

Control remoto por voz del robot móvil pioneer P3-DX

Diego Andrés Guffanti Martínez
 Director: Ing. Víctor Proaño Rosero
 Codirector: Ing. Hugo Ortiz

Departamento de Eléctrica y Electrónica, Escuela Politécnica del Ejército, Av. General Rumiñahui S/N. Sangolquí - Ecuador

Resumen— El presente artículo describe el diseño e implementación de un sistema de control remoto por voz para el robot móvil Pioneer P3-DX que posee el Departamento de Eléctrica y Electrónica de la ESPE. Como requisito previo para realizar el control de la plataforma robótica se han integrado dos entornos de programación distintos: MATLAB y ARIA. En el sistema de reconocimiento se utilizan herramientas de procesamiento de señales de manera que se elimine el ruido y se puedan extraer características fundamentales de la voz para que sean comparadas con cada una de las características almacenadas en una base de datos. La señal almacenada con menor medida de distancia a la grabada representa la de mayor similitud.

I. INTRODUCCIÓN

Los sistemas de reconocimiento de voz han sido por mucho tiempo un tema muy estudiado por los investigadores de inteligencia artificial. Hoy en día gracias a los avances tecnológicos contamos con herramientas que nos permiten continuar con estos estudios y comprender cada vez más la anatomía y funcionamiento del aparato fonador.

Gracias a MATLAB se hace posible el análisis y procesamiento de las señales de voz. Es por ello que una integración de este programa con ARIA resultaría de trascendental importancia tanto para este proyecto como para investigaciones futuras usando las plataformas robóticas Pioneer.

II. SISTEMA DE RECONOCIMIENTO DE VOZ

A. Obtención de información mediante micrófono

El sistema de reconocimiento de voz diseñado, emplea el micrófono inalámbrico Wireless tipo diadema marca EB-330H para la recepción de la voz mostrado en la figura 1.



Figura 1. Micrófono inalámbrico tipo diadema EB-330H

B. Muestreo

Teniendo en cuenta el teorema de muestreo de Nyquist y que el ancho de banda en el cual se encuentra la señal de voz humana es 100 - 3500Hz, la frecuencia de muestreo elegida es de 8000 muestras/s. Debido a que el formato de audio wav es un formato propio de Windows, con 8 o 16 bits y con índices de muestreo recomendados de 11,025 kHz, 22,05 kHz o 44,1 kHz, para configurar la función wavrecord desde MATLAB se emplea una frecuencia de muestreo de la señal de voz de $F_s=11025\text{Hz}$, de manera que almacena 11025 datos cada segundo de grabación, lo cual puede ser considerado una buena calidad de sonido. La figura 1 muestra la señal de voz grabada durante 2 segundos usando la función wavrecord de MATLAB.

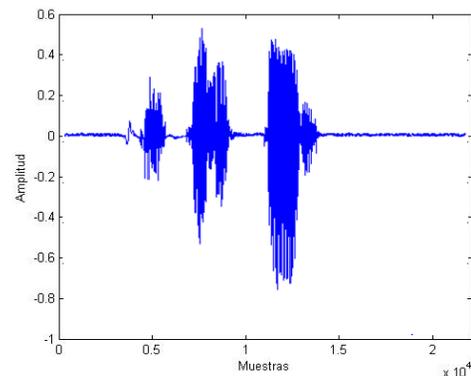


Figura 2. Señal de voz sin procesar muestreada a 11025 Hz durante 2 segundos.

C. Eliminación del ruido

Para este propósito se escogieron segmentos de 10ms. Si en un segmento la energía promedio es menor que un valor umbral proporcional a la energía promedio de la señal entera, dicho segmento es descartado. El valor umbral escogido es 20%

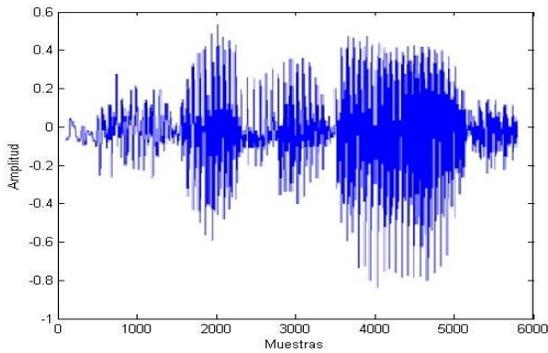


Figura 3. Señal de voz con ruido eliminado

D. Filtro de preénfasis

La etapa de preénfasis se realiza con el propósito de suavizar el espectro y reducir las inestabilidades de cálculo asociadas con las operaciones aritméticas de precisión finita. Además se usa para compensar la caída de -6 dB que experimenta la señal al pasar a través del tracto vocal.

