

Implementation of a Noise Reduction Algorithm in a Hearing Aid Device Based on a dsPIC

A. J. Uriz, *Student Member*, IEEE, P. D. Agüero, J. C. Moreira, J. C. Tulli and E. L. González

Abstract— Nowadays there are many people affected by hearing impairments. The cost of assistive listening devices leaves out of reach for many people a technological solution. In the search of an economical alternative, low-cost DSPs arises as a good choice. However, it could be difficult to decide which has the best cost - effectiveness trade off among all devices offered in the market. The main goal of this work, then, is to determine the memory requirements of three noise reduction methods. It is done in order to select the algorithm that uses fewer resources, which will be implemented on a hearing aid device based on a low-cost DSP. This assistive listening device through digital signal processing techniques can get the basic features of a high-end assistive listening device. Objective experiments are performed to analyze the performance of the implemented system..

Keywords— Hearing aid device, signal processing, SNR improvement, crosscorrelation DOA.

I. INTRODUCCIÓN

Las discapacidades auditivas son dolencias que afectan a un gran número de personas en nuestra Sociedad. En los últimos años se ha desarrollado una amplia variedad de dispositivos de asistencia para personas con deficiencias auditivas. Algunos de estos sistemas son desarrollados por Widex Inc.[1]. Esta compañía ofrece una amplia variedad de dispositivos de asistencia, desde audífonos analógicos hasta otros más modernos basados en procesadores digitales de señales (DSP). Estos últimos permiten obtener soluciones de mejor calidad que los analógicos. Debido a su capacidad de cómputo, los dispositivos digitales permiten llevar a cabo tareas como compresión espectral de voz, reducción de ruido y equalización dinámica, entre otras. La característica distintiva de este tipo de desarrollos es que permiten ofrecer una solución específica para la condición auditiva de cada usuario.

Una de las características más importantes de este tipo de sistemas es la relación señal a ruido (SNR) de la señal de salida. Este parámetro mide la relación entre las potencias de señal y ruido a la salida del sistema, y resulta fundamental maximizar dicho valor para el confort del usuario.

Hay muchos algoritmos que permiten mejorar la relación

señal a ruido en señales de audio. Uno de los más utilizados es el basado en la descomposición en valores singulares, en inglés *singular value decomposition* (SVD) [2], [3]. Esta técnica representa cada segmento de la señal a procesar como una matriz, a la cual se le calculan sus valores singulares. Entonces, se suponen los valores singulares de mayor valor asociados a componentes de señal, y los más pequeños al ruido. Luego, se reconstruye la señal utilizando una determinada cantidad de los aquellos valores singulares. En consecuencia, la señal reconstruida contiene menos ruido que la original. La principal desventaja de esta técnica es que para cada segmento de señal a procesar es necesario realizar una cantidad considerable de operaciones matriciales, lo que trae aparejado en primer lugar grandes requerimientos de memoria de datos, y en segundo un costo computacional que lo hace inapropiado para ser implementado en tiempo real en un DSP de bajo costo. Por otro lado, las técnicas de reducción de ruido basadas en la transformada wavelet discreta (TWD) [4]-[6], si bien son ampliamente utilizadas debido a su capacidad para trabajar con señales no estacionarias, requieren un gran número de operaciones, y además sus requerimientos de memoria de datos pueden ser excesivos para su implementación en un DSP de bajos recursos. En sistemas donde se dispone de más de un micrófono para la adquisición de sonidos, tal como en este caso, donde se existe un sistema de asistencia auditiva para cada oído, una opción razonable es aplicar el algoritmo de la correlación cruzada para mejorar la relación señal a ruido de la señal de salida del sistema [7]. Este algoritmo tiene la ventaja de requerir muy poca memoria de datos y además opera exclusivamente en el dominio temporal. Un algoritmo similar fue utilizado en [8] para determinar la dirección de arribo de una señal a un arreglo de micrófonos implementado utilizando microcontroladores de bajo costo dsPIC.

El objetivo de este trabajo es llevar a cabo la implementación de una técnica que permita mejorar la relación señal a ruido de salida de un sistema de asistencia auditiva basado en un procesador digital de señales de bajo costo. Los algoritmos son implementados en el DSP de Microchip dsPIC33FJ128GP802[9].

