Upzmg

cesar cordero

univercidad de la zona metropolitana de guadalajara

Sistema lógico anti retroalimentación

Sevilla cordero cesar Arturo /Joel de Jesús López Ascencio /Jonathan Alejandro Capuchino González

Índice

Contenido

[Introducción 2](#_Toc16798906)

[Objetivos 2](#_Toc16798907)

[Información Introductoria al tema 2](#_Toc16798908)

[Metodología y organización de la memoria 3](#_Toc16798909)

[Etapas 3](#_Toc16798910)

[Tecnologías utilizadas 4](#_Toc16798911)

[Que es un DSP 4](#_Toc16798912)

[Buffers sobre dsp 5](#_Toc16798913)

[Arquitectura de un DSP sobre lógica difusa en comparación a frecuencias y salidas 5](#_Toc16798914)

[El problema del feedback acústico 6](#_Toc16798915)

[Métodos de control automático del feedback acústico. 9](#_Toc16798916)

[Métodos de modulación de fase (PM) 9](#_Toc16798917)

[Métodos de reducción de ganancia en curvas a través de la toma de decisiones 9](#_Toc16798918)

[Control de feedback en frecuencias bajas 10](#_Toc16798919)

[Métodos de filtrado espacial. 10](#_Toc16798920)

[Diseño e implementación. 10](#_Toc16798921)

[Diseño del prototipo usando MATLABv 11](#_Toc16798922)

[Transformada de Hilbert mediante Filtro FIR 12](#_Toc16798923)

### Introducción

Desde la aparición de los sistemas de sonorización y PA (public addres), el feedback acústico ha sido un problema. Siempre que se utiliza un micrófono para captar una señal de audio, que luego es procesada, amplificada y vuelta a reproducir a través de la PA, es inevitable que la señal del altavoz sea realimentada al micrófono. Cuando esto ocurre, se crea un bucle cerrado que afecta al rendimiento del sistema, deteriorando la calidad del sonido y limitando la amplificación. Dentro de los efectos producidos por el feedback, el acople de ciertas frecuencias o howling effect es el más molesto.

### Objetivos

Resumen Esta memoria recoge información detallada acerca del proceso de diseño en lógica difusa, desarrollo e implementación del presente Trabajo Fin de Grado (TFG).

El objetivo del proyecto es la implementación de un algoritmo anti-feedback acústico digital utilizando un microprocesador especializado en el procesamiento de señales digitales: DSP. En concreto se trabaja con dos tipos de DSP, denominados Blackfin y SHARC.

* El estudio e implementación del algoritmo se ha realizado en dos fases:
* la primera fase, el prototipo, para el cual se ha utilizado el programa MATLAB, y
* la segunda, implementación del algoritmo final en el DSP, para la cual se ha utilizado el programa CrossCore de Analog Devices.

### Información Introductoria al tema

Este efecto es debido a las reflexiones del sonido producidas por las características de la sala, y no al sonido directo emitido por los altavoces o PAs, ya que en los escenarios se intenta que estos componentes y el micrófono no se encuentren enfrentados.

El control del feedback acústico se refiere al proceso de intentar anular o reducir lo máximo posible el efecto de feedback acústico. Normalmente, el encargado de que no se produzca es el técnico de sonido que, mediante un ecualizador paramétrico y técnicas de ecualización, busca y corrige manualmente las frecuencias afectadas.

Gracias a la evolución tecnológica y del mundo digital, es posible diseñar e implementar un algoritmo en un microprocesador que se encargue de esta tarea, llegando al punto de que no haya que preocuparse por este efecto tan molesto. En este proyecto se trata, desde el punto de vista teórico-práctico, el proceso de diseño e implementación de un algoritmo capaz de reducir el efecto de feedback lo máximo posible, teniendo en cuenta las características del dispositivo en el que se va a implementar y respetando la mejor relación posible entre calidad de audio y carga computacional del DSP.

## Metodología y organización de la memoria

Después de estudiar que necesidades cubrir, se ha dividido en bloques funcionales o subsistemas y se ha definido un listado de objetivos.

Primero se ha decidido programar el prototipo en un entorno controlado donde se puedan realizar simulaciones, con el fin de comprobar el buen funcionamiento del código y elegir las características adecuadas.

