

Reconocedor de Voz para Silla de Ruedas Basado en HMM e Implementado en DSP

Esteban Angarita León, Carolina Vargas Sánchez
stenba@gmail.com, aeda.cvs@gmail.com
Universidad Autónoma de Bucaramanga

Resumen

El reconocedor de voz desarrollado pretende identificar cinco (5) comandos diferentes, dichos por un hablante que haya realizado la fase de entrenamiento del sistema previamente.

En esa fase el usuario debe pronunciar varias repeticiones de cada uno de los comandos de voz para que el algoritmo de entrenamiento cree los patrones de comparación.

La voz se adquiere a través de un micrófono omnidireccional y la señal se limita en frecuencia con un filtro pasabanda antialiasing.

El procesamiento digital de la señal obtenida se realiza con una tarjeta DSP56F805EVM de Motorola. Primero se realiza la conversión A/D a 12 bits de resolución.

Las muestras resultantes se procesan con un filtro digital tipo IIR que permite el paso del espectro entre 0.3 y 3.4 KHz correspondiente al intervalo de interés.

Es aquí donde cobra importancia la programación del chip DSP, permitiendo el procesamiento de la señal en tiempo real, característica necesaria para la ejecución del algoritmo de reconocimiento de voz basado en los modelos ocultos de Markov (MOM o HMM).

A la señal que resulta después del filtro digital, se le calcula el contorno de energía, luego de un proceso de segmentación y inventanado.

En el algoritmo, ese proceso se aprovecha para ir determinando al mismo tiempo el vector de observaciones O , según la teoría de las MOM.

El algoritmo de reconocimiento de voz incluye los algoritmos forward, backward y de Baum-Welch, cuyos cálculos permiten obtener y comparar probabilidades. Las comparaciones, según el propósito sea de entrenamiento o de reconocimiento, servirán respectivamente para optimizar los modelos de cada comando o para identificar la acción que el sistema debe realizar de

acuerdo al comando introducido: adelante, atrás, izquierda, derecha, parar.

Las acciones se llevan a cabo por medio de los puertos PWM y GPIO del chip DSP. Las señales son enviadas a una tarjeta de potencia VNH2SP30-E para cada uno de los motores (principal y de dirección). Estas tarjetas reciben las señales PWM y de polaridad para el manejo de la velocidad y el sentido de giro de los motores.

1. Introducción

El reconocimiento de voz tiene un campo de aplicaciones bastante amplio y sigue creciendo con el paso del tiempo, pues cada día aumenta la demanda de dispositivos que revolucionen las aplicaciones actuales y que hagan más comfortable la vida del consumidor.

Las sillas de ruedas permiten a las personas con ciertas limitaciones físicas y/o neurológicas, movilizarse de una manera “sencilla” y, en la mayoría de los casos, independiente. Sin embargo, no en todos los casos una silla de ruedas convencional es la mejor opción, pues algunas de estas limitaciones obligan la intervención de otra persona que mueva la silla.

Por ejemplo, una persona con cuadriplejia no podría caminar ni tampoco moverse por sí sola en una silla de ruedas convencional, ni siquiera en una con mando por joystick.

El hecho de no poder caminar es una limitación que quizá no pueda revertirse o modificarse y con la cual muchos pacientes aprenden a vivir, sin embargo, una imposibilidad adicional es el no poder decidir a dónde ir, la manera de hacerlo y cuándo hacerlo, una silla de ruedas convencional impide tener esta autonomía a las personas que no pueden impulsarla por sí solas, es decir, se adquiere una limitación más.

Este proyecto ayuda a eliminar ese impedimento, habilitando nuevamente la autonomía de la persona

con discapacidad, posibilitando el uso de su voz como herramienta para desplazarse de forma autónoma, con miras al mejoramiento de su calidad de vida.

