Implementation of a Speech Recognition System in a DSC

A. G. Alvarez, D. A. Evin and S. Verrastro

***Abstract*— This paper describes the development and validation of an Embedded IsolatedWord Recognition System (IWR) for the Argentinian Spanish language, implemented on the STM32F4-Discovery platform. Its front-end extracts Mel Frequency Cepstral Coefficients (MFCC), while its classification step is based on the Dynamic Time Warping (DTW) algorithm. Since the system was conceived as a base platform for the research and development of speechbased command and control applications, it was designed to be modular and to meet real-time performance. The system includes a Real Time Operating System (RTOS) to manage various processing and control tasks, which can be easily reconfigured with different acquisition, processing and recognition parameters using a single file. The validation was done using a scenario of robotic control, achieving performance rates which demonstrates the practical usefulness of the system.**

***Keywords*— Embedded ASR, DSC, IWR, DTW, MFCC.**

I. INTRODUCCIÓN

El habla es considerada como una de las formas más naturales para transmitir pensamientos, ideas y emociones. Permite que las personas se comuniquen entre sí, no solo de forma instantánea, sino también simultánea. Por este motivo, no es sorprendente encontrar varios intentos, a lo largo de más de sesenta años, para producir dispositivos y sistemas que posean este mecanismo de interacción.

El estudio y la investigación de la tecnología del habla han logrado avances sustanciales en las últimas décadas. Sin embargo, dadas las dificultades de este problema, las soluciones propuestas son siempre limitadas a situaciones específicas, conocidas y en escenarios controlados. Por otro lado, los costos computacionales asociados con los reconocedores de habla, tales como procesamiento, velocidad, y memoria son relativamente altos.

Los avances en *sistemas embebidos* han tenido éxito no sólo en la reducción de costos y del consumo de energía, sino también en el aumento de su portabilidad, velocidad y simplicidad. Esos avances los condujeron a una notable proliferación en el mercado. De allí que existe hoy en día un gran interés en tener múltiples dispositivos con capacidades de interacción a nivel humano, siendo el habla uno de los principales [1].

II. TRABAJOS RELACIONADOS

La mayoría de las soluciones propuestas en la literatura que abordan el problema de *Reconocimiento Automático del Habla* (RAH) en Sistemas Embebidos, se basan en *Procesadores de Señales Digitales* (DSP) [2-6] o en Field Programmable Gate Arrays (FPGA) [7-9]. También pueden encontrarse algunas implementaciones basadas en Microcontroladores. La mayoría de ellas requieren algún canal de comunicación con un servidor remoto que procesa los datos recogidos y realiza el reconocimiento real [10-12]. Estos métodos, aunque poderosos, presentan latencias y consumos más elevados. A su vez, existen algunas aproximaciones de Sistemas RAH totalmente basados en microcontroladores. Por lo general, estas aplicaciones, utilizan características acústicas más sencillas con el fin de reducir el costo computacional. Van desde vectores de características muy simples [13-15], hasta algunos más complejos y robustos [16, 17]. Por último, otras implementaciones menos populares se basan en chips dedicados para llevar a cabo el proceso de reconocimiento [18-20].

Los avances en los Sistemas Embebidos mencionados anteriormente han dado lugar a un diseño híbrido basado en microcontroladores y en DSPs: *Controladores de Señales Digitales* (DSC) [1]. Al igual que los microcontroladores, los DSCs poseen respuestas rápidas a interrumpciones y ofrecen una gran cantidad de periféricos orientados al control. Pero también incorporan unidades multiplicar-acumular (MAC) de ciclo-único encontradas en los DSPs, registros de desplazamiento, y grandes acumuladores.

El objetivo de este trabajo fue diseñar, desarrollar y validar un sistema de Reconocimiento de Palabras Aisladas (RPA) en un Sistema Embebido para el lenguaje Español hablado en la Argentina, que pueda ser utilizado como una plataforma de base para la investigación y el desarrollo de aplicaciones de comando y control basadas en la voz. Uno de los requisitos para este proyecto fue desarrollar el sistema en un DSC genérico, adaptando tanto al procesamiento de audio como al algoritmo de reconocimiento de voz a los recursos computacionales y de memoria disponibles en dicho dispositivo.

El trabajo está organizado de la siguiente manera: la siguiente sección describe las arquitecturas de sistemas ASR típicas basadas en el enfoque de Comparación de Plantillas. A continuación, se describen las características del DSC seleccionado. Después de eso, se da un detallad el desarrollo del sistema, junto con su estructura lógica y de control. La siguiente sección muestra los resultados de la validación, y finalmente, el documento concluye con un breve resumen y algunas sugerencias para trabajos futuros.

