# Un sistema empotrado para DSP. Aplicación a sistemas de comunicaciones digitales

F. Segovia*, R. Serrano†, J. M. Górriz*, J. Ramírez* C. G. Puntonet‡ and J. González‡
*Departamento de Teoría de la Señal Telemática y Comunicaciones, Universidad de Granada, (España)
†ATIS (España)
‡Departamento de Arquitectura y Tecnología de Computadores, Universidad de Granada, (España)

***Resumen*—En los últimos años los sistemas embebidos basados en microprocesadores han progresado significativamente. Esto ha posibilitado el uso de potentes sistemas operativos y lenguajes de programación con un alto nivel de abstracción que facilitan el desarrollo de aplicaciones. Este trabajo muestra un sistema de comunicaciones diseñado para funcionar sobre una BeagleBoard, un dispositivo de bajo coste basado en el chip OMAP 3530. El sistema permite manejar señales de audio, compartirlas con otros dispositivos, de forma inalámbrica mediante la tecnología Bluetooth, y filtrar el ruido de las señales siguiendo la metodología de Wiener. Este trabajo supone el punto de partida para la implementación de un conjunto de aplicaciones de procesamiento de señales en la BeagleBoard.**

## I. Introducción

Durante los últimos años se ha producido un gran avance en el desarrollo de los llamados sistemas empotrados basados principalmente en microcontroladores [1], como los de la familia OMAP de Texas Instruments, que presentan procesadores multimedia con un consumo energético mínimo para propósitos específicos [2]. El incremento de potencia de estos procesadores permite trabajar en un nivel de abstracción superior mediante la instalación y configuración en ellos de sistemas operativos potentes como Linux, estableciendo un puente entre la programación clásica de los sistemas empotrados y la de un PC estándar. Este trabajo supone el punto de partida para la implementación de un conjunto de aplicaciones de procesamiento de señales en la BeagleBoard (BB) [2], de propósito específico y utilizable en un amplio repertorio de aplicaciones que van desde los servidores web hasta el procesamiento de vídeo en tiempo real.

La BB fue diseñada con la idea de desarrollar procesadores abiertos tanto en software como en hardware, de manera que se pudiera usar en un entorno educativo demostrando la habilidad de procesamiento del OMAP3530 implementado en chip (véase figura 1). En

Figura 1. BeagleBoard (plataforma de desarrollo)

la actualidad se ha aplicado en una enorme variedad de contextos como son los proyectos en desarrollo [2]:

- **BeagleBot.** La BB aplicada como núcleo de un robot que presenta un carácter autónomo evitando colisiones debidas al uso de sensores de rango ultrasónico.
- **BB Home security.** La BB aplicada al campo de la domótica para definir sistemas de control que monitoricen sensores y cámaras en una residencia para protección industrial y en el hogar.
- **Vehículo todo terreno autónomo.** El objetivo es construir un vehículo autónomo explorando la fusión de sensores en varios niveles que conducen al desarrollo de APIs para varios sensores, planificadores de rutas, constructor de mapas etc.

Para el desarrollo del sistema se dota a la BB de un Sistema Operativo (SO), en este caso GNU/LINUX, que dispone de un conjunto de librerías específicas para la aplicación a desarrollar (véase sección II). Esto permite el ahorro de una gran cantidad de horas de trabajo por parte del programador y aprovechar tanto las funcionalidades, como la potencia de la CPU en cuestión (basada en la arquitectura ARM). En los sistemas embebidos, a

