RECONOCIMIENTO DE VOZ

(palabras aisladas y conectadas)

Tesis dirigida por el Ing. Leonardo Canseco Rodríguez
Director de tesis: M. Ing. Abel Herrera Camacho
México D.f. de [redacted].

1999

ÍNDICE

- ***RESUMEN***

Este trabajo se enfoca a ser una tutoría en los conceptos y teorías implícitos en los sistemas de reconocimiento de palabras, experimentalmente y en la práctica. Dos aspectos del tema tienen que tener una especial atención. Primero, el reconocimiento de voz se trata como un problema clásico del reconocimiento de patrones mostrando como algunas ideas fundamentales del procesamiento de señales, teoría de la información, y la ciencia de la computación pueden ser utilizadas para proporcionar la capacidad de reconocimiento de palabras aisladas así como del reconocimiento de sencillas secuencias de palabras conectadas. Se describe cada parte de los métodos utilizados en el análisis de la voz, y que combinados resuelven algunos de los problemas más generales en el reconocimiento de voz. En particular se muestra como esas teorías pueden ser empleadas para mejorar la exactitud del reconocimiento en un modelo de reconocimiento de patrones acústico.

El sistema de reconocimiento descrito para las palabras aisladas, es entrenado por un locutor designado, son reconocidas a través del cálculo del error cuadrático medio. El patrón de referencia de cada voz que va a ser reconocido es guardado como un patrón de coeficientes de predicción (LPC) en el tiempo. La señal de entrada es reconocida como la palabra de referencia que produjo el error cuadrático medio mínimo.

Finalmente se describe un sistema que reconoce cadenas de palabras conectadas. Este sistema es substancialmente diferente al primero ya que reconoce cadenas de palabras recitadas sin pausas entre ellas. Esto puede ser posible por una generalización importante en el procedimiento del ajuste temporal entre los patrones de voz.

En este trabajo se reporta una aproximación a la comparación de patrones utilizada en el reconocimiento de palabras aisladas, nada más que ésta se encuentra enfocada al reconocimiento de palabras conectadas. Así, el principio general utilizado en este tipo de reconocimiento, esta basado en la comparación de patrones de desconocidas secuencias de voz y patrones de referencia conformadas por palabras aisladas; todos los patrones deben estar dentro de un vocabulario previamente establecido. Entonces, se destaca que el proceso de comparación entre patrones es eficientemente llevado acabo por medio de este algoritmo en un solo paso como en el caso de palabras asiladas, aunque éste tiene que ser adaptado a las necesidades del sistema de forma recursiva.

- ***INTRODUCCIÓN***

Una gran cantidad de información se ha aprendido acerca del proceso fundamental de la reproducción de voz y de su percepción. La meta de desarrollar un reconocimiento mecánico en una platica continua parece ser efímera; Sin embargo, el reconocimiento de voz ha hecho grandes avances en la década pasada, y ésta ha avanzado hasta el punto de que existen a la disposición varios sistemas comerciales. Esos sistemas comerciales, predominantemente reconocen palabras aisladas, y son entrenados por el propio usuario, lo cual hace que la confianza del reconocimiento fluctúe por encima de un 95% en ambientes ruidosos.

Como las capacidades de los sistemas de reconocimiento han mejorado, las tareas a las cuales han sido aplicadas han venido siendo más sofisticadas, y más difíciles. Tales como, información en las aerolíneas y reservaciones, adquisición de datos y manejo de ellos, etc.

Este trabajo se enfoca a ser una tutoría en los conceptos y teorías implícitos en los sistemas de reconocimiento de palabras, experimentalmente y en la práctica. Dos aspectos del tema tienen que tener una especial atención. Primero, el reconocimiento de voz se trata como un problema clásico del reconocimiento de patrones mostrando como algunas ideas fundamentales del procesamiento de señales, teoría de la información, y la ciencia de la computación pueden ser utilizadas para proporcionar la capacidad de reconocimiento de palabras aisladas así como del reconocimiento de sencillas secuencias de palabras conectadas. Se describe cada parte de los métodos utilizados en el análisis de la voz, y que combinados resuelven algunos de los problemas más generales en el reconocimiento de voz. En particular se muestra como esas teorías pueden ser empleadas para mejorar la exactitud del reconocimiento en un modelo de reconocimiento de patrones acústico.

Los puntos a tratar en este trabajo se tratan como siguen. Se comienza con un repaso de los aspectos básicos en una señal de voz y los del sistema de reconocimiento de patrones aplicado a reconocimiento de voz. Después nos enfocamos al método de predicción lineal. También se discuten otros aspectos del paradigma del reconocimiento de patrones, incluyendo las medidas de similitud, alineamiento temporal entre los patrones de voz, y estrategias estadísticas utilizadas en las reglas de decisión. Entonces describimos dentro de este marco de trabajo el sistema básico de reconocimiento de palabras, dando algunos detalles de su implementación, operación y ejecución.

En el capítulo 3 proporciona una discusión de la aplicación de las técnicas del reconocimiento de patrones en la construcción y diseño de sistemas de reconocimiento de voz enfocadas a realizar tareas específicas.

El sistema de reconocimiento descrito para las palabras aisladas, es entrenado por un locutor designado, son reconocidas a través del cálculo del error cuadrático medio. El patrón de referencia de cada voz que va a ser reconocido es guardado como un patrón de coeficientes de predicción (LPC) en el tiempo. El total del error cuadrático medio de una señal de entrada se minimiza óptimamente registrando los LPC de referencia dentro de los coeficientes de autocorrelación de entrada usando simultáneamente el algoritmo de ajuste dinámico en el

tiempo (DTW). La señal de entrada es reconocida como la palabra de referencia que produjo el error cuadrático medio mínimo.

Finalmente se describe un sistema que reconoce cadenas de palabras conectadas. Este sistema es substancialmente diferente al primero ya que reconoce cadenas de palabras recitadas sin pausas entre ellas. Esto puede ser posible por una generalización importante en el procedimiento del ajuste temporal entre los patrones de voz.

En este trabajo se reporta una aproximación a la comparación de patrones utilizada en el reconocimiento de palabras aisladas, nada más que ésta se encuentra enfocada al reconocimiento de palabras conectadas. Así, el principio general utilizado en este tipo de reconocimiento, esta basado en la comparación de patrones de desconocidas secuencias de voz y patrones de referencia conformadas por palabras aisladas; todos los patrones deben estar dentro de un vocabulario previamente establecido. La capacidad de normalización en el tiempo es permitida por el uso del algoritmo de ajuste dinámico en el tiempo (DTW). Entonces, se destaca que el proceso de comparación entre patrones es eficientemente llevado acabo por medio de este algoritmo en un solo paso como en el caso de palabras asiladas, aunque éste tiene que ser adaptado a las necesidades del sistema de forma recursiva. Este algoritmo es utilizado extensivamente en los experimentos de reconocimiento para el presente trabajo.

Estos dos métodos (reconocimiento de palabras aisladas y conectadas) representan un avance hacia la meta final en la investigación del reconocimiento de voz, comunicación máquina/humano por medio de la voz humana. Con el pasar de los años ha parecido ser efímero la solución de esta meta. Parte de la dificultad reside en el hecho del paradigma de extrapolación en el reconocimiento de patrones, ya que no proporciona un modelo que sea lo suficientemente general en donde refleje el proceso de comunicación entre seres humanos.

1. ANÁLISIS DEL MODELO DE RECONOCIMIENTO DE PATRONES APLICADO AL RECONOCIMIENTO DE VOZ.

1.1 La señal de voz y sus características

La voz es una de las tantas características humanas que puede ser procesada y sometida a distintos análisis; ya sea, sintetizada, reconocida, codificada etc. con el fin de darle un uso provechoso utilizando como herramienta el área del procesamiento de señales. En el presente trabajo nos abocamos a la tarea del reconocimiento de voz en diferentes circunstancias, reconocimiento de palabras aisladas y conectadas; los principales problemas que se necesitan vencer en el sistema de reconocimiento son:

- La complejidad del habla así como lo ambiguo que puede resultar ésta. El vocabulario se encuentra atenida al contexto de la comunicación que se lleve acabo entre los diferentes individuos. Además de que existen palabras que se pronuncian igual como "cocer" y "coser" teniendo distintos significados, e inclusive algunas palabras pueden tener distintos significados que sólo con el contexto se sabe la correcta interpretación de ella.

- El ruido de fondo. La mayoría de las señales se contaminan comúnmente con el ruido ambiental, debido a que existe un sin número de perturbaciones que interfieren en la señal provocando su distorsión (el ruido).

- Tiempo de duración de una palabra. Las señales de voz varían con el tiempo, es decir dos señales de voz siempre serán distintas en el tiempo aunque hayan sido pronunciadas por la misma persona; cada una se pronuncia en el tiempo con distintas variaciones de tiempo en su interior.

- Interpretación. La comunicación se realiza a través de frases que comprenden varias palabras en las cuales es difícil de identificar el inicio y fin de cada una de las palabras; incluso, dentro de una misma palabra resulta difícil encontrar las fronteras entre los distintos sonidos que la conforman (sordos y sonoros).