Al tener esta parte acabada, se ha continuado con su diseño esta vez sí, en el DSP, reprogramando el código utilizado para adaptarlo a las características de programación de cada DSP. Como cada uno de los dos DSP utilizados esta optimizado para un formato de programación (fixed-point o floating-point), se ha desarrollado un código distinto para cada uno, buscando las funciones necesarias y su implementación en los manuales de cada dispositivo. Finalmente, se han programado los dos dispositivos para probar su correcto funcionamiento y pulir detalles.

### Etapas

Tras la exposición de la metodología y organización del proyecto, se pretende comentar las diferentes etapas por las que se ha tenido que pasar desde el comienzo hasta la finalización del trabajo.

1. Investigación sobre los DSP y CrossCore.

2. Organización y búsqueda de las necesidades.

3. Elección de elementos hardware.

4. Investigación y aprendizaje sobre la programación en tiempo real de audio.

5. Familiarización con el lenguaje utilizado mediante prácticas ajenas al proyecto.

6. Investigación y aprendizaje sobre la teoría de control de feedback acústico y los distintos métodos existentes.

7. Elección del método de control de feedback por sus características.

8. Planteamiento del prototipo y su implementación en un entorno controlado (Matlab).

9. Investigación, programación y comprobación de diferentes algoritmos en Matlab. Realizando las pruebas necesarias para obtener información suficiente para la elección del algoritmo final.

10. Implementación del algoritmo, primero en el DSP Blackfin y luego en el SHARK, mediante CrossCore IDE.

11. Realización de pruebas y obtención de resultados.

### Tecnologías utilizadas

En este apartado se estudian las distintas tecnologías que forman parte del proyecto, tales como componentes hardware, software, entornos de programación, etc…

### Que es un DSP

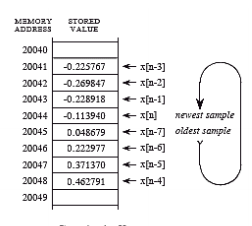
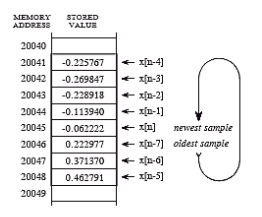
DSP (Digital Signal Processing) puede hacer referencia bien al área de la ciencia computacional que trabaja con un único tipo de datos, las señales, o bien al dispositivo especializado en esta tarea. En este proyecto, cuando se habla de DSP, se hace referencia al segundo término. Un dispositivo DSP es principalmente un microprocesador que se encarga de procesar señales digitales, pero consta de diversas cualidades que lo diferencian del resto de microprocesadores.

Los DSPs realizan de forma rápida y repetida operaciones matemáticas basadas en sumas y multiplicaciones (filtros digitales, FFT, …), y están especializados en el procesamiento de señales a “tiempo real”. Una señal de audio es continuamente convertida a digital en un ADC, procesada por el DSP y convertida de nuevo a forma analógica mediante un DCA.

Esto es lo que diferencia los DSP del resto de microprocesadores de propósito general, que realizan fundamentalmente tareas de manipulación de datos (por ejemplo, procesadores de texto o bases de datos) que pueden tomar más o menos tiempo de procesamiento.

A su vez, un DSP no es capaz de realizar otro tipo de tareas que otros microprocesadores pueden, pero es muy caro fabricar procesadores en todo tipo de tareas y por eso existen procesadores especializados como los DSPs. Otra cualidad es que pueden trabajar muestra a muestra o por bloques de muestras, cosa que afectará a la forma de trabajar con cada DSP, pero se hablará de esto más tarde. Dicho esto, para procesar rápidamente, por ejemplo, un filtro FIR, se necesita algo más que operaciones de sumas y restas. Es necesario el acceso rápido a los N últimos datos de entrada y a los N coeficientes del filtro, para ello se utiliza un buffer circular. Un buffer circular es una memoria temporal que se rellena de datos y sobrescribe los datos más antiguos, de esta forma no se sobrecarga la memoria.

