

INSTRUMENTACIÓN VIRTUAL PARA LA ADQUISICIÓN, GENERACIÓN Y ANÁLISIS DE SEÑALES

Iñigo J. Oleagordia Aguirre¹, José Ignacio San Martín Díaz², José Javier San Martín Díaz³

¹Universidad del País Vasco UPV/EHU. <jtpolagi@lg.ehu.es>

²Universidad del País Vasco UPV/EHU. <iepsadij@sb.ehu.es>

³Universidad del País Vasco UPV/EHU. <iepsmdij@sb.ehu.es>

RESUMEN

En la presente comunicación se realiza una descripción constructiva y funcional de un instrumento virtual (VI) diseñado e implementado en el entorno LabVIEW 6.1 que sirve como instrumentación complementaria a la tradicional en el ámbito de la docencia universitaria fundamentalmente. El instrumento virtual es capaz de captar señales desde tres fuentes distintas como son: mediante una tarjeta de adquisición de datos (DAQ), a través de un fichero, o bien empleando un generador interno de señales y que forma parte del mismo instrumento virtual. Una vez captada la información se procede a su procesamiento empleando diversos algoritmos matemáticos como cálculos estadísticos, filtrado digital y análisis espectral. El resultado de estas operaciones se representa gráficamente a la vez que se le proporciona al usuario la posibilidad de almacenarla en un fichero. A lo largo de la descripción se muestran algunos resultados obtenidos para poner de manifiesto la funcionalidad y capacidad del sistema desarrollado.

1. INTRODUCCIÓN

Actualmente la utilización de ordenadores ha evolucionado de tal forma que se han hecho fundamentales, si no imprescindibles, dentro de la estructura curricular de cualquier disciplina tecnológica tanto en el ámbito universitario, docencia e investigación, como en la industria. De hecho, diferentes secciones de la industria, como son las cadenas de producción, las comunicaciones y entre ellas los buses industriales, el transporte, los laboratorios de investigación y los sistemas de test y medida, entre otros, dependen en gran medida del soporte del ordenador. El desarrollo tecnológico a todos los niveles y especialmente en el campo de la electrónica implica la disponibilidad y empleo de instrumentación más específica y competitiva. Hoy en día una característica primordial es la capacidad de integración en sistemas de test y medida basados en ordenador y la flexibilidad para adaptarse fácilmente a los cambios impuestos por la metodología seguida. En este contexto la instrumentación virtual se presenta como una estrategia que asume satisfactoriamente los retos mencionados puesto que entre sus características más notables cabe citar: I) está definido por el usuario, II) es un sistema orientado a una aplicación y conectividad a una amplia gama de redes periféricas y otras aplicaciones, III) el software es la clave, minimizando los costes de mantenimiento y desarrollo. De hecho, un instrumento virtual es un módulo software, IV) es un sistema abierto, de funcionalidad flexible que emplea una tecnología evolutiva, V) se adapta rápidamente a los cambios tecnológicos.

Cualquier instrumento virtual implementado en LabVIEW posee dos partes: la primera corresponde al panel frontal que es la pantalla mediante la cual el usuario interactúa con la aplicación, y la segunda corresponde al diagrama de bloques que es el código fuente, en lenguaje G, donde se programa la aplicación.

En este trabajo se presenta el desarrollo de un prototipo de instrumento virtual de bajo coste multifunción orientado fundamentalmente a la docencia universitaria, complementando a la instrumentación tradicional, aunque es perfectamente posible su adaptación a otros ámbitos del campo industrial o de investigación. El software se ha realizado mediante programación modular, desarrollándose los siguientes bloques funcionales:

- **Módulo de adquisición de datos:** permite seleccionar la fuente de adquisición de las señales eléctricas que puede provenir de:
 - a) Una tarjeta de adquisición de datos ADQ, concretamente la PCI-6024E, en cuyo caso se configuran los canales adquisición.
 - b) Los datos almacenados en un fichero y que han sido registrados con anterioridad para su posterior análisis y representación.
 - c) Un generador interno de señales senoidal, triangular, cuadrada, diente de sierra, etc.
- **Módulo de procesamiento:** en el dominio del tiempo, de la frecuencia y estadístico.
- **Módulo de filtrado digital:** donde se realiza el filtrado empleando distintos tipos y topologías de filtros como FIR, IIR (Butterworth, Chebyshev, Bessel, ...). En este módulo también es posible elegir distintos tipos de ventanas como triangular, de Kaiser, Blackman, etc.
- **Módulo de simulación:** en el cual se puede representar el diagrama de Bode de diversas funciones de transferencia.

2. METODOLOGÍA

El desarrollo de la aplicación se ha realizado con un método “top-down” ya que es el más adecuado a las características de LabVIEW puesto que el lenguaje G puede enmarcarse dentro de los lenguajes procedurales además de ser un lenguaje concurrente. Entendiendo que un método estructurado “top-down” se basa en la subdivisión de programas complejos en subprogramas sencillos, con esta técnica cualquier aplicación por compleja que resulte se estructura en un árbol jerárquico de procedimientos sencillos que en LabVIEW se implementan mediante subVIs. En la figura 1 se muestra sólo una parte del árbol jerárquico ya que en una sola ventana no se puede visualizar toda la jerarquía debida a su extensión. Este módulo está compuesto de 129 Vis. Externamente, los VIs se almacenan agrupados temáticamente en 3 librerías distintas.

Este tipo de desarrollo lleva un coste relativamente alto en tiempo ya que se invierte un gran esfuerzo en la parte de diseño previa a la implementación, pero esta última, con herramientas adecuadas es mucho más sencilla en comparación con otros entornos. En este contexto los aspectos más relevantes en el desarrollo de la aplicación han sido: planificación y depuración (debugging), rendimiento, compartición de ficheros, y documentación.

2.1. Planificación y Depuración

La labor se centra en encontrar lo que se busca. Las herramientas empleadas se basan en la estructura jerárquica (Browse) y en el propio debugger de LabVIEW que mediante los puntos de prueba y ruptura y ejecución paso a paso permiten optimizar la funcionalidad de un subVI. También es de gran utilidad el comando de búsqueda (Find). Dentro de la planificación se ha prestado especial atención a la reutilización del software.