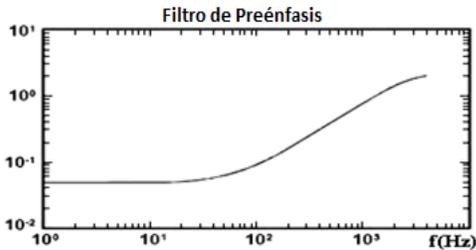


Figura 4. Respuesta frecuencial del filtro de preénfasis

E. Segmentación y Ventaneo

Debido a la inercia inherente a los órganos articulatorios se puede suponer que las características y propiedades de la señal de voz no varían apreciablemente en un intervalo corto de tiempo, que aproximadamente oscila entre [20 30ms], por lo cual es posible realizar un análisis espectral cuasi-estacionario sobre segmentos de señal de esta duración temporal. Durante la segmentación, la señal de voz es agrupada en tramas de 30ms (240 datos) con un desplazamiento inicial de 10ms.

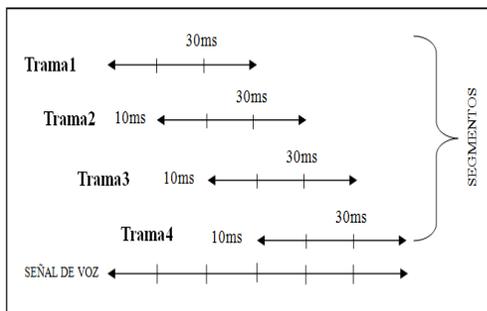


Figura 5. Segmentación de la señal de voz

Es necesario procesar las tramas de voz a través de una ventana de Hamming para eliminar los problemas causados por los cambios rápidos de la señal en los extremos de cada trama de voz. Por esta misma razón la segmentación se aplica con un desplazamiento inicial (para conseguir transiciones suaves entre tramas).

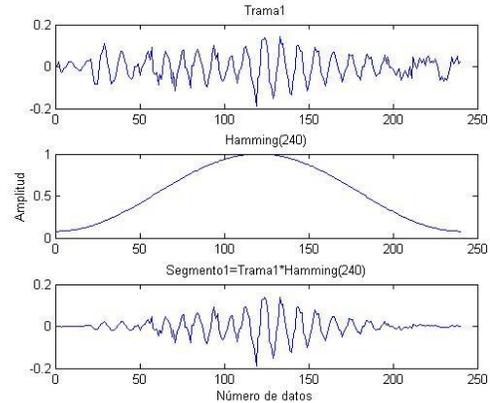


Figura 6. Ventaneo Hamming de un segmento o trama de la voz

F. Cepstrum

El cepstrum es una transformación común que se utiliza para obtener información de señal de voz de una persona. Se puede utilizar para separar la señal de excitación (que contiene las palabras y el tono) y la función de transferencia (que contiene la calidad de la voz).

El proceso de generación de la voz depende de dos factores básicos, uno de baja frecuencia llamado excitación $E(w)$, y uno de alta frecuencia generado por el sistema del tracto vocal $\theta(w)$. La convolución de estos dos términos genera la señal de voz en el dominio de la frecuencia como se muestra en la ecuación 1.

$$S(w) = E(w)\theta(w) \quad \text{Ecuación 1}$$

Aplicando la definición de cepstrum se obtiene la ecuación 2.

$$\begin{aligned} C_s(w) &= \log|S(w)| \\ C_s(w) &= \log|E(w)\theta(w)| \\ C_s(w) &= \log|E(w)| + \log|\theta(w)| \\ C_s(w) &= C_e(w) + C_\theta(w) \end{aligned} \quad \text{Ecuación 2}$$

Finalmente al regresar a la señal al dominio del “tiempo” mediante la Transformada Inversa de Fourier (llamado generalmente dominio quefrequency) se puede aplicar un filtro que permite pasar la parte más baja de la señal. Ésta etapa se denomina liftering.