### Buffers sobre dsp

### Arquitectura de un DSP sobre lógica difusa en comparación a frecuencias y salidas

Tomando como ejemplo un filtro FIR, uno de los procesos más típicos en el procesamiento de señales, es necesario que en un solo ciclo de reloj se realicen 4 accesos a memoria:

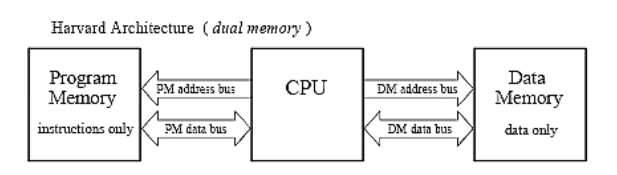
1. Tomar la instrucción sobre las variables obtenidas

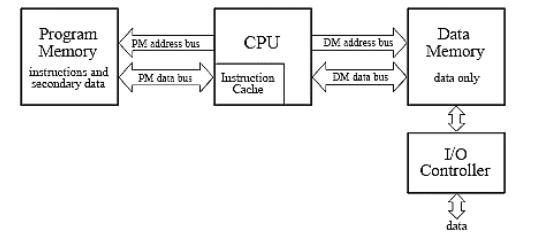
2. Acceso a un dato, en línea de retardo. (fuzificacion)

3. Acceso a dato del coeficiente del filtro.(deszusificacion)

4. Desplazamiento de posición del dato en la línea de retardo.(salida desfuzificada)

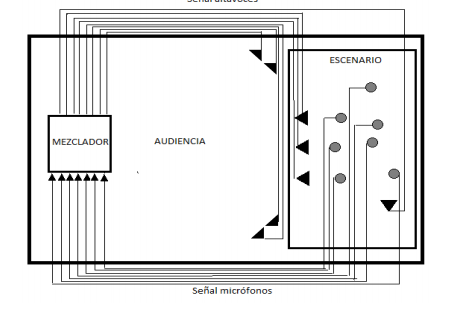
Debido a esto el acceso a la memoria debe ser rápido y eficiente, y que permita acceder a más de un dato simultáneamente.





### El problema del feedback acústico

En la siguiente figura se muestra la distribución típica de un sistema de PA en un escenario/conferencia.



Múltiples micrófonos están orientados para captar distintas fuentes de sonido de interés. La señal de audio captada por los micrófonos es dirigida a un mezclador, donde 20 se realizada el procesamiento necesario (amplificación, efectos, compresión, etc.) para después enviarla a los altavoces, que suelen estar organizados en grupos de altavoces que emiten señales distintas, ya que no es necesario emitir la misma señal hacia los músicos que hacia el público.

Normalmente, los micrófonos y los altavoces suelen estar orientados de forma que, teniendo en cuenta su directividad, la señal emitida por los altavoces no sea recibida directamente por los micrófonos. Aun así, es inevitable que debido a las reflexiones producidas en la sala no se produzca una realimentación.

Estas reflexiones constituyen un acoplamiento acústico indirecto entre los altavoces y los micrófonos.

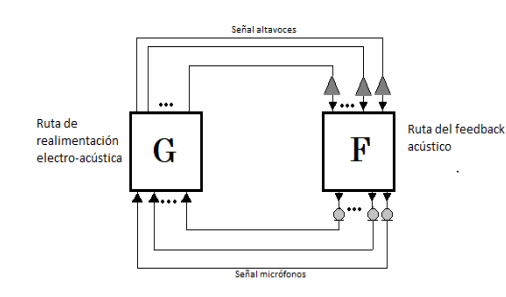


Diagrama de como sucede la retroalimentación

Cronograma

el sistema de PA es con un modelo en tiempo discreto como se muestra en la imagen anterior . De esta forma se pueden diferenciar dos rutas que recorre la señal de audio: la ruta del feedback acústico, que incluye las reflexiones de la señal en la sala hasta que es realimentada en los micrófonos, y la ruta electro-acústica, que comprende desde que la señal es captada por los micrófonos, procesada y amplificada para su retorno a los altavoces. Esta representación está basada en un sistema multicanal, con múltiples micrófonos y altavoces. Para la realización del proyecto se tiene en cuenta un modelo de un solo canal, ya que todos los métodos de control de feedback propuestos hasta la fecha son aplicables a solo un canal.

Hasta ahora se podría suponer que:

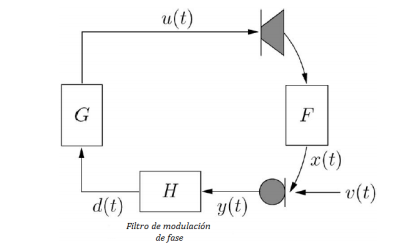
• El altavoz tiene respuesta lineal y plana.

• El micrófono tiene respuesta lineal y plana.

• La ruta de amplificación tiene respuesta lineal y plana (suponiendo que no se aplica ningún procesamiento con respuesta infinita)

• La ruta de feedback tiene respuesta lineal pero no plana.

