inicial
N1=[a0 a1 a2];
figure
freqz(N1,[1])
title('Respuesta en frecuencia del notch inicial')
figure
subplot(2,1,1)
plot(s)
title('señal de entrada')
subplot(2,1,2)
plot(x)
title('señal de contaminada')
%estimación de la amplitud de la interferencia de 60 Hz
c=0;
%cálculo de la parte real del coeficiente de la señal x[n]
for j=1:1000
    real=c+x(j)*cos(2*pi*120*j/1000);
    c=real;
end
%cálculo de la parte imaginaria del coeficiente de la señal
x[n]

```



```

d=0;
for q=1:1000
    img=d+x(q)*sin(2*pi*120*q/1000);
    d=img;
end
fase1=atan(img/real);
fase=(pi/2)-fase1;
A=(sqrt(real*real+img*img))*2/1000;
R1=0.05*sin(2*pi*0.2*n*T)+60;
for n=1:1000
    v2(n)=A*sin(2*pi*R1(n)*n*T);
end
% Proceso de adaptación
%G(Z)=go[0 -a1 -1] Función de Transferencia del Estimador
%z=Señal esimada
V(1)=a1;
G=[0 -a1 -1];
z=filter(G,[1],v2);
e=x-z;
c=a1;
for i=2:1000
    q=c+0.000001*e(i)*x(i-1);
    c=q ;
    V(i)=c;
end
alf=c;
G=[0 -alf -1];
z=filter(G,[1],z);
ef=x-z;
for k=1:500
    w(k)=fs*(k-1)/1000;
end
figure
subplot(2,1,1)
plot(x)
title('señal contaminada')
subplot(2,1,2)
plot(ef)
title('señal Salida')
figure
subplot(3,1,1)
xf=abs(fft(x));
plot(w,xf(1:500))
grid on
title('Espectro señal contaminada')
subplot(3,1,2)
efn=abs(fft(ef));
plot(w,efn(1:500))
grid on
title('Espectro señal salida sistema por cancelación')
subplot(3,1,3)

```

```

scf=abs(fft(s));
plot(w,scf(1:500))
grid on
title('Espectro señal original')
B1=fir1(50,[0.23 0.25],'stop');
ecfn=filter(B1,[1],x);
figure
subplot(2,1,1)
plot(ef)
title('señal salida sistema adaptado')
subplot(2,1,2)
plot(ecfn)
title('señal Salida notch convencional de orden 50')
figure
subplot(2,1,1)
plot(w,efn(1:500))
title('Espectro señal salida notch adap')
subplot(2,1,2)
d=abs(fft(ecfn));
plot(w,d(1:500))
title('señal Salida notch convencional de orden 50')

```

14. Programa simulacionseno.mat

```

%programa que simula el efecto del algoritmo del sistema
%por cancelación utilizando una
%variación senosoidal en su valor frecuencia de la
%interferencia.
n=1:1000;
fs=500;
T=1/500;
for k=1:500
    w(k)=500*(k-1)/1000;
end
R=0.05*sin(2*pi*0.5*n*T)+60;
for n=1:1000
    v1(n)=0.4*sin(2*pi*R(n)*n*T);
    v1n(n)=sin(2*pi*60*n*T);
end
v1f=abs(fft(v1));
v1nf=abs(fft(v1n));
figure
subplot(2,1,1)
plot(w,v1f(1:500))
title('Señal seno de frecuencia variable (modelo
senosoidal)')
subplot(2,1,2)
plot(w,v1nf(1:500))
title('Señal seno de frecuencia fija')

```

```

%señal de muestra:
n=1:1000;
fs=500;
s1=sin(2*pi*10*n*(1/fs));
s2=sin(2*pi*50*n*(1/fs));
s3=sin(2*pi*58*n*(1/fs));
s4=sin(2*pi*65*n*(1/fs));
s5=sin(2*pi*70*n*(1/fs));
s=s1+s2+s3+s4+s5;
%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*fs;
x=s+v1;
%Calculo de coeficientes iniciales
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a=a1;
a0=1;
a2=1;
%función de transferencia del notch inicial
N1=[a0 a1 a2];
figure
freqz(N1,[1])
title('Respuesta en frecuencia del notch inicial')
figure
subplot(2,1,1)
plot(s)
title('señal de entrada')
subplot(2,1,2)
plot(x)
title('señal de contaminada')
%estimación de la amplitud de la interferencia de 60 Hz
c=0;
%cálculo de la parte real del coeficiente de la señal x[n]
for j=1:1000
    real=c+x(j)*cos(2*pi*120*j/1000);
    c=real;
end
%cálculo de la parte imaginaria del coeficiente de la señal
x[n]
d=0;
for q=1:1000
    img=d+x(q)*sin(2*pi*120*q/1000);
    d=img;
end
fase1=atan(img/real);
fase=(pi/2)-fase1;
A=(sqrt(real*real+img*img))*2/1000;

R1=0.05*sin(2*pi*0.2*n*T)+60;

```

```

for n=1:1000
    v2(n)=A*sin(2*pi*R1(n)*n*T);
end
% Proceso de adaptación
%G(Z)=go[0 -a1 -1] Función de Transferencia del Estimador
%z=Señal esimada
V(1)=a1;
    G=[0 -a1 -1];
    z=(filter(G,[1],v2));
    e=x-z;
    c=a1;
    for i=2:1000
        q=c+0.000001*e(i)*x(i-1);
        c=q ;
        V(i)=c;
    end

alf=c;
    G=[0 -alf -1];
    z=filter(G,[1],v2);
    ef=x-z;
for k=1:500
    w(k)=fs*(k-1)/1000;
end
figure
subplot(2,1,1)
plot(x)
title('señal contaminada')
subplot(2,1,2)
plot(ef)
title('señal Salida')
figure
subplot(3,1,1)
xf=abs(fft(x));
plot(w,xf(1:500))
grid on
title('Espectro señal contaminada')
subplot(3,1,2)
efn=abs(fft(ef));
plot(w,efn(1:500))
grid on
title('Espectro señal salida sistema por cancelación')
subplot(3,1,3)
scf=abs(fft(s));
plot(w,scf(1:500))
grid on
title('Espectro señal original')

B=fir1(50,[0.23 0.25],'stop');
ecfn=filter(B,[1],x);

```

```

figure
subplot(2,1,1)
plot(ef)
title('señal salida sistema adaptado')
subplot(2,1,2)
plot(ecfn)
title('señal Salida notch convencional de orden 50')
figure
subplot(2,1,1)
plot(w,efn(1:500))
title('Espectro señal salida sistema por cancelación')
subplot(2,1,2)
d=abs(fft(ecfn));
plot(w,d(1:500))
title('señal Salida notch convencional de orden 50')

```

15. Programa evalfrecfija.mat

```

%Programa que simula el comportamiento del sistema de
cancelación ante la
%presencia de una interferencia de 60 Hz con frecuencia
fija.
%Se toma la señal de entrada X(n) que corresponde a la
señal ECG tomada de una
% base de datos con frecuencia de muestreo de 333.333 Hz y
se somete a la operación
%de resta junto con la señal V2 procesada a través de la
función estimadora G(Z).
%ec= señal electrocardiográfica (ver archivo original)

%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*333.333;
n=1:1000;
%Interferencia
v1=0.09*sin(2*pi*60*n*(1/333.333)+0.45);
ec=ec';
x=ec+v1;
%Calculo de coeficientes iniciales
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a=a1;
a0=1;
a2=1;
%función de transferencia del notch inicial
N1=[a0 a1 a2];

figure

```

```

freqz(N1,[1])
title('Respuesta en frecuencia del notch inicial')
figure
subplot(2,1,1)
plot(ec)
title('señal de entrada')
subplot(2,1,2)
plot(x)
title('señal de contaminada')
%estimación de la amplitud de la interferencia de 60 Hz
c=0;
%cálculo de la parte real del coeficiente de la señal x[n]
for j=1:1000
    real=c+x(j)*cos(2*pi*180*j/1000);
    c=real;
end
%cálculo de la parte imaginaria del coeficiente de la señal
x[n]
d=0;
for q=1:1000
    img=d+x(q)*sin(2*pi*180*q/1000);
    d=img;
end
fase=atan(img/real);
fase=(pi/2)-fase;
A=(sqrt(real*real+img*img))*2/1000;
%Señal de referencia V2
v2=A*sin(2*pi*60*n*(1/333.333)+fase);
% Proceso de adaptación
%G(Z)=go[0 -a1 -1] Función de Transferencia del Estimador
%z=Señal esimada
V(1)=a1;
G=2*[0 -a1 -1];
z=filter(G,[1],v2);
e=x-z;
c=a1;
for i=2:1000
    q=c+0.00000001*e(i)*x(i-1);
    c=q ;
    V(i)=c;
end

alf=c;
G=[0 -alf -1];
z=filter(G,[1],v2);
ef=x-z;

for k=1:500

```

```

    w(k)=333.333*(k-1)/1000;
end
figure
subplot(2,1,1)
plot(x)
title('señal contaminada')
subplot(2,1,2)
plot(ef)
title('señal Salida')
figure
subplot(3,1,1)
xf=abs(fft(x));
plot(w,xf(1:500))
grid on
title('Espectro señal contaminada')
subplot(3,1,2)
efn=abs(fft(ef));
plot(w,efn(1:500))
grid on
title('Espectro señal salida sistema por cancelación')
subplot(3,1,3)
ecf=abs(fft(ec));
plot(w,ecf(1:500))
grid on
title('Espectro señal original')
B=fir1(50,[0.35 0.37],'stop');
ecfn=filter(B,[1],x);
figure
subplot(2,1,1)
plot(ef)
title('señal salida sistema adaptado')
subplot(2,1,2)
plot(ecfn)
title('señal Salida notch convencional de orden 50')
figure
subplot(2,1,1)
plot(w,efn(1:500))
title('Espectro señal salida notch adap')
subplot(2,1,2)
d=abs(fft(ecfn));
plot(w,d(1:500))
title('señal Salida notch convencional de orden 50')

```

16. Programa evalfrecseno.mat

