IDENTIFICACIÓN DE COMANDOS DE VOZ UTILIZANDO LPC Y ALGORITMOS GENÉTICOS EN MATLAB®

Pérez Ibarra, J.[1]; Borrero Guerrero, H.[2]

Resumen: *Numerosas aplicaciones informáticas y tecnológicas requieren de la construcción de herramientas de hardware y software con alta capacidad de procesamiento de información y búsqueda de soluciones. En esta línea se encuentran el procesamiento de voz aplicado al control de procesos. Este documento describe el diseño e implementación de un sistema prototipo para la identificación de comandos de voz por computador, con el propósito de controlar la navegación de un agente robótico. El sistema consta de una aplicación para la adquisición y procesamiento de la señal de voz, y de un computador con tarjeta de sonido. Para el proceso de identificación de comandos de voz se aplicó el método LPC y el método de distorsión en el tiempo así como algoritmos genéticos.*

Palabras Clave: *Codificación por predicción lineal, robótica móvil, comparación de patrones, algoritmo genético.*

Abstract: *Several technological applications require the construction of hardware and software tools with high capacity for processing information and searchingof solutions. In this line lies the voice processing applied to process control. This paper describes the design and implementation of a prototype system for identifying voice commands by using a computer. The aim isto control the navigation of a robotic agent. The system consists of a software application for acquiring and processing the voice signal, and a computer with a sound card. The commands identification process is applied the LPC method(linear predictive coding) for speech signal, the method of distortion in time and genetic algorithms for finding solutions. Finally, communication interface navigation commands to the mobile robot are used.*

Key words: *Linear prediction coding, Mobile robotics, Pattern comparison, Genetic Algorithm*

[1]Ingeniero Electrónico
Universidad de los Llanos, Villavicencio, Colombia
juancapml15@hotmail.com

[2]Ingeniero Electrónico,Estudiante de Doctorado en Ingeniería Mecánica
Universidad de Sao Paulo – EESC, Sao Paulo, Brasil
h_borrelo@ieee.org

Grupo de Investigación en robótica (*giro*)
Facultad de Ingeniería de la Universidad de los Llanos

1. INTRODUCCIÓN

Con la implementación de sistemas de reconocimiento de voz se busca identificar y reunir todo el conocimiento presente en la señal de voz que emite un hablante (acústica, fonemas, palabras, significados y contextos). (Alezones, 2009;Chou, 2003).

La aproximación acústica-fonética al reconocimiento de la voz postula que el habla está compuesta por una secuencia de fonemas; que hay un número finito de estos en el lenguaje; que cada fonema puede ser caracterizado por un conjunto de propiedades, las cuales están presentes en la señal de voz y en su espectro; y por último que varias expresiones de un mismo fonema presentan características similares independientemente de su duración, intensidad y del locutor. Sin embargo, la implementación de un sistema basado en la mencionada aproximación acústico-fonética es muy compleja y poco innovadora (Chou, 2003).

En el mismo contexto de los sistemas computacionales para el reconocimiento de voz una aproximación más moderna propone identificar patrones del habla utilizando un método estadístico en el cual se debe ejecutar un subproceso de extracción de características, que por lo general es el resultado de alguna clase de análisis espectral; un subproceso de clasificación de patrones, en el cual se halla una medida de la similitud entre los patrones (en este caso de voz y vocalización) adquiridos frente a los patrones representativos y un subproceso de toma de decisión sobre el patrón correspondiente al sonido adquirido (Alezones, 2009; Chou, 2003).

En este documento se presenta una aplicación en la cual se tiene por objetivo general la identificación de palabras aisladas a partir de un sistema de auto-aprendizaje sin analizar coherencias o significados, y se exponen los diversos algoritmos y técnicas utilizadas para la extracción de características en la aplicación. En la Sección I se presentan los conceptos generales relacionados con el modelo LPC para reconocimiento de comandos voz, el *algoritmo de durbin-levinson,* el proceso de clasificación de patrones y algoritmos de búsqueda. En la sección II se exponen de forma general los aspectos relacionados con la implementación completa de un sistema de reconocimiento de voz consiste de la reunión de dispositivos y algoritmos que busca identificar y agrupar todo el conocimiento presente en una señal (comando) de voz y los módulos que lo componen, para luego presentar los agradecimientos, conclusiones y referencias correspondientes.

2. CONCEPTOS GENERALES

El desarrollo de la aplicación expuesta tiene cuatro fundamentos de carácter teórico, los cuales determinan cada aspecto del diseño y la implementación del sistema. Estos fundamentos son: el modelo de coeficientes de predicción lineal (LPC) para el reconocimiento de voz(Rabiner & Juang, 1993), el algoritmo de Durbin-Levinson para la solución de matrices Toeplitz(Rabiner & Juang, 1993), el alineamiento y normalización temporal de secuencias de parámetros como técnicas de clasificación de patrones(Chou & Juang, 2003). Finalmente los algoritmos genéticos como algoritmos de búsqueda de soluciones(Russell & Norvig, 2010).