Bucle de un solo canal.



### Métodos de control automático del feedback acústico.

En este apartado se habla de los diferentes métodos de control del feedback acústicos en el contexto de un sistema de un solo canal, con el fin de explicar brevemente las diferentes técnicas utilizadas, al igual que las ventajas y desventajas de cada uno. Los métodos automáticos de control del feedback se pueden clasificar en cuatro clases.

### Métodos de modulación de fase (PM)

Una de las primeras propuestas para el control del feedback acústico consiste en el desplazamiento de frecuencia o FS (Frequency shifting) de la señal del micrófono antes de ser amplificada y enviada a los altavoces

Se ha comprobado en diversos experimentos que, la distancia frecuencial media entre dos picos de magnitud en la respuesta de una sala, es aproximadamente de 10Hz, por lo tanto, el valor óptimo de FS se espera que sea alrededor de 5 Hz. Existen otras técnicas, como el uso de modulación de fase en la ruta electro-acústica con el objetivo de evitar la condición de fase en el criterio de Nyquist, u otras como la modulación de amplitud (AM) o modulación del retardo (Delay Modulation, DM). La realización de este proyecto se basa principalmente en la técnica de modulación de la frecuencia (FS) usando un filtro FIR de Hilbert, aunque también se han probado alguna de las otras. Se ha elegido esta técnica debido a su robustez y sencilla implementación comparada con los siguientes métodos, aunque cuenta con algunas desventajas de las que se hablará más adelante, donde se explica esta técnica con más detenimiento.

### Métodos de reducción de ganancia en curvas a través de la toma de decisiones

El enfoque más claro para el control del feedback es automatizar las acciones que un técnico haría para prevenir o eliminar el acople en sistemas de refuerzo de sonido. Estas acciones normalmente consisten en reducir la ganancia de la ruta electro-acústica mediante ecualización.

Dependiendo del ancho de la banda de frecuencias en la cual se reduce ganancia, se pueden distinguir entre tres métodos de reducción de ganancia:

### Control de feedback en frecuencias bajas

1. Métodos de control automático de ganancia(AGC), los primeros de este tipo en estudiarse. Si se detecta acople en alguna frecuencia se reduce la ganancia de toda la banda de frecuencias, y después de un intervalo de tiempo la ganancia se restaura a su estado inicial. Con los años se han propuesto mejoras que permiten realizar la reducción de ganancia en sub bandas de frecuencia más centradas en donde se produce el acople. La mayor fortaleza de este método es su fiabilidad, ya que si la ganancia se reduce lo suficiente se asegura que el sistema sea estable
2. Métodos de ecualización automática(AEQ), derivan de la mejora del método anterior. Si se detecta acople, la reducción de ganancia se aplica solo a las sub bandas donde se produce.
3. Métodos NHS, se reduce la ganancia en bandas estrechas de frecuencias (filtros noch) donde se produce el acople, frecuencias donde el bucle de ganancia es cercano a la unidad.

### Métodos de filtrado espacial.

La meta de estos métodos es alterar la respuesta del bucle 𝐺(𝜔,𝑡)𝐹(𝜔,𝑡) del sistema usando arrays de micrófonos y altavoces en los cuales la señal recibida/transmitida se procesa mediante los llamados beamforming filters, que se podría traducir como filtros formadores de haces.

El objetivo general es un array de micrófonos en los que el lóbulo principal de captación esté dirigido a la fuente mientras se tiene un nulo en la dirección del altavoz, o lo mismo con un array de altavoces, donde tenga máxima directividad hacia la audiencia y un nulo en la dirección de los micrófonos.

### Diseño e implementación.

Para diseñar, desarrollar e implementar este proyecto, se ha tenido que pasar por diversas etapas hasta conseguir el resultado final. Principalmente el desarrollo de este proyecto se puede dividir en dos etapas, la etapa de desarrollo de los algoritmos prototipo, y la etapa de implementación en DSP del algoritmo.

Aun así, antes de empezar con cada una de las etapas principales, ha hecho falta dedicar tiempo al aprendizaje de los distintos programas, teoría necesaria y formas de programación.

Se comenzó por investigar acerca de CrossCore y del DSP Blackfin, para obtener una idea de sus posibilidades y las características propias de la programación a tiempo real. Tanto el software y hardware ha sido proporcionado por la UPV, por lo tanto, no ha habido necesidad de realizar una búsqueda de cuales se iban a utilizar o cual era su coste.