```
%simulación de contaminación de una señal cardiaca con
%ruido de 60 Hz con frecuencia central que varía según una
%función seno
%ec= señal electrocardiográfica (ver archivo original)
n=1:1000;
T=1/333.333;
c=0;
for i=1:1000
    t(i)=c+T;
    c=t(i);
end
R=0.05*sin(2*pi*0.5*n*T)+60;
plot(t,R)
for n=1:1000
    v1(n)=0.2*sin(2*pi*R(n)*n*T);
    v1n(n)=0.3*sin(2*pi*60*n*T);
end
figure
plot(t,v1)
v1f=abs(fft(v1));
for k=1:500
    w(k)=333.333*(k-1)/1000;
end
figure
plot(w,v1f(1:500))
x=ec'+v1;
%frecuencia a atenuar
w0=2*pi*60;
% frecuencia de muestreo
ws=2*pi*333.333;
%Calculo de coeficientes iniciales
b=(w0/ws)*2*pi;
a1= -2*cos(b);
a=a1;
a0=1;
a2=1;
%función de transferencia del notch inicial
N1=[a0 a1 a2];
figure
freqz(N1,[1])
title('Respuesta en frecuencia del notch inicial')
figure
subplot(2,1,1)
plot(ec)
title('señal de entrada')
subplot(2,1,2)
plot(x)
```



```

title('señal de contaminada')

%estimación de la amplitud de la interferencia de 60 Hz
c=0;
%cálculo de la parte real del coeficiente de la señal x[n]
for j=1:1000
    real=c+x(j)*cos(2*pi*180*j/1000);
    c=real;
end
%cálculo de la parte imaginaria del coeficiente de la señal
x[n]
d=0;
for q=1:1000
    img=d+x(q)*sin(2*pi*180*q/1000);
    d=img;
end
fase1=atan(img/real);
fase=(pi/2)-fase1;
A=(sqrt(real*real+img*img))*2/1000;
%Señal de referencia V2
n=1:1000;
R1=0.005*sin(2*pi*0.5*n*T)+60;
for n=1:1000
    v2(n)=A*sin(2*pi*R1(n)*n*T);
end
% Proceso de adaptación
%G(Z)=go[0 -a1 -1] Función de Transferencia del Estimador
%z=Señal esimada
V(1)=a1;
G=[0 -a1 -1];
z=filter(G,[1],v2);
e=x-z;
c=a1;
for i=2:1000
    q=c+0.00000000001*e(i)*x(i-1);
    c=q ;
    V(i)=c;
end

alf=c;
G=[0 -alf -1];
z=filter(G,[1],v2);
ef=x-z;
for k=1:500
    w(k)=333.333*(k-1)/1000;
end
figure
subplot(2,1,1)
plot(x)
title('señal contaminada')
subplot(2,1,2)

```

```

plot(ef)
title('señal Salida')

figure
subplot(3,1,1)
xf=abs(fft(x));
plot(w,xf(1:500))
grid on
title('Espectro señal contaminada')
subplot(3,1,2)
efn=abs(fft(ef));
plot(w,efn(1:500))
grid on
title('Espectro señal sistema por cancelación')
subplot(3,1,3)
scf=abs(fft(ec));
plot(w,scf(1:500))
grid on
title('Espectro señal original')
B=fir1(50,[0.35 0.37], 'stop');
ecfn=filter(B,[1],x);
figure
subplot(2,1,1)
plot(ef)
title('señal salida sistema por cancelación')
subplot(2,1,2)
plot(ecfn)
title('señal Salida notch convencional de orden 50')
figure
subplot(2,1,1)
plot(w,efn(1:500))
title('Espectro señal salida sistema por cancelación')
subplot(2,1,2)
d=abs(fft(ecfn));
plot(w,d(1:500))
title('señal Salida notch convencional de orden 50')

```

17. Programa deteccionondaR.mat.

```

%Simulación del algoritmo para detección de la onda R
basado en la primera y segunda derivada.
%ec= señal electrocardiográfica (ver archivo original)
B=fir1(60,[0.052 0.148]);
freqz(B,[1])
title('Respuesta en Frecuencia')
y=filter(B,1,ec);
figure
subplot(2,1,1)
plot(ec)

```

```

grid on
title('Señal ecg')
subplot(2,1,2)
plot(y)
grid on
title('Señal Filtrada')
B0=[1 0 -1];
y0=filter(B0,[1],ec);
y0=abs(y0);
B1=[1 -2 0 1];
y1=filter(B1,[1],ec);
y1=abs(y1);
y2=1.3*y0+1.1*y1;
for i=1:1000
    if y2(i)>1
        y3(i)=1;
    else
        y3(i)=0;
    end
end
figure
subplot(4,1,1)
plot(ec)
grid on
title('Señal ecg')
subplot(4,1,2)
plot(y)
grid on
title('Filtro Pasa Banda')
subplot(4,1,3)
plot(y0)
grid on
title('primera derivada')
subplot(4,1,4)
plot(y1)
grid on
title('Segunda derivada')
figure
subplot(3,1,1)
plot(ec)
grid on
title('Señal ecg')
subplot(3,1,2)
plot(y2)
grid on
title('1.3Yo[n]+1.1Y1[n]')
subplot(3,1,3)
plot(y3)
grid on
title('Umbral')

```

18 . Programa detecRumbral.mat

```
%Programa que simula la detección de la onda r por medio
del algoritmo de Hamilton y Tompkins.
%p2= señal electrocardiográfica (ver archivo original)

%filtro pasa banda
[B,A]=fir1(20,[0.1 0.2]);
%Respuesta en frecuencia del filtro pasa banda.
freqz(B,A)
%Filtrado pasa banda
p=filter(B,A,p2);
fs=1/0.008;
for k=1:(N/2)
    w(k)=fs*(k-1)/N;
end
%Análisis espectral.
fec=abs(fft(p2));
fy=abs(fft(p));
figure
subplot(2,1,1)
plot(p2)
title('señal ECG de entrada')
subplot(2,1,2)
plot(p)
title('filtro Pasa banda')
figure
subplot(2,1,1)
plot(w,fec(1:(N/2)))
title('Espectro señal ECG')
subplot(2,1,2)
plot(w,fy(1:(N/2)))
title('Espectro señal ECG filtrada')
%filtro derivador
B1=[ 0.25 0.125 0 -0.125 -0.25 ];
B0=[1 0 -1];
figure
freqz(B0,[1])
p2=filter(B0,[1],p);
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(p2)
title('filtro derivador')
fy=abs(fft(p));
fy2=abs(fft(p2));
```

```

figure
subplot(2,1,1)
plot(w,fy(1:(N/2)))
title('Espectro señal ECG')
subplot(2,1,2)
plot(w,fy2(1:(N/2)))
title('Espectro señal ECG filtrada (Derivador)')
%cuadrado de la señal
p=p2.^2;
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(p)
title('función de cuadrado')
%integrador de ventana movil
B3=[0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ];
a=filter(B3,[1],p);
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(a)
title('Integrador con ventana móvil')
%grid on
%umbral
%picol pico total
%spki pico estimado de la señal
%npki pico estimado del rido
%umbral11 umbral aplicado
%umbral12 segundo umbral aplicado
%valor máximo del arreglo ec
temp=a(1);
for m=2:N
    temp2=a(m);
    if temp < temp2
        pico = temp2;
        temp =pico;
    else
        pico = temp;
    end
end
%normalización arreglo ec
p=a/pico;
%valor pico estimado spki=temp
%npki=temp2;
spki=mean(a);

```

```

npki=0.5;

spki=0.125 + 0.875*spki;
npki=0.125 + 0.875*npki;
umbral11=npki+0.25*(spki-npki);
umbral12=0.5*umbral11;
for m=1:N
    if p(m)>umbral11
        a(m)=1;
    else
        a(m)=0;
    end
end
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(a)
title('Onda R ')
%cáculo tiempos RR
temp=0;
r1=1;
z=0;
for m=1:N
    if a(m)==0
        temp=temp+1;
        z=0;
    else
        z=z+1;
        if z<=1
            r1=r1+1;
            t(r1)=temp;
            temp=0;
            z=z+1;
        else
            z=z+1;
            temp=temp+1;
        end
    end
end
end
L=length(t);
t=t(3:L);
t=0.008*t;
figure
stem(t)

```

19. Programa ondaRlineabase.mat