- Visualización de la señal de voz. La representación de una señal en el tiempo es lo más básico. La forma de onda de una palabra puede ser vista en la Fig.1a esta representación será crucial en la implementación del sistema de reconocimiento, porque con ella se podrá determinar el inicio y fin de una palabra aislada o de las palabras conectadas (un conjunto de palabras).

- Existe otro tipo de análisis gráfico, debido a las variaciones que se presentan en las señales de voz. La forma es conocida como espectro en frecuencia, presentada en espectogramas. Estas gráficas representan la intensidad de la señal en un cierto ancho de banda (eje de las ordenadas) en un intervalo de tiempo (eje de las abscisas). No se pueden hacer gráficas que posean una buena resolución en cualquiera de sus representaciones debido a que el tiempo y frecuencia se relacionan de manera inversa.

Por lo que existen espectogramas de banda ancha (pobre resolución en frecuencia y una buena resolución en el tiempo) y los de banda angosta (alta resolución en frecuencia y baja en el tiempo).

Métodos en el Dominio del Tiempo

EL método utilizado en el sistema de reconocimiento que se desarrollo en el presente trabajo, procesa la señal de voz en el domino del tiempo por su fácil implementación; además, provee una base útil en la estimación de algunas de sus características importantes, tales como la frecuencia fundamental y las formantes. Así como lo relativamente fácil que resulta el separar de la señal los sonidos sordos, sonoros o silencios; aunque en algunos casos resulte ser muy complicado, por lo general se pueden realizar dichas tareas.

Las Ventanas

Una suposición básica con la que trabajan los sistemas de reconocimiento de voz es que las propiedades de la voz varían relativamente muy poco en pequeños intervalos de tiempo. Esto hace que se analice y se procese la señal en tiempo corto, es decir, la señal de voz se segmenta en muchas partes, y cada parte será aislada y procesada, considerando que en toda la trama existen las mismas propiedades, es decir la trama se supone ser estacionaria.

Esto se realiza al aplicar a las señales una ventana, el procedimiento es tomar una trama de la señal original con el que se va a trabajar y éste se multiplica en un proceso simple. Se asume entonces que la señal es igual a cero fuero del intervalo de interés.

$$x_w(n) = x(n)\, w(n) \qquad 0 \le n \le N\text{-}1 \qquad 1.1$$

Para una frecuencia de muestreo de 10khz (10 a 20 ms de duración) el orden de las ventanas es de 100 a 200 muestras.

Existen diferentes tipos de ventanas, tales como: ventana rectangular, ventana Bartlett (triangular), ventana de Hanning, ventana de Hamming y ventana de Blackman; todas diferentes y con distintas aplicaciones. En este trabajo se trabajó con la ventana de Hamming definida por

$$w(n) = 0.54 - 0.46 \cos(2\pi n / N\text{-}1) \qquad 0 \le n \le N\text{-}1 \qquad 1.2$$

Energía y Magnitud Promedio

De las gráficas en el tiempo, la amplitud de la señal varía conforme transcurre el tiempo y lo que destaca de la amplitud es que en los sonidos sonoros la amplitud será mucho mayor que para los sonidos sordos. La energía en el tiempo corto de la señal de voz provee una representación en donde se reflejan dichas variaciones Fig. 1b, la energía de una señal discreta se define como

$$E=\sum_{m=-\infty}^{\infty} x^2(m) \qquad 1.3$$

Pero esta expresión resulta de poca utilidad, debido a que la información que presenta nos dice muy poco acerca de las propiedades que dependen del tiempo de la señal. Por lo que mejor se utiliza la definición en tiempo corto.

$$E_n = \sum_{m=n-N+1}^{n} x^2(m) \qquad 1.4$$

Escribiendo de nuevo la expresión

$$E_n = \sum_{m=-\infty}^{\infty} x^2(m)\, h(n-m) \qquad 1.5$$

donde,

$$h(n) = w^2(n) \qquad 1.6$$

La señal $x^2(m)$ es pasada a través de un filtro con respuesta al impulso $h(n)$. El problema es entonces seleccionar la ventana que mejor se acople a las necesidades del sistema.

La aplicación de esta herramienta comienza cuando queremos distinguir los segmentos de voz sonora de los de voz sorda, así como distinguir los segmentos de la voz de los silencios. La desventaja que tiene el utilizar la energía en tiempo corto es que resulta ser muy sensible a valores grandes de la señal ya que existe un cuadrado, enfatizándose en las diferencias de valores que existen entre muestra y muestra. Una forma de disminuir estas diferencias es el utilizar la función de magnitud promedio en vez de la función de energía

$$M_n = \sum_{m=-\infty}^{\infty} |x(m)|\, w(n-m) \qquad 1.7$$

en la cual se suman los valores absolutos de la señal en vez de sumar los cuadrados. Representación que resulta de igual utilidad que la función de energía Fig. 1.3. Aunque los rangos dinámicos de las funciones varíen en una raíz cuadrada y que las diferencias entre voz sonora y voz sorda no sean tan pronunciados para el caso de la función magnitud promedio.

Taza de Cruces por Cero

Una forma de medir el contenido espectral de una señal discreta es cuantificar cuantas veces la señal cruza por cero, es decir, si dos muestras sucesivas tienen distinto signo; y aunque las señales de voz son de banda ancha y que la interpretación espectral por medio de la taza de cruces por cero es muy poco precisa, se puede estimar las propiedades espectrales de la señal de voz. La taza de cruces por cero se define por

$$Z_n = \sum_{m=-\infty}^{\infty} \left|\mathrm{sgn}[x(m)] - \mathrm{sgn}[s(m-1)]\right| w(n-m) \qquad 1.8$$

$$\mathrm{sgn}[x(n)] = \begin{cases} 1 & x(n) \geq 0 \\ -1 & x(n) \langle 0 \end{cases} \qquad 1.9$$

$$W(n) = \begin{cases} 1/2N & 0 \leq n \leq N-1 \\ 0 & c.o.c \end{cases} \qquad 1.10$$

Estas expresiones checan las muestras por pares, se suman aquellas con signos distintos y el resultado se promedia sobre Nmuestras consecutivas (opcional).

En una señal de voz la taza de cruces por cero se utiliza bajo algunas consideraciones que dependen del modelo de voz. La energía de la voz sonora se concentra por debajo de los 3kHz, mientras que la energía de la voz sorda se encuentra en su mayoría en las altas frecuencias como se ve en la Fig. 1c.
La taza de cruces por cero se incrementa cuando en ese segmento existen las altas frecuencias e inversamente las bajas frecuencias involucran una menor taza de cruces por cero; de todo esto resulta una fuerte correlación entre la taza de cruces por cero y la

distribución de la energía con la frecuencia. De este punto podemos concluir que si la taza de cruces por cero es alta, la voz es sorda. Y que si la taza es baja se tiene voz sonora. Claro que esta conclusión no es del todo correcta por que hace falta definir otras cuestiones, pero mientras tanto esto nos servirá más adelante.

Función de Autocorrelación

La función de autocorrelación de una señal determinística y discreta se define por

$$\phi(k) = \sum_{m=-\infty}^{\infty} x(m)x(m+k) \qquad 1.11$$

Esta función es de suma utilidad en el sistema de reconocimiento por que nos ayuda a obtener otras propiedades de la voz, por ejemplo:

- Si la señal es periódica con periodo P, entonces: $\phi(k) = \phi(k+p)$, es decir, la función de autocorrelación resulta igualmente periódica con periodo P
- Es una función par: $\phi(k) = \phi(-k)$
- Tiene su máximo valor en $k=0$; ósea $|\phi(k)| \leq \phi(0)$ para toda k.
- La cantidad $\phi(0)$ es igual a la energía para señales determinísticas o es igual a la potencia promedio si las señales son periódicas o aleatorias.

Con la ayuda de estas propiedades, podemos observar que para señales periódicas, la función de autocorrelación presenta máximos en los puntos $0, \pm P, \pm 2P, \ldots$ Esto es, ignorando el punto de origen el periodo de la señal puede estimarse al localizar el primer máximo de la función; y resulta aún más importante ya que nos permite estimar la periodicidad en los siguientes segmentos de la señal, incluyendo las señales de voz. De la misma manera que para los métodos anteriores, conviene obtener una representación en tiempo corto.

$$R_n(k) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)x(m+k)w(n-k-m) \qquad 1.12$$

Volviendo a escribir la expresión

$$R_n(k) = \sum_{n=0}^{n=N-1-m} x_w(n)x_w(n+m) \qquad \text{m=1,2, ...,P} \qquad 1.13$$

esta es la expresión que utilizaremos después.

Filtrado de Pre-énfasis

En el espectro de la voz existe una caída de -6 dB/octava, conforme la frecuencia aumenta. Esto se debe a la combinación de una caída de -12 dB/octava ocasionada por la fuente de excitación de la voz y un incremento de + 6 dB/octava ocasionado por la radiación de la boca. Esto significa, que cada vez que la frecuencia aumenta al doble, la amplitud de la señal se reduce en un factor de 16. Por lo que se desea compensar este roll-off de -6 dB/octava en el rango apropiado, de manera que la medición del espectro tenga un rango dinámico similar a lo largo de todo su ancho de banda.