Así pues, una vez obtenido el DSP y el software, se empezó a aprender el lenguaje de CrossCore mediante la realización de programas básicos encontrados en libros en la web, pero sobretodo mediante la realización de las prácticas impartidas en 31 la asignatura de Electrónica Aplicada al Audio (EAA). Siguiendo la estructura de la asignatura, realizando pequeños programas de efectos de audio y trabajando con líneas de retardo, fue posible coger agilidad, práctica y soltura a la vez que se iba adentrando más en el lenguaje.

Cuando se consideró que el nivel adquirido era suficiente, se empezó con el estudio de los algoritmos anti-feedback, su comprensión y la forma de implementarlos mediante el lenguaje de programación. Tras el estudio del documento de van Waterschoot y Moonen, citados anteriormente, y por la recomendación del tutor del proyecto, se eligió el método de control de feedback acústico basado en el PFC, más específicamente se tomó como referencia el método de FS, con el cual se ha trabajado hasta la implementación final.

Llegados a este punto se observó que había distintas formas de programar el algoritmo, sobre todo a la hora de realizar las transformada de Hilbert, que puede ser obtenida mediante diversos métodos.

Así, teniendo métodos distintos, era necesario elegir el que mejor rendimiento y calidad era capaz de obtener, para ello se llegó a la conclusión de que CrossCore no era la plataforma más adecuada para esta tarea, ya que con él no puedes obtener datos objetivos sobre el funcionamiento y rendimiento del algoritmo implementado.

### Diseño del prototipo usando MATLABv

Como ya se ha mencionado anteriormente, el programa con el que se va a implementar el código que se introducirá en el DSP es el CrossCore, pero antes de implementar el código en esa plataforma especializada para programación en tiempo real, se decidió realizar un prototipo en un entorno controlado, en el que fuera posible realizar todas las pruebas necesarias para seleccionar el modelo de algoritmo que mejor se adapte después al DSP según sus limitaciones computacionales, comentadas anteriormente.

El programa elegido es Matlab, una de las herramientas matemáticas más potentes.

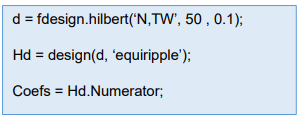
Gracias a Matlab se pueden generar diferentes tipos de señales y procesarlas con los diferentes algoritmos que se vayan a usar, a la vez que se obtienen resultados gráficos y numéricos que facilitan la tarea de análisis de los distintos algoritmos.

El único inconveniente es que no se trata de programación a tiempo real por lo que el código tiene que ser adaptado posteriormente para que su uso sea apto en el DSP.

### Transformada de Hilbert mediante Filtro FIR

Para esta función, Matlab cuenta con diversas funciones capaces de implementar el filtro FIR. La función utilizada fue fdesign.hilbert, que permite introducir dos parámetros, el número de coeficientes y el ancho de banda de transición (TW).

Esta función diseña el filtro con los parámetros introducidos y guarda los coeficientes, que se pueden obtener añadiendo .Numerator a la variable donde se guarda el filtro.



Ahora los coeficientes de un filtro de Hilbert de 50 coeficientes y un ancho de banda de transición de 0.1 se encuentran guardados en la variable Coefs.

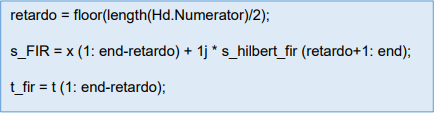
Como era necesario obtener un archivo externo con el valor de los coeficientes, se utilizó la función save, que permite crear un fichero .txt con los valores de los coeficientes en formato ASCII.



A la hora de diseñar el filtro FIR, se tuvo que tener en cuenta un factor muy importante, el retardo generado por el filtro. Siempre que se implementa un filtro FIR, la señal de salida se encuentra retarda N/2 muestras, donde N es el número de coeficientes del filtro.

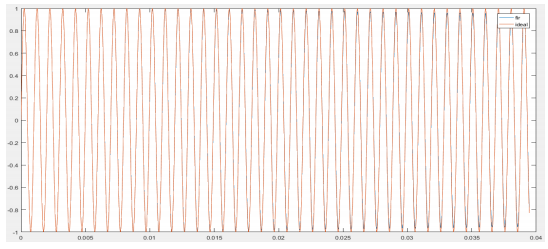
La corrección es necesaria, sobretodo en Matlab, ya que las dimensiones de los vectores tienen que ser las mismas si se desea operar con ellos o representarlos gráficamente.

