UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSI

FACULTAD DE CIENCIAS

Introducción al Labview con Modelación

TESIS PROFESIONAL

para obtener el título de

INGENIERO ELECTRONICO

PRESENTA:

José Reynaldo Hidalgo Colorado

ASESOR DE TESIS:

Dr. Raul Eduardo Balderas Navarro

SAN LUIS POTOSI, S. L. P. FEBRERO 2004

Agradecimientos

Hace poco más de cuatro años que mi vida ha ido cambiando por completo. Jamás había tenido la sensación, ni la más profunda intención de vivir solo. Quizás si hubiese sabido a todas las carencias que me enfrentaría y los problemas que me acogerían, jamás lo hubiera intentado. No obstante, con el paso del tiempo se aprende que la felicidad es gratis y fiel compañera de todos los días.

Quiero Agradecer a mis padres, Reynaldo y Josefina, por el apoyo que me brindaron, por la formación, por fomentar en mí el deseo de saber, de conocer lo novedoso y abrirme las puertas al mundo ante mi curiosidad insaciable. A mi hermano Oliver Josué Por tus comentarios, sugerencias y opiniones. Además de ser un buen amigo eres la mejor compañía para compartir el mismo techo

Deseo agradecer profundamente a la casualidad que la vida me otorgó al haberme puesto en un hogar maravilloso al nacer, el cual recuerdo aun de manera nostálgica. Sin el apoyo en todo sentido de mis padres y hermano, el placer cotidiano de vivir sería simple monotonía. Es difícil imaginar cómo sería el andar cotidiano sin recordar su comprensión, su apoyo inmenso y su amor. Gracias a mis padres, hermano y abuelo por compartir y dedicar gran parte de sus vidas conmigo y por darme aliento para la ardua tarea de caminar hacia la perspectiva de un nuevo día; de verdad serán inolvidables.

Los sabios consejos de Dr. Raul Balderas , asesor de Tesis, quien ha venido guiando desde hace un año mi formación no solamente académica, sino como persona, sin lugar a duda me han dado lugar a ver en la Naturaleza esa combinación de complejidad y sencillez que a la vez se presenta. De gran aprendizaje resultó para mi que la realización de esta Tesis no haya tenido resultados inmediatos; muy por el contrario, en ocasiones no encontraba la llave mágica que abre las puertas hacia el camino de las soluciones Quiero enfatizar mi agradecimiento hacia Raul Balderas por tener la paciencia ante mis dudas de novato y por escuchar atentamente los problemas que a lo largo de esta Tesis surgieron. Gracias también a Juan José Escalante y Rafael Rocha, quiénes propusieron varios métodos sofisticados y a la vez artísticos en la parte computacional de este trabajo.

Sin lugar a duda este trabajo no pudo haberse realizado sin la formación que recibí durante cuatro años en la Facultad de Ciencias (U.A.S.L.P.). Gracias a todos mis compañeros y amigos que contribuyeron realmente en mi formación, en especial a Marco A. Gallegos, Román Díaz de León e Iris Maldonado por todos sus consejos, sus formidables clases, su paciencia y su amistad a mi persona. Quiero agradecer también a Mario A. Martínez, Javier Briones, Tomas Reyna, Azael Bautista, Rosario Juárez. José Luis Castillo, Manuel Elizondo , Juan Luis Juárez ,Vicente Ferrer por su apoyo, sus consejos y compartir su amistad y tiempo en momentos especiales a lo largo de la carrera. Agradezco profundamente a la Facultad de Ciencias y todo su personal por las facilidades brindadas para la realización de esta Tesis. Gracias intensamente a mi Padre y a mis tíos Iván Hidalgo y Omar Hidalgo por haberme introducido al fascinante mundo de la electrónica desde pequeño.

Gracias a la amistad brindada, las sugerencias y sabios consejos y contribuciones a lo largo de mi vida a mis tíos: Cesar Hidalgo, Isabel Colorado Juan Manuel Hidalgo pues han resultado de gran utilidad. Quiero dar las gracias a la gente que revisó con paciencia este trabajo, mis sinodales: Raul Balderas , Dra. Marcela Mejia , M.C. Gustavo Pérez, y a el Director de la Facultad Fis. Benito Pineda Reyes Por su apoyo a lo largo de toda la carrera que incondicionalmente brindo

Será difícil olvidar los maravillosos momentos de los partidos de fútbol representando a mi facultad así como a la selección de fútbol de la U.AS.L.P los entrenamientos a las dos de la tarde. Gracias a el Prof. Antonio Loria del Regil por llevarme a dos torneos nacionales, así como a mis compañeros de la selección que en tantos momentos y viajes defendimos la playera y dejamos en alto el nombre de la U.A.S.L.P. .

Agradezco a mis primos Roberto, Juan Manuel, Cesar por su sincera amistad y por compartir conmigo muy agradables instantes. Como olvidar la sonrisa, el entusiasmo y el apoyo de mi abuelo Reynaldo Hidalgo Torres (que en paz descanse), presente siempre en mi pensamiento. Gracias también en todo sentido a Raúl Macias por su inmenso apoyo en momentos difíciles y por su grandiosa y verdadera amistad.

A mis amigas, que siempre están, estuvieron y seguirán estando, brindandome cariño y soporte. Akane Berrones, Alma Beatriz Díaz de León, Ivett Rasillo, Zuhey Puente, Monica Martínez, Sara Edith Roldan. Cuatro grandes amigos Arturo Rubio, Israel Martínez y Roberto Hernández, Jorge Díaz aunque lejos por el momento apoyaron moralmente este trabajo. No olvidaré los momentos que compartieron conmigo Samuel Acuña, Reymundo Rodríguez.

Gracias a Nahum Sifuentes Sígala por su intachable carácter, por su apoyo y por su amistad. Del mismo modo a Eyder Rodríguez, Ulises Martínez, Onesimo Castillo, Felipe López Sánchez por esas amplias charlas y por compartir una gran amistad. Igualmente gracias a Claudia Amaro, por ser una amiga, socia y colaboradora. A toda mi familia por su amistad y su inmenso apoyo. Agradezco a las cucas, que a pesar de tantos disturbios, la amistad y el apoyo mutuo no han cesado.

Quiero agradecer muy especialmente a Beatriz, mi querida chaparra, que durante un año tuvo la paciencia suficiente para apoyarme profundamente, para darme su comprensión, su cariño y su amor. Gracias por hacer de esos momentos un verdadero vivir.

Por último quiero dar las gracias a todos aquellos que me han devuelto una sonrisa, a todos aquellos que me ofrecieron un consejo en tiempos difíciles, a todos aquellos que han puesto de su parte para que el trajín diario sea más llevadero y muy en especial a la vida que, como dijera Violeta Parra, me ha dado tanto,....

"Hasta la victoria siempre "

Malcom Che Hidalgo

Resumen

Desde la introducción del concepto de instrumentación virtual (IV) en la década de los 80's y su impacto en diferentes disciplinas industriales y de investigación científica, se ha tenido la necesidad de incluirla en la currícula de las Carreras de Ingeniería Eléctrica y Electrónica así como de ciencias básicas. Esto, desafortunadamente, no ha sido el caso para todas las universidades en México. Esto se complica más aún cuando el estudiante requiera un entrenamiento práctico en procesamiento digital de señales aunque cuente con una sólida formación teórica previa. Además la falta de un esquema coherente impide la transición al aspecto práctico. Una herramienta de IV que ha sido la más importante a nivel comercial es LabView de National Instruments, que además de ser relativamente fácil de trabajar, se ha diseminado en casi todas las áreas de ingeniería, principalmente para el control y manejo de instrumentos.

La motivación del presente trabajo de Tesis es presentar un monograma que introduzca rápidamente a los interesados, que cuenten con una base sólida en teoría de filtros digitales y procesamiento digital de señales, a adquirir una experiencia en el manejo del Labview para que la combinen con habilidades de teoría de filtros digitales y procesamiento digital de señales y programación en MatLab.

Índice

Capítulo Uno

INTRODUCCIÓN

El concepto de instrumentación virtual (IV) se origina a partir del uso de la computadora personal (PC) como instrumento de medición de variables físicas (temperatura, presión, vibración, video, audio, imágenes, etc.), representadas por señales analógicas de corriente o voltaje eléctricos, entre otras. Un particular, este concepto va más allá de la simple medición de corriente o voltaje, pues también involucra el procesamiento, análisis, almacenamiento, distribución y despliegue de datos e información relacionada con la medición de una o varias señales específicas. El instrumento virtual utilizado para la adquisición de las señales comprende también la interfaz hombre-máquina, las funciones de análisis y procesamiento de señales, las rutinas de almacenamiento de datos y la comunicación rápida y eficiente con otros equipos. Es importante mencionar que la IV debe estar abierta a la adaptación a las nuevas tecnologías (Bluetooth, WiFi, 802.15, etc.)

El término de "virtual" surge de la funcionalidad y apariencia que –*por software*- el usuario puede definir para la PC utilizada como instrumento, creando una flexibilidad que no depende del fabricante como en el caso de los instrumentos tradicionales. El IV queda definido entonces como el conjunto de software y hardware que agregado a una PC permite a los usuarios interactuar con la computadora como si se estuviera utilizando un instrumento electrónico hecho a la medida. El software es la esencia del sistema de instrumentación virtual –como el hardware lo es para un instrumento tradicional-, ya que es el que le confiere la flexibilidad necesaria para variadas aplicaciones y modos de operación (termómetros, control de funcionamiento de motores, análisis de señales filtrado, etc...) seleccionadas por el programa. El IV se completa con la tarjeta electrónica (hardware) apropiada para la captura conversión y acondicionamiento de las señales eléctricas representativas de las variables físicas en el proceso a estudiar y controlar con algoritmos matemáticos.

El objetivo del presente trabajo es mostrar las ventajas proporcionadas por herramientas tan poderosas de instrumentación virtual LabVIEW para el diseño y el análisis en procesamiento digital de señales, así como en filtros digitales. Al final del trabajo se dicta una sesión en donde se usa LabView, en paralelo con Matlab, para la comprensión detallada en el diseño de control básico proporcional-integral-derivativo y en el diseño de un filtro con respuesta al impulso infinito con fase linealizada, entre otras aplicaciones. Se espera que con este monograma se asista a los alumnos de la facultad a las diferentes programas con los que cuenta en el laboratorio de simulación de la Facultad de Ciencias. La justificación de lo anteriormente dicho se puede resumir como sigue.

En las últimas tres décadas de la ingeniería se ha visto revolucionada en el campo de procesamiento digital de señales, donde los filtros forman una componente muy importante, así como el análisis basado en la transformada de Fourier; ya que estos se encuentran en una extensa gama de aplicaciones. Esencialmente, un filtro es "una caja negra" la cual puede modificar las componentes espectrales de una señal n-dimensional. En este Trabajo se les pretende llevar desde la adquisición o generación de señales filtrado a la presentación final un ejemplo de procesamiento de audio, donde se verá la importancia del Procesamiento digital paso a paso incluye como:

1) Generación de señales digitales y analógicas de ancho de banda limitada.
2) Procesamiento de señales, tanto generadas como las adquiridas por el puerto serial o paralelo y tarjetas PCI.
3) Filtrado de señales.
4) Basados en los puntos 1)- 4), un monograma de prácticas para la mejor comprensión de Lab-View, como finalidad de que el estudiante pueda tomarlo como un tutorial rápido y conciso.

Materiales de Hardware y métodos utilizados.

Hardware: Se utiliza una PC, del laboratorio de Instrumentación Virtual. Incluye ésta los puertos tradicionales: paralelo, serial y tarjeta de adquisición PCI de National Instruments

Señales: Se utilizan diferentes tipos de señales creadas/adquiridas por la PC, simulando diversos tipos de ellas.

Software: ***LabVIEW*** (Laboratory Virtual Instrument Engineering Workbench), uno de los más difundidos para aplicaciones de instrumentación virtual, que está basado en el lenguaje grafico de programación G para facilitar la interacción sistema-usuario y orientado particularmente a la adquisición, procesamiento y manejo de datos. Se emplea también el uso de procesamiento de señales virtuales y además de correr algunos ejemplos diseñados con Matlab (para el diseño y que permite ver el desempeño de funciones matemáticas, así como graficar y encontrar soluciones a sistemas lineales y no lineales) sobre Labview. Este software puede reemplazar la utilización de programas específicos de adquisición de datos usualmente provistos con la tarjeta de conversión A/D (National Instruments). Para la generación de algunos de los filtros digitales se utilizan también algunos programas en MATLAB.

Lenguaje G

Lab View.-

La particularidad del lenguaje G consiste en la posibilidad de programar gráficamente a partir de la asociación de objetos IV, reduciendo considerablemente los tiempos de desarrollo para lograr la adquisición, control, análisis y presentación de datos eficientemente. En la configuración de un IV virtual se distinguen tres partes componentes, como se ilustra en la Fig. 1.

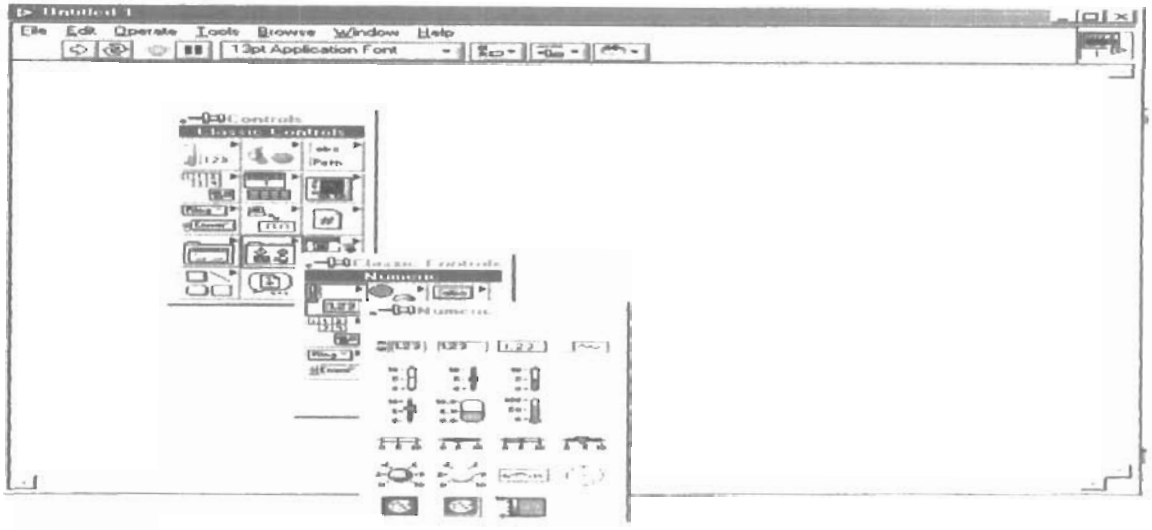


Figura 1.1 Panel frontal o interfaz interactiva de usuario, que simula el panel de un instrumento físico

El panel frontal puede tener perillas, botones de comando y otros controles que son las entradas del usuario. También se puede simular indicadores que constituyen la salida del programa. Los datos se pueden ingresar por mouse o teclado, visualizándose en pantalla los resultados del programa o proceso (Figs 2a y b).

Figura 1.2 a Código gráfico o diagrama en bloques

El diagrama es el código fuente del instrumento virtual, y se construye utilizando el lenguaje G de programación gráfica. Es este lenguaje de programación el que permite que una representación pictórica de los bloques del instrumento sea en realidad el programa que va a ejecutarse.

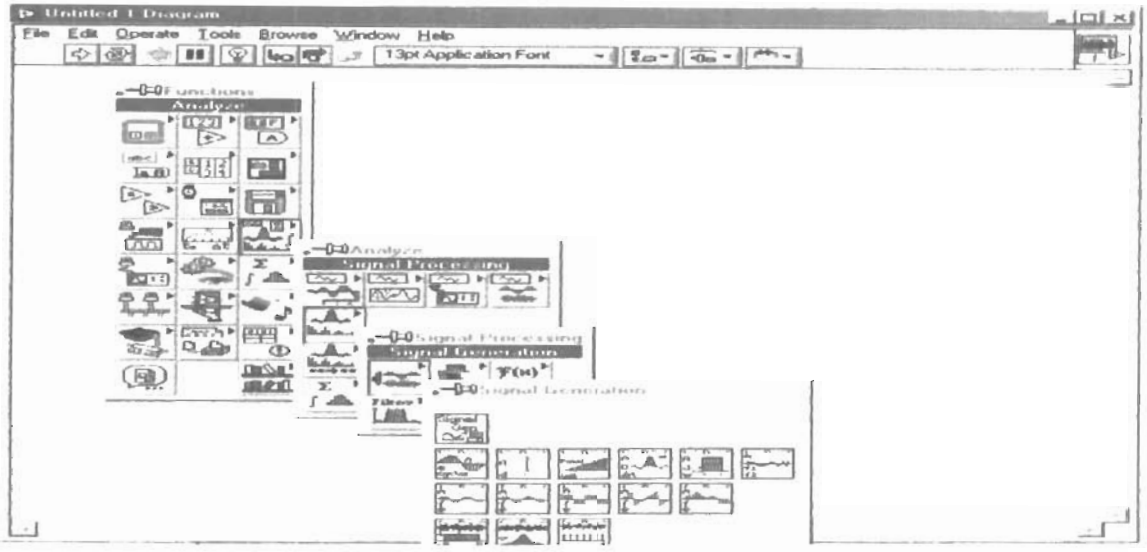


Figura 1.2b. Iconos y conectores son los componentes del código grafico

Los iconos representan módulos VI funciones o estructuras de control del programa. Los conectores indican el flujo de entrada y salida de los datos del programa en bloques. La naturaleza jerárquica y modular de los iconos posibilita su utilización como sub-módulos VI, permitiendo la elaboración de sub-rutinas y la programación modular de la figura 1.

Capítulo Dos

Generación de Señales

En este Capitulo discutiremos la generación de señales utilizando el sub VI ***signal generation by duration*** ubicado en *Funciones Analize signal procesing signal generation*. Además de hacer algunas operaciones con ellas, se muestra su relación con el Teorema de Nyquist.

2.1 Generación de Señales analógicas y digitales

El subprograma *signal generation by duration* es un sub-progrma establecido en el entorno Labview que nos permite generar diferentes tipos de señales acondicionando los controles adecuados para la generación de las diferentes ondas, como los son tipo el de onda deseado, su frecuencia, su amplitud, el número de muestras; así como un offset arbitrario (DC). Con este subprograma analizaremos los siguientes aspectos:

- Generación de señales
- Simulación de señales probando algoritmos cuando no se dispone de una tarjeta de adquisición
- Operaciones de suma, resta y multiplicación entre señales
- Generacion arbitraria de señales f(t) con un periodo determinado.