La estructura planteada para el desarrollo de este proyecto exige la integración de conocimientos en diversas áreas y técnicas de control de sistemas mecánicos y electrónicos, como el manejo de un controlador digital, el concepto de modulación de ancho de pulso para el control de motores y la implementación de un algoritmo capaz de reconocer patrones aislados de voz.

En el mercado se pueden adquirir sillas de ruedas electrónicas a precios sumamente elevados, lo cual es un impedimento para algunas personas discapacitadas. Por ello, se pretende brindar la oportunidad a una mayor cantidad de personas de acceder a este tipo de “comodidad/necesidad”, reduciendo, no sólo los costos de fabricación de una silla de ruedas electrónica, sino también haciendo posible su adquisición en el mercado local. Ya que la falta de oferta nacional, es también un impedimento para los pacientes que solicitan este tipo de productos.

2. Vocabulario del Reconocedor

El vocabulario seleccionado para este reconocedor consta de 5 comandos que representan las operaciones básicas que deberá realizar la silla de ruedas: adelante, atrás, izquierda, derecha y parar.

Esta selección del vocabulario se hizo principalmente obedeciendo a la naturaleza de cada palabra, a su significado, y además atendiendo a la sencillez y comodidad para memorizar los comandos de control de la silla. Por otro lado, la decisión se complementa con el análisis espectral de esas cinco palabras que deben ser lo suficientemente diferentes en sus características, de tal forma que el algoritmo sea capaz de reconocerlas individualmente sin confundirse entre una y otra.

A continuación se muestra la forma de onda de las palabras ‘Adelante’ y ‘Atrás’, con los correspondientes fonemas indicando el inicio de cada alófono en la señal de voz. Se muestra además el contorno de energía de las mismas y con base a este, se puede determinar que los 5 comandos son lo suficientemente diferentes.

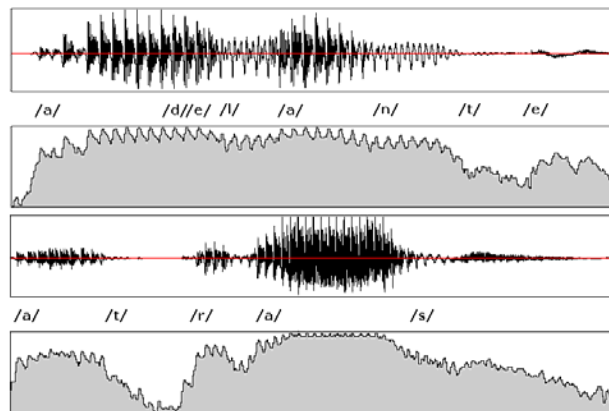


Figura 1. Formas de onda y contornos de energía de las palabras ‘Adelante’ y ‘Atrás’.

3. Filtro activo

3.1. Aliasing y Conversor A/D

El efecto alias (*aliasing*) es un solapamiento de frecuencias que ocurre cuando el muestreo no es lo suficientemente rápido, esto lo explica el criterio de Nyquist, el cual indica que una señal continua en el tiempo debe ser muestreada más de dos veces por período si va a ser convertida a una señal discreta en el tiempo.

Antes de hacer una conversión A/D se debe conocer la frecuencia más alta de la señal de interés para poder determinar la frecuencia de muestreo, que debe ser mayor que el doble de esa frecuencia máxima.

En el dominio digital es imposible eliminar los errores generados por el muestreo, por lo tanto para evitar el efecto alias debe filtrarse la señal en el dominio analógico (es decir antes de la conversión). Por ello, se limita la señal de voz para el reconocedor en una banda de 300Hz a 3400Hz. De esta forma la frecuencia de muestreo deberá ser mayor a 6800 muestras por segundo.

Para este caso, un muestreo con frecuencia de muy superior a 6.8kHz sería catalogado como sobre-muestreo, lo cual significa mayor fidelidad a la señal original. Sin embargo el sobre-muestreo implica también un conversor más rápido y más costoso. Es necesario, antes de determinar la frecuencia de muestreo, tener en cuenta la velocidad máxima del ADC disponible. Además debe tenerse en cuenta el espacio en memoria necesario según la operación que se requiera realizar a las muestras.