```
%Programa que simula el efecto de un corrimiento de la
linea de base en el algoritmo
%para detección de la onda R.
%ec = señal electrocardiográfica.
%r= ruido simulado
r=r';
x=ec+(r*10);
inter=r;
subplot(2,1,1)
plot(ec)
title('Señal ECG')
subplot(2,1,2)
plot(x)
title('Señal ECG')
N=1000;
%filtro pasa banda
[B,A]=fir1(20,[0.1 0.2]);
y=filter(B,A,ec);
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(y)
title('filtro Pasa banda')
%filtro derivador
B1=[ 0.25 0.125 0 -0.125 -0.25 ];
B0=[1 0 -1];
y2=filter(B0,[1],y);
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(y2)
title('filtro derivador')
%cuadrado de la señal
y3=y2.^2;
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(y3)
title('función de cuadrado')
%integrador de ventana movil
```

```

B3=[0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ];

y4=filter(B3,[1],y3);
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(y4)
title('Integrador con ventana móvil')
grid on
%umbral
%picol pico total
%spki pico estimado de la señal
%npki pico estimado del ruido
%umbral11 umbral aplicado
%umbral12 segundo umbral aplicado
%valor máximo del arreglo ec
a=y4(1);
for i=2:N
    d=y4(i);
    if a < d
        picol = d;
        a =picol;
    else
        picol = a;
    end
end
%normalización arreglo ec
y4=y4/picol;
%valor pico estimado spki
spki=mean(y4);
npki=0.5;
spki=0.125 + 0.875*spki;
npki=0.125 + 0.875*npki;
umbral11=npki+0.25*(spki-npki);
umbral12=0.5*umbral11;
for i=1:N
    if y4(i)>umbral11
        R(i)=1;
    else
        R(i)=0;
    end
end
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(R)

```



```

title('Onda R ')

%cáculo tiempos RR
r=0;
j=1;
uno=0;
for i=1:N
    if R(i)==0
        r=r+1;
        uno=0;
    else
        uno=uno+1;
        if uno<=1
            j=j+1;
            TT(j)=r;
            r=0;
            uno=uno+1;
        else
            uno=uno+1;
            r=r+1;
        end
    end
end
end
L=length(TT);
TT=TT(3:L);
TT= 0.002*TT;
temp=0;
for i=1:L-2
    t(i)=TT(i)+temp;
    temp=t(i);
end
figure
stem(TT)
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
stem(R)
title('Onda R ')
%con señal con línea de base.
%filtro pasa banda
ec=x;
[B,A]=fir1(20,[0.1 0.2]);
y=filter(B,A,ec);
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)

```

```

plot(y)
title('filtro Pasa banda')
%filtro derivador
B1=[ 0.25 0.125 0 -0.125 -0.25 ];
B0=[1 0 -1];
y2=filter(B0,[1],y);
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(y2)
title('filtro derivador')
%cuadrado de la señal
y3=y2.^2;
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(y3)
title('función de cuadrado')
%integrador de ventana movil
B3=[0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ];
y4=filter(B3,[1],y3);
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(y4)
title('Integrador con ventana móvil')
%grid on
%umbral
%picol pico total
%spki pico estimado de la señal
%npki pico estimado del rido
%umbral11 umbral aplicado
%umbral12 segundo umbral aplicado
%valor máximo del arreglo ec
a=y4(1);
for i=2:N
    d=y4(i);
    if a < d
        picol = d;
        a =picol;
    else
        picol = a;
    end
end
end
%normalización arreglo ec

```

```

y4=y4/picol;

%valor pico estimado spki
spki=mean(y4);
npki=0.5;
spki=0.125 + 0.875*spki;
npki=0.125 + 0.875*npki;
umbral11=npki+0.25*(spki-npki);
umbral12=0.5*umbral11;
for i=1:N
    if y4(i)>umbral11
        R(i)=1;
    else
        R(i)=0;
    end
end
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)
plot(R)
title('Onda R ')
%cáculo tiempos RR
r=0;
j=1;
uno=0;
for i=1:N

    if R(i)==0
        r=r+1;
        uno=0;
    else
        uno=uno+1;
        if uno<=1
            j=j+1;
            TT(j)=r;
            r=0;
            uno=uno+1;
        else
            uno=uno+1;
            r=r+1;
        end
    end
end
end
TT= 0.002*TT;
figure
subplot(2,1,1)
plot(ec)
title('señal ECG de entrada')
subplot(2,1,2)

```

```

stem(TT)
title('Intervalos calculados ')
20. Programa dominiotiempo.mat

```

```

%simulación donde se calculan los índices mencionados sobre
un arreglo de datos que contienen tiempos RR tomados de una
base de datos
%RR= tiempos RR . (ver archivo original)
N=length(RR);%promedio tiempos RR
prom=0;
for i=1:N
    prom=prom+RR(i);
end
prom=prom/N;
%Desviación estandar Tiempos RR
des=0;
for i=1:N
    des=des + ((RR(i)-prom)*(RR(i)-prom)/(N-1));
end
des=sqrt(des);
%diferencias sucesivas de NN
for i=1:N-1
    dNN(i)=RR(i+1)-RR(i);
end
%RMSSD: RMS de las diferencias sucesivas de NN
RMSSD=0;
for i=1:N-1
    RMSSD=RMSSD+dNN(i)*dNN(i);
end
RMSSD=sqrt((1/(N-1))*RMSSD);
%NN50: Número de diferencias de intervalos NN sucesivos
mayores que 50ms.
for i=1:N-1
    if dNN(i)>50
        d(i)=1;
    else
        d(i)=0;
    end
end
r=0;
j=0;
for k=1:N-1
    if d(k)==1
        r=r+1;
    else
        if r>1
            j=j+1;
            CNN(j)=r;
        end
        r=0;
    end
end

```

```

        end

    end
    if r>1
        j=j+1;
        CNN(j)=r;
    end
    NN50=0;
    for i=1:j
        NN50=NN50+CNN(i);
    end
    pNN50=(NN50/N)*100;
    plot(RR/1000,'o:')
    title('Tiempos RR')
    xlabel('muestra')
    ylabel('Tiempo (s)')
    RMSSD
    NN50
    pNN50

```

21. Programa PSDtiemposRR.mat

```

N=length(RR1);
%Arreglo que contiene la base de tiempo
c=0;
for i =1:N
    t(i)=c+RR1(i);
    c=t(i);
end
%Arreglo que contiene los valores en frecuencia
%con los que se evalúa la fórmula de lomb.
w=0.001:0.004:0.4;
N=length(RR1);
N1=length(w);

    %Proceso de enventanado.
    xm=0;
    s=0;
    for i=1:N
        xm= xm + RR1(i);
        W2= 0.54+0.46*cos(2*pi*t(i)/t(N));%Ventana de hanning.
        RR1(i)=RR1(i)*W2;
        s=s+W2;
    end

xm=xm/N;
NOR=N/(s*s);

```

```

for r=1:N1
    c=0;
    b=0;
    Sh=0;
    Ch=0;
    S2=0;
    C2=0;
    for i=1:N
        Sh=Sh+( (RR1(i)-xm)*sin( 2*pi*w(r)*(t(i)) ) );
        Ch=Ch+( (RR1(i)-xm)*cos(2*pi*w(r)*(t(i))) );
        S2=S2+sin(2*pi*w(r)*(t(i)));
        C2=C2+cos(2*pi*w(r)*(t(i))) ;
    end
    tao=(atan(S2/C2))/(4*pi*w(r));
    A=Ch*cos(2*pi*w(r)*tao)+Sh*sin(2*pi*w(r)*tao);
    B=Sh*cos(2*pi*w(r)*tao)-Ch*sin(2*pi*w(r)*tao);
    A2=(N/2)+0.5*C2*cos(4*pi*w(r)*tao)+0.5*S2*sin(4*pi*w(r)*tao);
    B2=(N/2)-0.5*C2*cos(4*pi*w(r)*tao)-0.5*S2*sin(4*pi*w(r)*tao);
    Pr(r)=NOR*((A*A)/A2)+((B*B)/B2));
end
Prlog=10*log10(Pr);
plot(w,Prlog)
figure
plot(w,Prlog)
title('Análisis en el dominio de la frecuencia')
xlabel('frecuencia (Hz)')
ylabel('Magnitud')
mini=min(RR1);
Tp=mini/2;
TN=max(t);
Tp=1/100;
ts=0:Tp:TN;
RI=interp1(t,RR1,ts,'splines');
figure
plot(ts,RI)
NR=length(RI);
Prom=sum(RI)/NR;
RIp=RI-Prom;
V=hamming(NR);
RIv=V.*RIp;
RIF=abs(fft(RIv));
F=1/Tp;
NR=NR-1;
for i=1:(NR/2)
    w1(i)=(i-1)*F/NR;
end
plot(w1,RIF(1:(NR/2)))

```

```

title('Análisis en el Dom. Frecuencia(FFT)')
xlabel('Frecuencia (Hz)')
ylabel('magnitud')

```

22. Programa dominofrecuencia.mat