El trabajo se organiza de la siguiente manera: en la Sección II se estudian las características más relevantes de los tres algoritmos bajo estudio, haciendo énfasis en los requisitos para su implementación. La Sección III presenta el equipo desarrollado y describe la implementación del sistema de reducción de ruido. A lo largo de la Sección IV se presentan las mediciones realizadas. Por último, en la Sección V se presentan las conclusiones del trabajo y las tareas a desarrollar

A.J. Uriz, CONICET - Facultad de Ingeniería (UNMDP), Argentina, ajuriz@conicet.gov.ar

P.D. Agüero Facultad de Ingeniería (UNMDP), Argentina, pdaguero@fi.mdp.edu.ar

J. Castiñeira Moreira, CONICET - Facultad de Ingeniería (UNMDP), Argentina, casti@fi.mdp.edu.ar

J.C. Tulli, Facultad de Ingeniería (UNMDP), Argentina, jctulli@fi.mdp.edu.ar

E.L. González, Facultad de Ingeniería (UNMDP), Argentina, elgonzal@fi.mdp.edu.ar

en el futuro.

II. ALGORITMOS PARA REDUCCIÓN DE RUIDO

Los dispositivos digitales de asistencia [1] son la mejor opción para personas con deficiencias auditivas. Estos sistemas están basados en un procesador digital de señales, los cuales procesan los sonidos de forma tal que puedan ser percibidos por una persona con una dolencia en particular. Las funcionalidades más importantes de este tipo de sistemas son:

- **Extensor de audibilidad.** Es la funcionalidad [1] más importante de este tipo de sistemas. Es el encargado de procesar espectralmente la señal adquirida de forma tal que un usuario con una deficiencia auditiva específica pueda percibirla.

- **Amplificador de sonidos débiles.** Este subsistema [1] permite al usuario percibir tanto los sonidos débiles como los de amplitudes más elevadas. Esta funcionalidad permite percibir tanto conversaciones y sonidos lejanos como también otros emitidos cerca del usuario.

Aunque existen diversas funcionalidades que permiten mejorar la usabilidad [10] de un dispositivo de asistencia auditiva, las más importantes son las que permiten mejorar la relación señal a ruido del sistema, lo cual es fundamental para maximizar el confort del usuario. En las siguientes subsecciones se desarrollan tres algoritmos ampliamente usados para mejorar la SNR de una señal.

A. Algoritmo basado en la descomposición en valores singulares

Las técnicas de reducción de ruido basadas en la descomposición en valores singulares (SVD) [2],[3] están basadas en una representación matricial de la señal. Esta técnica toma cada segmento de la señal a procesar y obtiene A , una matriz cuadrada Toeplitz del mismo.

Luego, mediante la descomposición en valores singulares es posible definir (1), donde la matriz A es representada mediante el producto de tres matrices U , Σ y V .

$$A = U\Sigma V^T \quad (1)$$

Donde U y V son matrices unitarias y Σ es una matriz diagonal que contiene en su diagonal principal los valores singulares de A ordenados de forma decreciente. La definición (1) se corresponde al modelo de rango completo de la descomposición en valores singulares.

Luego, si A es una matriz de dimensiones $M \times N$, el rango de A se define como $\text{rango}(A) = R < M$, y (1) puede ser reescrita como:

$$A = \sum_{k=1}^r \sigma_k u_k v_k^T \quad (2)$$

Donde u_k es la columna k^{th} de la matriz U ; v_k , la columna k^{th} de la matriz V y σ_k los primeros r valores singulares ordenados en orden decreciente. Luego, $\text{rango}(A) = r$, y $\sigma_{r+1}; \sigma_{r+2}; \dots; \sigma_m$ son nulos. Si se consideran los últimos $r - p$ valores singulares muy pequeños, se puede truncar (2). Si se supone $k = p$, resulta (3), donde la matriz

\hat{A} de rango p , es denominada aproximación de bajo rango de A .

$$p < r: \quad \hat{A} = \sum_{k=1}^p \sigma_k u_k v_k^T \quad (3)$$

En el caso de reducción de ruido, los valores singulares más grandes están asociados a las componentes de señal, mientras que los autovalores más pequeños se corresponden a componentes de alta frecuencia, y por lo tanto se suponen asociados al ruido contenido en la señal. Entonces, el proceso de truncamiento de los valores singulares es similar a aplicar un filtrado pasabajos a la señal. Hay diversas formas para determinar el valor de p , los más utilizados definen dicho valor en función de la SNR de cada segmento de la señal a procesar. En este método, para $N = 256$ y $p = 50$, los requerimientos de memoria de datos son de alrededor de $58KB$. En estas condiciones puede obtenerse una mejora de alrededor de $10dB$ [3] en la SNR de la señal a procesar.

