INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA

RECONOCIMIENTO DE COMANDOS DE VOZ UTILIZANDO LA HERRAMIENTA COMPUTACIONAL LABVIEW.

TESIS

QUE PARA OBTENER EL TITULO DE INGENIERO EN COMUNICACIONES Y ELECTRONICA.

PRESENTAN

CASTRO LÓPEZ FRANCISCO JAVIER.
MACEDO ZAMUDIO HIDELBERTO.

ASESORES:

DR. PABLO ROBERTO LIZANA PAULIN.
DR. SERGIO GARCÍA BERISTAÍN.

MÉXICO, D.F. NOVIEMBRE DE 2013

INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD PROFESIONAL "ADOLFO LÓPEZ MATEOS"

TEMA DE TESIS

QUE PARA OBTENER EL TITULO DE	**INGENIERO EN COMUNICACIONES Y ELECTRÓNICA**
POR LA OPCIÓN DE TITULACIÓN	**TESIS COLECTIVA Y EXAMEN ORAL INDIVIDUAL**
DEBERA (N) DESARROLLAR	**C. HIDELBERTO MACEDO ZAMUDIO** **C.FRANCISCO JAVIER CASTRO LOPEZ**

"RECONOCIMIENTO DE COMANDOS DE VOZ UTILIZANDO LA HERRAMIENTA COMPUTACIONAL LABVIEW"

DISEÑAR UN CIRCUITO VIRTUAL CAPAZ DE RECONOCER Y COMPARAR PATRONES DE COMANDOS DE VOZ, COMO SON ENCENDER, APAGAR, IZQUIERDA Y DERECHA; HACIENDO USO DE LA HERRAMIENTA COMPUTACIONAL LABVIEW.

- PRODUCCIÓN Y PERCEPCIÓN DE LA VOZ HUMANA
- FUNDAMENTOS PARA EL TRATAMIENTO DIGITAL DE LA SEÑAL DE VOZ
- ETAPAS DE RECONOCIMIENTO DE VOZ
- LA HERRAMIENTA COMPUTACIONAL LABVIEW
- DESARROLLO DEL PROYECTO

MÉXICO D.F. A 24 DE FEBRERO DE 2015

A S E S O R E S

ING. PABLO ROBERTO LIZANA PAULÍN

ING. SERGIO GARCÍA BERISTAÍN

ING. PATRICIA LORENA RAMIREZ RANGEL
JEFE DEL DEPARTAMENTO DE
INGENIERÍA EN COMUNICACIONES Y ELECTRONICA

Hoy pido sabiduría para construir un mañana mejor sobre los errores y la experiencia del ayer…

Tomas de Kempis

ÍNDICE

Objetivos generales.

Diseñar una herramienta virtual la cual permita la comunicación hombre-máquina.

Objetivos particulares.

Diseñar un circuito virtual capaz de reconocer y comparar patrones de comandos de voz, como son encender, apagar, izquierda y derecha; haciendo uso de la herramienta computacional LabVIEW.

Justificación.

A través del tiempo se han desarrollado interfaces que permiten la comunicación hombre-máquina, la alta demanda de comunicarse con distintos dispositivos exige una mejora día a día. La necesidad de emplear comandos de voz para controlar dispositivos o maquinaria, ya sea por falta de extremidades o por estas ser utilizadas en otras actividades son la motivación del presente proyecto. Ya que este circuito virtual se puede adaptar a distintos dispositivos no se hace un enfoque en uno solo, y así se deja abierta la aplicación para las necesidades y condiciones que se así lo requieran.

Introducción.

En el presente proyecto de tesis se propone, presenta y desarrolla los métodos y etapas utilizadas para el reconocimiento de patrones de voz, utilizando la herramienta computacional labVIEW. Su enfoque se basa en proporcionar soluciones a personas que carezcan de alguna extremidad, o que estas sean utilizadas para hacer otras labores.

La metodología utilizada está conformada por las siguientes etapas:

1.- Se realiza la adquisición de la señal de voz y la cuantificación y muestreo de la señal, se utiliza una frecuencia de muestreo de 22050, 16 bits y 1 canal.

2.- Se realiza una normalización a la señal de voz para evitar el error generado por el ruido de fondo.

3.- Se hace una segmentación de la señal de voz, se obtiene tramas con una duración de 20 ms y Se aplica un ventaneo a cada trama para evitar el error generado por los traslapes entre cada segmento.

4.- Se calculan los coeficientes cepstrales, para obtener características de la señal de voz esto con el fin de que sean comparadas.

5.- Se hace el cálculo de distancias entre la señal obtenida y la que se encuentra en la base de datos.

6.- Como la emisión de la señal de voz no se hace siempre a la misma velocidad se obtiene la distorsión dinámica temporal.

7.- Una vez reconocida la señal de voz se implementa un led indicador que señala la palabra dicha.

CAPITULO 1: PRODUCCIÓN Y PERCEPCIÓN DE LA VOZ HUMANA.

1.1 El sonido.

En el diccionario de la real academia española define el sonido como: sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido por un medio elástico como el aire [4].

1.2 La voz.

Es una onda de presión acústica que se genera voluntariamente a partir de movimientos de la estructura anatómica del sistema fonador humano. La producción de la voz comienza en el cerebro con la conceptualización de la idea que se desea transmitir, la cual se asocia a una estructura lingüística, seleccionando las palabras adecuadas y ordenándolas de acuerdo con unas reglas gramaticales.

La señal de voz genera una amplia gama de sonidos con la finalidad comunicativa, consiste en la creación de una onda de presión sonora que se propaga a través del aire a una velocidad de unos 340 m/s a temperatura de 15 °C. La concatenación del sonido y en un orden prefijado, característicos de cada idioma, constituyen el mensaje [4].

1.3 Producción de la voz humana. Fonética articulatoria.

El análisis de la lengua se realiza en tres niveles:

- Nivel fonológico. Se estudian las unidades lingüísticas mínimas: fonemas. El conjunto de los fonemas se establece por oposición, es decir, si se cambia un sonido de una palabra y cambia de significado se le considera fonema, por ejemplo en las palabras coco, loco y toco se cambia un fonema y el significado es distinto.

- Nivel morfosintáctico, se estudian las palabras estableciendo su género, número y tiempo, y la relación entre ellas.

- Nivel semántico. Se estudia el significado de las frases y su coherencia [4].

1.3.1 Fisiología y funcionalidad del aparato fonador.

El aparato fonador se divide en tres partes: Cavidades infraglóticas, Cavidad glótica y cavidades supraglóticas. Cada una de ellas realiza una misión distinta en la fonación, pero todas ellas son imprescindibles en la misma. En la figura 1.1 se presenta una descripción del aparato fonador.

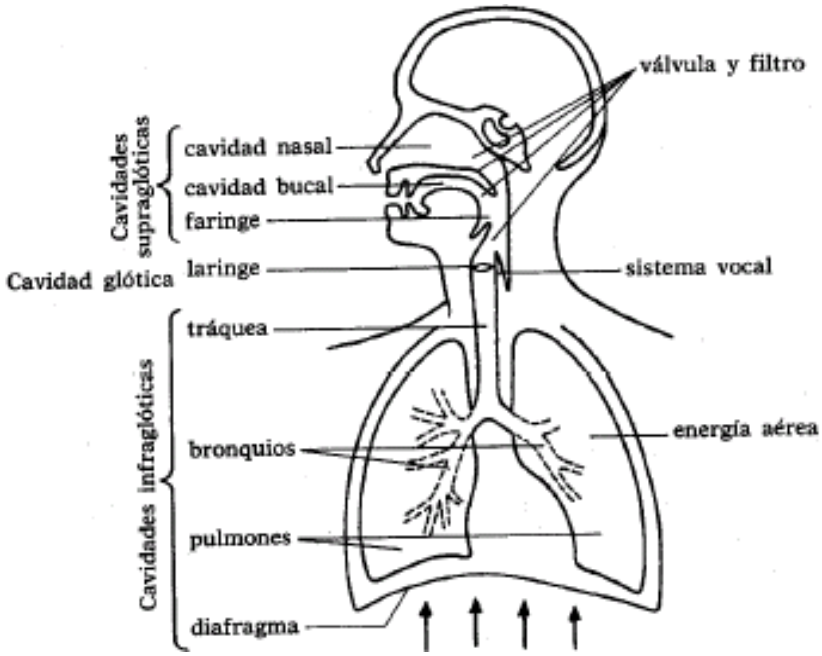


Figura 1.1 El aparato fonador.

1.3.2 Cavidades infraglóticas.

Tienen como misión proporcionar la corriente de aire espirada necesaria para producir el sonido. Está compuesta por diafragma, pulmones, bronquios y taquea.

El diafragma es un músculo situado por debajo de los pulmones y con forma de cúpula. Su misión es controlar el despliegue e hinchado de la cavidad pulmonar o su reducción y vaciado junto con los músculos pectorales, y con ello la respiración. Cuando se contrae el diafragma se ensancha la cavidad torácica producción de la inspiración del aire; al relajarse se reduce la cavidad torácica, produciendo la espiración del aire contenida en los pulmones.

1.3.3 Cavidad glótica.

Está formada por la laringe. La característica más interesante es la presencia en la misma de las cuerdas vocales, son las responsables de la producción de la vibración básica para la generación de la voz. Aunque se llaman tradicionalmente cuerdas vocales, en realidad se trata de 2 marcados pliegues musculosos. Cuando el aire sale de los pulmones pasa por a hendidura glótica (la glotis es el espacio triangular que queda entre las cuerdas vocales), haciéndolas vibrar. La vibración

producida que se emite puede variar en frecuencia e intensidad según varié la masa, longitud y tensión de las cuerdas vocales.

1.3.4 Cavidad supraglótica.

Existen cuatro cavidades: faríngea, nasal, bucal y labial.
La corriente de aire al pasar por la laringe, lo primero que se encuentra es con la faringe que es de donde arranca la raíz de la lengua. Aparece el primer obstáculo móvil: la úvula; es el apéndice final del paladar blando o velo del paladar. Cuando está unida a la pared faríngea, la corriente de aire sale exclusivamente por la boca, produciendo sonidos orales. Si el velo paladar esta caído, también se expulsara aire por la cavidad nasal. La cavidad nasal carece de elementos oviles, por lo cual su función es pasiva en la producción del habla.

La lengua es el órgano más móvil de la boca, registrando una actividad elevada durante el habla. Se divide en tres partes: raíz, dorso y ápice. El timbre del sonido será diferente según la forma sea cóncava, convexa o plana, o que se situé en la parte anterior, central o posterior. Los dientes son órganos pasivos en la medida en que están insertos en los maxilares; los inferiores son móviles por estar engarzados en la mandíbula inferior siendo esta activa en la circulación. El paladar es una amplia zona que va desde los alveolos hasta la úvula.

Finalmente tenemos los labios, elemento que posee bastante movilidad y, que por lo tanto, permite modificar sonidos.

Para la producción del habla se dan los siguientes elementos:

- Una fuente de energía, proporcionada por el aire a presión que se expulsa en la respiración.
- Un órgano vibratorio: las cuerdas vocales.
- Una caja de resonancia: las fosas nasales, la cavidad bucal y la faringe.
- Un sistema de articulación del sonido: lengua, labios, dientes y úvula

El proceso se inicia con la espiración del aire, al pasar a través de las cuerdas vocales las hace vibrar a una frecuencia determinada que depende de la tensión de las mismas; a esta frecuencia se le conoce como frecuencia fundamental; cuanto el tono es grave indica que la frecuencia es baja y cuando es agudo que la frecuencia es alta. Según como se encuentren articulados los órganos se formara una caja de resonancia distinta, la cual potenciara un conjunto de frecuencias y atenuara el resto. Aunque articulemos de forma similar los fonemas, aparecen características especiales de cada individuo, que es el timbre. Finalmente sale la voz este proceso explica el conjunto de fonemas sonoros. El resto de fonemas se producen por fricciones y explosiones de aire [4].

1.3.5 Clasificación de los fonemas

El punto de articulación identifica el lugar de las cavidades supraglóticas donde se produce la articulación del fonema.
En la figura 1.2 se muestran las partes que la componen.

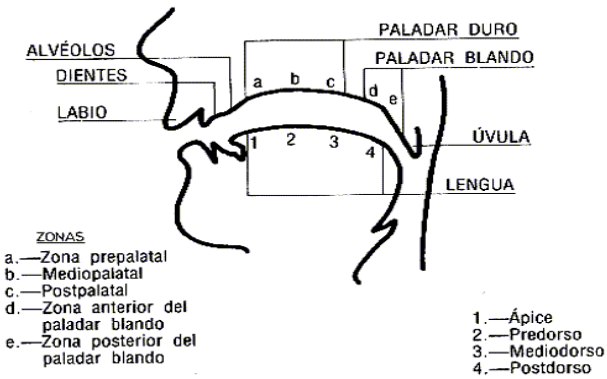


Figura 1.2 Cavidad bucal.

De acuerdo con los diferentes puntos de articulación se pueden distinguir los siguientes fonemas:
Consonantes:

- Bilabiales. Contactan los labios superiores e inferiores como se muestra en la siguiente figura: /p, b, m/. Ver figura 1.3.

Figura 1.3. Producción de las consonantes bilabiales.

- Labiodentales. Contactan el labio inferior con los incisivos superiores como se muestran en la siguiente figura: /f/. Ver figura 1.4.

Figura 1.4. Producción de las consonantes labiodentales.

- Linguodentales. Contacta el ápice de la lengua con los incisivos superiores: /t, d/. Ver figura 1.5.

Figura 1.5. Producción de las consonantes liguodentales.

- Linguoalveorales. Contacta el ápice o predorso de la lengua con los alveolos: /l, s, n, r/. Ver figura 1.6.

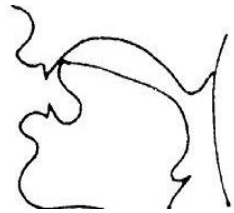


Figura 1.6. Producción de las consonantes Linguoalveorales.

- Linguovelares. Se aproxima o toca el postdorso de la lengua con el velo de paladar: /x, k, g/. Ver figura 1.7.

Figura 1.7 Producción de las consonantes Linguovelares

.

Vocales

- Anteriores. La lengua se aproxima a la región delantera o zona del paladar duro: /i, e/.
- Centrales. La lengua se encuentra en la parte central del paladar: /a/.
- Posteriores. La lengua se aproxima a la zona velar: /o, u/.

1.4 Percepción de la voz humana. El aparato auditivo.

El oído es un órgano cuya función es captar las ondas acústicas y transformarlas en impulsos nerviosos que el cerebro puede interpretar. Está formado por tres partes, ver figura 1.8.

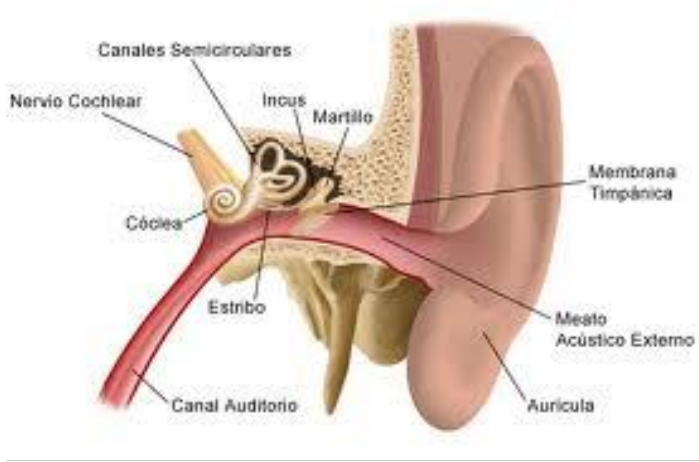


Figura 1.8. Partes del oído.

1. Oído externo

Está formado por el pabellón auditivo y por el conducto auditivo externo. El pabellón auditivo (conocido comúnmente como oreja) recoge las ondas sonoras y facilita su paso hacia el interior, realizando una amplificación de las mismas. El conducto auditivo externo acaba en el tímpano, una membrana que le separa del oído medio. El conducto auditivo externo conduce las ondas hacia la membrana y las amplifica, pero atenúa a los sonidos más agudos aquellos que podrían dañar la cóclea.

2. Oído medio

Se separa del oído externo a través del tímpano y se comunica con el oído interno a través de la ventana oval y la ventana redonda. Dispone de una cadena ósea de tres huesillos llamados martillo, yunque y estribo. También se encuentra la trompa de Eustaquio, un canal que se comunica con la faringe. La misión del oído medio es transmitir los sonidos desde el oído externo al interno realizando una adaptación de las impedancias acústicas. Cuando la intensidad es pequeña, la cadena ósea se mueve en conjunto produciendo un aumento de la misma; cuando la intensidad es grande se produce una disminución de la intensidad para evitar daños al oído interno.

3. **Oído interno**

   Está formado por el caracol y el órgano vestibular. El caracol es el órgano de audición y su función es percibir las frecuencias de las vibraciones sonoras; las convierte en impulsos nerviosos que transmite al cerebro para su interpretación.

   El órgano vestibular está formado por canales semicirculares que intervienen en el equilibrio del ser humano.

La percepción es la forma en que cada individuo siente los diferentes sonidos. Es una cuestión totalmente subjetiva.

El oído humano es capaz de percibir un rango de frecuencias que va desde los 20 Hz hasta los 20,000 Hz, aunque estos límites dependen de la persona en cuestión.
El conjunto de frecuencias y la sensación de diferencia entre las mismas no se perciben con la misma sensibilidad. Las frecuencias, altas y bajas se escuchan con menor intensidad, siendo la zona de 3,000 Hz la de mayor fuerza. La sensación de variación de frecuencias, en general, sigue una escala logarítmica; para notar la misma diferencia la misma diferencia entre varias frecuencias debemos ir duplicando su valor [4].

CAPÍTULO 2: FUNDAMENTOS PARA EL TRATAMIENTO DIGITAL DE LA SEÑAL DE VOZ.