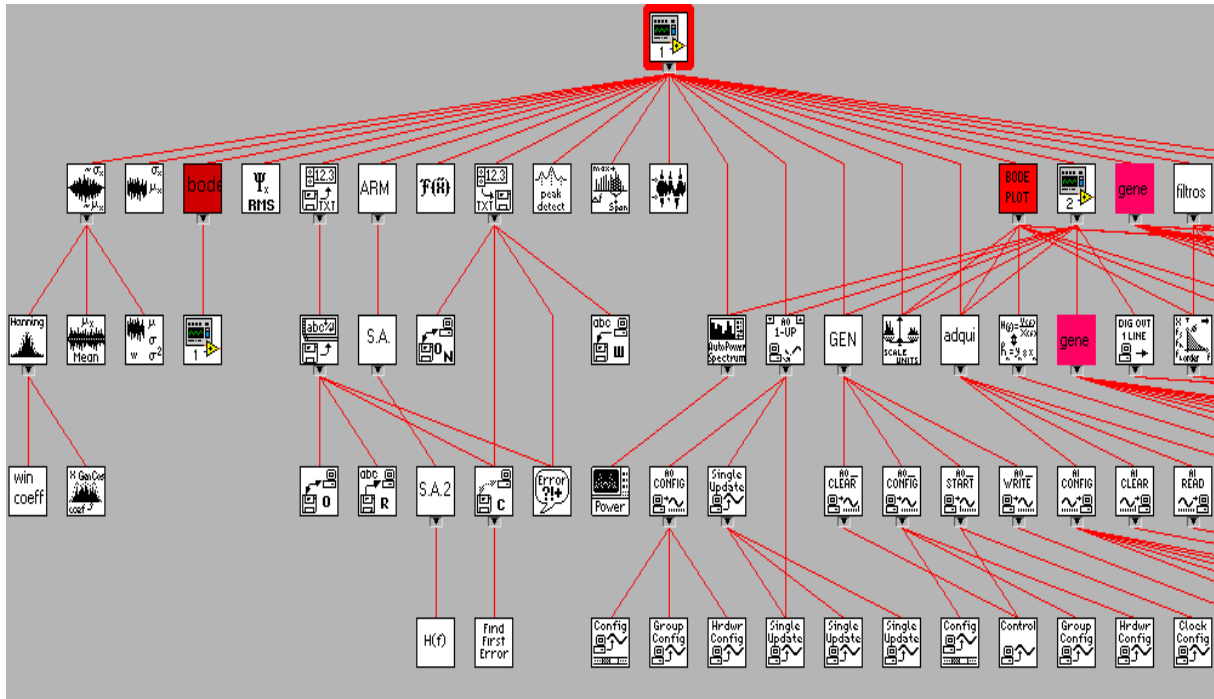


Fig.1. Estructura jerárquica parcial

2.2. Rendimiento

Los puntos más significativos son: la dificultad de análisis, la memoria utilizada y el tiempo empleado en cargar la aplicación. Las herramientas empleadas son el Profiler y VI Control VIs. Mediante el Profiler se puede analizar el tiempo de ejecución y la memoria que ocupa un determinado VI. Con esta información se optimiza la programación pues se determina perfectamente cuales con los subVIs que consumen más tiempo y memoria. Con la herramienta VI Control VIs se puede cargar un VIs de forma dinámica, es decir sólo cuando sea necesario. Así mismo se puede descargar dinámicamente un VI, diferir la carga de un VI hasta que se haya lanzado la aplicación así como modificar el tamaño y posición de los paneles frontales.

2.3. Documentación

En grandes proyectos la documentación de los mismos juega un relevante papel en el mantenimiento y comprensión de los mismos. Además de la documentación escrita se adjunta ficheros de ayuda “online” sobre todo para describir la funcionalidad de diversos VIs. Se han creado ficheros de ayuda tipo Windows a los que se puede acceder desde cada una de las pantallas. Para ello se emplea la función de LabVIEW “Control Online Help”. La herramienta SCC (Source Code Control) es de gran ayuda en la descripción y gestión del proyecto ya que describe la evolución del mismo facilitando el trabajo en equipo.

Por último, paralelamente a la fuerte estructuración de procedimientos y datos, la trazabilidad es otra de las características que aportan calidad al software, entendiendo que la trazabilidad entre las fases de análisis, diseño e implementación implica que los elementos que aparecen en el análisis tienen sus homólogos en el diseño y estos a su vez en la implementación.

3. DESARROLLO DEL SOFTWARE

En este apartado se muestra las principales características de software creado. No se pretende dar una explicación exhaustiva del programa sino simplemente mostrar sus principales posibilidades. Para que el instrumento virtual sea operativo se ejecuta la aplicación ADGEASE_1.VI apareciendo la correspondiente pantalla de trabajo. La configuración de las distintas opciones se realiza actuando sobre los correspondientes controles en el panel frontal. En la figura 2 se muestra la pantalla de trabajo junto a la interface de usuario.