Este algoritmo tiene dos desventajas, la primera es que si se toma un valor de p muy bajo, se perderían componentes importantes de la señal de la voz. En segundo lugar, este algoritmo requiere una gran cantidad de operaciones matriciales para cada segmento de señal a procesar, por ello se requiere una gran cantidad de memoria de datos para implementarlo, y además su costo computacional lo hace inapropiado para una implementación en tiempo real en un DSP de bajo costo.

B. Algoritmo basado en la Transformada wavelet discreta

En los últimos años, los algoritmos para reducción de ruido basados en la Transformada Wavelet Discreta (TWD) [5],[6] han sido ampliamente utilizados para señales de voz. Esto se debe a su capacidad de trabajar con señales no estacionarias.

Este tipo de algoritmos se dividen en tres pasos: análisis, umbralamiento y síntesis. La primera etapa realiza un análisis multiresolución de la señal, descomponiendo la misma en componentes de señal pura y de detalle para cada nivel de análisis. Luego, el proceso de umbralamiento filtra la señal de acuerdo a los requerimientos del sistema. Por último, las componentes filtradas de cada nivel de análisis son utilizadas para sintetizar la versión filtrada de la señal.

De acuerdo a [6], para la implementación de este algoritmo con cuatro niveles de análisis, se requieren $N \times N + 4 \cdot N$. Por lo que si se toma $N = 256$, y datos de 16 bits, se requieren $137KB$ de memoria de datos. Esta cantidad de memoria de datos puede reducirse al estudiar la matriz de dimensiones $N \times N$ que se utiliza para el cómputo de la TWD, la cual debido a su naturaleza puede ser reducida a dos vectores fila de N elementos cada uno, los cuales deben ser desplazados para el cálculo de cada uno de los elementos. Al aplicar estas consideraciones, se requieren solo $8KB$ de memoria de datos para llevar a cabo la implementación del algoritmo.

La mejora obtenida por este algoritmo está en el orden de los $6dB$ [4],[6]. Aunque este método requiere menos recursos que el descripto basado en la SVD, debido a que el reductor de ruido es una funcionalidad adicional del sistema, se desea

utilizar un método que requiera aún menos memoria de datos.

C. Algoritmo basado en la correlación cruzada

El algoritmo basado en la correlación cruzada permite mejorar la relación señal a ruido del sistema, considerando los dispositivos de asistencia auditiva en cada oído del usuario como un arreglo de micrófonos. En este método, se toma la señal adquirida por cada elemento que compone el arreglo de micrófonos, y se la correlaciona con el fin de establecer el retardo existente entre la señal recibida por ambos dispositivos. Una vez determinado dicho retardo, se desplaza una de las señales con el fin de compensarlo, y luego se las promedia. Si se asume que ambas señales adquiridas están contaminadas por ruido blanco gaussiano no correlacionado, la señal resultante del promedio contiene menos ruido que la original. En la Fig. 1 se aprecia una representación de la topología presentada.

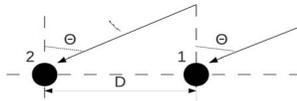


Figura 1. Esquema del arreglo de micrófonos utilizado.

Este método se basa en el hecho que la señal proveniente de la dirección θ_s arriba primero al micrófono 1, luego recorre una distancia ξ y, es recibida por el micrófono 2. Entonces, si: $\xi = D \sin \theta_s$, la dirección de arribo de la señal puede ser determinada usando (4).

$$\theta_s = \sin^{-1}\left(\frac{v \cdot \tau}{D}\right) \quad (4)$$

Donde v es la velocidad del sonido, y τ el tiempo que le lleva a la señal atravesar la distancia D . Entonces, una vez que se obtiene un valor de τ del rango $-T < \tau < T$, la dirección de arribo puede ser estimada. Pero, en este trabajo, τ es usado para determinar el desplazamiento temporal entre las señales que son adquiridas por cada micrófono. Entonces, con el fin de ajustar temporalmente ambas señales adquiridas, una de las mismas es desplazada $d = \tau / T_s$ muestras, siendo T_s el período de muestreo. En este trabajo, la distancia τ es obtenida calculando el valor máximo del resultado de la correlación cruzada $\Phi(\tau)$ (5) entre las señales adquiridas por cada micrófono $X_1(t)$ y $X_2(t)$.

$$\Phi(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} X_1(t) X_2(t + \tau) \quad (5)$$

Donde $\Phi(\tau)$ es un vector de $2N - 1$ elementos.