```

N=length(RR1);
%Arreglo que contiene la base de tiempo
c=0;
for i =1:N
    t(i)=c+RR1(i);
    c=t(i);
end
%Arreglo que contiene los valores en frecuencia
%con los que se evalúa la fórmula de lomb.
w=0.001:0.004:0.4;
N=length(RR1);
N1=length(w);
    %Proceso de enventanado.
    xm=0;
    s=0;
for i=1:N
    xm= xm + RR1(i);
    W2= 0.54+0.46*cos(2*pi*t(i)/t(N));%Ventana de hanning.
    RR1(i)=RR1(i)*W2;
    s=s+W2;
end
xm=xm/N;
NOR=N/(s*s);
for r=1:N1
    c=0;
    b=0;
    Sh=0;
    Ch=0;
    S2=0;
    C2=0;
    for i=1:N
        Sh=Sh+( (RR1(i)-xm)*sin( 2*pi*w(r)*(t(i)) ) );
        Ch=Ch+( (RR1(i)-xm)*cos(2*pi*w(r)*(t(i)))));
        S2=S2+sin(2*pi*w(r)*(t(i)));
        C2=C2+cos(2*pi*w(r)*(t(i))) ;
    end
    tao=(atan(S2/C2))/(4*pi*w(r));
    A=Ch*cos(2*pi*w(r)*tao)+Sh*sin(2*pi*w(r)*tao);
    B=Sh*cos(2*pi*w(r)*tao)-Ch*sin(2*pi*w(r)*tao);

```

```

A2=(N/2)+0.5*C2*cos(4*pi*w(r)*tao)+0.5*S2*sin(4*pi*w(r)*tao
);
B2=(N/2)-0.5*C2*cos(4*pi*w(r)*tao)-
0.5*S2*sin(4*pi*w(r)*tao);
Pr(r)=NOR*((A*A)/A2)+((B*B)/B2));
end
Prlog=10*log10(Pr);
plot(w,Prlog)
figure
plot(w,Prlog)
title('Análisis en el dominio de la frecuencia')
xlabel('frecuencia (Hz)')
ylabel('Magnitud')
dw=0.004;
%cálculo VLF (<0.04)
A=0;
B=0;
for i=1:4
    A=A+Pr(2*i);
    B=B+Pr(2*i+1);
end
VLF=(dw/3)*(Pr(1)+4*(A+Pr(10))+2*B+Pr(11));

%cálculo LF (0.04-0.15)
A=0;
B=0;
for i=1:14
    A=A+Pr(2*i+10);
    B=B+Pr(2*i+1+10);
end
LF=(dw/3)*(Pr(11)+4*(A+Pr(40))+2*B+Pr(42));

%calculo de la HF(0.15 - 0.4)
A=0;
B=0;
for i=1:28
    A=A+Pr(2*i+42);
    B=B+Pr(2*i+1+42);
end
HF=(dw/3)*(Pr(41)+4*(A+Pr(100))+2*B+Pr(100));
%calculo potencia total
A=0;
B=0;
for i=1:49
    A=A+Pr(2*i);
    B=B+Pr(2*i+1);
end
PT=(dw/3)*(Pr(1)+4*A+2*B+Pr(100));
LFnu=LF*100/(PT-VLF);
HFnu=HF*100/(PT-VLF);
disp('Indices del análisis en el dominio de la frecuencia')

```



```
VLF  
disp('')
```

```
LF  
disp('')  
HF  
disp('')  
LFnu  
disp('')  
HFnu  
disp('')
```

APENDICE I I

1. Programa decimación.c

```
#include<std.h>
#include<log.h>           //Soporte para monitorizar el DSK
#include<math.h>
#include<tms320.h>        //Soporte para librerías de la familia //tms320
#include<type.h>          //Soporte para definir nuevas variables
#include<board.h> //Soporte para inicializar la tarjeta
#include<codec.h>
#include <swi.h> //Archivo soporte para manejo del //codec
#include<mcbbsp54.h> //Manejo de los puertos seriales McBSP
#include<dsplib.h>
HANDLE hHandset; //Directiva para manejo del codec
s16 data; //Definición de data como variable tipo //s16
int m=0,z=0,r,r1,rt,N;
float a[8000]={0};
float p[2000]={0};
float p2[2000]={0};
void delay(int);
void main()
{
    hHandset = codec_open(HANDSET_CODEC); //Habilitar el codec de audio
    codec_dac_mode(hHandset, CODEC_DAC_15BIT); //Configuración del codec
    de audio
    codec_adc_mode(hHandset, CODEC_ADC_15BIT);
    codec_sample_rate(hHandset,SR_2000); /* 2KHz frecuencia de muestreo*/
    for(m=0;m<8000;m++)
    {
        //Espera la llegada de una muestra
        while (!MCBSP_RRDY(HANDSET_CODEC)) {};
        //Coloca la muestra en el buffer de salida
        data = *(volatile u16*)DRR1_ADDR(HANDSET_CODEC);
        // coloca muestra en el codec
        a[m]=data;
        a[m]=((a[m]-1086.691515)*0.1/2174.01)+0.110;
    }
    //Filtrado pasa bajos a frecuencia de corte pi/4
    z=0;
    for (r=0;r<4;r++){
        for (m=0;m<6;m++){
            p[m]=a[m+z];}

        for (m=6;m<2000;m++){
            p[m]=0.00848592006158*a[m+z]+0.06975525008428*a[m-
            1+z]+0.24503094177825*a[m-2+z]+0.35345577615177*a[m-
            3+z]+0.24503094177825*a[m-4+z]+0.06975525008428*a[m-
            5+z]+0.00848592006158*a[m-6+z];
        }
    }
}
```

```

for (m=0;m<2000;m++){
a[m+z]=p[m];}
z=z+2000;
}
z=0;
//Decimación por factor 4
for(m=0;m<2000;m++){
z=4*m+1;
p2[m]=a[z];
}
}

```

2. Programa filtrolineadabase.c

```

#include<std.h>
#include<log.h>           //Soporte para monitorizar el DSK
#include<math.h>
#include<tms320.h>        //Soporte para librerías de la familia //tms320
#include<type.h>          //Soporte para definir nuevas variables
#include<board.h> //Soporte para inicializar la tarjeta
#include<codec.h>
#include <swi.h> //Archivo soporte para manejo del //codec
#include<mcbbsp54.h> //Manejo de los puertos seriales McBSP
#include<dsplib.h>
#include "time.h"
HANDLE hHandset;//Directiva para manejo del codec
s16 data; //Definición de data como variable tipo //s16
int m=0,z=0,r,r1;
float a[8000]={0};
float p[2000]={0};
float p2[2000]={0};
float tb[11]={0,0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2};
float c;
float d[6]={0};
float poli[6]={0};
float INV[6][6]={
{0.97377622378 , -7.21459790211, 17.40056818185,-18.22006118884,
8.60467657343,-1.50240384615},
{-7.21459790211, 142.94843871357,-478.75919652972, 595.93607347033,-
313.39916593522, 58.88588408056},
{17.40056818183, -478.75919652964, 1764.58280216482,-
2315.77911018811, 1260.30246878353,-242.57562099045},
{-18.22006118881, 595.93607347023,-2315.77911018810,
3138.40212518050,-1746.40219803934, 341.69254139480},
{8.60467657342, -313.39916593517, 1260.30246878359,-
1746.40219803942, 987.61237675873,-195.62550079843},
{-1.50240384615, 58.88588408055,-242.57562099048, 341.69254139483,-
195.62550079844, 39.12510015967},
};

```

```
float vma[11]={-0.03381196,-0.02255912,-0.001520712, 0.00981942,
0.01339368, 0.03524481, 0.02928216, 0.03983014, 0.03836351, 0.01064724,
0.01363879};
```

```
void main()
{
//Habilitar el codec de audio
hHandset = codec_open(HANDSET_CODEC);
//Configuración del codec de audio
codec_dac_mode(hHandset, CODEC_DAC_15BIT);
codec_adc_mode(hHandset, CODEC_ADC_15BIT);
codec_sample_rate(hHandset,SR_2000); * 2KHz frecuencia de muestreo*/
```

```
for(m=0;m<8000;m++)
{
//Espera la llegada de una muestra
while (!MCBSP_RRDY(HANDSET_CODEC)) {};
//Coloca la muestra en el buffer de salida
data = *(volatile u16*)DRR1_ADDR(HANDSET_CODEC);
// coloma muestra en el codec
a[m]=data;
a[m]*=0.150/7626;
}
//Filtrado pasa bajos a frecuencia de corte pi/4
a[0]=a[1];
a[1]=a[2];
a[2]=a[3];
for (m=3;m<8000;m++){
a[m]=0.06295639775563*a[m]+0.87408720448874*a[m-
1]+0.06295639775563*a[m-2];
}
z=0;
//Decimación por factor 2
for(m=0;m<2000;m++){
z=4*m+1;
p[m]=a[z];
}
//pormedio
z=0;
a[0]=p[0];
for (r=1;r<=10;r++){
c=0;
for(r1=0;r1<200;r1++){
c+=p[r1+z]; }
a[r]=c/200;
z=z+200;
}
}
```