2.1 Dominio en el tiempo.

Cuando se diseñan circuitos electrónicos de comunicaciones, se emplean para analizar y pronosticar con base a la distribución de potencias y frecuencias de la información, esto se hace con un método matemático llamado *análisis de señales.*

Un osciloscopio básico es un instrumento de dominio del tiempo. La pantalla del tubo de rayos catódicos es una representación de la amplitud de la señal de entrada en función del tiempo, y se le suele llamar *forma de onda de la señal.* En esencia, una forma de onda de la señal muestra la forma y la magnitud instantánea de la señal con respecto al tiempo, pero no necesariamente indica el valor de la frecuencia. Con un osciloscopio, la desviación vertical es proporcional a la amplitud de la señal total de entrada, y la deflexión horizontal es una función del tiempo (frecuencia de barrido) [1]

2.2 Dominio de la frecuencia.

El analizador de espectro es un instrumento en el dominio de la frecuencia. En esencia no se despliega ninguna forma de onda en la pantalla de tubo de rayos catódicos. En vez de lo anterior se muestra una gráfica de amplitud contra frecuencia (la cual se conoce como *espectro de frecuencia*). En un analizador de espectro, el eje horizontal representa la frecuencia y el eje vertical representa la amplitud. En consecuencia existirá una deflexión vertical para cada frecuencia que está presente en la entrada. De hecho la forma de onda de entrada se "barre" a una frecuencia variable, con la ayuda de un filtro de paso de banda con Q elevado cuya frecuencia central esta sincronizada con la velocidad de barrido horizontal del tubo de rayos catódicos. Cada frecuencia que está presente en la forma de onda entrada produce una línea vertical en la pantalla del tubo de rayos catódicos (estas son las *componentes espectrales*). La deflexión vertical (altura) de cada línea es proporcional a la amplitud de la frecuencia que representa. Una representación en el dominio de la frecuencia de la onda muestra el contenido de la frecuencia, pero no indica necesariamente la forma de onda o la amplitud combinada de todas las componentes de entrada en un instante específico de tiempo [1]

2.3 Transformada Fourier.

Para las formas de onda de tipo senoidal en el dominio del tiempo la frecuencia se puede encontrar mediante la ecuación 2.1:

$$f = \frac{1}{T} \tag{2.1}$$

donde:

f=frecuencia [Hz]

T=periodo[s]

Para formas de onda no senoidales utilizaremos la transformada de Fourier.

La transformada de Fourier (FT) por sus siglas en inglés, nos proporciona los componentes de tipo senoidal en $\omega(t)$.

Por lo tanto la transformada de Fourier (FT) de una forma de onda $\omega(t)$, se obtiene con la ecuación 2.2

$$W(f) = F[\omega(t)] = \int_{-\infty}^{\infty}[\omega(t)]e^{-j2\pi ft}\,dt \tag{2.2}$$

donde:

$F[\omega(t)]$= denota la transformada de $[\omega(t)]$.

f= es el parámetro de frecuencia [Hz].

Esto define al término frecuencia que es el parámetro f en la transformada de Fourier.

La FT se emplea para encontrar las frecuencias $\omega(t)$, se selecciona algún valor de f como por ejemplo $f = f_0$ y se calcula $|W(f_0)|$ en general al integral FT se evalua una y otra vez para todos los valores posibles de f sobre el rango $-\infty < f > \infty$ para encontrar todas las frecuencias en $\omega(t)$. La evaluación directa de la integral FT puede ser difícil, así que una lista de técnicas alternativas de evaluación bastante útil es:

1. Integral directa.
2. Tablas de transformadas de Fourier o transformadas de Laplace.
3. Teorema de la FT.
4. Superposición para dividir en 2 o más componentes simples.

5. Diferenciación o integración de $\omega(t)$.
6. Integración numérica de la integral FT en la pc por medio de funciones de integración en MATLAB.
7. Transformada rápida de Fourier (FFT) en la PC por medio de función FFT en MATLAB

A partir de la ecuación 2.2 y debido a que $e^{-j2\pi ft}$ es complejo, entonces $W(f)$ es una función compleja de frecuencia por lo tanto $W(f)$ puede ser compuesta por 2 funciones, $X(f), Y(f)$

Tal que:

$$W(f) = X(f) + jY(f) \tag{2.3}$$

Lo cual es idéntico a escribir un número complejo en termino de pares de números reales que puede graficarse en un sistema de coordenadas cartesianas, por esta razón se le conoce a esta ecuación como forma en cuadratura o forma cartesiana de la misma forma esta ecuación puede escribirse equivalentemente en términos de un sistema de coordenadas polares, en lugar de par de funciones reales; denotar magnitud y fase. Tal como se aprecia en la ecuación 2.4.

$$W(f) = |W(f)|e^{-j\theta f} \tag{2.4}$$

donde:

θ=Angulo [°]

Podemos obtener ambas componentes de la siguiente forma:

Para magnitud, mediante la ecuación 2.5.

$$|W(f)| = \sqrt{X(f)^2 + Y(f)^2} \tag{2.5}$$

Y para frecuencia, mediante la ecuación 2.6.

$$\theta f = \tan^{-1}\frac{Y(f)}{X(f)} \tag{2.6}$$

Esto se conoce como la forma de magnitud y fase en forma polar, para determinar la presencia de ciertos componentes de frecuencia se examina el espectro de magnitud $|W(f)|$ a lo cual en ingeniería se le conoce libremente como espectro [2]

2.4 **La serie de Fourier**.

Esta serie se usa en análisis de señales para representar las componentes senoidales de una onda periódica no senoidal. Es decir, para cambiar una señal en el dominio del tiempo a una señal en el dominio de la frecuencia. En general, se puede obtener una serie de Fourier para cada función periódica, en forma de una serie de funciones trigonométricas mediante la ecuación 2.7.

$$f(t) = A_0 + A_1 \cos \propto + A_2 \cos 2 \propto + A_3 \cos 3 \propto + \cdots + A_n \cos n \propto + B_1 \sin \beta + B_2 \sin 2\beta + B_3 \sin 3\beta + \cdots + B_n \sin n\beta \quad (2.7)$$

donde:

$\propto = \beta$

A= amplitud de la señal.

La ecuación 2.7 indica que la forma de onda *f (t)* comprende un valor promedio (A_0) de cd, una serie de funciones cosenoidales en las que cada termino sucesivo tiene una frecuencia que es múltiplo entero de la frecuencia del primer término cosenoidal de la serie, y una serie de funciones senoidales en la que cada termino sucesivo tiene una frecuencia que es múltiplo entero de la del primer término senoidal de la serie. No hay restricciones para los valores relativos de las amplitudes de los términos seno y coseno. La ecuación 2.7 se enuncia en palabras como sigue: cualquier forma de onda periódica está formada por un componente promedio y una serie de ondas senoidales y cosenoidales relacionadas.

2.5 Teorema de Muestreo:

Para que una señal sea discreta en tiempo se utilizan técnicas de muestreo, a pesar de que hay pérdida de mucha información al muestrear una señal, es necesario para así trabajar con valores finitos. El teorema de muestreo es uno de los de mayor utilidad porque se aplica a los sistemas digitales de comunicación, y es otra aplicación de una expansión de series ortogonales.

Teorema de muestreo: Cualquier forma de onda física puede representarse sobre el intervalo $-\infty < t > \infty$ mediante la ecuación 2.8.

$$w(t) = \sum_{n=-\infty}^{n=\infty} an \frac{\sin\{\pi f_s\left[t-\left(\frac{n}{f_s}\right)\right]\}}{\pi f_s\left[t-\left(\frac{n}{f_s}\right)\right]} \quad (2.8)$$

Donde an se representa con la ecuación 2.9.

$$an = f_s \int_{-\infty}^{\infty} w(t) \frac{\sin\{\pi f_s\left[t-\left(\frac{n}{f_s}\right)\right]\}}{\pi f_s\left[t-\left(\frac{n}{f_s}\right)\right]} dt \qquad (2.9)$$

Y donde:

f_s = un parámetro al cual se le asigna un valor conveniente mayor a cero.

$w(t)$=está limitada en un ancho de banda "B" *[Hertz]* y $f_s \geq 2B$, entonces la ecuación 2.9 se convierte en la representación de la función de muestreo, donde la ecuación 2.10, lo indica

$$an = w(n/f_s) \qquad (2.10)$$

donde:

n=es el número de muestra.

f_s= frecuencia de muestreo.

Esto es que, para $f_s \geq 2B$, los coeficientes de la serie ortogonal son simplemente los valores de la forma de onda generados cuando se obtiene una muestra dada $1/f_s$ segundos

2.6 Frecuencia de muestreo (Teorema de Nyquist).

El teorema de Nyquist establece la frecuencia mínima de muestreo (f_s) que se puede realizar en un determinado sistema. Para que una muestra se produzca con exactitud en el receptor, se debe muestrear cuando menos dos veces cada ciclo de la señal analógica de entrada. En consecuencia, la frecuencia mínima de muestreo es igual al doble de la frecuencia máxima de la entrada de audio.

De la ecuación 2.10 se tiene:

$$a_n = \int_{-f_s/2}^{f_s/2} w(f) e^{+j2\pi f\left(\frac{n}{f_s}\right)} df \qquad (2.11)$$

donde:

a_n= es la representación de la función de muestreo.

Resulta que la máxima velocidad de muestreo permitida para la reconstrucción de una forma de onda limitada por banda sin errores está dada por la expresión $f_s = 2B$

A esto se le conoce como la *frecuencia de Nyquist.*

2.7 Técnicas de cuantificación.

En la cuantificación el valor de cada muestra de la señal se representa como un valor elegido de entre un conjunto finito de posibles valores.
Se conoce como error de cuantificación, a la diferencia entre la señal de entrada (sin cuantificar) y la señal de salida (ya cuantificada). Para tener un bajo error de cuantificación se utilizan diferentes tipos de cuantificación:

2.7.1 Cuantificación uniforme:

En los cuantificadores uniformes o lineales la distancia entre los niveles de reconstrucción es siempre la misma, la mayoría usan un número de niveles que es una potencia de dos. No hacen ninguna suposición acerca de la señal a cuantificar, de allí que no proporcionen los mejores resultados. Pero son los más fáciles y menos costosos a implementar.

2.7.2 Cuantificación logarítmica:

Para evitar desperdicio de niveles de reconstrucción y de ancho de banda se utiliza un método sencillo para mejorar el incremento de la distancia entre los niveles de reconstrucción conforme aumenta la amplitud de la señal. Para conseguir esto se hace pasar la señal por un compresor logarítmico antes de la cuantificación. Esta señal comprimida puede ser cuantificada uniformemente. A la salida del sistema la señal pasa por un expansor. A esta técnica se le llama compresión.

2.7.3 Cuantificación no uniforme:

Este cuantificador utiliza la función de la distribución de probabilidad, conociendo esto se puede ajustar los niveles de reconstrucción a la distribución de forma que se minimice el error cuadrático medio.

2.7.4 Cuantificación vectorial:

Este método cuantifica los datos en bloques de N muestras. En este tipo de cuantificación, el bloque de N muestras se trata como un vector N-dimensional

CAPITULO 3. ETAPAS DE UN RECONOCIMIENTO DE VOZ.

3.1 El MICRÓFONO

El micrófono es un transductor electroacústico, su función es transformar la presión acústica ejercida sobre su cápsula por las ondas sonoras en energía eléctrica.
Cuando un micrófono está operando, las ondas de sonido hacen que vibre el elemento magnético del micrófono causando una corriente eléctrica, la calidad de la señal tiene las siguientes características:

- Sensibilidad: es la eficiencia del micrófono, la relación entre la presión sonora que incide y la tensión eléctrica de salida. Por lo que tenemos una relación directamente proporcional a mayor presión sonora mayor tensión eléctrica a la salida.

- Fidelidad: se define como la respuesta en frecuencia del micrófono, indica la variación de la sensibilidad con respecto de la frecuencia.

- Directividad: indica en qué dirección existe mayor amplitud de la señal captada.

  Existen diferentes tipos de micrófonos clasificados de acuerdo a su directividad:

  a) Omnidireccionales: captan todos los sonidos, sin importar la dirección desde donde lleguen.
  b) Bidireccionales: captan tanto el sonido que llega por su parte frontal, como por su parte posterior.
  c) Unidireccionales o direccionales: captan el sonido en una sola dirección mientras que son relativamente sordos a las otras direcciones.

- Ruido de fondo: es la tensión que obtenemos del micrófono sin que exista presión acústica sobre él, el valor de este regularmente está entorno a los 60 dB.

- Rango dinámico: es el margen que hay entre el nivel pico y el nivel de ruido de fondo, esta expresado en dB.

- Impedancia interna: es la resistencia en función de la frecuencia, que tiene el micrófono al paso de la corriente. La impedancia se caracteriza según su valor:
  a) Lo-Z Baja impedancia (alrededor de 200 Ω)
  b) Hi-Z Alta impedancia (1 K Ω o 3 K Ω e incluso 600 Ω)
  c) VHi-Z Muy alta impedancia (más de 3 K Ω).

Los micrófonos de clasifican en tres grupos:

Según el tipo de transductor:

- Micrófono de Condensador o Capacitor: su característica es que las ondas sonoras provocan el movimiento oscilatorio del diafragma, el cual actúa como una de las placas de un capacitor, esta vibración provoca una variación en la energía almacenada y en el condensador, que forma el núcleo de la capsula microfónica. Esta variación genera una tensión eléctrica que es la señal de salida del sistema. La señal de salida de este sistema es análoga.

Se dividen en tres tipos:

a) Micrófono de condensador DC.
b) Micrófono de condensador electret.
c) Micrófono de condensador de radiofrecuencia (RF).

- Micrófono Dinámico: trabajan por medio de inducción electromagnética, la vibración del diafragma provoca el movimiento de una bobina móvil o cinta corrugada ancladas a un imán permanente que genera un campo magnético que a su vez genera una tensión eléctrica, que es la señal de salida. Esta señal eléctrica es análoga.

  Hay dos tipos básicos:
  a) Micrófono de bobina móvil o dinámica.
  b) Micrófono de cinta.

- Micrófono piezoeléctrico: utilizan el fenómeno de piezoelectricidad, cuando las ondas sonoras hacen vibrar el diafragma el movimiento de este hace que se mueva el material contenido en su interior (cuarzo, sales de Rochélle, carbón, etc.). La fricción entre estas partículas generan sobre la superficie del material una tensión eléctrica, la respuesta en frecuencia de estos micrófonos es muy irregular.
  Tipos de micrófonos piezoeléctricos.
  a) Micrófono de carbón
  b) Micrófono de cristal
  c) Micrófono de cerámica.

El diagrama de conexión para el micrófono se muestra en la figura 3.1.

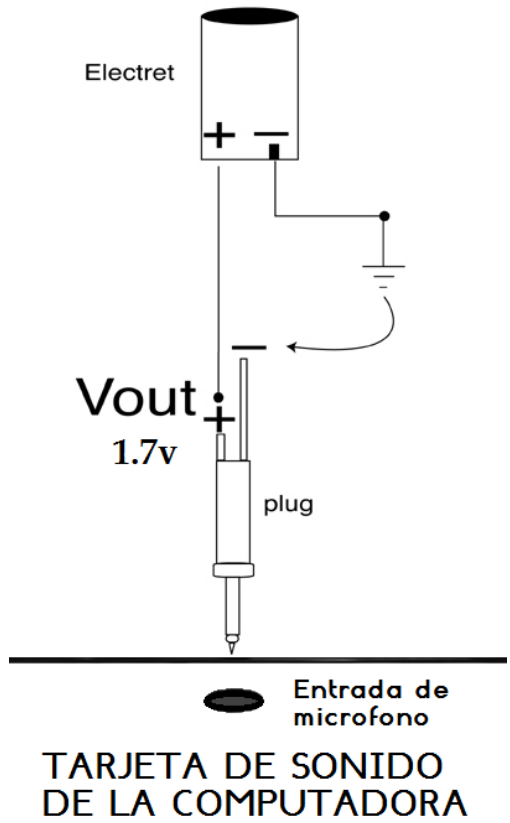


Figura 3.1 Diagrama de conexión del micrófono electret.

3.2 Normalización.

La técnica de normalización ayuda a evitar la degradación de una señal de voz. El problema se presenta cuando existen diferentes locutores cada uno con características diferentes en la señal de voz emitida debido a diferencias fonéticas y fonologías de los locutores, otra razón por la que un sistema de reconocimiento de voz puede bajar su desempeño es por las diferencias que hay entre los datos de entrenamiento y los datos de evaluación.

Existen diferentes obstáculos para lograr que un reconocedor de voz sea robusto, la robustez en reconocimiento de voz, se refiere a la necesidad de mantener una buena precisión aun cuando la calidad de la voz de entrada este degradada, o cuando las características acústicas, articulatorias o fonéticas de la voz en los ambientes de entrenamiento y evaluación difieran. Para ser más específicos, la normalización de una señal de voz, es hacer que el ruido aditivo, efectos de filtración lineal, fenómenos de transducción o transmisión, así como fuentes impulsivas de interferencia no produzcan cambios en la señal de voz.

La técnica de normalización consta de fijar un límite de amplitud superior y un límite de amplitud inferior, con el fin de tener un mayor control en la señal de voz adquirida.

3.3 Filtros de respuesta de impulso finito (FIR).

FIR es una abreviatura en inglés para *Finite Impulse Response* o respuesta finita al impulso, son también conocidos como filtros no recursivos o filtros de convolución, se caracterizan por ser filtros digitales que como su nombre lo indica si a la señal de entrada se tiene una señal impulso, a la salida se tendrá un numero finito de términos, y además de ser los filtros más sencillos para el diseño. Los filtros FIR realizan una convolución de los coeficientes del filtro con una secuencia de valores de entrada y producen una secuencia de igual numeración a los valores de salida. La ecuación que define la convolución finita de un filtro FIR es:

$$y_i = b_0 x(i) b_1 x(i-1) + \cdots + b_{N-1} x(i-N+1) = \sum_{k=0}^{n-1} h(k).x(i-k) \quad (3.1)$$

donde:

x: es la secuencia de entrada para filtrar.

y: es la secuencia filtrada.

h: es los coeficientes del filtro FIR.

Los filtros FIR tienen las siguientes características:

- Pueden lograr fase lineal debido al coeficiente de simetría en la realización.
- Son estables.
- Permiten filtrar señales mediante la convolución.

Por lo tanto se puede asociar un retraso en la secuencia de salida, como se muestra en la ecuación 3.2:

$$r = (n-1)/2 \quad (3.2)$$

donde:

r= retardo [s].

n= es el número de coeficientes del filtro FIR.