A. MODELO LPC PARA RECONOCIMIENTO DE COMANDOS VOZ

Una señal de voz digitalizada puede ser modelada como un conjunto de parámetros en el dominio del tiempo, asumiendo que cada muestra $s[n]$ $s[n]$ de la señal es una combinación lineal de los pp datos anteriores; así, es posible realizar una predicción lineal de la señal al modelarla como el filtro FIR de orden p representado en la *ecuación 1,* donde los coeficientes $\hat{a}_k \hat{a}_k$ son asumidos como constantes (Chou, 2003;Oppenheim, 1999;Rabiner, 1993).

$$s(n) \approx \hat{a}_1 s(n-1) + \hat{a}_2 s(n-2) + \cdots + \hat{a}_p s(n-p)$$

(Eq.1)

Este modelo es útil especialmente en las tramas donde la señal de voz es cuasi-estacionaria, en las cuales el filtro FIR con coeficientes $\hat{a}_k \hat{a}_k$ constantes ofrece una buena aproximación al modelo del tracto vocal de la producción de la voz que se muestra en la Figura 1, en el cual la voz humana$s(n) s(n)$ es modelada por la señal $u(n) u(n)$resultado de la conmutación entre un tren de impulsos de frecuencia controlada y una señal de ruido aleatorio, que se someten a un control de ganancias GG que determina la intensidad de la voz y a un filtro digital variante en el tiempo.

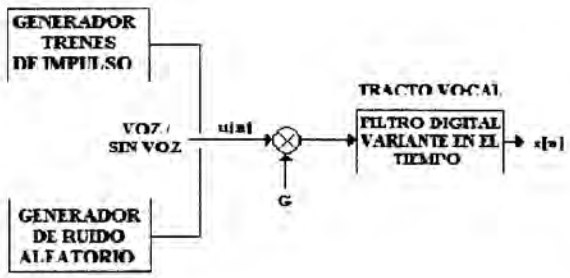

Figura 1. Modelo de producción de voz para LPC

El problema del análisis de predicción lineal básicamente consiste en determinar los coeficientes $\hat{a}_k \hat{a}_k$ del filtro directamente de la señal de voz, de forma que la respuesta en frecuencia de éste coincida con la respuesta en frecuencia de la señal de voz analizada. Este problema se reduce a la función de auto-correlación presente en la *ecuación 2*, y al sistema de ecuaciones de predicción lineal presente en la *ecuación 3*. (Alezones, 2009; Rabiner, 1993).

$$r_n(m) = \sum_{n=1}^{N-m} s_n[n] s_n[n+m], \quad 1 \le m \le p$$

(Eq.2)

$$r_n(i) = \sum_{k=1}^{p} \hat{a}_k r_n(|i-k|), \quad 1 \le i \le p$$

(Eq.3)

En donde NN es el número de muestras de la señal.

La solución de estas ecuaciones implica calcular $r_n(i)$ para $1 \le i \le p r_n(i)$ para $1 \le i \le p$ para luego resolver el sistema de ecuaciones simultáneas de orden p, que se expresa en forma matricial en la *ecuación 4*.

$$\begin{bmatrix} r_n(1) \\ r_n(2) \\ \vdots \\ r_n(p) \end{bmatrix} = \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_p \end{bmatrix} \begin{bmatrix} r_n(0) & r_n(1) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & & r_n(p-2) \\ \vdots & & \ddots & \vdots \\ r_n(p-1) & r_n(p-2) & \cdots & r_n(0) \end{bmatrix}$$

(Eq.4)

Esta matriz $R_{pxp} R_{pxp}$ no solo es simétrica, sino también tiene todos los elementos de las diagonales iguales, se conoce como matriz Toeplitz y es eficientemente solucionada por el Algoritmo de Durbin-Levinson (Rabiner, 1993).

B. ALGORITMO DE DURBIN-LEVINSON

El algoritmo de Durbin-Levinson es un algoritmo recursivo que soporta la solución de sistemas de ecuaciones lineales asociados a una matriz simétrica Toeplitz y normalmente se utiliza para obtener los conjuntos de parámetros LPC $(\hat{a}_k)$ $(\hat{a}_k)$ a partir de un conjunto de coeficientes de auto correlación $(r_n)(r_n)$ (Rabiner, 1993)

El método consiste en proceder recursivamente empezando con un predictor de orden 1, donde posteriormente se incrementa el orden recursivamente usando la solución de orden menor para obtener la solución de orden siguiente. Este algoritmopuede expresarse por medio de las siguientes ecuaciones recursivas:

$$E^{(0)} = r_n(0)$$

$$k_i = \frac{-\left\{r_n(i) - \sum_{j=1}^{i-1} \hat{a}_j^{(i-1)} r(|i-j|)\right\}}{E^{(i-1)}}, \quad 1 \le i \le p$$

$$\hat{a}_i^{(i)} = k_i$$

$$\hat{a}_j^{(i)} = \hat{a}_j^{(i-1)} - k_i \hat{a}_{i-j}^{(i-1)}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)}$$

(Eq.5-9)

C. CLASIFICACIÓN DE PATRONES

Las diversas *técnicas de clasificación de patrones* buscan situar el conjunto de características o parámetros extraídos en una clase o grupo previamente definido. En la aplicación expuesta se utiliza la técnica de medición de distancia entre patrones sujeta al alineamiento y normalización temporal de las secuencias a comparar. (Chou & Juang, 2003)