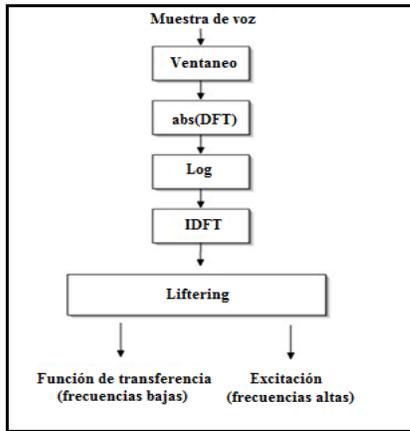


Figura 7. Etapas para el cálculo del cepstrum de una señal

Luego de la etapa liftering, se obtiene la señal de voz debida únicamente al sistema o tracto vocal y no a la excitación. A partir de esta señal se pueden obtener los coeficientes cepstrales mostrados en la figura, que serán usados para el sistema de reconocimiento de voz.

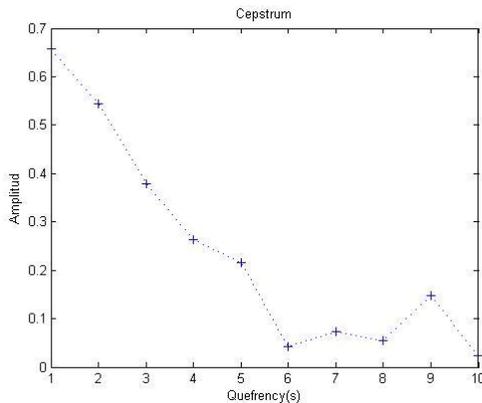


Figura 8. Coeficientes cepstrales para un segmento o trama de voz

G. Distancia

Una vez determinado lo anterior, se mide la distancia Euclidiana punto a punto entre las dos matrices cepstrum correspondientes a la palabra dicha y la almacenada en una base de datos de MATLAB. Para ello se ocupa la ecuación 3.

$$d = \sqrt{\sum_{i=1}^D |f_i - f'_i|^2} \quad \text{Ecuación 3}$$

Al generar un vector que almacene las medidas de distancia obtenidas, podemos identificar cuál es la menor de ellas y de ésta manera deducir cuál fue la palabra reconocida.

III. INTERFAZ MARLAB-ARIA

La interfaz de MATLAB-ARIA es una aplicación escrita en lenguaje C++, que permite llamar a muchos comandos y acciones de ARIA desde MATLAB. Los programas escritos en código C++ y que pueden ser llamados desde MATLAB se denominan Archivos-MEX. Estos Archivos-MEX poseen

subrutinas que MATLAB automáticamente puede cargar y ejecutar.

Para comprender las entradas y salidas que posee una función MEX se puede analizar la siguiente línea de código:

```
void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const
                 mxArray *prhs[])
```

La tabla 1 muestra los argumentos de una función MEX.

C/MEX	SIGNIFICADO	CÓDIGO .M
Nlhs	Número de variables de salida	Nargout
Plhs	Arreglo de punteros mxArray para las variables de salida	Varargout
Nrhs	Número de variables de entrada	Nargin
prhs	Arreglo de punteros mxArray para las variables de entrada.	varargin

Tabla 1. Argumentos de una función MEX

La compuerta de comunicación entre los software MATLAB y ARIA está escrita en lenguaje C++ desde Microsoft Visual Studio.NET 2003 y ha sido compilada en MATLAB R2009a con el propósito de generar un archivo MEX llamado ariainterface.mex. Por ejemplo si desde el entorno de MATLAB se desea ordenar al robot girar hacia la derecha, con una velocidad de 5grados/s, se procede de la siguiente manera:

```
ariainterface(4,5)
```

Donde 4 corresponde al número de función implementada en el código C++, en este caso es la función setRotVel, y 5 es el valor de la velocidad deseada en grados/s. Entonces tenemos:

```
Nrhs =2      Número de datos de salida
Prhs[0]=4    Función 4, setRotVel.
Prhs[1]=5    Velocidad 5°/s
```

La figura 9 muestra el esquema de cómo la interfaz MATLAB-ARIA interactúa con MATLAB, el entorno de simulación MobileSim y con el robot Pioneer P3-DX.