En el dominio continuo la frecuencia de una señal x(t) es medida en Hz o ciclos por segundo y se denota por Ω. Para su correspondiente señal digital, x(n), obtenida partir de muestrear x(t) ($x(nT) = x(t=n/f_s)$) en tiempos discretos a una razón de muestro $f_s(=1/T_s)$, la frecuencia se denota por ω. Sean $X_a(j\Omega)$ y $X(j\omega)$ las transformadas de Fourier de x(y) y x(n), dadas por:

$$X_a(j\Omega) = \int_{-\infty}^{\infty} x(t)e^{-j\Omega t}dt \quad \text{y} \quad X(j\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \quad . \qquad (2.1)$$

Se puede demostrar que la relación entre X(jΩ) y X(jω) es:

$$X(j\omega) = \frac{1}{T_S}\sum_{m=-\infty}^{\infty} X_a\left[j\left(\frac{\omega}{T_S} - \frac{2\pi}{T_S}m\right)\right]. \qquad (2.2)$$

Las frecuencias analógicas y digitales están relacionadas como $\omega = \Omega T_S$. Esta frecuencia digital, ω, es conocida como frecuencia normalizada. Con unidades de ciclos /muestras. Algunos programas de generación de señales tienen en la entrada el control de la frecuencia, f, eso es contemplando que usamos la frecuencia normalizada, con unidades de ciclos sobre muestras. Esta frecuencia tiene un rango de 0.0 a1.0, el cual corresponde a la frecuencia real en un rango de 0 a la frecuencia de muestreo f_s. Esta frecuencia esta alrededor de 1.0; así que la frecuencia normalizada de 1.1 es equivalente a 0.1. Como ejemplo, una señal que es muestreada con el teorema de Nyquis (fs/2) señal muestreada al doble de la frecuencia. Por cada ciclo, (esto es dos ejemplos/ciclo). Esto corresponde a la frecuencia normalizada de ½ ciclo /muestra =0.5 ciclos/muestra. El reciproco de la frecuencia normalizada, 1/f , nos dá el número de veces que la señal a sido muestreada en un ciclo. Cuando usemos algún subprograma que requiera la frecuencia normalizada de una señal de entrada, se tendrá que convertir nuestras unidades de frecuencia para normalizar las unidades de ciclos/muestras. Nosotros deberemos normalizar las unidades de señal generadas de: Onda senoidal, cuadrada, diente de sierra, triangular y aleatoria Si trabajamos la frecuencia en unidades de ciclos, podemos convertir los ciclos a ciclos /ejemplos por dividiendo los ciclos en el número de ejemplos generados. Solo necesitamos dividir la frecuencia (en ciclos) por el numero de muestras: por ejemplo una frecuencia de 2 ciclos es dividido por 50 ejemplos, el resultado de la frecuencia normalizada es de f=1/25 ciclos/ejemplos

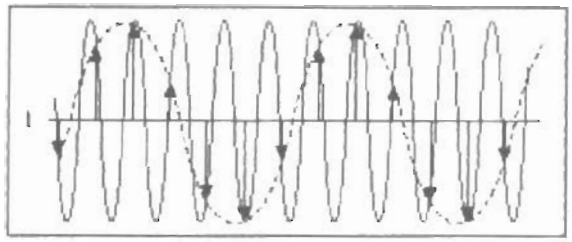


Fig. 2.1 linea puntcada muestra 4 ciclos y 12 muestras, 3 por cada ciclo

2.2 Herramientas para la generación de señales

SIGNAL GENERATION BY DURATION: generación de señales por duración, genera una señal dada por el tipo de onda. Sus controles, numero de ejemplos de la señal de salida, amplitud , fase, frecuencia la cual de inicio es 10 Hz; offset , duración y por supuesto el tipo de onda.

BUILT-in ARRAY : junta varios arreglos dependiendo de el número de elementos de las dimensiones del arreglo

WAVEFORM GRAPH: conecta datos de salida directamente para graficarlos.

CASE STRUCTURE: se usa para mas de un sub-diagrama, funciona como cualquier case en lenguajes de programación; seleccionando la opción indicada dependerá del caso de instrucciones que obedezca.

SUBSTRACT : calcula la diferencia de las entradas

ADD: calcula la suma de las entradas

MULT: genera el producto de las entradas

2.3 Generación de dos señales

La generación de dos señales es el inicio a todo el desarrollo de procesamiento digital de señales. En esta ocasión simularemos en un programa diferentes tipos de señales en una misma gráfica, cuando no tenemos a la mano una tarjeta de adquisición, como habiamos mencionado con anterioridad. Los subprogramas son de mucha ayuda para la comprensión de este tema y en particular tomaremos **Signal generation by duration** que se ubica en el sub directorio **Funciones>Analize> signal procesing> signal generation**. Una vez obtenido el subprograma generamos los controles adecuados para una buena observación,

en este caso ponemos controles en: tipo de onda, numero de muestras a la salida, fase, frecuencia, amplitud y offset. Repetimos la operación para obtener dos diferentes tipos de señales, a la salida de cada una utilizamos un *built array* para generar dos ondas en una misma grafica y a la salida de este utilizamos una grafica de ondas. (Ver Fig. 2.2)

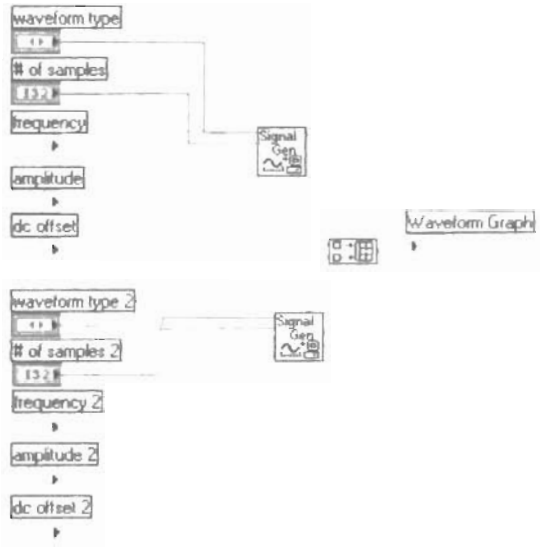


Fig 2.2-a Diagrama de bloques de Generador de señales

Correr el programa en el panel de control para la observación de las diferentes tipos de señales que podemos generar.

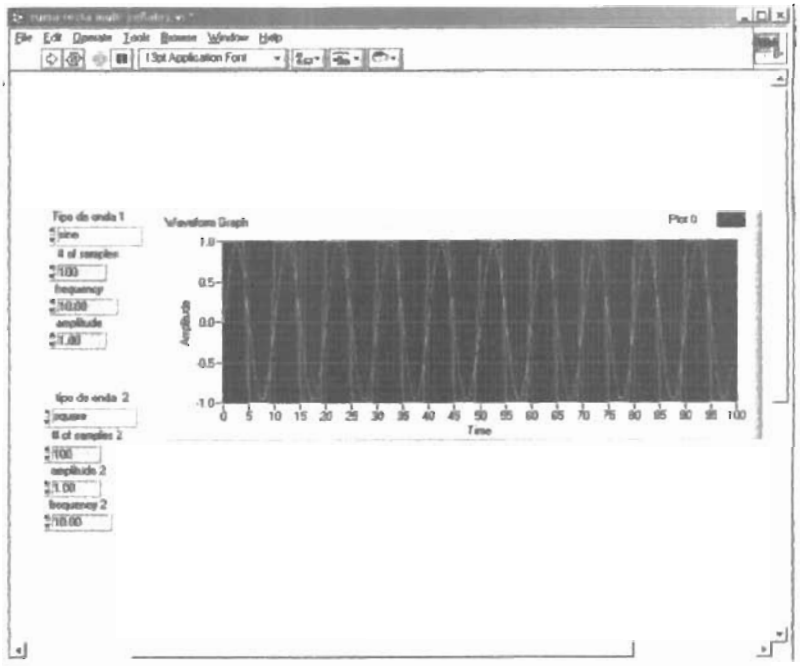


Fig2.2-b Panel de control de Dos señales generadas en un mismo grafica

aquí mostramos dos ondas en una misma gráfica para observar mejor bajaremos la frecuencia de 10 a 1.

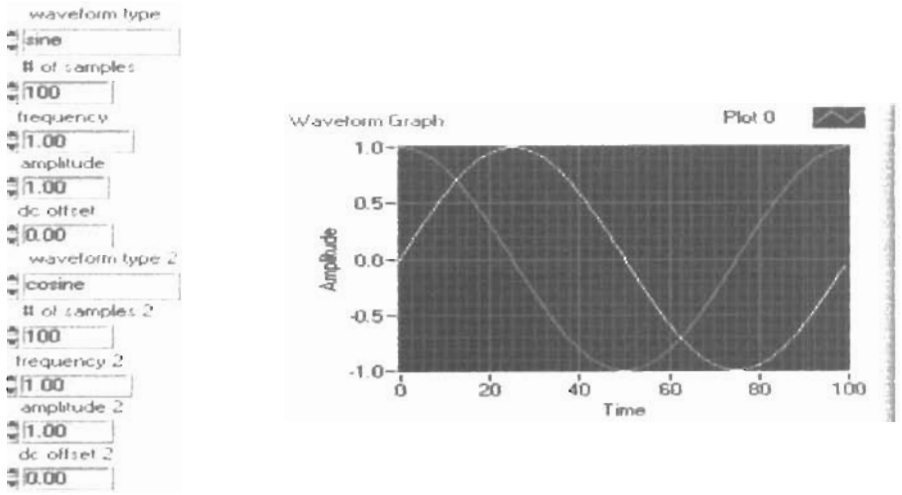


Fig 2.3-a La generación de dos ondas seno y coseno

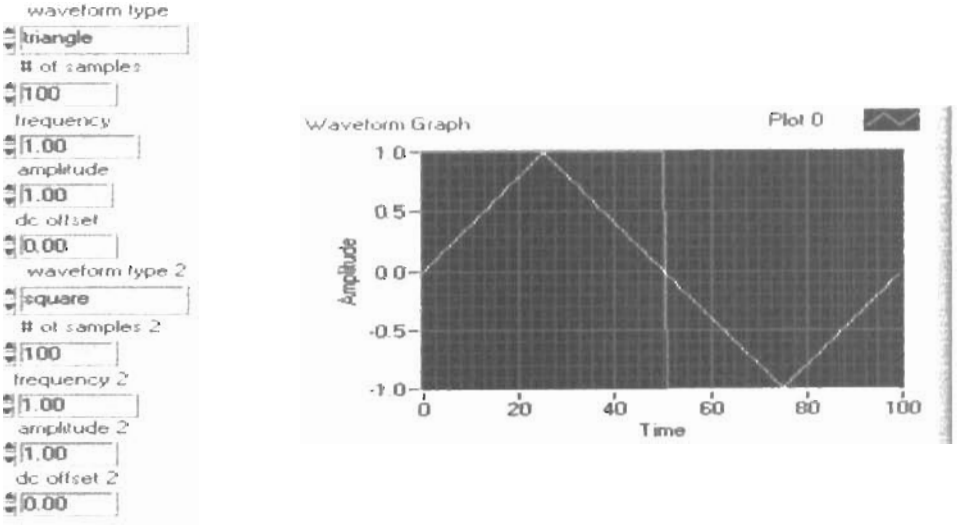


Fig 2.3-b La generación de dos ondas triangular y cuadrada

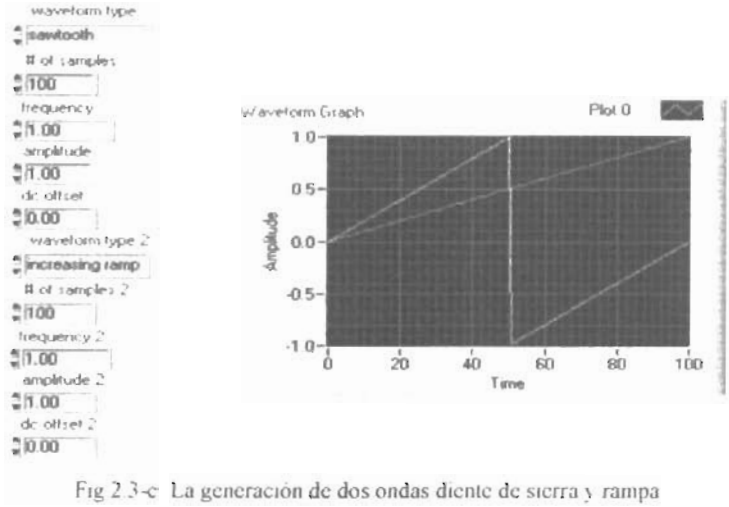


Fig 2.3-c La generación de dos ondas diente de sierra y rampa

2.4 Suma, resta y multiplicación de señales

La manipulación de los diferentes tipos de onda es de importancia así como el desempeño y las operaciones básicas que podamos efectuar con ellas utilizando. El programa anterior será de mucha utilidad ya que solo con algunas modificaciones podremos adecuarlo para hacer operaciones con las ondas que nos aparecen en las graficas. El uso de un subprogama ***case structure***, ubicado en ***functions >structures***, nos permite seleccionar un caso especifico de operación utilizando un control. En este seleccionaremos para tres diferentes tipos de operadores, suma, resta y multiplicación uno para cada caso utilizamos un control en el cual lo encontramos en el ***submenu de controls sing & enum***. Así el programa en el diagrama de bloques es de la siguiente manera (Fig. 2.4):

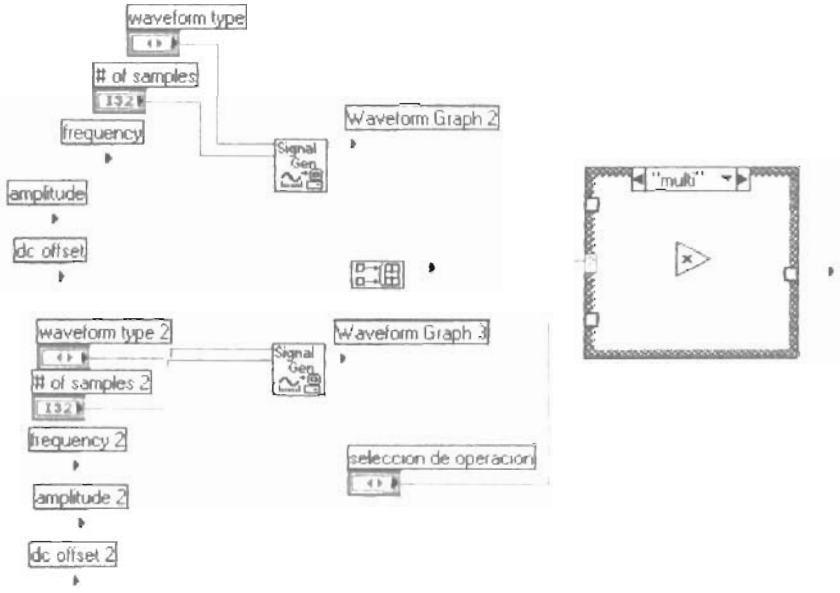


Fig 2.4 Diagrama de bloques del sub VI para suma resta y multiplicación de señales

En este caso notamos que seleccionado la operación de multiplicación, el panel de control nos muestra los diferentes graficas generadas por las operaciones básicas para este caso suma producto diferencia de las diferentes ondas. Utilizaremos para ondas diferentes de frecuencia 1 Hz y amplitud de 1, veremos la suma el producto y la diferencia de ellas.

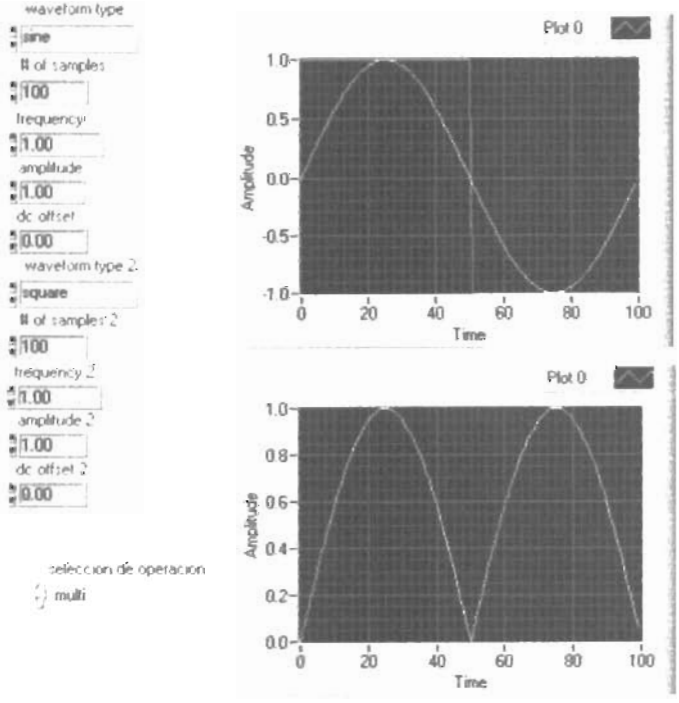
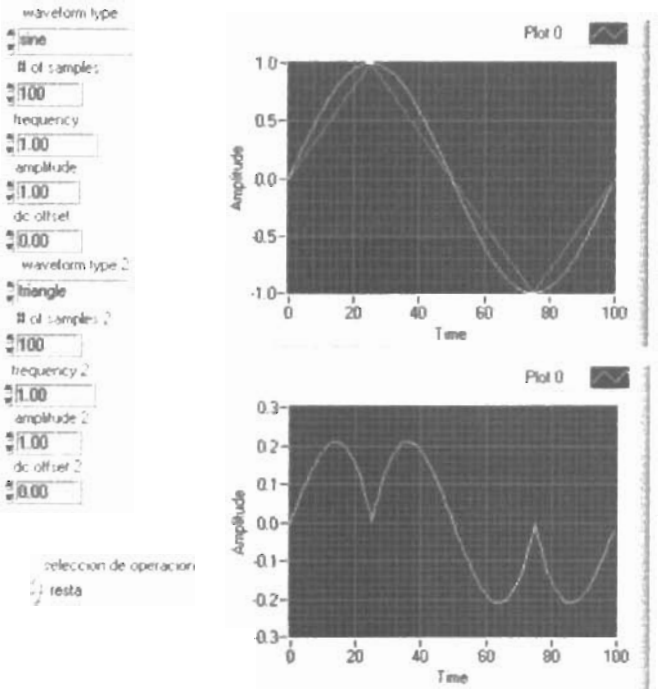


Fig. 2.5 Panel de control en el cual primero mostramos las dos ondas y depuse se multiplican

Ahora bien observaremos en los siguientes puntos:



2.6 Panel de control en el cual se muestra una resta de dos señales

En esta gráfica se muestra la diferencia de dos ondas. En el primer caso tenemos la onda seno y en el segundo la triangular; vemos como la gráfica resultante nos muestra la diferencia de estas dos. Se puede observar que la onda seno casi a su perfección solamente quitándole una sección en forma triangular. En la siguiente un caso de onda seno el cual le sumamos o le restamos la onda cuadrada.