La ventaja de los filtros FIR es que pueden diseñarse para que presenten fase lineal, la linealidad de fase implica que se verifiquen ciertas condiciones de simetría:

- Un sistema no causal con respuesta impulsional conjugada simétrica tiene una Función de Transferencia real.

- Un sistema no causal con respuesta impulsional conjugada antisimétrica tiene una Función de transferencia imaginaria pura.

  Si consideramos sistemas FIR con coeficientes reales, una secuencia conjugada simétrica se dice que es una secuencia par, y una secuencia conjugada antisimétrica es una secuencia impar. Dependiendo del número de coeficientes del filtro y del tipo de simetría tenemos varias posibilidades; ver tabla 3.1

Tabla 3.1. Tipos de simetría y números de coeficientes de un filtro.

Tipo.	Número de términos.	Simetría.
I	Impar.	Simétrico. $h(k) = h(N-1-k)$
II	Par.	Simétrico. $h(k) = h(N-1-k)$
III	Impar.	Antisimétrico. $h(k) = -h(N-1-k)$
IV	Par.	Antisimétrico. $h(k) = -h(N-1-k)$

Durante la adquisición de voz puede haber pérdida de ganancia en las altas frecuencias, ya sea por la etapa de resonancia o debido a la respuesta del micrófono. Una ventaja del filtro FIR es que se puede diseñar pasa alta, con esto aseguramos una señal más uniforme y así no se pierde información durante la segmentación.

3.4 Segmentación.

La segmentación de una señal consiste en dividir una señal en diferentes tramas basándose en algún criterio. La forma más común para la segmentación de señales de voz es por fonemas aunque también se hace con silabas. Para la segmentación de voz se utilizan distintas técnicas como lo es la segmentación manual, la segmentación basada en modelos ocultos de markov, redes neuronales artificiales, modelos estadísticos y el filtrado paramétrico; en los que generalmente se realizan en base a espectrogramas, curvas de energía, entonación y además de otros estudios utilizados para el análisis de la voz.

En el caso más simple, la segmentación consiste en encontrar los limites más precisos que definan a cada segmento o unidad fonética, cada segmento presenta dos limites o marcadores que miden el tiempo desde el inicio de la emisión, en el que se encuentran

el principio y final de cada segmento. Una señal de voz puede tener muchos segmentos y así la ubicación correcta de todos sus límites puede ser un problema más complejo. Más aún si se consideran todas las variaciones asociadas con los distintos lenguajes.

3.4.1 Algoritmo evolutivo para la segmentación de voz.

Para obtener los vectores de características se parte de un análisis por tramos de la señal de voz:

$$\mathrm{x(t;k)} = \tau(\mathrm{k})\{v(t;n)\}, \quad 0 < k < N_x$$

donde:

x(t; k): son los vectores caracterizados por tramos de la señal de voz.

$\tau(\mathrm{k})$: es un operador para la transformación de dominio.

$\{v(t;n)\}$: son los tramos de voz en el tiempo.

La segmentación da como resultado un conjunto $\emptyset = \{E_m\}$ donde cada segmento E_m contiene vectores de características x(t) con determinado grado de pertenencia. Sobre esta definición general se hacen dos restricciones.

La primera es considerar que la segmentación es totalmente exclusiva, es decir, cada vector de características puede pertenecer a sólo un segmento. Ver ecuación 3.3.

$$\mathrm{x(t;k)} \in E_{j1} \leftrightarrow \mathrm{x(t;k)} \in E_{j2} \forall j2 \neq j1 \tag{3.3}$$

Esto permite describir el grado pertenencia asociado a cada vector.

La segunda restricción está en que no se debe de invertir el orden temporal de los vectores caracterizados de cada segmento. Las dos restricciones se pueden expresar conjuntamente mediante la ecuación 3.4

$$\mathrm{X(t;k)} \in E_{j1} \Delta \mathrm{x}(t_2;\mathrm{k}) \in E_{j2} \leftrightarrow t_1 < t_2 \forall j_1 < j_2 \tag{3.4}$$

Dadas estas restricciones, se puede representar la segmentación mediante el vector de los marcadores del primer elemento de cada segmento, ver ecuación 3.5

$$\emptyset = [M_1, M_2, \dots, M_{N_\emptyset}] \text{ Con } N_\emptyset = |\emptyset| + 1 \tag{3.5}$$

donde:

$\emptyset$ = vector de marcadores

Ya que se incluyen los marcadores inicial y final y además $1 < M_1 < M_2 < \cdots < M_{N_\emptyset} \leq T + 1$.

3.5 Ventaneo.

Para entender como una ventana determinada afecta el espectro de frecuencias de una señal, es necesario comprender más acerca de las características de las frecuencias en las ventanas. El ventaneado de una señal, es equivalente a la convolución del espectro

de la señal original con el espectro de la ventana, Incluso si no se utiliza alguna ventana, la señal se convolucióna con una ventana rectangular de altura uniforme, por la naturaleza de tomar un instante en el tiempo de la señal de entrada. Una trama real de una ventana que muestra la característica de frecuencia de una ventana es un espectro continuo con un lóbulo principal y varios lóbulos laterales. El lóbulo principal se centra en cada componente de frecuencia de la señal de dominio del tiempo, y los lóbulos laterales se acercan a cero. La resolución de la frecuencia, está limitada por el ancho del lóbulo principal del espectro de ventana. La capacidad de distinguir dos componentes de frecuencia estrechamente espaciados aumenta a medida que el lóbulo principal se estrecha, provocando así que la resolución espectral mejore y la energía de la ventana se extiende en sus lóbulos laterales.

Para seleccionar una ventana espectral, se debe tener en cuenta el contenido de frecuencia de la señal a ventanear se debe utilizar:

- elegir una ventana con un lóbulo tasa *de roll-off* de lado alto, si la señal contiene componentes de frecuencia interferente, fuerte y distante de la frecuencia de interés
- la ventana con un nivel de lóbulo lateral máximo bajo, si hay fuertes señales de interferencia cerca de la frecuencia de interés.
- elegir una ventana con un lóbulo principal muy estrecho. Si la frecuencia de interés está cerca de dos o más señales, ya que la resolución espectral es importante.
- elegir una ventana con un amplio lóbulo principal, si la precisión de la amplitud de un solo componente de frecuencia es más importante que la ubicación exacta del componente en un contenedor de frecuencia determinado.
- usar la ventana Uniforme (sin ventana), si el espectro de la señal es más bien plana o banda ancha en contenido de frecuencia.

3.5.1 Ventana Hamming.

El método de ventana de Hamming es el más utilizado para el análisis de señales de voz, la ventaja de este método es una alta resolución en el dominio de frecuencia y su fuga espectral es pequeña, ya que la atenuación de la túnica de lado es más de 43 dB [7].

Se define de acuerdo a la ecuación 3.6.

$$W_H(n) = 0.54 - 0.46\cos(\frac{2n\pi}{N-1}) \quad (3.6)$$

donde: n= es elemento a ventanear.

La relación entre el período de muestreo $T[s]$ y el número de muestras para el análisis de N, y la resolución de frecuencia nominal del espectro calculado $\Delta f(Hz)$ se expresa como:

$$\Delta f = \frac{1}{TN} \tag{3.7}$$

donde:
T= periodo de muestreo.
N= número de muestras.

3.6 Coeficientes cepstrales (cepstrum).

El cepstrum, o coeficiente cepstral $c(\tau)$, se define como la transformada inversa de Fourier del espectro de amplitud logarítmica en tiempo corto. El término cepstrum es esencialmente un término acuñado que incluye el significado de la transformada inversa del espectro. El parámetro independiente para el cepstrum se llama *quefrency*, que obviamente está formado a partir de la característica de función de dominio de la frecuencia. Ya que el cepstrum es la transformada inversa de la función de dominio de la frecuencia, la *quefrency* se convierte en el parámetro de dominio de tiempo. La característica especial de la cepstrum es que permite la representación por separado de la envolvente espectral y la estructura fina, basado en el modelo de circuito equivalente separable lineal descrito. La voz $x(t)$ puede ser considerada como la respuesta del filtro de tracto vocal, equivalente articulación impulsada por una fuente de pseudoperiodica $g(t)$, entonces $x(t)$ puede ser dada por la convolución de $g(t)$, y del tracto vocal de respuesta de impulso h (t) como se muestra en la ecuación 3.8.

$$x(t) = \int_0^t g(\tau)h(t-\tau)\,d\tau \tag{3.8}$$

Que es equivalente a la ecuación 3.9.

$$X(w) = G(w)H(w) \tag{3.9}$$

donde: $X(w)$, $G(w)$, $y\ H(w)$ son la transformada de Fourier de $x(w), g(w), y\ h(w)$ respectivamente.

Si $g(\tau)$ es una función periódica, $|X(w)|$ está representado por las líneas espectrales, los intervalos de frecuencia de los cuales son el recíproco de la periódica fundamental de $g(\tau)$. Por lo tanto, cuando $|X(w)|$ se calcula mediante la

transformada de Fourier de la secuencia de tiempo de la muestra por un período de onda de voz corto, exhibe picos agudos con intervalos iguales a lo largo del eje de frecuencia, la ecuación 3.10 define su logaritmo.

$$\log|X(w)| = \log|G(w)| + \log|H(w)| \tag{3.10}$$

El cepstrum, que es la transformada inversa de Fourier del log | X (w) |, es:

$$c(\tau) = F^{-1}\log|X(w)| = F^{-1}\log|G(w)| + F^{-1}\log|H(w)| \tag{3.11}$$

donde $c(\tau)$: son los coeficientes cesptrales de la función periódica.

Donde F es la transformada Fourier. El primer y segundo términos en el lado derecho de la ecuación 3.10, que corresponden con la estructura fina espectral y la envolvente espectral, respectivamente. El primero es el patrón periódico, y el último es el patrón global a lo largo del eje de frecuencia. En consecuencia, se producen grandes diferencias entre la transformada inversa de Fourier de las funciones de ambos elementos indicados en la ecuación 3.11.

Principalmente, la primera función en el lado derecho de la ecuación 3.11 indica la formación de un pico en la región de alta *quefrency*, y la segunda función representa una concentración en la región de baja *quefrency* de 0, 2 o 4 ms. El periodo fundamental de la fuente $g(t)$, entonces se puede extraer desde el pico en la región de alta quefrency. Por otro lado, la transformada de Fourier de los elementos de baja - *quefrency* produce la envolvente espectral logarítmica que se puede obtener a través de la transformada de exponencial. El orden máximo de elementos de baja *quefrency* utilizado para la transformación determina la suavidad de la envolvente espectral. El proceso de separación de los elementos cepstrales en estos dos factores se llama *liftering,* que se deriva de filtrado. Cuando el valor cepstrum se calcula con la DFT (transformada discreta de Fourier por sus siglas en ingles), es necesario establecer el valor base de la transformación, N, lo suficientemente grande como para eliminar el *aliasing* similar a la producida durante el muestreo de forma de onda, el cepstrum queda definido por la ecuación 3.12.

$$c_n = \frac{1}{N}\sum_{k=0}^{N-1}\log|X(k)|\, e^{i2\pi kn/N} \qquad (0 \le n \le N-1) \tag{3.12}$$

donde:

c_n= coeficiente cepstrum obtenido al evaluar en el elemento n.

K= el coeficiente del cepstrum

3.6.1 Parámetros delta cepstrum.

Se obtiene la estimación de la rapidez de variación de la función temporal de cada parámetro en cada segmento de cada muestra, es decir, el promedio de la función de tiempo de cada coeficiente cepstral de cada segmento en cada muestra, mediante el cálculo de la pendiente del polinomio de orden uno que mejor se aproxima a esta función temporal, lo que se conoce como Parámetros Delta cepstrum.

Realizar este cálculo ayuda a darle mayor certeza en el reconocimiento a nuestro algoritmo, pues realiza el cálculo para la aproximación de la derivada del espectro instantáneo de la voz, siendo este más robusto frente a la variabilidad del interlocutor y del entorno. Nos proporciona más información sobre la señal de voz para que el algoritmo supere errores de reconocimiento debidas al hablante o el medio.

Este cálculo se realiza aproximando la derivada mediante un polinomio ortogonal de primer orden, que utiliza los datos de un intervalo, centrado en el segmento actual, de longitud 2k+1 segmentos. Ver ecuación 3.13.

$$\Delta c(n,t) = \frac{\sum_{k=-K}^{K} k\ c(n,t+k)}{\sum_{k=-K}^{K} k^2} \tag{3.13}$$

donde:
k: representa el radio del intervalo, es decir, el desplazamiento hacia adelante y hacia atrás.

Una vez que ya se tiene calculado el vector cepstral pesado $\hat{C}(m)$, y el vector cepstral derivado para cada segmento $\hat{C}(m)$, la concatenación de ambos forma el vector observación "O" final correspondiente a dicho segmento, como se muestra en la ecuación 3.14.

$$O = \left\{ \hat{C}(m),\ \Delta\hat{C}(m) \right\}. \tag{3.14}$$

3.7 Distancias euclidianas.

Una característica fundamental en los sistemas de reconocimiento de voz es la forma en que es comparada la señal de voz adquirida con los patrones de referencia. Para poder realizar estas operaciones es necesario definir una medida de distancia entre los vectores característicos. Por ejemplo si x_i y t_i con $i = 0,1,2,\dots,L$ son las componentes de dos vectores característicos, el método de distancia euclidiana centra la clase en un patrón de características resultado de la

media del número de muestras inicialmente tomadas. La distancia desde cualquier muestra nueva a ese centro se mide mediante la ecuación 3.15:

$$d = \sqrt{\sum_{i=1}^{L}(x_i - t_i)^2} \tag{3.15}$$

donde:

L= la dimensión del vector de características
x_i= la i-ésima componente del vector de características
t_i = la i-ésima componente del patrón.

3.8 Distorsión dinámica temporal (DTW).

El método distorsión dinámica temporal (DTW por sus siglas en ingles *Dynamic Time Warping*) se basa en la variación en la escala del tiempo, de dos palabras a comparar. El problema que se presenta cuando se pronuncia una palabra es que esta no siempre se realiza a la misma velocidad, lo que produce importantes distorsiones temporales. Estas distorsiones afectan no sólo a la palabra considerada sino también a sus componentes acústicos. Las variaciones temporales no son generalmente proporcionales a la velocidad de locución y podrán variar de locutor a locutor. Es por esto que se hace necesario un procedimiento que permita comparar dos palabras, sin considerar las distorsiones temporales. Los métodos que se usan para realizar lo expuesto se basan en algoritmos de programación dinámica. Ver figura 3.2.

Si tenemos una señal de voz patrón de prueba $T = \{t_1, \dots, t_N\}$ y consideramos un patrón de referencia $R = \{r_1, \dots, r_M\}$

La distorsión temporal $D(T, R)$ va a estar basada en una suma de distancias locales entre elementos $d(t_i, r_j)$, apartir de un alineamiento especial $\emptyset$ el cual alinea a T y R mediante un mapeo de punto a punto $\emptyset = (\emptyset_t, \emptyset_r)$, de longitud $k_\emptyset$ por lo que el alineamiento óptimo minimiza la distorsión global, la ecuación 3.16 muestra la distancia.

$$D_\emptyset(T, R) = \frac{1}{M_\emptyset}\sum_{k=1}^{k_\emptyset} d(t_{\emptyset_t}(k), r_{\emptyset_r}(k))m_k \tag{3.16}$$

donde:

k= es el número correspondiente al elemento en evaluación.

t= son los elementos del patrón de prueba.

r= son los elementos del patrón de referencia.

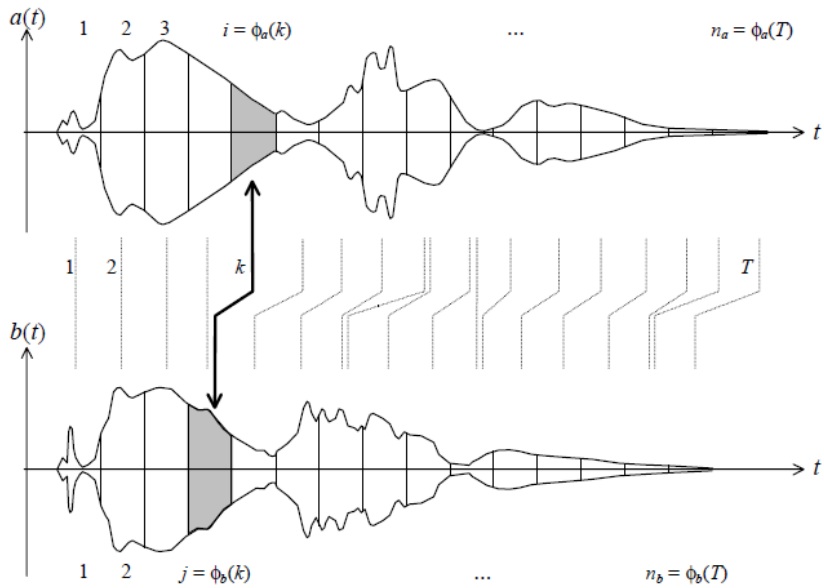


Figura 3.2 Grafica de la alineación mediante la distorsión dinámica temporal. Se realizan las comparaciones entre cada segmento de la señal de voz adquirida (arriba) y la señal de voz en la base de datos (abajo).

El objetivo es comparar dos emisiones desde el punto de vista del cepstro de sus tramas fonéticamente homólogos, es decir aquéllos que corresponden a fonemas iguales. En este contexto, cada elección de coeficientes trae aparejada una decisión acertada o no sobre qué tramas han de considerarse homólogos. Se define a partir de dicha alineación, una distancia entre ambas emisiones igual a la suma de las distancias entre los vectores cepstrales que representan a cuadros homólogos

La mejor alineación es aquélla que permite comparar entre sí cuadros fonéticamente parecidos, es decir, aquéllos cuya distancia cepstral sea inherentemente pequeña. Dado que esta condición debe cumplirse para cada par de cuadros de A y B a comparar, equivale a requerir que la distancia total sea mínima. En otras palabras, la distorsión temporal óptima es aquélla que hace mínima la distancia entre A y B, la ecuación se muestra a continuación. La ecuación 3.16 muestra la distancia.

$$d_{\emptyset}(A,B) = \sum_{K=1}^{T} d(A_{\emptyset_a}(k), B_{\emptyset_b}(k)) \qquad (3.16)$$

donde:

k= segmento evaluado.
T= periodo de la señal.