```

for (r=0;r<6;r++) {
    c=0;
    for(r1=0;r1<11;r1++) {
        c +=(a[r1]*pow(tb[r1],r));
    }
    a[r+11]=c;
}

c=0;
for (r=0;r<11;r++) {
    c+=a[r];
}
a[11]=c;
//multiplicación
for (r=0;r<6;r++){
    c=0;
    for (r1=0;r1<6;r1++) {
        c +=(INV[r1][r]*a[r1+11]);
    }
    a[r]=c;
}

for (r=0;r<2000;r++){
    c=0;
    for (r1=0;r1<6;r1++) {
        c +=a[r1]*pow(t[r],r1);
    }
    p2[r]=c;
}

for (r=0;r<2000;r++){
    t[r]=p[r]-p2[r];
}

}

```

3. Programa notch.c

```
#include<std.h>
#include<log.h>           //Soporte para monitorizar el DSK
#include<math.h>
#include<tms320.h>        //Soporte para librerías de la familia //tms320
#include<type.h>          //Soporte para definir nuevas variables
#include<board.h> //Soporte para inicializar la tarjeta
#include<codec.h>
#include <swi.h> //Archivo soporte para manejo del //codec
#include<mcbbsp54.h> //Manejo de los puertos seriales McBSP
#include<dsplib.h>
#include "time.h"
HANDLE hHandset;//Directiva para manejo del codec
s16 data; //Definición de data como variable tipo //s16
float a[8000];
int m=0,z=0,r,r1;
//float a[8000]={0};
float p[2000]={0};
float p2[2000]={0};
float tb[11]={0,0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2};
float c,u,f;
float d[6]={0};
float fase,A;
float pi=3.14159265358979;
float INV[6][6]={
    {0.97377622378 , -7.21459790211, 17.40056818185, -18.22006118884,
    8.60467657343, -1.50240384615},
    {-7.21459790211, 142.94843871357, -478.75919652972, 595.93607347033,
    -313.39916593522, 58.88588408056},
    {17.40056818183, -478.75919652964, 1764.58280216482,-
    2315.77911018811, 1260.30246878353,-242.57562099045},
    {-18.22006118881, 595.93607347023,-2315.77911018810,
    3138.40212518050,-1746.40219803934, 341.69254139480},
    {8.60467657342, -313.39916593517, 1260.30246878359,-
    1746.40219803942, 987.61237675873,-195.62550079843},
    {-1.50240384615, 58.88588408055, -242.57562099048, 341.69254139483,
    -195.62550079844, 39.12510015967},
    };

void main()
{
    //borrar variables.
    //Habilitar el codec de audio
    hHandset = codec_open(HANDSET_CODEC);
    //Configuración del codec de audio
    codec_dac_mode(hHandset, CODEC_DAC_15BIT);
    codec_adc_mode(hHandset, CODEC_ADC_15BIT);
    codec_sample_rate(hHandset,SR_2000);
    /* 2KHz frecuencia de muestreo*/
```

```

for(m=0;m<8000;m++){
//Espera la llegada de una muestra
while (!MCBSP_RRDY(HANDSET_CODEC)) {};
//Coloca la muestra en el buffer de salida
data = *(volatile u16*)DRR1_ADDR(HANDSET_CODEC);
// coloca muestra en el codec
a[m]=data;
a[m]=((a[m]-1086.691515)*0.1/2174.01)+0.110;
}

//Filtrado pasa bajos a frecuencia de corte pi/4
z=0;
for (r=0;r<4;r++){
for (m=0;m<6;m++){
p[m]=a[m+z];
}

for (m=6;m<2000;m++){
p[m]=0.00848592006158*a[m+z]+0.06975525008428*a[m-
1+z]+0.24503094177825*a[m-2+z]+0.35345577615177*a[m-
3+z]+0.24503094177825*a[m-4+z]+0.06975525008428*a[m-
5+z]+0.00848592006158*a[m-6+z];
}

for (m=0;m<2000;m++){
a[m+z]=p[m];
}
z=z+2000;
}

z=0;
//Decimación por factor 4
for(m=0;m<2000;m++){
z=4*m+1;
p2[m]=a[z];
}

//Filtrado pasa bajos corte 100Hz.
p[0]=p2[0];
p[1]=p2[1];
p[2]=p2[2];
p[3]=p2[3];
p[4]=p2[4];
for (m=5;m<2000;m++){
p[m]=0.01974533929363*p2[m]+ 0.13240450223511*p2[m-
1]+0.34785015847126*p2[m-2]+0.34785015847126*p2[m-
3]+0.13240450223511*p2[m-4]+0.01974533929363*p2[m-5];
}

```

```

//pormedio
z=0;
a[0]=p[0];
for (r=1;r<=10;r++){
    c=0;
    for(r1=0;r1<200;r1++){
        c+=p[r1+z];
    }
    a[r]=c/200;
    z=z+200;
}
//arreglo de datos

for (r=0;r<6;r++) {
    c=0;
    for(r1=0;r1<11;r1++) {
        c +=(a[r1]*pow(tb[r1],r));
    }
    a[r+11]=c;
}

c=0;

for (r=0;r<11;r++) {
    c+=a[r];
}
a[11]=c;

//multiplicación y cálculo de la solución
//del sistema de ecuaciones.
//se obtienen los coeficientes del polinomio
for (r=0;r<6;r++)
{
    c=0;
    for (r1=0;r1<6;r1++)
    {
        c +=(INV[r1][r]*a[r1+11]);
    }
    a[r]=c;
}

//Calculo de la función estimada
//de la línea de base utilizando
//los coeficientes encontrados

for (r=0;r<2000;r++)
{
    c=0;
    for (r1=0;r1<6;r1++)

```



```

{
    c += a[r1]*pow(t[r],r1);
}
p2[r]=c;
}
//Resta: señal contaminada - función estimada.
for (r=0;r<2000;r++){
    t[r]=p[r]-p2[r];
}

//Filtro 60Hz.
c=0;
//cálculo de la magnitud de la componente de 60 Hz de la señal.
for (m=0;m<2000;m++){
    c += t[m]*cos(2*pi*240*m/2000);
}
u=0;
for (m=0;m<2000;m++){
    u +=t[m]*sin(2*pi*240*m/2000);
}
fase=u/c;
fase=atan(fase);
A=(sqrt(u*u+c*c))*2/2000;
//v2 estimada.
for(m=0;m<2000;m++){
    f=0.05*sin(2*pi*0.5*m*0.002)+60;
    p2[m]=A*cos((2*pi*f*m*0.002)+fase);
}

//filtro notch para 60 HZ.
//a1: coeficiente que será adaptado.
//b=(Wo/Ws)*2*pi. Wo=frecuencia que se desea atenuar.
//Ws=frecuencia de muestreo.
//a1=-2cos(b)=c;
c=-1.457937254;

a[0]=p2[0];
a[1]=p2[1];
//Función estimadora G(z)
//a[q] toma el valor de la señal estimada.
for (m=2;m<2000;m++){
    a[m]=-c*p2[m-1]-p2[m-2];
}
//calculo de e.
for (m=0;m<2000;m++)
{
    p[m]=t[m]-a[m];
}

```

```

//Proceso de adaptación
a[0]=c;
for(m=2;m<2000;m++){
    c +=0.001*p[m]*t[m-1];
    a[m-1]=c;
}
a[2000]=c;

//Señal de salida
p[0]=t[0];
p[1]=t[1];
p[2]=t[2];
for (m=2;m<2000;m++){
    p[m]=t[m]+c*t[m-1]+t[m-2];
}

}

```

4. Programa deteccionqrs.c

```

#include<std.h>
#include<log.h>           //Soporte para monitorizar el DSK
#include<math.h>
#include<tms320.h>        //Soporte para librerías de la familia //tms320
#include<type.h>          //Soporte para definir nuevas variables
#include<board.h> //Soporte para inicializar la tarjeta
#include<codec.h>
#include <swi.h> //Archivo soporte para manejo del //codec
#include<mcbbsp54.h> //Manejo de los puertos seriales McBSP
#include<dsplib.h>
//#include "time.h"
HANDLE hHandset;//Directiva para manejo del codec
s16 data; //Definición de data como variable tipo //s16
int m=0,z=0,r,r1,c;
float a[8000]={0};
float p[2000]={0};
float p2[2000]={0};
float t[500]={0};
float ecg[2000]={0};

float temp,temp2,pico;
void main()
{

```

```

//Habilitar el codec de audio
hHandset = codec_open(HANDSET_CODEC);
//Configuración del codec de audio
codec_dac_mode(hHandset, CODEC_DAC_15BIT);
codec_adc_mode(hHandset, CODEC_ADC_15BIT);
codec_sample_rate(hHandset, SR_2000);
/* 2KHz frecuencia de muestreo*/

r1=0;
for(m=0;m<500;m++){
t[m]=0;
}
for(c=0;c<21;c++){
for(m=0;m<8000;m++){
//Espera la llegada de una muestra
while (!MCBSP_RRDY(HANDSET_CODEC)) {};
//Coloca la muestra en el buffer de salida
data = *(volatile u16*)DRR1_ADDR(HANDSET_CODEC);
// coloca muestra en el codec
a[m]=-data;
//a[m]=-((a[m]-1086.691515)*0.1/2174.01)-0.410;
}

