

COLECTOR DE DATOS PARA VIBRACIONES BASADO EN DSP

Ing. Javier I. Roitman*, Rodrigo Alcoberro*, Ing. Claudio Verrastro**
*Grupo de Inteligencia Artificial,
Facultad Regional Buenos Aires.
Universidad Tecnológica Nacional.*

Palabras Clave: DSP, portátil, multirate, vibraciones, colector de datos.

Resumen:

Se presenta un colector de datos portátil basado en tecnología de Procesamiento Digital de Señales (DSP) para captura de parámetros de vibraciones en elementos rotativos de máquinas de mediano y gran porte, para su uso en el mantenimiento preventivo y predictivo de tales máquinas. Sus características principales son: el empleo del DSP tanto para el procesamiento de las señales adquiridas como para la gestión y presentación gráfica de las mediciones, evitándose el uso de un segundo procesador a tal efecto; una construcción robusta y confiable carente de partes móviles; gran capacidad de memoria no volátil y extensa autonomía. Dada la potencia del equipo y flexibilidad del diseño centrado en el software, se planea en un futuro dotarlo de herramientas de análisis automatizadas aplicando técnicas de inteligencia artificial.

1. Introducción

1.1. Función del equipo

El colector de datos fue diseñado para medir y almacenar el estado de vibración, tanto en aceleración como en velocidad, de más de 1000 puntos. Estas mediciones pueden ser de valores pico, eficaz o espectro de frecuencias con 400 líneas de resolución. Funciona además como analizador de primera instancia ya que posee distintos tipos de alarmas configurables. Una de las ventajas de la utilización de este tipo de equipos es que no se debe detener la máquina a ensayar y disminuye considerablemente el riesgo de sufrir paradas por fallas imprevistas.

Como elemento transductor se utiliza un acelerómetro activo que entrega un nivel de señal adecuado al circuito de entrada del equipo.

El equipo está centrado alrededor de un procesador digital de señales ADSP-2181, de 16 bits de palabra de datos y punto fijo. El diseño toma ventaja de la gran velocidad de procesamiento de estos dispositivos empleándolos no sólo para las tareas de adquisición, procesamiento y medición de la señal, que es la aplicación clásica de un DSP, sino también, por consideraciones de tamaño y consumo, para ejecutar la interfaz con el usuario, la administración de la base de datos jerárquica, la presentación gráfica de las mediciones y la emulación de un UART (Universal Asynchronous Receiver Transmitter) para la

comunicación serie con una computadora personal compatible PC.

La construcción sumamente robusta y un paquete de baterías recargables de NiMH, permite obtener una unidad muy confiable, liviana, de bajo costo, con gran autonomía, alta velocidad de procesamiento y fácil manejo reemplazando con ventaja a una versión anterior basada en una notebook 486.

La plataforma de hardware está integrada por un kit de desarrollo estándar de DSP, una segunda placa de circuito impreso que aloja la memoria y las interfaces de pantalla y teclado, además de la electrónica para el acondicionamiento de la señal del sensor, el cargador de baterías y una pantalla gráfica LCD.

Cuenta con un convertidor A/D de dos canales, 16 bits de resolución y una tasa máxima de muestreo de 48Ks/s. El procesador, de arquitectura Harvard modificado, posee una capacidad de memoria RAM interna de 16384 palabras de 24 bits para código y datos, y otras tantas, de 16 bits, sólo para datos. La frecuencia de reloj es de 33MHz, con un rendimiento del orden de los 33MIPS. Para el almacenamiento masivo de los datos, cuenta con un dispositivo NVRAM de 2MB, el más grande disponible al momento de realizado el desarrollo. La interfaz con el usuario está implementada mediante una pantalla LCD gráfica de 320x240 pixeles de resolución, iluminación de descarga gaseosa y un teclado de cuatro teclas, cuyas funciones son asignadas dinámicamente. La Fig. 1 muestra el aspecto real del equipo.

* UTN FRBA E-mail: jroit@sinectis.com.ar

** UTN FRBA - CNEA

La programación se estructura en dos grandes bloques. En primer lugar, la adquisición de los datos, procesamiento y medición, realizados íntegramente en lenguaje ensamblador con el fin de obtener el máximo rendimiento y lograr un aprovechamiento integral de los recursos del procesador. En segundo término, la administración de la base de datos, transferencia de datos, presentación de mediciones, configuración del equipo y demás tareas operativas están codificadas en lenguaje ANSI-C. Esto permitió su desarrollo y depuración en PC, al tiempo que facilitó la manipulación de tipos de datos complejos como ser cadenas de caracteres, números en coma flotante y estructuras. Entre los algoritmos de procesamiento de señales implementados se cuentan la adquisición y filtrado de señales a múltiples frecuencias de muestreo por decimado e interpolado (procesamiento multi-rate), transformada rápida de Fourier, integración, cálculo de verdadero valor eficaz, y en un futuro próximo, detección de envolvente.

Otro punto de interés en la programación es que careciendo el equipo de una UART, la misma se ha emulado por programa, alcanzando velocidades de transferencia de 115200 bps mientras opera en el trasfondo de otros procesos.



Fig. 1: Fotografía del equipo

1.2. Campo de Aplicación

El equipo se puede utilizar para recolectar las vibraciones provenientes de equipos rotativos. En la industria en general, aeronáutica, automotrices, etc. que adopten el sistema de mantenimiento predictivo y preventivo como método de trabajo. Por su robustez y elevada autonomía cuadra perfectamente en la categoría de equipo de medición en campo.