diferencia del PC de sobremesa, la parte hardware suele ser muy específica y restrictiva, por lo que se debe ajustar con gran precisión y eficiencia cada uno de los aspectos del sistema, a fin de minimizar el uso de memoria y el tamaño del sistema operativo y optimizar, al mismo tiempo, el rendimiento y las funcionalidades del mismo. Una vez configurado el SO, se monta una capa superior donde programar las aplicaciones de usuario, en este caso la plataforma JAVA, que es una de las mejores opciones en la actualidad para el desarrollo de software, dado su amplia biblioteca estándar, su capacidad para generar código para ejecutar en distintas plataformas y su flexibilidad y abstracciones de alto nivel. Finalmente se propone, en la sección III, el desarrollo de un sistema de comunicaciones digitales que presenta una etapa de filtrado en receptor basado en la metodología de Wiener [3]. La finalidad de esta aplicación es servir de ejemplo de la potencialidad del marco de desarrollo. El sub-sistema emisor transmite todo tipo de señales por Bluetooth, por ejemplo señales de voz, que se ven afectadas por ruido ambiente; el sub-sistema receptor las recibe y filtra, representándolas en pantalla y analizando la calidad de la respuesta del sistema. El repertorio de algoritmos de filtrado es variado, recogiendo algunos de los algoritmos más importantes en la literatura, como el algoritmo Normalizado de mínimos cuadrados (NLMS) [4], [5]. Estos algoritmos se aplicarán en un repertorio variado de esquemas de filtrado, por ejemplo el problema de supresión de ruido [3], en el que se suprime una interferencia de banda ancha, que contamina la señal deseada. Este proceso se realiza mediante la adquisición de la señal contaminada, su retardo en $k$ instantes en el receptor y su posterior filtrado. El criterio de diseño del filtro es la minimización del error cuadrático medio, definido como la diferencia entre la salida del filtro, que evoluciona adaptativamente siendo un algoritmo iterativo para el conjunto de pesos que lo definen, y la señal contaminada de entrada.

## II. Un sistema empotrado de bajo coste: la BeagleBoard

La tarjeta BB [2] es una plataforma de bajo coste diseñada específicamente para su uso por parte de la comunidad "Open Source". No es una plataforma completa de desarrollo dado no permite acceder a todas las funcionalidades e interfaces del OMAP 3530.

### *II-A. Descripción hardware*

En la figura 2 se muestra el diagrama de bloques de la BB en el que se observan las numerosas opciones

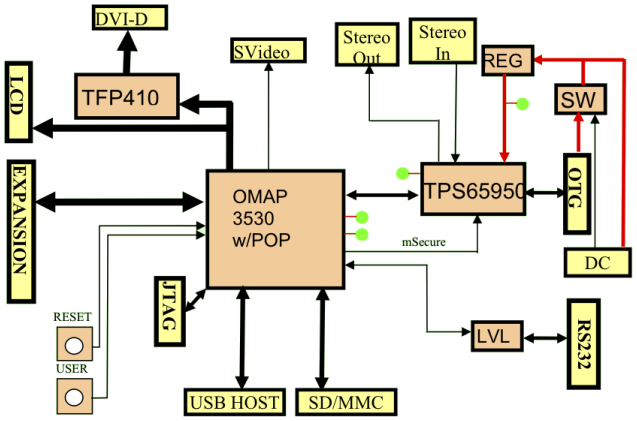
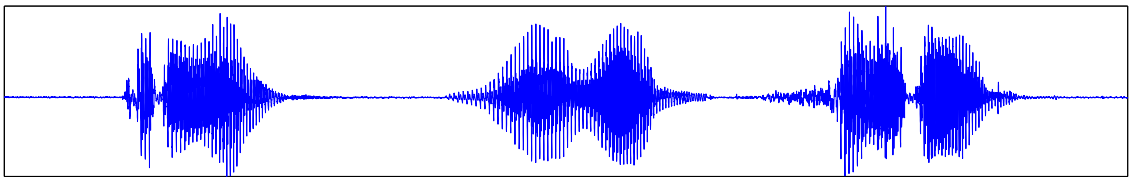



Figura 2. Diagrama de bloques de la BeagleBoard

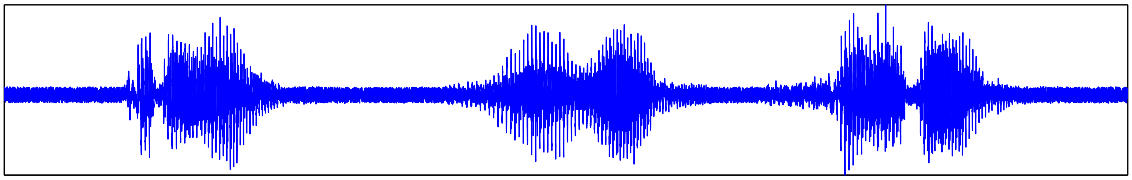
de conectividad que ofrece aproximándose en gran medida a lo que podemos encontrar en un PC y que se controlan a través de un microprocesador de apoyo que se comunica con el microprocesador central mediante el protocolo I2C [6]. El microprocesador central de la serie OMAP (del inglés "Open Multimedia Application Platform") de Texas Instruments está pensado para el desarrollo de aplicaciones multimedia en dispositivos móviles [7]. Estos microprocesadores actualmente se utilizan en multitud de teléfonos móviles de alta gama, como la serie N de Nokia. Dentro de la serie OMAP, la OMAP3 es la familia de mayores prestaciones, contando con procesadores suficientemente potentes como para ejecutar sistemas operativos de un peso considerable, como Linux. El OMAP 3530 cuenta en un solo chip con un núcleo ARM Cortex A8 funcionando a 720 MHz, una GPU, un DSP y un ISP, que lo hacen indicado para aplicaciones tales como el procesamiento de vídeo en tiempo real.