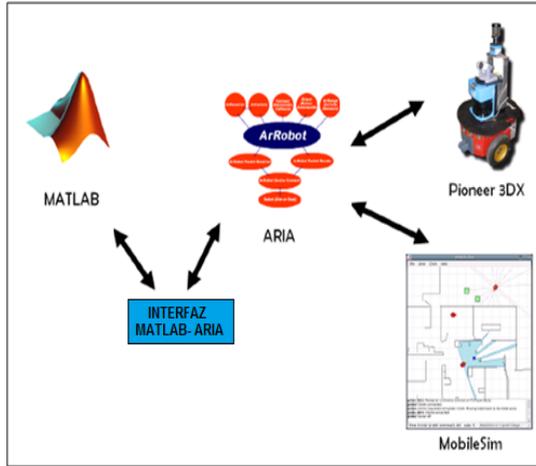


Figura 9. Funcionamiento de la interfaz MATLAB-ARIA

COMANDO DE VOZ	ACCIÓN
Conectar	Establece la conexión entre la aplicación y el simulador MOBILESIM o el robot físico.
Fin	Desconecta la aplicación del simulador MOBILESIM o el robot físico
Adelante	El robot se mueve hacia adelante con la velocidad y aceleración establecidas en el panel de configuración
Atrás	El robot se mueve hacia atrás con la velocidad y aceleración establecidas en el panel de configuración
Paro	El robot se detiene.
Giro uno	El robot gira hacia la derecha.

Tabla 2. Comandos de movimiento del robot

IV. CONTROL POR VOZ DEL ROBOT PIONEER P3-DX EMPLEANDO LA INTERFAZ MARLAB-ARIA

El sistema, permite controlar remotamente por voz el robot móvil Pioneer P3-DX desde MATLAB. Se ha desarrollado mediante una recopilación de dos programas: El Sistema de Reconocimiento de Voz y la interfaz MATLAB-ARIA. El entorno gráfico del sistema es el que se muestra en la figura 10.

V. EXPERIMENTACIÓN

Debido a que la plataforma Pioneer P3-DX es para ambientes interiores, las pruebas se llevaron a cabo en entornos interiores y estáticos es decir sin presencia de objetos en movimiento. Estas pruebas se efectuaron tanto con el simulador MOBILESIM como con el robot Pioneer P3-DX del laboratorio de robótica de la Escuela Politécnica del Ejército. La figura 11 y 12 muestran las pruebas de funcionamiento del sistema.

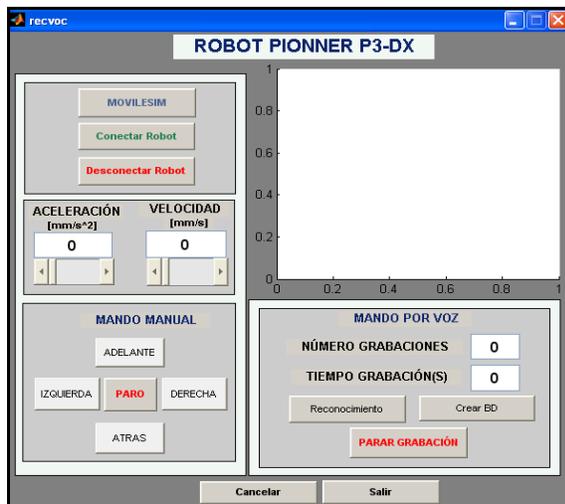


Figura 10. Interfaz de la aplicación de CONTROL REMOTO POR VOZ desarrollada en MATLAB

El usuario del sistema posee siete comandos de voz para controlar el robot, entre ellos encontramos: Conectar, Fin, Adelante, Atrás, Paro, Giro uno y Giro dos.

En la tabla 2 se describen las acciones que realiza cada uno de los comandos de voz cuando una palabra es reconocida correctamente.