El DSP56F805 de Motorola, permite hasta 1.66 millones de muestras por segundo –utilizando

conversión simultánea –. Una conversión simple le toma 8.5 ciclos de reloj que equivalen a $1.7\mu\text{s}$, es decir 588235 muestras por segundo. Esto es debido a la frecuencia máxima de reloj del ADC de 5MHz con período de 200ns.

Para poder cumplir con el teorema de muestreo y evitar el fenómeno de alias, se diseñó un filtro pasa-banda de acuerdo con la naturaleza de la señal de voz. Se determinó que el rango de frecuencias de interés en la señal de voz está entre los 300Hz y los 3400Hz, por tanto se eligió como frecuencia de muestreo 8000 muestras por segundo, lo que significa que se necesita una frecuencia de corte para la etapa pasa-bajas menor a 4.0kHz para este filtro.

3.2. Diseño e implementación del filtro activo

Para seleccionar las características del filtro se tuvo en cuenta, además de la banda de interés de la señal, la resolución del conversor A/D del DSP. Se debe cuidar que las componentes dentro de la banda de paso del filtro sean detectadas por el conversor.

La resolución del conversor permite saber cuánto deberá ser la atenuación en la banda de rechazo para que las frecuencias no deseadas no vayan a ser detectadas por el conversor y no se genere el fenómeno alias.

La resolución del conversor A/D del DSP56F805 es de 12bits, lo que quiere decir que una atenuación de -73dB en la señal hará que sea muestreada como un cero, sin embargo, el número efectivo de bits, ENOB, de este conversor está entre 9 y 10 bits, lo que para un FS^1 de 6.6V equivale a 12.91mV y 6.45mV respectivamente o -54dB y -60dB.

Por otro lado, el fabricante anota que para un desempeño óptimo del conversor se debe mantener un valor mínimo en la entrada que sea mayor o igual a 25mV, ya que valores por debajo podrían convertirse a un cero digital. Si la entrada del conversor está configurada como diferencial ($FS=6.6V$), estos 25mV equivalen a -49dB. Según lo anterior se determinó diseñar el filtro pasa-banda con atenuación de -60dB en las frecuencias de corte.

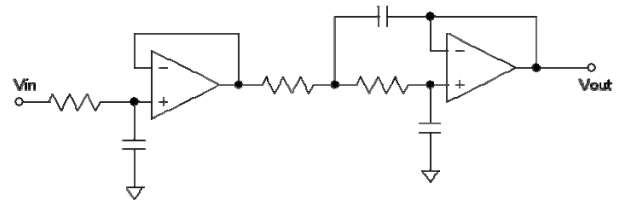


Figura 2. Esquema general de una etapa pasa-bajas SAB positivo de orden 3.

En la figura anterior se muestra esquemáticamente la conformación de una etapa pasa-bajas de implementación SAB² positivo, la cual necesita menos componentes que otras formas de implementación activa.

En el filtro pasa-banda se utilizaron 4 etapas similares y una sola etapa pasa-altas, también de implementación SAB positivo cuya configuración es igual a la de la Figura 2 cambiando los resistores por condensadores y viceversa.

4. Filtro Digital

La finalidad del filtro digital que se implementa en este proyecto es limitar la banda de frecuencias de la señal de voz desde 300Hz hasta 3400Hz como se realizó con el filtro análogo anteriormente, pero de manera más selectiva. Se diseñó entonces un filtro IIR por medio de la herramienta FDATool de Matlab con las siguientes características:

- Filtro tipo IIR elíptico.
- Orden del filtro: 28.
- Diseño en una sola sección.
- Frecuencia de muestreo de 8kHz (es la frecuencia de muestreo de la señal de entrada al filtro).
- Filtro pasa-banda.
- Banda de paso desde 300Hz hasta 3.4kHz
- Bandas de transición de 50Hz cada una.
- 80dB de atenuación las bandas de rechazo.
- 0.001dB de rizado máximo permitido en la banda de paso.