Esto es referido como Pre-énfasis. En un sistema de procesamiento digital de señales, el Pre-énfasis puede ser implementado ya sea por un filtro analógico paso altas de primer orden con una frecuencia de corte de 3 dB en algún punto entre los 100 Hz y 1 kHz (la posición exacta no es crítica), el cual precede al filtrado anti-traslape y al convertidor A/D; o con un filtro digital paso altas que procese a la señal de voz digitalizada. Este filtrado digital puede ser implementado al usar la ecuación en diferencias

$$y[n] = x[n] - ax[n-1] \qquad 1.14$$

Donde $y[n]$ es la muestra actual que se obtiene a la salida del filtro de Pre-énfasis, $x[n]$ es la muestra actual que se tiene a la entrada del filtro; $x[n-1]$ es la muestra anterior y a es una constante usualmente escogida entre 0.9 y 1. Calculando la transformada Z a la ecuación anterior

$$Y(z) = X(z) - az^{-1}X(z) = (1 - az^{-1})X(z) \qquad 1.15$$

Donde z^{-1} representa el operador de retardo por muestra. La función de transferencia $H(z)$ del filtro es:

$$H(z) = Y(z) / X(z) = 1 - a z^{-1} \qquad 1.16$$

Para el caso de segmento de silencio, no existe la necesidad de aplicar el filtrado, ya que no existen cambios espectrales que necesiten ser eliminados. Sin embargo, por su simplicidad, el Pre-énfasis es normalmente aplicado a los segmentos de silencio también.

Detección de Inicio y Fin de Palabras Aisladas

El problema de localizar donde inicia y donde termina una palabra resulta importante en muchas áreas del procesamiento de voz, y es de particular importancia para este trabajo encontrar estos aspectos que son determinantes para el buen funcionamiento del sistema de reconocimiento, es decir, solamente trabajaremos con los segmentos de voz eliminando silencios y ruido.

El algoritmo empleado en este trabajo es conocido como el método de Rabiner-Sambur. Este algoritmo mezcla dos herramientas para la detección de inicio y fin de una palabra.

Primero se debe de obtener trama por trama *n* muestras de la magnitud promedio de la señal así como su taza de cruces por cero. De manera de obtener una caracterización estadística del ruido de fondo se considera que las primeras 10 tramas no contienen voz. Utilizando esta estadística se calculan unos umbrales que nos servirán para detectar el inicio y fin más adelante. El primer umbral (ITU) nos sirve para establecer el intervalo en el cual la energía promedio de la señal siempre será excedida y se asume que el inicio y el fin se encontrarán fuera de este intervalo. Entonces, verificando las tramas desde el punto donde se rebaso por primera vez el umbral ITU hacia el inicio (final) de la grabación, hasta un punto N1 (N2) donde la magnitud promedio cae por debajo de un umbral menor ITL y que es determinado momentáneamente como el inicio (final) de la palabra. Resulta conveniente suponer que el inicio y fin no se encuentran en estos puntos, por lo que el siguiente paso será verificar hacia el inicio (final) de la grabación desde N1(N2) la tasa de cruces por cero y compararlas contra un umbral ITZC. Esto se realiza para las 25 tramas que preceden a N1. Y si la tasa de cruces por cero excede el umbral ITZC 3 o más veces, el inicio N1 es recorrido hasta el punto en donde el umbral fue excedido por primera vez. En caso contrario el inicio se escoge el primer punto N1. Se realiza un procedimiento similar para determinar el final de la palabra. Un ejemplo típico del proceso ya realizado se encuentra conformado por las tres imágenes de la Fig. 1

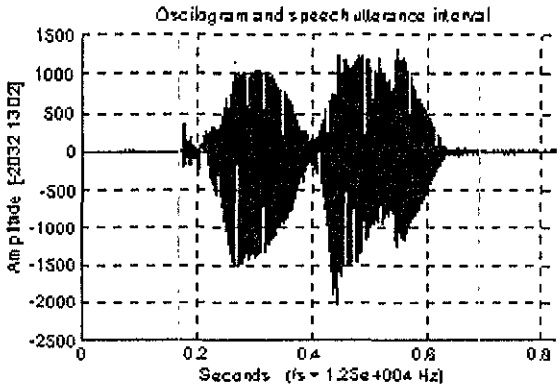

Fig. 1a. Muestra de una señal de voz (palabra conectada de dos dígitos, 2 y 3) en el tiempo. Los ejes verticales denotan el inicio y fin de palabra después de haber pasado por el proceso de Rabiner

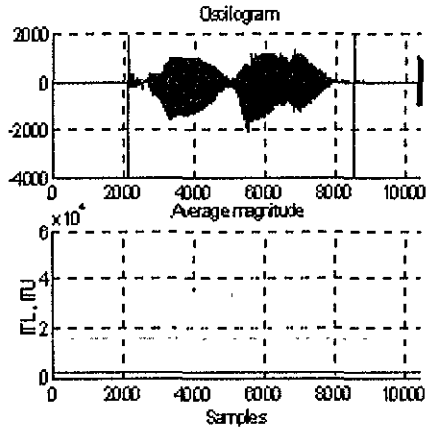

Fig. 1b. Muestra la energía promedio de la señal; así como los umbrales ITL e ITU en el proceso de detección de inicio y fin de palabras.

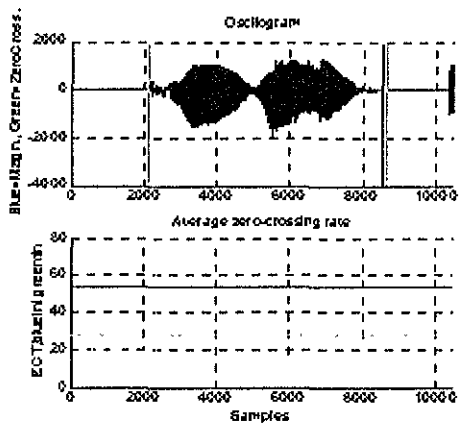

Fig. 1c. Muestra la taza de cruces por cero de la palabra conectada; así como los umbrales IZCT para el inicio y fin de palabra.

FALTA PAGINA

No. 11

1.2 *El modelo del reconocimiento de patrones*

La Fig. 2 muestra el modelo de reconocimiento de patrones que es usado ampliamente en los sistemas de reconocimiento de palabras aisladas.

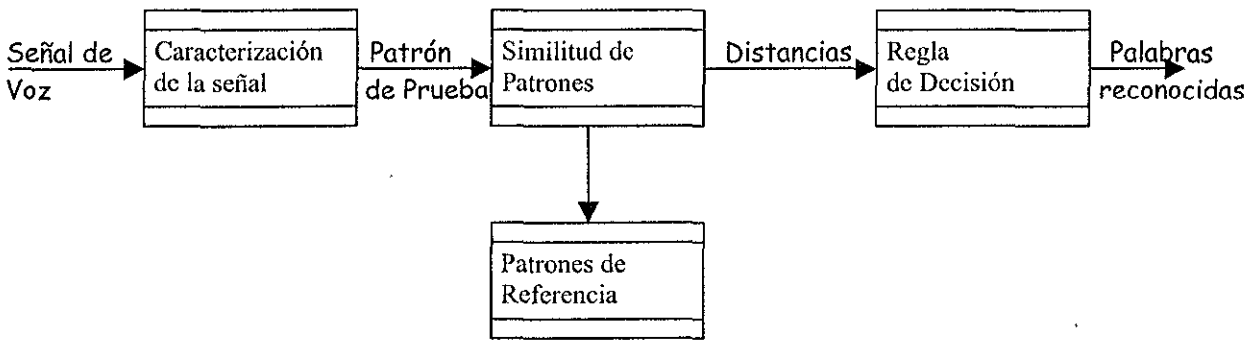

Fig. 2 Modelo de reconocimiento de patrones adaptado al reconocimiento de palabras

Del modelo podemos destacar que hay 3 procedimientos básicos:

1. Caracterización de señales
2. Determinación de la similitud entre las señales
3. Regla de decisión

A la entrada del modelo se tiene una señal acústica (en este caso será una palabra aislada o una cadena de palabras). En la salida del modelo se encuentra la mejor estimación de la señal que se encuentra en la entrada. A la salida del modelo se puede tener un conjunto de las mejores estimaciones, que después serán referidas a un nivel más alto de procesamiento en el sistema de reconocimiento.

Existen otras versiones del modelo de la Fig. 2, pero su esencia es la misma; de hecho no se ha propuesto un modelo alternativo. La experiencia de trabajar con este modelo y del por qué ha sido enfocado al reconocimiento y a otras muchas aplicaciones, es debido a que:

1. Es invariable a diferentes vocabularios de habla, usuarios, características del lenguaje, algoritmos para determinar la similitud, y reglas de decisiones;
2. Fácil de implementar;
3. Funciona bien en la práctica.

Cualquiera de los puntos anteriores justifica el uso del modelo de la Fig. 2, las tres razones en conjunto definitivamente obligan a utilizar este modelo; por lo tanto, el presente trabajo se referirá al modelo de la Fig. 2 a lo largo de los capítulos.