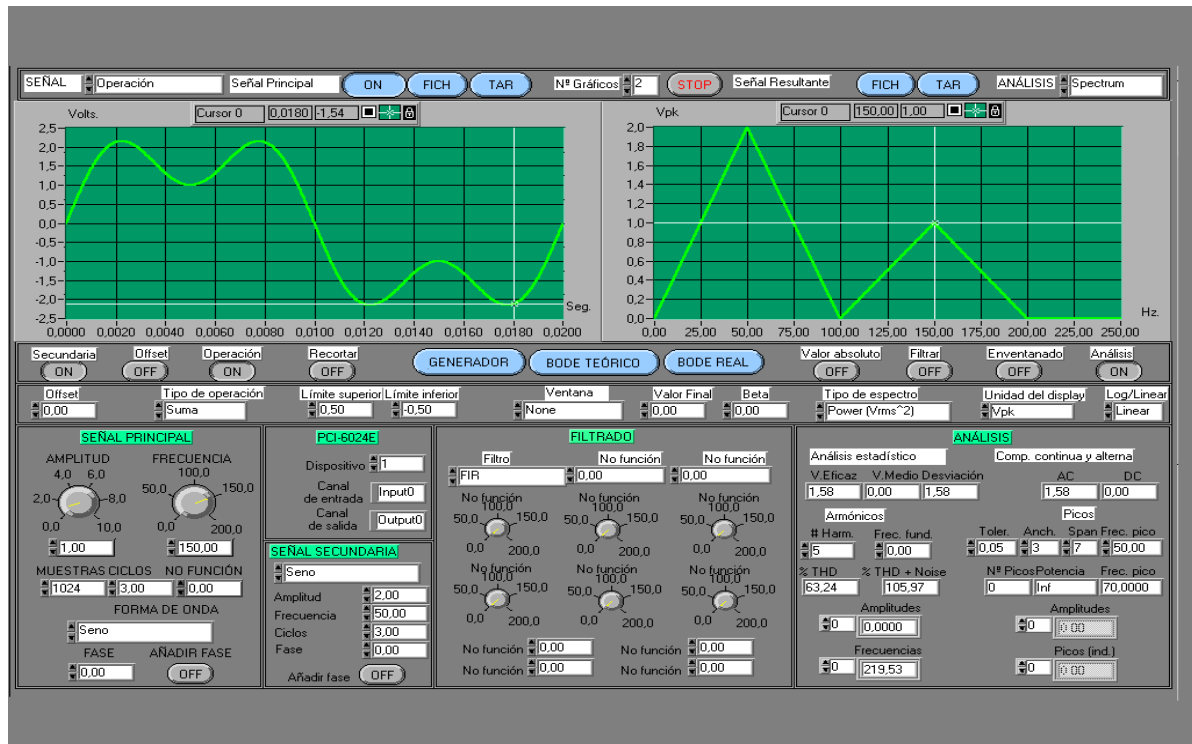


Fig.2. Pantalla de trabajo e interface de usuario del instrumento virtual ADGEASE_1.VI

3.1. Módulo de adquisición

La primera acción a realizar en el VI es la adquisición de la señal que vendrá representada por un número determinado de valores. Como se ha indicado la adquisición de señales se puede efectuar desde tres fuentes distintas e independientes cuya funcionalidad se describe seguidamente:

3.1.1. Tarjeta de adquisición (DAQ)

La tarjeta empleada es la PCI-6024E que dispone de 16 canales de entrada analógicos, 2 canales de salida analógicos y 8 líneas DIO. La frecuencia máxima de muestreo es de 200KHz por canal.

Cuando la aplicación adquiere datos repetidamente y además los almacena en un fichero se incrementa la eficiencia del software si no se abre y cierra el fichero cada vez que se accede a él. La técnica de dejar archivos abiertos entre las operaciones escritas se denomina disk streaming. Por las ventajas que reporta esta técnica es la empleada en esta aplicación. Los ejemplos mostrados a continuación muestran estas ventajas. En el primer ejemplo, figura 3(a) el VI debe abrir y cerrar el archivo durante cada iteración del lazo. El segundo ejemplo, figura 3(b), emplea la técnica disk streaming para reducir el número de veces que el VI debe

interactuar con el sistema de operación para abrir y cerrar el fichero. Abriendo el archivo una vez antes de que comience el lazo, y cerrándolo después de que finalice el lazo, se guardan dos operaciones del archivo en cada iteración del lazo.

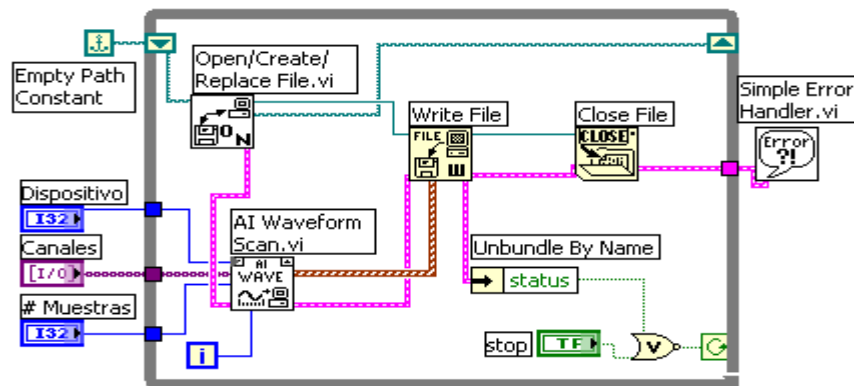


Fig. 3 (a). Adquisición sin optimizar el acceso a disco

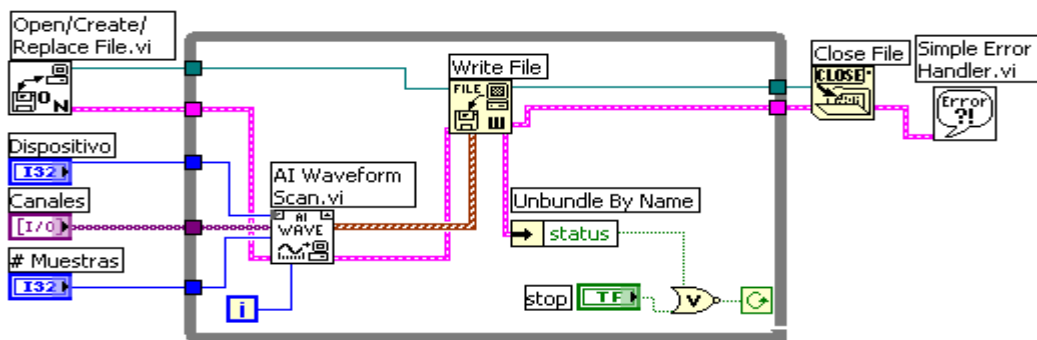


Fig. 3 (b). Adquisición optimizando el acceso a disco. Técnica disk streaming

Tomando como referencia el diagrama de bloques de la figura 3(b) se construye el VI de la figura 4 que permite visualizar los datos que se están adquiriendo a la vez que su almacenamiento en un fichero para su posterior análisis y representación.