Entonces, si se supone que las señales que arriban a ambos micrófonos son iguales, si el elemento de valor máximo de $\Phi(\tau)$ está ubicado en el centro del mismo, ambas señales fueron recibidas en simultáneo por ambos micrófonos. En cambio, si el elemento de valor máximo del vector resultado está desplazado d elementos y el período de muestreo es T_s , se puede concluir que la señal ha arribado a ambos micrófonos con un retardo $\tau = d \cdot T_s$.

Una vez calculado el retardo entre ambas señales, el algoritmo propuesto desplaza una de las señales para compensar dicho retardo y promedia ambos vectores. Como consecuencia de esta operación, el ruido blanco contenido en la señal promediada se reduce $3dB$ respecto a cada una de las señales adquiridas.

Una ventaja de este algoritmo respecto a los descritos en las subsecciones previas es que su implementación solo requiere un vector adicional de $2N - 1$ elementos para almacenar el resultado de la correlación cruzada. De esta forma, si se toma $N = 256$, y datos de 16 bits, se requieren 2560 bytes de memoria de datos para su implementación. Se puede verificar que esta cantidad de memoria es inferior a la requerida por los otros métodos, y por lo tanto es apropiado para su implementación en el dispositivo propuesto.

III. IMPLEMENTACIÓN

Debido a los resultados surgidos del análisis llevado a cabo en la Sección II, se optó por implementar el algoritmo de reducción de ruido basado en la correlación cruzada. En las siguientes subsecciones se presenta el dispositivo desarrollado, y se describe en detalle la implementación del algoritmo.

A. Equipo desarrollado

Tal como se describió previamente, el objetivo de este dispositivo de asistencia auditiva es procesar los sonidos de forma tal que una persona con una dolencia en particular, pueda percibirlos de forma correcta. Las características más importantes del dispositivo desarrollado se describen a continuación:

- **Preamplificador de micrófono:** Amplifica la señal extraída del micrófono de forma tal que pueda ser procesada de una forma adecuada.

- **Filtro antisolapamiento:** Con el fin de limitar el ancho de banda de la señal que será adquirida por el conversor analógico a digital (ADC), es necesario implementar un filtro pasabajos. Los parámetros de diseño del filtro se calcularon en base a las especificaciones del DSP. En este caso, la frecuencia de muestreo es de 16KHz y el conversor analógico a digital tiene una resolución de 12 bits. Entonces, la frecuencia de rechazo máxima es de 8KHz, y la atenuación a partir de la misma debe ser de al menos 72dB. Estas condiciones fueron satisfechas al utilizar un filtro Sallen-Key elíptico de octavo orden. Además, con el fin de reducir las discrepancias entre el diseño y la implementación del filtro, se optó por utilizar un filtro integrado MAX7404 [11], el cual integra el filtro requerido y además tiene su frecuencia de corte programable a través de un oscilador interno.

- **Control automático de ganancia (AGC):** Una vez que las señales son filtradas, es necesario ajustar su nivel de forma tal que se optimice el uso de conversor analógico a digital. Además, debido a que la distancia entre el micrófono y las fuentes de sonido es variable, es necesario amplificar por distintos factores las señales con el fin de que el sistema pueda

procesarlas. La mejor forma de resolver este problema es a través de un control automático de ganancia, el cual amplifica el nivel de señal por un factor variable de forma tal de optimizar el uso del rango de entrada del ADC. En consecuencia, los sonidos débiles son amplificados por un factor mas elevado que otros sonidos de mayor amplitud. De esta forma, esta etapa del sistema facilita la percepción de los sonidos débiles.

- **Procesador digital de señales:** El procesador digital de señales adquiere la señal de salida del AGC. Luego, la procesa de acuerdo a los requerimientos del usuario y la resintetiza utilizando el conversor digital a analógico (DAC) integrado. Las características mas relevantes del DSP utilizado son descriptas mas adelante.