Del primer bloque se puede observar que (caracterización de la señal) la señal básicamente se procesa con una técnica de reducción de datos; es decir, un número largo de datos (en este caso las muestras de la señal de voz en el tiempo grabadas a una velocidad de muestreo apropiado) es transformado en un pequeño conjunto de características que equivalen a la señal de entrada, en el sentido de que en ellas se encuentran descritas fielmente los rasgos más significativos de la señal acústica.

Para representar la señal de voz se han propuesto bastantes y diferentes tipos de caracterizaciones, fluctuando desde un simple conjunto de característico de la energía y cruces por cero, hasta conjuntos de características más complejos y completos, tales como el espectro en tiempo-corto y los coeficientes de predicción (LPC). La decisión de tomar alguno de todos los posibles conjuntos que caracterizan la onda acústica depende fuertemente de las restricciones y de las condiciones impuestas por el sistema, por mencionar: velocidad del proceso, respuesta en tiempo real, complejidad en el procesamiento, etc., pero para tener un buen criterio en la selección se debe tener en cuenta principalmente:

1. Tiempo de cómputo
2. Almacenamiento
3. Fácil implementación.

Y sobre todos los criterios, el sistema debe acertar al reconocimiento; es complejo llevar acabo esta tarea ya que todos los criterios dependen de muchas variables propias del sistema.

2. CARACTERIZACIÓN DE LA SEÑAL DE VOZ

2.1. Modelando la señal de voz con el modelo LPC

Esta técnica para procesar la señal de voz es común que se utilice en los sistemas de reconocimiento de palabras (aisladas o conectadas). Esta técnica originalmente fue propuesta por Itakura [1].

El modelo LPC se asemeja verdaderamente al modelo básico de la reproducción de voz, en el que la señal acústica se modela como la salida de un sistema lineal variante en el tiempo excitado por cada pulso cuasiperiodico (para sonidos de voz) o por el ruido aleatorio (sonidos afonizados). El modelo LPC provee un robusto, confiable, y acertado método para la estimación de los parámetros que caracterizan al sistema lineal variante en el tiempo.

En el análisis LPC, la sección del tracto vocal del modelo de reproducción de voz se representa por medio de un filtro digital lineal variante con el tiempo. Este filtro debe representar los efectos de la radiación de los labios, la forma del pulso glotal, y el acoplamiento de la cavidad nasal cada vez que se requiera. La representación de la voz la efectúa en una tasa de transmisión o almacenamiento de bits muy baja. La importancia de esta técnica radica tanto en su capacidad de proveer unos muy buenos parámetros que estiman a la voz, como una relativa rapidez en la velocidad de cálculo.

La idea básica que esta detrás de este análisis, es que a una muestra de voz se le puede aproximar como una combinación lineal de muestras de voz que acaban de suceder o pasar. Minimizando la suma del error cuadrático medio entre las muestras actuales y la de las que han sido linealmente predecidas, por cada muestra de voz, un único conjunto de coeficientes del predictor puede ser determinado. esto es

$$s(n) \approx a_1 s(n-1) + a_2 s(n-2) + \ldots + a_p s(n-p), \qquad 2.1$$

donde los coeficientes $a_n,\ldots,a_p$ se suponen constantes sobre la trama de voz analizada. La ecuación anterior se convierte en igualdad al incluir un termino de excitación $Gu(n)$, quedando

$$s(n) = \sum_{i=1}^{p} a_i\, s(n-i) + Gu(n), \qquad 2.2$$

donde $u(n)$ es la excitación normalizada y G es su ganancia. Al expresar esta ecuación en el dominio de z se obtiene la relación

$$S(z) = \sum_{i=1}^{p} a_i\, z^{-i} S(z) + GU(z), \qquad 2.3$$

conduciéndonos a la función de transferencia,

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)} \qquad 2.4$$

La interpretación de esta última ecuación está dada en la Fig. 2.1, donde se muestra la fuente de excitación normalizada, *u(n)*, siendo escalada por la ganancia *G*, y actuando como entrada al sistema all-pole, *H(z) = 1/A(z)*, para producir la señal de voz, *s(n)*.

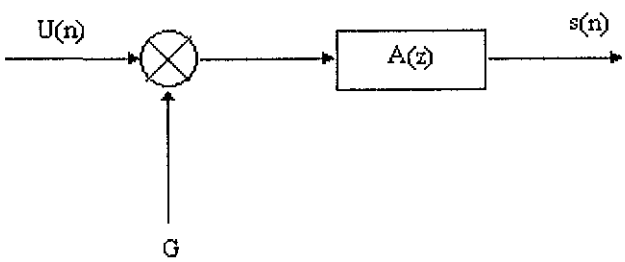

Fig. 2a Modelo lineal de predicción para voz

Basados en el previo conocimiento de que la función de excitación para voz es un tren de pulsos cuasiperiódico (para sonidos de voz), o una fuente aleatoria de ruido (para sonidos ensordecedores), entonces el modelo apropiado de síntesis de voz corresponderá al análisis LPC, como se muestra en la Fig. 2b la fuente de excitación normalizada depende de un switch en donde su posición se controla por los caracteres de la voz (voz/sin voz), que escoge un tren de pulsos cuasiperiódico como excitación para sonidos de voz, o una secuencia de ruido aleatorio para sonidos donde no hay voz. La ganancia *G* apropiada de la fuente es estimada de la señal de voz, y la fuente escalada es usada como entrada a un filtro digital *(H(z))*, que se controla por los parámetros característicos de la voz que son producidos por el tracto vocal. De esta manera, los parámetros de este modelo son cuando hay sonidos de voz, cuando no los hay, periodo para sonidos de voz, el parámetro de la ganancia, y los coeficientes del filtro digital, *{ak}*. Todos estos parámetros varían lentamente con el tiempo.

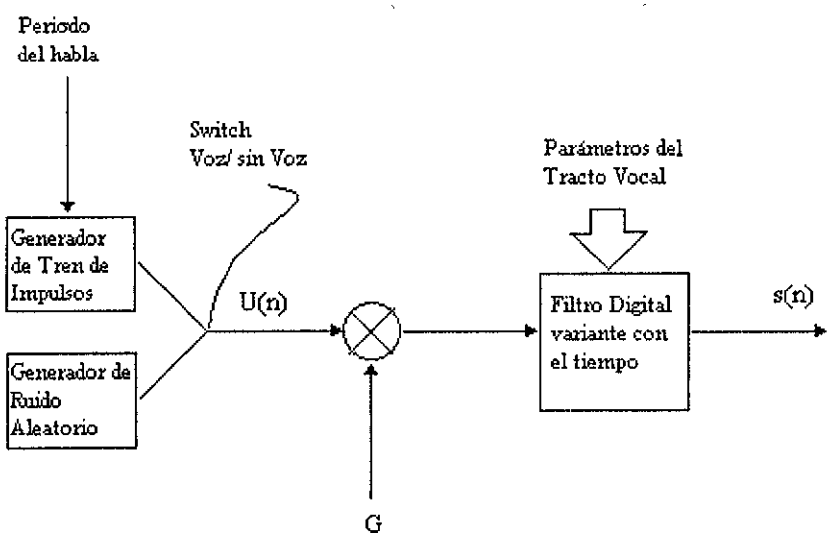

Fig. 2b Modelo de síntesis de voz basado en el modelo LPC

Además la filosofía del modelo LPC se encuentra íntimamente relacionada con el conocimiento de que la fuente de excitación para la señal de voz es esencialmente un tren de pulsos cuasiperiódicos (para la voz sonora) o una fuente de ruido aleatorio (para la voz sorda) que sirve de excitación a un sistema lineal variante con el tiempo, y que precisamente el modelo LPC provee un método robusto, realizable y preciso para estimar los parámetros que caracterizan a este sistema lineal e invariable con el tiempo.

Análisis de ecuaciones LPC

De acuerdo al modelo de la figura 2a, la relación exacta entre *s(n)* y *u(n)* es

$$s(n) = \sum_{k=1}^{p} a_k \, s(n-k) + Gu(n) \qquad 2.5$$

Consideramos la combinación lineal de las muestras de voz pasadas como la estimación $\hat{s}(n)$ definida como,

$$\hat{s}(n) = \sum_{k=1}^{p} a_k \, s(n-k) \qquad 2.6$$

Formamos el error de predicción, *e(n)*, de la siguiente forma,

$$e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \qquad 2.7$$

con función de transferencia del error,

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^{p} a_k \, z^{-k} \qquad 2.8$$

Ahora el problema básico en el análisis de predicción lineal es determinar el conjunto de coeficientes de predicción, *{ak}*, directamente de la señal de voz, de modo que las propiedades espectrales de la voz se mantengan constantes. Y ya que las características espectrales de la voz varían con el tiempo, los coeficientes de predicción en un tiempo dado n, deben estimarse en un segmento corto de la señal de voz alrededor de este tiempo. Entonces la idea fundamental es encontrar un conjunto de coeficientes de predicción que minimicen el error de predicción cuadrático medio sobre un segmento corto de la forma de onda de la voz. (Normalmente este análisis se realiza sobre tramas sucesivas de voz, con un espaciamiento del orden de *10 ms* entre tramas).