### *II-B. Descripción software*

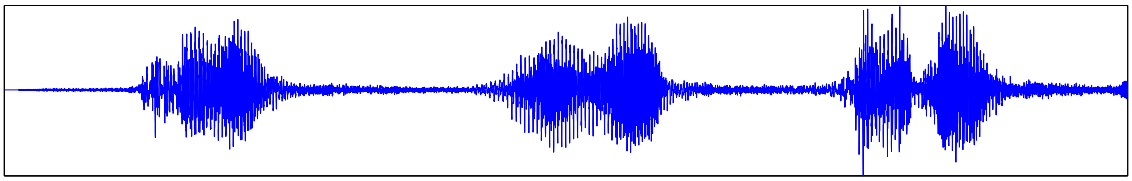
Para el arranque de un sistema operativo libre en esta placa de desarrollo se opta por la distribución Ånström [8], una novedosa distribución creada por un pequeño grupo de gente que han trabajado en proyectos como OpenEmbedded, OpenZaurus y OpenSimpad, intentando unir esfuerzos para conseguir una distribución estable y amigable con el usuario para un gran abanico de dispositivos empotrados como es el caso de la BB. Se han incluido las librerías básicas para su funcionamiento, incluyendo un entrono grafico Gnome, librerías para manejo de dispositivos bluetooth como blu-z y la maquina virtual de Java para el funcionamiento del programa desarrollado.



(a) Seña de audio de ejemplo. La transcripción fonética es "tres, nweβe, θero"



(b) Señal (a) tras añadirle ruido uniforme



(c) Señal (b) tras procesada usando el esquema de filtrado LMS

Figura 3. Funcionamiento del sistema de comunicaciones desarrollado.

La BB utiliza como disco de almacenamiento, una tarjeta SD de 4 GB, en la que se ha instalado la distribución libre Ånström usando dos particiones de disco realizadas con el software Gparted. La primera partición del tipo Fat32 con una capacidad de 100Mb, donde se encuentra el archivo de carga del SO que permite una vez se conecta la BB arrancar el SO Ånström que se instala en la segunda partición. La segunda partición ocupa el resto de la capacidad de la tarjeta del tipo EXT3 donde se encuentra todo el SO y la aplicación de filtrado y comunicaciones desarrollada. Para la correcta instalación del SO en la tarjeta SD de la BB, se utiliza el SO Ubuntu en su versión 11.10. Para ello se descomprime el contenido del archivo con la imagen del SO Angström en la segunda partición de la tarjeta SD usada en la BB y se copian los archivos arranque que se integran dentro de la compilación en la primera partición de la tarjeta SD por orden estricto: MLO, Uboot y Uimage.

## III. DSP MEDIANTE LA BEAGLEBOARD

Tradicionalmente el procesamiento digital de señales (DSP) suele llevarse a cabo sobre un hardware especializado, debido al propósito del mismo y al elevado número de operaciones numéricas que requiere. Un hardware optimizado es especialmente importante cuando el procesamiento de la señal se realiza en tiempo real, como es el caso, por ejemplo, de los procedimientos de cancelación de ruido que realiza un teléfono móvil. El chip OMAP3530 integrado en una BB, junto con el software adecuado, proporcionan la capacidad de cómputo necesaria para el procesamiento de señales, además de algunas ventajas propias de los sistemas empotrados, como son su bajo coste y unos requerimientos energéticos contenidos.