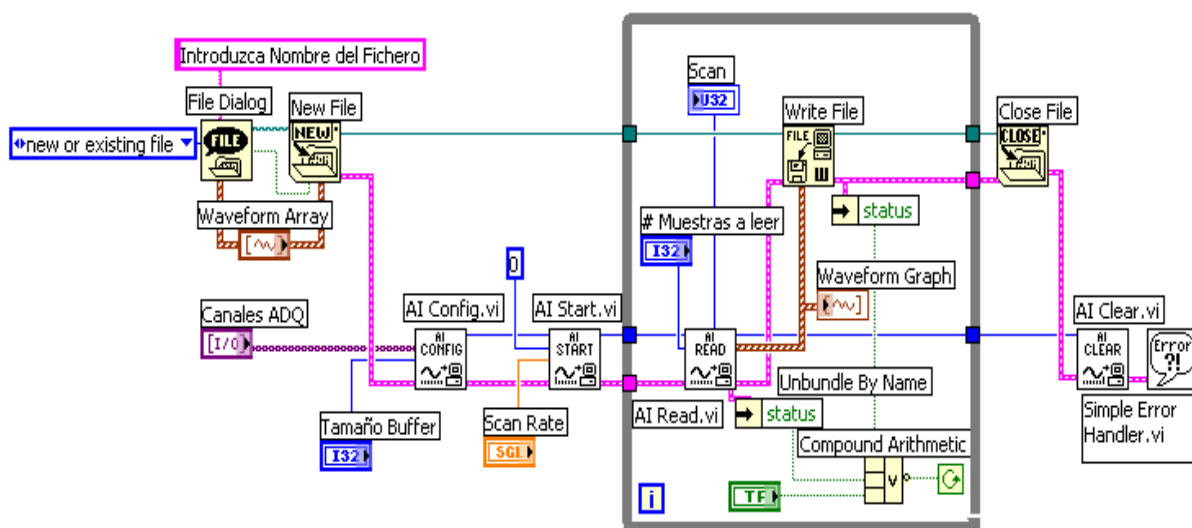


Fig. 4. Adquisición de señales y almacenamiento mediante la técnica disk streaming

Seguidamente se describe de forma resumida la funcionalidad de este Vi. La función File Dialog.Vi muestra en pantalla un cuadro de diálogo donde se especifica el path y nombre del fichero donde se almacenan las muestras capturadas por la tarjeta. La función New File.Vi crea el fichero especificado mientras que Write File.Vi escribe los datos y Close File.Vi cierra el fichero después que los datos se han almacenado. La función AI Config.Vi configura la adquisición de las señales analógicas para los canales especificados. Este Vi también configura el hardware de la tarjeta y define un buffer para la operación de adquisición. En AI Start.Vi comienza la adquisición a la frecuencia muestreo especificada. La función AI Reads.Vi lee los datos del buffer de adquisición que son almacenados mediante Write File.Vi. AI Clear Vi resetea las características funcionales asociadas al proceso de adquisición. La función Unbundle by Name devuelve del cluster el elemento especificado por el nombre.

3.1.2. Adquisición a través de fichero.

En la figura 5 se muestra el diagrama en bloques básico a partir del cual se leen y representan los datos almacenados en un fichero. Al ejecutarse este VI aparece un cuadro de diálogo en el cual se especifica el nombre del fichero cuyo contenido se quiere leer y visualizar.

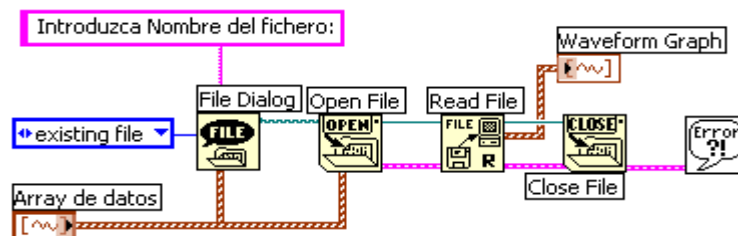


Fig. 5. Diagrama de bloques básico para la adquisición de señales almacenadas en un fichero.

3.1.3. Señal generada internamente.

Básicamente se pueden dividir en dos grupos: señales periódicas (senoidal, cuadrada, diente de sierra, etc.) y señales de test tales como la función impulso o un pulso de amplitud y duración determinada por el usuario. Para las señales periódica se define: la frecuencia, amplitud, número de puntos y fase inicial. Para generar señales mediante software se hace uso de las funciones de librería, tales como Sine Wave.Vi, Triangle Wave.Vi, etc. que para este fin dispone el programa LabVIEW. La señal generada puede sacarse al exterior a través de uno de los canales analógicos de salida de la DAQ para actuar sobre un circuito y analizar posteriormente la salida. En la figura 6 se muestran dos señales generadas por este procedimiento.

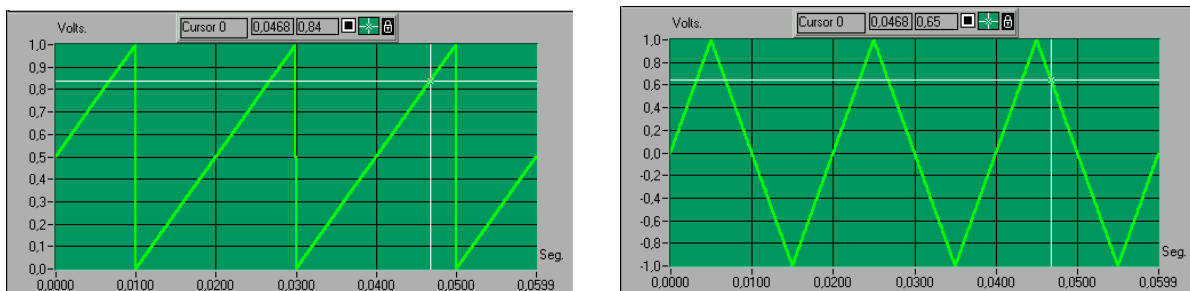


Fig. 6. Señales diente de sierra y triangular generadas internamente por software

Cuando se emplea este procedimiento para generar señales la tarjeta DAQ impone una limitación en la frecuencia de la señal generada ya que solamente puede llegar a las 10.000 muestras por segundo. En la figura 7 se muestra un detalle de la programación correspondiente al generador.