Para ello, se tuvo que eliminar las N/2 ultimas muestras de la señal original y del vector de tiempo, y las N/2 primeras muestras de la señal procesada (transformada).

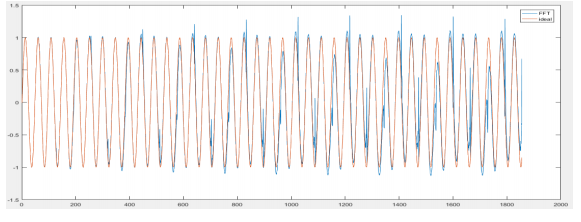


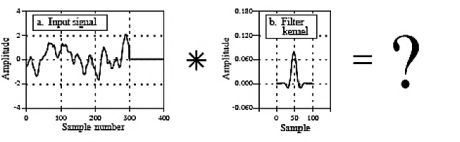
Modelado difuso con las variables echas en mat lab

Barrido y donde puede guardar en cada canal en el sistema de memorias de un dsp.



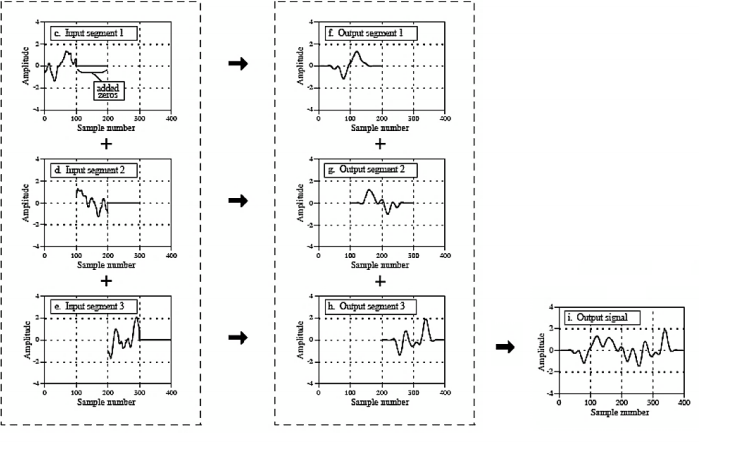
Compensación de filtrado de las graficas de la entada y la salida y la corrección.





Variables donde de acuerdo al sistema en donde va a retro alimentar sistema

En la toma de decisiones donde y como va aplicar los filtros de acuerdo al sistema y la condiciones de muestreo en las memorias.



En esta parte podemos ver como el sistema puede tomar una decisión con las variables de salida.

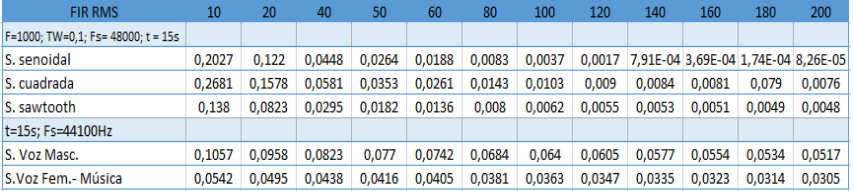
Antes de implementar este método, se decidió realizar primero las pruebas con la función de Matlab fftfilt(D,x,Nfft) donde D es el filtro FIR, x la señal de entrada y Nfft el número de puntos de la fft. Tras realizar las pruebas, se observó que los picos de la señal habían desaparecido y se obtenía una calidad de audio final aceptable.

Materiales

Para el código de pruebas se creó un documento de Matlab en el que se implementaron conjuntamente los diferentes métodos para obtener la transformada de Hilbert, se realizaba la ecuación de anti-feeback y con la señal obtenida se calculaba el valor RMS comparándola con la señal anti-feedback ideal, obtenida mediante la función hilbert. Se realizaron diversas pruebas generando diferentes tipos de señales simples (senoidal, cuadrada, sawtooth), grabaciones de voz y música, las cuales se 38 procesaban utilizando distintos parámetros de cada algoritmo implementado. Los resultados se reunieron en 3 tablas, una por cada método implementado.

Tabla donde podemos ver la compensación de acuerdo a los valores básicos para evitar el margen de error y aplicar el filtro para evitar descompesaciones grandes y ver que frecuencia de manera LOG corta la frecuencia retroalimentada.