1.3. Características

Las características mas sobresalientes del colector son las siguientes:

- Procesador: ADSP2181 a 33MHz
- Cuantización: 16 bits
- Ancho de banda: 10Hz a 10KHz ($F_s = 32kHz$)
- Escala de frecuencias: 100Hz a 10KHz en pasos 1,2,5.
- Resolución espectral: 400 líneas nominales.
- Señal de Entrada: Acelerómetro activo.
- Escala de aceleración: 0.2G a 100G.
- Escala de velocidad: 0.2mm/s a 100mm/s.
- Cambio de escalas: Automático, en pasos 1,2,5.
- Promediado: 1, 2, 5, 10, 20 o 50 mediciones
- Mediciones: Espectro de aceleración.
Espectro de velocidad.
Valor Eficaz de aceleración.
Valor Pico de aceleración.
Valor Eficaz de velocidad.
Valor Pico de aceleración.
- Capacidad de almacenamiento de aproximadamente 1000 mediciones en una memoria RAM no volátil de 2MB de capacidad.
- Vínculo a PC serie RS232 a 115200 bps máximo.
- Autonomía de mas de 8 horas con alimentación propia interna.
- Medición manual o automática.

2. Arquitectura del equipo

2.1. Diagrama en bloques

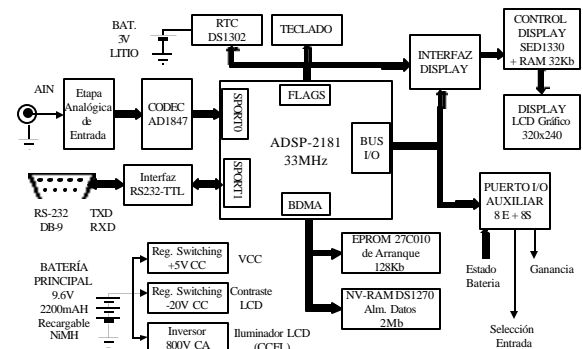


Fig. 2: Diagrama en bloques

El equipo posee una etapa analógica de entrada que tiene como función adaptar la impedancia de salida del sensor a la de entrada del CODEC y ajustar la amplitud de la señal de salida del acelerómetro a 20mV/G.

El CODEC AD1847 posee un convertidor A/D Sigma-Delta con una resolución de 16 bits y una frecuencia de muestreo máxima de 48 kS/s. Utiliza la técnica de sobremuestreo, muestreando la señal a una frecuencia mucho mayor que la frecuencia de Nyquist con una resolución de un bit. Luego posee un conformador de ruido que disminuye el ruido en la

banda de frecuencias por debajo de $F_s/2$, siendo F_s la frecuencia de Nyquist. Un filtro digital del tipo FIR y un decimador logran la resolución de 16 bits y la tasa deseada respectivamente.

El DSP carga el programa en la memoria de Programas interna (RAM) durante el arranque del equipo desde la memoria EPROM 27C010 mediante un proceso llamado BOOTSTRAP LOADING. El DSP comienza este proceso cargando y ejecutando las primeras 32 palabras de la EPROM a través del BDMA (Byte-wide Direct Memory Access). A su vez, este fragmento de código configura el BDMA para cargar el resto del programa.

Como memoria de almacenamiento principal para las mediciones y algunas variables de estado se utiliza una NVRAM (Non-Volatile Random Access Memory) Dallas DS1270 que posee un tiempo de retención de 5 años y una capacidad de 2Mb. Este dispositivo tiene la forma de un híbrido, incorporando la RAM, la batería y los circuitos de protección contra escritura (cuando la tensión de alimentación principal cae por debajo de un umbral) y conmutación de la fuente de alimentación. El tiempo de acceso es de 100ns. Los datos desde la memoria y a ella se transfieren a través de BDMA.

Posee un reloj de tiempo real (RTC) DS1302 con alimentación independiente y una pequeña memoria que se comunica con el DSP en forma serie a través de una interfaz serie hecha por software.

La interfaz de pantalla funciona como un bus bidireccional de 8 bits de baja velocidad ya que el DSP es mucho más rápido que el controlador de la pantalla. Esta interfaz posee un registro que mantiene los datos provenientes del DSP el tiempo necesario para que puedan ser tomados por dicho controlador, y un buffer TRI-STATE para leer la memoria de video o el estado. Las señales de CHIP SELECT, WRITE, READ, se manejan manualmente.

El controlador de pantalla es un SED1330 que permite utilizar hasta dos capas superpuestas de texto y gráficos. La forma en que interactúan estas capas es programable. Su frecuencia de reloj es de 10 MHz e incorpora 32 KB de memoria RAM. El generador de caracteres interno conforma la norma JIS 5 x 7

La pantalla es de cristal líquido con una resolución de 320 x 240 pixels en modo gráfico y 30 renglones por 40 columnas en modo texto. Es la pantalla de mayor resolución que puede manejar este controlador. Está iluminada por una lámpara de descarga gaseosa de alto rendimiento.

El DSP se conecta al CODEC a través del puerto serie 0 (SPORT0). El puerto serie 1 (SPORT 1) se utiliza

para emular la UART. El puerto de FLAGS es de 8 bits, programable bit a bit como entrada o salida, y se utiliza para controlar el RTC, el teclado y la interfaz para la pantalla. El bus de I/O es de 8 bits y a través de él se intercambian los 8 bits de datos entre el DSP y la pantalla gráfica.

La alimentación es autónoma y está compuesta por un paquete de baterías de NiMH, un regulador conmutado de 5VCC para alimentar la lógica y otro de -20VCC para el contraste del LCD. La iluminación del LCD es provista por un inversor que entrega 800VCA.

2.2. Estructura del programa

La señal procesada por las etapas analógicas de entrada es muestreada a 32Ks/s. Debido a que las frecuencias de muestreo disponibles en el convertidor A/D no coincidían con las requeridas por el proyecto, se eligió tomar muestras a esta frecuencia y luego emplear un algoritmo de conversión de frecuencia de muestreo por programa. De esta forma fue posible obtener un amplio rango de frecuencias tales que los espectros resultasen con la resolución y ancho de banda requeridos.