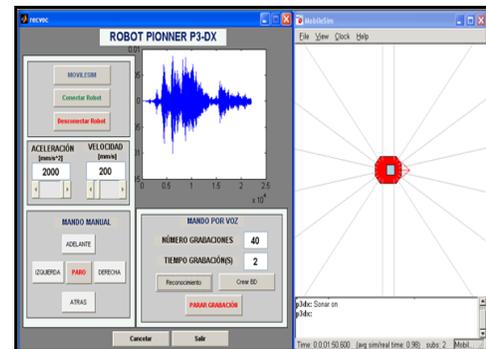


Figura 11. Pruebas de funcionamiento PC - SIMULADOR MOBILESIM



Figura 12. Pruebas de funcionamiento PC-ROBOT

Con el propósito de advertir al usuario que el sistema se encuentra en línea se incorporó un sonido de alerta. En la tabla 3 se muestran los resultados considerando que el operador pronuncia una palabra al escuchar el sonido de alerta:

COMANDO	ACIERTOS	DESACIERTOS
Conectar	4	0
Fin	3	1
Adelante	4	0
Atrás	4	0
Paro	4	0
Giro Uno	2	2
Giro Dos	4	0
SUMA	25	3
	TOTAL CASOS	28

Tabla 3. Resultados obtenidos para las pruebas finales

Los resultados de la tabla anterior se muestran en la figura 13.



Figura 13. Taza de aciertos para pruebas finales

La fórmula empleada para el cálculo de la tasa de aciertos se muestra en la ecuación 4.

$$\text{tasa de aciertos}(\%) = \frac{\text{No.de aciertos}}{\text{No.de casos}} \times 100 \quad \text{Ecuación 4}$$

$$\text{tasa de aciertos}(\%) = \frac{25}{28} \times 100$$

$$\text{tasa de aciertos}(\%) = 89.29\%$$

VI. CONCLUSIONES

- El sistema de reconocimiento de voz requirió la implementación de una interfaz entre MATLAB y ARIA. Esta comunicación se implementó en código C++ mediante el uso de archivos MEX, que toman el dato de la palabra reconocida en MATLAB y la traducen a un comando directo de ARIA.

- Para realizar el reconocimiento es necesario analizar la señal de voz de manera segmentada para entender la evolución temporal, ya que si se usa una ventana muy grande los cambios locales se omiten y si es muy pequeña se reflejan demasiados cambios puntuales.
- El uso de MATLAB permitió reducir la complejidad del procesamiento digital de señales y ayudó a filtrar la voz para enfatizar únicamente las características acústicas que dependen del tracto vocal, llamadas formantes.
- La interfaz gráfica desarrollada empleando la herramienta GUIDE de MATLAB tiene un diseño flexible, que puede ser mejorado a fin de incorporar nuevas funciones al sistema.
- La interfaz de comunicación fue escrita en lenguaje C++ con el compilador de Visual Studio .NET 2003 para evitar conflictos debido a que de esa manera están generadas las librerías de ARIA.

VII. REFERENCIAS

- Borgstrom, J. (3 de Abril de 2005). ARIA and Matlab Integration With Applications. Recuperado el 21 de Noviembre de 2012, de <http://www8.cs.umu.se/education/examina/Rapporter/JonasBorgstrom.pdf>
- Velásquez, G. (26 de Abril de 2008). Sistema de reconocimiento de voz en MATLAB. Recuperado el 14 de Marzo de 2013, de http://biblioteca.usac.edu.gt/tesis/08/08_0223_EO.pdf
- Getreuer, P. (15 de Abril de 2010). *Writing MATLAB C/MEX Code*. Recuperado el 22 de Marzo de 2013, de <http://classes.soe.ucsc.edu/ee264/Fall11/cmex.pdf>
- MobileRobots. (Enero de 2006). *Pioneer 3 Operation Manual*. Recuperado el 9 de Febrero de 2013, de http://www.ist.tugraz.at/_attach/Publish/Kmr06/pioneer-robot.pdf

VIII. BIOGRAFÍA



Guffanti Martínez Diego Andrés. Nació en Riobamba, Ecuador, el 1 de Enero de 1990. Se graduó en la Escuela Particular Pio XII de la ciudad de Santo Domingo en el 2002 y estudió la especialidad de Físico Matemático en el Instituto Técnico Superior Julio Moreno Espinosa, donde se graduaría en el año 2008, actualmente

es egresado de la carrera de Ingeniería Electrónica en Automatización y Control de la Escuela Politécnica del Ejército, Campus Sangolquí. Hoy en día ejerce la docencia en la Unidad Educativa Federico González Suárez y en el instituto de asesorías académicas DELTA. Entre sus campos de interés están los fenómenos físicos y electrónicos.
e-mail: diegogufy090@hotmail.com
telf.: (593)02 2755494