A. G. Alvarez, Estudiante de la Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Argentina, alegasalv@gmail.com

D. A. Evin, Investigador del Laboratorio de Investigaciones Sensoriales, INIGEM, CONIET-UBA, Buenos Aires, Argentina, diegoevin@gmail.com

S. Verrastro, Profesor de la Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, Argentina, sebastian_verrastro@yahoo.com.ar

III. ÁREA DE ESTUDIO

Los sistemas RAH se pueden clasificar de acuerdo con el grado de restricciones impuestas en la señal acústica: *Habla Aislada*, *Conectada* o *Continua* [21, 22]. Se refieren al reconocimiento de palabras individuales emitidas aisladamente unas de otras, palabras pronunciadas secuencialmente con una pausa entre ellas, y secuencias de palabras producidas naturalmente y sin restricciones en sus límites, respectivamente. En consecuencia, el nivel de dificultad de la tarea aumenta a medida que la restricción del habla se vuelve menos estricta.

Debido a los bajos recursos disponibles en el dispositivo y a los objetivos de este trabajo, el resto del documento se centra en el sistema RPA. Como se muestra en la Fig. 1, un sistema RPA típico comprende 3 etapas: *Front-End*, *Detección de Punto Final* y *Reconocimiento de Patrones Acústicos*.

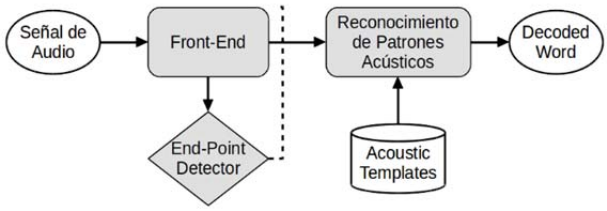

Figura 1. Sistema de Reconocimiento de Palabras Aisladas.

III.1. FRONT-END

El *Front-End*, también conocido como *Análisis Acústico* o *Extractor de Características*, recibe la señal de habla pura y extrae una secuencia de vectores de características acústicas. Esta etapa es crítica para reconocedores de alta calidad, porque esas características o descriptores son la base fundamental sobre la cual operará el reconocedor. Deben contener información útil para caracterizar y discriminar los sonidos que componen la palabra recitada, y al mismo tiempo deben eliminar redundancias y ser robustos frente al ruido. Entre las características acústicas más comunes, podemos mencionar: la Codificación Predictiva Lineal (LPC), la Predicción Lineal Perceptual (PLP), y los MFCC [21, 22]. Siendo estos últimos las características más populares en la literatura y también la representación seleccionada para este trabajo.

III.2. RECONOCIMIENTO DE PATRÓNES ACÚSTICOS

El bloque del *Reconocimiento de Patrones Acústicos* infiere la palabra más probable, en base a las características del habla observadas y a los modelos de reconocimiento. Según su estructura de modelado, existen varias estrategias para llevar a cabo tal inferencia, entre ellas encontramos: las aproximaciones basadas en la *Comparación de Plantillas*, la *Estadística*, y las *Redes Neuronales Artificiales* (ANN) [21-23]. En el enfoque de Comparación de Plantillas después de establecer un conjunto de prototipos ejemplares para cada palabra en el léxico, la etapa de reconocimiento aplica alguna medida de distancia para encontrar la plantilla más similar a la palabra recitada. El enfoque estadístico calcula un modelo acústico para cada palabra o sub-palabra de tal manera que representa la distribución de probabilidad de sus *vectores acústicos*, luego estos *modelos acústicos* son utilizados durante la fase de decodificación para buscar el modelo que mejor explica la secuencia de vectores de características observada. Finalmente, el Reconocimiento de Patrones basado en ANN entrena las mismas para clasificar directamente la secuencia observada de características acústicas, o para ser utilizadas en una combinación híbrida con el marco estadístico.

El estado del arte para los ASR está dado por el enfoque estadístico, en particular, utilizando los *Modelos Ocultos de Markov* (HMM). La mayoría de ellos sin embargo, también aplican algún clasificador discriminativo como ANN o *Suported Vector Machines* (SVM) en un marco híbrido.

Aunque el enfoque basado en Comparación de Plantillas ha sido superado, sigue siendo interesante para aplicaciones como las reportadas en este trabajo; donde el vocabulario a reconocer es pequeño, la segmentación de palabras se resuelve fácilmente, y cuando los recursos disponibles para el reconocimiento son muy limitados. También tiene la ventaja de tener un procedimiento de entrenamiento más sencillo y menos exhaustivo. En particular, en este trabajo, el algoritmo DTW fue utilizado para comparar las secuencias de vectores de características acústicas.