### *III-A. Modelado del canal*

Las señales de audio pueden verse afectadas por diferentes formas de ruido debido, por ejemplo, al ruido ambiente existente durante la captura de la señal o al canal de comunicaciones en caso de que la señal haya sido transmitida de un dispositivo a otro, etc. El sistema de comunicaciones desarrollado permite simular estos ruidos y añadirlos artificialmente a las señales. Más concretamente hemos definido dos tipos de ruido:

- **Ruido uniforme.** Los valores del ruido se generan aleatoriamente y con la misma probabilidad dentro de un intervalo preestablecido.
- **Ruido gaussiano.** Los valores del ruido se generan a partir de una distribución gaussiana de media y desviación típica determinada.

Tanto el intervalo, en el caso del ruido uniforme, como la media y desviación típica, en el caso del ruido gaus-


Figura 4. Ventana principal de la aplicación

siano, están parametrizados y pueden ser modificados por el usuario. No obstante, para evitar que el ruido sea tan elevado que oculte la señal, estos parámetros serán relativos a los de la señal.

La figura 3 contiene una representación de tres señales de audio que ilustran el funcionamiento del sistema. La primera (rectángulo superior) es un sonido de la base de datos SpeechDat-Car [9], [10], cuya transcripción fonética es "tres, nweβe, θero". La segunda (rectángula central) es una señal ruidosa formada al añadir ruido uniforme a la primera señal. En este caso el ruido está compuesto por valores procedentes de una distribución uniforme es el rango $[-x, x]$, donde $x = \frac{1}{2} * (S_{max} - S_{min})$ y $S_{max}$ y $S_{min}$ son el valor mayor y el valor menor de la señal respectivamente. Por último, la tercera señal (rectángulo inferior) se obtiene al procesar la señal anterior usando un esquema de filtrado LMS.

### *III-B. Filtrado adaptativo*

El filtrado de una señal tiene como finalidad mejorar su relación señal-ruido (SNR) reduciendo el ruido de la misma. Existen en la literatura diferentes métodos para realizar este proceso, si bien en este trabajo nos hemos centrado en los filtros adaptativos más sencillos, que van ajustando sus parámetros iterativamente a lo largo del tiempo en base a un criterio de minimización del error cuadrático medio. Más concretamente, hemos implementado una variante de los filtros LMS, ampliamente usados para este propósito. En líneas generales, estos filtros constan de dos procesos básicos, que se repiten iterativamente:

- Un proceso de filtrado que implica:
  - Generar la salida del filtro lineal en respuesta a la señal de entrada.
  - Calcular una estimación del error comparando esta salida con la respuesta deseada
- Un proceso adaptativo que ajusta los parámetros del filtro automáticamente de acuerdo al error estimado en el paso anterior.

Una característica significativa de la familia de algoritmos LMS es su simplicidad, ya que no requiere realizar mediciones de las funciones de correlación, ni inversiones matriciales. A continuación mostramos algunas variantes de estos algoritmos:

- **LMS Estándar.** Es el primer algoritmo de esta familia que se desarrolló y también el más simple [11]. Su funcionamiento es el siguiente:
  1. Establecer $\widehat{\mathbf{w}}_0 = \mathbf{0}$ a menos que se disponga de los coeficientes del filtro $\widehat{\mathbf{w}}_k(n)$. En tal caso, usarlos para seleccionar un valor apropiado para $\widehat{\mathbf{w}}_0$.
  2. Para $n = 0, 1, 2, ...$, calcular:

$$e(n) = d(n) - \widehat{\mathbf{w}}_n^H \mathbf{u}(n) \quad (1)$$

$$\widehat{\mathbf{w}}_{n+1} = \widehat{\mathbf{w}}_n + \mu \mathbf{u}(n) e^*(n) \quad (2)$$

donde $d(n)$ es la señal deseada y $\widehat{\mathbf{w}}_n^H \mathbf{u}(n)$ es la salida del filtro, que se calcula como el producto interno entre el vector de coeficientes del filtro $\widehat{\mathbf{w}}_k(n)$ y la señal de entrada $\mathbf{u}(n)$. El

superíndice $H$ denota transpuesta conjugada y el asterisco conjugación.