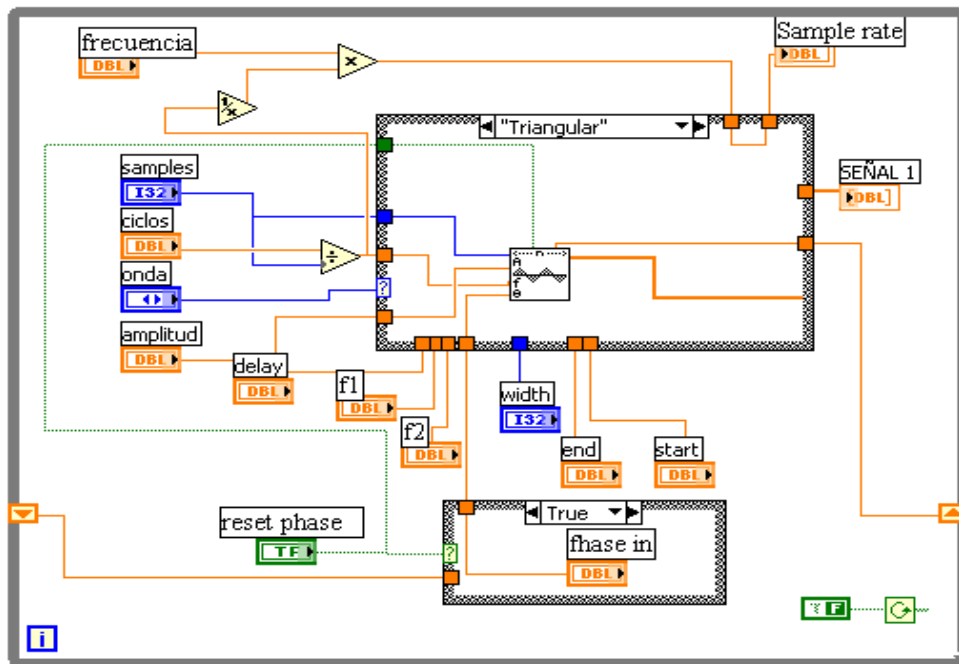


Fig. 6. Generador de señales. Detalle de programación.

3.2. Módulo de procesamiento

Este módulo consta de las siguientes unidades funcionales: unidad algebraica, unidad estadística y unidad de análisis espectral.

3.2.1. Unidad algebraica.

Las operaciones matemáticas a realizar son: suma, resta multiplicación y división entre dos señales que pueden adquirirse a través de la tarjeta, estar almacenadas en ficheros ó generadas internamente por software. El sistema está protegido contra la división entre 0. En la gráfica 7 se muestra la suma de dos señales de frecuencia distinta $f_1 = 50$ Hz y $f_2 = 100$ Hz

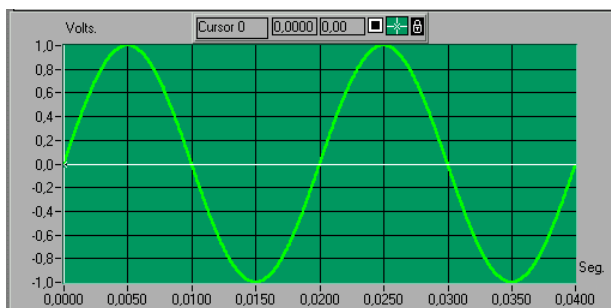


Fig.7(a) Señal senoidal (s1) de 50Hz

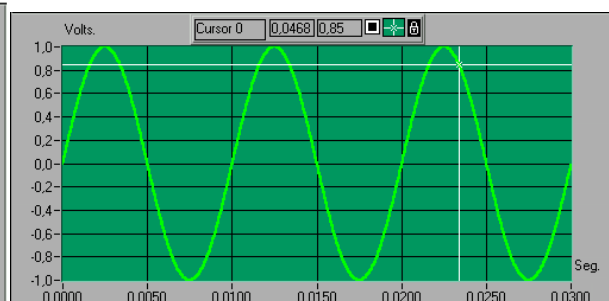


Fig.7(a) Señal senoidal (s2) de 100Hz

3.2.2. Unidad estadística

Aprovechando las funciones de librería de LabVIEW se ha implementado este bloque estadístico que permite calcular prácticamente todas las variables estadísticas empleadas en ingeniería tales como valor medio, varianza, desviación estándar y moda. Dentro de este bloque también se ha incluido el cálculo del valor eficaz (RMS) así como el valor de la componente continua de una señal.

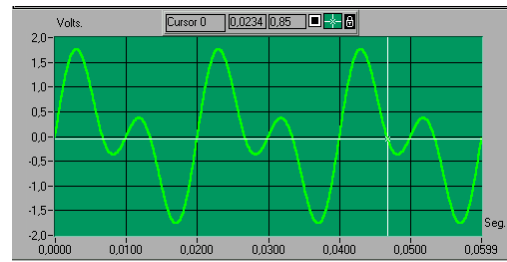


Fig. 7(c) Señal resultante de sumar s1+s2

3.2.3. Unidad de procesamiento

La función básica de este bloque es Real FFT. Vi que calcula la FFT $X[k]$ del conjunto de muestras de entrada $x[n]$ en el dominio del tiempo. La FFT se calcula mediante el algoritmo decimation in time si el número de datos $x[n]$ es potencia de dos, de lo contrario calcula la DFT. En la figura 8 se muestra el núcleo de esta unidad. El conjunto de señales es generada mediante el subVI GEN_1.Vi de programación muy similar al de la figura 6.