El equipo puede realizar una de tres mediciones básicas: valor pico, valor eficaz o espectro. Los resultados de las mediciones son luego promediados. Las frecuencias de muestreo, los tipos de medición y la cantidad de mediciones a promediar son seleccionadas por el usuario, ya sea a través de una secuencia o ruta de mediciones previamente confeccionada y que se descarga en el equipo desde una computadora PC compatible, o bien en forma directa para realizar mediciones adicionales fuera de ruta.

También es posible recoger información de variables de proceso (presión, temperatura, etc.). Estos valores son ingresados por teclado.

Cada medición de la ruta lleva asociada dos máscaras de alarma (una de precaución y otra de emergencia), valores límite y valores históricos. Para mediciones espectrales, la máscara puede contener hasta diez bandas de frecuencia, cada una con sus propios umbrales de precaución y emergencia. Esto permite ajustarla a la firma espectral particular de cada máquina.

2.3. Interfaz con el usuario

La interfaz con el usuario emplea la pantalla y el teclado. La operación resulta simple e intuitiva. Las teclas están dispuestas verticalmente, duplicadas a ambos lados del equipo a fin de facilitar su uso con una sola mano para operadores diestros y zurdos (Ver Fig. 1).

Un menú principal controla las principales funciones del colector: operación, recepción de rutas conteniendo las mediciones a realizar, transmisión de las mediciones efectuadas, posibilidad de realizar mediciones fuera de ruta y control de configuración y estado.

Una serie de sub-menús permite recorrer la ruta en forma jerárquica, seleccionando la máquina a medir y dentro de la misma, el punto, tipo y parámetros de la medición.

Existen dos modos de operación: manual y automático. En el modo manual, el operador selecciona la medición a realizar, pudiendo acceder en forma aleatoria a cualquiera programada en la ruta. En el modo automático, el equipo recorre secuencialmente la ruta partiendo de la medición actual y hasta completar la ruta o ser detenido por el operador. El equipo informa al operador cuando ha completado una medición, cuál es el próximo punto a medir, y pide confirmación antes de continuar. El operador sólo debe ocuparse de colocar el sensor en los lugares indicados.

La Fig. 3 muestra un ejemplo de la pantalla, durante la medición de espectros. El valor RMS global, la fecha y resultado de la medición histórica anterior y los parámetros de la medición se muestran en el costado derecho. También pueden verse el cursor de medición y las máscaras de alarma de precaución (línea de trazos) y emergencia (línea llena).

La Fig. 4 muestra una pantalla de medición escalar usada para mostrar los valores pico y eficaz

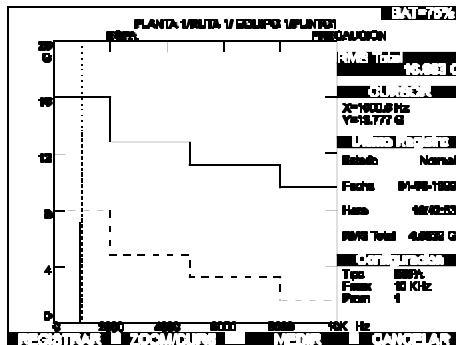


Fig. 3: Pantalla de medición de espectros

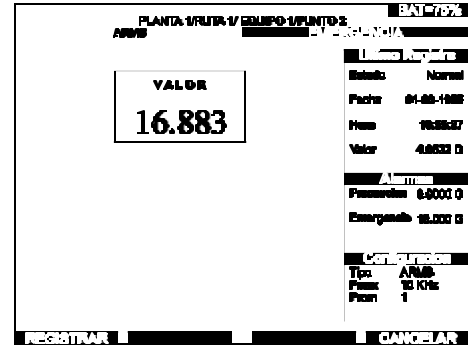


Fig. 4: Pantalla de medición de valor eficaz

3. Algoritmos implementados

3.1. Conversión de la frecuencia de muestreo

Al emplearse la FFT como algoritmo de cálculo del espectro y, estando la resolución espectral fijada por especificaciones en 400 líneas, las frecuencias de muestreo para cada ancho de banda quedan unívocamente determinadas.

El algoritmo FFT requiere vectores de entrada de longitud $N=2^K$ y su resolución espectral nominal es de F_s/N , donde F_s es la frecuencia de muestreo. En nuestro caso, para obtener 400 líneas de salida, $N=1024$.

Luego de muestrear la señal con una frecuencia de 32KHz se aplican los algoritmos de interpolación y decimado a fin de obtener las frecuencias de muestreo necesarias.

Interpolación significa generar una señal con una frecuencia de muestreo L veces mayor que la de entrada. Para lograrlo se intercalan $L-1$ muestras de valor cero entre cada muestra original, obteniéndose L muestras de salida por cada una de entrada. Este proceso tiene dos efectos colaterales: el primero es una reducción de la amplitud de la señal, ya que la energía de cada muestra de entrada se debe repartir entre L muestras de salida; el segundo es la producción de alias de la señal de entrada alrededor de su frecuencia de Nyquist original. La reducción de amplitud se corrige multiplicando cada muestra por un factor L . Los espectros alias se eliminan con filtrado pasabajos, el cual suaviza la salida, asignando valores distintos de cero a las $L-1$ muestras insertadas.

Decimar, por el contrario, significa generar una señal de salida con una frecuencia de muestreo M veces menor que la de entrada. El proceso puede verse como un sub-muestreo digital. Se filtra la señal con un filtro pasabajos

anti-alias. Luego, por cada M muestras de entrada filtrada se toma una y se descartan las otras $M-1$.