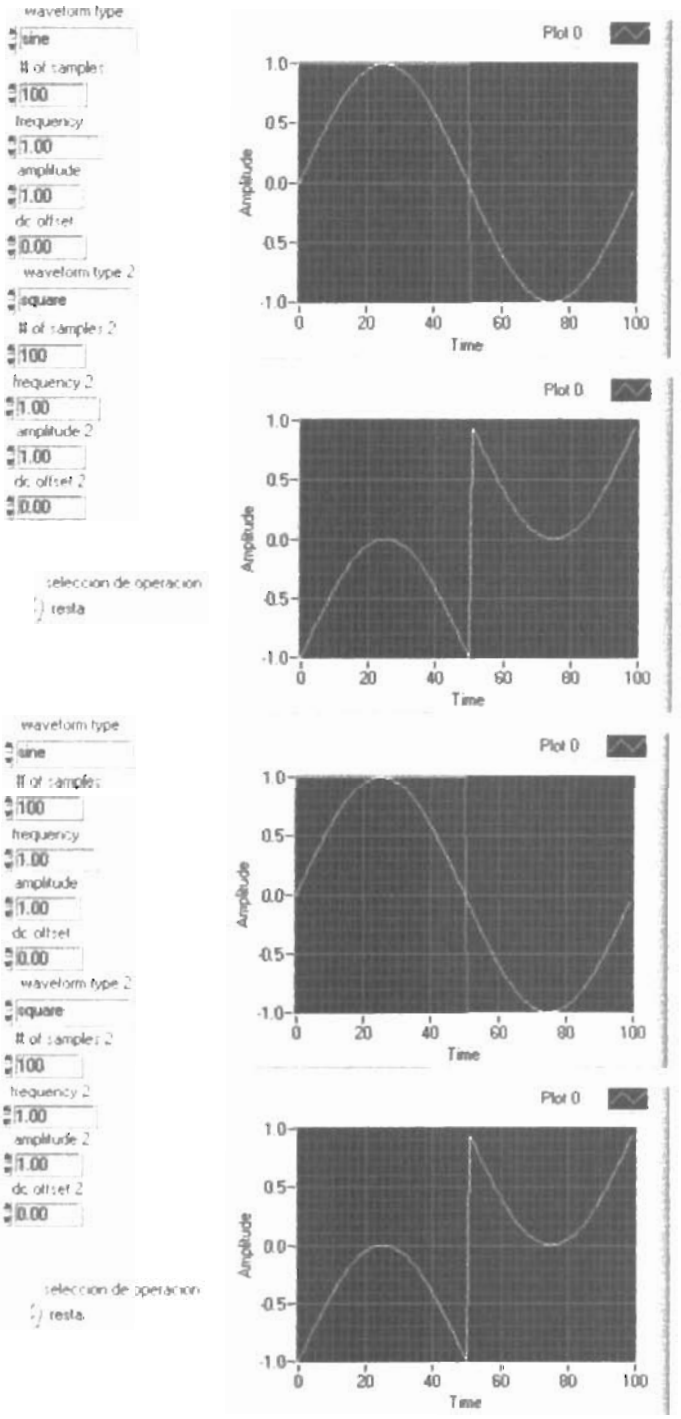


Fig. 2.7 Panel de control el cual muestra operaciones de suma y resta de iguales señales

2.5 Señales arbitrarias

Esto es de ayuda en aplicaciones para la determinación de funciones de transferencia de sistemas analógicos. El sub programa *Formula generation.vi* es el medio indicado para

generar el siguiente subprograma: lo encontramos en *waveform generation Analizis Function* del panel de control. A éste sub programa le adecuamos algunos controles para adecuar la frecuencia y amplitud, además de inicializar la señal generada y por supuesto la formula, la cual introducimos por medio de un arreglo *de index array*. El diagrama queda de la siguiente manera.

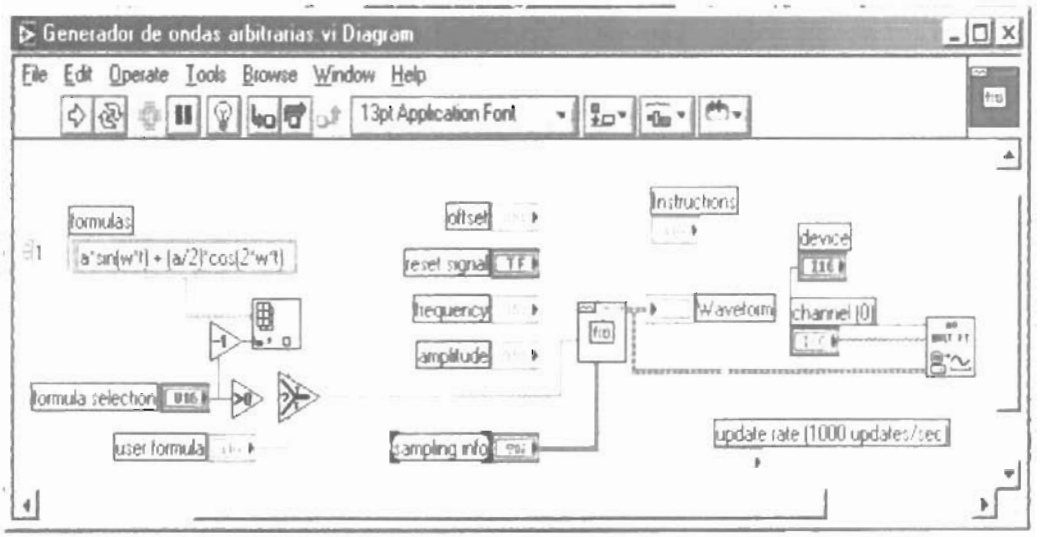


Fig. 2.8 Generador de ondas arbitrarias diagrama de bloques

Y el panel de control se muestra como a continuación:

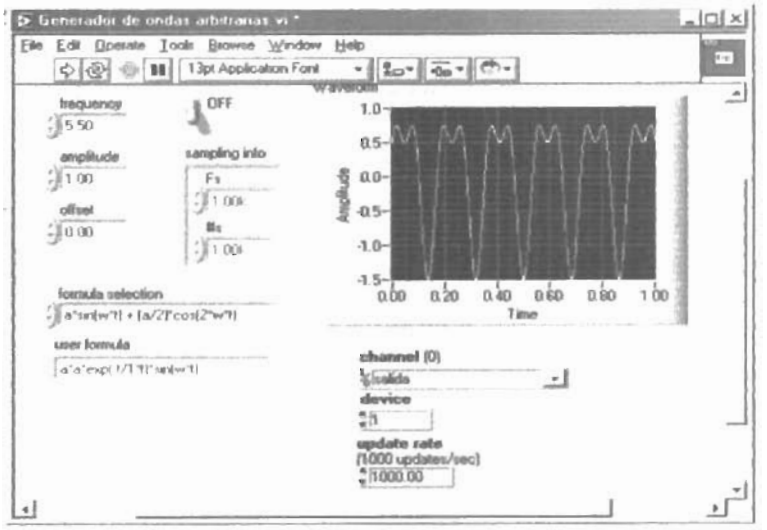


Fig. 2.8 Generador de ondas arbitrarias Panel de control

2.6 **Convolución, Deconvolución y Correlación: caso continuo**

La siguiente descripción del programa nos ofrece operaciones como la convolución, la correlación y la deconvolución. Aquí tenemos que tomar la importancia de la convolución, que la definimos como la integración del producto de dos funciones sobre el rango en el tiempo deseado (usualmente de $-\infty$ a $+\infty$). Si una señal de entrada se convoluciona con la respuesta al impulso unitario de la señal, el resultado de la señal es la misma que si la señal hubiera pasado a través de un sistema físico con la misma respuesta al impulso (o respuesta en frecuencia). Convolucón en el dominio del tiempo es equivalente a la multiplicación en el dominio de frecuencia, lo cual es equivalente a la suma en el dominio del espectro. Además incluimos la deconvolución que es la operación inversa de la convolución. Cuando una señal es deconvolucionada por la respuesta al impulso de un sistema, es básicamente reconstruir la señal para que si ha pasado por el sistema haya tenido una ideal respuesta al impulso (correspondiendo a la bandera en la respuesta en frecuencia). Las señales pueden ser deconvolucionadas para analizar las otras respuestas deseadas . La deconvolucion se puede computar el dominio de la frecuencia, en donde el espectro de la señal es dividido por la respuesta en frecuencia del sistema. A esto lo llamamos ecualización. Desde que la respuesta en frecuencia es compleja (esto es que contiene ambos magnitud ganancia y la parte de la fase), esta ecualización también es correcta para respuesta de fase.

El coeficiente de correlación R es una medida estadística de la relación entre dos variables. El valor puede ser de un rango de -1 (100% correlación negativa) a +1 (100% correlación positiva), 0 significa que no hay correlación. Este valor R es usado para designar los coeficientes de correlación. Estas son herramientas de vital importancia en el procesamiento digital de señales ya que son operadores los cuales describen la respuesta a un sistema lineal invariante en el tiempo, en donde la convolución puede ser evaluada en diferentes maneras aquí la mostramos gráficamente al seleccionar dos tipos de señales y ver su respuesta. Para la creación del programa se utilizo el mismo programa anterior solo cambiamos los operadores anteriores suma resta multiplicación por convolución, correlación y en la opción de deconvolución. Primero en este caso particular hacemos convolución para XY y después deconvolucionarlos solo con Y para obtener la

deconvolucion; así también en el panel de control adecuamos el control con los nombres correctos.

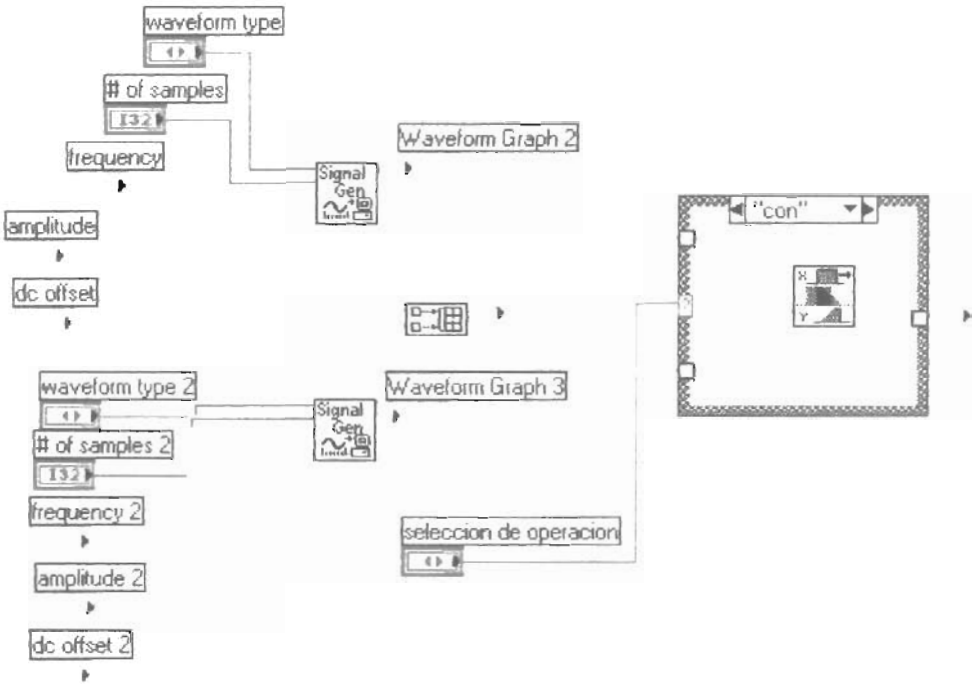


Fig 2.9-a Diagrama de bloques para la convolucion

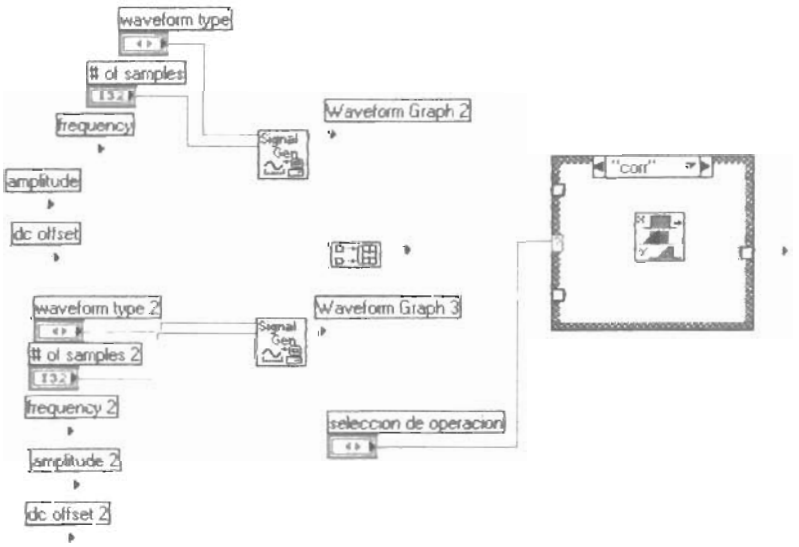


Fig. 2.9-b Diagrama de bloques para la correlación de ondas

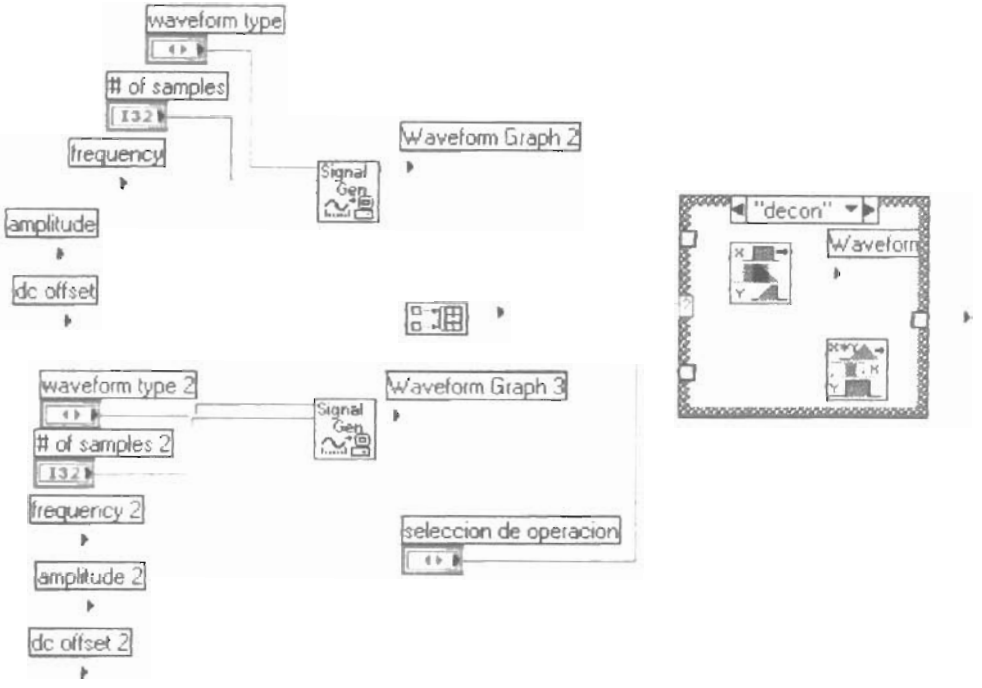


Fig 2.9-c Diagrama de bloques para la deconvolucion de ondas en el dominio del tiempo

Ahora observaremos en el panel de control para algunos ejemplos determinados el comportamiento de cada uno de los operadores.

Observamos la **convolucion** de una onda seno con un coseno.

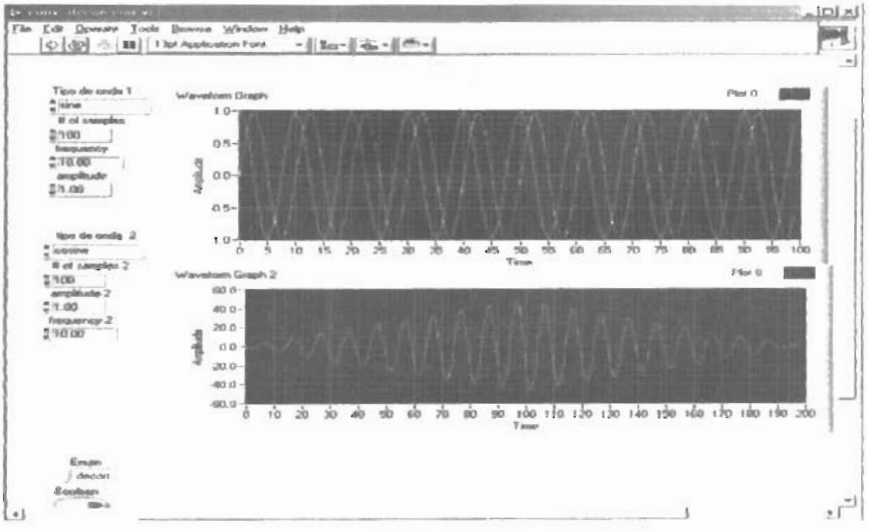


Fig 2.10-a Convolucion entre ondas seno y coseno

Ahora observaremos un poco la **correlación** dedos señales para este ejemplo ultilizamos para observar un poco mejor con ondas con frecuencia y amplitud de 1