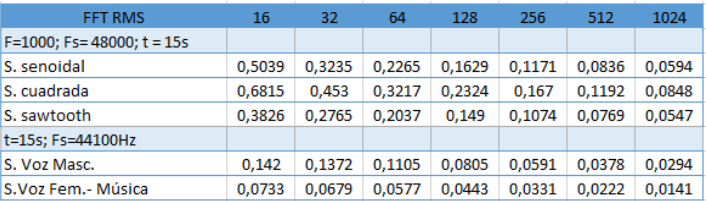
En filtros FIR

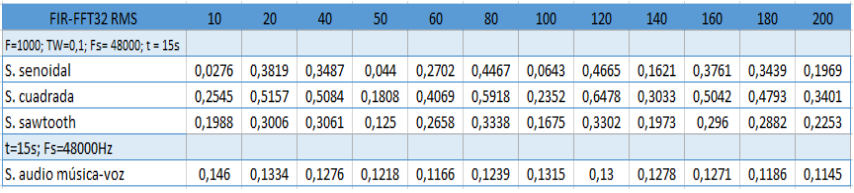


En ella se puede apreciar que con esta técnica se obtiene valores de error bastante pequeños en todos los tipos de señales utilizadas, que, como era de esperar, se van reduciendo conforme aumenta el número de coeficientes.

Con el uso de la técnica de FFT, también se pueden observar valores de error pequeños, aunque no tan buenos como en el caso anterior. Como era de esperar, conforme se incrementa el número de puntos de la FFT el valor de error se reduce. En este caso el resultado auditivo no fue tan bueno. Debido a los picos de distorsión que se comentan en el apartado de la FFT, a la hora de reproducir una señal de audio 39 se puede apreciar un chasquido que ensucia la señal y no la hace nada agradable para el oído.

Todo es con de acuerdo al tiempo.





Tras realizar las pruebas con la técnica del filtro FIR basado en FFT, que resolvió el problema de la técnica anterior, se obtuvieron unos valores de error muy dispares. Como se puede observar en la tabla 3, los valores están bastante alejados del cero, aun incrementando el número coeficientes (el número de puntos de FFT tiene un valor fijo de 32 debido a que el DSP Blackfin trabaja con un máximo de 32 muestras).



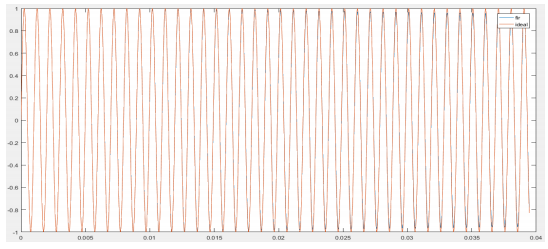
Memorias chicléales memorias fijas .

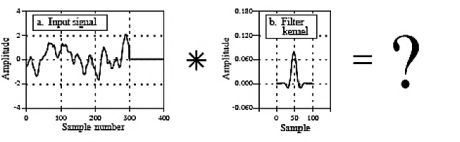
Variable con respecto el tiempo

Memorias del dsp

Por cada canal se ara el proceso de decisiones por lo cual ara un proceso con las variables respecto al tiempo y respeto ala frecuencuancia retro alimentada variables   
t= tiempo (ajustable   
fs= frecuencia = rango 20hz a 20khz   
f=frecuencia repetica con : respecto a t(tiempo)  
variables de salida   
fir = filtro a utilizar con respecto Fs  
gain= cuanto va restar con respecto al tiempo y la intensidad.

Grafica de desiciones por cada canal.





Aun así, la prueba auditiva fue satisfactoria comparada con los datos obtenidos. La señal reproducida mantenía una calidad bastante buena, que iba mejorando conforme aumentaba el número de coeficientes. Hay que decir que la componente tonal se distorsionaba, sobre todo con una señal de audio musical, y que la oscilación moduladora era apreciable, pero podía ser reducida con valores más bajos de fm. El chasquido producido en la técnica de FFT desaparecía, haciendo su escucha mucho más agradable.

El método que se implementará para realizar el algoritmo antifeedback en el DSP será mediante la técnica del filtro FIR, debido a que es la que menor valor de error y mayor calidad de audio proporciona, además de ser la más sencilla de implementar en el lenguaje de CrossCore y la que menos recursos de DSP consumirá.