Para empezar con las ecuaciones que se deben resolver, y así poder determinar los coeficientes de predicción, se define el segmento de voz y el segmento de error en un tiempo n como,

$$\begin{aligned} s_n(m) &= s(n+m) \\ e_n(m) &= e(n+m) \end{aligned} \qquad 2.9$$

y lo que buscamos es minimizar la señal de error cuadrático medio en el tiempo n,

$$E_n = \sum_m e^2(m) \qquad 2.10$$

y al usar la definición de *en(m)* en términos de *s(m)*, se puede escribir como,

$$E_n(m) = \sum_m \left[s_n(m) - \sum_{k=1}^{p} a_k \, s_n(m-k) \right]^2 \qquad 2.11$$

Para resolver esta ecuación, se deriva parcialmente *En* con respecto a cada *{ak}* y el resultado se iguala a cero,

$$\frac{\partial E_n}{\partial a_k} = 0, \quad k = 1,2,\ldots,p \qquad 2.12$$

quedando,

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} \hat{a}_k \sum_m s_n(m-i)s_n(m-k) \qquad 2.13$$

Reconociendo que los términos de la forma $\sum_m s_n(m-i)s_n(m-k)$ representan las covarianzas de los segmentos *Sn(m)*, se pueden escribir como,

$$\phi(i,k) = \sum_m s_n(m-i)s_n(m-k) \qquad 2.14$$

Por lo que la ecuación 2.14 se puede escribir en forma mas compacta,

$$\phi_n(i,0) = \sum_{k=1}^{p} \hat{a}_k \, \phi_n(i,k) \qquad 2.15$$

Ecuación que describe un conjunto de p ecuaciones con p incógnitas. Y para obtener los coeficientes de predicción óptimos tenemos que calcular el conjunto resultante de ecuaciones simultáneas.

Existen varias formas de calcular los coeficientes de predicción: método de covarianzas, método de autocorrelación, el del filtro inverso, el de estimación espectral, el de máxima probabilidad y el método del producto interno. En reconocimiento de voz (y en este trabajo), se utiliza el método de autocorrelación debido a su eficacia computacional así como estabilidad inherente. Este método siempre produce un filtro de predicción cuyos ceros se encuentran dentro del círculo unitario en el plano *Z*.

Método de autocorrelación

Una forma de determinar los límites de las sumatorias de las ecuaciones es asumir que el segmento, *Sn(m)*, es igual a cero fuera del intervalo $0 \le m \le N-1$. Esto puede escribirse de la forma

$$s_n(m) = s(m+n)w(m) \qquad 2.16$$

Donde *w(m)* es una ventana de longitud finita (usualmente ventana de Hamming), que es igual a cero fuera del intervalo $0 \le m \le N-1$.

El efecto de esta suposición en los límites de la sumatoria para las expresiones que contienen a *En* puede verse al considerar la ecuación. Claramente, si *Sn(m)* es diferente de cero solo para $0 \le m \le N-1$, entonces el correspondiente error de predicción, *en(m)*, para un predictor de orden p, será diferente de cero en el intervalo $0 \le m \le N-1+p$. Entonces, para este caso *En* se expresa apropiadamente como,

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \qquad 2.17$$

y *fn(i, k)* se puede expresar,

$$\phi_n(i,k) = \sum_{m=0}^{N+p-1} s_n(m-i)s_n(m-k), \quad \begin{matrix} 1 \le i \le p \\ 0 \le k \le p \end{matrix} \qquad 2.18$$

Como esta última ecuación es sólo función de *i-k* (en lugar de ser función de dos variables *i* y *k*), la función de covarianza, *fn(i, k)*, se reduce a una simple función de autocorrelación,

$$\phi_n(i,k) = r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k), \quad \begin{matrix} 1 \le i \le p \\ 0 \le k \le p \end{matrix} \qquad 2.19$$

Además como la función de autocorrelación es simétrica, *rn (-k) = rn (k)*, las ecuaciones LPC pueden expresarse como,

$$\sum_{k=1}^{p} r_n\left(|i-k|\right)\hat{a}_k = r_n(i), \qquad 1 \le i \le p \tag{2.20}$$

y se puede expresar en forma matricial,

$$\begin{bmatrix} r_n(0) & r_n(1) & r_n(2) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \cdots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \cdots & r_n(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \cdots & r_n(0) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \vdots \\ \hat{a}_p \end{bmatrix} = \begin{bmatrix} r_n(1) \\ r_n(2) \\ r_n(3) \\ \vdots \\ r_n(p) \end{bmatrix} \tag{2.21}$$

Esta matriz de orden *pxp* con los valores de autocorrelación, es una matriz Toeplitz (simétrica con los elementos de la diagonal principal iguales) que puede resolverse eficientemente con el uso de varios procedimientos numéricos. Como el algoritmo de Levinson-Durbin.

Obtención de los Coeficientes de Predicción LPC

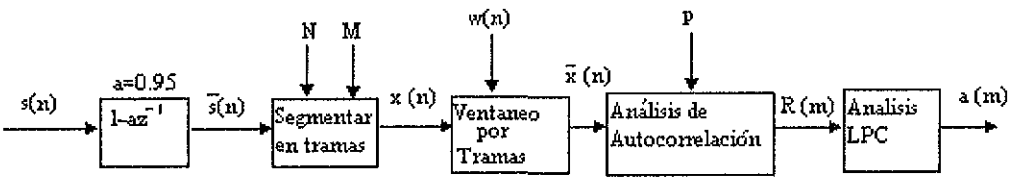

Fig. 2c El procesamiento de la señal para extraer las características con modelo para reconocimiento

La Fig. 2c muestra a manera de diagrama de bloques el proceso de obtención de los coeficientes LPC. En este sistema cada trama de N muestras de voz es procesada, un vector con datos característicos se parametriza. Para obtener este vector, primero se preenfatiza la señal de voz (espectralmente la señal se aplana un poco y reduce las inestabilidades computacionales asociadas con la finita precisión aritmética del sistema de reconocimiento). Usando un sistema digital de primer orden con una función de transferencia

$$H(z) = 1 - az^{-1} \quad a=0.95 \tag{2.22}$$

Dando la señal,

$$\check{S}(n) = s(n) - as\ (n-1) \qquad 2.23$$

la señal es después apilada en secciones de N muestras (tramas) listas para ser parametrizadas. Típicamente el tamaño de las tramas se establece de 15 a 50 *ms*. Por ejemplo, N=150 a 500 para una velocidad de muestreo de 10 *kHz*, las tramas consecutivas son espaciadas por M muestras. Claramente cuando M < N, ocurre un traslape entre las tramas adyacentes. En este traslape inherentemente ocurre una suavización entre los vectores que contienen los coeficientes de caracterización. Los valores típicos de M son M = N / 3 (relación 3 a 1), M = N / 2 (relación 2 a 1), o M = N (sin traslape).

Si determinamos la lth trama de voz como $x_l(n)$, tenemos,

$$x_l(n) = \check{s}(Ml + n) \quad n = 0,1,\ldots,N-1, \qquad 2.24$$

$$l = 0,1,\ldots,L-1$$

donde l=0 será la primera trama y l=L será la Lth trama de voz; al analizar una porción de la onda de voz y con el fin de eliminar los efectos que ocurren al principio y la final de la trama se aplica a cada trama una ventana $w(n)$ que permite disminuirla señal de voz hacia cero en la parte inicial y final de cada trama, quedando la señal ventaneada como:

$$\tilde{x} \approx x(n) * w(n) \qquad 2.25$$

La ventana típica que se utiliza en los sistemas LPC, es la ventana de Hamming definida como:

$$W(n) = 0.54 - 0.46\cos(2p\ n / N - 1) \qquad 2.26$$

El siguiente paso en el procesamiento es el realizar un análisis de autocorrelación de la trama ventaneada, quedando

$$R_l(m) = \sum_{n=0}^{N-1-|m|} \tilde{x}(n)\,\tilde{x}(n+m), \qquad m = 0,1,\ldots,p \qquad 2.27$$

donde p es el orden del sistema. (Típicamente los valores de p tienen un rango de 8 a 12) El conjunto característico

$$X(l) = \{ R_l(0), R_l(1), ..., R_l(p) \} \qquad 2.28$$

es frecuentemente usado como la salida del analizador LPC para el reconocimiento, aunque ambos patrones, de referencia y de prueba pueden ser derivados de las características de la Eq. (2.28) un completo análisis LPC requiere que el error medio cuadrático (en el dominio del tiempo) del modelo all-pole sea apropiado para el espectro de la trama de datos, y éste se encuentra resolviendo simultáneamente el conjunto de ecuaciones lineales. Se pueden obtener con el empleo de métodos numéricos. Como lo es el algoritmo de Levinson-Durbin que a continuación se presenta.