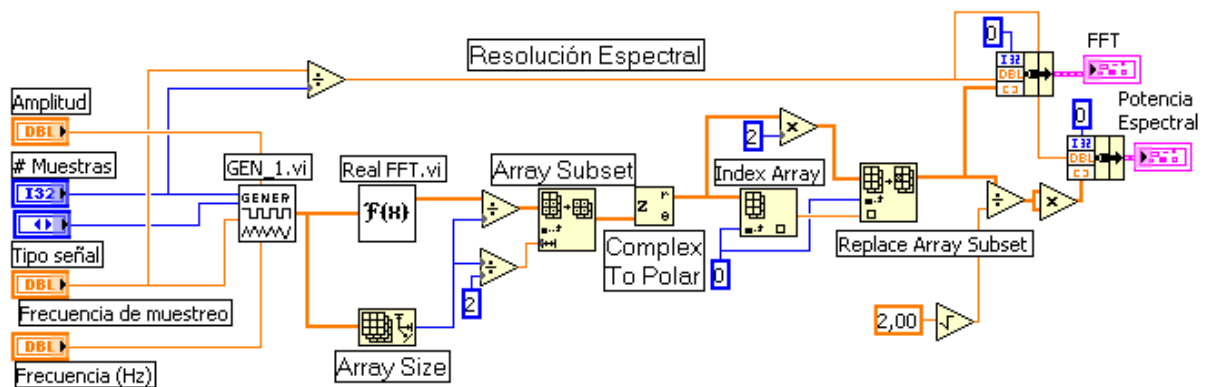


Fig. 8. Núcleo básico de la unidad de análisis espectral

La FFT, de un conjunto de N muestras $x[n]$ en el dominio del tiempo, se define por la ecuación:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j \cdot \frac{2 \cdot \pi}{N} \cdot k \cdot n} ; 0 \leq k \leq N-1 \quad [1]$$

que separadas en su parte real e imaginaria queda:

$$R(X[k]) = \sum_{n=0}^{N-1} x[n] \cdot \cos\left(\frac{2 \cdot \pi}{N} \cdot k \cdot n\right) ; I(X[k]) = - \sum_{n=0}^{N-1} x[n] \cdot \text{sen}\left(\frac{2 \cdot \pi}{N} \cdot k \cdot n\right) \quad [2]$$

Puesto que la salida de Real FFT .Vi es un array complejo con las partes reales $R(X[k])$ e imaginaria $I(X[k])$ de la FFT, la función Complex to Polar.Vi transforma en el formato módulo argumento. Si la señal es muestreada a la frecuencia f_s el intervalo de tiempo entre

dos muestras consecutivas es $\Delta t = \frac{1}{f_s}$ que en el dominio de la frecuencia corresponde a un espaciado $\Delta f = \frac{f_s}{N} = \frac{1}{N \cdot \Delta t}$ denominado resolución espectral. En la figura 8 se aprecia este cálculo al dividir la frecuencia de muestreo entre el número de muestras. En la figura 9 se muestra la gráfica resultante de aplicar este análisis en el dominio de la frecuencia a una señal cuadrada de 10 Hz de frecuencia.

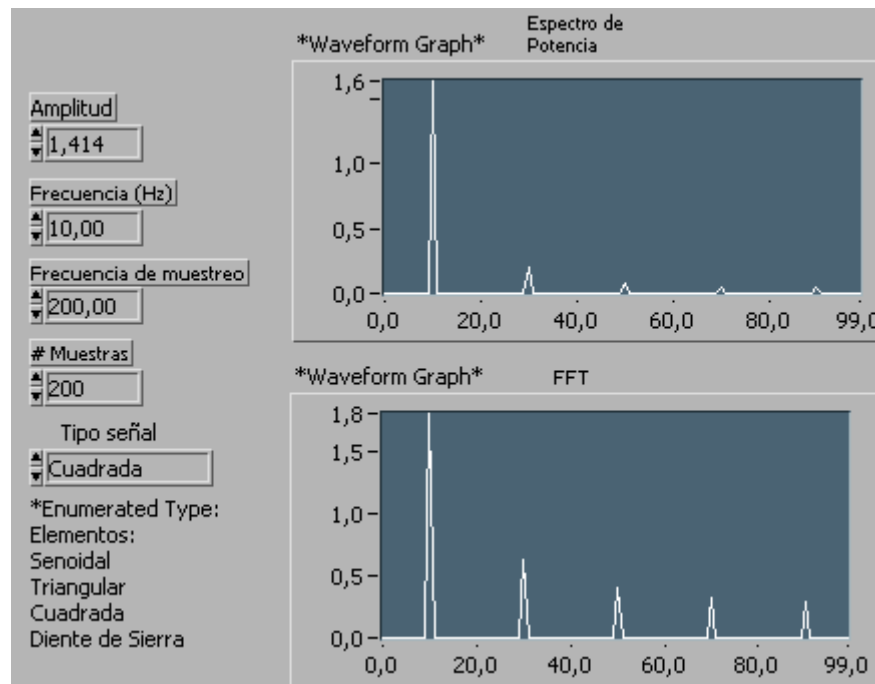


Fig. 9. Gráfica correspondiente al análisis espectral.

En la pantalla de trabajo mostrada en la figura 2, en la parte superior, se aprecia el resultado de este análisis aplicado a una señal compuesta de otras dos señales senoidales de 50 y 150 Hz respectivamente. En el diagrama de la figura 9 también se calcula la potencia de $x(t)$ en el dominio de la frecuencia calculando la potencia de cada componente frecuencial $X(k)$. Una vez hecho esto se suman estas potencias para obtener la potencia total P . El conjunto de las potencias de las componentes $\left(\frac{X(k)}{\sqrt{2}}\right)^2 = \frac{|X(k)|^2}{2}$ en función de la frecuencia se denomina espectro de potencia de $x(t)$.

3.3. Módulo de filtrado digital

Podemos considerar la acción de filtrado como aquella mediante la cual se modifica el contenido frecuencial de una señal. El objetivo básico en este módulo es analizar y comprobar la funcionalidad tanto de los filtros digitales con arquitectura FIR (Finite Impulse Response) modelizados por la ecuación en diferencias finitas [3], como de los filtros IIR (Infinite Impulse Response) definidos por la ecuación [4], ambas en el dominio del tiempo.

$$y[n] = \sum_{k=0}^N b_k \cdot x[n-k] \quad [3] \quad ; \quad \sum_{i=0}^N a_i \cdot y[n-i] = \sum_{k=0}^M c_k \cdot x[n-k] \quad [4]$$

Las muestras $x[n]$, e $y[n]$ corresponden a la señal de entrada y salida respectivamente. Los coeficientes a_i , b_k , y c_k con características de cada topología y tipo del filtro elegido. En este contexto, LabVIEW posee una amplia librería de funciones para implementar tanto filtros FIR como IIR, por ejemplo las Vis para implementar filtros digitales de Butterworth, Chebyshev, Elípticos y de Cauer y ventana de tipo Hamming, Blackman, Kaiser, etc.