Eligiendo convenientemente los factores L y M , es posible trasladar una señal de una frecuencia de muestreo F_S a otra $F_S * L/M$. Este proceso se denomina **Conversión Racional de Frecuencia de Muestreo**. Cuando se combinan etapas de interpolación y decimado en cascada, los filtros de suavizado y anti-alias quedan también en cascada, pudiendo eliminarse aquel de mayor frecuencia de corte.

La Fig. 5 muestra la estructura del convertidor de frecuencia de muestreo básico. El bloque L inserta $L-1$ muestras de valor cero por cada una de la secuencia de entrada $x(n)$, incrementando su frecuencia de muestreo L veces y produciendo la secuencia $x(k)$. En el otro extremo, el bloque M realiza la operación contraria, removiendo $M-1$ muestras de la secuencia $w(k)$ para producir la secuencia de salida $y(m)$ de frecuencia de muestreo M veces inferior a $w(k)$. Los bloques $h_1(k)$ y $h_2(k)$ representan los filtros de suavizado y anti-alias, que al quedar en cascada, se reemplazan por un único filtro $h(k)$.

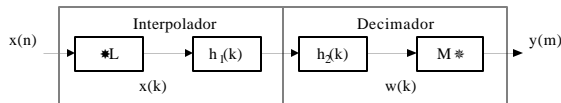


Fig. 5: Convertidor de frecuencia de muestreo

Para una aplicación como esta, es conveniente elegir un filtro de tipo FIR (respuesta finita al impulso). Ya que su salida sólo depende de las entradas previas y no tiene realimentación, puede lograrse una implementación más eficiente en términos de tiempo de cálculo al no tener que calcular la salida para cada muestra. Sólo se calculan aquellas salidas que serán efectivamente usadas, produciéndose de esta forma la reducción de la frecuencia de muestreo. El diseño del filtro puede hacerse por cualquier procedimiento convencional.

Un detalle particular del diseño de filtros para análisis espectral es que como el extremo superior del espectro se desecha, puede permitirse alias en esa región y de esta forma obtener un filtro de menor orden.

Por último, cuando se desean implementar factores de interpolado o decimado grandes (mayores o iguales a 10) es conveniente emplear varias etapas en cascada en lugar de una sola, tal que el producto de los factores sea igual al factor original. Cuando se utilizan factores grandes, es necesario filtrar con bandas de transición menores y por ende, con filtros de mayor orden. Con

etapas en cascada se pueden usar filtros de menor orden, resultando en una implementación más eficiente y que requiere menos memoria y tiempo de cálculo (Proakis y Manolakis, 1996).

El convertidor empleado en este equipo utiliza una etapa de interpolación con factor L ajustable y hasta tres etapas idénticas de decimado en cascada con factor M fijo igual a 5, logrando factores de decimado de 5, 25 y 125. Los filtros permiten una cierta cantidad de alias en la banda de transición, ya que la misma será descartada para el análisis. De esta forma se obtiene la respuesta en frecuencia requerida (ver Fig. 6) con filtros de sólo 84 derivaciones. El tiempo de cálculo de cada filtro es de $2.7\mu s$.

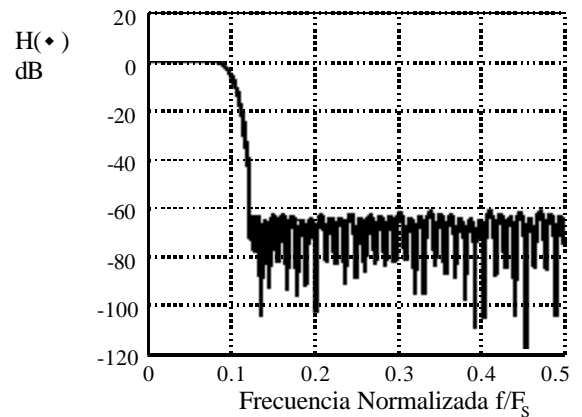


Fig. 6: Respuesta en frecuencia de los filtros

La Tabla 1 indica para cada uno de los alcances de ancho de banda del equipo, F_{MAX} , los factores de interpolado, decimado, frecuencia de muestreo F_S y la resolución nominal de los espectros obtenidos en cada caso.

F_{MAX} (Hz)	Factor de Interpolado	Etapas de Decimado x5	F_S (Hz)	Resol. (Hz/linea)
100	1	3	256	0.25
200	2	3	512	0.5
500	1	2	1280	1.25
1000	2	2	2560	2.5
2000	4	2	5120	5
5000	2	1	12800	12.5
10000	4	1	25600	25

Tabla 1: Factores de interpolado y decimado

3.2. Adquisición de datos

Por lo expresado arriba, se requieren entonces vectores de entrada de 1024 muestras, tomadas con la frecuencia de muestreo adecuada para poder obtener un espectro de 400 líneas con la resolución espectral pedida.

El convertidor A/D obtiene muestras a 32KHz. Cada vez que toma una muestra, produce una interrupción al DSP. La atención de esta interrupción primero pasa la muestra por el conjunto de interpolado y decimado. Si dicho conjunto produce una salida, la almacena en un buffer.

Existen dos de tales buffers, ambos de 1024 posiciones. En cualquier momento, uno está recibiendo las muestras de señal mientras que el otro sirve de entrada a las rutinas de cálculo. Cuando el buffer de señal se llena, se lo intercambia con el de cálculo. Como el tiempo de cálculo de todos los procesos de medición es mucho menor que el tiempo necesario para adquirir las 1024 muestras, no existe riesgo de interferencia. A modo de referencia, el tiempo de cálculo máximo está en el orden de los 4ms; pero se necesitan 40ms para llenar el buffer tomando muestras a la máxima frecuencia de 25600Hz. Por ende, es el tiempo de adquisición y no el de cálculo quien acota la velocidad de medición, ya que este último es constante mientras que el primero aumenta al reducirse la frecuencia de muestreo para obtener los diferentes anchos de banda requeridos.