III.3. DETECCIÓN DE PUNTO FINAL

Como su nombre indica, la *Detección de Punto Final* o *Detección de Actividad de Vocal* (VAD), analiza cada segmento de voz, separando a los que contienen información acústica relacionada con el habla de aquellos correspondientes al ruido de fondo [24]. En el enfoque basado en la comparación de plantillas, el rendimiento de esta etapa es crítica [23], ya que si un segmento de ruido es considerado como parte de la expresión de entrada, éste puede inducir una selección incorrecta. Antes de detallar la ejecución de los bloques antes mencionados, se describen las características de la plataforma del DSC seleccionada; ya que sus recursos, especialmente su capacidad de memoria y su potencia de procesamiento, limitaron y encaminaron las decisiones de implementación.

IV. CARACTERÍSTICAS DE LA PLATAFORMA

El sistema fue desarrollado en la placa *STM32F4-Discovery*. La CPU de esta placa es un microcontrolador STM32F407VGT6, que ofrece un *ARM Cortex-M4F* de 32 bits con *1 MB* de memoria flash y *192 KB de RAM*. Su velocidad de reloj va hasta los *168 MHz*, y tiene una MAC de un ciclo-único y una *Unidad de Punto Flotante* (FPU), lo que la hace apropiada para el estudio y desarrollo de aplicaciones de procesamiento de sonido y video.

Esta plataforma contiene también un micrófono digital omnidireccional incorporado (MP45DT02) para la adquisición de audio. Este es un *Sistema Microelectromecánico* (MEMS), que entrega una señal de *Modulación por Densidad de Pulsos* (PDM), que utiliza técnicas de modelado de ruido y de sobre-muestreo, aumentando la *Relación Señal a Ruido* (SNR) en la banda de audio [25]. Esta señal se lee a través del bus Inter-

Integrated Circuit Sound (I2S), que a su vez está interconectado con el periférico de *Acceso Directo a Memoria* (DMA). El DMA adquiere paquetes de datos desde el periférico y los almacena en la memoria interna sin ninguna intervención de la CPU, reduciendo la carga computacional y aumentando su rendimiento [26].

Para fines de depuración y de configuración, la aplicación utiliza una unidad flash, conectada al *Bus Serie On-The-Go Full USB*.

V. DESCRIPCIÓN GENERAL DEL SISTEMA

El sistema de reconocimiento fue implementado utilizando el *Cortex Microcontroller Software Interface Standar* (CMSIS). Esta, posee una *Capa de Abstracción de Hardware* (HAL) que define una interfaz con herramientas genéricas, facilitando así la portabilidad hacia otros procesadores Cortex-M y a chips de otros fabricantes. También se basa en el Sistema operativo *FreeRTOS* para la gestión de hilos de procesamiento y en el HAL de ST-Microelectronics para el control de periféricos.

El sistema de RPA comienza con la ejecución de una máquina de estados capaz de seleccionar entre tres tareas principales: *calibración*, *entrenamiento* y *reconocimiento*. Todas ellas interactúan con las tareas de captura y de procesamiento de audio y otras funciones del sistema de menor importancia. Como se ha mencionado anteriormente, hay un archivo de configuración global que el usuario puede modificar para ajustar las diferentes variables del proceso. Esto le confiere flexibilidad al sistema, especialmente durante el arranque y para la evaluación comparativa de rendimiento. El esquema mostrado en la Fig. 2 representa la interacción de los sistemas.

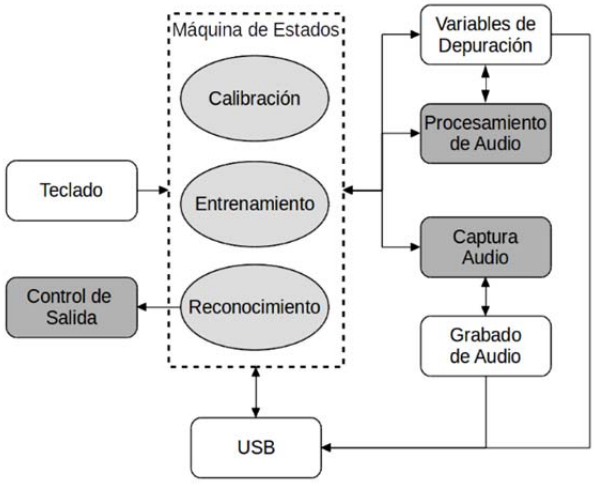

Figura 2. Diagrama en bloques del Sistema.

Durante la fase de entrenamiento, el usuario registra las plantillas para las palabras del vocabulario. El procedimiento de calibración estima la relación señal a ruido y los niveles de ruidos de fondo, que se utilizan para configurar los umbrales del detector de punto final. Finalmente la fase de reconocimiento captura y calcula los segmentos de las características acústicas de lo enunciado, y realiza el comparación de los patrones acústicos, reconociendo a la palabra correspondiente.