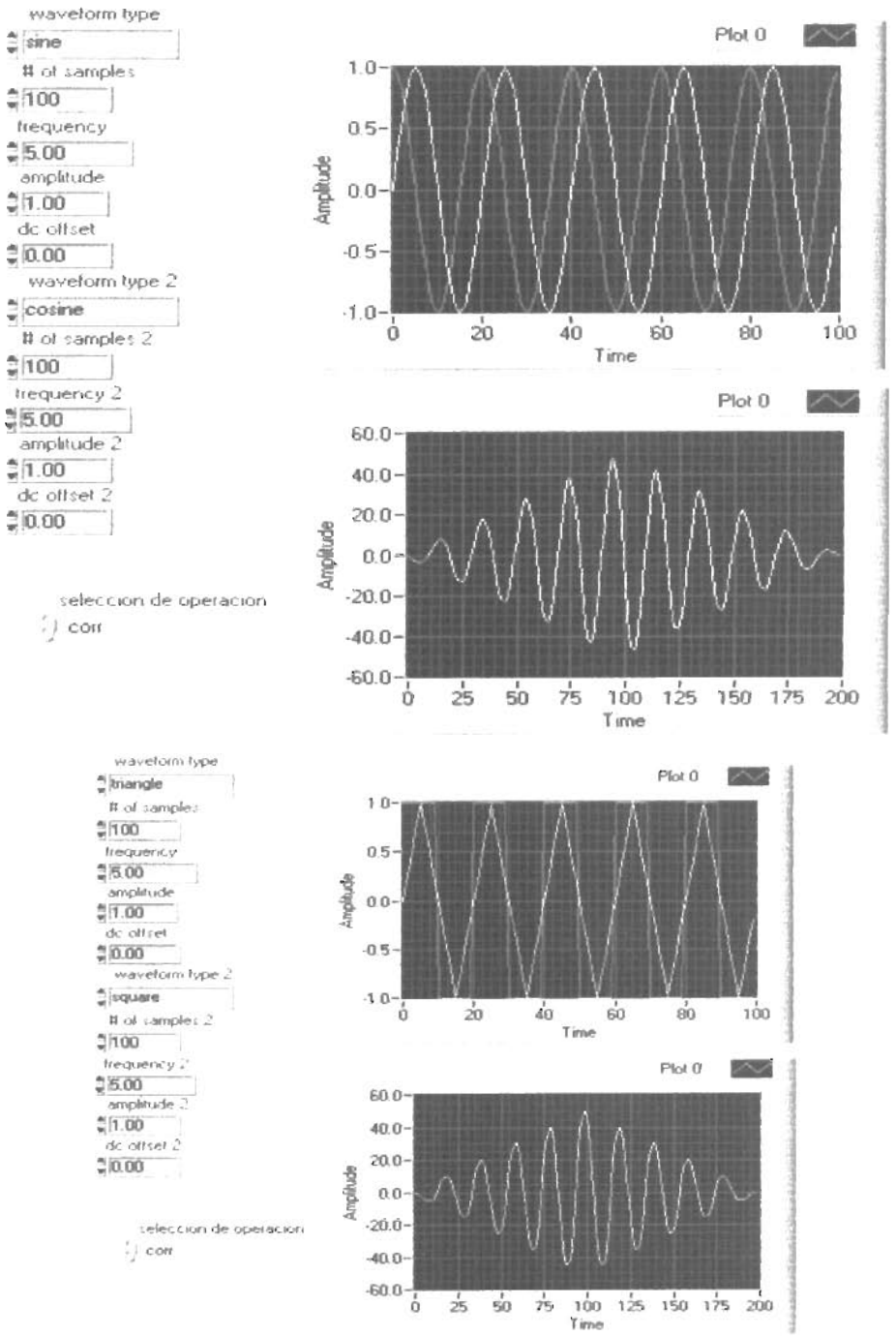


Fig 2.11 a) correlación de séno y cósenó. b) correlación cuadrada y triangular

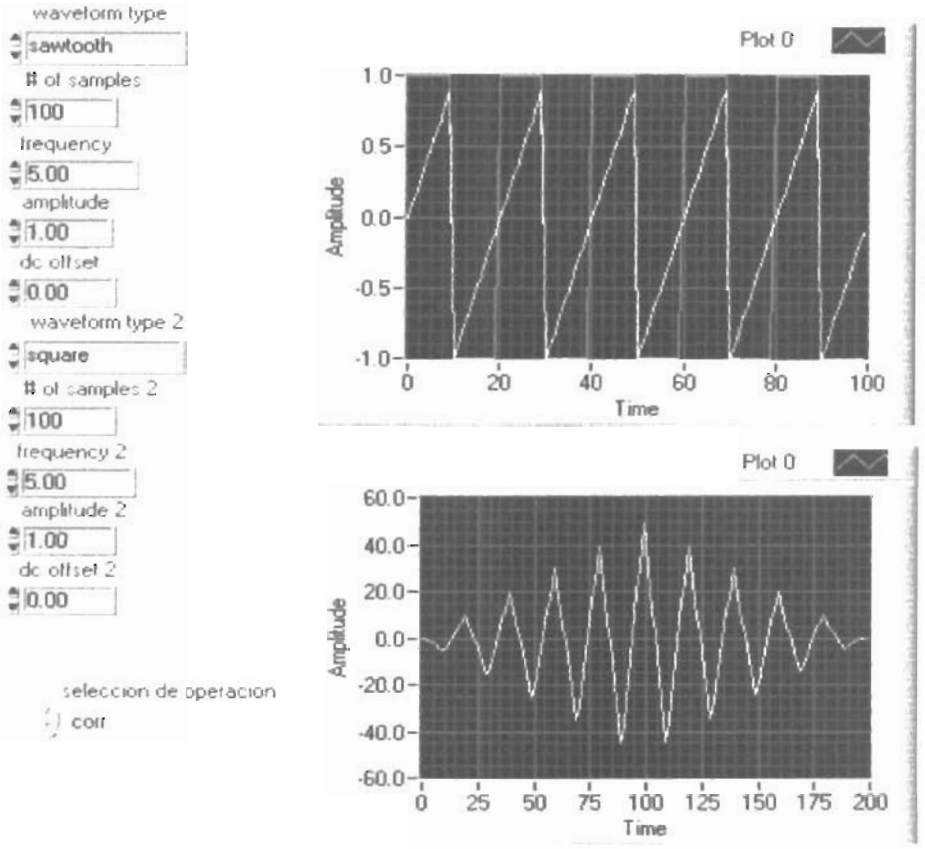


Fig.2.11-c Correlación diente de sierra y una cuadrada

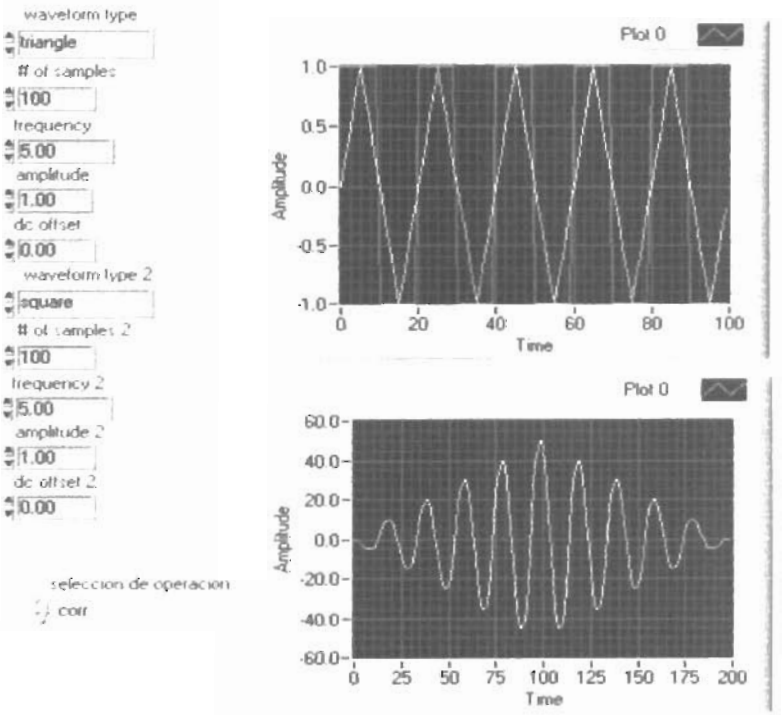


Fig.2.11-d Correlación triangular y una cuadrada

Capítulo Tres

PROCESAMIENTO DE SEÑALES

En este Capítulo discutiremos las funciones básicas de transformar una señal del dominio del tiempo al dominio de la frecuencia.

La transformada discreta de Fourier (DFT) y la transformada rápida de Fourier (FFT)

El subprograma ***signal generation by duration*** es un subprogrma establecido en Labview que nos permite generar diferentes tipos de señales acondicionando los controles adecuados para la generación de las diferentes ondas como los son tipo de onda, frecuencia, amplitud, numero de muestras y el nivel de DC (offset). Con este subprograma analizaremos los siguientes puntos:

- Comprender el funcionamiento de DFT y la FFT representados en un IV.
- Determinar los espacios de la frecuencia entre ejemplos de FFT
- Familiarizar con la respuesta espectral y como se diferencian DFT y FFT
- Comprender como interpretar la información en el dominio de la frecuencia para cada una
- una DFT/FFT y sus respuestas espectrales.

HERRAMIENTAS DEL PROGRAMA:

Transformada de Fourier: Operación matemática que resuelve una señal dada en suma de senos y cosenos, ampliamente usada. Provee información de su composición harmónica.

Algoritmo DFT: La transformada discreta de Fourier es una técnica computacional para calcular la transformada de Fourier y una versión rápida de esta técnica es la transformada rápida de Fourier.

3.1 Por que la frecuencia es tan importante?

Algunas medidas son fáciles en el dominio de la frecuencia pero difíciles en el dominio del tiempo; al adquirir datos en el dominio del tiempo la transformada rápida de Fourier la convierte del dominio del tiempo a el dominio de la frecuencia. En el dominio del tiempo la señal se asume como la suma de senos en diferentes frecuencias. Los ejemplos obtenidos de una señal de una tarjeta de adquisición DAQ es la representación de la señal en el dominio del tiempo. Esta representación proporciona la amplitud de la señal en instantes de la duración del tiempo en el cual la señal es muestreada. Como sea da la mayoría de los casos, lo que nos interesa conocer es el contenido de la frecuencia Este dominio nos ayuda a comprender mejor acerca de la señal y el sistema de el cual fueron generadas. Esta es utilizada ampliamente en los campos de análisis espectral, acústica, imágenes médicas, análisis numéricos, instrumentación y telecomunicaciones. El algoritmo el cual nosotros usamos para transformar las muestras de datos del dominio del tiempo hacia el dominio de la frecuencia es conocido como Transformada Discreta de Fourier o DFT; Esto no es más que la relación que establece entre los dominios del tiempo vs. Frecuencia

Para entender lo anterior, supongamos que se han obtenido N muestras de una señal obtenida por una tarjeta de adquisición DAQ. Si nosotros aplicamos la DFT a los N ejemplos de esta señal en el dominio del tiempo, el resultado será el mismo largo de N ejemplos, pero la información contenida será en el dominio de la frecuencia.

Figra 3.1 Representación en el dominio del tiempo y el dominio de la frecuencia

3.2 Información en Fase y magnitud

Hemos visto que las N muestras de la señal de entrada resultan en N muestras de salidas en la DFT. Esto es, que el número de ejemplos de ambas en el tiempo y en la frecuencia es el mismo. Esta la señal de entrada tiene dos partes real e imaginaria, aunque algunas veces la imaginaria es cero. Asi, por eso la Transformada discreta de Fourier es compleja, conteniendo información en Magnitud y Fase. Esto se muestra a continuación. En la grafica de la Fig. 3.2 se pueden observar dos picos en la representación del dominio de la frecuencia, pero esto significa que son los armónicos de la señal la cual contiene información de armónicos positivos como armónicos negativos.

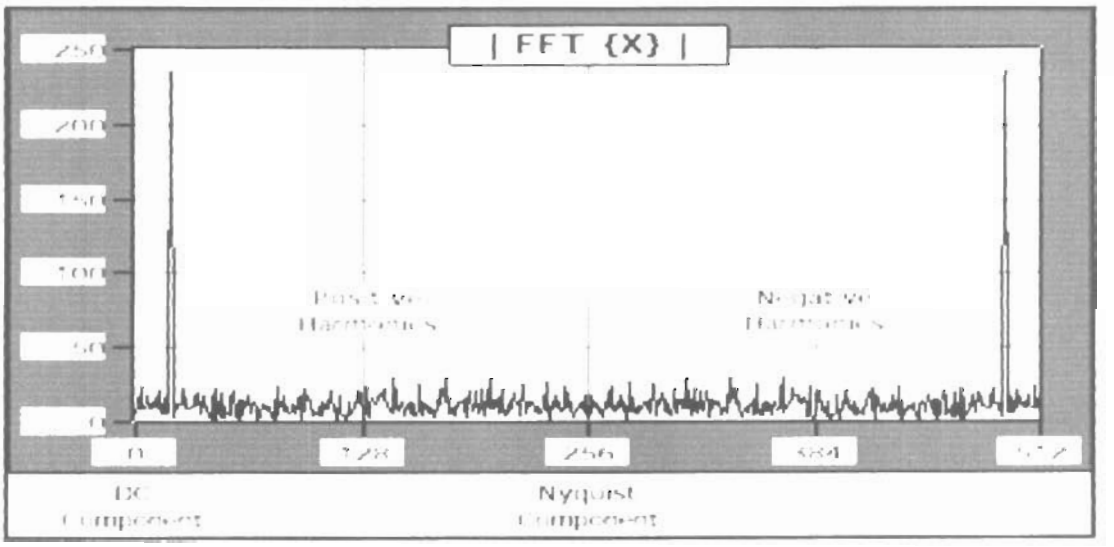


Figura 3.2 Dominio de la frecuencia. FFT a una secuencia de entrada x, parte real e imaginaria

3.3 La transformada rápida de Fourier

Un caso especial de al transformada de Fourier (discreta), es el caso de la transformada rápida de Fourier. Una implementación directa de la transformada discreta de Fourier en N muestras requiere aproximadamente operaciones complejas de N al cuadrado y esto hace que el tiempo en el proceso sea mucho más largo, a medida de que N crece. Sin embargo, cuando el tamaño de al secuencia es una potencia de dos, $N=2^m$ con m igual a 1,2,3..., el cómputo de la DFT se puede reducir aproximadamente a N*Log(N) operaciones, lo cual en tiempo significa una reducción considerable. Este calculo hecho en la DFT hace mucho

más rápido el procesamiento digital de señales, el cual este caso especial ya esta hecho en un algoritmo el cual llamamos Transformada Rápida de Fourier. La transformada rápida de Fourier no es mas que un algoritmo el cual calcula un caso especial de DFT cuando el numero de ejemplos es potencia de 2, en algunos casos la transformada rápida de Fourier puede ser utilizada en algunos otros casos especiales en secuencias largas. La ventaja de este algoritmo incluye la rapidez y menor uso de memoria. La transformada discreta de Fourier puede procesar eficientemente cualquier secuencia de cualquier tamaño pero es más lenta y la transformada rápida de Fourier utiliza menos memoria por que almacena inmediatamente los resultados durante el procesamiento. En el siguiente programa observaremos como una señal en el dominio de la frecuencia cuenta con dos partes real e imaginaria utilizamos un generador de onda senoidal adecuando los controles de Frecuencia, número de ondas, y frecuencia de muestreo, al salir la onda es como si esta fuera la onda en el dominio del tiempo. Para transformarla a el dominio de la frecuencia utilizamos el subprograma Real FFT, el cual encontramos en ***Functions>Signal Processing>Frecuency domain>Real FFT.***

La salida del subprograma Real FFT sumamos esa salida con la salida de la señal real adecuándolas con un ***array size*** y se suman las dos la salida de esta suma se cambia de complejo a polar con el subprograma ***Complex to polar.VI*** y en la salida podemos observar la información pero en dominio de la frecuencia, en el cual observamos dos picos esto es que esta tanto la parte real como la imaginaria.

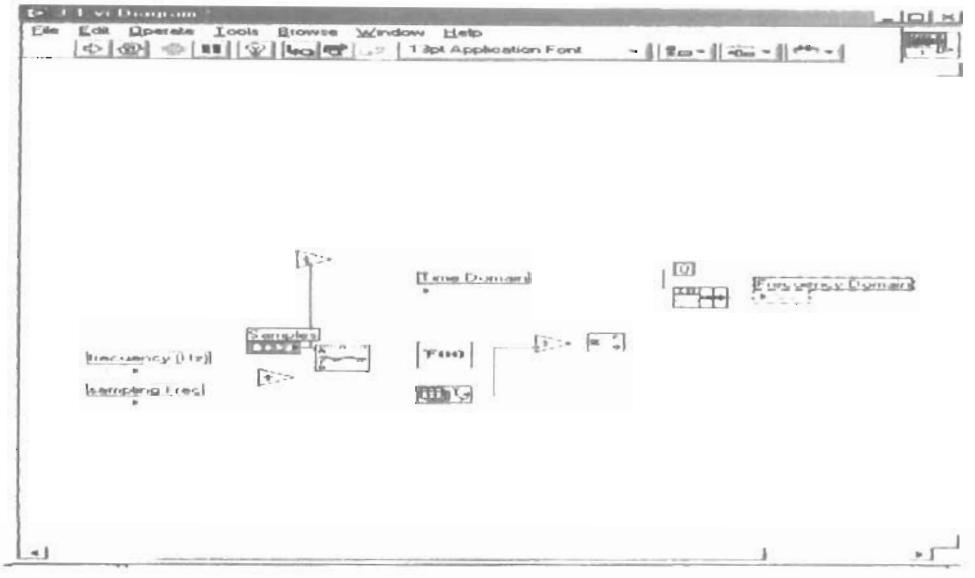


Figura 3.3 VI por el cual obtenemos la transformación del domino del tiempo al dominio de la frecuencia

Aquí podemos examinar la frecuencia del espectro es decir la transformada de Fourier. En los controles generamos una frecuencia de 40 Hz y en la gráfica se nos muestran dos picos: uno en 40 y el otro en 60. El pico en 60 representa la frecuencia de -40Hz, la grafica que se muestra en la Fig. 3.4 es conocida como FFT doble, ya que nos muestra las frecuencias positivas como negativas, como se vio anteriormente.

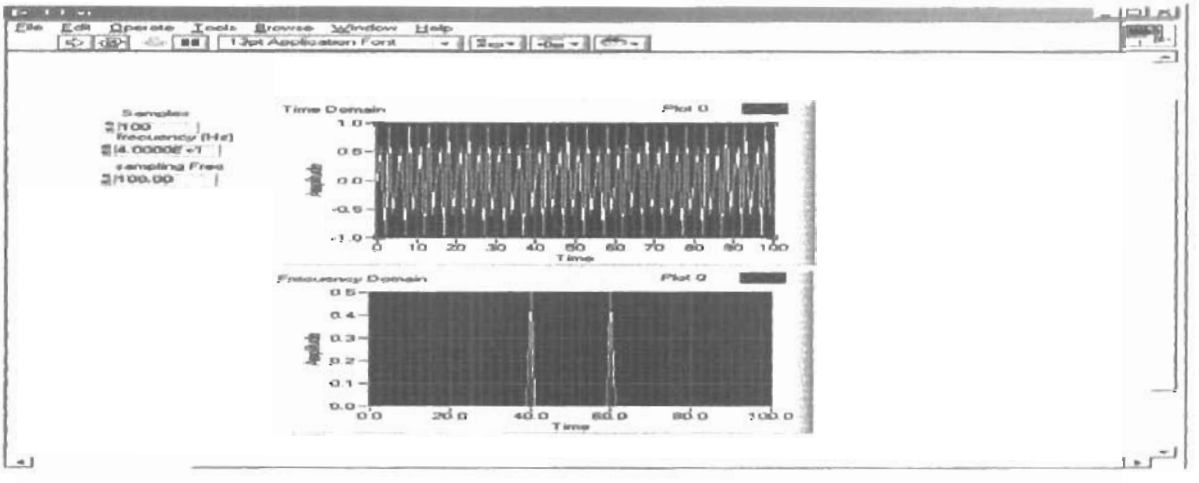


Figura 3.4 En el subprograma se ve la parte real e imaginaria solo se puede ver hasta cierto rango de frecuencia que no interfieran.

Ahora observaremos cuando tenemos una frecuencia de 10Hz. En la frecuencia, la grafica nos (ver Fig. 3.5) dos picos, el primero en 10 y el segundo en 90. El de 90 representa la frecuencia en -10Hz. Aqui también cabe hacer notar que la representación de la onda seno se observa de mejor manera.