La determinación de dicha distancia mínima requeriría, en principio, la inspección de todos los posibles caminos y el cálculo de la distancia correspondiente a cada uno de ellos. Las condiciones de contorno establecen que el primer cuadro del segmento de referencia debe compararse con el primer cuadro del segmento a comparar y, análogamente, el último cuadro de uno debe compararse con el último cuadro del otro. Un algoritmo más eficiente para resolver problemas de optimización de este tipo es la programación dinámica, que permite reducir la cantidad de casos explorados apoyándose en el denominado principio de optimalidad. Este principio establece que si un camino que une dos puntos X y Z pasando por una serie de puntos intermedios, es óptimo según determinado criterio, entonces al subdividirlo en dos tramos XY e YZ, cada uno de ellos también es el camino óptimo entre sus respectivos extremos. La programación dinámica aplica este principio recursivamente aumentando en una unidad la cantidad de puntos en cada iteración. Una ecuación para determinar el "costo" parcial para conectar el punto (1, 1) con el (i, j) es la mínima distancia parcial posible, abarcando todos los caminos posibles se obtiene mediante la ecuación 3.18.

$$\boldsymbol{\varphi_{(i,j)}(A,B) = \min_{\emptyset}\{\frac{d_{\emptyset,h}(A,B)}{\emptyset_b(h)} = (i,j)\}} \quad \textbf{(3.18)}$$

donde:

φ= mínima distancia parcial.
i, j= son los respetivos valores de cada iteración.
A, B= son los vectores en evaluación.

De igual forma se define el "costo" entre el punto medio y el punto (l, m) donde queremos que se haga el cálculo (i, j) con la ecuación 3.19.

$$\zeta((l,m),(i,j)) = d\left(A_{\phi_a(k)}, B_{\phi_b(k)}\right)m(k) + d\left(A_{\phi_a(k-1)}, B_{\phi_b(k-1)}\right)m(k-1) \quad \textbf{(3.19)}$$

donde:

(l, m)=elemento en donde se define el costo.
(i, j)= elemento en evaluación

Por lo que el algoritmo final queda definido por la ecuación 3.20.

$$\Psi_{(i,j)}(A,B) = \min_{(l,m)}\left\{\Psi_{(l,m)}(A,B) + \zeta\left((l,m),(i,j)\right)\right\}. \quad (3.20)$$

donde:

Ψ= distorsión dinámica temporal.

CAPITULO 4: LA HERRAMIENTA COMPUTACIONAL LABVIEW.

4.1 Introducción al LabVIEW.

LabVIEW es una plataforma de programación gráfica que ayuda a ingenieros a escalar desde el diseño, hasta pruebas y desde sistemas pequeños hasta grandes sistemas. Ofrece integración sin precedentes con software legado existente, IP y hardware al aprovechar las últimas tecnologías de cómputo. Labview ofrece herramientas para resolver los problemas de hoy en día y la capacidad para la futura innovación, más rápido y de manera más eficiente [7].

4.2 La instrumentación virtual.

Un instrumento virtual es un módulo software que simula el panel frontal de un instrumento de medida y se apoya en hardware como tarjetas de adquisición, tarjetas DSP (procesador digital de señal por sus siglas en ingles), instrumentos accesibles vía GPIB (*General Purpose Interface Bus*) bus de interfaz de uso general , RS-232 (Recommended Standard 232) norma recomendada 232, USB (*Universal Serial Bus*) bus universal serial, de este modo, cuando se ejecuta un programa que funciona como instrumento virtual o VI (*Virtual Instrument*) el usuario ve en la pantalla su computadora un panel cuya función es idéntica a la del instrumento físico, facilitando la visualización y control del aparato.

4.3 Programación gráfica.

Cuando se crea un VI en LabVIEW se trabaja con dos ventanas una en la que se implementa el panel frontal y otra que soporta el nivel de programación llamada diagrama de bloques, para la creación del panel frontal se dispone de una librería de controles e indicadores de todo tipo y la posibilidad de crear más, por el propio usuario.

Cuando un control es pegado desde la librería en el panel frontal se acaba de crear una variable cuyos valores vendrán determinados por lo que el usuario ajuste desde el panel; inmediatamente aparece una terminal una terminal en la ventana de programación representándolo.

4.4 Formato UFF.

UFF (Formato Universal de Archivo por sus siglas en inglés) es utilizado por *National Instruments* para poder compartir archivos e información entre sus diferentes herramientas de software. Fue desarrollado inicialmente para estandarizar la transferencia de datos entre el diseño asistido por computadora (CAD) y la prueba asistida por computadora (CAT).

El formato define un encabezado que contiene información general sobre los datos contenidos en el archivo (tipo de función, la dirección de respuesta), así como el canal de información específica (nombre del canal, unidades, tipo de datos, etc.). Para reducir el espacio de almacenamiento, así como el tiempo de carga, se introdujo el formato de archivo binario universal. Estos archivos contienen la misma información que el tipo ASCII, la única diferencia es la forma en que los valores de los datos se almacenan

4.5 Componentes de un diagrama en labview.

Como se mencionó anteriormente, los programas creados en labview reciben el nombre de instrumentos virtuales, cada VI consta de tres componentes:

1.- Un panel frontal (*front panel*). Es la interfaz del usuario.
2.- Un diagrama de bloques (*block diagram*). Contiene el código fuente gráfico que define la funcionalidad del VI.
3.- Icono conector. Identifica a cada VI de manera que podemos utilizarlo dentro de otro VI recibe el nombre subVI que es una subrutina hecha con anterioridad dentro de otro VI.

4.6 Características de los Diagrama bloques.

LabVIEW nos proporciona un diagrama de bloques para realizar una tarea específica, las características de estos bloques y su uso se presenta a continuación:

Flat Séquense Estructure (estructura de secuencia plana)

Este bloque consiste en uno o más subdiagramas, marcos, segmentos o tramas que se ejecutan secuencialmente. Se utiliza la estructura de secuencia plana para asegurarse de que un subdiagrama se ejecuta antes o después de otro sub diagrama.

Los comandos en la estructura de secuencia plana se ejecutan de izquierda a derecha y cuando todos los comandos dentro de los segmentos (o tramas) de la estructura son válidos. Esto significa que los datos de entrada de una trama pueden depender de la salida de la trama anterior. Ver figura 4.1.

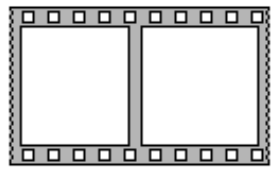


Figura 4.1. Diagrama bloque estructura de secuencia plana

***Wait (ms) Function* (función de espera en ms).**

Espera el número especificado de milisegundos y devuelve un valor del contador en milisegundos. Ver figura 4.2.

Figura 4.2. Diagrama bloque función de espera en ms.

Variable local.

Cuando se crea una variable local, aparece en forma de icono en el diagrama de bloques. Esta variable local puede recibir información y pasarla hacia el bloque donde esté conectada, entonces mandar información a una variable local es como pasar datos a cualquier otra terminal. Una variable local, puede acceder a un objeto del panel frontal tanto como una entrada y una salida. Ver figura 4.3.

Figura 4.3. Diagrama bloque de una variable local.

***True / False constant* (constante Verdadero / Falso).**

Este bloque entrega un valor falso o verdadero. Ver figura 4.4.

Figura 4.4. Diagrama bloque Verdadero / Falso constante.

***Acquire Sound Express VI* (Adquisición exprés de sonido).**

Adquiere datos de un dispositivo de sonido. Este VI Exprés configura automáticamente una tarea de entrada, adquiere los datos, y borra la tarea una vez finalizada la adquisición. Ver figura 4.5.

Figura 4.5. Diagrama bloque para la adquisición exprés de sonido.

Sus conexiones de entrada y salida son descritas en la tabla 4.1 y 4.2.

ENTRADAS:

Tabla 4.1. Conexiones de entradas para el diagrama boque adquisición exprés de sonido.

PARÁMETRO	DESCRIPCIÓN
***Device* (dispositivo)**	Enumera los dispositivos de sonido que estén conectados.
Duration* (duración) *en seg.	Establece el número de segundos que desea adquirir sonido.
***Sample rate* (frecuencia de muestreo) en Hz.**	Especifica la Frecuencia de muestreo en [Hz].
***#Channels* (número de canales)**	Especifica el número de canales, 1 para Monofónico y 2 para Estéreo.
Resolution* (resolución) *en bits	Especifica la calidad de cada muestra en bits. El valor predeterminado es de 16 bits

SALIDA:

Tabla 4.2. Conexiones de salida para el diagrama boque adquisición exprés de sonido.

***Data* (dato).**	Devuelve los datos que este *VI* Exprés adquiere desde el dispositivo seleccionado con los valores que se especifiquen en el cuadro de diálogo de configuración. Se puede convertir esta salida a una forma de onda o una matriz unidimensional de ondas (uno por canal) utilizando la función de conversión de datos dinámicos.

Una gran ventaja que LabVIEW brinda, es que varios de sus bloques se pueden configurar también por medio de un cuadro de dialogo, tal como se observa en la figura 4.6.

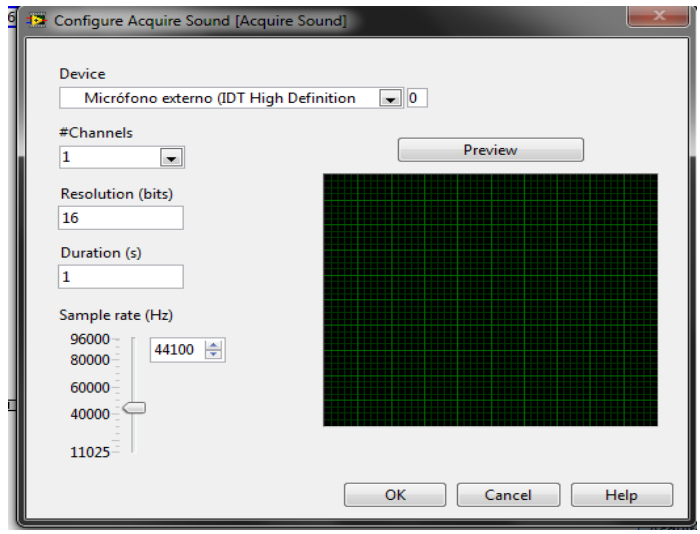


Figura 4.6. Cuadro de dialogo para las características del diagrama boque adquisición exprés de sonido.

Waveform Graph. (*Forma* de onda Gráfico)

Despliega en el panel frontal una gráfica que puede ser configurada para graficar tiempo, frecuencia, voltaje, corriente. Ver figura 4.7.

Figura 4.7. Diagrama boque Forma de onda Gráfico

La grafica obtenida de una señal de tiempo contra amplitud se muestra en la figura 4.8.

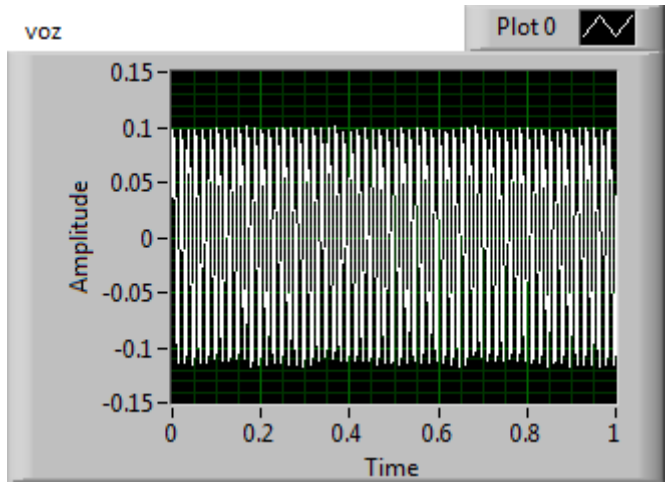


Figura 4.8. Grafica correspondiente al bloque Forma de onda Gráfico

***Convert from Dynamic Data Express VI* (convertidor exprés dinámico de datos).**

Convierte el tipo de datos dinámicos a numérico, Booleano, forma de onda y los tipos de datos de matriz para su uso con otros VI y funciones. Ver figura 4.9.

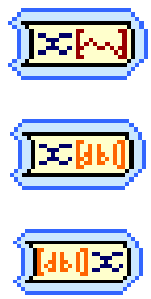


Figura 4.9. Diagrama bloques convertidor exprés dinámico de datos VI.

***Array Max & Min Function* (función arreglo de máximos y mínimos)**

Devuelve los valores máximo y mínimo que se encuentran en la serie de datos que le son introducidos, junto con los índices para cada valor. El conector muestra los tipos de datos predeterminados para esta función polimórfica. Ver figura 4.10.

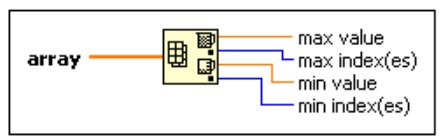


Figura 4.10. Diagrama bloque función arreglo de máximos y mínimos.

***Index Array Function* (función índice del arreglo).**

Devuelve el elemento del subconjunto de n-dimensión de la matriz en el índice. Cuando se cablea un arreglo para esta función, cambia de tamaño automáticamente para mostrar las entradas de índice para cada dimensión de la matriz que se conecte. También se puede agregar elementos adicionales o terminales a un subarreglo cambiando el tamaño de la función. Ver figura 4.11.

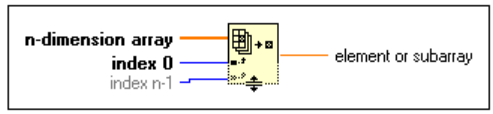


Figura 4.11. Diagrama bloque función índices del arreglo.

***Get Waveform Components* (Ver Componentes de forma de onda)**

Entrega los componentes de la onda analógica que se seleccione. Se puede seleccionar que tipo de componente se desea dando clic en medio de la figura 4.12.

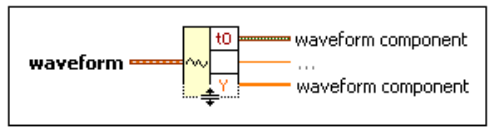


Figura 4.12. Diagrama bloque para Ver los Componentes de forma de onda.

donde:

t0= devuelve el tiempo de activación de la forma de onda.

Y=devuelve los valores de los datos de la forma de onda.

Dt= devuelve el intervalo de tiempo en segundos entre datos en la forma de onda.

Attributes (atributos): devuelve los nombres y valores de todos los atributos de la forma de onda.

***Array Size Function* (función del tamaño de una matriz).**

Devuelve el número de elementos en cada dimensión del arreglo. Ver figura 4.13.

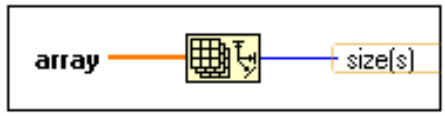


Figura 4.13. Diagrama bloque función del tamaño de una matriz

***Case* Structure (caso estructurado).**

Puede tener uno o más subdiagramas, o casos, los cuales se ejecutan en orden cuando la estructura superior se realiza. El valor que se conecta a la terminal de selección determina el caso a ejecutar y puede ser de tipo booleano, cadena, un entero de tipo enumerado o de error.

Para desplazarse por las subdiagramas o casos disponibles, haga clic en la flechas de incremento y decremento en la etiqueta del selector. Después de crear un caso, puede agregar, duplicar, cambiar o eliminar los subdiagramas. También se pueden crear múltiples entradas y túneles de salida.

Este bloque también puede ser análogo a una estructura IF. Ver figura 4.14.

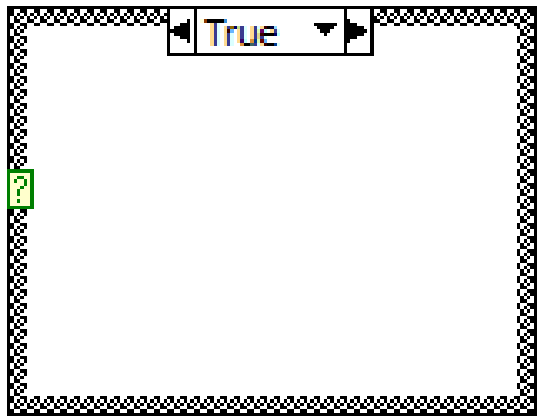


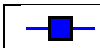
Figura 4.14. Diagrama bloque caso estructurado.

***For Loop* (bucle para).**

Este bloque se ejecuta *n* veces, donde *n* es el valor conectado a la terminal de recuento (N). La terminal de iteración (i) proporciona el conteo de iteraciones en el bucle, que va desde 0 a *n*-1. Ver tabla 4.3 y figura 4.15.

Tipos de entradas:

Tabla 4.3. Tipo de entradas para el diagrama bloque bucle para.

	El túnel pasa los datos de entrada y salida del bucle sin manipulación adicional.
	Los registros de desplazamiento guardan los datos de la iteración anterior y los pasan a la siguiente iteración del bucle.
	Los túneles de auto-*index* leen y procesan un elemento de la matriz o arreglo por cada iteración del bucle.

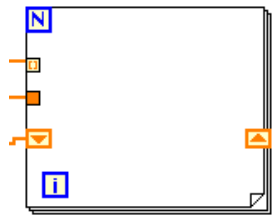


Figura 4.15. Diagrama bloque bucle para.

***Array Subset Function* (Función subconjunto de matrices).**

Devuelve una porción del arreglo comenzando en el índice y sus elementos son de la de longitud indicada. Ver figura 4.16 y tabla 4.4.

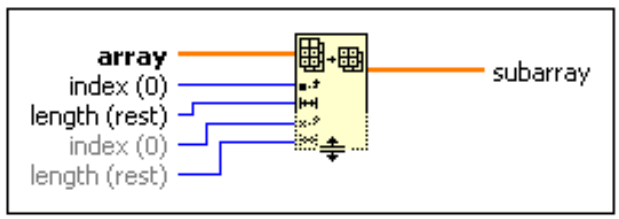


Figura 4.16. Diagrama bloque Función subconjunto de matrices.

ENTRADAS:

Tabla 4.4. Datos de entrada para el bloque Función subconjunto de matrices.