La herramienta genera unos vectores de coeficientes que son Num[28] para el numerador y Den[28] para el denominador, son presentados en formato hexadecimal y pueden ser encontrados en el documento de este trabajo.

¹ FS es la sigla para el inglés *Input Full Scale*, que quiere decir alcance, fondo de escala o también llamado *span*.

² Sigla para el inglés *Positive Gain Single Amplifier Biquads*.

Tales vectores son el resultado del proceso de diseño del filtro realizado por la herramienta. Los valores de los coeficientes son la base de la implementación del filtro dentro del programa de reconocedor de voz.

El proceso de filtrado digital de una señal obedece a la solución de la función de transferencia del filtro seleccionado, la cual para un filtro IIR se denota así:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}},$$

donde z^{-i} es un operador de retardo de i posiciones, cada a_i y b_i son los coeficientes del filtro, mientras que $(M + 1)$ y $(N + 1)$ son el número total de coeficientes en el numerador y en el denominador, respectivamente, y el mayor entre M y N indica el orden del filtro.

Generalmente en los diseños de filtros IIR, los valores de N y M son iguales, como sucede en el filtro diseñado en Matlab para este proyecto, donde el orden del filtro es 28 y hay 29 coeficientes para el numerador y 29 para el denominador.

Ya que la función de transferencia de un sistema es la razón de la señal de salida entre la señal de entrada, la solución a esta función de transferencia es una ecuación en diferencias de la siguiente forma:

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) - \dots - a_N y(n-N) + b_0 x(n) + b_1 x(n-1) + \dots + b_M x(n-M)$$

El filtro digital tipo IIR es un filtro recursivo ya que utiliza no sólo valores pasados de la señal de entrada (filtros no recursivos, por ejemplo filtros FIR), sino también valores pasados de la salida, como se ve en la ecuación anterior. Esto implica que a medida que aumenta el orden del filtro también aumenta la cantidad de operaciones que se deben realizar por cada muestra de una señal y la cantidad de espacio necesario en memoria para almacenar los datos anteriores.

Sin embargo, al comparar un filtro recursivo (IIR) con uno no recursivo (FIR) que ejerzan las mismas características de filtrado sobre una señal de entrada, el filtro no recursivo requiere ser de orden muy superior, siendo entonces mayor el número de

operaciones necesarias para implementar un filtro no recursivo. Para este caso, por ejemplo, un filtro FIR de Matlab con las mismas condiciones de diseño tiene orden mínimo de 764 (FIR *equiripple*).

5. Extracción de Características

5.1. Segmentación y Inventanado

La señal de voz no es periódica ni estacionaria durante la producción de cada palabra, por lo tanto su análisis debe ser especial y corresponderse con estas propiedades. Cada palabra se compone de distintos sonidos que duran muy poco y que se fusionan con los sonidos aledaños en la misma palabra o la frase que se está diciendo, esto hace que el tiempo en que la señal correspondiente a un alófono es estacionaria se acorte aún más (entre los 10ms y 40ms dependiendo del alófono, de su contexto y del hablante).

Por tanto el análisis de la voz debe hacerse limitado a tramas consecutivas de la señal, la longitud de cada trama dependerá de la cantidad mínima de tiempo establecida como la duración de segmentos estacionarios.

Para tomar un segmento de una señal dada, ésta debe multiplicarse por una función cuyo valor fuera del intervalo que delimita el segmento es nulo o cero, como lo es una señal cuadrada, por ejemplo.

El proceso de inventanado conlleva efectos tanto en el dominio temporal como en el dominio de la frecuencia y, tanto su magnitud como sus ventajas, dependen del tipo de ventana que se utilice. En el tiempo ocurre el efecto de ponderación de las muestras dentro del intervalo por los coeficientes de la función de la ventana elegida, mientras que, como efecto semejante en el dominio de la frecuencia, ocurre la convolución del espectro de la señal con la transformada de Fourier de la función de la ventana.