V.1. FRONT-END

En este bloque funcional el audio es capturado primero a través del micrófono incorporado. A continuación, a través del DMA, la señal PDM se mueve a un buffer cíclico que emplea una estrategia de doble buffering [26]. Esta permite procesar la mitad del buffer mientras que la otra mitad está siendo llenada. Después de esto, el formato de audio se transforma de PDM a *Modulación por Impulsos Codificados* (PCM) utilizando la funcionalidad proporcionada por ST-Microelectronics. Este conversor emplea un decimador, un filtro pasa bajos y uno pasa alto. Estos dos últimos componentes se implementan mediante filtros de *Respuesta Infinita al Impulso* (IIR), con frecuencias de corte establecidas a 10 Hz y 8 kHz, respectivamente.

V.1.1. Acondicionamiento de Señal

Una vez adquirida la señal acústica, primero es normalizada dentro del rango [-1,1] y luego se pasa a través de un *Filtro de Pre-Énfasis*. Este filtro se utiliza para compensar la atenuación del espectro de habla a altas frecuencias, y se aproxima a la inversa de la respuesta en frecuencia de la transmisión de la boca [23, 27]. La implementación de esta compensación, se da por un filtro de *Respuesta Finita al Impulso* (FIR) de primer orden que se muestra en la Ec.(1).

$$H_{[z]} = 1 - a \cdot z^{-1} \qquad 0{,}9 \leq a \leq 1{,}0 \tag{1}$$

V.1.2. Ventaneo

La señal de habla se puede considerar como un proceso cuasi-estacionario porque tiene un comportamiento estacionario considerando cortos períodos de tiempo (de 5 a 100 milisegundos). Por lo tanto, el análisis del espectro en tiempo corto de la señal requiere su segmentación en tramas de entre 20 y 30 milisegundos. Para este propósito se prefieren ventanas de borde suave (Hamming, Hanning), en vez de ventanas cuadradas, ya que las primeras introducen menos distorsión en la señal transformada. Para suavizar la variación espectral sobre los bordes de las ventanas consecutivas, se hace necesario el uso de una superposición entre 10 y 15 milisegundos [6, 23, 28].

V.1.3. MFCC

Como se ha descrito anteriormente, los MFCC son uno de los descriptores de habla más populares en este campo. Estas características se obtienen mediante la combinación de *Bancos de Filtros de Mel* y del *Cepstrum* de la señal [6, 27]. El primero está dado por un conjunto de bancos de filtros triangulares que se aproxima a la respuesta en frecuencia del oído humano. Mientras que el segundo es una transformación bien estudiada definida como la *Transformada Inversa de Fourier* (IFT) del logaritmo del cuadrado del módulo del espectro (Ec. 2).

$$\mathcal{F}^{-1}\{\log\|\mathcal{F}(f_{(t)}{}^2)\|\} \tag{2}$$

El modelo de fuente-filtro para la producción del habla implica la convolución de una fuente de excitación con la

respuesta al impulso del tracto vocal [23]. El cepstrum da la posibilidad de separar la señal de excitación del tracto vocal, aplicando un filtro simple que se muestra en la Ec.(3) [29]. Este filtro es comúnmente llamado *lifter* porque se aplica en el dominio cepstral.

$$H_{(k)} = 1 + \frac{K}{2}\sin\left(\frac{\pi k}{K}\right) \qquad 0 \le k < K \tag{3}$$

Además, en lugar de aplicar la IFT convencional, se aplica una *Transformada de Coseno Inverso tipo II* con el fin de optimizar la compresión [30].

$$C_{[k]} = \sum_{n=0}^{N-1} x_{[n]} \cos\left(\frac{\pi k}{N} \cdot \left(n + \frac{1}{2}\right)\right) \quad 0 \le k < N-1 \tag{4}$$

V.1.4. Deltas y Energía

La eficiencia del MFCC puede ser altamente mejorada mediante la adición de la energía de cada segmento de voz y el comportamiento dinámico a través de tramas consecutivas [28]. La primera de ellas se calcula rápidamente a través de la Ec.(5) después del proceso de ventaneo. La última, está dada por los coeficientes delta y delta-delta, que también son denominados coeficientes de velocidad y aceleración. Estos coeficientes se pueden obtener tomando las derivadas instantáneas, pero se convertirían en estimaciones ruidosas. Por lo tanto, se aplica a menudo una *regresión lineal* dada por la Ec.(6) [29, 31]. Esta etapa final, se lleva a cabo utilizando una línea de retardo por derivada.

$$E_{(m)} = \frac{1}{K}\sum_{k=0}^{K-1} {x_{(k,m)}}^2 \tag{5}$$

$$d_t = \frac{\sum_{\theta=0}^{\Theta} \theta(c_{(t+\theta)} - c_{(t-\theta)})}{2\sum_{\theta=0}^{\Theta} \theta^2} \tag{6}$$

El procesamiento completo se puede observar en la Fig. 3.