Array (arreglo).	Puede ser una matriz n-dimensional de cualquier tipo.
Index (índice).	Específica el primer elemento, fila, columna o página a incluir en la parte de la matriz que desea devolver. Si el índice es inferior a 0, la función la trata como 0. Si el índice es mayor o igual que el tamaño de la matriz, la función devuelve una matriz vacía.
Lenght (largo).	Específica cuántos elementos, filas, columnas o páginas se incluirán en la parte de la matriz que desea devolver. Si largo del arreglo es mayor que el tamaño de la matriz, la función devuelve sólo la mayor cantidad de datos de que se disponga.

***Hamming Window VI* (ventana Hamming VI).**

Aplica una ventana Hamming a la señal que esté conectada a la entrada de este bloque. Ver figura 4.17 y tablas 4.5 y 4.6.

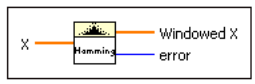


Figura 4.17. Diagrama bloque ventana Hamming VI.

ENTRADA:

Tabla 4.5. Datos de entrada del diagrama bloque ventana Hamming VI.

X	Es la señal a la que se le aplicara la ventana.

SALIDAS:

Tabla 4.6. Datos de salida del diagrama bloque ventana Hamming VI.

ventana *X*	Es la señal a la que se le aplico la ventana.
Error	Devuelve una advertencia si el VI funciona mal.

***Array To Matrix* (Arreglo de matriz)**

Convierte un array a una matriz de elementos del mismo tipo que los elementos del arreglo. Ver figura 5.18.

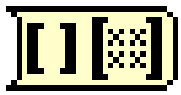


Figura 4.18. Diagrama bloque arreglo de matriz.

Módulo NI LabVIEW *MathScript RT* (Módulo de LabVIEW para Escritura Matemáticas).

El Módulo para Escritura Matemáticas añade programación textual orientada a matemáticas al entorno de desarrollo gráfico LabVIEW con compilador para archivos .m.

MathScript ofrece una interfaz de línea de comando en la cual se pueden cargar, almacenar, diseñar y ejecutar scripts de archivos .m. Conecta las variables de E/S basadas en texto con las entradas y salidas de LabVIEW. Ver figura 4.19.

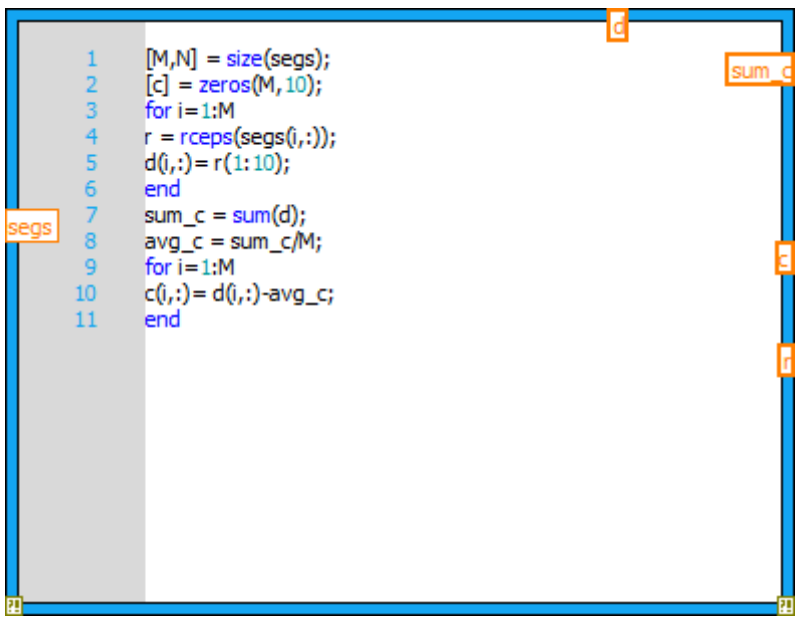


Figura 4.19. Módulo NI LabVIEW *MathScript RT.*

Compound Arithmetic Function **(Función Aritmética compuesta).**

Realiza operaciones aritméticas con una o más variables, series, clúster o entradas booleanas. Para seleccionar la operación (sumar, multiplicar, AND, OR o XOR), se tiene que hacer clic en la función y seleccionar la operación que se desea. Ver figura 5.20.

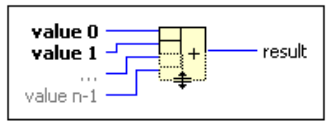


Figura 4.20. Diagrama bloque Función Aritmética compuesta.

Sound File Read Simple **(leer un simple archivo de audio).**

Lee datos de un archivo wav. Este VI abre automáticamente, lee y cierra el archivo WAV. Ver tabla 4.7 y 4.8.

ENTRADAS

Tabla 4.7. Datos de entrada para el bloque leer un simple archivo de audio.

number of samples/ch (número de muestras por canal)	Específica la frecuencia de muestreo por canal.
Path (camino)	Específica la ruta del archivo de sonido. Si la ruta está vacía o no válida, el VI devuelve un error. El valor predeterminado es <*not A Path*>.
position mode (modo de posición)	Especifica dónde comienza la operación de lectura. Puede ser en el comienzo del archivo o en la ubicación actual de la marca de archivo. El valor predeterminado es la ubicación actual.
Position offset (compensación de posición).	Específica la distancia desde la ubicación determinado por el *position mode* para iniciar la lectura. El valor predeterminado es 0.
Error in (error de entrada).	Devuelve una advertencia si el VI funciona mal.

SALIDAS

Tabla 4.8. Datos de salida para el bloque leer un simple archivo de audio.

path out (error de salida)	Identifica la ruta del archivo que fue dada en *path.*
Data (dato)	Entrega los datos del archivo de sonido que fue leído.
Offset (compensar)	Indica la nueva ubicación de la marca de archivo en relación con el principio del archivo, en unidades de muestras. El valor predeterminado es 0.
error in (error de entrada)	Devuelve una advertencia si el VI funciona mal.

***Matrix Size Function* (Función Tamaño Matriz).**

Devuelve el número de elementos en cada dimensión de la matriz. Ver figura 4.21.

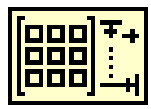


Figura 4.21. Diagrama bloque Función Tamaño Matriz.

CAPITULO 5: DESARROLLO DEL PROYECTO.

En el presente trabajo se presentan las etapas del procesamiento de la señal de voz (obtención de señales de voz, muestreo de la señal, técnicas de reconocimiento de patrones para procesar señales de voz) con el fin de que sean comparadas y reconocidas, para esto se utiliza herramientas computacionales como LabVIEW.
Para lograr el objetivo del presente proyecto se consideran los pasos que se muestran en la siguiente figura:

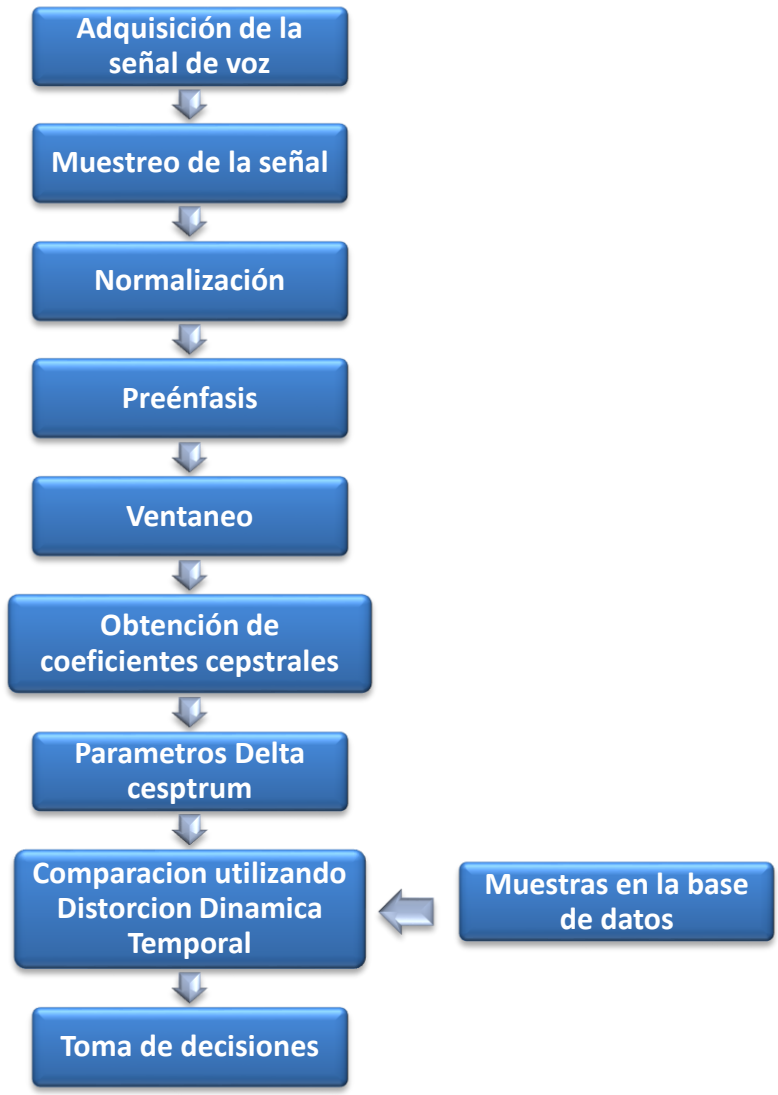


Figura 5.1 diagrama bloques del proyecto.

5.1 Caracterización del micrófono.

Se adquiere la señal de voz utilizando un micrófono electret, por su respuesta a las frecuencias del habla, el cual se conecta a la tarjeta de sonido de la computadora que alimenta al micrófono con 1.7 V siendo innecesario dar alimentación externa al mismo. Para su caracterización se realizó la conexión mostrada en la figura 5.2, las frecuencias a evaluar y los valores obtenidos del voltaje de salida se muestra en la tabla 5.1.

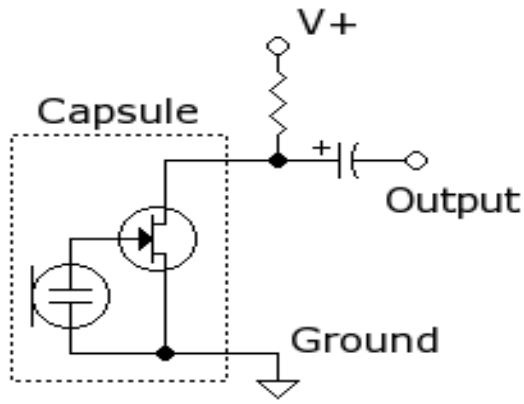


Figura 5.2. Diagrama de conexión para caracterización del micrófono.

donde: capsule (encapsulado), output (salida), groun (tierra).

Y donde los valores de los componentes son:

- Resistor de 2.2 KΩ.
- Capacitor de 1 μf.
- Micrófono electret.

Tabla 5.1.Valores obtenidos a diferentes frecuencias con micrófono electret.

Frecuencia (Hz)	Vsal. (mV)
20	76
40	80
100	76
200	76
300	76
400	76
500	100
600	100
700	100
800	100
1000	100
2000	100
3000	120
4000	110
5000	100
6000	90
7000	90
8000	80
9000	80
10000	80
11000	76
12000	76
13000	76
14000	76
15000	76
16000	76
17000	76
18000	76
19000	76
20000	76

Graficando los datos obtenidos (figura 5.2), se observa que si hay una respuesta a las frecuencias del habla, por lo cual se procede a utilizar este micrófono para el desarrollo del presente proyecto de tesis, en la figura 5.3 se muestra la respuesta en frecuencia.

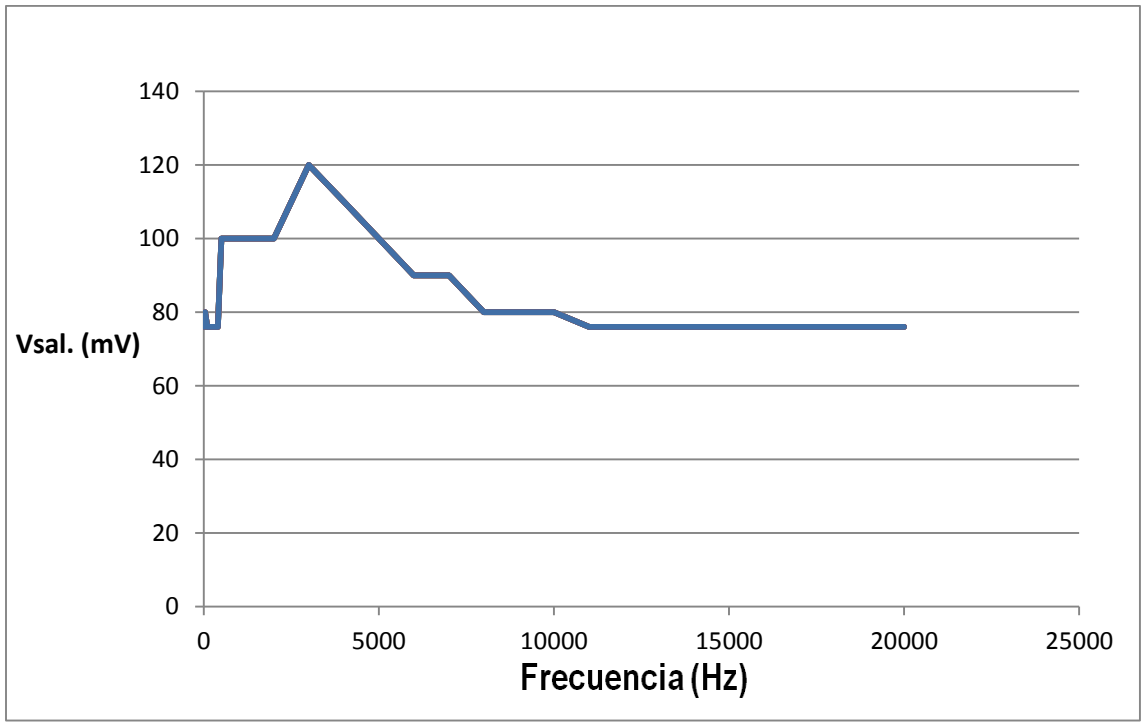


Figura 5.3 grafica que muestra la respuesta en frecuencia de el micrófono.

5.2 Adquisición de la señal de voz.

Para dar eficiencia al programa, existe un botón que el usuario puede activar para realizar el reconocimiento de una palabra y un botón que finaliza el mismo. Para hacer este proceso posible, todo el algoritmo se encuentra dentro de un ***Case estructure*** (Estructura de Caso).

Comenzamos con la adquisición de la señal de voz. Ya que la duración promedio de los comandos de voz de las palabras **izquierda, derecha, encender y apagar** es de 2.5 segundos, la adquisición de señal se realiza durante 2700 ms para hacer finita la señal de voz y abarcar el tiempo de pronunciación de cada palabra. Se utilizan indicadores para que el usuario sepa cuándo comienza y termina la adquisición de señal.

Para tener un sistema de referencia, se utiliza un bloque que proporciona Labview llamado ***Flat Sequence Structure*** (estructura de secuencia plana) como lo muestra la figura 5.4, este bloque ejecuta secuencialmente los comandos dentro de cada subdiagrama.

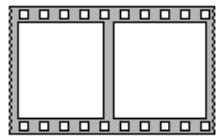


Fig. 5.4. Diagrama bloque Flat Sequence Structure

Dentro de la secuencia de ***Flat Sequence Structure***, se inserta un contador llamado ***Wait* (ms) *Function*** (función de espera), el cual realiza el conteo por 300 ms y 600 ms según se necesite en cada trama de la etapa de adquisición, en la figura 5.5 se muestra el diagrama bloque de esta función.

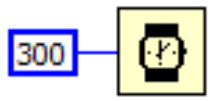


Fig. 5.5. Diagrama bloque *Wait (ms) Function para un tiempo de 300 ms.*

Como se ha dicho anteriormente, se utilizan indicadores conectados a una constante ***True*** (verdad) que es equivalente a un 1 lógico, y permanecen activados el tiempo de ejecución de cada subdiagrama donde se encuentren y continúan así hasta que algún otro valor altere su estado. En la figura 5.6. se muestra el diagrama bloque.

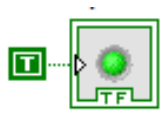


Fig. 5.6. Diagrama bloque de una constate *true.*

Entonces los indicadores irán activándose así como las tramas de la estructura ***Flat Sequence Structure*** se vallan ejecutando**.**

Después de haberse ejecutado todos los subdiagramas donde se activaron los indicadores y transcurrió el tiempo de adquisición de señal, se desactivan todos los indicadores para que el usuario sepa que el tiempo de adquisición ha transcurrido. Para realizar esta operación se devuelve una constante ***False*** (falso) la cual es equivalente a un 0 lógico, a

cada uno de los indicadores por medio del empleo de **Variables Locales.** Las Variables Locales representan objetos que han sido colocados en la programación de Labview, pueden recibir información y devolverla al objeto que estén representando. En la figura 5.7 se muestra el diagrama bloque completo.

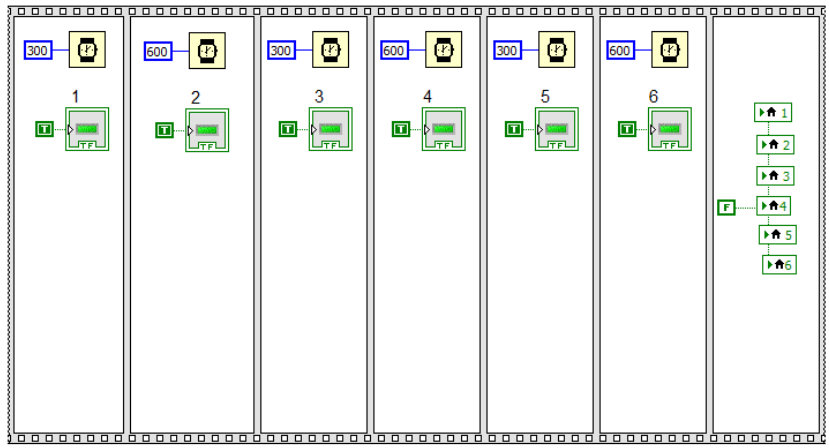


Fig. 5.7. Diagrama bloque completo para todos los indicadores.