Para disminuir ese efecto convolutivo es necesario seleccionar una ventana con lóbulo principal angosto, ya que si es muy ancho se provocará suavizado espectral, y con lóbulos secundarios pequeños (en magnitud, dB), de lo contrario se obtendría un espectro lobulado.

Después de una serie de comparaciones entre las ventanas rectangular, Hann, Hamming, Kaiser, Blackman, Bartlett-Hanning y Tukey, todas con longitud de 40 muestras, se seleccionó la ventana tipo Hann, definida de la siguiente forma:

$$W_{HANN}(n) = \begin{cases} 0, & n < m_1 \\ 0.5 \left(1 - \cos \left(2\pi \frac{n}{m_2 - m_1 + 1} \right) \right), & m_2 \leq n \leq m_1 \\ 0, & n > m_2 \end{cases}$$

Donde $(m_2 - m_1 + 1)$ es la longitud de la ventana.

Ya que cada palabra comprenderá 12080 muestras, según las muestras destinadas para cada ventana y las muestras de solapamiento entre ventanas, se obtendrán 149 ventanas por palabra.

5.2. Cálculo del contorno de energía de la señal

La etapa de extracción de características de la señal permite codificarla, crear un patrón que identifique la señal y reemplace las muestras, con el fin de hacer más sencilla la comparación entre varias señales y reducir el espacio en memoria que ocupa la representación de la señal.

En señales de voz con miras al reconocimiento, son varias las características que se pueden extraer, entre los más nombrados están el cálculo de energía, potencia, densidad de cruces por cero, pitch y coeficientes cepstrales.

Para este trabajo se eligieron la energía de la señal y la densidad de cruces por cero para caracterizar la señal de voz, estos valores serán calculados para cada ventana y luego normalizados según un libro de códigos.

La energía de una señal en función del tiempo es una propiedad estadística de la señal, es un valor promedio, se utiliza en señales estacionarias o segmentos estacionarios de señales para caracterizarlas. Su cálculo se resume con la siguiente fórmula:

$$E_s(v) = \sum_{n=m_1}^{m_2+1} s^2(n) \cdot w^2(n - m_1)$$

Donde v es el número de la ventana que está entre 1 y 98 para el caso de este trabajo, m_1 y m_2 son los límites de la ventana como se dijo anteriormente, de tal forma que la longitud de la ventana es $(m_2 - m_1 + 1)$ muestras y $w(n)$ es la función ventana.

Ya que el reconocedor de voz está desarrollado sobre un procesador DSP Motorola, los cruces por cero de un intervalo se pueden calcular fácilmente

con la ayuda de un evento generado por el mismo *bean* del conversor A/D. Para el desarrollo del algoritmo, las ventanas se dividen en varios grupos, dependiendo de la longitud de las ventanas y de la porción de ventana en solapamiento, para el caso de este trabajo con ventanas de 30ms y solapamiento de 20ms, se necesitan tres grupos de ventanas diferentes para poder realizar los cálculos.

A manera de ejemplo se pueden crear dos grupos de ventanas, pares e impares, ya que las ventanas consecutivas estarán solapadas, es decir que la ventana $n + 1$ inicia antes que se haya alcanzado el final (longitud) de la ventana n , sin embargo, la ventana $n + 2$ inicia cuando ya se ha superado el final de la ventana n , ambas impares (para $n = 0$), lo mismo ocurrirá entre las ventanas $n + 3$ y $n + 1$. Esto se puede observar en la Figura 3.