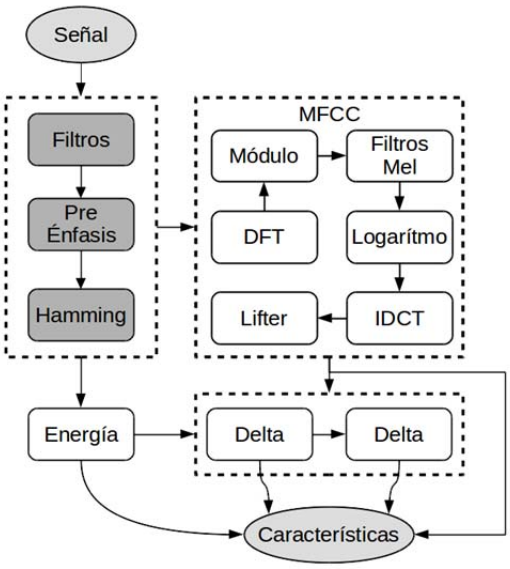

Figura 3. Proceso de extracción de características

V.2. DETECTOR DE PUNTO FINAL

El Detector de Punto Final no es un problema resuelto, ni una tarea trivial cuando se trata de sistemas embebidos [32]. Debe ser suficientemente robusto para hacer frente a diferentes ambientes de ruido, pero también rápido y eficiente, manteniendo bajo costo computacional en relación al proceso de reconocimiento. Un trabajo previo [33] presentó un sencillo pero eficiente algoritmo VAD basado en tres descriptores: *Energía*, *Planitud Espectral* y *Frecuencia Dominante*. El Detector de Punto Final introducido aquí aplica dichos descriptores, pero utiliza un algoritmo diferente para la toma de decisiones. En primer lugar el ruido de fondo se estima a través de un proceso de calibración durante un corto período de tiempo. Durante este proceso se calculan, la media y la desviación estándar para los descriptores, y todos los umbrales se actualizan en consecuencia. Por último, el algoritmo VAD, detallado en Algoritmo 1, ejecutado durante el tiempo de corrida, donde THD significa Umbral, SF significa Planitud Espectral, DF significa Frecuencia Dominante y E significa Energía.

Algoritmo 1 Algoritmo VAD

```
1: for cada segment de habla (m) do
2:     if (SF(m) > THDSF) o [(E(m) > THDE) y
       (THDmin_DF < DF(m) < THDmax_DF)]          then
3:         if age++ > min_age then
4:             Comenzar el procesamiento
5:     else
6:         if está procesando then
7:             if timeout++ > max_timeout then
8:                 Parar el procesamiento
```

V.3. ENTRENAMIENTO

El sistema de entrenamiento consiste en la grabación de muestras para cada palabra a ser reconocida. Este procedimiento genera un vector de características para cada palabra grabada, que se almacena en la memoria. Ellos formarán el *Diccionario*, *Plantillas* o *Vocabulario* del sistema. Para mejorar el rendimiento del sistema es muy deseable diseñar el vocabulario seleccionado maximizando las diferencias fonéticas entre palabras, tales que la probabilidad de confusión entre ellas se reduzca al mínimo.

V.4. RECONOCIMIENTO

Como se mencionó anteriormente, la etapa de reconocimiento de nuestro sistema se basa en el algoritmo DTW. El DTW se utiliza para comparar la emisión con las plantillas pre-grabadas. El algoritmo utiliza la programación dinámica para alinear un par de series temporales cuyas longitudes generalmente difieren. Este proceso caracteriza a la similitud entre las dos series de tiempo, buscando la ruta óptima que minimice la distancia entre ellas. Por lo tanto, este método hace posible la obtención de pequeñas distancias entre dos versiones de la misma palabra, incluso si se pronuncian a diferentes velocidades.

El algoritmo DTW por lo general tiene una heurística asociada para obtener alineaciones razonables. Normalmente, ésta obedece mayormente a tres restricciones: *condiciones de límite*, *continuidad* y *monotonía*. Las restricciones más

comunes son aquellas en donde: las aperturas y los finales de dos secuencias deben alinearse, los caminos locales deben ser continuos y en donde el camino no pueden ir hacia atrás.