- **LMS Normalizado (NLMS).** Esta variante normaliza la expresión $\mu\mathbf{u}(n)e^*(n)$ mediante $\|\mathbf{u}(n)\|^2$, modificando la magnitud (pero no la dirección) del vector de gradiente estimado, lo cual provoca que la convergencia del algoritmo sea más rápida [12]. La ecuación 2 modificada es, por tanto:

$$\widehat{\mathbf{w}}_{n+1} = \widehat{\mathbf{w}}_n + \mu\frac{\mathbf{u}^*(n)e(n)}{\|\mathbf{u}(n)\|^2} \tag{3}$$

  donde $\mu$ es el tamaño del paso, $e(n)$ es el error en el instante $n$ y $\mathbf{u}(n)$ es la señal de entrada del filtro.
- **LMS normalizado por momento (MNLMS).** Al igual que la variante anterior, esta variante normaliza la expresión $\mu\mathbf{u}(n)e^*(n)$ [13], quedando la ecuación 2 de la siguiente forma:

$$\widehat{\mathbf{w}}_{n+1} = \widehat{\mathbf{w}}_n + \frac{\mu\mathbf{u}^*(n)e(n)}{\mu\|\mathbf{u}(n)\|^2 + 1} \tag{4}$$

  donde $\mu$ es el tamaño del paso, $e(n)$ es el error en el instante $n$ y $\mathbf{u}(n)$ es la señal de entrada del filtro.

Los filtros adaptativos, van adaptando iterativamente sus coeficientes con el fin de que la señal de salida (señal filtrada) se parezca cada vez más a la señal deseada (señal sin ruido). Una forma de evaluar esta adaptación es mediante el error cuadrático instantáneo $e_c(n)$, que se define como sigue:

$$e_c(n) = (d(n) - u(n))^2 \quad \forall n \tag{5}$$

donde $d(n)$ y $u(n)$ son la señal deseada y la señal de salida respectivamente.

### *III-C. Sistema de comunicaciones*

La figura 4 muestra el sistema de comunicaciones desarrollado. Ha sido implementado para procesar señales de audio y permite visualizar simultáneamente tres señales: i) una señal de partida, ii) la señal de partida con ruido aditivo, y iii) esta última señal una vez filtrada. Está diseñado para, partiendo una señal de audio almacenada en formato Wave [14], comprobar los efectos del ruido, tanto uniforme como Gaussiano y aplicarle uno o más filtros de forma de que se suavicen los efectos del ruido. Además calcula el error cuadrático instantáneo del proceso de filtrado, según se ha definido en la ecuación 5.

El sistema permite el envío de señales entre diferentes dispositivos. Dicha transmisión se realiza mediante Bluetooth, una tecnología concebida para la creación de redes inalámbricas de área personal (WPANs), esto

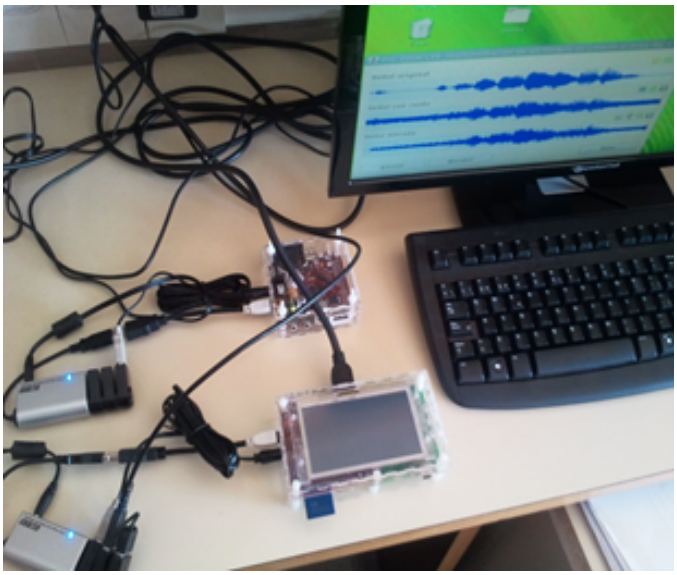


Figura 5. Sistema de comunicaciones funcionando *in situ*

es, redes de corto alcance destinadas a conectar dispositivos próximos entre sí. Dado que nuestro sistema está destinado a utilizarse en recintos de tamaño reducido (típicamente un aula o laboratorio de una facultad o escuela universitaria), la limitación del protocolo Bluetooth relativa a la distancia entre los dispositivos no supone un problema y, en cambio, sí que aporta una ventaja importante: permite crear redes Ad-hoc, sin necesidad de dispositivos adicionales (como puntos de acceso, routers, etc.).