3.3. Transformada rápida de Fourier y compensación de ventana

Para la estimación del espectro de la señal se emplea la Transformada Rápida de Fourier o FFT. La longitud de los vectores de entrada se ha fijado en 1024 (2^{10}) muestras.

Al operar sobre señales aleatorias, cada vector no contiene una cantidad entera de periodos. Ello introduce una distorsión en el espectro estimado conocida como **fuga espectral** (spectral leakage), característica de la FFT. La forma de minimizar dicha distorsión consiste en multiplicar el vector de entrada por una función conocida como **ventana**. Las funciones de ventana tienen la característica de poseer un máximo central y decaer gradualmente hacia cero en los extremos, en forma acampanada. La elección de una en particular de las muchas conocidas depende de la aplicación. En nuestro caso, encontramos que la ventana de Hanning es de uso extensivo para el análisis de vibraciones, por lo que se optó por dicha función. Su expresión es:

$$v(n) = 0.5 * \left(1 - \cos \frac{2\pi n}{N-1} \right) \quad \begin{matrix} N = 1024 \\ 0 \leq n \leq N-1 \end{matrix}$$

La transformada discreta de Fourier (DFT) se expresa como:

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) * W_N^{nk} \quad 0 \leq k \leq N-1$$

$$W_N^{nk} = e^{-j \frac{2\pi}{N} nk}$$

$$\tilde{x}(n) = x(n) * V(n)$$

dónde $V(n)$ es la función de ventana.

La FFT se implementa aprovechando el hecho de que el término exponencial de la DFT, W_N^{nk} , es periódico de periodo N y tiene simetría par. Esto permite reducir la cantidad de operaciones necesarias para calcular la DFT de aproximadamente N^2 a $N * \log_2(N)$.

En general los algoritmos FFT tienen dos grandes condicionantes. En primer lugar, la cantidad de datos de entrada N debe ser potencia de 2; en nuestro caso $N=2^{10}=1024$. En segundo lugar, los datos deben llegar ordenados en forma **“Bit-Reversal”**. Este reordenamiento de los datos o **“scrambling”** es generado por hardware propio del DSP, en forma transparente y eficiente.

La elección de N condiciona la máxima resolución espectral posible, que es:

$$R = \frac{f_s}{N}$$

donde f_s es la frecuencia de muestreo en Hz, y R es la resolución espectral nominal en Hz por línea.

Esta resolución es un límite máximo y sirve para fijar el eje de frecuencias. En la práctica la misma se ve reducida por la función de ventana empleada. En el caso de la ventana de Hanning, se reduce a la mitad. Esto significa que aunque la escala de frecuencia sea de R Hz por línea, no se puedan diferenciar picos situados a menos de $2 * R$ Hz.

A partir del espectro complejo, se calcula el módulo de las primeras 400 componentes y se las divide por $\sqrt{2}$. Los valores resultantes representan el espectro de amplitud eficaz de la aceleración o velocidad en el ancho de banda deseado.

Además de afectar la resolución espectral, la función de ventana reduce la amplitud de las componentes observadas. Para mantener constante el nivel de energía en el espectro estimado, es necesario normalizarlo por la

energía de la ventana empleada (Proakis y Manolakis, 1996, Shanmuga n y Breipohl, 1988).

Sea U la energía de la ventana, calculada como

$$U = \frac{1}{N} \sum_{k=0}^{N-1} V^2(k)$$

Luego, el espectro normalizado será

$$\hat{X}'(k) = \frac{\hat{X}(k)}{\sqrt{U}}$$

De esta forma, el valor eficaz global calculado a partir del espectro coincide con el calculado a partir de la forma de onda de la señal.

3.4. Mediciones

Además de la estimación de espectro, el equipo realiza otras dos mediciones: el valor pico y el valor eficaz o RMS de la señal.

Estos valores se emplean para obtener una caracterización de banda ancha (o global) del comportamiento vibratorio de la máquina bajo estudio. En general se emplea el valor eficaz, reservándose el valor pico para cuando se deben detectar valores muy reducidos, evitándose el gran error de cálculo asociado a la raíz cuadrada cuando su argumento es pequeño.

Las tres mediciones (espectro, valor pico y valor eficaz) pueden realizarse tanto sobre señales de aceleración como de velocidad.

Al medir sobre máquinas reales en funcionamiento, los resultados presentan grandes variaciones debido a la existencia de un sinnúmero de fuentes de ruido tanto eléctricos como mecánicos. La forma usual de reducir dicha variabilidad es promediar los valores medidos sobre un cierto número de observaciones. La cantidad de observaciones a promediar es seleccionada por el usuario.

3.5. Base de datos

La función principal del equipo es la de recoger datos sobre el comportamiento vibratorio de un conjunto de máquinas de una determinada planta industrial.

Debe llevar entonces la información sobre cuáles son las máquinas que serán estudiadas, que mediciones hay que hacer sobre cada una, en qué parte específica de la máquina deben ser hechas y cuáles son los parámetros particulares de cada una.

Estos parámetros son definidos por el personal encargado de los estudios de **Mantenimiento Predictivo**, mediante un software especial que corre en una computadora tipo PC. Existen siete mediciones posibles, a saber: aceleración pico, aceleración eficaz, espectro de aceleración, velocidad pico, velocidad eficaz, espectro de velocidad y variables de proceso. Estas últimas no se miden sino que se ingresan directamente por teclado. Todas las mediciones incorporan alarmas de precaución y emergencia, fecha de la última medición, su valor y su estado. La frecuencia máxima, cantidad de promedios y otros parámetros característicos de cada medición son programables.