1) Medición de la distancia entre dos patrones

Si $\hat{a}_{xk} = [x_1, x_2, x_3 \dots x_{T_x}] \hat{a}_{xk} = [x_1, x_2, x_3 \dots x_{T_x}]$ es el patrón representativo de un fonema identificado y $\hat{a}_{yk} = [y_1, y_2, y_3 \dots y_{T_y}] \hat{a}_{yk} = [y_1, y_2, y_3 \dots y_{T_y}]$ es el patrón correspondiente a un fonema por identificar, donde $x_i x_i$ y $y_i y_i$ son los i-esimos vectores de parámetros en los patrones, y $T_x T_x$ y $T_y T_y$ son las longitudes de estos; la distancia entre $\hat{a}_{xk}\hat{a}_{xk}$ y $\hat{a}_{yk}\hat{a}_{yk}$ se representa por $d\left(x_{i_x}, y_{i_y}\right)$ $d\left(x_{i_x}, y_{i_y}\right)$ ó $d(i_x, i_y) d(i_x, i_y)$ y se define como:

$$d(\hat{a}_{xk}, \hat{a}_{yk}) = d(i_x, i_y) = \sum_{i=1}^{min(T_x, T_y)} d(x_i, y_i)$$

(Eq.10)

2) Alineamiento y normalización temporal

Tomando como referencia la *ecuación 10*, el valor de $d(i_x, i_y) d(i_x, i_y)$ debe ser subjetivamente significativo, matemáticamente coherente e indiferente a las fluctuaciones de la velocidad del habla, así que es necesario normalizar toda tasa de fluctuación presente en la velocidad de habla y alinear las secuencias $\hat{a}_{xk}\hat{a}_{xk}$ y $\hat{a}_{yk}\hat{a}_{yk}$ en el dominio del tiempo para conservar el orden en la expresión. La medición de la distancia de una forma más general debe integrar unas funciones de distorsión $\varphi_x \varphi_x$ y $\varphi_y \varphi_y$, que permitan determinar la distorsión acumulada al final de la expresión a evaluar. El "mejor" par de funciones de distorsión equivale al mejor "camino" a través de un plano $(i_x, i_y)(i_x, i_y)$ formado por las dos expresiones a comparar, escoger ese camino implica escoger entre un gran número de pares de funciones, lo cual es un interesante problema de minimización como se puede apreciar en la *ecuación 11*.

$$d(\hat{a}_{xk}, \hat{a}_{yk}) \triangleq \min_{\varphi} d_{\varphi}(\hat{a}_{xk}, \hat{a}_{yk})$$

(Eq.11)

3) Restricciones de normalización

Las funciones de distorsión anteriormente descritas son significativas para la normalización temporal entre diferentes instancias de una expresión, siempre que se impongan algunas restricciones a dichas funciones, pues una minimización sin restricciones puede fácilmente dar como resultado caminos perfectos para expresiones enormemente distintas. Las siguientes son típicas restricciones que se consideran necesarias para el adecuado alineamiento y normalización temporal:

- Restricciones de punto final

Los puntos de inicio y fin para las funciones de distorsión son los límites fijos de las expresiones en el tiempo, tal como se puede apreciar en la *ecuación 12*.

$$\varphi_x(1) = 1, \quad \varphi_y(1) = 1,$$
$$\varphi_x(T) = T_x, \ \varphi_y(T) = T_y$$

(Eq.12)

- Condiciones de monotonía

Como se puede apreciar en la *ecuación 13*, se elimina la posibilidad de funciones de distorsión con pendientes negativas, las cuales generarían marchas atrás en la alineación temporal:

$$\varphi_x(k+1) \geq \varphi_x(k), \qquad \varphi_y(k+1) \geq \varphi_y(k)$$

(Eq.13)

- Restricciones de continuidad local

Se reduce al mínimo la posibilidad de omitir segmentos a la señal de voz al determinar muy claramente los pasos. Un camino dado como una secuencia de pasos, equivale a pares de incrementos en las coordenadas indexados del plano. Por ejemplo, si se tienen los pasos P_1® (1,1)(0,1), P_2® (1,0) y P_3® (0,1)(1,0), un camino C = { (1,1)(0,1)(1,0)(0,1)(1,0)} es el camino recorrido por la secuencia de pasos P_1, P_2 y P_3; y, dado que la posición inicial es (1,1), las posiciones a lo largo del camino son [(1,1)(2,3)(3,3)(4,4)].

- Restricciones de camino global

Cada restricción de continuidad local permite una expansión distinta para los caminos, esto causa que ciertas posiciones del plano (i_x, i_y) (i_x, i_y) no puedan hacer parte de los caminos y por lo tanto quedan excluidas de la solución. Esta nueva restricción determina el rango de puntos en el plano que pueden alcanzarse en el camino que une el punto inicial $(1,1)(1,1)$ con el punto final $(T_x,T_y)(T_x,T_y)$.

D. ALGORITMOS DE BÚSQUEDA

Un algoritmo de búsqueda es un método mediante el cual un agente puede encontrar la secuencia de pasos que tenga el mejor rendimiento en la solución de un problema (Russell & Norvig, 2010)

En el caso presentado en este documento el problema consiste en el desarrollo de la *ecuación 11*, el objetivo es construir un camino entre el punto inicial $(1,1)(1,1)$ y el punto final (T_x,T_y) (T_x,T_y) cuya solución optima es el camino con el menor valor de $d_\varphi d_\varphi$. Para dicho propósito se optó por el desarrollo de un algoritmo genético (Russell & Norvig, 2010) y se rechazó la ampliamente utilizada técnica DTW (Alezones 2009) &(Rabiner & Juang, 1993). Esta decisión se tomópor dos razones, el algoritmo genético puede obtener soluciones muy cercanas a las óptimas con mucho menos recursos de memoria y tiempo que DTW; el algoritmo genético es un método poco aplicado a la identificación de comandos