Para aplicaciones en tiempo real tales como las reportadas aquí, el tiempo cuadrático y la complejidad espacial del DTW crean la necesidad de métodos para acelerar el algoritmo. Se han propuesto tres estrategias principales [23, 34]: *limitaciones en la búsqueda*, *abstracción de datos* e *indexación*. En este trabajo, se aplicó la primera, que emplea el *Paralelogramo de Itakura*, que reduce el espacio de búsqueda a explorar.

Para optimizar aún más la memoria, nuestro algoritmo no necesita alocar la totalidad del tamaño de la matriz DTW. En lugar de ello, se requieren sólo dos filas para evaluar la ruta local en cada paso. Esto reduce el espacio de asignación de $M \times N$ a $2 \times N$.

VI. RESULTADOS Y VALIDACIÓN

Para validar la ejecución de cada bloque del sistema RPA hemos establecido un modo de depuración, que una vez seleccionado desde el archivo de configuración, guarda los valores de cada variable intermedia. Estas variables pueden ser comparadas numéricamente contra los valores correspondientes producidos por una implementación paralela en Matlab. Los resultados de la prueba mostraron que, la diferencia numérica en el cálculo de extracción de características entre el FPU de precisión simple del STM32F4 y la PC era aceptable (aproximadamente 10^{-4}).

Cuando se observa el procesamiento del algoritmo DTW, aunque el error se mantiene en el mismo intervalo entre palabras de 1 a 2 segundos de duración, éste crece a medida que la longitud de lo enunciado aumenta. Esto es causado por la aritmética de punto flotante y la forma en que funciona el DTW, en donde sumar números consecutivos de diferente orden eleva el error.

La evaluación del algoritmo VAD reveló excelentes segmentaciones en ambientes de alta SNR, y un buen rendimiento cuando ésta disminuye. Se encontró que la medida de Planitud Espectral fue una de las características más fiables para los propósitos del VAD.

Para validar el rendimiento del sistema, se empleó un caso de estudio que consiste en una aplicación de control de robot que contiene 9 palabras. El sistema se entrenó con una sola plantilla para cada palabra y 25 versiones de cada una fueron emitidas en secuencia por un solo locutor.

Este procedimiento se repitió tres veces y se hizo en dos escenarios diferentes: una *Cámara Anecoica* que tiene una muy alta SNR, y en un *Ambiente Normal*, para conseguir una mejor estimación. Las medidas utilizadas para caracterizar el rendimiento del sistema fueron la *tasa de error por palabra* (WER) y el *factor de tiempo real* (RT). La Tabla 1 muestra el rendimiento promedio de cada palabra objetivo. Se encontró una WER promedio general de 1,04% y 2,81% para los escenarios respectivos. El RT reportado fue de 1,43 que está en línea con el nivel de rendimiento reportado en la literatura. Este resultado demuestra la utilidad del sistema desarrollado.

TABLA I
TASA DE ERROR DE PALABRAS DEL SISTEMA

Palabras	**WER**			
	Cámara Anecoica		Ambiente Normal	
	μ[%]	σ[%]	μ[%]	σ[%]
Arriba	0	0	0	0
Abajo	0	0	16	8
Adelante	1,33	2,31	0	0
Izquierda	0	0	2,67	4,62
Derecha	0	0	4	4
Acelerar	2,67	2,31	0	0
Frenar	2,67	2,31	1,33	2,31
Velocidad constante	0	0	0	0
Total	1,04	1,54	2,81	3,53

VII. CONCLUSIÓN

Este trabajo describe el diseño, implementación y validación de un sistema IWR embebido basado en tecnología de DSC para el idioma Español de Argentina. En comparación contra los sistemas RAH completamente embebidos, esta implementación fue capaz de extraer características acústicas complejas según el estado del arte y logró un excelente reconocimiento y rendimiento computacional, a pesar de las limitaciones de la plataforma. Su Front-End calcula características acústicas MFCC, mientras que su componente de reconocimiento de patrones se basa en DTW. El sistema diseñado resultó en una plataforma modular y configurable que será utilizada para la investigación y el desarrollo de aplicaciones de comando y control basados en el habla.

El sistema fue validado contra una aplicación basada en PC, y su rendimiento para el reconocimiento de palabras se evaluó bajo un escenario de control robótico. Los resultados mostraron un buen factor de tiempo real y niveles de WER compatibles con la utilidad práctica del sistema. Además, el sistema demostró un rendimiento superior al de las aplicaciones con microcontroladores y comparable con los sistemas basados en DSPs.