Para comprobar la funcionalidad de estos filtros se realizó el diseño de la figura 10 correspondiente a un filtro Butterworth de paso bajo y de orden variable donde, desde el panel frontal, se puede elegir la frecuencia de corte f_L y el orden. La señal de entrada $x[n]$ está formada por una senoidal $s[n]$ y ruido $r[n]$. Como puede comprobarse en la figura 11 mediante la acción del filtrado se elimina la señal de ruido $r[n]$, también puede comprobarse el régimen transitorio en la señal de salida del filtro.

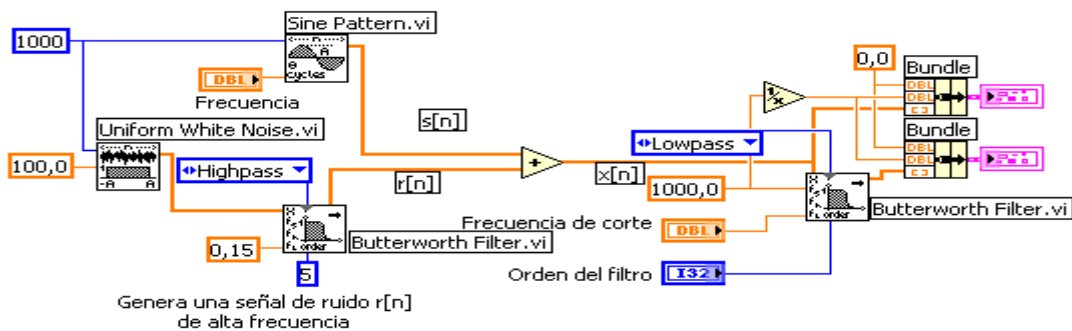


Fig. 10. Implementación de un filtro Butterworth de paso bajo y orden variable

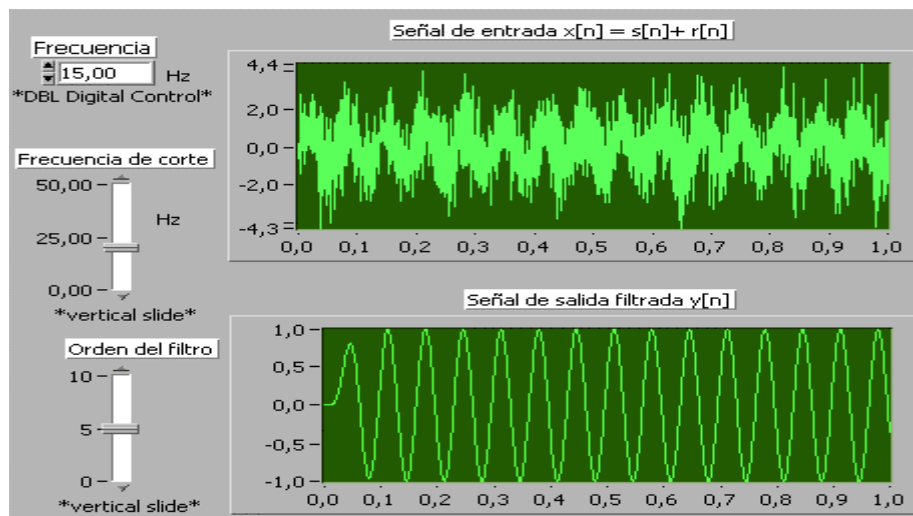


Fig. 11. Representación gráfica de la acción de filtrado

Este VI básico ha sido el punto de partida para la implementación del módulo de filtrado digital. En la figura 12 se muestra un detalle constructivo del mismo donde, como puede observarse la elección del filtro se hace con una estructura Case en la cual mediante los correspondientes controles habilitados por los nodos de propiedad, property nodes, se introducen en el diagrama de bloques los parámetros del filtro seleccionado. En la figura 13 se muestra el panel frontal de esta módulo así como la señal de entrada, que es una senoidal modulada linealmente en frecuencia, la señal filtrada por un filtro Butterworth de paso de banda y la respuesta frecuencial del mismo caracterizada por las gráficas del módulo y la fase de su función de transferencia.