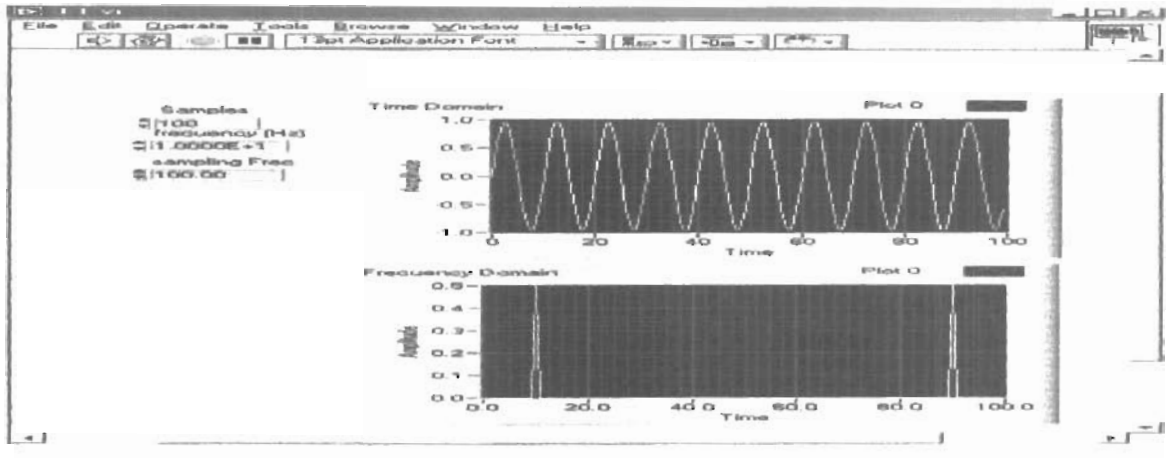


Figura 3.5 Frecuencia de muestreo de 10 Hz, se observa mejor la onda senoidal

El porqué la frecuencia de muestreo (sampling Frecuency) es igual a 100Hz se debe a que solo se pueden muestrear frecuencias menores de 50Hz. Ahora al cambiar la frecuenta a 48 Hz es la ultima que se podrá observar, si pusiéramos una frecuencia de 49.5Hz este superior al nivel permitido para observar que la lectura en la grafica será errónea porque esta frecuencia sin ser superior a lo establecido por el teorema de Nyquist hace que los picos se junten en uno solo. Ya que se junta mucho con 49.5 Hz y solo se observa un gran pico en el centro, también se observa que la grafica en el dominio del tiempo la frecuencia se distorsiona.

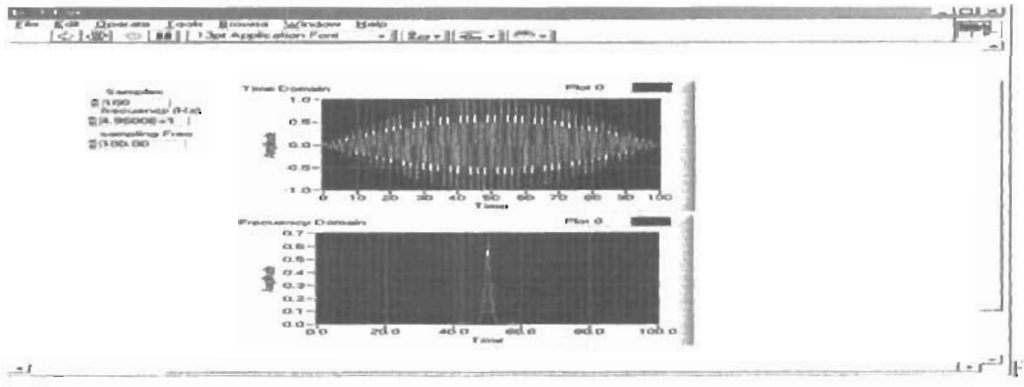


Figura 3.6

Frecuencia de muestreo de 49.5 Hz, se observa que los dos picos se juntan en uno solo

3.4 **Diferencia entre la representación del FFT y la respuesta espectral.**

El espectro en cada componente de la frecuencia representada por la DFF/FFT puede ser obtenida si la magnitud de los componentes de frecuencia se elevan al cuadrado. Asi el espectro en la k-esima componente de frecuencia es dado por $|X[k]|^2$. La grafica mostrara el cuadrado de cada una de las componentes de frecuencia esto es conocido como respuesta espectral. Porque DFT/FFT son señales reales y simétricas, el valor absoluto de el cuadrado de los componentes es el mismo para le valor de las frecuencias positivas y negativas. Debido a que la potencia, obtenida elevando al cuadrado la magnitud de DFF/FFT, la respuesta espectral es siempre real y toda la información de la fase se pierde. Si se desea información en la fase se debe de usar la FFT o DFT. Podemos usar la respuesta espectral en aplicaciones en las cuales la fase no sea tan importante, por ejemplo al calcular el armónico en una señal. En el siguiente ejemplo observaremos las diferencias entre la FFT y la representación de la respuesta espectral. El siguiente es el diagrama de bloques del VI.

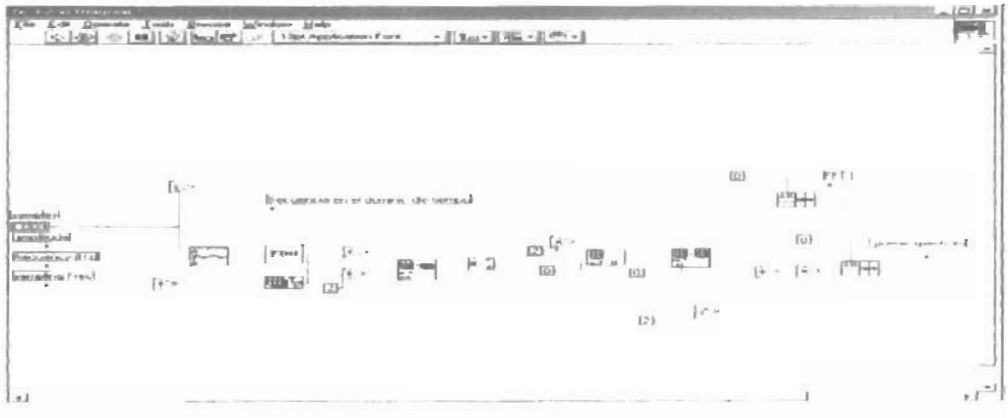


Figura 3.7 Diagrama de bloques del sub programa para observar la diferencia entre fft y respuesta espectral

En el panel de control ponemos los siguientes valores en las entradas (amplitud = 1.414, frecuencia (Hz) = 20 Hz y la frecuencia de muestreo = 100 Hz). El número de muestras es 100. El resultado se muestra en la Fig. 3.8.

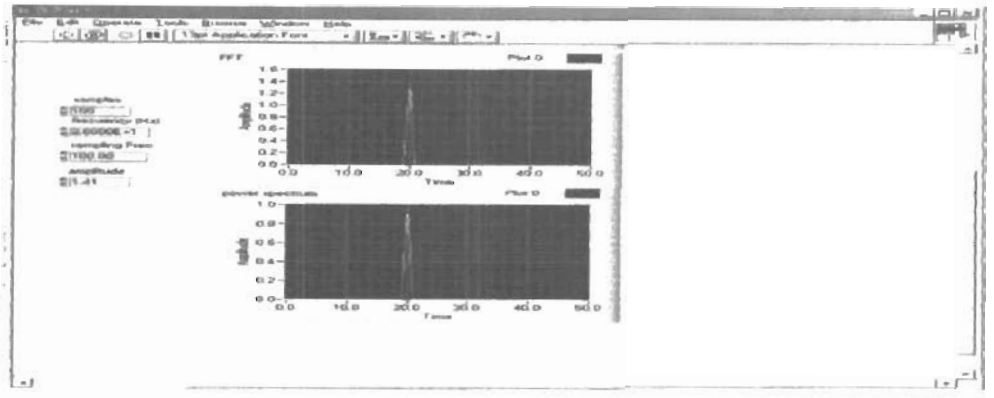


Figura 3.8 Panel de control del VI para conocer la diferencia entre fft y respuesta espectral.

Podemos observar las diferencias entre ambos resultados ya que la respuesta espectral es obtenida elevando al cuadrado la magnitud de fft. Cada punto se eleva al cuadrado Ahora en la siguiente ejemplo ponemos solo 1 en la amplitud y observamos la gráfica

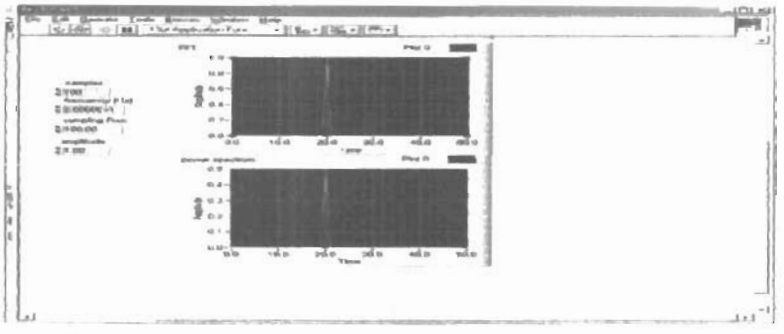


Figura 3.9. Diferencia entre fft y respuesta espectral con la amplitud =1

Aquí observamos la amplitud que esta en 1 y vemos como la respuesta espectral esta solo en 0.5 (ver. Fig. 3.9).

Capítulo Cuatro

Filtros Digitales

Dentro de las aplicaciones del procesamiento digital de señales, la de con filtros digitales es la más importante ya que permite manipular de manera directa las señales muestreadas en base a un algoritmo preestablecido. Sin embargo, es necesaria la comprensión de dicho tópico a manera de simulación entre los estudiantes, antes de diseñar sistemas de aplicación con filtros digitales con hardware. Para tal fin, en este capitulo abordaremos acerca de las caracteristicas de diferentes tipos de filtros digitales y el cómo podemos usarlos en aplicaciones en procesamiento digital de señales (PSD). Los puntos a tratar son:

4.1 Principios básicos de filtrado y porqué es necesario
4.2 Caracteristicas de diferentes tipos de filtros ideales
4.3 Conocer las diferencias y ventajas de filtros digitales sobre filtros analógicos
4.4 Comprender las principales diferencias entre filtros IIR y los filtros FIR.
4.5 Caracteristicas de los filtros IIR y FIR

4.1 Filtrado y Filtros Ideales

Filtrado es el proceso por el cual la respuesta en frecuencia contenida en una señal y sistema es alterada y es la herramienta más usada en el procesamiento de señales. Un ejemplo común es simplemente el control en nuestro estéreo de los bajos y los medios en el control: el bajo controla o elimina las bajas frecuencias contenidas en la señal y los medios altera o/y elimina las altas frecuencias. Algunas otras aplicaciones son eliminar ruidos eléctricos. En relación a esto, los filtros alteran o remueven frecuencias no deseadas. Dependiendo del rango de frecuencia que ellas tengan para atenuar, los filtros pueden clasificarse en *pasabandas* (PB), *pasaaltos* (PA), *rechabanda* (RB), *pasabajos* (PB) y

notch. Es posible, además, hacer transformaciones entre diferentes tipos de filtros. En la Fig. 4.1 (a) y (b) se muestra un ejemplo típico de un filtro PB de fase lineal en modo absoluto y ganancia relativa.

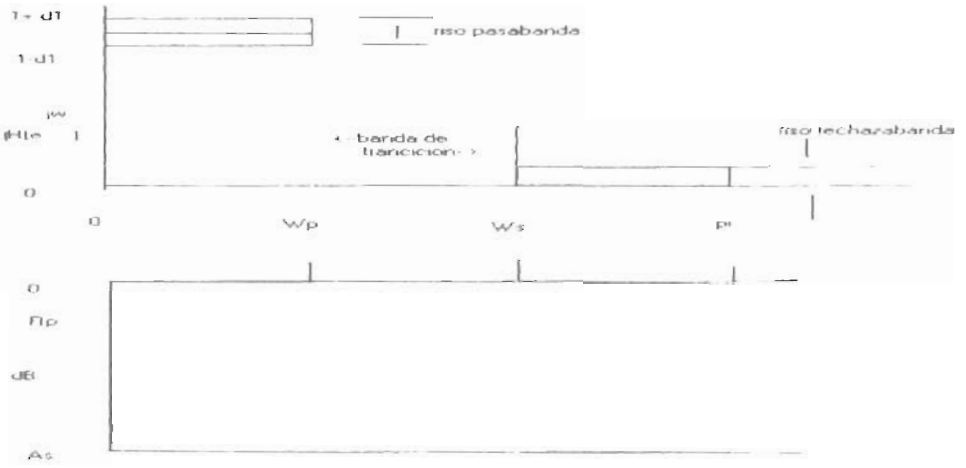


Figura 4.1 Respuesta en frecuencia en la especificación de un filtro de fase lineal (a) absoluta y (b) relativa

4.2 Ventajas de Filtros digitales sobre filtros analógicos

Un filtro analógico tiene señal analógica en las entradas y salidas. Ambas, entrada x(t) y la salida y(t), son funciones de variables continuas en el tiempo y toman un número infinito de valores. Afortunadamente, para los filtros analógicos, la teoría está bien desarrollada y por lo tanto, el pasar a su correspondiente digital no es mucho problema. En aplicaciones, se incluyen audio, telecomunicaciones y diagnóstico médico. Las ventajas de los filtro digitales son las siguientes.

- Son programados por software y son fáciles de simular
- Estos requieren solo operaciones aritméticas como multiplicación, suma resta división y son relativamente fáciles de implementar
- Estos son estables (esto quiere decir que no cambian con el tiempo o la temperatura) y son predecibles
- Tienen un mejor desempeño en cuanto a calidad de salida
- Estos no sufren errores de fabricación o variación en componentes

4.3 Diferencias entre FIR e IIR

Los filtros digitales a su vez se dividen en dos: filtros FIR y filtros IIR. Esta clasificación se basa a las diferencias en la respuesta al impulso: FIR, cuya respuesta al impulso unitario tiene una longitud finita e IIR, cuya respuesta al impulso es una secuencia de longitud infinita. Cualquiera que sea la respuesta al impulso del filtro sea finita o no, depende de cómo sea calculada la salida. La diferencia básica entre los filtros FIR y los IIR es que, para los FIR la salida solo depende de la entrada de valores pasados; es decir estos no tienen retroalimentación; mientras que para los IIR la salida depende no solo de los valores de salida pasados si no que también de valores todavia anteriores y futuros a la entrada actual. Asi, se escriben las funciones de transferencia para ambos filtros FIR e IIR, respectivamente como:

$$H(z) = b_0 + b_1z^{-1} + b_2z^{-2} + \ldots + b_Mz^{-M}$$

y

$$H(z) = (b_0 + b_1z^{-1} + b_2z^{-2} + \ldots + b_Mz^{-M})/(1 + a_1z^{-1} + a_2z^{-2} + \ldots + a_Nz^{-N}),$$

En donde los conjuntos de valores $\{a_i\}$ y $\{b_j\}$ son los coeficientes del filtro y el orden del filtro es el máx[M,N].

4.4 Ventajas y desventajas entre los filtros IIR y FIR

Comparando los desempeños de los diferentes filtros, una de las ventajas de los filtros IIR sobre los FIR, es que los filtros IIR usualmente requieren menor coeficientes para diseñar y desempeñar un filtro de similares operaciones. Así los filtros IIR ejecutan mucho más fácil y no requiere memoria extra para ejecutarse. La desventaja de los IIR es que la respuesta en la fase no es lineal. Si la aplicación no requiere información relacionada con la fase, los filtros IIR son los más apropiados. Una de las desventajas de los filtros IIR es que el que sean retroalimentados hace que estos sean mas difíciles de diseñar he implementar. En lo que sigue, se discuten ejemplos relacionados con estos dos tipos de filtros con LabView.

Para un desarrollo detallado de la teoría general de filtros se recomienda el libro de Oppenheim o Proakis.

4.5 Características de los diferentes tipos de filtros

Para visualizar las características de los filtros FIR, utilizaremos el subprograma *FIR windowed filter design* para familiarizarse con su desempeño. El panel frontal de este sub VI es el siguiente:

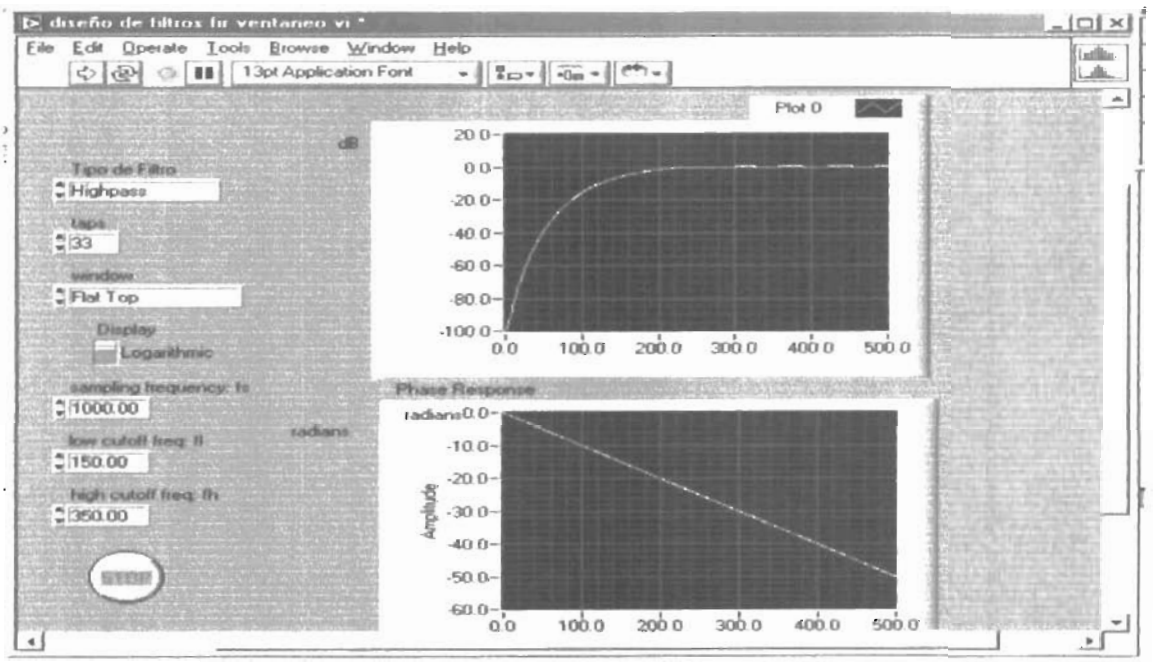


Fig 4.2 Panel de control de FIR -Ventaneo

En éste tenemos controles sobre el tipo de filtro este puede ser: PB, PA, RB; así como el tipo de ventana que queremos utilizar, ya sea Hanning, Hamming, Hanning, Kaiser-Bessel, Triangular, Blackman-Harris. Así también tenemos un control de despliegue que nos permite mostrar la respuesta en la magnitud (logarítmica o lineal). También se tienen los controles ya conocidos como frecuencia de muestreo, así como frecuencia de corte abajo y frecuencia de corte alto. El diagrama de bloques es como a continuación se muestra:

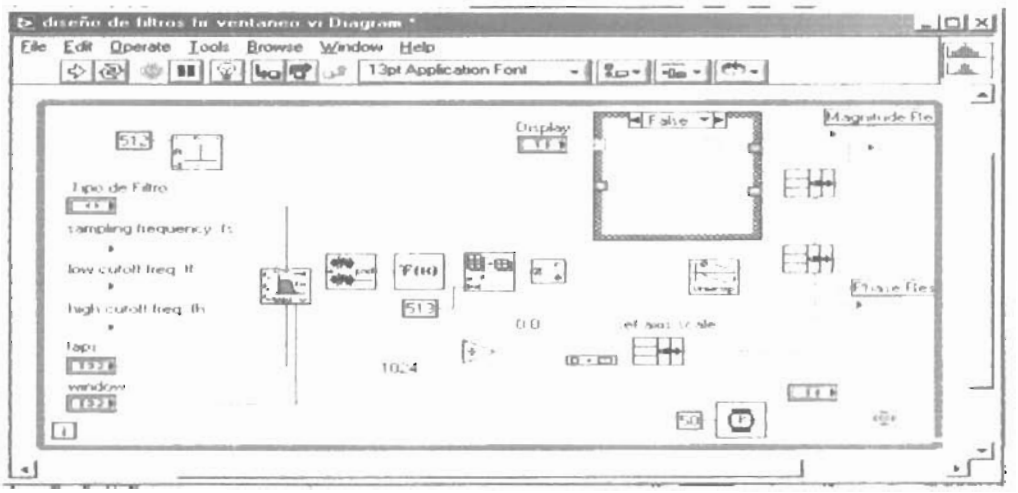


Fig 4.3 Panel de control de FIR con ventaneo

En esta parte del VI es importante hacer mención sobre los controles que aquí se muestran. A primera instancia tenemos un subprogma llamado ***FIR windowed Filter***, a el cual se le adecuan todos sus controles como son, tipo de filtro, frecuencia de muestreo, frecuencia baja de corte y frecuencia alta de corte y número de coeficientes que es usados en el filtro (ver Ec.); asi como el tipo de ventana que usamos para truncar las muestras.