Al principio de una jornada de trabajo, la información de una o más plantas es bajada al colector vía RS-232. El operador sale a recorrer las plantas asignadas siguiendo la lista que el equipo le presenta en la pantalla. El colector cuenta con un modo automático que facilita dicha tarea, realizando de una sola vez todas las mediciones asociadas a cada punto y pasando de un equipo al siguiente una vez realizadas todas sus mediciones. El operador sólo debe cambiar el sensor de lugar y presionar una tecla. Cuando se completa la jornada, los datos recogidos son descargados vía RS-232 para su posterior análisis, registro y seguimiento histórico.

En vista de la organización jerárquica de la información de plantas, se decidió modelarla con una estructura jerárquica de cinco niveles. El nivel de PLANTAS es el superior y el de MEDICIONES, el inferior. Cada nivel es, a su vez, una lista doble enlazada de ítems de igual jerarquía. Cada uno de estos ítems contiene una referencia descendente al primero de sus sub-ítems. Las referencias ascendentes se guardan en una pila FIFO, independiente de la pila del compilador C.

En razón de su complejidad, tanto la base de datos como todo el resto de las rutinas relacionadas con la interfaz usuario están codificadas en lenguaje C.

El volumen de datos requerido para una jornada de trabajo es considerable y debe ser mantenido en el equipo aún ante una falla de su batería principal. Luego de un extenso estudio de soluciones disponibles, se optó por una memoria RAM estática con respaldo por batería de litio interna (NV-RAM) y una capacidad 2Mb. Dicha capacidad es suficiente para más de 1000 mediciones, dependiendo de la selección particular de las mismas.

3.6. Presentación gráfica

Se implementó en el equipo un conjunto de rutinas de manejo básico de pantalla tanto en modo gráfico o de texto estas rutinas son:

- **Líneas** continuas, de trazos y punteadas.
- Impresión de **cadena de caracteres** en video normal e invertido.
- Volcado de **imágenes en mapa de bits**.
- **Caracteres de doble tamaño**.

Estas rutinas están implementadas en lenguaje ensamblador compatible con C ya que son llamadas desde las rutinas de menú de usuario realizadas en ANSI-C.

El sistema controlador - pantalla funciona realizando un barrido de forma similar a un tubo de rayos catódicos, con una parte de la línea "activa" y otra parte de "retrasado". Se debe acceder a la memoria del display durante el retrasado, o tiempo de línea inactiva para no interferirlo, ya que el controlador de pantalla no realiza la contención del bus. Se debe chequear un bit de la palabra de estado para asegurarse de no acceder a la memoria de vídeo de doble puerto en el momento en que la pantalla se encuentra en la parte activa de la línea.

En la rutina de líneas se debió tener especial atención ya que la memoria de vídeo se organiza en forma de bits empaquetados de a ocho y esto complica las rutinas al no poder acceder a un bit en particular.

Si un objeto dibujado se superpone con otro o con caracteres se realiza una XOR con lo cual el pixel (o los pixeles) en cuestión cambian de encendido a apagado y viceversa. Para imprimir caracteres en video invertido se dibuja un fondo oscuro en la capa de gráficos y se imprime sobre ésta.

Para la presentación de algunas mediciones se desarrolló una rutina que imprime números del doble de tamaño con tipografía **Times New Roman**.

Para imprimir dibujos se desarrolló una rutina que imprime gráficos del tipo mapa de bits.

3.7. Comunicación serie

El sistema de comunicaciones se organiza según el modelo de capas de OSI:

- Nivel 1 Capa física: **RS232**
- Nivel 2 Capa de enlace de datos: **Protocolo 3964 (modificado)**.
- Nivel 7 Capa de aplicación: **Protocolo propio**.

El DSP utilizado para el proyecto no tiene una UART dedicada (hardware), por lo que se implementó una UART por software. El principio de funcionamiento de esta UART emulada es el muestreo de los datos provenientes del puerto serie 1 (SPORT 1) a una tasa mayor que la máxima tasa de transferencia de la UART. En esta implementación la máxima tasa de transferencia es de **115200 bits/s** y se muestrea tres veces cada bit. Entonces:

$$115200 * 3 = 345600 \text{ muestras/s}$$

Una vez que se tomaron las tres muestras de un bit se realiza una "votación" para decidir si lo recibido fue un cero o un uno (lógica 2 de 3).

Como reloj de muestreo se utiliza el temporizador que dispone el DSP. Una vez recibido el bit de START se reciben los datos. Estos entran en un registro de desplazamiento para luego ser ingresados en memoria. Todo este proceso trabaja mientras el equipo continúa su funcionamiento normal. Esto es posible gracias a la gran velocidad de procesamiento del DSP y su excelente rendimiento para las comunicaciones.

Para sincronizar el comienzo de la recepción o bit de START se utiliza una entrada de interrupción externa (IRQ1), conectada a la línea de recepción. Iniciada la recepción de un carácter, la misma debe completarse dentro de 0.5s (periodo de time-out). El protocolo de línea sigue el estándar RS-232, con un bit de arranque (START), 8 de datos y uno de parada (STOP). No se emplea paridad.

El protocolo 3964 se modificó para cumplir mejor con los requerimientos del proyecto. Originalmente está orientado a bytes y limitado a paquetes de 256 bytes de longitud.

El protocolo posee un sistema de handshake para establecer la comunicación entre equipos, definiéndose uno de los equipos como **Emisor** y el otro como **Receptor**. Previo al envío de cada paquete, el emisor interroga al receptor. Si el mismo se encuentra listo, le envía un paquete. En caso contrario, espera y reintenta. Luego de cada paquete enviado, el emisor espera un asentimiento por parte del receptor, confirmando o rechazando los datos recibidos. En caso de rechazo, el emisor reintenta hasta un máximo de tres veces. Un error no recuperado provoca el fin de la comunicación.