VIII. TRABAJO FUTURO

Se han evaluado otras mejoras. En cuanto a su robustez, las técnicas de reducción de ruido, tales como Filtro Winner o Sustracción Espectral se pueden aplicar sin un incremento considerable de los costos computacionales. Para una mayor precisión de reconocimiento y para un reconocimiento independiente de locutor, se debe implementar la Cuantización Vectorial. En relación con el rendimiento del DTW, se deben probar otras limitaciones y otros algoritmos de aceleración, además de mejorar las restricciones de normalización y de ruta. Por último, las técnicas basadas en estadísticas para el reconocimiento de voz implementadas con HMM están actualmente en desarrollo. Esto hará que sea posible comparar el grado de conveniencia entre la complejidad y el rendimiento del sistema.

REFERENCIAS

[1] P. Sinha, "Speech processing in embedded systems", Springer US, 2010.

[2] U. Suryawanshi and S. R. Ganorkar, "Hardware Implementationof Speech Recognition Using MFCC and Euclidean Distance," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 03, no. 08, pp. 11248–11254, Aug. 2014.

[3] S. Li and H. Ren, "An isolated word recognition system based on DSP and improved dynamic time warping algorithm," *IEEE Int. Conf. Prog. Informatics Comput.*, vol. 1, pp. 136–139, 2010.

[4] K. Joshi, N. Kolhare, and V. M. Pandharipande, "Implementation of Speech Recognition System using DSP Processor ADSP2181," *Int. J. Electron. Signals Syst.*, vol. 1, no. 3, 2012.

[5] J. XinXing and S. Xu, "Speech Recognition Based on Efficient DTW Algorithm and Its DSP Implementation," *Procedia Eng.*, vol. 29, pp. 832–836, Jan. 2012.

[6] Y. Meng, "Speech Recognition on DSP: Algorithm Optimization and Performance Analysis," Ph.D. dissertation, University of Hong Kong, 2004.

[7] T. Sledevic, G. Tamulevicius, and D. Navakauskas, "Upgrading FPGA Implementation of Isolated Word Recognition System for a Real-Time Operation," *Elektron. ir Elektrotechnika*, pp. 123–128, 2013.

[8] A. Aldahoud, H. Atoui, and M. Fezari, "Robust Automatic Speech recognition System Implemented in a Hybrid Design DSP-FPGA," *Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 6, no. 5, pp. 333–342, Oct. 2013.

[9] S. Pan, C. Lai, and B. Tsai, "The implementation of speech recognition systems on FPGA-based embedded systems with SoC architecture," *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 11, pp. 6161–6175, 2011.

[10] M. Dharmale and M. Mahamune, "Robotic Automationthrough Speech Recognition," *Int. J. Sci.Res. Publ.*, vol. 3, no. 6, pp. 1–4, 2013.

[11] H. Heidari, S. Gobee, and N. Jaiswal, "Isolated Word Command Recognition for Robot Navigation," *Eng. Procedia*, vol. 41, no. IRIS, pp. 412–419, Jan. 2012.

[12] A. Vijayaraj and N. Velmurugan, "Limited speech recognition for controlling movement of mobile robot implemented on atmega162 microcontroller," *Int. J. Eng. Sci. Technol.*, vol. 2, no. 10, pp. 347– 350, 2009.

[13] N. Kandpal, Y. Mandke, and A. Patwardhan, "Implementation of Voice Recognition in Low Power Microcontroller," in *Int. Proc. Comput. Sci. Inf. Technol.*, vol. 30, 2012, pp. 111–115.

[14] C.-H. Chang, Z.-H. Zhou, S.-H. Lin, J.-C. Wang, and J.-F. Wang, "Intelligent appliance control using a low-cost embedded speech recognizer," in *Int. Conf. Comput. Netw. Technol.*, no. 1, 2012, pp. 311–314.

[15] Q. Qu and L. Li, "Recognition Module Based on STM32," in *Int. Symp. Commun. Inf. Technol.*, 2011, pp. 73–77.

[16] B. Kamdar, M. Bhisham, and D. Shah, "Real Time Speech Recognition using IIR Digital Filters Implemented on an Embedded System," in *Int. Conf. Commun. Inf. Comput. Technol.*, 2012, pp. 1–5.

[17] V. Naresh, B. Venkataramani, A. Karan, and J. Manikandan, "PSoC based isolated speech recognition system," in *Int. Conf. Commun. Signal Process.*, 2013, pp. 693–697.

[18] Y. Xing and W. Chen, "Design of speech recognition robot based on MCU," in *Int. Conf. Intell. Human-Machine Syst. Cybern.*, vol. 1. Ieee, Aug. 2012, pp. 253–256.

[19] H. Liu, Y. Qian, and J. Liu, "English speech recognition system on chip," *Tsinghua Sci. Technol.*, vol. 16, no. 1, pp. 95–99, 2011.