En la parte de los filtros IIR el subprograma que nos ayuda a explicar más en detalle ese funcionamiento es el *IIR Filter Design*, el cual nos muestra en el panel de control el siguiente diagrama.

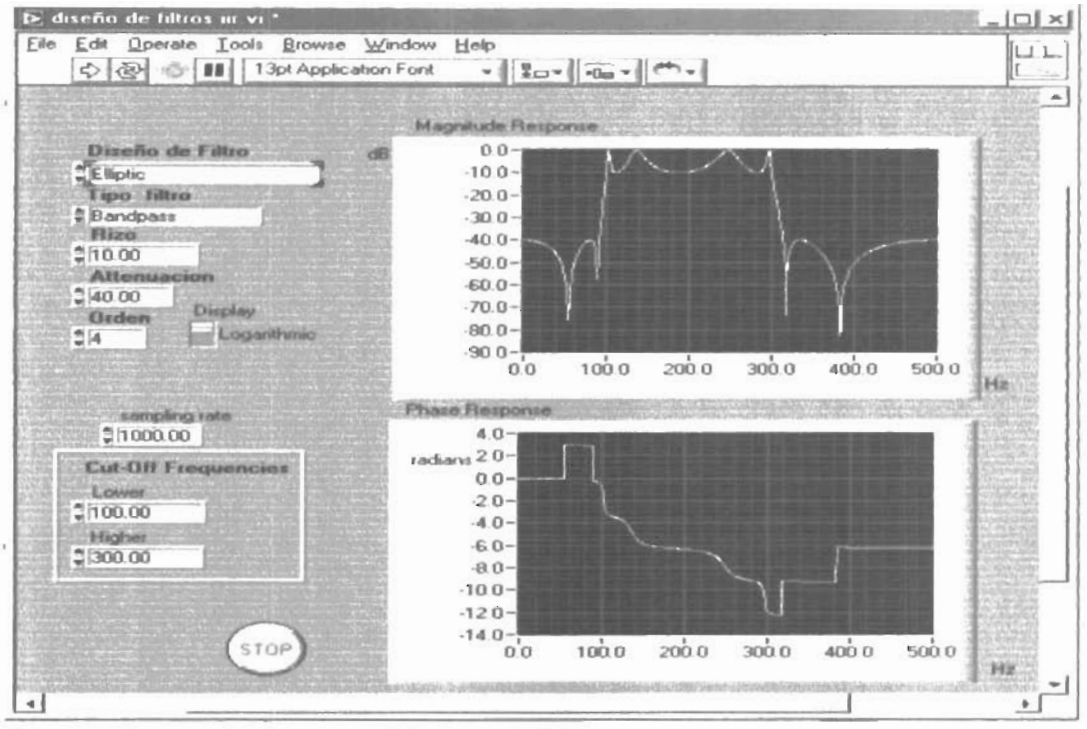


Fig 4.4 Panel de control diseño de IIR

El tipo de controles que tenemos aqui son los siguientes: el filter ***desig*** que nos designa qué tipo de filtro podemos adecuar: Eliptico, Chebyshev, inv-Chevishev, Butterworth y Bessel. Los tipos de filtros que nos permite escoger son: PB, PA y RB. Otro control importante es en relación al rizo; así como la atenuación y el número de orden del filtro. También tiene un control de visualización para observar el comportamiento lineal o logaritmico (en dB). En la parte de "Respuesta de Fase" tenemos los controles de Frecuencias de corte para la más alta o la más baja, según sea el caso.

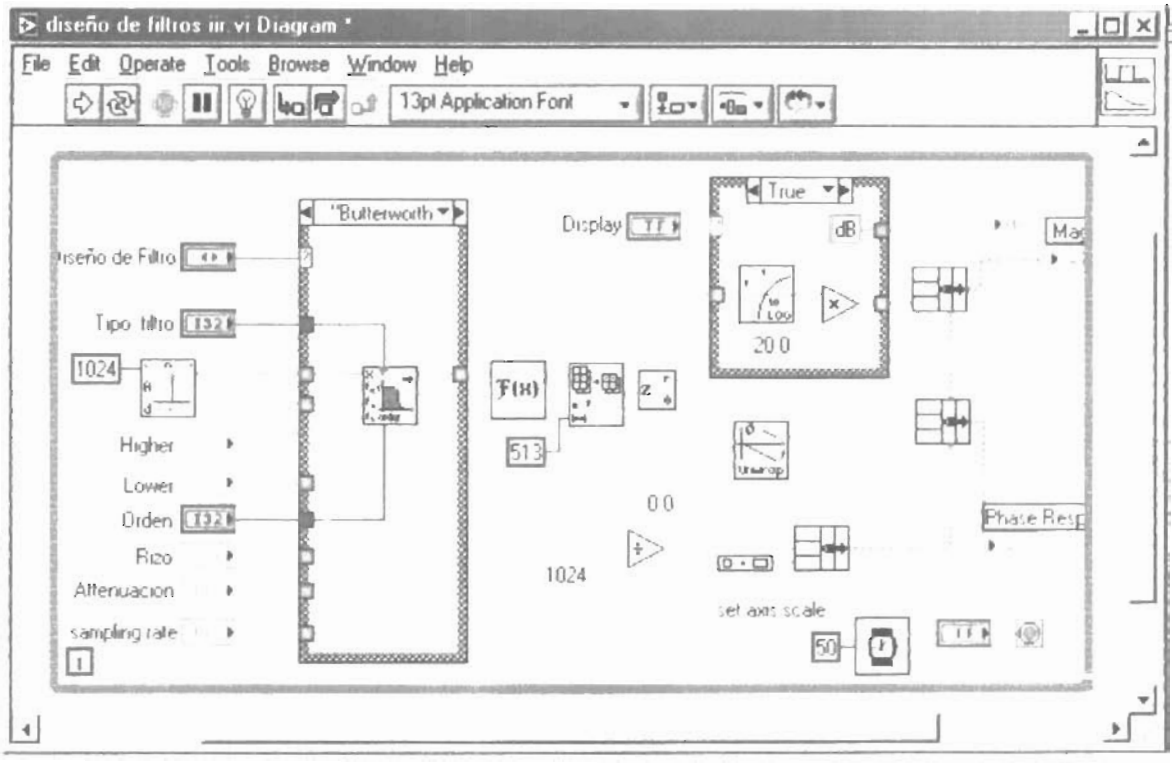


Fig. 4.5 Diagrama de bloques diseño de IIR

Algunos ejemplos de lo anterior podria ser un diseñar un filtro rechazabanda con ventana hamming con una frecuencia de rechazabanda en 150Hz y 350Hz, observar la respuesta lineal. Adecuamos los controles necesarios para dar las caracteristicas anteriormente pedidas y observamos la grafica.

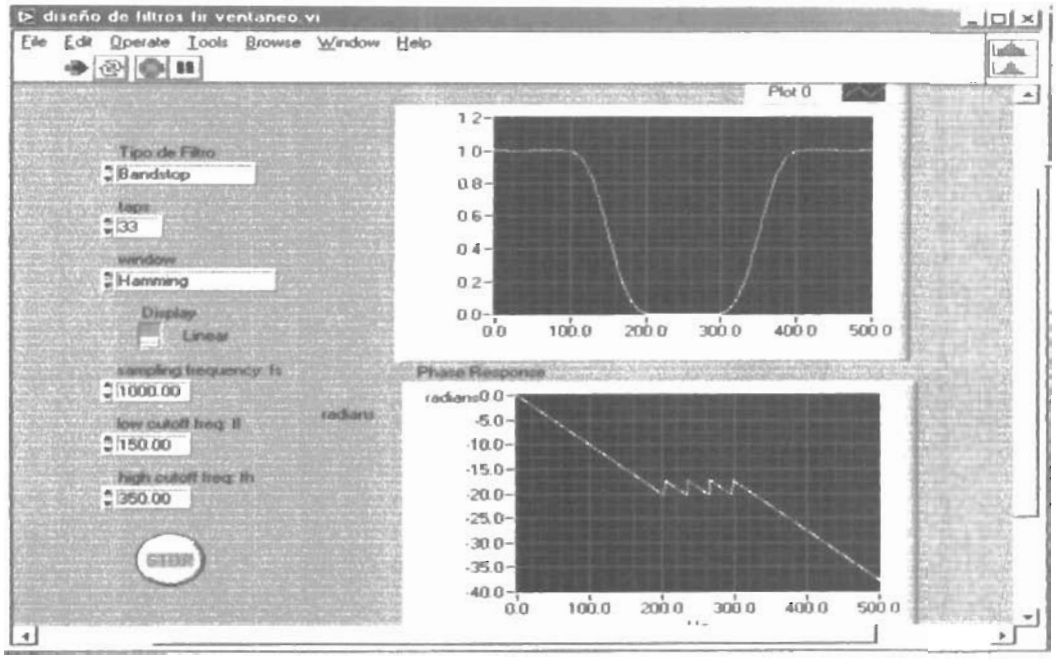


Fig. 4.6 Ejemplo de diseño de filtro con ventaneo

Para observar un ejemplo del otro tipo de filtro ponemos el siguiente diseño.

Filtro Butterworth, Pasabajo con una frecuencia de corte de 100Hz.

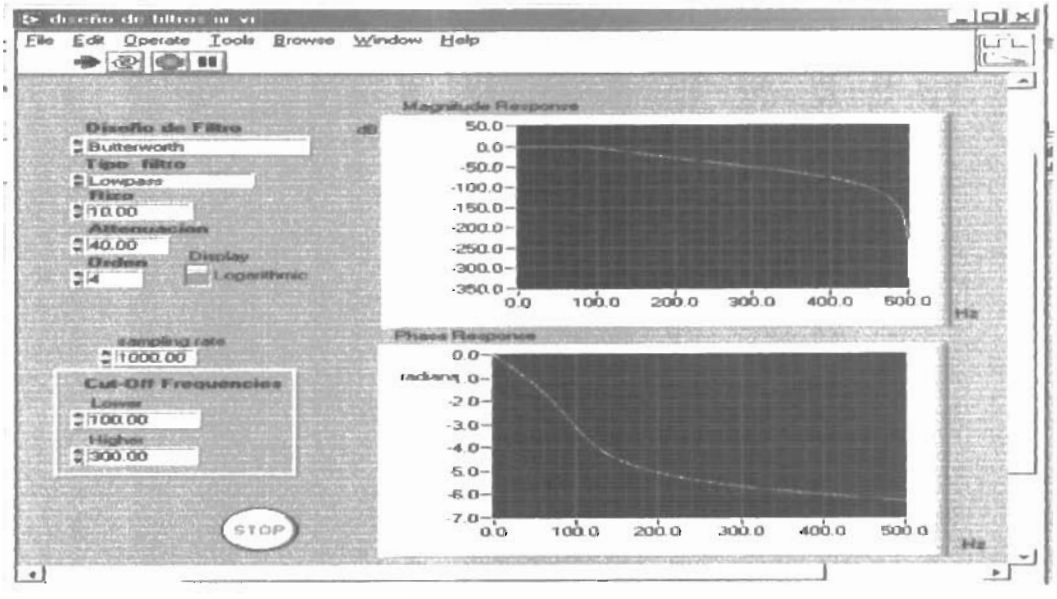


Fig. 4.7 Ejemplo de diseño de filtro IIR

4.6 Usando MatLab en LabView.

Una parte que no podemos dejar de mencionar es en este trabajo es la importancia que tiene MatLab como herramienta de diseño y ayuda para la simulación de efectos modelado matemático y diseños. Aquí LabView nos proporciona la oportunidad de trabajar en el con un ambiente totalmente de MatLab y con su lenguaje completo.

El MatLab Scrip lo encontramos en ***Functions>Matematics>formula>MATLAB SCRIPT***.

En éste bloque podemos realizar una gran variedad de operaciones en las que destaca el poder utilizar el código del lenguaje MATLAB. En las figuras 4.4. a 4.7 se muestra los dicho líneas arriba.

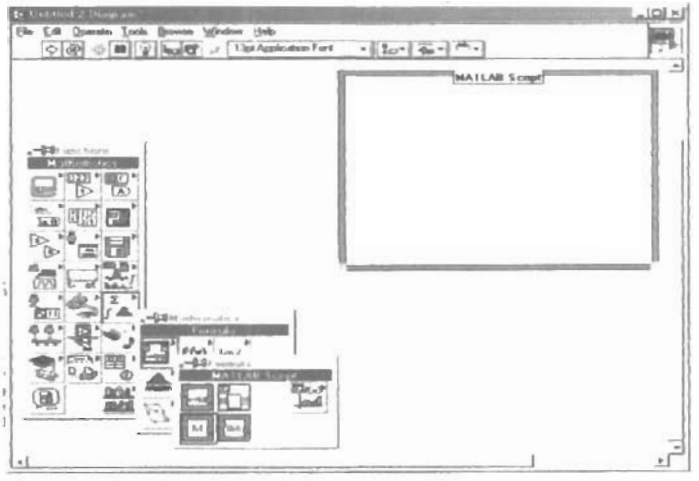


Fig 4.8 El bloque para utilizar Matlab Scrip

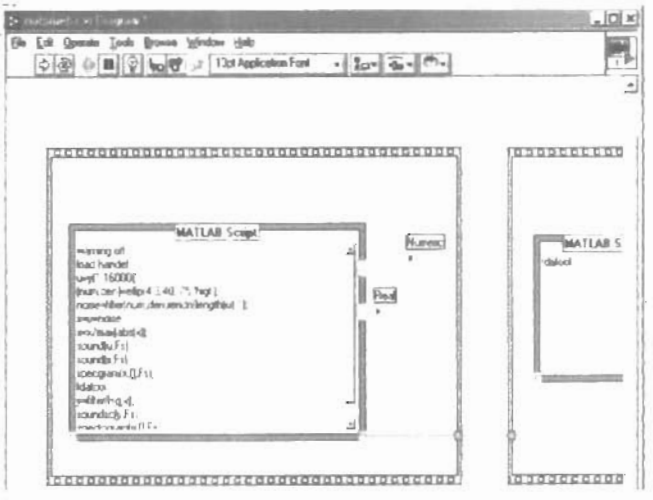


Fig 4.9 Ejemplo de Sub Vi con Instrucciones en Matlab

Capítulo Cinco

APLICACIONES

En este Capitulo se define una serie de aplicación de los cuales estuvimos tratando en los Capítulos anteriores, desde la generación de una señal, pasando por la adquisición y su representación. Se trata además el diseño de un control PID.

