

SISTEMA DOPPLER ULTRASÓNICO BASADO EN UN SISTEMA LINUX-GNU

REFERENCIA PACS: 43.35Yb 43.80QF

Cárdenas Francisco J.¹; García Fabian¹; Moreno Eduardo^{1,2}.

¹Departamento de Ingeniería en Sistemas Computacionales y Automatización, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Autónoma México. P.O. Box 20-276, Del. A. Obregón, 01000 México DF, México.

cardenas@uxdea4.iimas.unam.mx

fabian@uxdea4.iimas.unam.mx

²Instituto de Cibernética, Matemáticas y Física, Calle D No. 353, Vedado Habana, Cuba.
moreno@uxdea4.iimas.unam.mx

ABSTRACT

We present the development of a system based on a GNU based linux system for the visualization of a spectral image from a doppler signal.

RESUMEN

Aprovechando los desarrollos hechos en Linux para multimedia, se presenta un sistema de Ultrasonido Doppler con despliegue espectral para aplicaciones médicas. Utilizando GNOME ("GNU Network Object Model Environment") como herramienta de desarrollo y el lenguaje de programación Python, se logró un sistema en tiempo real para el despliegue de imágenes espectrales basado en FFT. El sistema propuesto utiliza adicionalmente las capacidades comerciales de multimedia de un PC Pentium II.

INTRODUCCIÓN

El rápido desarrollo tecnológico en el campo de la informática, está a la vista de todos nosotros. El desempeño numérico de los microprocesadores comerciales ha llegado a tal punto, que los podemos considerar para el desarrollo de aplicaciones de control y análisis de señales en tiempo real.

En este trabajo nos vamos a centrar en el análisis de un sistema de despliegue de imágenes espectrales en tiempo real de señales doppler de onda continua, para la

auscultación médica.

El trabajo esencial de este sistema es la captura de la señal ultrasónica a través de una tarjeta de audio convencional. Ésto es factible porque la naturaleza de la señal doppler ultrasónica, tiene un ancho de banda de 5 KHz (1,2,3), que tomando en cuenta a Nyquist, la frecuencia de muestreo debe ser por lo menos de 10 KHz. La frecuencia de muestreo de las tarjetas de audio de los sistemas multimedia de uso comercial es de 44 KHz.

Por otro lado, para poder hacer el procesamiento digital de la señal, debemos tener un algoritmo y procesador numérico rápidos. Las pruebas de desempeños que hemos hecho para el cálculo de FFT de 1024 muestras complejas, con el logaritmo FFTW, en una computadora con procesador Intel Pentium II a 300 MHz, son del orden de los 0.8 ms. Este resultado, comparado con el publicado por Analog Devices para su Sharc ADSP2106X es de de 4 ms.

Hay que añadir además, el tiempo empleado para graficar el espectro, que involucra el despliegue del vector transformado más el desplazamiento de la imagen en la pantalla, para dar el efecto de que la gráfica está corriendo de derecha a izquierda. Ésto demora unos cuantos milisegundos

Las premisas anteriores nos llevaron al desarrollo del sistema. A lo largo del artículo detallaremos más los datos presentados, que fueron medidos en una computadora portátil con procesador Intel Pentium II a 333 MHz, Marca "Acer", modelo "TravelMate 331".

SISTEMA GNU BASADO EN LINUX.

El "software libre" tiene sus orígenes y filosofía en la FSF ("Free Software Foundation"), creada por Richard Stallman. A este organismo también se le denomina GNU ("GNU is Not Unix") (4,5). Aquí se organizan y se supervisan las licencias de uso (copy left).

Uno de los hijos predilectos de GNU, es el sistemas operativo LINUX, que gracias a su madurez de un poco más de diez años, ha dado pie a aplicaciones para el usuario final como GNOME.

GNOME (GNU Network Object Model Environment) desarrollo a cargo del mexicano Miguel de Icaza, tiene entre sus principales componentes: GTK (Gimp Toolkit), GDK (Graphics Drawing kit) y GLIB (Colección de funciones ampliamente usadas por GTK) y éstas fueron el resultado del desarrollo del proyecto The GIMP (GNU Image Manipulation Program), además de usar CORBA (ORBIT) para la intercomunicación de aplicaciones.