//Filtrado pasa bajos a frecuencia de corte pi/4
z=0;
for (r=0;r<4;r++){
for (m=0;m<6;m++){
p[m]=a[m+z];
}
for (m=6;m<2000;m++){
p[m]=0.00848592006158*a[m+z]+0.06975525008428*a[m-
1+z]+0.24503094177825*a[m-2+z]+0.35345577615177*a[m-
3+z]+0.24503094177825*a[m-4+z]+0.06975525008428*a[m-
5+z]+0.00848592006158*a[m-6+z];
}

for (m=0;m<2000;m++){
a[m+z]=p[m];
}
z=z+2000;
}

z=0;
//Decimación por factor 4
for(m=0;m<2000;m++){
z=4*m+1;
p2[m]=a[z];
ecg[m]=a[z];
}

```

```

//Filtrado pasa banda

for(m=0;m<20;m++){
p[m]=p2[m];
}

for (m=21;m<2000;m++){
p[m]= 0.0*p2[m]-0.00643641562731*p2[m-1]-0.02034406261393*p2[m-2]-
0.04271590624320*p2[m-3]-0.06429410758026*p2[m-4]-
0.06805036637528*p2[m-5]-0.03903518661005*p2[m-
6]+0.02417597652525*p2[m-7]+0.10439338618338*p2[m-
8]+0.17169615839554*p2[m-9]+0.19795049173953*p2[m-
10]+0.17169615839554*p2[m-11]+0.10439338618338*p2[m-
12]+0.02417597652525*p2[m-13]-0.03903518661005*p2[m-14]-
0.06805036637528*p2[m-15]-0.06429410758026*p2[m-16]-
0.04271590624320*p2[m-17]-0.02034406261393*p2[m-18]-
0.00643641562731*p2[m-19]-0.0*p2[m-20] ;

}

//filtro derivador
p2[0]=p[0];
p2[1]=p[1];
p2[2]=p[2];
p2[3]=p[2];
for (m=4;m<2000;m++){
p2[m]=p[m]+0*p[m-1]-p[m-2];
}

//cuadrado de la señal.
for (m=0;m<2000;m++){
p[m]=p2[m]*p2[m];
}

//integrador de ventana móvil
temp=0;
for (m=0;m<20;m++){
a[m]=p[m];
}
for (m=20;m<2000;m++)
{
    for(z=m-20;z<m;z++)
    {
        temp= temp + 0.05*p[z];
    }
a[m]=temp;
temp = 0;
}

```

```

a[0]=0;
a[1]=0;
//Valor máximo

temp=a[0];
for (m=1;m<2000;m++) {
    temp2=a[m];
    if (temp < temp2){
        pico = temp2;
        temp =pico;
    }
    else
        pico = temp;
}

//normalización del arreglo
//temp=valor medio
temp=0;
temp2=0;
for (m=0;m<2000;m++)
{
    p[m]=a[m]/pico;
    temp = temp + p[m];
    temp2= temp2 + p[m]*p[m];
}

//valor pico estimado spki=temp
//npki=temp2
temp=temp/2000;
temp2=temp2/2000;
temp2=sqrt(temp2);
temp=0.125 + 0.875*temp;
temp2=0.125 + 0.875*temp2;

//cálculo del umbral

temp=temp2+0.25*(temp-temp2);
//umbral12=0.5*umbral11;

//aplicación del umbral
for (m=0;m<2000;m++) {
    if (p[m]>temp){
        a[m]=1;
    }
    else
        a[m]=0;
}

```

//cálculo de los tiempos RR.

```
temp=0;
z=0;
for (m=0;m<2000;m++) {
    p[m]=0;
    if (a[m]==0){
        temp=temp+1;
        z=0;
    }
    else
    {
        z=z+1;
        if (z<=1){

            t[r1]=temp*0.002;
            r1=r1+1;
            temp=0;
            z=z+1;
        }
        else{
            z=z+1;
            temp=temp+1;
        }
    }
}

}

z=0;
for (m=0;m<r1;m++)
{
    if (t[m]>0.3){
        p[z]=t[m];
        z=z+1;
    }
    else{
        z=z+0;
    }
}

}

}
```

5. Programa indicesdominiodel tiempo.c

```
#include<std.h>
#include<log.h>           //Soporte para monitorizar el DSK
#include<math.h>
#include<tms320.h>        //Soporte para librerías de la familia //tms320
#include<type.h>          //Soporte para definir nuevas variables
#include<board.h> //Soporte para inicializar la tarjeta
#include<codec.h>
#include <swi.h> //Archivo soporte para manejo del //codec
#include<mcbbsp54.h>      //Manejo de los puertos seriales McBSP
#include<dsplib.h>
#include "tiempos.h"

int i,r,j;
float prom,des,RMSSD,NN50,pNN50;
float dNN[346];
float CNN[346];
float d[346];

void delay(int);
void main()
{

//promedio tiempos RR
//RR:Datos de entrada.
prom=0;
for (i=0;i<346;i++)
{
    prom=prom+RR[i];
}
prom=prom/346;

//Desviación estandar Tiempos RR
des=0;
for (i=0;i<346;i++)
{
    des=des +((RR[i]-prom)*(RR[i]-prom)/(346-1));
}
des=sqrt(des);
//diferencias sucesivas de NN
for (i=0;i<345;i++)
{
    dNN[i]=RR[i+1]-RR[i];
}
//RMSSD: RMS de las diferencias sucesivas de NN
RMSSD=0;
for (i=0;i<345;i++)
{
```

```

    RMSSD=RMSSD+dNN[i]*dNN[i];
}
RMSSD=RMSSD/(345);
RMSSD=sqrt(RMSSD);
//NN50: Número de diferencias de intervalos NN sucesivos mayores que 50ms.
for(i=0;i<345;i++) {
    if (dNN[i]>50)
        d[i]=1;
    else
        d[i]=0;}

r=0;
j=0;
for (i=0;i<345;i++)
{
    if (d[i]==1)
        r=r+1;
    else
    {
        if (r>1){
            CNN[j]=r;
            j=j+1;
        }
        r=0;
    }
}
if (r>1){
j=j+1;
CNN[j]=r;
}
NN50=0;
for (i=0;i<j;i++) {
    NN50=NN50+CNN[i];
}
pNN50=(NN50/346)*100;

}

```


6. Programa lomb.c

```
#include<std.h>
#include<log.h>           //Soporte para monitorizar el DSK
#include<math.h>
#include<tms320.h>        //Soporte para librerías de la familia //tms320
#include<type.h>          //Soporte para definir nuevas variables
#include<board.h> //Soporte para inicializar la tarjeta
#include<codec.h>
#include <swi.h> //Archivo soporte para manejo del //codec
#include<mcbbsp54.h> //Manejo de los puertos seriales McBSP
#include<dsplib.h>
//#include "tiempo.h"
//#include "muestras.h"
//#include "frecuen.h"
#include "TRR2.h"
int i,r;
float xm,s,NOR;
float c,b,Sh,Ch,S2,C2;
float tao,A,B,A2,B2;
float Pr[212],t[305],w[212];
float arg;
float W,dw;
float pi=3.14159265358979;
void main()
{

dw=0.004;
  c=0;
  for(i=0;i<36;i++){
    t[i]=c+RR1[i];
    c=t[i];
  }
  c=0.001;
  w[0]=c;
  for(i=1;i<100;i++){
    w[i]=c+dw;
    c=w[i];
  }

  xm=0;
  s=0;
  for(i=0;i<36;i++){
    xm= xm + RR1[i];
    arg=(2*pi*t[i])/t[35];
    W= 0.54+0.46*cos(arg);
    RR1[i]=RR1[i]*W;
    s=s+W;
  }
  xm=xm/36;
```

```

NOR=36/(s*s);
for (r=0;r<100;r++){
    c=0;
    b=0;
    Sh=0;
    Ch=0;
    S2=0;
    C2=0;
    for (i=0;i<36;i++)
    {
        Sh=Sh+( (RR1[i]-xm)*sin( 2*pi*w[r]*(t[i]) ) );
        Ch=Ch+( (RR1[i]-xm)*cos( 2*pi*w[r]*(t[i]) ) );
        S2=S2+sin(2*pi*w[r]*(t[i]));
        C2=C2+cos(2*pi*w[r]*(t[i]));
    }
    tao=(atan(S2/C2))/(4*pi*w[r]);
    A=Ch*cos(2*pi*w[r]*tao)+Sh*sin(2*pi*w[r]*tao);
    B=Sh*cos(2*pi*w[r]*tao)-Ch*sin(2*pi*w[r]*tao);
    A2=(36/2)+0.5*C2*cos(4*pi*w[r]*tao)+0.5*S2*sin(4*pi*w[r]*tao);
    B2=(36/2)-0.5*C2*cos(4*pi*w[r]*tao)-0.5*S2*sin(4*pi*w[r]*tao);
    Pr[r]=NOR*( ((A*A)/A2) + ((B*B)/B2) );
}