- **Etapa de salida:** Con el fin de minimizar los efectos de interferencias electromagnéticas, el DAC posee salida diferenciales. Por ello, es necesario obtener la diferencia de ambas señales y luego amplificar la señal resultante. Estas operaciones son llevadas a cabo utilizando dos amplificadores operacionales.

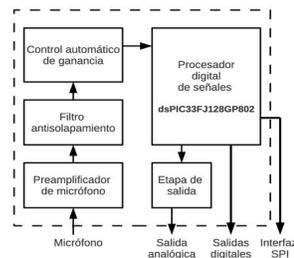


Figura 2. Sistema base implementado.

1) Procesador digital de señales

El procesador digital de señales utilizado en este proyecto es el dsPIC33FJ128GP802 [9] de Microchip. Las principales características de este DSP de bajo costo son:

- **40KB de memoria de programa.** Permite utilizar compiladores cruzados para su programación.
- **16KB de memoria de datos.** De los cuales 2KB son utilizados para acceso directo a memoria (DMA).
- **Bajo costo.** US\$ 4. Inferior a otros DSP comerciales.
- **Bus de datos de 16 bits.**
- **ADC integrado de 12 bits@500ksps.**
- **DAC integrado de 16 bits@100ksps.**
- **Módulo de comunicación SPI integrado.** Permite establecer un enlace entre el dsPIC y diversos dispositivos.

Además, debe tenerse en cuenta que las librerías [13] están disponibles de manera gratuita. La mayor ventaja del dsPIC33FJ128GP802 es que permite realizar en **simultáneo** dos tareas, que en este caso resultan ser la de adquisición de datos de un segmento y el procesamiento en paralelo de los datos del segmento anterior. Esto se realiza utilizando el módulo de DMA del dispositivo [9], el cual trabaja de forma independiente al procesador principal. Por este motivo se reduce casi a la mitad el tiempo de procesamiento de cada uno de los segmentos, lo cual es un factor crítico a la hora de

obtener un dispositivo que funcione en tiempo real. Otro aspecto a tener en cuenta, es que la utilización de técnicas de DMA aumenta el rendimiento del sistema, ya que reduce al mínimo las fuentes de interrupción del programa principal. Es decir, los dispositivos realizan la transferencia de datos utilizando el módulo DMA, y por ello no agrega retardos de ejecución al programa principal. En particular los tiempos que se reducen son:

- Tiempo de procesamiento de la rutina de interrupción.
- Tiempo de acceso, almacenamiento y lectura de la pila ó stack del sistema.
- Tiempo de acceso a los periféricos.

El extensor de audibilidad fue implementado en este caso utilizando el algoritmo de compresión espectral SPINC [12]. La principal ventaja de esta técnica es que utiliza un factor de compresión unitario para las frecuencias por debajo de los 1000Hz, y un factor elevado para el rango de frecuencias por encima de los 1000Hz. De esta forma, los formantes mas importantes de la voz no son afectados, mientras que para los formantes ubicados en frecuencias mas altas se obtiene una compresión espectral elevada.

A la hora de implementar el algoritmo FFT en el dsPIC se debió realizar una variación al algoritmo ya que el mismo opera en aritmética de punto flotante, lo cual no es soportado por la arquitectura del dsPIC utilizado. Las operaciones de punto flotante debían ser emuladas en software, aumentando considerablemente los tiempos de procesamiento. Por dicho motivo se implementó el algoritmo de FFT en aritmética de punto fijo de 16 bits, trabajando en base a escalamientos de magnitud.

El tiempo de procesamiento del algoritmo de compresión espectral es $T_{SPINC} = 0,29ms$, el cual es muy inferior al tiempo de cálculo de las rutinas implementadas para el cálculo de la FFT y la IFFT (1,85ms y 1,83ms respectivamente).