Ventanas solapadas, sobre la señal de voz:

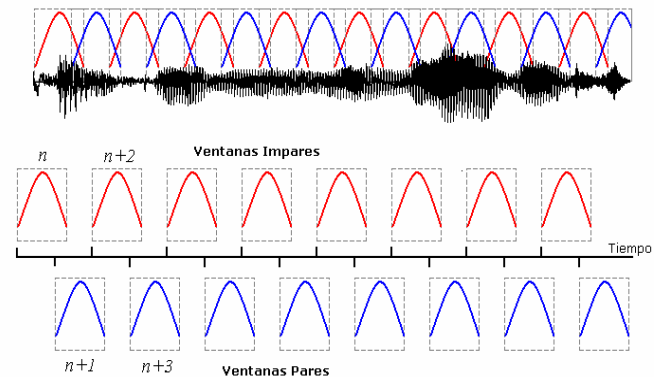


Figura 3. Ejemplo de ventanas solapadas, separadas en pares e impares.

Así, sólo se necesitarían dos contadores de cruces por cero, uno para ventanas pares y otro para ventanas impares, que irán aumentando cada vez que ocurra una interrupción por cruce por cero y se reiniciarán cuando se haya completado cada ventana.

Debe tenerse en cuenta que la figura es sólo un ejemplo para poder explicar mejor el solapamiento de las ventanas, sin embargo, se necesita mayor porción de ventana solapada para poder caracterizar toda la señal, y por consiguiente, se necesitan por lo menos tres grupos de ventanas.

6. Aplicación de los Modelos Ocultos de Markov (HMM)

6.1. Generalidades

En el diseño del algoritmo se crearon las variables que definen los elementos básicos de un modelo. Para esto se seleccionó el número máximo de estados como seis ya que es el mayor número de fonemas diferentes en las palabras seleccionadas como comandos. A su vez, M tiene un valor de 128, que será la cantidad de códigos normalizados o símbolos de observaciones. Se crearon también un grupo de matrices A, B y Pi para cada palabra del vocabulario. Esas variables conformarán los parámetros del modelo generativo de cada palabra que se inicializan con ciertos valores predefinidos como base y luego, mediante el algoritmo de Baum–Welch en la fase de entrenamiento, son modificados.

Al realizar la extracción de características de la señal de voz y codificar los resultados se obtiene un patrón que representa la palabra que ha sido adquirida, este patrón está compuesto por vectores de observaciones.

La codificación consiste en cambiar el resultado del cálculo de la energía, a valores normalizados según un libro de códigos previamente establecido, que corresponde al vector $V = \{V_1, V_2, V_3, \dots, V_M\}$.

A partir de los vectores de observaciones creados al final de la codificación de las tres repeticiones, se generan los valores óptimos de los parámetros del modelo que representa la palabra que se está entrenando.

6.2. Elementos básicos de cada modelo

Los modelos preliminares para el algoritmo de Markov creados para las palabras del reconocedor, son iguales entre sí en todos los parámetros. Son modelos tipo izquierda–derecha con saltos hasta de un estado, cada uno con seis estados y distribuciones uniformes en las matrices A, B y π . En la figura 4 se encuentra la representación más sencilla del modelo, donde los círculos con etiquetas entre S1 y S6 representan cada uno de los estados, las líneas representan las posibles transiciones y la flecha de las líneas indica la dirección de la transición. Las inscripciones a_{ij} sobre las líneas, señalan la probabilidad de transición desde el estado S_i hacia el estado S_j , estos subíndices también indican la posición del valor de esa probabilidad en la matriz de

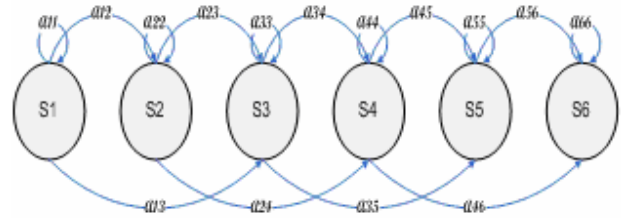


Figura 4. Representación de los estados y las transiciones.