GNOME desde el punto de vista del usuario, es un conjunto de aplicaciones que conforman un ambiente integrado de trabajo, muy parecido a lo que ofrece OLE de MS-WINDOWS. Desde la perspectiva del programador es un "framework" para el desarrollo de aplicaciones.

GNOME como "framework" de desarrollo, no está limitado a un lenguaje de programación. Esto quiere decir que tiene mecanismos de desarrollo para diversos lenguajes de programación como Python (6).

Python es un lenguaje de programación interpretado, que maneja los conceptos de orientación a objetos y de módulo. Con el manejo de éste último, se extiende el poder del lenguaje, para incorporarse a su ambiente de desarrollo, por ejemplo los proyectos GTK y GDK.

Otra característica a destacar del "software libre" es su transportabilidad. Esta es consecuencia de la gran comunidad de programadores especialistas en diferentes plataformas y sistemas operativos, que vía la red INTERNET, se involucran en los desarrollos de sistemas comunes. El resultado es, el tener a Linux funcionando en diversas plataformas como SUN o Silicon Graphics, etc. y las herramientas como Python, GTK y GDK las podamos conseguir para ambientes como MS-Windows por citar algún ejemplo.

Esta transportabilidad, no solo se debe al intérprete, sino a la transportabilidad de los sistemas; como las bibliotecas para la creación de interfaces gráficas, el manejo de los dispositivos como las tarjetas de audio y gráficas. Todo esto da como resultado el desarrollo de un sistema que no estará limitado a solo un ambiente de trabajo.

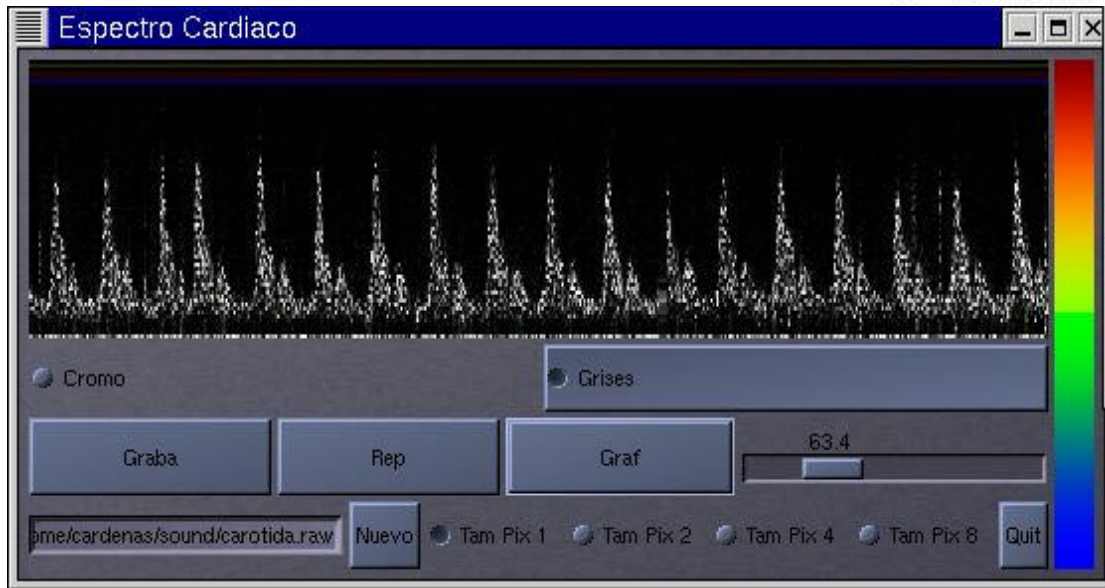
DESCRIPCIÓN DEL SISTEMA

Los requerimientos que debe tener el sistema son:

- Tiempos de la FFT de 1024 puntos complejos del orden de los 80 ms.
- Tarjetas gráficas con aceleración de despliegue, además de las bibliotecas: GDK y OpenGL, que permitan una manipulación directa a su memoria. Funciones como el copiado y la manipulación de bloques de memoria de la propia tarjeta de video.
- La Tarjeta de audio debe tener una resolución mínima de 8 bits y frecuencias de muestreo de al menos 10 KHz. Las características comunes de las tarjetas comerciales hoy en día son: convertidores de 16 bits a una frecuencia de muestreo de 44 KHz, con dos canales (stereo).
- Las bibliotecas y los manejadores de las tarjetas de audio y video, debidamente instaladas en el sistema.