Se midieron los tiempos de procesamiento utilizando una frecuencia de muestreo $f_{sampling} = 16,288KHz$, y en dichas condiciones se registró un tiempo total de procesamiento para cada segmento de 256 puntos de 15,72ms y un tiempo de adquisición de 3,97ms. Ambos tiempos fueron medidos por separado debido a que ambas tareas se ejecutan en paralelo. Entonces, se concluye que debido a que el tiempo total de procesamiento es menor al de adquisición, el sistema trabaja en tiempo real. Otro aspecto que debe considerarse es que en esta configuración, los requerimientos de memoria de datos y de programa son bajos (34% y 5% respectivamente), mientras que la memoria DMA disponible fue utilizada en su totalidad. Esto impone una limitación en la capacidad del sistema.

B. Descripción de la implementación del algoritmo

Tal como se describió previamente, el algoritmo basado en la correlación cruzada utiliza la señal proveniente de un arreglo de micrófonos para mejorar la SNR de salida del sistema. En este caso, el arreglo de micrófonos está compuesto por dos sistemas como el mostrado en la Fig. 2. Entonces, el

sistema base cuyo micrófono está en el oído derecho del usuario, adquiere la señal y la transmite al sistema base del oído izquierdo utilizando un enlace SPI [14]. De esta forma, el dispositivo ubicado en el oído izquierdo del usuario calcula la correlación cruzada entre los datos adquiridos por el mismo y los recibidos del otro sistema. Para establecer el enlace, se utilizó el módulo SPI que posee el dsPIC33FJ128GP802 [9], [14]. Una de las principales ventajas de esta interfaz serie sincrónica es que permite transferir datos hasta a 10Mbps, lo cual es un factor fundamental para obtener un sistema que opere en tiempo real. Para llevar a cabo esta implementación, es necesario configurar al sistema del oído derecho en modo SPI maestro y al del oído izquierdo en modo SPI esclavo. Además, con el fin de reducir los efectos de las interferencias entre las líneas de comunicación y de datos, el enlace fue establecido utilizando el modo SPI Framed [9],[14].

Para la implementación, la frecuencia de reloj de sistema de ambos dsPIC debe ser exactamente la misma. Por lo tanto, se utilizó un cristal externo para generar la señal de reloj del sistema maestro. La misma señal fue utilizada como fuente de reloj del sistema esclavo. Una consideración especial debe hacerse para conectar el oscilador del sistema maestro al esclavo: el efecto de la capacidad de la compuerta de entrada del sistema esclavo, la cual es del orden de los 60pF. Además, si se tiene en cuenta la capacidad parásita del cable utilizado para comunicar la señal de reloj, se obtiene una capacidad equivalente de 70pF, que afecta la frecuencia de oscilación del sistema maestro. Este problema fue resuelto conectando un condensador de 8,2pF en serie con el cable. Entonces, la capacidad resultante se reduce notablemente y el reloj del sistema opera normalmente. A la hora de realizar esta mejora debe tenerse especial precaución en que la amplitud de la señal de reloj que ingresa al sistema esclavo sea la suficiente para excitarlo. En la Fig. 3 se presenta un esquema del sistema desarrollado.

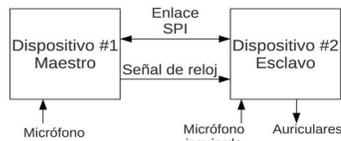


Figura 3. Sistema completo implementado.

Una vez que ambos sistemas son sincronizados y la comunicación es establecida, se aplica el método de reducción de ruido. El algoritmo desarrollado calcula la correlación cruzada entre el vector recibido por SPI y el vector adquirido por el convertor AD del sistema esclavo.

Tal como se comentó previamente, el maestro transmite un vector de $2.N$ muestras por cada segmento a procesar, siendo $N=256$. El receptor utiliza los datos recibidos y las últimas $3.N$ muestras adquiridas por su convertor AD para aplicar la mejora propuesta.

En primer lugar, se definen dos punteros. El primero, llamado p_0 , apunta al elemento central del vector con los datos adquiridos, mientras que el segundo puntero ($p_{reception}$)

apunta al elemento en la posición $N/4$ del vector recibido a través del sistema maestro. La estructura de dichos vectores es presentada en la Fig. 4.

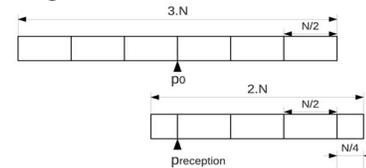


Figura 4. Estructura del vector adquirido (superior) y el recibido por SPI (inferior), junto con los respectivos punteros.