El paquete empleado tiene la siguiente estructura:

SeqNr	Comando	Largo	Dato(0) - Dato(m)	BCC
-------	---------	-------	----------------------	-----

INFO(0) INFO(n)

SeqNr:	Número de secuencia	16 bits.
Comando:	Código de comando	16 bits
Largo:	Largo del campo de datos	16 bits
Dato(i):	Datos	n*16 bits
BCC:	Paridad	8 bits

Fig. 7: Formato de la trama

El BCC es una suma de comprobación que se calcula mediante la XOR de toda la trama, byte a byte.

4. Lenguajes de programación. Problemas y soluciones

4.1. Requisitos del proyecto

En los diseños de equipos de medición que incluyen procesadores digitales de señal (DSP), lo usual es que un procesador tipo x86 o un micro-controlador cumplan el rol de ejecutar la interfaz usuario, manejando la pantalla y el teclado.

En nuestro caso, quedó claro desde los primeros estudios de factibilidad que el DSP tendría muy poca carga de trabajo, típicamente menor al 5% del tiempo. Se decidió entonces por un diseño de una sola CPU, con la programación codificada en dos lenguajes: Ensamblador para el código de medición, a fin de lograr velocidad, eficiencia y un óptimo uso del DSP; y C para la interfaz usuario y base de datos, a fin de poder manipular tipos de datos como arreglos, estructuras, cadenas y números en coma flotante fácilmente. El código que maneja la pantalla y comunicaciones también está estructurado en dos capas, estando en ambos casos la más próxima al hardware escrita en Ensamblador y la o las superiores en lenguaje C.

Como la base de datos y gran parte de la interfaz usuario debían ser diseñadas en paralelo con el hardware y software de manejo de pantalla y comunicaciones (de los cuales dependen), el lenguaje C posibilitaría el uso de un compilador y depurador estándar para PC como herramienta intermedia de desarrollo.

De hecho, ambos módulos, junto con un sinnúmero de rutinas auxiliares, fueron desarrollados y probados sobre una PC, simulando el hardware faltante con librerías escritas ex profeso. Cuando el hardware estuvo terminado, dichas librerías fueron reemplazadas por las definitivas, escritas en Ensamblador, pero con la misma interfaz. El traslado del código entre una y otra

plataforma, demandó mínimas modificaciones debidas a incompatibilidades y errores del compilador del DSP.

4.2. Requisitos del hardware

Tanto el DSP como el convertidor A/D empleados requieren rutinas de inicialización muy específicas, no provistas por el compilador C, y que deben escribirse en lenguaje Ensamblador.

Por otro lado, el convertidor A/D opera en una modalidad llamada "Auto-Buffering", similar a una transferencia por acceso directo a memoria (DMA). Este modo requiere la reserva permanente de recursos del DSP que entran en conflicto con los requisitos del entorno de lenguaje C.

Además, las rutinas de cálculo de FFT emplean la casi totalidad de los registros punteros del procesador, lo cual también supone un conflicto con el entorno del C.

Estas incompatibilidades fueron resueltas planteando dos entornos separados: un entorno anfitrión constituido por el programa principal en lenguaje C, y un entorno de servicio, constituido por las rutinas de medición. Estos entornos se comunican por medio de variables globales escritas por el Ensamblador y leídas por el C, y una única rutina que oficia de punto de conexión para transferir la ejecución entre uno y otro entorno. Cuando dicha transferencia se produce, el contexto del lenguaje C (registros y pila) es guardado en un espacio privado de RAM, se inicializa el convertidor A/D y su puerto asociado (SPORT0), se realiza la medición pedida, se desactiva el hardware y se restaura el contexto del C.

Las rutinas de interrupción del temporizador, apagado y botón de abortar están escritas de forma que sean compatibles con ambos entornos. El temporizador se emplea para la lectura del teclado y el estado de la batería y para la comunicación serie. La interrupción de apagado suministra un aviso anticipado de tal evento, permitiendo al equipo resguardar su estado en memoria no volátil. El botón de abortar se utiliza como medida de seguridad y herramienta de depuración, ofreciendo al programador la interrupción segura de lazos de programa en la eventualidad de que se encuentren bloqueados. Este botón no está accesible en la versión final del equipo.

4.3. Requisitos del compilador C

El compilador C requiere la reserva de una serie de registros internos del DSP para su propio uso. Esto provoca un conflicto con los requisitos del hardware, como se explicó anteriormente.

El DSP no está específicamente desarrollado para ejecutar programas compilados en lenguajes de alto

nivel. Especialmente, carece de una pila para el paso de parámetros entre subrutinas. El compilador C emula esta función reservando una zona de RAM de datos para guardar la pila y un juego de punteros para hacer las veces de Tope y Base de la pila (los registros SP y BP, respectivamente, de un x86).

4.4. Librerías

Debido a las características particulares del entorno de DSP empleado, muy pocas librerías estándar del compilador C están disponibles. En particular, sólo las librerías **string.h** (manejo de cadenas de caracteres) y **math.h** (emulación de punto flotante) fueron usadas en este proyecto.

Además, fue necesario desarrollar una serie de librerías de soporte, cuyas funciones incluyen: manejo de la pantalla gráfica, conversión y formato de datos para su presentación, manejo de teclado y reloj de tiempo real y pantallas de ingreso de parámetros.

Cabe destacar que esta última librería fue escrita con técnicas tan cercanas a la programación orientada a objetos como fue posible en un compilador C (carente de objetos). Esta metodología permitió un sustancial ahorro de espacio de programa y facilitó el reuso de código para implementar diferentes pantallas.

Estas librerías fueron diseñadas con el criterio de ser tan genéricas como fuese posible a fin de que pudieran ser reusadas en proyectos futuros sobre la misma plataforma de hardware.