Inicialización:

$$\begin{aligned}
&E_{LP}^{(0)} = r(0) \\
&Para 1 \le i \le p \\
&\{ \\
&k_i = \frac{\left[r(i) - \sum_{j=1}^{L-1} \alpha_j^{(i-1)} r(|i-j|) \right]}{E^{(i-1)}} \quad 1 \le j \le i-1 \quad \} \qquad 2.29 \\
&\alpha_i^{(i)} = k_i \\
&\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \\
&E^{(i)} = (1 - k_i^2) E^{(i-1)}
\end{aligned}$$

Las iteraciones son realizadas para $i = 1, 2, 3, ..., p$, y la solución final esta dada por:

$$a_m = coeficientes\ LPC = \alpha_m^{(p)} \qquad 2.30$$

El modelo resultante all-pole de la trama tiene la forma

$$A_l(z) = \frac{G}{1 + \sum_{m=1}^{P} a_l(m) z^{-m}} \qquad 2.31$$

Si se define $a_l(0) = 1$, entonces se puede reescribir la Eq. (2.29)

$$A_l(z) = \frac{G}{\sum_{m=0}^{P} a_l(m) z^{-m}} \qquad 2.32$$

donde G es un factor de ganancia, y el conjunto de coeficientes al(m), $m = 1, 2, \ldots, p$ define el modelo all-pole.

3. AJUSTE DINÁMICO EN EL TIEMPO

3.1. Registro de los modelos en el tiempo

Después de haber procesado las señales trama por trama, el siguiente paso en el proceso del reconocimiento del modelo de la Fig. 2, consiste en determinar la similitud entre el patrón de prueba y el de referencia; para ello, se recurre a un algoritmo que pueda alinear en el tiempo ambas señales y determinar su similitud al mismo tiempo.

Para poder determinar la similitud entre las señales es necesario identificar claramente las características que ocurren al momento de registrar los modelos, se pueden resumir en los siguientes puntos:

1. Variaciones entre locutores. Nunca suenan iguales dos personas; es decir, la señal de voz contiene variables que dependen exclusivamente del propio locutor así como lo difícil que resulta separar la información fonética de una palabra.
2. Ambigüedad. Las variables acústicas no son mapeadas una a una, es decir, cuando no podemos escuchar correctamente lo que fue pronunciado, el cerebro puede reconstruir la palabra sobre la base del contexto y el conocimiento del lenguaje en el que se está hablando. En repetidas ocasiones tenemos que pedir que se nos deletree la palabra que no entendemos.
3. Variaciones de voz en un individuo. Estas siempre estarán presentes aún cuando el locutor este muy bien entrenado. Ellas incluyen:

    - Murmullos. Palabras cortas como "Y" o "O" que son frecuentemente reducidas a simples murmullos. Algunas sílabas son opacadas
    - Variaciones fonéticas. Las frecuencias fundamentales, así como la duración de las transiciones pueden cambiar con el tiempo.
    - Coarticulaciones. Las características fonéticas de los sonidos son afectadas por el contexto. Por ejemplo existen vocales que tienen diferente función según sean pronunciadas, ya sean nasales o no.
    - Variaciones temporales. La duración de una palabra puede cambiar o algunas partes de sus partes

4. El ruido y la interferencia. El ancho de banda del ruido hace débiles a los sonidos Fricativos causando que los predictores estimen parámetros incorrectos; también hace más difícil la detección del inicio y fin de las palabras.

La técnica clásica para determinar la similitud entre los patrones, consiste simplemente en alongar o comprimir uniformemente la señal de prueba, hasta que se tengan la misma longitud de la señal de referencia. La efectividad de este proceso depende simplemente de la compresión o la expansión de la escala en el tiempo, por supuesto esto no es suficiente para llevar acabo una correcta alineación en el tiempo.

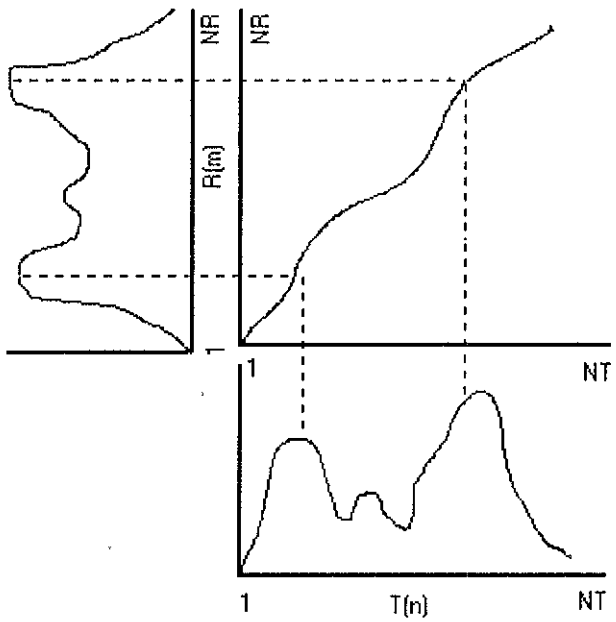

Fig. 3a. Ejemplo del registro en el tiempo del patrón de referencia y el de prueba.

La Fig. 3a, nos da una clara imagen de lo que el algoritmo debe realizar; es decir, hacer la función de alinear el patrón de prueba $T(t)$ y un patrón de referencia $R(t)$. La meta de dicho algoritmo es encontrar una función de alineación $w(t)$ que fuera mapeando R con las correspondientes partes de T. El criterio para determinar la correspondiente similitud se encuentra en el cálculo de una función de distancia $D\ (\ T,\ R\ )$ que minimice las pequeñas discrepancias que hay entre los patrones y que hagan máximas las diferencias. Así, que el problema lo podemos pasar a la forma discreta,

$$T = \{ T(1), T(2), \ldots T(NT) \} \qquad 3.1$$

$$R = \{ R(1), R(2), \ldots R(NR) \} \qquad 3.2$$

la óptima alineación describe un camino a una curva que relaciona el eje m del patrón de referencia y el eje n del patrón de prueba (ambos patrones están representados en el tiempo), de la siguiente forma:

$$m = w(n) \qquad 3.3$$

Las condiciones iniciales y finales de la Fig. 3a pueden ser expresadas formalmente como restricciones sobre w(n) de la forma,

$$W(1) = 1 \qquad 3.4$$

$$W(NT) = NR \qquad 3.5$$

muchas técnicas han sido propuestas para determinar la función w de alineación en el tiempo entre ambas señales tales como:

1. El alineamiento lineal en el tiempo

$$m = w(n) = 1 + (n-1)(NR-1)/(NT-1) \qquad 3.6$$

2. Unión de eventos en el tiempo. Es decir, encontrar los instantes en los que ocurren eventos significativos en ambos patrones (de referencia y de prueba), y alinearlos en el tiempo,

$$m_1 = w(n_1) \qquad 3.7$$
$$m_2 = w(n_2) \qquad 3.8$$
$$\vdots$$
$$m_Q = w(n_Q) \qquad 3.9$$

Y un arreglo funcional para w(n) se encuentra basado en esas restricciones.

3. Máxima correlación. La función de alineación *w(n)* se varía para maximizar la correlación entre los patrones de referencia y de prueba.

$$R^* = \max_{w(n)} \sum_n (T(n)R(w(n))) \qquad 3.10$$

Donde la optimización es realizada en una manera restringida.

4. Ajuste dinámico en el tiempo. La curva de ajuste se determina resolviendo el problema de optimización de:

$$D^* = \min_{w(n)} \left[\sum_{n=1}^{NT} d(T(n), R(w(n))) \right] \qquad 3.11$$

Donde *d(T(n), R(w(n))* es la distancia entre la trama *n* del patrón de prueba y la trama *w(n)* del patrón de referencia. En los siguientes subtemas, se desarrollará la aproximación del ajuste dinámico en el tiempo (DTW) para poder comparar dos señales.

3.2. Medición de la similitud de señales trama por trama

Se debe definir la manera de optimar la función de distancia entre dos tramas caracterizadas. Varias posibles distancias pueden ser utilizadas, dependiendo de la forma en que se presente el conjunto de datos que están caracterizando a una señal; por ejemplo: La distancia Euclideana, de Covarianza, Espectral y logarítmica LPC.