En segundo lugar, los primeros N elementos a partir de cada puntero son correlacionados utilizando la instrucción *VectorCorrelate()* de la librería de lenguaje C30 de Microchip[13]. El resultado de esta operación es almacenado en un vector de $2.N-1$ elementos. Si el elemento de valor máximo de dicho vector está ubicado en el centro del mismo, no existe retardo entre las señales. En cambio, si existe una diferencia de d muestras entre el centro del vector resultante y la posición del máximo elemento, ambas señales tienen un retardo $\tau = d.T_s$, donde T_s es el período de muestreo. Una representación de las operaciones realizadas se presenta en la Fig. 5.

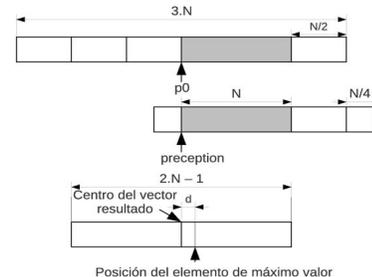


Figura 5. Esquema de los vectores utilizados para el cálculo de la correlación cruzada.

Luego, una vez definido el retardo d , el puntero $p_{reception}$ es desplazado con el fin de compensar dicho retardo. Este retardo puede ser positivo o negativo, dependiendo del resultado de la correlación. Entonces, los primeros $3.N/2$ a partir de los punteros p_0 y $p_{reception} + d$ son promediados y el resultado se almacena en un vector de $3.N/2$ elementos que se utiliza para sintetizar la señal de salida utilizando el método de la suma solapada en el dominio del tiempo (TD-OLA). Para realizar esta operación, se definen dos punteros p_{EVEN} y p_{ODD} para definir los segmentos de síntesis pares e impares respectivamente.

Finalmente, los $2.N$ elementos a partir del puntero p_0 son desplazados N posiciones a la izquierda y el sistema queda listo para procesar un nuevo segmento.

IV. EXPERIMENTOS

Un conjunto de mediciones fueron realizadas para analizar el funcionamiento del sistema. En primer lugar se midieron los

tiempos de ejecución de cada una de las tareas.

La Tabla I muestra que para cada segmento de $N=256$ a procesar la suma de los tiempos de ejecución de la rutina de transmisión de datos por SPI, la ejecución del algoritmo de reducción de ruido por correlación y los restantes tiempos de procesamiento involucrados, es inferior al tiempo de adquisición. Por lo tanto, el sistema funciona en tiempo real.

Luego, con el fin de verificar la correcta implementación del algoritmo, se midió la SNR de salida del sistema para una señal sinusoidal de 1KHz y para vocales sostenidas. La mejora obtenida fue del orden de los 3dB , lo cual se ajusta a lo planteado en la teoría. La SNR del sistema implementado fue es 49dB .

El uso de recursos del sistema implementado, se presenta en la Tabla I.

TABLA I
TIEMPO TOTAL DE PROCESAMIENTO, TIEMPO DE ADQUISICIÓN, Y USO DE MEMORIA DE DATOS, DMA Y MEMORIA DE PROGRAMA

T. Procesamiento	T. Adquisición	RAM %	DMA %	ROM %
6,09ms	15,72ms	96	100	8

En primer lugar, la Tabla I muestra que la memoria de datos se utiliza casi en su totalidad y la memoria disponible para acceso directo a memoria es totalmente utilizada. Mientras que, en segundo lugar, se utiliza un porcentaje bajo de memoria de programa. Por lo tanto, se podría optimizar el código de forma tal de hacer un mejor uso de la memoria de programa.

V. CONCLUSIONES

En este trabajo se presentó un dispositivo de asistencia auditiva de bajo costos capaz de implementar las funcionalidades mas importantes de un dispositivo comercial de alta gama.

Además, en este trabajo se llevó a cabo un estudio del uso de recursos de tres algoritmos para reducción de ruido en señales de voz con el fin de implementarlos en el dispositivo. Se demostró que es posible implementar el algoritmo basado en la correlación cruzada en el DSP utilizado y que el mismo opera en tiempo real. La mejora de la SNR del sistema está de acuerdo con la estimada de forma teórica.

En el futuro, por un lado se pretende optimizar otros algoritmos de reducción de ruido de forma tal que puedan ser implementados en el dispositivo. Por otro lado, se desea estudiar tecnologías que permitan reducir el tamaño del desarrollo de forma tal que mejore su usabilidad por parte del usuario.