### Conclusiones

El proyecto ha conseguido cumplir el objetivo de implementar un algoritmo para el control del feedback acústico en dos modelos de DSP distintos. A su vez, se ha conseguido estudiar a fondo las posibilidades que ofrece trabajar con CrossCore Embedded Studio y con el DSP como herramienta de procesamiento de audio. De esta forma se cumple el objetivo de punto de partida del proyecto. Respecto a los dos modelos de DSP se ha podido conocer las capacidades que ofrece cada uno. El modelo Blackfin, más limitado computacionalmente, pero perfecto para nuevos usuarios que quieran comenzar en la materia, además es adaptable para programar en coma flotante, aunque se consume más recursos. El modelo mucho más profesional y con unas características que lo hacen capaz de procesar una gran cantidad de datos, además, su formato nativo de programación es coma flotante, facilitando mucho su uso. A partir de los resultados obtenidos del algoritmo anti-feedback se puede concluir que, el método PFC FS es eficaz, robusto y de fácil implementación, con un buen rendimiento en señales simples de voz, pero aun así cuenta con limitaciones, como la 49 distorsión en la calidad de audio final, sobre todo en señales musicales, y en la cantidad de incremento de señal estable que puedes obtener sin llegar a percibir la moduladora. Mirando hacia futuras líneas de estudio, creo que sería posible mejorar este método, optimizándolo, y ejecutándolo conjuntamente con otra técnica que sea capaz de suplir sus carencias. Además, otro objetivo de interés sería conseguir implementar estos métodos de control del feedback acústico en sistemas multicanal, cosa que aumentaría su versatilidad. Finalmente, pienso que los métodos de control de feedback son una herramienta que puede ser de mucha utilidad en sistemas de refuerzo de sonido, pero aún queda mucho para que sea completamente rentable su uso, sean capaces de adaptarse a las diferentes situaciones de sonorización y producir mejores resultados que la labor que hace un técnico de sonido para eliminar ese efecto tan molesto llamado feedback acústico.

### Bibliografía

- Analog Devices, CrossCore® Embedded Studio 2.5.0 C/C++ Compiler and Library Manual for Blackfin Processors (Rev. 1.9), en

http://www.analog.com/media/en/dsp-documentation/softwaremanuals/cces-BlackfinCompiler-library-manual.pdf [Consulta: Junio,10,2017] -

Analog Devices, CrossCore® Embedded Studio 2.6.0 C/C++ Library Manual for SHARC Processors (Rev. 1.9), en http://www.analog.com/media/en/dsp-documentation/softwaremanuals/cces-SharcLibrary-manual.pdf [Consulta:Junio,11, 2017] -

Analog Devices, ADSP-BF706 en http://www.analog.com/en/products/processors-dsp/blackfin/adspbf706.html#product-overview –

Analog Devices, ADSP-21489 en http://www.analog.com/en/products/processors-dsp/sharc/adsp21489.html#product-overview - DUOANDIKOETXEA, J. (2000), Fourier Analysis, American Mathematical Society - MARPLE, S. L. "Computing the Discrete-

Time Analytic Signal via FFT." IEEE Transactions on Signal Processing. Vol. 47, 1999. - MathWorks, FFT-based FIR filtering using overlap-add method, en https://es.mathworks.com/help/signal/ref/fftfilt.html [Consulta: Marzo, 2017] - MathWorks, Discrete-time analytic signal using Hilbert transform, en https://es.mathworks.com/help/signal/ref/hilbert.html [Consulta: Marzo, 2017] - OPPENHEIM, A. V. Y SCHAFER, R.W. (2010). Discrete-Time Signal Processing, 3 rd Edition, Pearson Education 51 - SMITH, S. W. (1997). The Scientist and Engineer’s Guide to Digital Signal Processing, en http://www.dspguide.com/pdfbook.htm [Consulta: Junio, 2017] - SOGORB, T. C. (2016). Electrónica Aplicada al Audio. Módulo 1: Procesadores Digitales de Señal (Primera Parte). Escuela Politécnica Superior de Gandía. - VAN WATERSCHOOT, T. Y MOONEN, M. (2011). “Fifty years of acoustic feedback control: state of the art and future challenges,” Proc. IEEE, vol. 99, no. 2, pp. 288-327, Feb. 2011. - Wikipedia, Hilbert Transform, en https://en.wikipedia.org/wiki/Hilbert\_transform [Consulta: Agosto, 2017]