transiciones entre estados A. En la matriz A de este modelo las probabilidades de transiciones desde un estado S_i hacia cualquier estado S_j están distribuidas en partes iguales, como se observa a continuación:

$$A = \begin{Bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{612} & a_{63} & a_{64} & a_{65} & a_{66} \end{Bmatrix} = \begin{Bmatrix} 0.\bar{3} & 0.\bar{3} & 0.\bar{3} & 0 & 0 & 0 \\ 0 & 0.\bar{3} & 0.\bar{3} & 0.\bar{3} & 0 & 0 \\ 0 & 0 & 0.\bar{3} & 0.\bar{3} & 0.\bar{3} & 0 \\ 0 & 0 & 0 & 0.\bar{3} & 0.\bar{3} & 0.\bar{3} \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{Bmatrix}$$

Debe notarse que la suma de los elementos de cada fila suman la unidad, lo cual es por regla estadística, ya que la totalidad de las probabilidades de que estando en un S_i pase a cualquier otro, deben sumar el 100%.

Por su parte, el vector de probabilidades de que un estado S_i sea el primero en la secuencia de estados observados, es decir el vector π , también se crea inicialmente con distribución uniforme, así cada estado del modelo tiene la misma probabilidad de ser el primero. Ya que son seis estados, cada uno tendrá probabilidad de 1/6 de ser el primero.

$$\pi \{ \pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_6 \} = \{ 0.1\bar{6}, 0.1\bar{6}, 0.1\bar{6}, 0.1\bar{6}, 0.1\bar{6}, 0.1\bar{6} \}$$

La matriz B, de probabilidades de observaciones por cada estado, es de dimensiones NxM=6x128.

Para esta matriz se crearon funciones de distribución diferentes para cada estado (fila de la matriz), teniendo en cuenta que $\sum_{k=1}^M b_j(k) = 1$. Entre

las cuales hay funciones de línea recta con pendiente negativa y con pendiente positiva y porciones de funciones senoidales.

La secuencia de observaciones se obtiene mediante comparación de los cálculos de energía localizada con el libro de códigos.

El libro de códigos es una distribución de segmentos de valores entre el mínimo y el máximo valor posible de energía de una ventana. Estos valores límites son 0 y 3850, este último se obtiene de la sumatoria de los 240 valores puntuales de energía de cada ventana (cuyo resultado conforma el valor de energía localizada o energía de ventana), tomando en cada uno el valor máximo posible de energía de una muestra que es el valor decimal que representa el voltaje máximo de entrada que es de +3.3V dividido en 10000 (esta división se hace con el fin de manejar números más pequeños en el programa) y finalmente elevado al cuadrado (la potencia al cuadrado es según la fórmula del cálculo de energía puntual $E_{muestra} = s(n)^2 * w(n)^2$), esto

es $\left(\frac{65535}{10000}\right)^2 = 42.94967$. Entonces la suma

$$\sum_{n=1}^{240} 42.94967 * w(n)^2 = 3849.3644 \text{ es el valor}$$

máximo de energía de una ventana.

La distribución de los segmentos de valores se realizó mediante una función de orden 2, donde el límite inferior del segmento es $0.235 * n^2$ y el límite superior es $0.235 * (n+1)^2$ y donde $0 \leq n \leq 127$.

Esto quiere decir que, calculado el valor de energía de una ventana, éste se compara con los segmentos de valores mediante una sentencia for y, cuando se encuentra el segmento correspondiente, en el vector de observaciones se guarda el valor de n como código o símbolo de observación. Es decir que el vector de observaciones es una lista de índices que apuntan a uno de los segmentos de valores.

7. Funcionamiento del Prototipo

El núcleo del prototipo se centra en el DSP56F805EVM, tarjeta que tiene como circuito

principal el Procesador DSP56F805. Este circuito es el que realiza todas las operaciones matemáticas y acumulaciones del algoritmo. La tarjeta contiene además memoria flash, conversor A/D, comunicaciones serial y en paralelo, pines PWM y otros dispositivos que podemos encontrar en su hoja de características.