En (Russell & Norvig, 2010) se plantea que un algoritmo genético es un algoritmo de búsqueda. Inicialmente este genera un conjunto aleatorio de soluciones (población inicial) que han sido representadas en una cadena de datos de un alfabeto finito (preferiblemente binario), cada solución se evalúa mediante una función para determinar si es una solución completa del problema.A partir de esta población inicial, en cada iteración del algoritmo es generado un conjunto de sucesores que son el resultado de la combinación de dos estados anteriores o de la modificación de alguno de estos, esto se repite hasta que se alcance un valor conveniente a la función de evaluación. Su nombre deriva de la evidente similaridad con el proceso evolutivo de selección natural y del proceso de reproducción sexual. (Russell & Norvig, 2010)

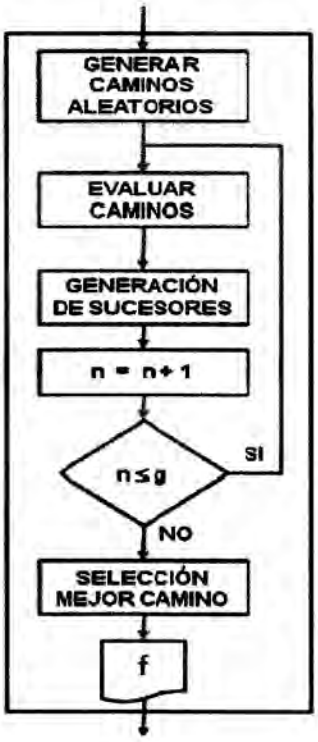

Figura 2. Diagrama de flujo para el algoritmo genético

De acuerdo con el diagrama de flujo que se muestra en la figura 2, el algoritmo comienza con un conjunto de caminos generados aleatoriamente (población), donde cada camino (individuo) está representado en una cadena de un alfabeto finito. En este caso, cadenas con [1 2 3] según el paso dado desde cada punto en el camino. Ej. [1 1 3 2 1 2 1 3]. Luego cada camino se mide con una función de evaluación, en este caso, la distorsión acumulada a lo largo del camino. Como tercer paso la probabilidad de ser elegido es una función decreciente (inversamente proporcional) al valor resultante en la función de evaluación. Se procede a cruzar pares de individuos seleccionados aleatoriamente (reproducción), y cada uno de los nuevos individuos se somete a una "poco probable" mutación aleatoria. Enseguida se vuelve al paso 3 hasta que se logre una medida aceptable para algún camino, o hasta realizar un número de iteraciones (generaciones) del algoritmo.

3. IMPLEMENTACIÓN

La implementación completa de un sistema de reconocimiento de voz consiste de la reunión de dispositivos y algoritmos que busca identificar y agrupar todo el conocimiento presente en una señal de voz. En este documento se propone y expone un sistema de reconocimiento de palabras indiferente del hablante.

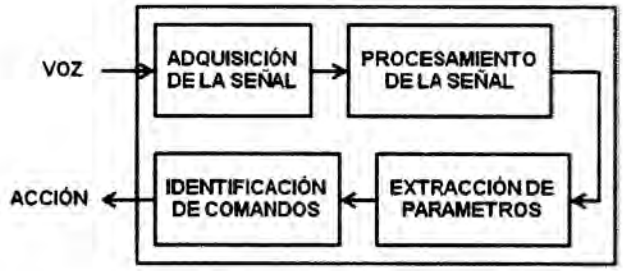

Figura 3. Diagrama de bloques del sistema de reconocimiento

El reconocimiento de palabras constituye un problema complejo que involucra manipulación de hardware, procesamiento de señales, métodos de análisis de datos y algoritmos de búsqueda de soluciones. Estas cuatro tareas definen los componentes del sistema tal como se representa en el diagrama de bloques de la figura 4, que como se puede apreciar está constituido por un módulo de adquisición de datos que convierte el fenómeno físico de la voz en una señal eléctrica; un módulo de procesamiento de datos que realiza las transformaciones adecuadas a la señal para que pueda extraerse de ella información relevante; un módulo de extracción de características que toma la señal procesada y por medio de métodos estadísticos extrae un conjunto de parámetros que caracterizan la señal y un módulo de identificación de comandos cuyos algoritmos buscan determinar a qué comando corresponde la señal de voz adquirida. La figura 4 muestra el diagrama de flujo completo del sistema.