En este trabajo el sistema trabaja calculando las distancias por medio del cálculo de la distancia logarítmica LPC. Esta distancia resulta ser extremadamente eficiente y fue propuesta por Itakura, de la siguiente forma

$$d(T,R) = log\ [a_R V_T a_R^t / a_T V_T a_T^t] \qquad 3.12$$

Donde a_R y a_T, son los vectores de coeficientes LPC de referencia y de prueba respectivamente, y V_T es la matriz de coeficientes de autocorrelación de la trama de prueba. La interpretación de la distancia dada en la ecuación (3.1) se muestra también en la Fig. 3.b donde el suscrito R denota la referencia, y el suscrito T denota la prueba. El denominador del término en el paréntesis cuadrado puede ser obtenido pasando la señal de prueba $S_T(n)$ a través del sistema LPC inverso

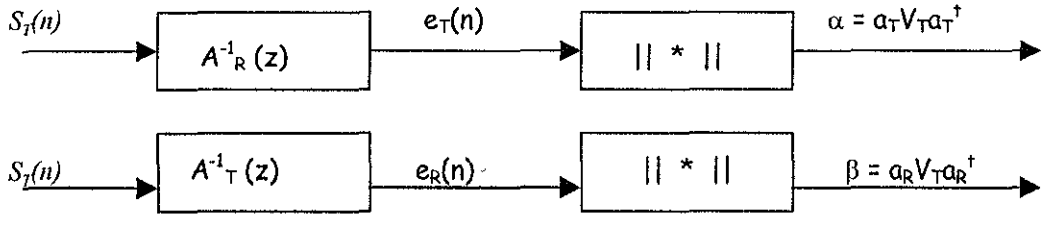

Fig. 3.b Interpretación de la parametrización de la distancia LPC

de la prueba, $A_T^{-1}(z)$ con respuesta,

$$A_T^{-1}(z) = \sum_{m=0}^{P} a_m T^{Z^{-m}} \qquad 3.13$$

Dando el error de las señales

$$e_T(n) = \sum_{m=0}^{P} a_{mT} S_T(n-m) \qquad 3.14$$

y calculando su energía da

$$\alpha = \left\| e_T(n) \right\| = \sum_{n=0}^{N-1} \left[e_T(n) \right]^2 \qquad 3.14$$

Similarmente, el término del numerador puede ser obtenido pasando la misma señal de prueba $S_T(n)$ através del sistema LPC inverso de la referencia $A_R^{-1}(z)$ con respuesta,

$$A_R^{-1}(z) = \sum_{m=0}^{P} a_m R^{Z^{-m}} \qquad 3.15$$

Dando el error de la señal

$$e_R(n) = \sum_{m=0}^{P} a_{mR} S_T(n-m) \qquad 3.16$$

Con energía

$$\beta = \left\| e_R(n) \right\| = \sum_{n=0}^{N-1} \left[e_R(n) \right]^2 \qquad 3.17$$

Así, de (3.1), (3.7) y (3.4) obtenemos

$$d\,(T, R) = log\,(\beta / \alpha) \qquad 3.18$$

Mostrando que la distancia entre los dos conjuntos característicos se relacionan con la diferencia en los conjuntos característicos LPC, que a la vez se relaciona con las diferencias entre las tramas de T y R. Uno de los aspectos más importantes es la velocidad de su cómputo,

aunque los cálculos de las distancias en cualquier sistema de reconocimiento consumirá el mayor tiempo de cómputo.

La distancia LPC como está descrita en (3.1) necesitan un largo número de multiplicaciones y adiciones. La distancia de (3.1) puede ser expresada de la forma

$$d(T,R) = \log\left[\sum_{k=0}^{P-m} \tilde{V}(m) R_R^a(m)\right] \qquad 3.19$$

Donde

$$V_T(m) = \frac{\tilde{V}(m)}{E_T} \qquad 3.20$$

Donde E_T es el error normalizado del análisis LPC de la trama en prueba y

$$R_R{}^a(m) = \sum_{k=0}^{p-m} a_R(k) a_R(k+m) \qquad 3.21$$

$R_R{}^a(m)$ es la autocorrelación del vector finito

$$A_R = [1, a_R(1), a_R(2), \ldots, a_R(p)] \qquad 3.22$$

Usando la ecuación (3.9), los cálculos de distancia requieren $(p+1)$ multiplicaciones y adiciones, y un *log*.

Analizando, ambas distancias, Euclidiana y distancia LPC son razonablemente buenos candidatos para utilizarce en el cálculo de las distancias dentro de los sistemas de reconocimiento; ambas son ampliamente utilizadas en los sistemas de reconocimiento para palabras aisladas y conectadas.

3.3. El marco de trabajo de DTW

El problema del ajuste de las señales en el tiempo se resuelve frecuentemente con el proceso conocido como DTW (Dinamic Time Warping). Este proceso crea una línea no uniforme resultante de la comparación de dos señales, y con ello logramos obtener una función que nos diga que tan similares son. Un claro ejemplo esta en la Fig. 3c las señales que serán comparadas se encuentran a lo largo de los ejes *(x,y)*, y la diagonal ondulada muestra gráficamente la comparación que hubo entre ellas, entre más distorsionada la línea más desiguales fueron las señales comparadas. Si la línea resultante pasa a través del punto *(i,j)*, entonces la *ith* muestra de la señal *A* se alinea con la *jth* muestra de la señal *B*. La diagonal debería ser una recta para cuando son iguales las muestras. En otras palabras, la curva resultante ajusta con una expansión o compresión uniforme la comparación entre dos señales, ósea, la diagonal sufre de una distorsión.

Las entradas al proceso del ajuste dinámico son dos funciones en el tiempo, se puede usar la amplitud o los coeficientes LPC de las señales. A la salida del proceso se encuentra la función de ajuste junto con el grado de distorsión que se hubo entre las dos señales.

Los contornos que van a ser comparados son definidos como funciones escaladas en el tiempo.

$$A = a_1, a_2, \ldots, a_i, \ldots, a_M, \qquad 3.23$$
$$B = b_1, b_2, \ldots, b_j, \ldots, b_N, \qquad 3.23$$

El problema es el ajustar las a's con las b's así como el minimizar la discrepancia en cada par de muestras que se comparan. Esto se puede representar esquemáticamente dibujando una porción de la Fig. 3a como se muestra en la Fig. 3c definiendo la función de ajuste como:

$$C = c(1), c(2), \ldots, c(k), \ldots, c(K) \qquad 3.24$$

Donde cada c es un par de puntos de las muestras que están siendo comparadas,

$$c(k) = [i(k), j(k)] \qquad 3.25$$

en la figura,

$$C = (1,1), (2,2), (3,2), (4,3), (5,4),\ldots \qquad 3.26$$

y las líneas punteadas la correspondencia asignada entre *a(4)* y *b(6)*,

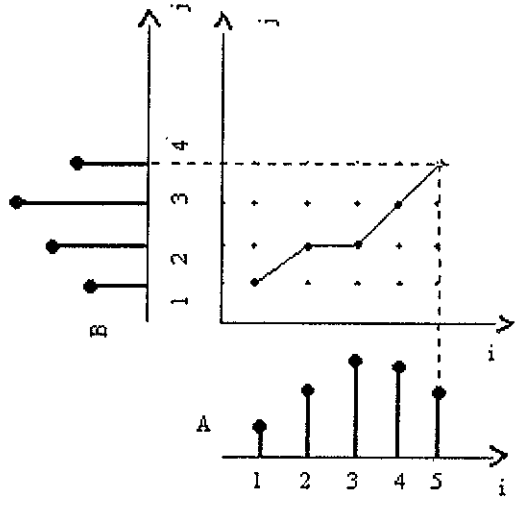

Fig. 3c Detalle del proceso de ajuste dinámico en el tiempo.
El punto *(5,4)* se alinea con *a(5)* y *b(4)*

Para cada *c(k)* existe una función costo

$$d[c(k)] = \delta(a_{i(k)} - b_{j(k)}) \qquad 3.27$$

la cual refleja la discrepancia entre el par de muestras. La típica función costo es le cuadrado de la diferencia entre las muestras,

$$d[c(k)] = (a_{i(k)} - b_{j(k)})^2 \qquad 3.28$$

Si los contornos están determinados por paquetes de coeficientes de predicción, entonces la distancia utilizada es frecuentemente la de Itakura.

La función de ajuste tiene que minimizar la función costo.

$$D(c) = \sum_{k=1}^{K} d[c(k)] \qquad 3.29$$

Sujeta a las siguientes restricciones:

1. La función tiene que ser monótonamente creciente:

$$i(K) \geq i(k-1) \qquad y \qquad j(k) \geq j(k-1)$$

2. La función debe de comparar los puntos finales de A y B:

$$i(1) = k(1) = 1 \qquad i(k) = M \qquad j(k) = N$$

3. La función no debe de saltarse ningún punto:

$$i(k) - i(k-1) \leq 1 \qquad j(k) - j(k-1) \leq 1$$

4. Existen usualmente algunos límites globales sobre la máxima cantidad del ajuste dinámico. El más simple es probablemente

$$| i(k) - j(k) | < Q$$

donde Q es el ancho de la ventana. Alternativamente, un límite global puede ser impuesta sobre la pendiente de la función de ajuste; Esto puede ser llevado a cabo restringiendo el dominio del proceso, usando un paralelogramo como el que se muestra en la Fig. 3d. El método del paralelogramo es seguro cuando M = N, si M = 2N o N = 2M, el paralelogramo se colapsa en una línea recta y no habrá ningún ajuste.

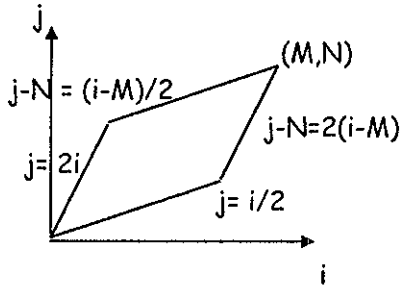

Fig. 3d. Paralelogramo definido por las restricciones o pendientes en el ajuste dinámico en el tiempo.