## IV. Conclusiones

Se ha presentado el desarrollo de un sistema de comunicaciones digitales sobre un dispositivo empotrado. El dispositivo usado ha sido una BeagleBoard [2], basada en el chip OMAP3530 de Texas Instruments. El sistema se ha desarrollado en varias capas. En primer lugar se ha preparado una versión mínima del sistema operativo GNU/Linux capaz de ejecutarse en dicho hardware. A continuación se ha instalado una máquina virtual de Java que ha proporcionado un entorno de programación de alto nivel y las bibliotecas necesarias para manejar el dispositivo. Por último, usando el lenguaje de programación Java, se ha implementado una aplicación que, mediante una interfaz gráfica, es capaz de manejar señales de audio, añadir diferentes tipos de ruido a las mismas y filtrar dicho ruido usando algoritmos basados en la metodología de Wiener [3]. Además, el sistema permite el envío de señales entre diferentes dispositivos mediante Bluetooth. Existen numerosas alternativas para la extensión y mejora del sistema desarrollado en este

trabajo, algunas de las cuales están siendo ya abordadas en la actualidad. Por un lado, cabe la posibilidad de extender el sistema dotándolo de la capacidad de obtener las señales de audio a partir de un micrófono e implementando otros tipos de comunicaciones, como redes WIFI. Otro objetivo, más ambicioso, consistiría en extender su funcionamiento para trabajar con imágenes además de señales de audio o, incluso, dotarlo de una interfaz web que permita usar el sistema a través de Internet.

## AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el MICINN bajo el proyecto TEC2008-02113, la Consejería de Innovación y Ciencia y Empresa (Junta de Andalucía, España) bajo los proyectos de excelencia P07-TIC-02566, P09-TIC-4530 y P11-TIC-7103, y la Universidad de Granada bajo el proyecto de Innovación Docente "IMPLEMENTACIÓN DE UN MÓDULO DE COMUNICACIONES DIGITALES" (Ref 09-139).

## REFERENCIAS

[1] S. Kuo, B. H. Lee, and W. Tian, *Real-time digital signal processing: implementations and applications*. John Wiley & Sons, Jun. 2006.

[2] "Beagleboard.org." [Online]. Available: http://beagleboard.org/

[3] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*, 1st ed. Wiley, Apr. 1996.

[4] J. M. Górriz, J. Ramírez, S. Cruces-Alvarez, C. G. Puntonet, E. W. Lang, and D. Erdogmus, "A novel LMS algorithm applied to adaptive noise cancellation," *IEEE Signal Processing Letters*, vol. 16, no. 1, pp. 34–37, Jan. 2009.

[5] J. M. Górriz, J. Ramírez, S. Cruces-Alvarez, D. Erdogmus, C. G. Puntonet, and E. W. Lang, "Speech enhancement in discontinuous transmission systems using the constrained-stability least-mean-squares algorithm." *Journal of the Acoustical Society of America*, vol. 124, no. 6, pp. 3669–3683, 2008.

[6] D. Paret, *El bus 12C: de la teorÃa a la prÃ¡ctica*. Paraninfo, 1995.

[7] G. Martin and F. Schirrmeister, "A design chain for embedded systems," *Computer*, vol. 35, no. 3, pp. 100–103, Mar. 2002.

[8] "The ångström distribution." [Online]. Available: http://www.angstrom-distribution.org/

[9] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen, "SpeechDat-Car. A large speech database for automotive environments," in *International Conference on Language Resources & Evaluation*, 2000.

[10] J. M. Górriz, J. Ramírez, E. W. Lang, and C. G. Puntonet, "Hard c-means clustering for voice activity detection," *Speech Commun.*, vol. 48, no. 12, pp. 1638–1649, Dec. 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1222682.1223178

[11] S. Haykin, *Adaptive Filter Theory*, 4th ed. Prentice Hall, Sep. 2001.

[12] Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. Eugene Dong, and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, 1975.

[13] S. C. Douglas and T. H. Meng, "Normalized data nonlinearities for LMS adaptation," *IEEE Transactions on Signal Processing*, vol. 42, no. 6, pp. 1352–1365, Jun. 1994.

[14] E. Fleischman, "WAVE and AVI codec registries," http://tools.ietf.org/html/rfc2361. [Online]. Available: http://tools.ietf.org/html/rfc2361