Para hacer la adquisición de la señal de voz se utiliza un bloque llamado ***Acquire Sound Express VI*** (adquisición exprés de sonido) Figura 5.8. el cual es ejecutado mientras se ejecutan las instrucciones para indicar el tiempo de adquisición de señal. En este bloque se caracteriza la señal de voz adquirida dándole una duración de 2700 ms por lo ya mencionado anteriormente. Como las frecuencias del habla humana van desde los 100 Hz hasta 8 KHz y haciendo uso del teorema de Nyquist podríamos utilizar una frecuencia de muestreo de 16 KHz, mas sin embargo para captar la mayoría de las frecuencias audibles se utiliza el rango de audición humana que va desde los 20 Hz hasta los 20 KHz, la frecuencia de muestreo seria de 40 KHz con esto aseguramos todas las frecuencias audibles, haciendo un ajuste para frecuencias estándar en formato WAV. se utiliza la frecuencia de muestreo de 44,100 Hz. También es de un solo canal, y la resolución de 16 bits, utilizado comúnmente en un cd. de audio.

Fig. 5.8

El proceso anterior se encierra en otra ***Flat Sequence Structure*** (estructura de secuencia plana), para dar orden al proceso de adquisición y análisis de voz, quedando como se muestra en la figura 5.9.

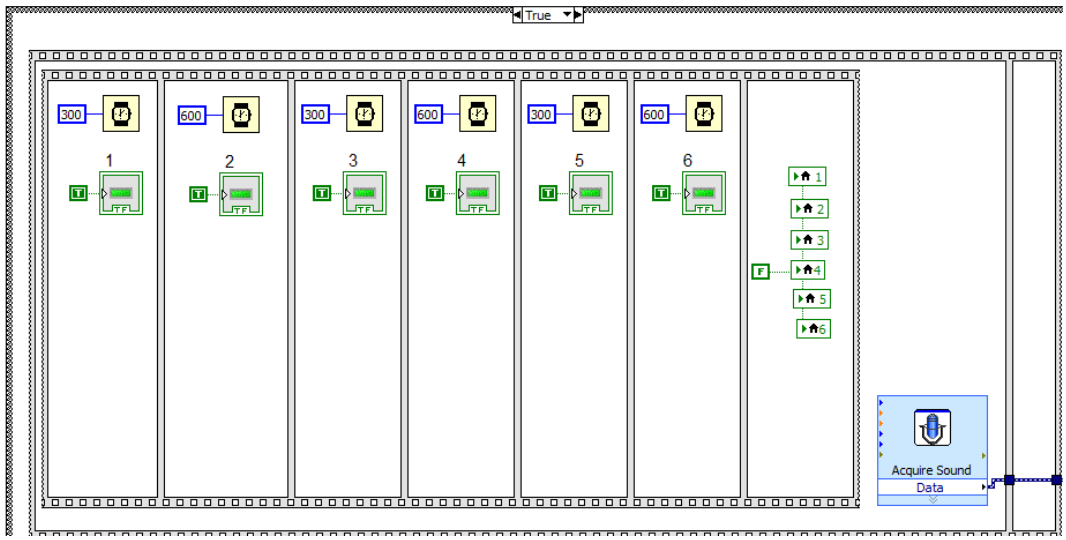


Figura 5.9. Diagrama bloque para un tiempo de referencia de 2700 ms durante la adquisición de voz.

5.3 Normalización.

Para poder trabajar con la señal se transformarla en una forma de onda, para esto se utiliza un bloque llamado ***Convert from Dynamic Data Express VI***, que transformara la señal en una forma de onda gráfica como lo muestra la figura 5.10.

Figura 5.10. Diagrama bloque para ***Convert from Dynamic Data Express VI.***

Antes de que la señal comience a ser analizada la normalizamos en un rango de 1 a -1 para tener un mayor control sobre los niveles de amplitud que la señal pueda tener, con el bloque ***Normalizate waveform*** ver figura 5.11.

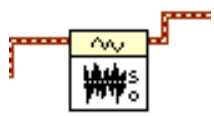


Figura5.11. Diagrama bloque ***Normalizate waveform.***

Para el siguiente paso se utiliza una propiedad muy útil de LabView, la creación de ***SubVI's,*** un *SubVI* es un función de Labview diseñada por el usuario y cumple con la

finalidad que este le haya dado. Es muy útil para recuperar espacio en la pantalla y organizar los procesos dentro de un algoritmo muy extenso.

Dentro del siguiente SubVI ver figura 5.12. se encuentra el proceso de análisis de la señal.

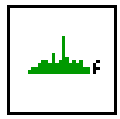


Figura 5.12. Diagrama bloque *del* ***subVI*** *creado.*

El primer paso a realizar dentro del primer *SubVI* es obtener los componentes de nuestra forma de onda con el bloque ***Get Waveform Components*** (componentes de una forma de onda), donde se obtienen los componentes de la señal en forma de un arreglo lineal y el intervalo de tiempo en segundos entre datos de la forma de onda, ver figura 5.13.

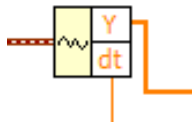


Figura 5.13. Diagrama bloque ***Get Waveform Components.***

5.4 Preénfasis.

Se realiza preénfasis en la señal aplicando un filtro FIR, (filtro de respuesta al impulso por sus siglas en ingles), como su nombre lo dice al tener en la entrada un impulso, este filtro lo delimita a valores finitos dando ganancia a las altas frecuencias y atenuando las bajas frecuencias para así tener una señal más uniforme. La figura 5.14. muestra el diagrama bloque de un filtro FIR.

Figura 5.14. Diagrama bloque para el filtro FIR.

5.5 Muestreo.

Muchos sistemas de reconocimiento de voz emplean como unidad básica del habla al fonema o la silaba, segmentando su señal en función de estas unidades. Pero en el ´presente proyecto se propone una segmentación arbitraria. Se trata a la señal como una señal analógica cualquiera segmentándola en tramas de 20 ms con un 25 % de

traslape, utilizando ventana de Hamming para reducir el error que pueda surgir en los traslapes entre segmentos. Para realizar este ventaneo, primero se obtiene el número de elementos que componen la señal por segundo con el bloque ***Array Size Function*** ver Figura 5.15, en el cual se guardan los valores que entrega el filtro y a su salida proporciona el número de elementos en cada dimensión.

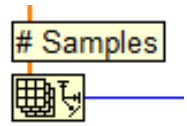


Figura 5.15. Diagrama bloque ***Array Size Function***.

Se multiplica el número de muestras por segundo por el periodo de la señal, obtenido anteriormente, después se obtiene el número de tramas dividiendo entre el tamaño propuesto que fue de 20 ms. Como se propone un traslape del 25%, se multiplica por 1.25 para obtener el número de tramas necesario considerando dicho traslape.
Para obtener la longitud de la trama se divide el tamaño propuesto de la trama entre el periodo de la señal, ver figura 5.16

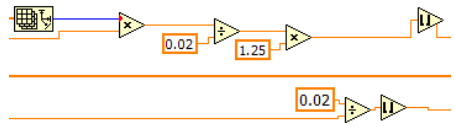


Figura 5.16.. Diagrama bloque algoritmo para obtener el número de tramas por segmento.

5.6 Ventaneo.

Después se introduce a un ciclo FOR que ejecuta el número de tramas calculado previamente, ya adentro se multiplica el tamaño de la trama por 0.75 (por el traslape de 25%) y se multiplica por el número de iteración, después se introduce en la terminal *Index* del bloque ***Array Sub Set***, esto le da la posición que se necesita a cada trama dentro del *array* final. La longitud que se calculó para cada trama se introduce en la termina *Lenght* para ir dando el tamaño a nuestras tramas. Cada trama se multiplica por una ventana de Hamming, para reducir el error que pueda surgir entre traslapé ver figura 5.17.

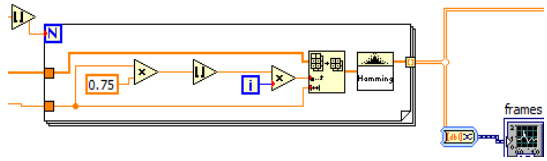


Figura 5.17. Diagrama bloque donde se aplica la ventana haming.

Para no ocupar mucho espacio dentro del primer SubVI y optimizar la programación se utiliza otro SubVI ver figura 5.18, donde se calculan los componentes cepstrales de la señal ya segmentada.

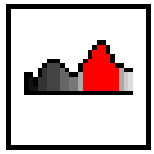


Figura 5.18 Diagrama bloque del subVI creado.

Dentro de este SubVI se obtienen las características principales de cada segmento de voz, calculando los componentes cepstrales de cada uno. Pero solo se utiliza los primeros 15 coeficientes de cada trama, siendo suficiente para caracterizarlas.

5.7 Obtención de coeficientes cepstrales.

Este proceso es encapsulado en un SubVI, ya que se tiene la señal segmentada en tramas se convierte en matriz y se obtiene el número de Filas, para que el proceso se ejecute ese número de veces. Se toman cada una de las tramas y se hacen pasar primero por ***Convert from Dynamic Data Express VI*** para trasformar los datos y puedan ser utilizados por el bloque **Real Cepstrum**, que calcula el cepstrum de cada trama, a este bloque se le especifica el número de puntos donde se debe realizar el cálculo, aunque se utilizan solo 10 es conveniente darle un valor elevado para tener más resolución en el cálculo; también se le especifica el método para realizar el cálculo y el tipo de ventana a utilizar, como ya se hizo un ventaneo, posteriormente no es necesario utilizar una, después con el bloque ***Unbundle by Name***, obtenemos el componente resultante que representa el cepstrum calculado, después se introduce en el bloque ***Array Sub Set.*** Como solo se ocupan los primero 10 coeficientes **se** le indica en la terminal ***Index*** que comience siempre en el primer término y en la terminal ***Length*** se le indica que utilice solo 10 datos, que serán los coeficientes que se necesita.

Los coeficientes obtenidos se introducen al bloque ***Build Array*** para ir construyendo un nuevo arreglo que contenga los coeficientes de cada trama juntos, en una matriz donde cada fila corresponda a una trama y cada columna a un coeficiente cepstral. El arreglo resultante se introduce los ***Shift Registrer*** del ciclo For que almacenaran el valor entregado y lo regresaran en la iteración siguiente, este valor se introduce de nuevo en ***Build Array*** para ir formando la matriz de coeficientes.

Se conecta una constante con cero en los ***Shift Registrer*** para borrar todos los datos cuando se hayan terminado las iteraciones. En la figura 5.19 se observa este proceso.

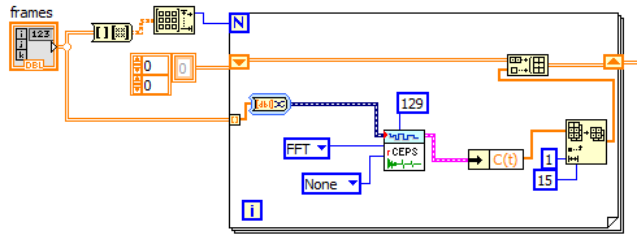


Figura 5.19. Diagrama bloque para una matriz de registros.

Después se normalizan los componentes de cada trama para controlar los niveles de cada coeficiente y evitar errores en el sistema. A cada uno de los vectores se le resta la media tipificando por tanto cada una de las componentes cepstrales.

Se calcula la media haciendo un promedio por columna, cuando la matriz se introduce en el ciclo FOR las filas se convierten en columnas y las columnas en filas, por lo que para hacer el promedio por columna primero se Transpone, luego se introduce columna por columna en el ciclo, y se obtiene el promedio sumando todos los coeficientes y dividiendo entre el número de filas.

Después se le resta el promedio a la matriz de coeficientes columna por columna. Este proceso se observa en la figura 5.20.

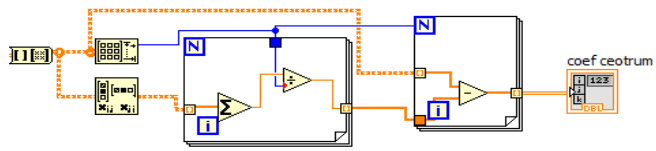


Figura 5.20. Diagrama bloque para la obtención de la media.

El proceso completo dentro de este SubVI se observa en la figura 5.21.

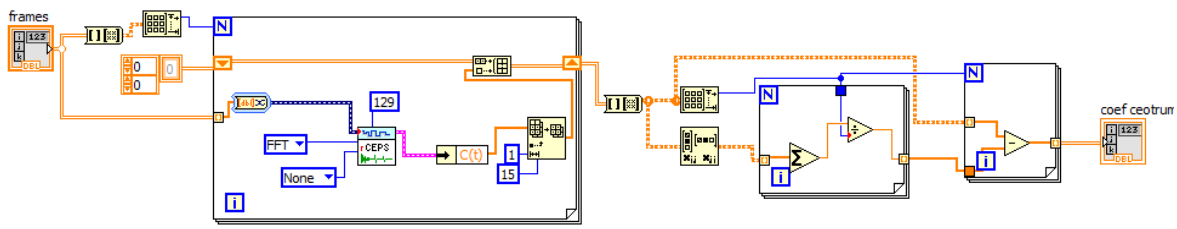


Figura 5.21. Diagrama bloque final para la obtención de los coeficientes cepstrales.

5.8 Obtención de Parámetros delta cepstrum.

Todavía dentro del primer SubVI que fue mencionado se obtiene la estimación de la rapidez de variación de la función temporal de cada parámetro, en cada segmento de cada muestra, es decir; el promedio de la función de tiempo de cada coeficiente cepstral de cada segmento en cada muestra, mediante el cálculo de la pendiente del polinomio de orden uno que mejor se aproxima a esta función temporal, lo que se conoce como Parámetros Delta Cepstrum. En esta parte es conveniente utilizar el módulo ***MathScript*** (Figura 5.22). Realizar este cálculo nos ayuda a darle mayor certeza en el reconocimiento a nuestro algoritmo, pues realiza el cálculo para la aproximación de la derivada del espectro instantáneo de la voz, siendo este más robusto frente a la variabilidad del interlocutor y del entorno. Nos proporciona mayor información sobre la señal de voz para que el algoritmo supere errores de reconocimiento debidas al hablante o el medio.

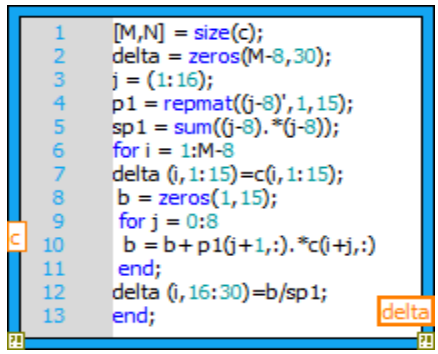


Figura 5.22. Diagrama bloque ***MathScript*** para la obtención de parámetros delta cepstrum.

Descripción del código.

"[M,N] = size(c);" :se calcula el tamaño de la matriz de coeficientes cepstrales.

"delta = zeros(M-8,30);" :se reserva espacio de memoria para realizar los cálculos del vector observación, que deberá estar formado por 15 coeficientes cepstrales y 15 parámetros delta, como marca la ecuación 4.9.

"j = (1:16);" :como la K propuesta fue de 7, entonces el cálculo se realizara en intervalos de 15 segmentos, porque se tomaron 15 coeficientes cepstrales.

"p1 = repmat((j-8)',1,15);" :hace una matriz donde se repite la columna j-8 15 veces, para hacer el cálculo desde 7 hasta -7.

"sp1 = sum((j-8).*(j-8));" :este representa el divisor de la ecuación ***4.8***, donde se hace la sumatoria de K al cuadrado y se suma el vector j-8 multiplicándose por sí mismo en cada elemento.

"for i = 1:M-8" :se inicia un ciclo FOR para las filas.
"delta (i,1:15)=c(i,1:15);" :se igualan los primeros 15 términos de cada fila a los coeficientes cepstrales.

"b = zeros(1,15);" :se reserva espacio para el cálculo de las deltas siguientes en este vector.

"for j = 0:8" se inicia un ciclo FOR para las columnas.

"b = b+ p1(j+1,:).*c(i+j,:);" :se realiza la parte del dividendo de la ecuación 4.8, donde se multiplica cada elemento de k (desde -7 hasta 7, realizado con la matriz p1) por cada coeficiente cepstral realizando un desplazamiento en cada iteración.

"end;" :fin del primer ciclo.
"delta (i,16:30)=b/sp1;" :se termina la ecuación realizando la división de las deltas entre el cuadrado la sumatoria del cuadrado de cada elemento de K y se iguala a los últimos 15 elementos de cada fila en la matriz que fue reservada al inicio. Para completar el vector observación como marca la última ecuación.

"end;" Fin del programa.

Quedando como lo muestra la figura 5.23 la programación dentro del primer SubVI que fue mencionado:

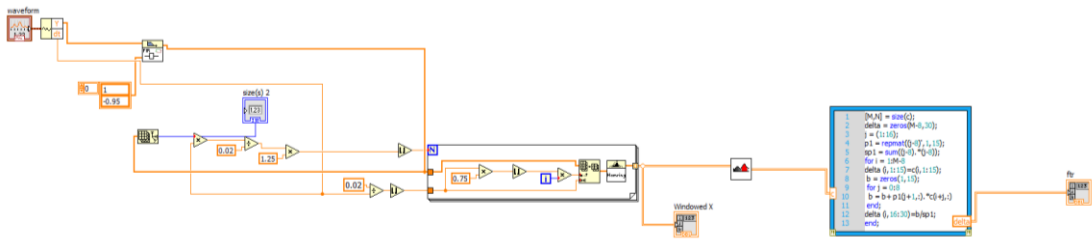


Figura 5.23 Diagrama bloque completo de la etapa análisis de la señal de voz.

Para darle mayor resolución al algoritmo, se incluyeron 5 muestras diferentes de cada una de las cuatro palabras que componen el vocabulario de la base de datos. Estas muestras están guardadas en formatos **UFF**, a una frecuencia de muestreo de 22,050 Hz, 16 bits de resolución y a un sólo canal.

5.9 Creación de la Base de Datos.

Para realizar la lectura y procesamiento de los archivos que componen la base de datos se utiliza el siguiente SubVI ver figura 5.24.

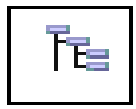


Figura 5.24 Diagrama bloque del subVI creado para lectura y procesamiento.