REFERENCIAS

- [1] Widex Inc. <http://www.widex.com/>, Lyngø, Denmark.
- [2] H. G. Gauch, Jr., Noise Reduction By Eigenvector Ordinations, Ecological Society of America, Vol. 63, No. 6, pp. 1643-1649, 1982.
- [3] E. V. de Payer, Preprocesado de la señal de voz: el método de la descomposición de subespacios, Revista Argentina de Bioingeniería, Vol.16, No.1. June 2010.
- [4] V. Balakrishnan, N. Borgesa and L. Parchment, Wavelet Denoising and Speech Enhancement, 2006.
- [5] T. Young and W. Qiang, The realization of Wavelet Threshold noise filtering Algorithm, Proceedings of 2010 Conference on Measuring

Technology and Mechatronics Automation. pp 953-956. 2010.

- [6] Ch. Dolabdjian, J. Fadili and E. Huertas Leyva, Classical low-pass filter and real-time wavelet-based denoising technique implemented on a DSP: a comparison study, The European Journal Applied Physics. Vol.20, pp 135-140. 2002.
- [7] A. K. Tellakula, Acoustic Source Localization Using Time Delay Estimation, Degree Thesis. Bangalore, India: Supercomputer Education and Research Centre Indian Institute of Science, 2007.
- [8] S. Takahashi, T. Morimoto, Development of Small-size and Low-priced Speaker Detection Device Using Micro-controller with DSP functions. Proceedings of the IMECS 2011, Vol.1, 2011.
- [9] Microchip Inc., dsPIC33FJ128GPX02/X04 Data Sheet, High Performance 16-bit Digital Signal Controllers. <http://www.microchip.com/>, 2009.
- [10] International Organization for Standardization (ISO), ISO/IEC 9241. <http://www.iso.org/>, 1999.
- [11] Maxim Inc., MAX7400/MAX7403/MAX7404/MAX7407 8th-Order, Lowpass, Elliptic, Switched-Capacitor Filters, 1999.
- [12] E. Terhardt, The SPINC Function for scaling of frequency in Auditory Models. Journal of Acoustic. Vol.77 pp.40-42, 1992.
- [13] Microchip Inc. 16-Bit Language Tools Libraries. <http://www.microchip.com/>, 2005.
- [14] Microchip Inc., Serial Peripheral Interface (SPI) - dsPIC33F/PIC24H FRM. <http://www.microchip.com/>, 2011.



Alejandro José Uriz was born in Mar del Plata, Argentina in 1984. He Obtained the Degree of Electronic Engineer in Universidad Nacional de Mar del Plata (UNMDP) in 2010. Currently he is studying towards his PhD degree. Nowadays, he is working in the Communications Lab - UNMDP. His current research interests include rehabilitation engineering and speech processing.



Pablo Daniel Agüero was born in Mar del Plata, Argentina in 1977. He received his degree of Electronic Engineer at this university in 2002. He moved to University Politècnica de Catalunya to make its PhD. He worked in the speech group of that university in the area of speech synthesis in several projects and networks. Currently he is finishing his PhD on "Speech synthesis applied to Speech-to-speech Translation".



Jorge Castiñeira Moreira was born in Mar del Plata, Argentina in 1962. received his degree of Electronics Engineer at University of Mar del Plata, Argentina, on March 1990. He continued his studies to receive his MSc in Digital Signal Processing Applications in Communications Systems in 1996 and his PhD in Communication Systems, at Lancaster University, Lancaster UK, on 2000. Currently he is an Associate Professor in UNMDP and responsible for the teaching area Communications. He is also director of the research project "Communication and Information Theory applied to Data Networks" in the Mar del Plata University.



Juan Carlos Tulli was born in Mar del Plata, Argentina in 1947. He received the Eng. degree in electric engineering from UNMDP in 1976. Nowadays, he is Full Professor and Director of the Communications Lab - UNMDP. His current research interest is rehabilitation engineering.



Esteban Lucio González was born in Mar del Plata, Argentina in 1959. He received his degree of Electronic Engineer at UNMDP in 1989. In 1991 he moved to Antarctica as a Scientific leader, responsible for San Martín Antarctic Base Laboratory. In 1998 he got his Master of Science in Digital Signal Processing Applications in Communications Systems at Lancaster University, England.