El prototipo tiene una tarjeta de adquisición, que recibe la señal de voz, la filtra y la envía al conversor A/D. Este conversor envía la señal muestreada al DSP, que realiza una acción con base en esa señal. Esta tarjeta es alimentada por una batería de 5V y por una señal de -5V proveniente de un regulador que es alimentado con una batería de 9V.

La acción realizada por el DSP es enviada por los puertos de entrada y salida de propósito general y por dos de los pines PWM.

Estas acciones llegan a la tarjeta de potencia, cuyo circuito principal es el puente H VNH2SP30-E, que posee dos pines de manejo de polaridad, dos de bloqueo, dos de salida y un pin de PWM. Este puente H puede manejar hasta 40A según su hoja de datos. Las salidas de este circuito van a ser las señales de control de los motores principal y de dirección.

La tarjeta de potencia y el DSP son alimentados por una batería de 12V y 12Ah, suficiente para controlar el prototipo en su muestra.

8. Bibliografía

- [1] AMBARDAR, Ashok. Procesamiento de señales analógicas y digitales. 2 ed. México D.F. : Thomson Editores, 2000. 811p.
- [2] BERNAL BERMÚDEZ, Jesús, *et al.* Reconocimiento de voz y fonética acústica. México D.F. : Alfaomega y Ra-ma editorial, 2000. 332p.
- [3] CÁRDENAS MAESTRE, Laura Constanza y MONTEZUMA GÓMEZ, Libardo Andrés. Sistema de Registro de Señales Eléctricas para Apoyo Diagnóstico de Patologías Cardíacas Basado en Electrocardiografía y DSP. Tesis de Grado. Bucaramanga : UNAB. 2006. 153p.
- [4] GRUPO DE TRATAMIENTO Avanzado de Señal, Universidad de Cantabria. Análisis Localizado de la Señal en el Dominio del Tiempo. Diapositivas. Disponible en formato PDF.

- [5] MARI, Jean-Francois, *et al.* Automatic Word Recognition Based on Second-Order Hidden Markov Models. En IEEE Transactions on speech and audio processing, vol. 5, N° 1. (Enero, 1997); pág. 22.
- [6] MAXIM DALLAS SEMICONDUCTOR. The Basics of Anti-Aliasing: Using Switched-Capacitor Filters. Artículo (Application Note 928). Enero 11 de 2002. Disponible en línea en :
http://www.maximic.com/appnotes.cfm/appnote_number/928
- [7] MICROSTAR LABORATORIES INC. DAPL IIR Filter Module Manual. Documentación. Disponible en línea en :
<http://www.mstarlabs.com/docs/manuals/DAPLIFM.PDF>
- [8] MUÑOZ MONER, Antonio Faustino. Sensórica e instrumentación: Mecánica de alta precisión. 2000.
- [9] OROPEZA RODRÍGUEZ, José Luis. Algoritmos y Métodos Para el Reconocimiento de Voz en Español Mediante Sílabas. En Computación y Sistemas, vol.9, N° 3, pág. 270. Disponible en formato PDF.
- [10] RABINER, Lawrence R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. En Proceedings of the IEEE, vol.77, N° 2. (Febrero, 1989); pág. 257.
- [11] RAMÍREZ ROJAS, Juan Carlos. Control de Velocidad de los Motores de Corriente Continua de una Silla de Ruedas. Trabajo de Grado. Bucaramanga, UIS. 2003. 178h.
- [12] RÍOS MAESTRE, Antonio. Universidad Autónoma de Barcelona. Capítulo 4 Fonemas y Alófonos del Español. Estudios de Lingüística Española, revista universitaria electrónica, vol. 4, 1999. Disponible en línea en :
<http://elies.rediris.es/elies4/Cap4.htm>
- [13] SOLIMAN, Samir S. Señales y sistemas: continuos y discretos. Madrid : Prentice Hall, 1999. 542p.
- [14] THE DIGITAL SIGNAL processing handbook. Salem: CRC Press, 1998.