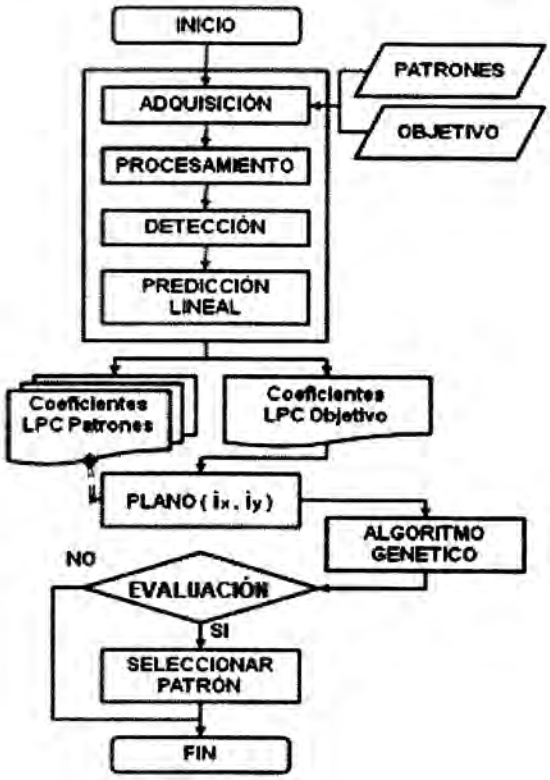

Figura 4. Diagrama de flujo del sistema de reconocimiento

Previo a la puesta en marcha de la navegación del robot móvil se almacenan los coeficientes LPC de los patrones a comparar, luego el sistema es habilitado para adquirir algún comando que controlara la navegación del robot (objetivo), y determinar en él un patrón de coeficientes LPC.

A. MODULO DE ADQUISICION DE DATOS

El primer paso en el proceso de identificación de comandos de voz consiste en la implementación de un sistema de adquisición de datos, la función de éste componente básicamente se resume en capturar una señal análoga de voz por medio de un micrófono conectado a la tarjeta de sonido de un computador la cual ésta amplifica y muestrea la señal de voz convirtiéndola en una señal eléctrica discreta, luego convierte ésta en un código binario (cuantización) que es transferido por medio de uno o más canales y por último, almacenarlo en la memoria del sistema. Este módulo convierte cada segundo de señal acústica en un vector de datos $s[n]s[n]$ de 16000 muestras cuantizados a 16 bits. $fs = 16000\ Hz$ $fs = 16000\ Hz$.

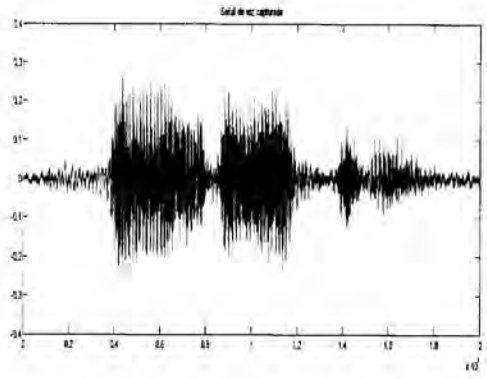

Figura 5. Señal acústica correspondiente a la palabra "DERECHA".

B. MODULO DE PROCESAMIENTO

Este módulo realiza tres transformaciones a la señal que facilitan la extracción de información relevante de ella, estas son: (1) un filtrado que normalice el espectro de la señal de voz, acentuando las altas frecuencias; (2) un segmentado que cumple una doble función de reducir la complejidad del cálculo y obtener tramas cuasi-estacionales; y (3) un ventaneo que compense el fenómeno de Gibbs producido en el segmentado

1) Filtro de preénfasis

Al producirse la voz se atenúan las frecuencias más altas mientras se acentúan las más bajas, de modo que con el fin de normalizar espectralmente la señal, esta se procesa en un filtro digital pasa-altas de bajo orden, el cual se representa con la ecuación (eq.14).

$$H(z) = 1 - a \cdot z^{-1} \rightarrow \ddot{S}[n] = S[n] - a \cdot S[n-1] \quad (Eq.14)$$

El valor recomendado para a es $^{15}/_{16}$ = 0.9375, dado que alcanza una ganancia de 32 dB en la frecuencia más alta de la señal sobre la más baja.

2) Segmentación

En el modelo LPC la generación de los diversos fonemas está a cargo del tracto vocal, el cual puede ser simulado por un filtro de respuesta infinita al impulso que cambia en intervalos de tiempo cercanos a los 30 ms, tiempo durante el cual se considera que la señal de voz es cuasi-estacionaria, esta propiedad puede ser aprovechada. El sistema implementado segmenta la señal adquirida en tramas de 32 ms (N = 512 datos) solapadas entres si por 16 ms (m = 256 datos), se obtienen 62.5 tramas/s. La Figura 6 muestra una de esas tramas.

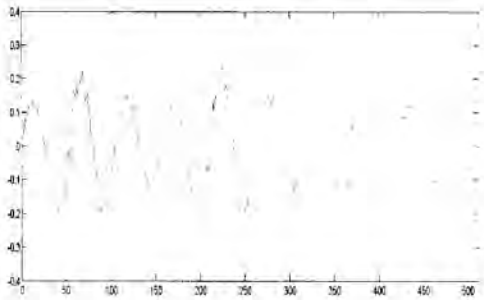

Figura 6. Segmento de señal de una expresión de la palabra "DERECHA"

3) Ventaneado en el tiempo

La segmentación de la señal genera discontinuidades en los extremos de cada trama, se aplica una "ventana" que amortigüe el fenómeno de Gibbs producido al truncar la señal (Oppenheim, Schafer, & Buck, 1999). Dada una ventana $w(m)w(m)$ de longitud N, el resultado de segmentar y ventanear el segmento de voz $s_n[m]s_n[m]$ es:

$$s_n[m] = \begin{cases} s[m] \cdot w[m], & 1 \le m \le N \\ 0, & 1 > m > N \end{cases}$$

(Eq.15)

La ventana más utilizada en procesamiento de voz es la ventana Hamming, la cual viene dada por la siguiente función:

$$w[n] = 0.54 + 0.46\cos\left(\frac{2\pi n}{N}\right), \qquad 1 \le n \le N$$

(Eq.16)

C. MODULO DE EXTRACCIÓN DE PARAMETROS

En el módulo de extracción de características se realizan tres pasos. (1) Un algoritmo detecta la voz en la señal adquirida, (2) una sumatoria de auto-correlación normaliza las frecuencias del espectro y (3) el algoritmo de Durbin-Levinson determina los coeficientes de predicción lineal $\hat{a}_k\hat{a}_k$.