[20] S. K. Nanda and A. P. Dhande, "Microcontroller implementation of a voice command recognition system for human-machine interface in embedded systems," *Int. J. Electron. Commun. Soft Comput. Sci. Eng.*, vol. 1, no. 1, pp. 5–8, 2005.

[21] S. K. Gaikwad, B. W. Gawali, and P. Yannawar, "A Review on Speech Recognition Technique," *Int. J. Comput. Appl.*, vol. 10, no. 3, pp. 16–24, 2010.

[22] M. Anusuya and S. Katti, "Speech recognition by machine: A review," *Int. J. Comput. Sci. Inf. Secur.*, vol. 6, pp. 181–205, 2009.

[23] L. R. Rabiner and B. Juang, "Fundamentals of speech recognition", *1st ed. Prentice-Hall International*, Inc., 1993.

[24] K. Kroschel and M. Grimm, "Robust speech recognition and understanding", *I-Tech Education and Publishing*, 2007.

[25] T. Kite, "Understanding PDM Digital Audio," *Audio Precision*, Inc, Tech. Rep., 2012.

[26] D. Katz and R. Gentile, "Embedded media processing", *Newnes*, 2006.

[27] L. R. Rabiner and R. W. Schafer, "Introduction to Digital Speech Processing", *Found. Trends Signal Process.*, vol. 1, no. 12, pp. 1–194, 2007.

[28] D. Jurafsky and J. H. Martin, "Speech and Language Processing", *2nd ed. Pearson Education International*, 2008.

[29] M. Nilsson and M. Ejnarsson, "Speech Recognition using Hidden Markov Model performance evaluation in noisy environment," *Ph.D. dissertation*, Blekinge Institute of Technology, 2002.

[30] X. Huang, A. Acerd, and H.-W. Hon, "Spoken Language Processing", *1st ed. Prentice-Hall*, Inc., 2001.

[31] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, "HTK Book", 2002.

[32] P. Khoa, "Noise robust voice activity detection," *Master Thesis*, Nanyang Technological University, 2012.

[33] M. Moattar and M. Homayounpour, "A simple but efficient real-time voice activity detection algorithm," *Eur. Signal Process. Conf.*, pp. 2549–2553, 2009.

[34] S. Salvador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space," *Intell. Data Anal.*, vol. 11, no. 5, 2007.

A. G. Alvarez, Estudiante de Ingeniería en Electrónica de la Universidad Tecnológica Nacional, Facultad Regional Buenos Aires (UTN-FRBA). Investigador del Grupo de Inteligencia Artificial y Robótica de la UTN-FRBA. Exbecario del DAAD. Estudio en la Technishe Univesität Dresden y fue partícipe del grupo de investigación Sprachtechnolgie, Institut für Akustik und Sprachkommunikation. Actualmente es System Developer en Savetrack S.A. Sus líneas de interés son: visión artificial, reconocimiento del habla, inteligencia artificial y sistemas de control. Posee publicaciones en congresos y ha participado en diferentes proyectos de investigación.ff

D. A. Evin, Doctor en Ciencias de la Computación de la Universidad de Buenos Aires, 2011. Bioingeniero de la Universidad Nacional de Entre Ríos, 2005. Investigador Asistente de CONICET desde 2014, del Laboratorio de Investigaciones Sensoriales, INIGEM-UBA CONICET; y docente e investigador del departamento de Computación en la Facultad de Ingeniería, Universidad Nacional de Entre Ríos desde 2003. Sus líneas de interés son: reconocimiento del habla, inteligencia computacional, e ingeniería biomédica. Posee múltiples publicaciones en revistas y congresos y ha participado en diferentes proyectos de investigación nacionales e internacionales.

Sebastián Verrastro, Ingeniero Electrónico de la Universidad Tecnológica Nacional, 2005, MBA, 2010. En 2000 comenzó su carrera docente en la materia Análisis de Señales y Sistemas, siendo en la actualidad profesor interino de la materia Proyecto Final de la carrera de Ingeniería Electrónica de la UTN. Es Investigador UTN categoría "C". Es Director de proyectos PID UTN-FRBA: "Diseño y testing de IC implantables", "Investigación y desarrollo de un IC Transceptor por Radiofrecuencia, Pasivo y de Bajo Consumo", "Desarrollo de Unidad Autónoma Voladora" y "Diseño del Chip de un Tag RFID". Es Director de los siguientes proyectos del Ministerio de Educación en UTN-FRBA: "OPAMP médico y educativo", "Extrusor Filamento Plástico" y dirigió el proyecto ANR600: "Mejora de la productividad en cultivos con Computadora de Abordo". Sus áreas de interés son: microelectrónica, automatización y control, procesamiento de señales e imágenes y gestión.