}

```

7. Programa indicesdominofrecuencia.c

```
#include<std.h>
#include<log.h>           //Soporte para monitorizar el DSK
#include<math.h>
#include<tms320.h>        //Soporte para librerías de la familia //tms320
#include<type.h>          //Soporte para definir nuevas variables
#include<board.h> //Soporte para inicializar la tarjeta
#include<codec.h>
#include <swi.h> //Archivo soporte para manejo del //codec
#include<mcbbsp54.h> //Manejo de los puertos seriales McBSP
#include<dsplib.h>
//#include "tiempo.h"
//#include "muestras.h"
//#include "frecuen.h"
#include "TRR2.h"
int i,r;
float xm,s,NOR;
float c,b,Sh,Ch,S2,C2;
float tao,A,B,A2,B2;
float Pr[212],t[305],w[212];
float arg;
float W,dw;
float pi=3.14159265358979;
float VLF,LF,HF,PT,LFnu,HFnu;
void delay(int);
void main()
{

dw=0.004;
c=0;
for(i=0;i<36;i++){
t[i]=c+RR1[i];
c=t[i];
}
c=0.001;
w[0]=c;
for(i=1;i<100;i++){
w[i]=c+dw;
c=w[i];
}

xm=0;
s=0;
for(i=0;i<36;i++){
xm= xm + RR1[i];
arg=(2*pi*t[i])/t[35];
W= 0.54-0.46*cos(arg);
RR1[i]=RR1[i]*W;
```

```

    s=s+W;
}
xm=xm/36;
NOR=36/(s*s);
for (r=0;r<100;r++){
    c=0;
    b=0;
    Sh=0;
    Ch=0;
    S2=0;
    C2=0;
    for (i=0;i<36;i++)
    {
        Sh=Sh+( (RR1[i]-xm)*sin( 2*pi*w[r]*(t[i]) ) );
        Ch=Ch+( (RR1[i]-xm)*cos( 2*pi*w[r]*(t[i]) ) );
        S2=S2+sin(2*pi*w[r]*(t[i]));
        C2=C2+cos(2*pi*w[r]*(t[i]));
    }
    tao=(atan(S2/C2))/(4*pi*w[r]);

    A=Ch*cos(2*pi*w[r]*tao)+Sh*sin(2*pi*w[r]*tao);
    B=Sh*cos(2*pi*w[r]*tao)-Ch*sin(2*pi*w[r]*tao);
    A2=(36/2)+0.5*C2*cos(4*pi*w[r]*tao)+0.5*S2*sin(4*pi*w[r]*tao);
    B2=(36/2)-0.5*C2*cos(4*pi*w[r]*tao)-0.5*S2*sin(4*pi*w[r]*tao);

    Pr[r]=NOR*( ((A*A)/A2) + ((B*B)/B2) );

}

//Cálculo VLF
A=0;
B=0;
for (r=0;r<4;r++){
    A=A+Pr[2*r+1];
    B=B+Pr[2*r+2];
}
VLF=(dw/3)*(Pr[0]+4*(A+Pr[9])+2*B+Pr[10]);

//Cálculo LF (0.04-0.15)
A=0;
B=0;
for (r=0;r<15;r++){
    A=A+Pr[2*r+10+1];
    B=B+Pr[2*r+2+10];
}
LF=(dw/3)*(Pr[10]+4*(A)+2*B+Pr[41]);

//calculo de la HF(0.15 - 0.4)
A=0;
B=0;

```

```

for (r=0;r<28;r++){
    A=A+Pr[2*r+42+1];
    B=B+Pr[2*r+2+42];
}
HF=(dw/3)*(Pr[41]+4*(A)+2*B+Pr[99]);

//cálculo potencia total
A=0;
B=0;
for (r=0;r<49;r++){
    A=A+Pr[2*r+1];
    B=B+Pr[2*r+2];
}
PT=(dw/3)*(Pr[0]+4*A+2*B+Pr[99]);
LFnu=LF*100/(PT-VLF);
HFnu=HF*100/(PT-VLF);

}

```

8. Programa final

```
#include<std.h>
#include<log.h>           //Soporte para monitorizar el DSK
#include<math.h>
#include<tms320.h>        //Soporte para librerías de la familia //tms320
#include<type.h>          //Soporte para definir nuevas variables
#include<board.h> //Soporte para inicializar la tarjeta
#include<codec.h>
#include <swi.h>           //Archivo soporte para manejo del             //codec
#include<mcbbsp54.h>      //Manejo de los puertos seriales McBSP
#include<dsplib.h>
#include "time.h"
HANDLE hHandset;//Directiva para manejo del codec
s16 data; //Definición de data como variable tipo //s16

int m=0,z=0,r,r1,rt,N;
float a[8000]={0};
float p[2000]={0};
float p2[2000]={0};
float RR[1000]={0};
float tb[11]={0,0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2};
float c,u,f,cont;
float d[6]={0};
float fase,A;
float pi=3.14159265358979;
float INV[6][6]={
    { 0.97377622378, -7.21459790211, 17.40056818185, -18.22006118884,
      8.60467657343, -1.50240384615},
    { -7.21459790211, 142.94843871357, -478.75919652972,
      595.93607347033, -313.39916593522, 58.88588408056},
    { 17.40056818183, -478.75919652964, 1764.58280216482,-
      2315.77911018811, 1260.30246878353,-242.57562099045},
    { -18.22006118881, 595.93607347023,-2315.77911018810,
      3138.40212518050, -1746.40219803934, 341.69254139480},
    { 8.60467657342, -313.39916593517, 1260.30246878359,-
      1746.40219803942, 987.61237675873,-195.62550079843},
    { -1.50240384615, 58.88588408055, -242.57562099048, 341.69254139483,
      -195.62550079844, 39.12510015967},
    };

float prom,des,RMSSD,NN50,pNN50;
float dNN[346];
float CNN[346];
float d1[346];
int i,j;
float dw,Sh,Ch,B2;
float VLF,LF,HF,PT,HFnu,LFnu;
float IND[8];
```

```

float TIME;

void main()
{
//Habilitar el codec de audio
hHandset = codec_open(HANDSET_CODEC);
//Configuración del codec de audio
    codec_dac_mode(hHandset, CODEC_DAC_15BIT);
    codec_adc_mode(hHandset, CODEC_ADC_15BIT);
    codec_sample_rate(hHandset,SR_2000);
    /* 2KHz frecuencia de muestreo*/

    rt=0;
    TIME=0;

    while(TIME<300){

        for(m=0;m<8000;m++)

        {
            //Espera la llegada de una muestra
            while (!MCBSP_RRDY(HANDSET_CODEC)) {};
            //Coloca la muestra en el buffer de salida
            data = *(volatile u16*)DRR1_ADDR(HANDSET_CODEC);
            // coloca muestra en el codec
            a[m]=data;
            a[m]=((a[m]-1086.691515)*0.1/2174.01)+0.110;
        }

        //Filtrado pasa bajos a frecuencia de corte pi/4
        z=0;
        for (r=0;r<4;r++)
        {
            for (m=0;m<6;m++)
            {
                p[m]=a[m+z];
            }
            for (m=6;m<2000;m++)
            {
                p[m]=0.00848592006158*a[m+z]+0.06975525008428*a[m-
                1+z]+0.24503094177825*a[m-2+z]+0.35345577615177*a[m-
                3+z]+0.24503094177825*a[m-4+z]+0.06975525008428*a[m-
                5+z]+0.00848592006158*a[m-6+z];
            }
            for (m=0;m<2000;m++)
            {
                a[m+z]=p[m];
            }
            z=z+2000;
        }
    }
}

```

```

}

z=0;
//Decimación por factor 4
for(m=0;m<2000;m++)
{
z=4*m+1;
p2[m]=a[z];
}

//Filtrado pasa bajos corte 100Hz.

for(m=0;m<4;m++)
{
p[m]=p2[m];
}
z=0;
r=200;
c=0;
r1=0;

a[0]=p[0];
for (m=3;m<2000;m++)
{
p[m]= 0.01809893300751*p2[m]+0.05429679902254*p2[m-
1]+0.05429679902254*p2[m-2]+0.01809893300751*p2[m-
3]+1.76004188034317 *p[m-1]-1.18289326203783*p[m-
2]+0.27805991763455*p[m-3];
c+=p[m-3];
z=z+1;
if(z==r){
a[r1+1]=c/200;
//promedio
c=0;
r1=r1+1;
r=r+200;
}
else{
c=c;
}
}

a[r1+1]=c/200;

```



```
//arreglo de datos
```

```
for (r=0;r<6;r++)
{
    c=0;
    for(r1=0;r1<11;r1++)
    {
        c +=(a[r1]*pow(tb[r1],r));
    }
    a[r+11]=c;
}
```

```
c=0;
```

```
for(r=0;r<11;r++)
{
    c+=a[r];
}
a[11]=c;
```

```
//multiplicación y cálculo de la solución
//del sistema de ecuaciones.
```

```
//se obtienen los coeficientes del polinomio
```

```
for (r=0;r<6;r++){
c=0;
    for (r1=0;r1<6;r1++)
    {
        c +=(INV[r1][r]*a[r1+11]);
    }
    a[r]=c;
}
```

```
//Calculo de la función estimada de la línea de base utilizando
```

```
//los coeficientes encontrados
```

```
c=0;
```

```
u=0;
```

```
for (r=0;r<2000;r++)
```

```
{
    c=0;
    for (r1=0;r1<6;r1++)
    {
        c +=a[r1]*pow(t[r],r1);
    }
}
```