Donde los archivos de la base de datos son leídos por medio del bloque ***Load From UFF*** **(**lectura desde un archivo UFF**)**, donde sólo le especifica la dirección donde está almacenado el archivo, y después se le aplica el procedimiento que fue descrito anteriormente para obtener las características de la señal adquirida por el micrófono. El diagrama de conexión se observa en la figura 5.25.

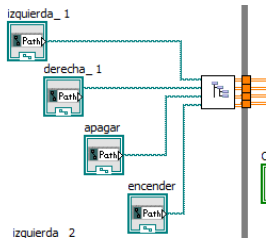


Figura 5.25. Diagrama bloque para la lectura de cada una de las palabras.

Quedando como lo muestra la figura 5.26 programación dentro de este SubVI:

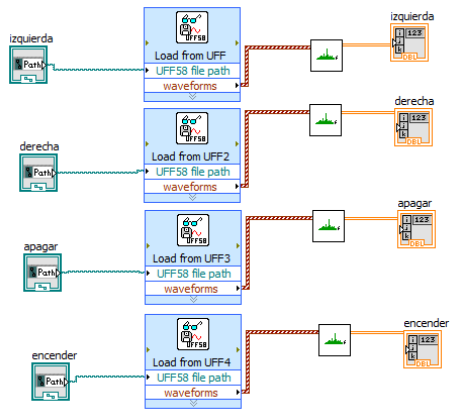


Figura 5.26. Diagrama bloque para caracterizar cada una de las palabras guardadas.

La base de datos fue grabada en la cámara sub-amortiguada de la ESIME Zacatenco, para evitar fuentes de señales no deseadas en las grabaciones de audio, con un micrófono electret, conectado a la tarjeta de audio de una computadora portátil, laptop. Se adquiere

señal durante 2700 ms, a una frecuencia de muestreo de 22,050 Hz, con 16 bits de resolución y a un solo canal, como se ha dicho anteriormente. Al iniciar el tiempo de reconocimiento se enciende un indicador durante 500 ms, para que el usuario se prepare para hablar, después se enciende un segundo indicador, que dura 2,700 ms, en ese transcurso de tiempo se ejecutan los bloques ***Acquire Sound Express VI*** (adquisición exprés de sonido), que registra la señal del micrófono con las características que se han mencionado anteriormente, de la señal que este bloque captura se obtiene una gráfica para que el usuario vea la señal adquirida, y también se manda a la entrada del bloque ***Save to UFF*** (guardar como UFF), que guardara la señal adquirida como un archivo de formato UFF, al cual sólo se le especifica la dirección donde se almacenara el archivo. Después se enciende un tercer indicador por 500 ms, para que el usuario sepa que el tiempo de adquisición ha terminado. Por último se desactivan todos los indicadores por medio del uso de Variables Locales. Todo este proceso se puede ver en la figura 5.27.

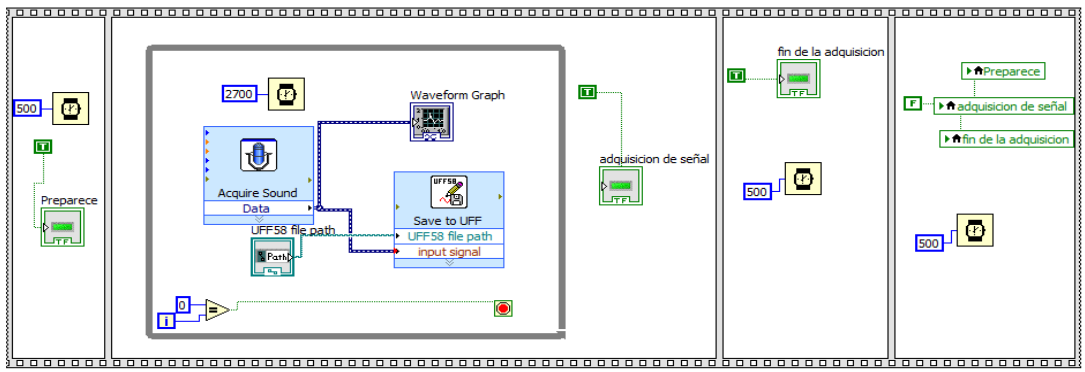


Figura 5.27. Diagrama bloque del Instrumento Virtual usado para grabar la base de datos.

El Panel Frontal de este Instrumento Virtual está conformado por 3 indicadores colocados en la parte superior central, el primero concientiza al usuario que el tiempo de adquisición esta próximo, el segundo indica que el tiempo de adquisición está transcurriendo y el tercero indica que el tiempo de adquisición ha transcurrido. En la parte central se encuentra una gráfica de la señal adquirida. En la parte inferior central se encuentra un cuadro de dialogo donde se puede especificar la ruta donde deseamos guardar el archivo UFF. Ver figura 5.28.

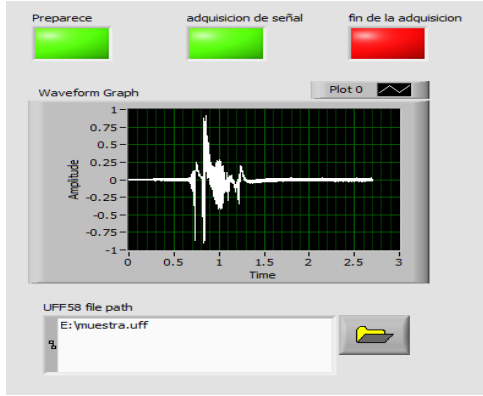


Figura 5.28. Panel Frontal del Instrumento Virtual usado para grabar la base de datos.

5.10 Distancias euclidianas y mínima distorsión temporal.

Después se inicia el proceso de comparación de la señal adquirida con todas las muestras de las palabras contenidas en la base de datos. Este proceso también fue contenido en un SubVI, que se aprecia en la figura 5.29, para evitar que el código fuese muy extendido.

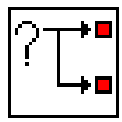


Figura 5.29. SubVI creado para el proceso de comparación entre señal adquirida y señal guardada.

Dentro de este SubVI se utiliza otro para comenzar el proceso de comparación que aparece en la figura 5.30.

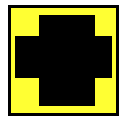


Figura 5.30. SubVI creado para el inicio del proceso de comparación.

Dentro de este SubVI se calcula primero la distancia euclidiana, que fue explicada en la ecuación 4.10, entre los coeficientes de la señal adquirida y cada una de las muestras de las palabras. Como la voz es muy variante en el tiempo y la pronunciación de una palabra depende varios factores como el estado de ánimo y la velocidad de pronunciación, los fonemas que componen una palabra no siempre son iguales cuando se pronuncia en repetidas ocasiones, por lo que no es suficiente calcular la distancia euclidiana entre los coeficientes de la señal adquirida y las muestras de la base de datos, se necesita calcular la Menor Distorsión Temporal, para obtener la distancia optima entre los resultados de las distancias euclidianas de los coeficientes de la señal

adquirida y la muestra de voz. En este punto también fue necesario utilizar el bloque ***MathScript Node*** (Nodo de código matemático) para implementar código de MathLab y realizar el cálculo de la Menor Distorsión Temporal, como está indicado en las ecuación 3.18. Ver figura 5.31.

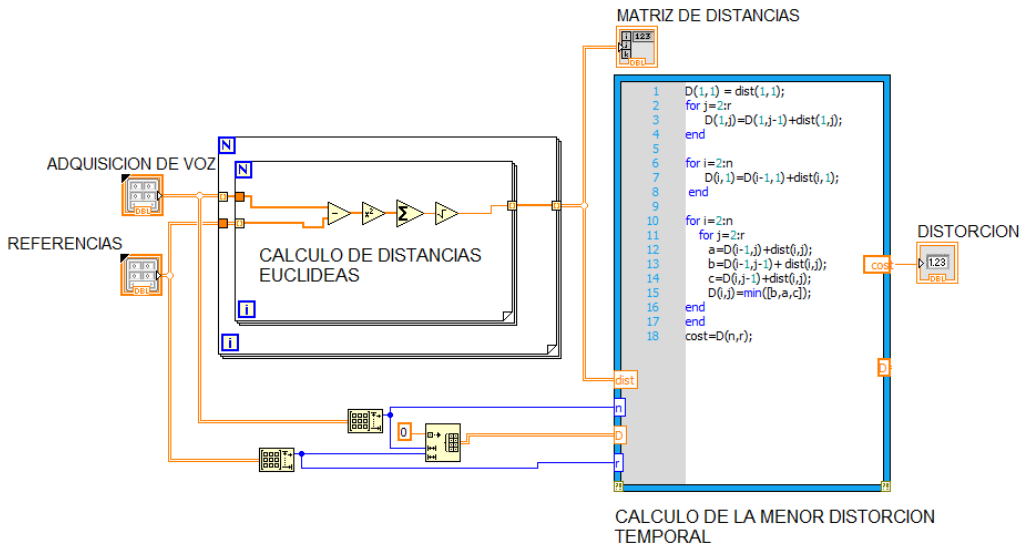


Figura 5.31. Calculo de distancias euclidianas y menor distorsión temporal.

Explicación del código.

"D(1,1) = dist(1,1);" :iguala el primer término de la matriz del vector observación con el primer término de la matriz reservada para establecer las condiciones de control.

"for j=2:r" :como ya se mencionó un camino que une dos puntos X y Z pasando por una serie de puntos intermedios es óptimo según determinado criterio, entonces al subdividirlo en dos tramos XY e YZ cada uno de ellos también es el camino óptimo entre sus respectivos extremos, primero se calcula el costo para encontrar la mínima distancia entre los supuestos puntos Y y Z.

"**D(1,j)=D(1,j-1)+dist(1,j);"** :primero entre columnas.

"end" :fin.

"for i=2:n" :inicia ciclo FOR.

"D(i,1)=D(i-1,1)+dist(i,1);" :después entre filas para realizarlo entre cada elemento de ambas matrices.

" **end"** :fin.

"for i=2:n" :inicia ciclo FOR**.**

"for j=2:r" :luego se calcula el costo para encontrar la mínima distancia entre los supuestos puntos X y Y pasando por Z, entre los coeficientes antes, durante y después de la iteración en proceso.

"a=D(i-1,j)+dist(i,j);"

" b=D(i-1,j-1)+ dist(i,j);"

" c=D(i,j-1)+dist(i,j);"

" D(i,j)=min([b,a,c]);" :se elige la menor distancia que se haya encontrado.

"end" :fin.

"end" :fin.

"cost=D(n,r);" :regresa el valor obtenido.

5.11 Calculo de máximos y mínimos.

Después de tomar este paso, el valor de la Distorsión Temporal pasa por otro SubVI (figura 5.32.) donde se compara con un valor máximo y mínimo, los cuales fueron obtenidos realizando repeticiones de la palabra correspondiente para observar rango de valores que pueden tomar y así dar mayor certeza al algoritmo.

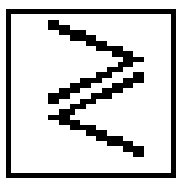


Figura 5.32. SubVI para cálculo de máximos y mínimos.

En la figura 5.33 se aprecia la programación utilizada para este paso, donde sólo se utilizaron un par de comparadores, una compuerta And y una secuencia Case Estructure (Estructura de Caso).

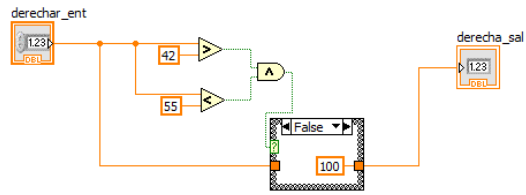


Figura 5.33 diagrama bloque para obtención de máximos y mínimos.

Se repite el proceso para comparar cada muestra de la base de datos con la señal que fue obtenida del micrófono.

5.12 Comparación y toma de decisiones.

Después de haber obtenido el valor de la menor distorsión lineal se comparan las distorsiones lineales obtenidas, la palabra con la menor distorsión lineal es la que se identifica como la palabra dicha por el usuario, ver figura 5.34.

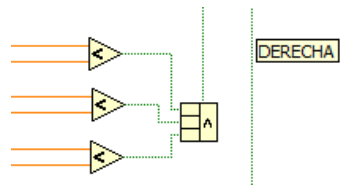


Figura 5.34. Diagrama de conexión para comparación de distorsión lineal.

Quedando como lo muestra la figura 5.35 la programación utilizada en el proceso de comparación entre la señal obtenido por el micrófono y la base de datos:

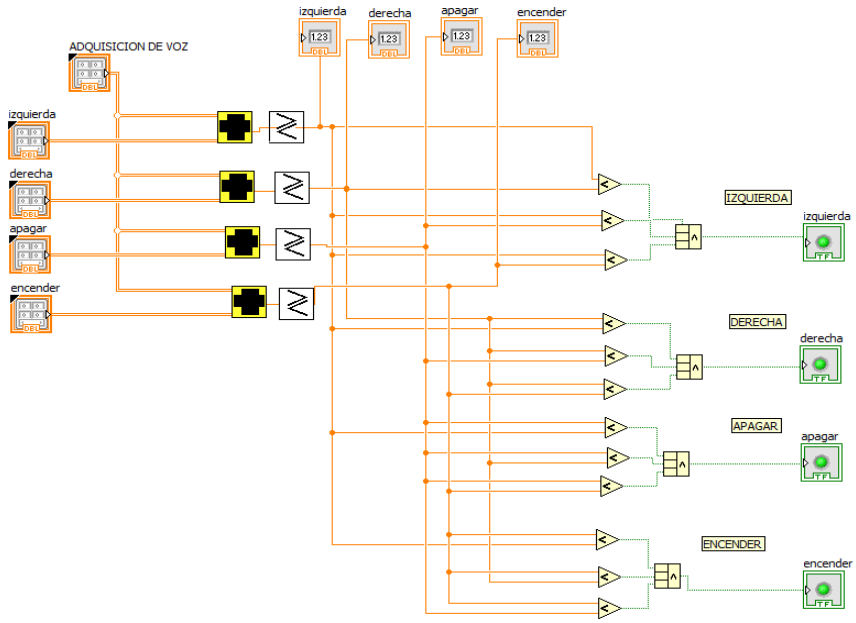


Figura 5.35. Diagrama bloque completo de proceso de comparación.

Hasta ahora sólo se ha explicado el proceso para comparar la señal adquirida por el micrófono con un grupo de palabras de la base de datos, como se dijo anteriormente, la base de datos está compuesta por 5 grupos de palabras para dar mayor resolución. El proceso antes expuesto se repite para los 5 grupos de palabras.

Después de que el algoritmo calcula, de cada grupo de palabras, cual fue la más parecida a la adquirida por el micrófono, se calcula cuantas repeticiones obtuvo cada palabra, con el siguiente SubVI que se muestra en la figura 5.36.

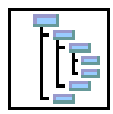


Figura 5.36 diagrama bloque para el nuero de repeticiones.

Donde por medio del uso del bloque ***Case Structure*** se suma un 1 cada vez que la palabra se repite en una de las 5 comparaciones, la programación de este SubVI se indica en la figura 5.37.

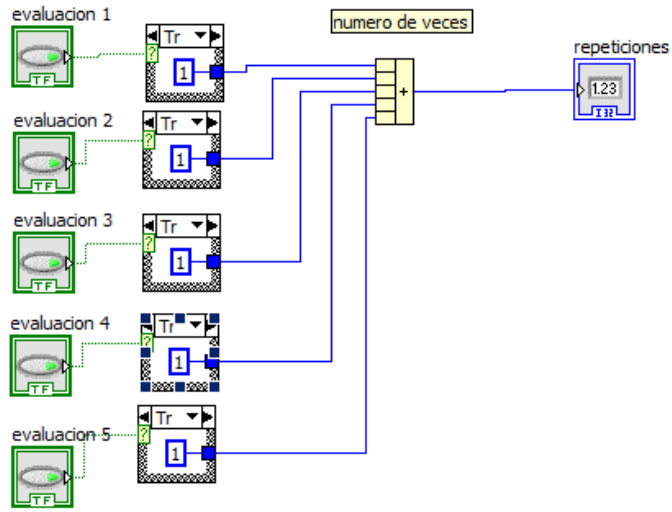


Figura 5.37. Diagrama bloque para la suma del nuero de repeticiones.

Se calcula el número de repeticiones en cada grupo para cada palabra. Luego se calcula cual fue la que recibió más repeticiones, por medio del siguiente SubVI mostrado en a figura 5.38.

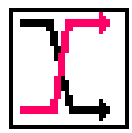


Figura 5.38. SubVI para el número de veces que se repite la palabra.

Donde sólo se compara el número de repeticiones que obtuvo cada palabra por todos los grupos, la palabra que obtuvo mayor repetición es la que será seleccionada como la palabra dicha por el usuario. Quedado como lo muestra la figura 5.39 la programación dentro de este SubVI:

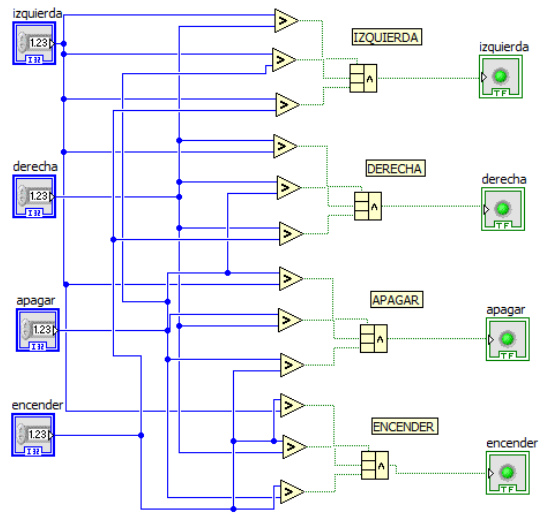


Figura 5.39. Diagrama bloques de la programación para el número de veces que se repite la palabra.

Por último, para que el usuario sepa que palabra fue reconocida, se utiliza el bloque ***One Button Dialog,*** este bloque despliega en la pantalla un mensaje. Para realizar su función, se le conecta una constante que indica el texto a mostrar y puede ser activado mediante el uso de un **Case *Structure***.