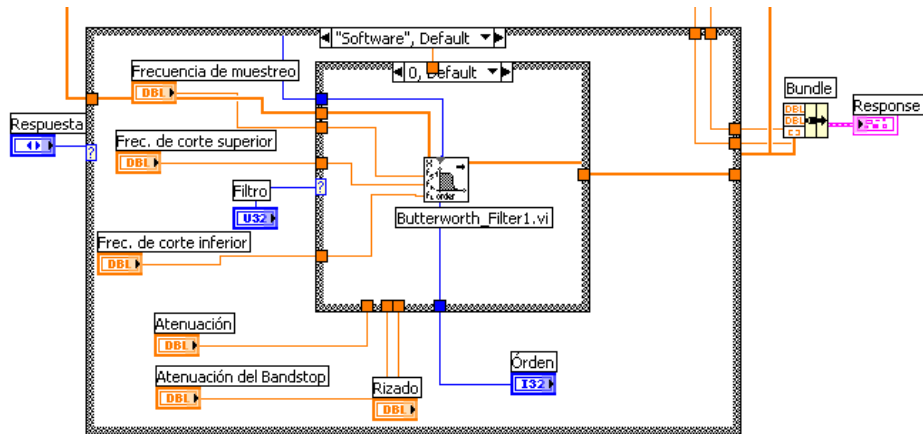


Fig. 12. Detalle de la programación del módulo de filtrado digital

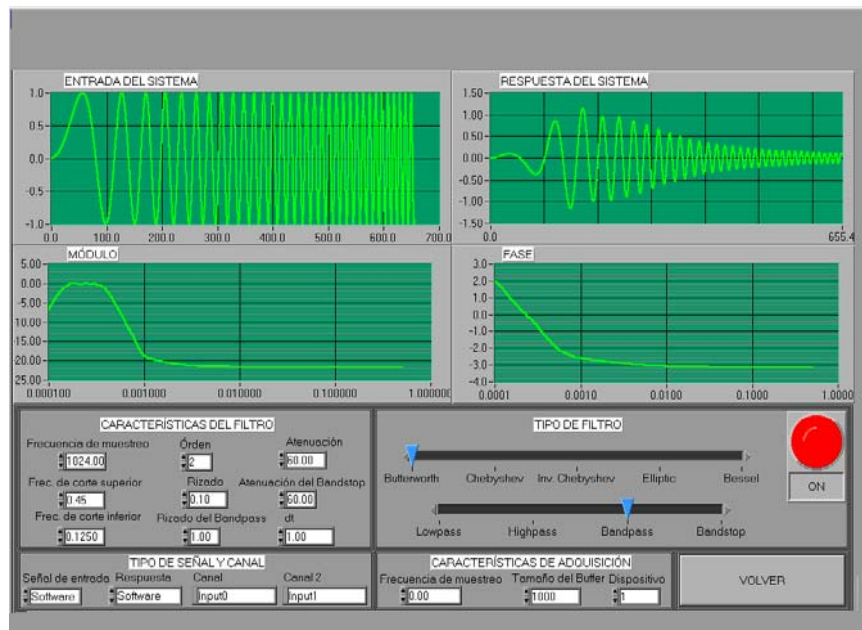


Fig. 12. Panel frontal y pantalla gráfica del módulo de filtrado digital

3.4. Módulo de simulación

El diagrama de Bode de una función de transferencia es una herramienta gráfica de gran utilidad para el análisis y diseño de sistemas lineales de control de tiempo continuo. El programa representa el diagrama de Bode de cualquier función de transferencia modelada por la ecuación:

$$G(s) = K \cdot \frac{(s + c_1) \cdot (s + c_2) \cdot (s + c_3)}{(s + p_1) \cdot (s + p_2) \cdot (s + p_3)} \quad [5]$$

En este caso, tanto el módulo (en dB) como la fase se han calculado empleando la estructura formula node que permite ejecutar fórmulas matemáticas que serían complicadas de crear utilizando las diferentes herramientas matemáticas que LabVIEW incorpora en sus librerías. En la figura 13 se observa una gráfica correspondiente a una simulación, y en la figura 14 un detalle de la programación. Se ha optado por esta modalidad, con fines experimentales, en lugar de emplear la función de librería Transfer_Function1.Vi como en la programación del módulo de filtrado digital cuya interface de usuario corresponde a la figura 12.

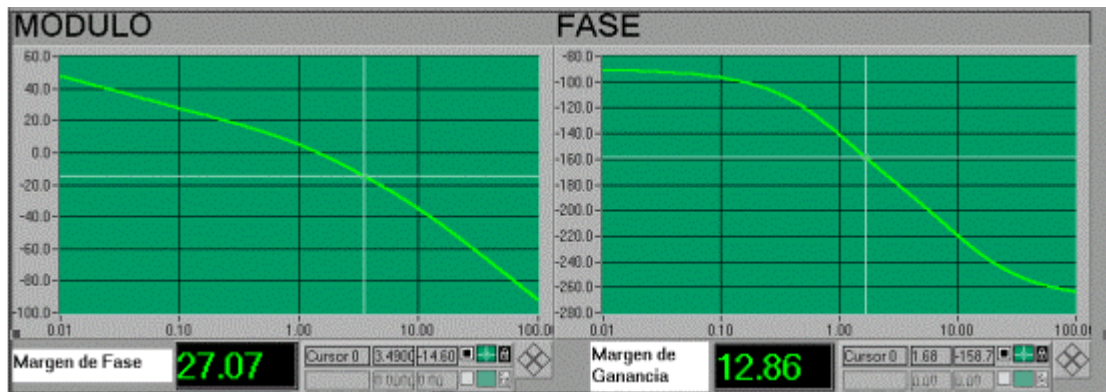


Fig. 13. Representación gráfica del diagrama de Bode con simulación

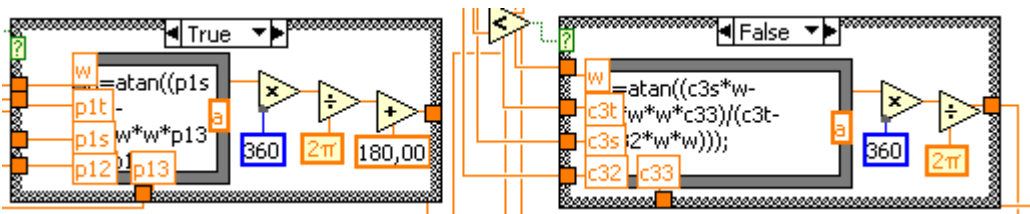


Fig. 14. Detalle de programación del diagrama de Bode con simulación

4. CONCLUSIONES

- El mayor logro ha consistido en la integración en un único entorno de trabajo de cuatro módulos con funcionalidades distintas, operativamente complementarias entre sí y que proporcionan al usuario una perspectiva de la unidad de conocimiento propia de la ingeniería electrónica.
- La adquisición de señales, para su posterior procesamiento y visualización, puede realizarse desde tres fuentes distintas: hasta 16 señales externas analógicas en tiempo real, desde un fichero, o generadas internamente. Los resultados finales también pueden almacenarse en otro fichero.
- Por su estructura modular es un sistema abierto y flexible pudiendo, por ejemplo sustituirse la adquisición de señales a través de la DAQ por otra tarjeta basada en un μC y conectada al PC vía RS232, así ampliarlo para su activación remota por Internet o su interacción con el programa Matlab.

10. BIBLIOGRAFÍA

- [1] A.M. Lázaro, LabVIEW. *Programación Gráfica para Control de Instrumentación*, Paraninfo, Madrid, 2001.
- [2] M.L. Chugani, A.R. Samant, and M. Cerna, *LabVIEW SignalProcessing*, Prentice-Hall, S. River USA, 1998
- [3] R. Jamal, H. Pichlink, *LabVIEW applications and solution*. Prentice-Hall, S. River USA, 1998.
- [4] R.H. Bishop, *Learning with LabVIEW*, Prentice-Hall, S. River USA, 2001.
- [5] [1] A.M. Lázaro, D. Biel, J. Olivé, J. Prat, F. J. Sánchez., *Instrumentación virtual. Adquisición, procesado y análisis de señales*. Ediciones UPC, Barcelona, 2001.
- [6] B. Paton, *Sensors, Transducers & LabVIEW*. Prentice-Hall, S. River USA, 1998.