Los puntos a tratar serian los siguientes:

1. Generación de una señal aleatoria representación en osciloscopio
2. Adquisición de una señal
3. FFT y la respuesta espectral a una señal adquirida
4. Filtrado de la señal y su representación.
5. Control de la velocidad angular de un motor DC usando PID

5.1 Para la generación de señales, usamos las creadas por fórmula u ondas normales (seno, coseno, triangular, cuadrada, diente de sierra). En este caso el sub vi nos muestra la salida utilizamos el subprograma AO generate waveform (generador de ondas analógicas). Al cual solo le adecuamos el dispositivo de salida, asi como el canal por el cual se generara las salidas, por medio de la tarjeta (Figuras 5.1 a, b y c). Las ondas a generar serán

a) $\sin(\omega t)\sin(2\pi t)$

b) cos(2*pi(1)*t) + sen(6*pi(1)*t)

c) sin(w*t)+cos(4*pi(1)*t)

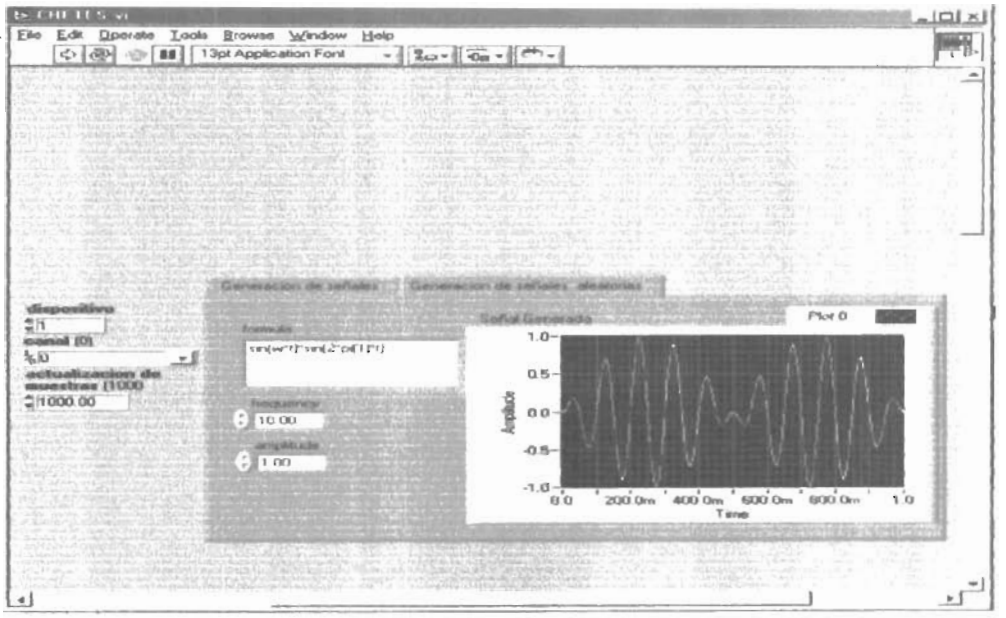


Figura 5.1a) Panel de control para la generación de señales aleatorias por formula

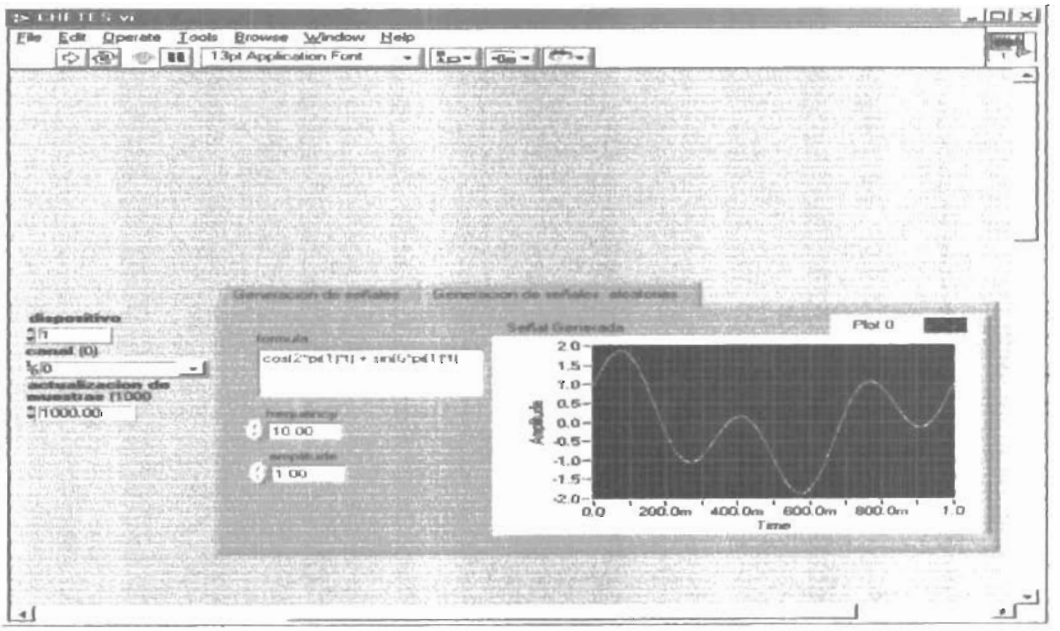


Figura 5.1 b)Panel de control para la generación de señales aleatorias por formula

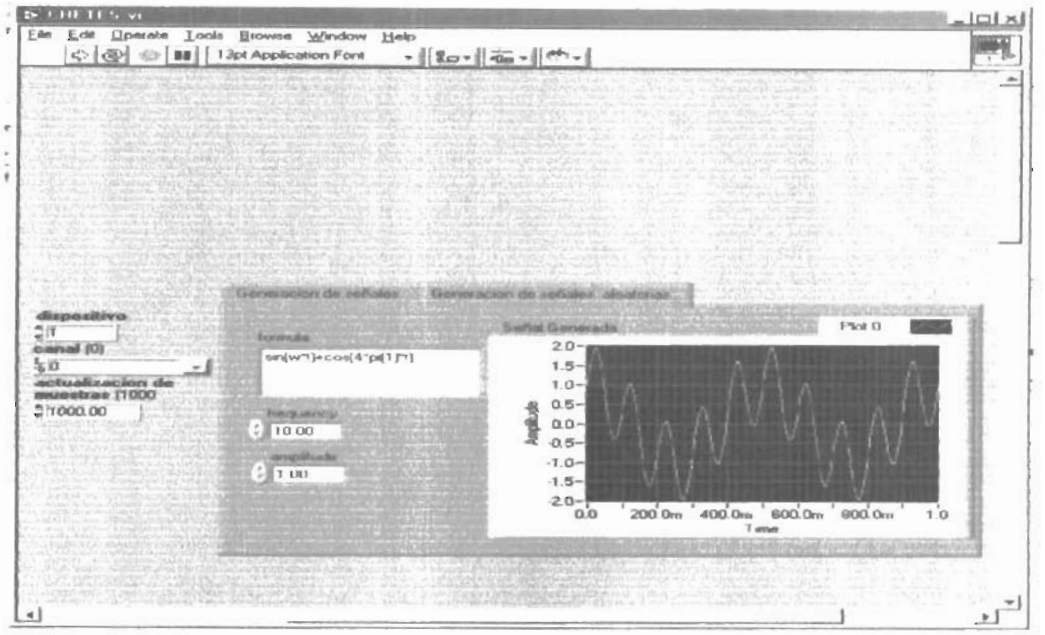


Figura 5.1 c) Panel de control para la generación de señales aleatorias por formula

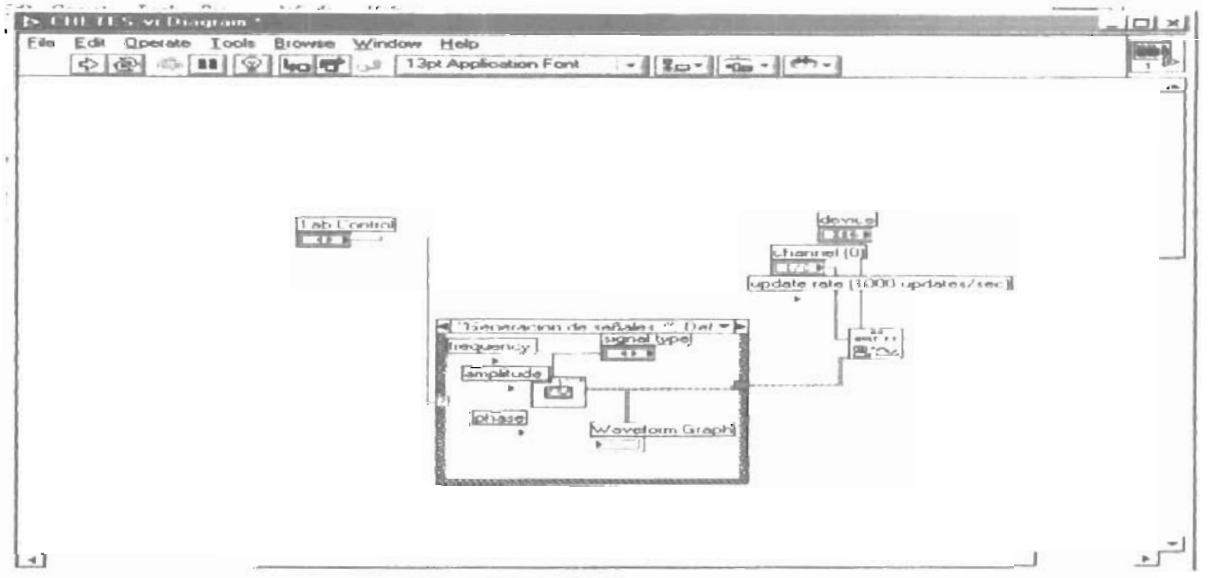


Figura 5.3 Diagrama de bloques. Generación de señales aleatorias por formula

La importancia de la salida de esas ondas se puede utilizar para generar cualquier tipo de onda que queramos generar, así como para saber el comportamiento de una señal solamente conociendo su formula, la salida en el osciloscopio, es al siguiente.

5.2 La adquisición de una señal es más fácil una vez observada la generación de señal. Ahora utilizaremos ***AL adquire waveform***, a la cual le adecuamos el canal por el que deseamos adquirir la señal, así como el dispositivo que estemos usando y el número de ejemplos que vamos a adquirir

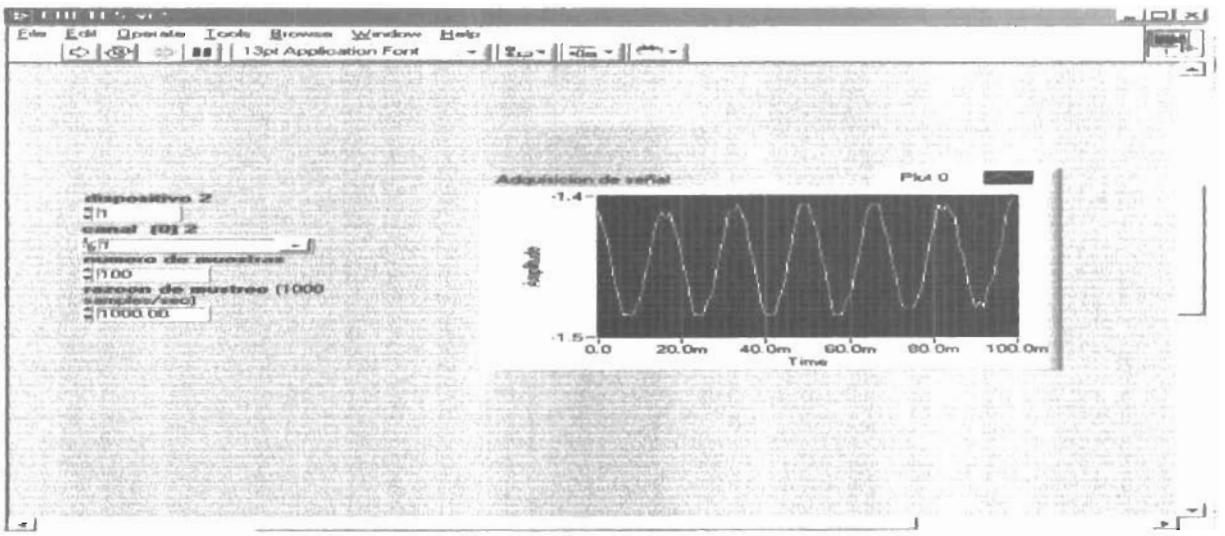


Figura5 4 Panel de control Adquisición de señales.

Así se muestra como una señal es adquirida y se muestra en el panel frontal

5.3 El uso del ejemplo de la transformada de Fourier puede verse a continuación cuando a la señal de entrada la correlación con una señal generada por nosotros después esa misma adquirirla observaremos la respuesta correlacionada, así como también el ejemplo de la FFT así como la respuesta espectral compararlas observar como es al respuesta de la señal una vez que la pasamos por FFT y la respuesta espectral.

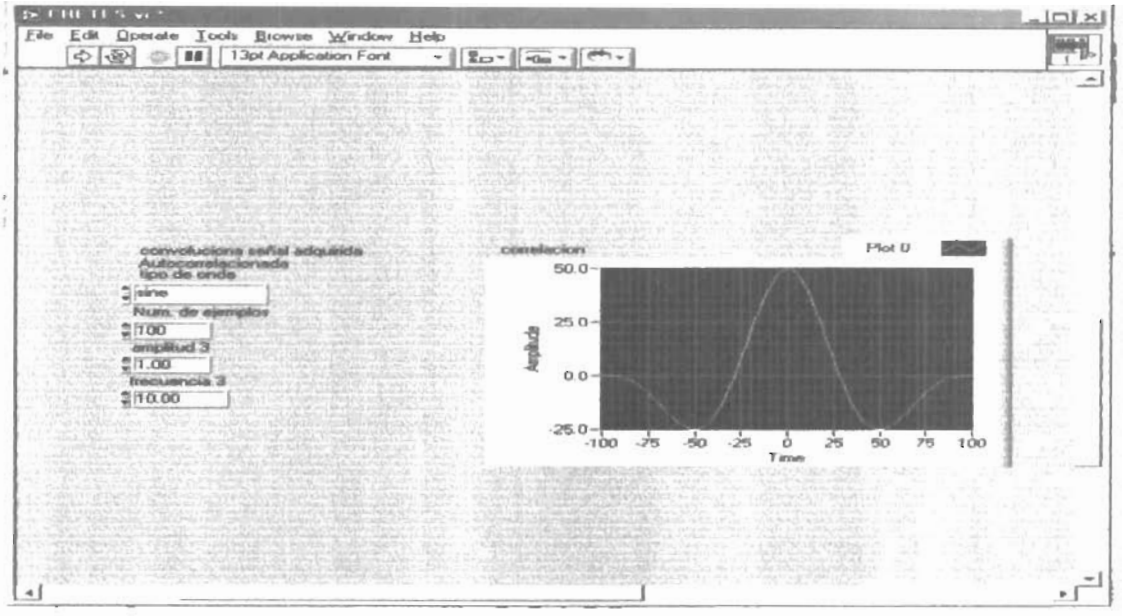


Figura5.5 Panel de control Autocorrelacion

Aqui observaremos que una señal impulso correlacionadas es igual a 1

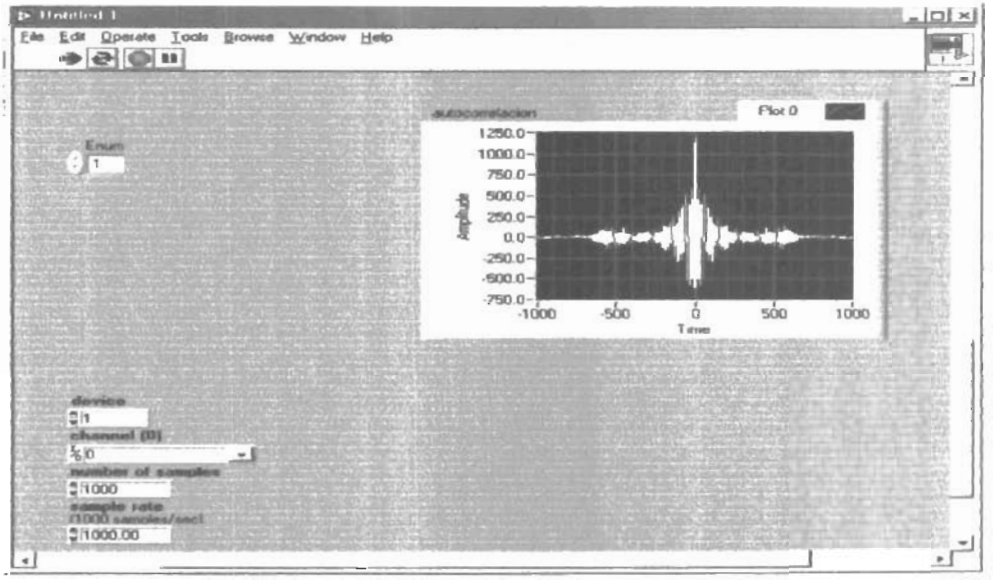


Figura5.6 Autocorrelacion Señal adquirida

A continuación compararemos la Transformada Rápida de Fourier con la respuesta espectral. Con el ejemplo de señal adquirida que trabajos anteriormente nos muestra la FFT y la respuesta espectral a la señal.

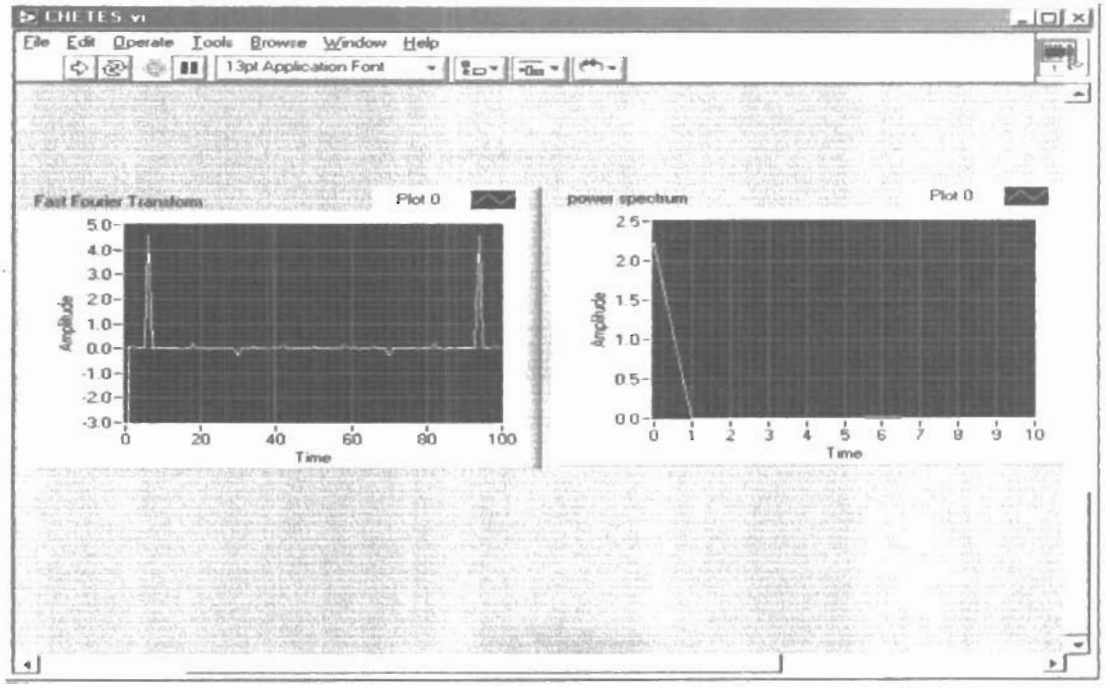


Figura5.7 Panel de control Transformada Rápida de Fourier y Respuesta espectral

5.4 En esta parte se observara la señal la cual generamos, adquirimos y procesamos en tan diferentes maneras ahora pasarla por un filtro diseñado por nosotros en el subprograma Generación de FIR con ventaneo nos ayuda para este ejemplo a ver el desempeño de los filtros lo importante es observar la señal y las respuestas en las siguientes figuras.