Para que el algoritmo funcione solo cuando el usuario lo desea, se encierra todo el proceso dentro de un ***Case Structure,*** el cual se activa por un botón que el usuario puede utilizar a placer, y se finaliza con otro botón dispuesto para el usuario. La programación interna de todo el algoritmo se muestra en la figura 5.40.

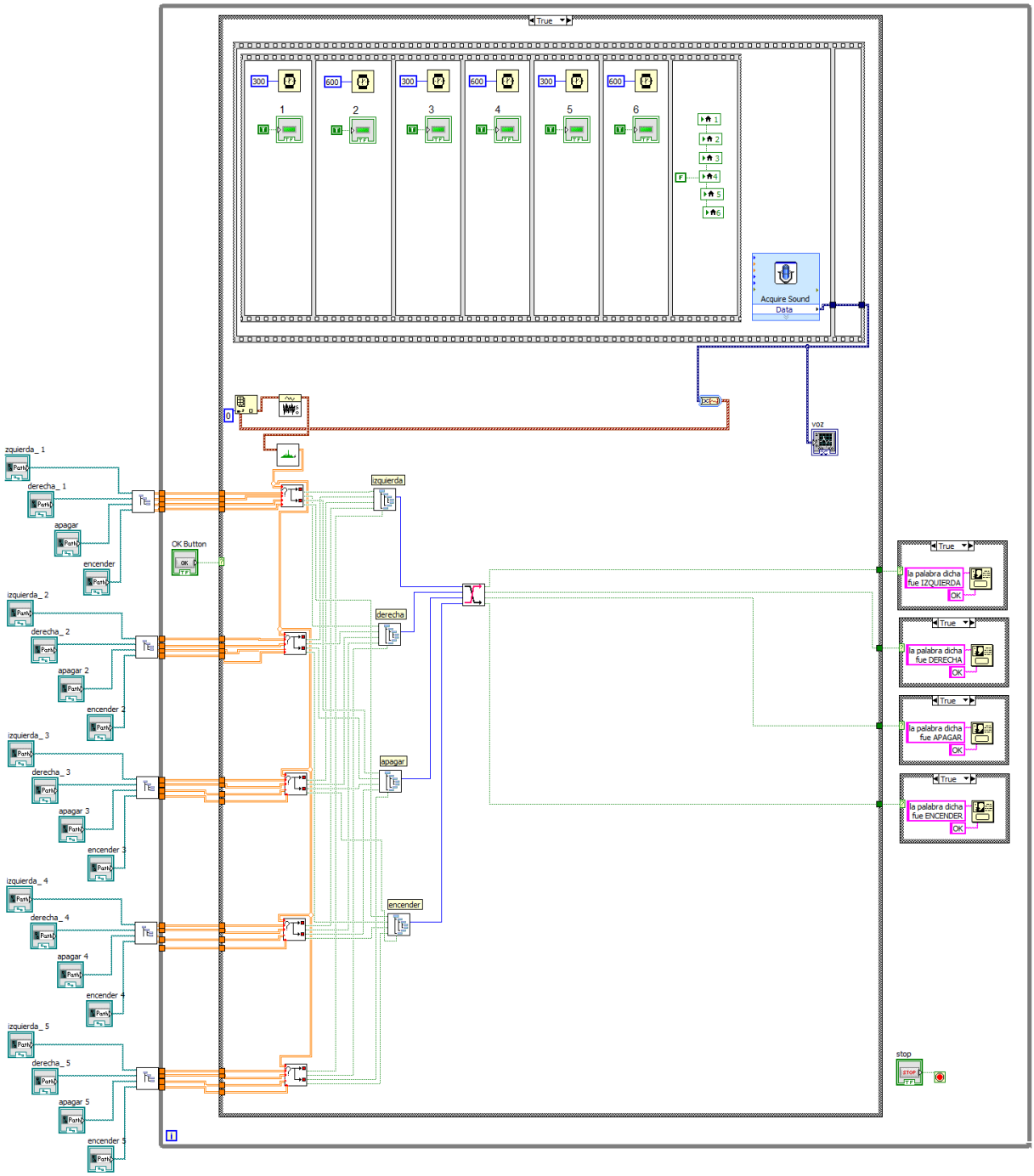


Figura 5.40. Diagrama bloques final.

A continuación se muestra el panel frontal (figura 5.41), donde se ubican en la parte superior una gráfica que muestra la señal adquirida por el micrófono, a su derecha se encuentran 2 *botonews,* uno para iniciar el reconocimiento de una palabra y el otro para detener al programa.

Debajo de la gráfica se encuentran 6 indicadores, para que el usuario sepa cuánto tiempo tiene para decir una palabra. Estos indicadores están en 3 diferentes colores, verde, amarillo y rojo, el verde indica que es una momento muy seguro para pronunciar una palabra, el amarillo indica ya no queda mucho tiempo para decir una palabra pero todavía es posible realizar una identificación y el rojo indica que ya no es muy conveniente decir una palabra pues podría no ser registrada completamente.

Debajo es estos indicadores se encuentran 20 recuadros para agregar las palabras que compondrán muestras base de datos, las palabras ENCENDER, APAGAR, IZQUIERDA Y DERECHA, fueron escogidas para componer nuestra base de datos por ser palabras muy comunes en cualquier sistema automatizado.

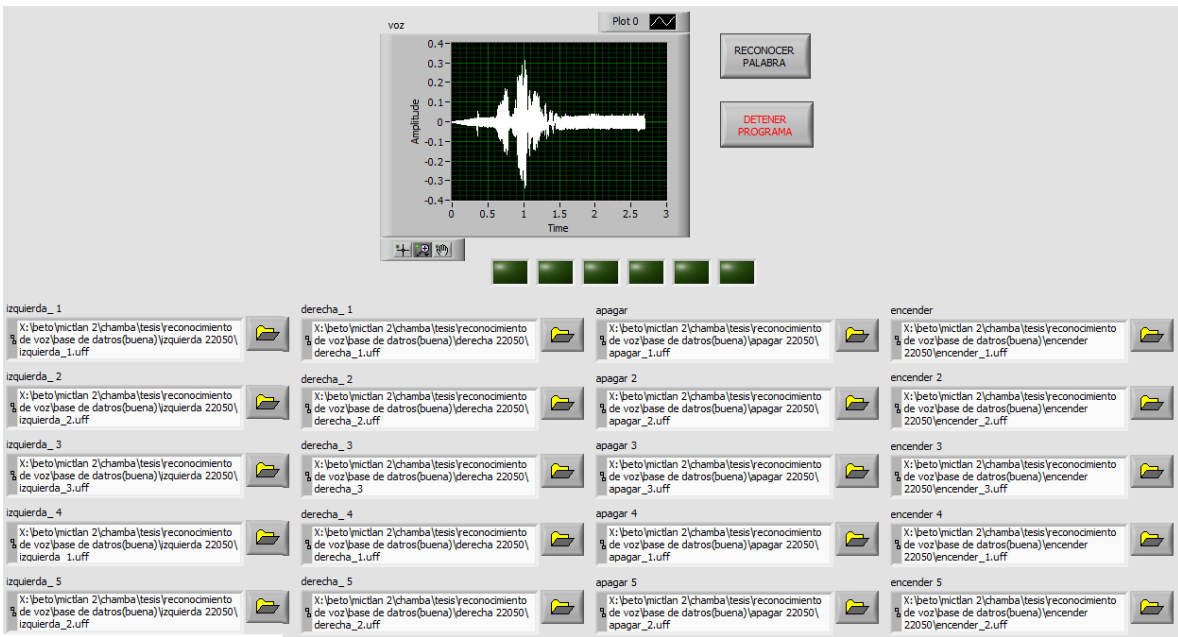


Figura 5.41 panel frontal que muestra la gráfica de la palabra adquirida y la base de datos.

A continuación se muestra el funcionamiento de la herramienta virtual para cada una de las palabras ver figuras 5.42, 5.43, 5.44, 5.45.

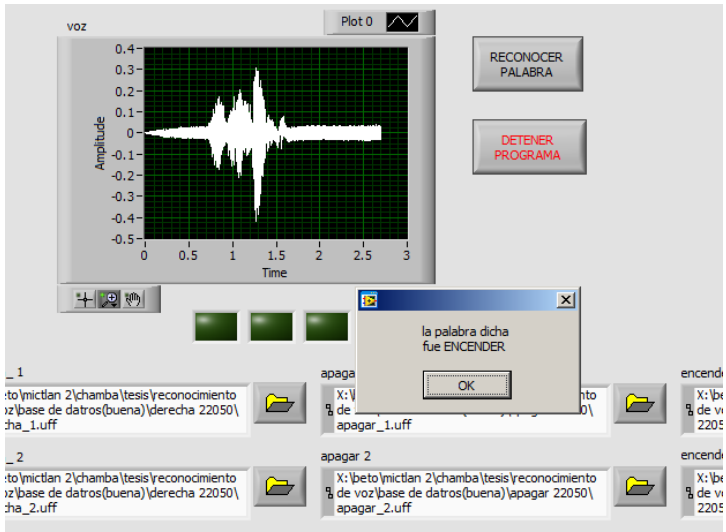


Figura 5.42 Panel frontal que muestra la gráfica de la palabra encender.

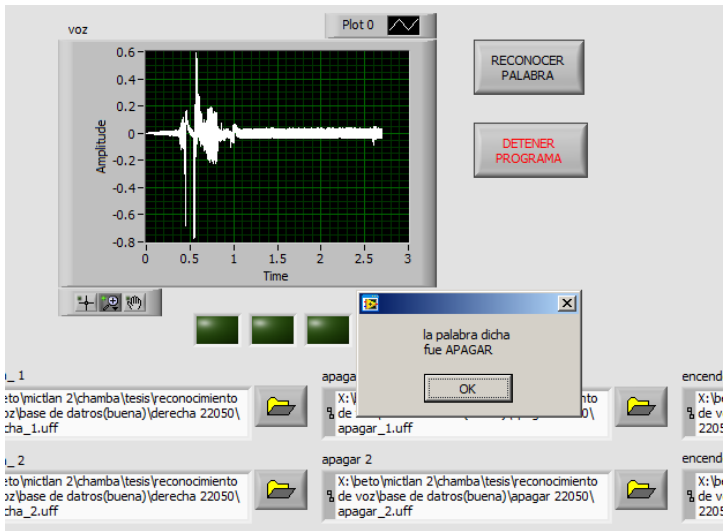


Figura 5.43 Panel frontal que muestra la gráfica de la palabra apagar.

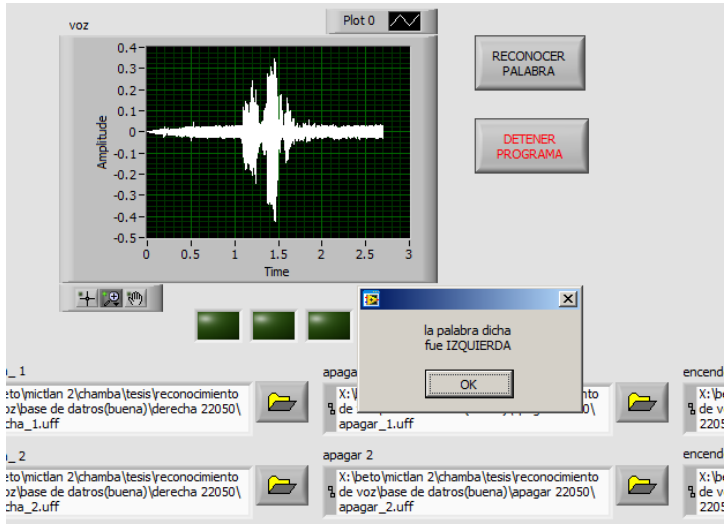


Figura 5.44 Panel frontal que muestra la gráfica de la palabra izquierda.

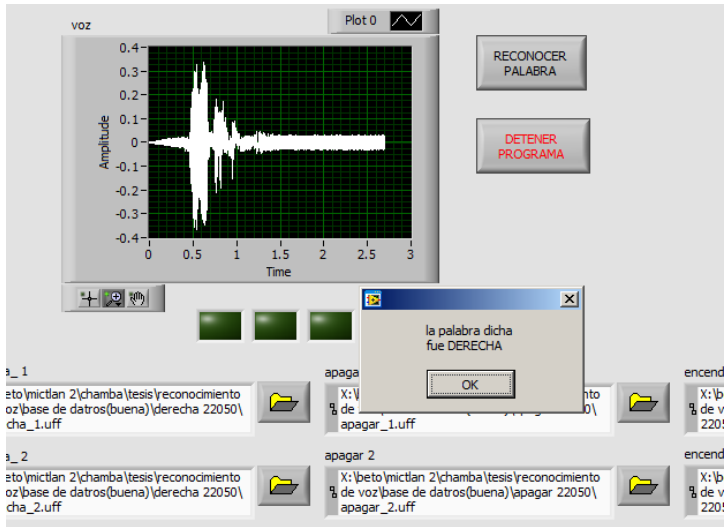


Figura 5.45 Panel frontal que muestra la gráfica de la palabra derecha.

CONCLUSIONES

El Análisis de la voz requiere herramientas con una gran capacidad de procesamiento debido a su característica cambiante, ya que la señal que representa una palabra siempre es diferente sin importar las veces que se repita. Labview resulta ser una herramienta muy poderosa en el análisis de señales, gracias a su versatilidad y su compatibilidad con otras herramientas como Mathlab. Este trabajo demuestra que el desarrollo de herramientas virtuales puede ser una alternativa recomendable a la falta de sistemas de control reales que pueden ser muy costosos, logrando así un importante ahorro y la posibilidad de adaptar o cambiar las herramientas a las necesidades de los usuarios.

Se comprobó que para el análisis de la voz tomar como unidad de análisis, muestras de 20 ms de la señal también es una alternativa, pues las técnicas modernas de reconocimiento de voz emplean como unidad mínima del habla el fonema y la silaba, basando sus algoritmos en la separación y reconocimiento de cada una, exigiendo del software mayores recursos y tiempo de procesamiento.

Aunque el sistema propuesto utiliza algoritmos matemáticos complejos resulta ser fácil de entender y puede ser utilizado en otras aplicaciones, como el reconocimiento de alteraciones en el comportamiento del ritmo cardíaco en las personas, pues las personas que presentan deficiencias cardíacas como soplos en el corazón tienen un patrón de ritmo cardiaco diferente a lo normal.

El desarrollo de esta herramienta resulto ser difícil al principio, pues algunos de los conceptos utilizados no se dominaban, además de que el desarrollo de Sistemas de reconocimiento del habla es muy complejo, y todavia no hay mucho avance en este tema, aún en la actualidad se están desarrollando nuevos algoritmos que realicen un análisis de la voz más eficaz y rápido, por lo que encontrar información que indique que algoritmo se debe seguir fue difícil y confuso al principio.

El apoyo técnico que brinda Nacional Instruments es muy eficiente, pues cuentan con un foro público donde las personas pueden resolver dudas o consultar novedades sobre sus productos y también cuentan con las especificaciones técnicas de todos los bloques que pueden ser utilizados en la programación dispuesta al público en la misma página.

En conclusión el presente trabajo alcanzo las metas deseadas en su efectividad para el reconocimiento de palabras aisladas en medios controlados o con presencia de ruido moderado, siendo posible su aplicación en otras áreas como el control de maquinaria compleja o en la vida diaria como sistema para el control de las funciones de un automóvil

evitando que el conductor pierda concentración en el manejo del mismo para controlar otras funciones del automotor.

BIBLIOGRAFÍA.

[1] TOMASI, WAYNE, Sistemas de comunicaciones electrónicas,
PEARSON EDUCACIÓN, México 2003.

[2] COUCH, W. LEÓN, II, sistemas de comunicaciones digitales y analógicos.
Séptima edición. PEARSON EDUCACIÓN, México, 2008.

[3] SANJIT K. MITRA, procesamiento de señales digitales, un enfoque basado en computadora. Tercera edición, MCGRAW-HILL INTERAMERICANA, México 2007.

[4] BERNAL BERMÚDEZ JESÚS, BOBADILLA SANCHO JESÚS, GÓMEZ VILDA PEDRO.
Reconocimiento de voz y fonética acústica. ALFAOMEGA GRUPO EDITOR.

[5] FURUI, SADAOKI Digital Speech Processing, Synthesis and Recognition MARCEL DEKKER INC., Estados Unidos 2001.

[6] documento extraído de la web
http://www.tdx.cat/bitstream/handle/10803/6911/05_hernandoPericas_capitol_4.pdf;jsessionjse=6FA098D3BBF72FD180286539B9D29569.tdx2?sequence=5, Fecha de consulta 20/10/2013

[7] documento extraído de la web http://www.ni.com/white-paper/4278/en/#toc3, Fecha de consulta 5/11/2013.

ANEXOS

Panasonic Microphone Cartridges

Unidirectional Back Electret Condenser Microphone Cartridge

Series: **WM-55A103** (back electret type)
WM-56A103 (tail electret type)

■ Features

- Unidirectional microphone for general use

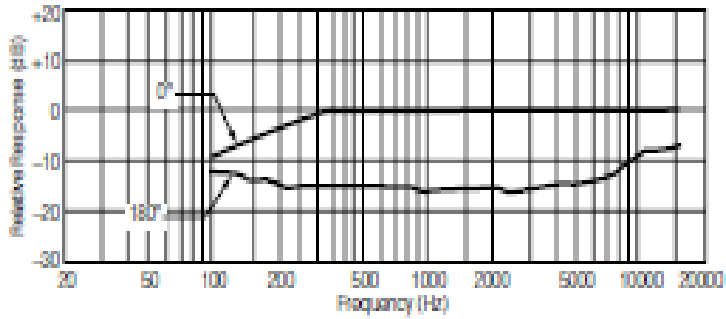
■ Sensitivity

Vs = 1.5V RL = 680Ω	WM-55A103	WM-56A103
	–47±4dB	–50±3dB

■ Specifications

Sensitivity	–47±4dB (at L = 50cm) (0dB = 1V/pA, 1kHz)
Impedance	Less than 680Ω
Directivity	Unidirectional (Cardioid)
Frequency	100–16,000 Hz
Max. operation voltage	10V
Standard operation voltage	1.5V
Current consumption	Max. 0.5 mA
Sensitivity reduction	Within –3 dB at 1V
S/N ratio	More than 60 dB

■ Typical Frequency Response Curve



■ Dimensions in mm (not to scale)

WM-55A103/56A103