1) Detector de señal

Para localizar los eventos de voz se calcula la potencia presente en cada trama mediante una suma de cuadrados *(Eq.17)*, luego se identifican las tramas que superen un determinado valor de potencia y por último se reorganizan las tramas detectadas en un nuevo vector.

$$P = \sum_{m=1}^{N} (s_n[m])^2$$

(Eq.17)

2) Algoritmo de Durbin-Levinson

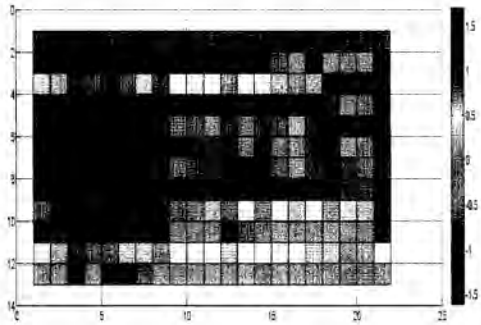

Figura 7. Representación gráfica de $\hat{a}_k\hat{a}_k$ para la palabra "DERECHA"

La complejidad de este algoritmo hace de éste el único que se implementó mediante una función prediseñada por MATLAB®, la función levinson recibe como argumento de entrada los vectores de auto-correlación y el valor de p, y entrega como salida los coeficientes $\hat{a}_k\hat{a}_k$. Este algoritmo genera una matriz de salida S x PS x P, como la matriz 31 x 12 que se muestra en la Figura 5, la cual caracteriza una expresión de 31 tramas de voz cada una representada por un conjunto de 12 predictores.

D. MODULO DE IDENTIFICACIÓN DE COMANDOS

El módulo de identificación de comandos contiene un grupo de algoritmos que buscan determinar a qué comando corresponde la señal de voz adquirida; para esto se realiza la técnica de medición de distancia entre patrones sujeta al alineamiento y normalización temporal de las secuencias a comparar.

1) Medición de la distancia entre dos patrones

Es necesario recordar que $d(\hat{a}_{xk}, \hat{a}_{yk})$ $d(\hat{a}_{xk}, \hat{a}_{yk})$ es la medida de la distancia entre dos instancias completas, y que cada una está formada por una serie de tramas de predictores $\hat{a}_k\hat{a}_k$. En este sistema la medida de la distancia entre dos tramas x_ix_i y y_iy_i, se define de la siguiente forma:

$$d(x_i, y_i) = \sum_{k=1}^{p} (x_i[k] - y_i[k])^2$$

(Eq.18)

2) Alineamiento y normalización temporal

El alineamiento temporal consiste en una correspondencia entre los patrones de voz $\hat{a}_{xk}\hat{a}_{yk}$ $\hat{a}_{xk}\hat{a}_{yk}$, esta relación se puede representar por medio de un plano cartesiano en el cual los ejes son i_xi_x y i_yi_y, y en cada punto de dicho plano se calcula el valor de la distancia $d(x_i, y_i)d(x_i, y_i)$.
La medición de la distancia entre dos patrones requiere de la generación de un camino entre los puntos $(1,1)(1,1)$y $(T_x,T_y)(T_x,T_y)$ en el cual se realice las *(Eq.10 y 18)*.

3) Restricciones de normalización

Para que la medida de la distancia sea subjetivamente significativa, los caminos generados deben cumplir el conjunto de restricciones expuestas. A continuación se explica cómo se implementó cada restricción:

- Restricciones de punto final

Se estableció que todo el camino se construya hasta alcanzar el punto $(T_x,T_y)(T_x,T_y)$.

- Restricciones de continuidad local

Dar un paso requiere que este cumpla una restricción de continuidad local, en este proyecto se determinó que la restricción TIPO III sería la determinante al generar aleatoriamente los distintos pasos.

TIPO III P1 ®(2, 1)(2, 1)
P2 ®(1, 1)(1, 1)
P3 ®(1, 2)(1, 2)

- Restricciones de camino global

Se evalúa cada paso si cumple con las restricciones de camino global que resultan como consecuencia de las restricciones de continuidad local, de ser así se continua al siguiente paso, de lo contrario se borra el paso para que se vuelva a dar.

4) Algoritmo genético

Se ha planteado la ejecución de un algoritmo genético cuya descripción general se ha expuesto y los detalles de su implementación se exponen a continuación. La figura 6 muestra el diagrama de flujo del algoritmo genético.

El algoritmo genético solo tiene dos parámetros iniciales, que se han determinado de forma heurística $ps = 20\,individuos$

$ps = 20\,individuos$ y $g = 25\,generac.$ $g = 25\,generac.$. Los siguientes son los pasos que sigue el algoritmo genético:

- Generar una población inicial de caminos para el plano relativo a los patrones $\hat{a}_{xk}\hat{a}_{yk}\hat{a}_{xk}\hat{a}_{yk}$.
- Determinar la condición de fin al algoritmo.
- Evaluar la generación actual $(d(x_i, y_i)d(x_i, y_i))$.
- Se asignan pesos en una "ruleta" que son proporcionales al valor de evaluación. Se cruzan individuos según los pesos, y cada uno de los nuevos individuos se somete a una "poco probable" mutación.
- Generar una nueva población de caminos.
- Al terminar las generaciones, elegir el mejor.