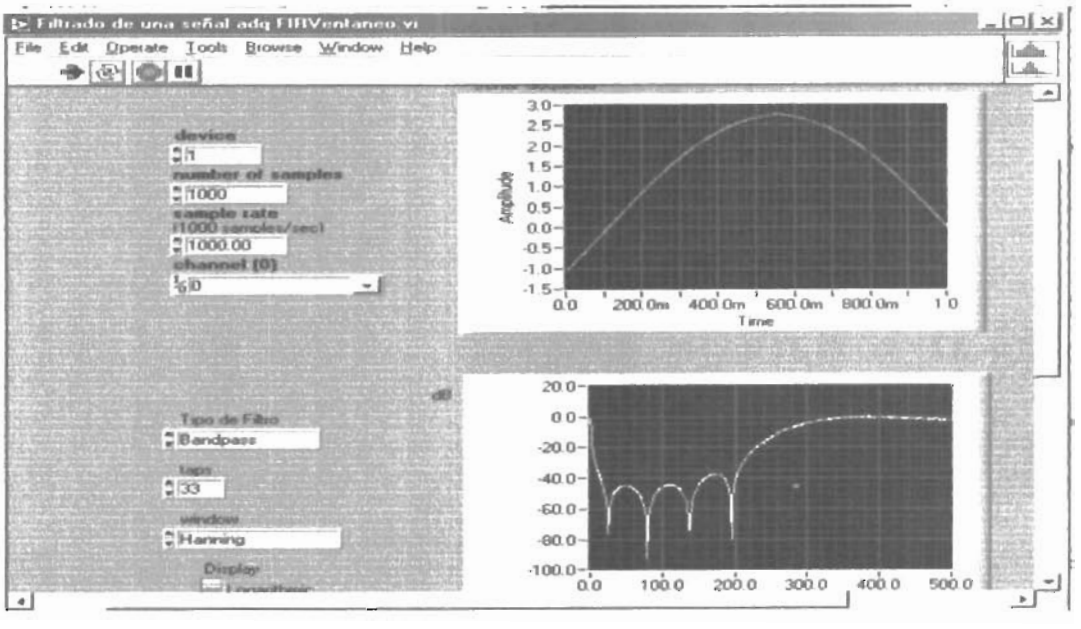


Figura5.8 Panel de control Diseño de filtros

5.5 Control PID. Sea el circuito que consiste de la fuente de voltaje a regular y el motor como se muestra a continuación:

Figura 5.7. Diagrama electrico del motor y su convención mecánica

La función de transferencia del sistema es:

$$\frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{JLs^2 + (RJ + Lb)s + bR + K^2} \qquad (1)$$

$$\frac{\Theta(s)}{V(s)} = \frac{K}{s[JLs^2 + (RJ + Lb)s + bR + K^2]} \qquad (2)$$

donde
J es el momento de inercia del rotor (Kg. m^2/s^2)
b es el factor de amortiguamiento del sistema (carga) (CMS)
K es la constante de la fuerza electromotriz (N m/Amp)
R es la resistencia eléctrica (ohms)
L es la inductancia eléctrica (henrios)
V es el voltaje de entrada (volts)
Θ es el ángulo de salida (radianes)
y $\omega = \dot{\Theta}$ es la velocidad angular (rad/seg).

La ecuación (2) se puede representar en forma de ecuaciones de estado. Si escogemos la posición del motor, la velocidad del motor y la corriente en el motor como variables de estado, las ecuaciones de estado son:

$$\frac{d}{dt}\begin{pmatrix}\Theta\\ \dot{\Theta}\\ i\end{pmatrix} = \begin{pmatrix}0 & 1 & 0\\ 0 & -\frac{b}{J} & \frac{K}{J}\\ 0 & -\frac{K}{L} & -\frac{R}{L}\end{pmatrix}\begin{pmatrix}\Theta\\ \dot{\Theta}\\ i\end{pmatrix} + \begin{bmatrix}0\\ 0\\ \frac{1}{L}\end{bmatrix} V$$

$$y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}\Theta\\ \dot{\Theta}\\ i\end{bmatrix}$$

Nos interesa controlar la velocidad angular ω entonando el voltaje de entrada V(t), o bien controlar la posición angular Θ entonando el voltaje de entrada V(t). Supongamos que tenemos los siguientes valores para J, b, K, R y L:

J=0.01; =0.1; K=0.01; R=1;L=0.5;
Para la Ec. 1, la respuesta al escalón unitario (entrada de 1 Volts) es:
num=K;
den=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2)];
step(num,den,0:0.001:0.2). Es resultado es:

Figura 5.9 Respuesta de la Ec. 1, para un voltaje de entrada escalon unitario

Para la Ec. 2, la respuesta al escalón unitario (entrada de 1 Volts) es.

num=K,
den=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2) 0],
step(num,den,0:0.001:0.2)

El resultado seria:

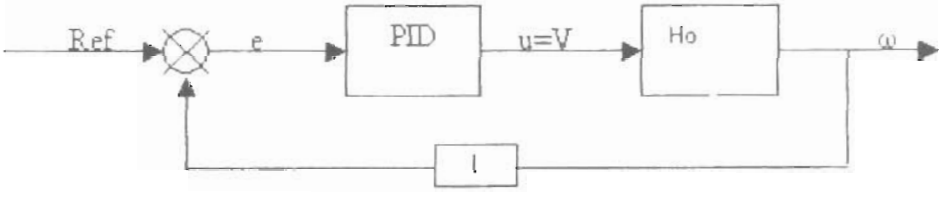
Figura 5.10 Respuesta de la Ec. 2, para un voltaje de entrada escalon unitario

Vemos que en la Fig. 5.9 se tiene que la velocidad de 48 rad/seg se obtiene con un escalón unitario (V = u(t) volts) en un tiempo de 0.18 seg; mientras que para la posición angular es de 8 radianes, como se muestra en la Fig. 5.10. Esto seria satisfactorio si de antemano no se

imponen restricciones de amortiguamiento al sistema y de tiempos requeridos. Sin embargo, pudiera suceder que se desee una velocidad angular determinada y tiempos de posición angular estrictamente en un tiempo preestablecido. De modo que se requiere un algoritmo de control PID que sea capaz de predecir, a partir de los valores J,b,R, L y K, tales que optimicen las Ecs. 1 o Ec. 2. En lo que sigue, presentamos el modo de optimizar las constantes PID para la ecuación 1 o 2. Una vez logrado esto diseñaremos un IV con lo obtenido en la presente modelación con Matlab

1) **Control de Velocidad**.

Supongamos que deseamos entonar la velocidad a una referencia; digamos a una frecuencia de rotación $2\pi f_0$. Para esto, la planta + el controlador (en este caso PID) será el siguiente:



La respuesta en frecuencia del controlador es:

$$K_P + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_P s + K_I}{s};$$

y por lo tanto la respuesta en frecuencia total será:

$$\frac{\Theta(s)}{V(s)}_{PID} = \frac{K_D s^2 + K_P s + K_I}{s} \qquad \frac{K}{JLs^2 + (RJ + Lb)s + bR + K^2}$$

Como ejemplo, supongamos que deseamos tener una velocidad final de 1 rad/seg, con un tiempo final de alcance de 2 segundos, un saturación (overshoot) de no más de 5% y que el error final de estado estacionario menor del 1%. De nuevo para los valores anteriores y para una constante proporcional $K_P = 100$, tenemos que:

J=0.01; b=0.1; K=0.01; R=1,L=0.5, Kp=100, num=K,

den=[(J*L) ((J*R)+(L*b)) ((b*R)+K.^2)].

```
numf=Kp*num,
denf=den;
[numfc,denfc]=cloop(numf,denf),
t=0:0.01:5;
step(numfc,denfc,t)
title('Control Proporcional')
```

El resultado es como sigue:

Figura 5.11 Respuesta a la velocidad angular para un impulso unitario con control proporcional

De la figura anterior vemos que efectivamente hemos logrado que en menos de 2 segundos se obtenga la velocidad de 1 rad/seg, pero el pico inicial (que ocurre alrededor de 0.3 seg) es mayor del 5%. Por lo tanto debemos entonar las otras dos constantes K_D y K_I. Hagamos primero a éstos últimos iguales a 1. Tenemos entonces:

```
J=0.01;b=0.1;K=0.01;R=1;L=0.5;num=K;
den=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2)];
Kp=100;Ki=1;Kd=1;
numcontrol=[Kd, Kp, Ki]; dencontrol=[1 0];
numf=conv(num,numcontrol);
denf=conv(den,dencontrol);
[numfcontrol,denfcontrol]=cloop(numf,denf);
%También se puede usar: [numfc,denfc]=feedback(numf,denf,1,1,-1);
step(numfc,denfc)
title('Control PID con Ki, Kd =1');
```

El resultado es:

Figura 5.12 Respuesta a la velocidad angular para un impulso unitario con control proporcional, integral y derivativo y

En la gráfica anterior se ha eliminado el pico; sin embargo, se tiene que el estado estacionario se obtiene después de 150 segundos. Es necesario encontrar el valor óptimo para las constantes PID. Sean Ki =200,0 Kd=10 y Kp =100

```
J=0.01;b=0.1,K=0.01;R=1,L=0.5;
SP=1,
num=K,
den=[(J*L) ((J*R)+(L*b)) ((b*R)+K^2)],
Kp=100,Ki=200;Kd=10,
numcontrol=[Kd, Kp, Ki], dencontrol=[1 0],
numf=conv(num,numcontrol), denf=conv(den,dencontrol),
[numfcontrol,denfcontrol]=feedback(numf,denf,1,1,-1),
ans=step(SP*numfcontrol,denfcontrol)
title('Control PID con Kp=100, Ki=200, Kd =10')
xlabel('Tiempo, seg'),
ylabel('Velocidad angular (rad/seg)')
axis auto
t=0:0.01:1.19
ans1=ans , plot(t,ans1)
```

El resultado es:

Figura 5.13 Respuesta a la velocidad angular para un impulso unitario con control proporcional, integral y derivativo que cumplen con las restricciones dadas en el diseño. Ver texto para más detalles

Vemos que finalmente el resultado es satisfactorio en relación a las condiciones dinámicas establecidas. Una vez que se ha optimizado las constantes PID, se procede a realizar el instrumento virtual. A continuación se muestra el IV para el control con las constantes PID encontradas.

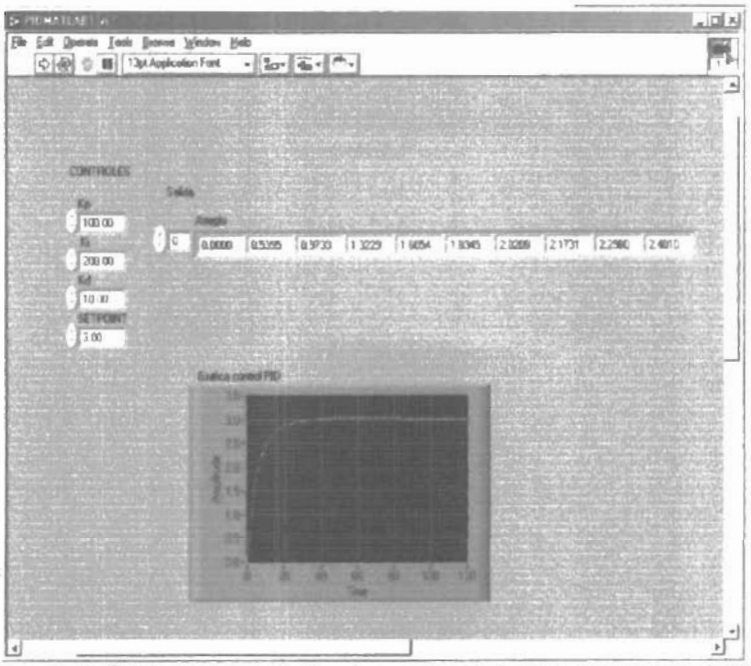


Figura 5.14. Salida de el control del pid con setpoint =3. como instrumento virtual

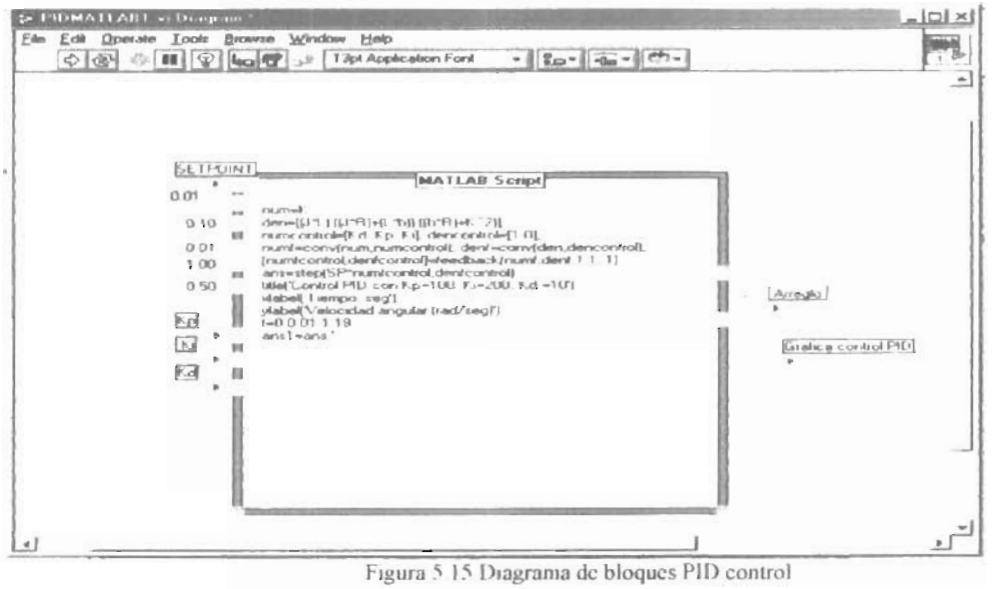


Figura 5.15 Diagrama de bloques PID control

Aquí ahora podemos modificar las diferentes variables constantes del PID que son los valores de Kp Ki o Kd para ver y comprender este punto modificándolos. Con un Kp = 20 vemos el desempeño que es el siguiente

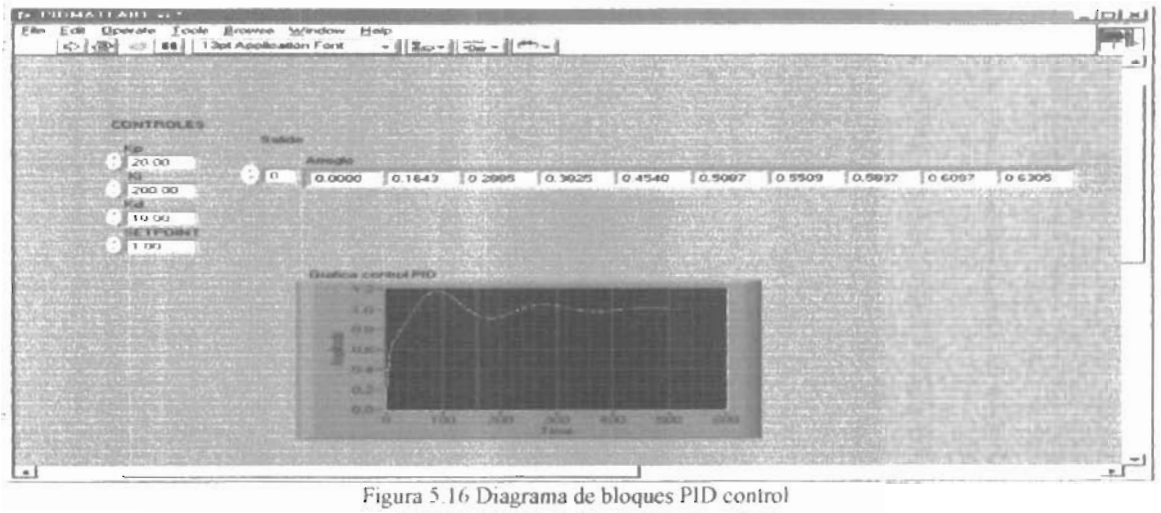


Figura 5.16 Diagrama de bloques PID control

Observamos que el overshoot al inicio es muy grande y rebasa las especificaciones que le dimos al inicio y el tiempo es de 6.0s para que se estabilice en el tiempo y referencia deseados.

Conclusiones

En el presente trabajo de se ha mostrado un proceso básico y útil para poder introducir a aquellos que deseen adentrarse en la instrumentación virtual (IV), basada en LabView, como una herramienta adicional en su currícula. Se asume que el lector cuenta con un entrenamiento básico en sistemas lineales, instrumentación electrónica y nociones en programación con Matlab. Lo anterior es necesario debido a que la gran mayoría de los algoritmos de control implementados con IV, se basan formalmente en la modelación de sistemas analógicos y digitales con herramientas computacionales. Esto definitivamente provee una sólida base para poder plantear e implementar módulos virtuales, más allá que el simple proceso de conexión de íconos, los cuales corren el riesgo de ser ineficientes y con falta de robustez en caso de que el diseñador no comprenda detalladamente el proceso involucrado en la finalidad de su aplicación solucionada con IV.

Asimismo, la cada vez creciente necesidad de aplicar el procesamiento digital de señales en sistemas de IV hace imperativo dominar la teoría de filtros digitales. Mas aún, si con herramientas computacionales de análisis como Matlab permiten implementarse con IV, hacen mucho más poderoso el proceso de optimización en las etapas del diseño, desde la modelación de algoritmos de control PID, lógica difusa, filtros IIR de fase linealizada introducidos recientemente, hasta la introducción de las tecnologías de comunicación ópticas e inalámbricas más recientes; todas ellas a ser integradas en instrumentos virtuales. Se provee al lector interesado con las referencias 5 a 10 citadas en la bibliografía dada al final de esta Tesis.

Bibliografía

1. Ver, por ejemplo, ***LabView 6i Adds Internet Features to Data Acquisition***, **por** Donald L. Shirer, Computing in Science & Engineering--July 2001,Volume 3, Issue 4, pp. 8-11.

2. http:\\www.ni.com

3. **LabView Signal Processing, M.L. Chugani, A.R. Samant, y M. Cerna, National Instruments, Virtual Instrumentation Series, Prentice Hall., 1998.**

4. Ver, por ejemplo, "*Real Time Digital Signal Processing in the Undergraduated Curriculum*", IEEE Transactions on Education, Vol. 46, pp. 95 (2003)

5. Design of Chebyshev-type IIR filters with approximately linear phase characteristics", Electronics and Communications in Japan (Part III: Fundamental Electronic Science). Volume 87, Issue 2 , pp. 1 – 9 (2004).

6. **LabView Signal Processing**, M.L. Chugani, A.R. Samant, and M. Cerna, National Instruments, Narional Instrumentation Series. Prentice Hall. 1998

7 ***Modeling and Simulation of Oscillator Based Random Number Generators*** *by Craig S. Petrie and J. Alvin Connelly, School of Electrical and Computer Engineering, Georgia Institute of Technology.*

8. **An Automated System for Frequency Response Analysis With Application to an Undergraduate Laboratory of Electrical Machines**, F. S. Sellschopp , Marco A. Arjona, a ser publicado en IEEE Transactions on Education (2004).

9. **The SIVA Demonstration Gallery for signal, image, and video processing education**, Rajashekar, U. Panayi, G.C. Baumgartner, F.P. Bovik, A.C. Dept. of Electr. & Comput. Eng., Texas Univ., Austin, TX, USA IEEE Transactions on Education, 2002.

10. **The future of electrical and computer engineering education.** Berry, F.C. DiPiazza, P.S. Sauer, S.L. , Rose-Hulman; IEEE Transactions on Education Vol. 467 - 476, Volumen 46(4) (2003).