Toda la programación del sistema se hizo en Python. Se requirieron los módulos de GTK, GDK, ESOUND, OpenGL. para Python.

La apariencia del sistema la podemos ver en la siguiente figura:



En donde hay que destacar, que todos los componenetes gráficos, como botones, deslizadoras y demas las proporciona GDK.

RESULTADOS Y ANÁLISIS

En la siguiente tabla, mostramos los tiempos que se emplea para calcular la FFT de diferentes tamaños de vectores de dos algoritmos diferentes, el FFTPack y el FFTW (7). La computadora empleada es una Acer TravelMate 331T a 333 MHz y es un sistema GNU basado en Linux.

Puntos	FFTPack Tiempos en ms.	FFTW Tiempos en ms.
256	0.22	0.35
512	0.42	0.46
1024	0.81	0.69
2048	1.99	1.97
4096	5.66	4.83

Tomando en cuenta estos resultados, vemos que el tiempo para tomar una muestra de 512 puntos y calcular la FFT es del orden de los .46 ms.

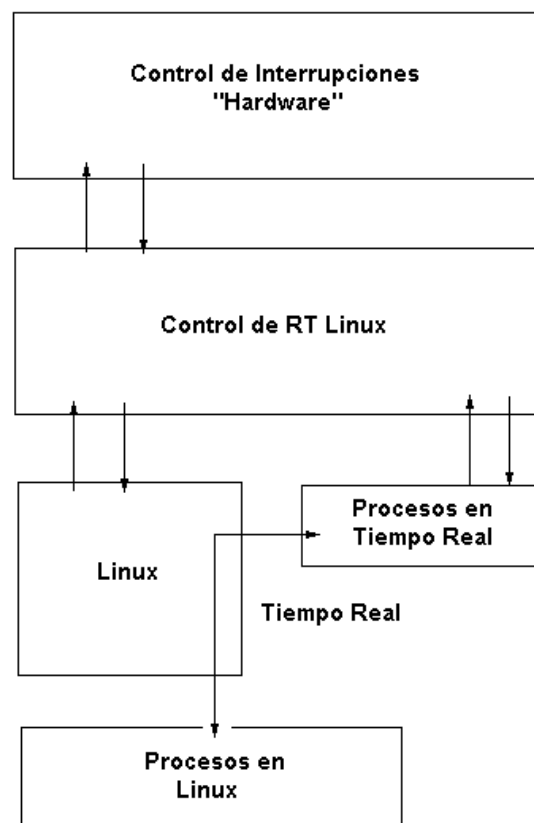
Para medir el tiempo total, el de cálculo y graficación, se hizo directamente en el programa. Poniendo marcas de tiempo, desde que hace la lectura de la tarjeta de sonido, hasta que finaliza el graficado. El tiempo ya promediado despues de hacer 98 bloques de 33 datos cada uno, da un tiempo de 4 ms utilizando el algoritmo FFTW.

CONCLUSIONES

La primera impresión, para el desarrollo de un sistema de adquisición y procesamiento de señales de ultrasonido doppler, utilizando un hardware comercial es buena.

Hay que tomar en cuenta, que el sistema esta en una plataforma multitarea, por lo tanto, tenemos que saber bien que procesos están corriendo en el sistema. En una estación de trabajo personal, esto es controlable, a menos que no se tenga conocimiento de algún proceso que haga muchas llamadas de entrada/salida, como un cron o esté activo netscape.

Si somos exigentes y queremos tener control sobre las llamadas de entrada/salida la opción es usar RTLinux (Real Time Linux), que permite tener un control directo entre las Interrupciones y el Kernel, como lo vemos en la siguiente figura:



REFERENCIAS

(1) Peter Atkinson, John P Woodcock, Doppler Ultrasound and its use in clinical measurement Academic Press 1982

- (2) Jorgen Arendt J. Estimation of blood velocities using ultrasound. Cambridge 1996
- (3) David H. Evans, W. Norman McDicken, Doppler Ultrasound Physics, Instrumentation and Signal Processing. Willey 1999.
- (4) GNU Home Page www.gnu.org
- (5) GNOME Home Page www.gnome.org
- (6) Python Home Page www.python.org
- (7) FFTW Home Page www.fftw.org