4.5. Espacio y Velocidad

El compilador trabaja siempre en modo de optimización automática, pudiendo optarse por optimizar el código generado en velocidad (por defecto) o en tamaño.

Debido a la gran extensión del programa C (cerca de 10240 palabras de código, más 4096 palabras de ensamblador puro, sobre 16384 palabras de espacio total), se tuvo que cambiar la optimización al modo de mínimo tamaño. Aún así fue necesario estudiar y re-codificar muchas rutinas para reducir manualmente el largo del ejecutable, a fin de que cupiese en la memoria del DSP.

Como el código C no tiene exigencias de rendimiento, la pérdida de velocidad resultante no fue notoria. Más aún, en muchos casos, la optimización manual tanto redujo el código como aumentó su velocidad.

5. Evaluación experimental. Resultados obtenidos

5.1. Conversor A/D

La mínima resolución de amplitud necesaria para las medidas a realizar se fijó inicialmente en 12 bits. Esta resolución requería el uso de un preamplificador con dos rangos de ganancia ($\times 1$ y $\times 10$) conmutables por programa para cubrir todo el rango de amplitud del equipo y de un algoritmo de auto-rango.

Se optó entonces por emplear un conversor de 16 bits y un único rango de ganancia. De esta forma se mejoró notablemente el rango dinámico del equipo y se simplificó la etapa analógica. Además se redujeron notablemente los tiempos requeridos para realizar las mediciones al no ser necesario determinar previamente la ganancia a emplear ni esperar a que el amplificador se establezca luego del cambio de ganancia. Esto es de particular importancia en las mediciones de baja frecuencia, las cuales pueden tomar varios segundos.

Los 16 bits del conversor resultaron entonces totalmente aprovechados, brindando resolución adicional en la medición de señales de gran amplitud y permitiendo medir señales de pequeña amplitud dentro de la resolución mínima necesaria.

5.2. Medición de espectros

En la Fig. 3 se observa una medición de prueba de un espectro. Se excitó al equipo con una señal senoidal de 1KHz de frecuencia y amplitud similar a la observada en máquinas reales. La presentación resulta limpia y precisa, a pesar del reducido tamaño de la pantalla y de la gran cantidad de procesamiento previo. La escala vertical lineal no permite observar ninguna componente espúrea, resultando las mismas de amplitud inferior a 0.02G RMS, frente a 13.78G RMS de componente fundamental (se las observó empleando la función de zoom). El tiempo de medición resultó esencialmente el tiempo de adquisición de datos (40ms para $F_{MAX}=10KHz$; 4s para $F_{MAX}=100Hz$; por cada medición promediada.) ya que todo el procesamiento se realiza en tiempo real. El cálculo del espectro en punto fraccionario fijo brinda entre cuatro y cinco cifras significativas, más que suficientes para los propósitos del equipo y toma menos de 5ms. El error en la medición no supera el 1%.

5.3. Medición de valores globales

En la Fig. 4 se observa una medición de valor RMS de la misma señal del punto 5.2. En la práctica, el nivel de ruido que se encuentra al realizar estas mediciones es tan grande que, aún con un número elevado de promedios, los dos dígitos menos significativos resultan

ilegibles. Por este motivo se decidió truncarlos mostrándose sólo los tres más significativos. El tiempo requerido para medir es el mismo que para espectros.

6. Futuras ampliaciones

Como futuras mejoras de en las prestaciones del equipo se prevén:

- Adición de una función de espectro de envolvente para la detección temprana de fallas en rodamientos. El detector de envolvente estará implementado íntegramente por programa.
- Análisis detallado de los datos adquiridos empleando técnicas de inteligencia artificial como ser redes neuronales, para el reconocimiento de patrones de falla, sistemas expertos y lógica difusa para evaluar en forma automática fallas mecánicas incipientes en los elementos bajo estudio, estimar la progresión de las mismas y predecir los tiempos de falla
- Ajuste automático de los parámetros de detección y alarma bajo diferentes condiciones de la máquina.
- Desarrollo de un equipo de monitoreo continuo. Determinación y auto-ajuste de los parámetros de alarma. Detección inteligente de condiciones de alarma (evitar paradas innecesarias de la maquina bajo monitoreo)
- Adquisición y análisis de múltiples canales de señal en forma simultánea (real-time). Posibilidad de relacionar información entre canales para determinar condiciones de alarma con mayor precisión.

7. Conclusiones

El empleo de un muy avanzado procesador digital de señales ó DSP como núcleo de este equipo, junto con memorias de gran capacidad, una pantalla gráfica amplia y la carencia absoluta de partes móviles permitieron lograr un equipo de costo, dimensiones y peso reducidos pero de alto rendimiento y elevada confiabilidad.

La sencillez de la plataforma de hardware y la implementación de la totalidad de los procesos de medición en software le confieren la característica de “**instrumento de medición virtual**” ya que la misma plataforma puede ser, en un futuro, reprogramada para realizar otras tareas de medición o bien, una misma plataforma portar un conjunto de programas para diversas mediciones.

Referencias

- Analog Devices, *ADSP-2100 Family User Manual*, Prentice Hall (1995)
- Analog Devices, *Digital Signal Processing Applications Using The ADSP-2100 Family*, Prentice Hall (1992)
- Dallas Semiconductor, *DS1270 Preliminary Data Sheet*, Dallas Semiconductor (1996)
- Oppenheim A. V. and Schafer R. W., *Digital Signal Processing Prentice*, Hall (1975)
- Proakis J. G. and Manolakis D. G., *Digital Signal Processing*, Prentice Hall (1996)
- Sam Shanmugan K. and Breipohl A. M., *Random Signals, detection, estimation and data analysis*, Willey (1988)