5) La interfaz de usuario

MATLAB® ofrece la posibilidad de crear interfaces graficas de usuario (GUI) que permiten interactuar con el desarrollo de los programas sin la necesidad de escribir códigos.
Se diseñó una interfaz que pudiera realizar las siguientes tareas:

- Iniciar y finalizar los sistemas de navegación e identificación de comandos.
- Adquirir los patrones relativos a cada comando del sistema de navegación.
- Adquirir e identificar en línea los comandos durante la navegación.

La figura 16 muestra la interfaz implementada, en esta el botón Navegar (1) inicia y detiene el sistema de navegación, el botón Adquirir objetivo (2) activa por 1 segundo el sistema de adquisición, identifica el comando dicho y envía al robot la orden correspondiente, cada botón del grupo (3) activa por 1 segundo el sistema de adquisición y almacena el patrón resultante como uno de los comandos (ADE = frente, IZQ = izquierda, DER = derecha, REV = reversa), en el espacio de la parte inferior (4) se muestra el resultado de la identificación.

Figura 8. Interfaz para Adquisición e Identificación de Comandos

RESULTADOS

Es fundamental realizar un conjunto de pruebas experimentales que permitan valorar el desempeño del sistema de reconocimiento de comandos de voz, este conjunto de pruebas se diseña teniendo en cuenta el objetivo del sistema, las condiciones normales de operación y la dependencia al locutor.

En primer lugar, dado que el objetivo del sistema es identificar palabras aisladas sin analizar significados o coherencias, su desempeño se valora al ordenarle que identifique un comando de voz adquirido, repitiendo la prueba la cantidad de veces que sea necesaria hasta establecer una aproximación estadística a la probabilidad de acierto del sistema.

En segundo lugar, una característica del sistema que resulta muy relevante a la hora de diseñar la prueba radica en el hecho que éste se adapta al locutor, de modo que cada hablante deberá "entrenar" al sistema con sus propios comandos. La prueba debe aplicarse a un grupo de hablantes con voces lo más disimiles posible.

Por último, es necesario establecer las condiciones de ruido en las que operaría el sistema para poder definir las condiciones en que se probará; se definió que el sistema debe ser probado en tres ambientes: exterior-campo, exterior-urbano e interior-hogar. El ambiente exterior-campo presenta fuentes de ruido generalmente animales con niveles hasta de 75 dBA, el ambiente exterior-urbano presenta fuentes generalmente de tráfico automotor y con niveles hasta de 100 dBA, y el ambiente interior-hogar presenta niveles de ruido hasta de 80 dBA producidos por lo general en electrodomésticos.

NOTA: A pesar de los diversos entornos que se puedan analizar, es necesario que el entrenamiento del sistema (el momento en el que se adquieren los comandos de voz) se haga en condiciones de silencio.

Teniendo en cuenta las anteriores condiciones, el sistema se probó en un conjunto de 4 personas (un hombre mayor, una mujer adulta, una niña y un niño) que posteriormente al entrenamiento tomaron 20 muestras de cada palabra en cada uno de los entornos descritos. Una vez realizada la prueba, el número de aciertos se registró en la Tabla 1.

En la siguiente tabla, ENT = entorno, COM = comando, UR = urbano, CA = campo, HG = hogar, y los comandos se encuentran abreviados de igual forma que en la interfaz.

LOCUTOR	ADULTO			ADULTA		
ENT. / COM.	UR	CA	HG	UR	CA	HG
ADE	17	17	17	17	18	17
DER	15	17	17	16	16	18
DET	19	19	19	18	19	20
IZQ	19	20	20	19	20	19
DET	16	17	17	17	17	17

LOCUTOR	NIÑO			NIÑA		
ENT. / COM.	UR	CA	HG	UR	CA	HG
ADE	17	18	17	16	18	17
DER	16	17	17	16	17	16
DET	19	18	19	18	18	18
IZQ	19	20	19	20	19	20
DET	16	17	17	16	16	17

Tabla 1. Número de aciertos del sistema
Teniendo en cuenta Ruido y Locutor

Los resultados obtenidos se representan en la Figura 9, se puede apreciar el buen desempeño del sistema, y cómo este presenta una mejor respuesta cuando se ejecuta en un ambiente con poco ruido.

No existe una diferencia significativa entre los distintos casos que pueda presentar la prueba, el sistema presenta resultados similares en adultos hombres, adultos mujeres y niños hombres (89%); así como en un entorno exterior-campo o en el interior de un hogar (90%).

Los resultados indican que el sistema tiene una efectividad promedio del 89%, siendo un poco más baja en el ambiente exterior-urbano. Una característica notable del sistema que se aprecia en los resultados es su poca dispersión en los resultados, expresada en una baja varianza (s^2 = 0,00018) y una desviación estándar de 1,33 aciertos. Los autores consideran que estos resultados se deben a dos condiciones, la primera, que dichos errores son inherentes al manejo por parte del usuario, y la segunda, y más importante, que tan baja tasa de error se deben más a los explícitos condicionamientos que se imponen al sistema que a las mismas características de éste.