```
p2[r]=c;
```

```
t[r]=p[r]-p2[r]; //Resta: señal contaminada - función estimada.
```

```
c += t[r]*cos(2*pi*240*r/2000); //cálculo de la magnitud de la componente de
60 Hz de la señal.
```

```
u += t[r]*sin(2*pi*240*r/2000);
```

```
}
```

```
//Filtro 60Hz.
```

```
fase=u/c;  
fase=atan(fase);  
A=(sqrt(u*u+c*c))*2/2000;
```

```
//filtro notch para 60 HZ.  
//a1: coeficiente que será adaptado.  
//b=(Wo/Ws)*2*pi. Wo=frecuencia que se desea atenuar.  
//Ws=frecuencia de muestreo.  
//a1=-2cos(b)=c;  
c=-1.457937254;  
a[0]=p2[0];  
a[1]=p2[1];  
for (m=2;m<2000;m++)  
{  
f=0.05*sin(2*pi*0.5*m*0.002)+60;  
p2[m]=A*cos((2*pi*f*m*0.002)+fase);//v2 estimada  
a[m]=-c*p2[m-1]-p2[m-2];//Función estimadora G(z)  
p[m-2]=t[m-2]-a[m-2];//calculo de e.  
c +=0.001*p[m]*t[m-1];//Proceso de adaptación  
}
```

```
//Señal de salida  
p[0]=t[0];  
p[1]=t[1];  
p[2]=t[2];  
for (m=3;m<2000;m++)  
{  
p[m]=t[m]+c*t[m-1]+t[m-2];  
}
```

```
//Detección QRS
```

```
//Filtrado pasa banda  
for(m=0;m<20;m++){  
p2[m]=p[m];  
}  
for (m=21;m<2000;m++){  
p2[m]= 0.0*p[m]-0.00643641562731*p[m-1]-0.02034406261393*p[m-2]-  
0.04271590624320*p[m-3]-0.06429410758026*p[m-4]-0.06805036637528*p[m-  
5]-0.03903518661005*p[m-6]+0.02417597652525*p[m-  
7]+0.10439338618338*p[m-8]+0.17169615839554*p[m-  
9]+0.19795049173953*p[m-10]+0.17169615839554*p[m-  
11]+0.10439338618338*p[m-12]+0.02417597652525*p[m-13]-  
0.03903518661005*p[m-14]-0.06805036637528*p[m-15]-
```

```

0.06429410758026*p[m-16]-0.04271590624320*p[m-17]-
0.02034406261393*p[m-18]-0.00643641562731*p[m-19]-0.0*p[m-20] ;

}

```

```

//filtro derivador
p[0]=p2[0];
p[1]=p2[1];
p[2]=p2[2];
p[3]=p2[3];
for (m=4;m<2000;m++){
p[m]=p2[m]+0*p2[m-1]-p2[m-2];//filtro derivador
}
for (m=0;m<2000;m++){
t[m]=p[m]*p[m];//cuadrado de la señal.
}
//integrador de ventana móvil
c=0;
for (m=0;m<20;m++){
a[m]=t[m];
}

for (m=20;m<2000;m++){
    for(z=m-20;z<m;z++)    {
        c += 0.05*t[z];
    }
a[m]=c;
c = 0;
}

```

```

//Valor máximo
c=a[0];
for (m=1;m<2000;m++) {
    u=a[m];
    if (c < u){
        f = u;
        c =f;
    }
    else
        f = c;
}

```

```

//normalización del arreglo
//temp=valor medio
c=0;
u=0;
for (m=0;m<2000;m++)
{

```

```

p[m]=a[m]/f;
c = c + p[m];
u = u + p[m]*p[m];
}

//valor pico estimado spki=temp
//npki=temp2
c=c/2000;
u=u/2000;
u=sqrt(u);
c=0.125 + 0.875*c;
u=0.125 + 0.875*c;
//cálculo del umbral
c=u+0.25*(c-u);
//umbral12=0.5*umbral11;
//aplicación del umbral

for (m=0;m<2000;m++)
{
    if (p[m]>c){
        a[m]=1;
    }
    else
        a[m]=0;
}

//cálculo de los tiempos RR.
r=0;
z=0;
for (m=0;m<2000;m++) {
    if (a[m]==0){
        r=r+1;
        z=0;
    }
    else
    {
        z=z+1;
        if (z<=1){
            t[r1]=r*0.002;
            r1=r1+1;
            r=0;
            z=z+1;
        }
        else{
            z=z+1;
            r=r+1;
        }
    }
}
}

```

```

for (m=0;m<r1;m++) {
  if (t[m]>0.3){
    RR[rt]=t[m];
    TIME=TIME+RR[rt];
    rt=rt+1;
  }
  else{
    rt=rt+0;
  }
}

}

//promedio tiempos RR
prom=0;
c=0;
f=0;
for (i=0;i<rt;i++){
  {
    prom=prom+RR[i];
    t[i]=c+RR[i];
    c=t[i];
  }
}
prom=prom/rt;
//Desviación estandar Tiempos RR
des=0;
for (i=0;i<rt;i++){
  des=des +((RR[i]-prom)*(RR[i]-prom)/(rt-1));
}
des=sqrt(des);
//diferencias sucesivas de NN
for (i=0;i<rt;i++) {
  dNN[i]=RR[i+1]-RR[i];
}
//RMSSD: RMS de las diferencias sucesivas de NN
RMSSD=0;
for (i=0;i<rt;i++) {
  RMSSD=RMSSD+dNN[i]*dNN[i];
}
RMSSD=RMSSD/(rt);
RMSSD=sqrt(RMSSD);
IND[0]=RMSSD;
//NN50: Número de diferencias de intervalos NN sucesivos mayores que 50ms.
for(i=0;i<rt;i++) {
  if (dNN[i]>50)
    d1[i]=1;
  else

```

```

    d1[i]=0;
}

r=0;
j=0;
for (i=0;i<rt;i++)
{
    if (d1[i]==1)
        r=r+1;
    else
    {
        if (r>1){
            CNN[j]=r;
            j=j+1;
        }
        r=0;
    }
}

if (r>1){
j=j+1;
CNN[j]=r;
}

NN50=0;
for (i=0;i<j;i++) {
    NN50=NN50+CNN[i];
}

pNN50=(NN50/rt)*100;
IND[1]=NN50;
IND[2]=pNN50;

c=0.001;
p[0]=0.001;
for(i=1;i<100;i++){
    p[i]=c+0.004;
    c=p[i];
}

for (i=0;i<rt;i++)
{
    fase=(2*pi*t[i])/t[rt-1];
    cont= 0.54+(0.46*cos(fase));//cálculo del valor de la ventana
    p2[i]=RR[i]*cont;//enventanado de la muestra
    f=f+cont;
}
cont=rt/(f*f);

```

```

for (r=0;r<100;r++){
    Sh=0;
    Ch=0;
    dw=0;
    c=0;
    for (i=0;i<rt;i++){
        {
            Sh=Sh+( (p2[i]-prom)*sin( 2*pi*p[r]*(t[i]) ) );
            Ch=Ch+( (p2[i]-prom)*cos( 2*pi*p[r]*(t[i]) ) );
            c=c+sin(2*pi*p[r]*(t[i]));
            dw=dw+cos(2*pi*p[r]*(t[i]));
        }
        fase=(atan(c/dw))/(4*pi*p[r]);

        A=Ch*cos(2*pi*p[r]*fase)+Sh*sin(2*pi*p[r]*fase);
        u=Sh*cos(2*pi*p[r]*fase)-Ch*sin(2*pi*p[r]*fase);
        f=(rt/2)+0.5*dw*cos(4*pi*p[r]*fase)+0.5*c*sin(4*pi*p[r]*fase);
        B2=(rt/2)-0.5*dw*cos(4*pi*p[r]*fase)-0.5*c*sin(4*pi*p[r]*fase);

        a[r]=cont*( ((A*A)/f) + ((u*u)/B2) );
    }
}

```

```

//Cálculo VLF
A=0;
u=0;
for (r=0;r<4;r++){
    A=A+a[2*r+1];
    u=u+a[2*r+2];
}
VLF=(0.004/3)*(a[0]+4*(A+a[9])+2*u+a[10]);
IND[3]=VLF;
//Cálculo LF (0.04-0.15)
A=0;
u=0;
for (r=0;r<15;r++){
    A=A+a[2*r+10+1];
    u=u+a[2*r+2+10];
}
LF=(0.004/3)*(a[10]+4*(A)+2*u+a[41]);
IND[4]=LF;
//calcula de la HF(0.15 - 0.4)
A=0;
u=0;
for (r=0;r<28;r++){
    A=A+a[2*r+42+1];
    u=u+a[2*r+2+42];
}
HF=(0.004/3)*(a[41]+4*(A)+2*u+a[99]);

```

```

IND[5]=HF;
//cálculo potencia total
A=0;
u=0;
for (r=0;r<49;r++){
    A=A+a[2*r+1];
    u=u+a[2*r+2];
}
PT=(0.004/3)*(a[0]+4*A+2*u+a[99]);

```

```

LFnu=LF*100/(PT-VLF);

```

```

HFnu=HF*100/(PT-VLF);

```

```

IND[6]=LFnu;

```

```

IND[7]=HFnu;

```

```

}

```