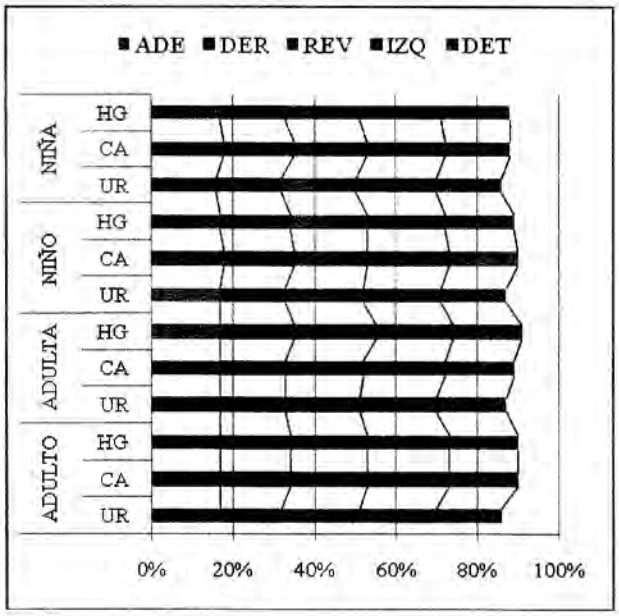

Figura 9. Porcentaje acumulado de aciertos teniendo en cuenta Ruido y Locutor

CONCLUSIONES

El proceso de diseño, implementación y prueba del sistema de identificación de comandos de voz arroja las siguientes conclusiones:

- No es necesario implementar una gran cantidad de algoritmos o algoritmos muy complejos para dar solución al problema de la identificación de palabras. Es suficiente con delimitar muy claramente las condiciones de operación que este tendrá y así, escoger la solución más eficiente.

- Dada la extensa bibliografía sobre el uso de la Transformada Rápida de Fourier en procesamiento de señales, ésta se constituyó en una primera opción para la extracción de características, ya que es reconocida su capacidad de caracterizar una señal en muy pocos parámetros, sin embargo, la FFT es propia del análisis espectral que se realiza en la aproximación acústico-fonética al reconocimiento de voz. En este documento se expone sobre la búsqueda de soluciones desde el análisis estadístico de la aproximación de comparación de patrones.

- Si se cuenta con un entorno de desarrollo lo suficientemente versátil (como MATLAB), es posible implementar aplicaciones muy potentes con diseños muy simples.

- La solución al problema de la generación de caminos se afrontó desde dos soluciones posibles, una solución recursiva y una solución estocástica. DTW es un sencillo algoritmo recursivo que genera caminos en el plano de forma completa y optima, pero en esta clase de desarrollos las prioridades son la complejidad (o costo) y la innovación, así que se escogió el Algoritmo Genético que aunque puede no dar soluciones completas y optimas, si puede acercarse a estas con muy pocos recursos de tiempo y memoria.

- La utilización del Algoritmo Genético es asociada comúnmente al resultado de la inspiración en la naturaleza para la creación de algo nuevo, sin embargo, en esta presentación se observa como este algoritmo para la búsqueda de soluciones corresponde más a la combinación de métodos estadísticos con algoritmos de inteligencia artificial, que a la simple copia de las formas naturales.

- El uso de Redes Neuronales Artificiales dominó muchas de las horas de prueba y pre-diseño y aunque el aprendizaje en esta disciplina de la inteligencia computacional generó un gran avance, no fue posible implementar una Red Neuronal que generara una respuesta al problema de identificación en el corto tiempo que se requería. El uso de redes neuronales artificiales resulta de mayor utilidad en un dispositivo de procesamiento paralelo que optimice sus potencialidades. Esta tarea se divisa como una línea de investigación a futuro.

- Algunos aspectos que surgen como interrogantes, y que merecerían una investigación posterior son:

¿Cuál es el efecto que se presenta en la eficiencia y eficacia del algoritmo al aumentar la cantidad de comandos de voz que se quieran identificar? ¿Cuál es la capacidad que tiene el algoritmo de diferenciar comandos de voz muy similares?

AGRADECIMIENTOS

Los autores agradecen el soporte ofrecido por el instituto de investigaciones de la Orinoquia colombiana que apoyó financieramente el desarrollo del proyecto de investigación que generó el resultado presentado en este documento. Se presenta un agradecimiento especial a los Ingenieros Zulia Alezones así como Yeison Baquero por la colaboración y asesoría en el tema de la investigación. También se agradece al personal que integra el grupo de investigación en robótica (*giro*).

REFERENCIAS

Alezones, Z,. Baquero Y,. Borrero H. (2009). *Reconocimiento de Palabras Aisladas Utilizando LPC Y DTW, para control de navegación de un mini-robot.* Colombia 2009,II Congreso Internacional Ingeniería Mecatrónica – UNAB. ISSN: 2145-812X

Chou, W., & Juang, B. H. (2003). *Pattern recognition in speech and language proccessing*. New York: CRC Press.

Oppenheim, A., Schafer, R., & Buck, J. (1999). *Discrete-Time Signal Processing* (2da ed.). New Jersey: Prentice Hall.

Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. New Jersey: Prentice-Hall, Inc.

Russell, S., & Norvig, P. (2010). *Artificial Intelligence, a Modern Approach* (Third ed.). New Jersey: Pearson Education, Inc.