La función de ajuste se puede ver como el proceso de encontrar un camino que tenga un costo mínimo, a través del espacio que se genera al hacer la comparación de dos señales como se puede ver en la Fig. 3c, del punto *(1,1)* al punto *(M,N)* hay una función costo que muestra un índice de discrepancia entre los correspondientes puntos de las dos señales.

A primera vista, parecería que *D(C)* tendría que ser evaluada y comparada para un gran número de caminos posibles, lo cual, resulta prohibitivo. La programación dinámica mantiene en control de este problema, haciendo notar que el mejor camino de *(1,1)* hacia cualquier punto dado es independiente del que pasa más allá de ese punto. Así el costo total de *[i(k), j(k)]* es el costo de ese mismo punto mas el costo mínimo del camino acumulado:

$$D(C_k) = d\left[c(k)\right] + \underset{permitido\ c(k-1)}{MIN}\left[D(C_{k-1})\right] \qquad 3.30$$

El suscrito *"permitido c(k-1)"* se refiere al mínimo sobre todos los posibles sucesores de *c(k)*. De esta manera, las restricciones 1 y 3 ya mencionadas, conducen a que haya solo tres sucesores permitidos: si *c(k) = (i,j)*, esos serán: *(i,j-1), (i-1,j) y (i-1,j-1)*. Así sólo se necesita considerar tres posibles sucesores de avance por punto.

3.4. La regla de decisión para el reconocimiento

El último paso del modelo de reconocimiento de patrones de la Fig. 1 es la regla de decisión que escoge cual patrón o patrones de referencia se asemejan más al patrón de prueba desconocido. Aunque existe un variado número de aproximaciones se pueden aplicar en esta etapa, solo dos tipos de regla de decisiones son utilizados en los sistemas más prácticos, normalmente la regla *NN* y la *KNN*.

La regla *NN*

La regla *NN* asume que tenemos V patones de referencias, R^i, $i = 1,2, \ldots V$, y para cada patrón obtenemos el promedio de la distancia D^i resultante del algoritmo DTW. Entonces la regla simple es:

$$i^* = \arg\min_i D^i \qquad 3.31$$

se escoge el patrón, R^{i^*} con el promedio mínimo de distancia como el patrón reconocido. En algunas aplicaciones, como se verá más adelante, explícitamente escoger i^* no es necesariamente la mejor opción; se usará una lista ordenada por distancias de candidatos. En este caso, el conjunto de distancias Di es reordenado para dar un nuevo conjunto $D^{[i]}$ tal que,

$$D^{[1]} <= D^{[2]} \leq \ldots \leq D^{[v]} \qquad 3.32$$

y el conjunto de índices que da el nuevo ordenamiento $[i]$ se retiene en

$$[i] = Tabla(i) \qquad 3.33$$

donde $Tabla\ (i)$ es el índice original del ith elemento en el reordenado arreglo de distancias.

La regla KNN

La regla KNN es aplicada cuando cada entidad de referencia es representada por dos o más patrones de referencia, como sería usado para hacer patrones de referencia independientes del locutor. Así si se asume que hay p patrones de referencia para cada uno de los V (palabras de referencia), y se denota la jth ocurrencia del ith patrón como $R^{i,j}$, $1 \leq i \leq V$, $1 \leq j \leq P$,

entonces si se denota la distancia del DTW por la *jth* ocurrencia del *ith* patrón como $D^{i,j}$, y si reordenamos las P distancias de la *ith* palabra tal que,

$$D^{i,[1]} \leq D^{i,[2]} \leq \ldots \leq D^{i,[P]} \qquad 3.34$$

entonces para la regla KNN se calcula el promedio de la distancia,

$$r^i = \frac{1}{k}\sum_{k=1}^{K} D^{i,[k]} \qquad 3.35$$

Y se escoge el índice del patrón reconocido como:

$$i^* = \arg\min_i r^i \qquad 3.36$$

Similarmente para la regla NN, se puede calcular una lista ordenada del promedio de las distancias (r^i) para los casos cuando una dista de candidatos se necesite.

3.5. Variaciones del algoritmo DTW

Una de las más grandes desventajas en el algoritmo DTW presentado en la sección anterior es el asumir que los patrones de referencia y de prueba se alinean precisamente al inicio y al final de las tramas. Cuando se hizo una buena detección de inicio y fin de palabra, las restricciones descritas anteriormente son aceptables y no alteran el reconocimiento. Sin embargo en los casos donde no se pueda llevar acabo la correcta comparación de toda la palabra de prueba con la palabra de referencia entonces la restricción del punto final del algoritmo DTW es inadecuada. De esta forma, se han propuesto varias variaciones en la detección del punto final con el fin de solucionar esta restricción.

La Fig. 3e, ilustra las tres variaciones sobre el algoritmo DTW que se proponen. La primera variación, Fig. 3e.1 en ella se puede observar la restricción de los puntos finales, con un rango de 2 a 1 en su pendiente. Este método ya ha sido discutido en el cual se asume que hay una perfecta alineación entre los puntos finales de los patrones. La segunda variante, Fig. 3e.2 se puede ver que no hay restricción alguna para acoplar los puntos finales de los patrones, mas tiene una restricción en la pendiente de 2 a 1. Se retienen todas las restricciones locales del camino, pero las condiciones de acople de los puntos finales esta sin restricción, es decir

$$1 \le w(1) \le 1 + \delta \qquad 3.37a$$

$$NR - \delta \le w(NT) \le NR \qquad 3.37b$$

Donde δ es un "offset" parámetro del algoritmo DTW. Claramente, si $\delta = 0$, el algoritmo DTW se convierte idénticamente al primer método ya descrito. Para valores de δ distintos de cero, la región de (n, m) se incrementa en el plano donde el camino toma su rumbo.

De la tercera variante del algoritmo DTW (Fig. 3e.3) se puede observar que no existe ningún tipo de restricción. No utiliza restricción para acoplar los puntos finales de los patrones como en la segunda propuesta. En adición a ésta las restricciones del camino toman la siguiente forma,

$$m^*(n-1) - \varepsilon \le m(n) \le m^*(n-1) + \varepsilon \qquad 3.38$$

donde $m(n)$ es el rango sobre m para buscar el camino óptimo para cada valor de n, $m^*(n-1)$ es el m índice donde $D_A(n-1, m)$ fue mínimo,

$$m^*(n-1) = argmin_m [DA(n-1, m)] \qquad 3.39$$

Y ε es un parámetro muy amplio. Como se muestra en la Fig. 3e el algoritmo rastrea el camino local que sea óptimo de manera que los cálculos sean reducidos. El mismo camino determinará la alineación de los puntos finales entre los patrones. Esta variante es muy útil en aplicaciones de reconocimiento donde sólo uno de los puntos finales es aproximadamente bien conocido, por ejemplo, en palabras conectadas.

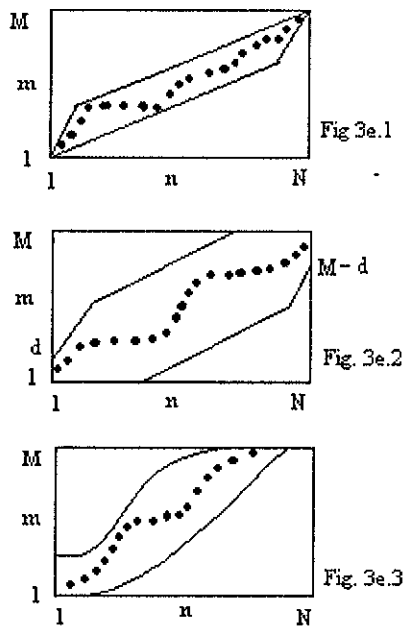

Fig. 3e Detalle de las tres caminos de las típicas técnicas del algoritmo DTW
Usadas en los sistemas de reconocimiento.

4. TÉCNICAS E IMPLEMENTACIÓN DE SISTEMAS DE RECONOCIMIENTO DE PATRONES

4.1. Reconocimiento de palabras aisladas

Hasta este punto hemos detallado el procedimiento para implementar un sistema de reconocimiento basado en la extracción de las características más importantes en una señal de voz por medio de un sistema LPC, y usando un algoritmo DTW con un cuantificador de distancias que nos indica el grado de similitud entre las señales comparadas. Analizando el sistema de reconocimiento de palabras aisladas, se observa que tiene 2 modos distintos de operación, normalmente:

Modo de Entrenamiento. - Este consiste en la adquisición de todo el vocabulario que será usado como la base de referencia. Cada parlante o conjunto de parlantes recitan cada palabra del vocabulario deseado (una o más veces) sobre algún sistema que los registre. El siguiente paso en el proceso, es aplicar un filtrado de pre-énfasis a cada una palabra de la base de referencia y un ventaneo (Hamming). Después continuamos con la obtención de las estimaciones del vocabulario, en el que para cada palabra se obtiene un conjunto de coeficientes de autocorrelación estimados cada 120 muestras (16 ms) usando un traslape entre tramas de 20 muestras. Después hay que encontrar los puntos en donde las palabras terminan y comienzan; en otras palabras, hay que separar la voz de los sonidos de fondo. Este punto es importante porque:

- Errores en la detección del punto final incrementan la probabilidad de error en el correcto reconocimiento. Grandes errores en la localización de estos puntos hacen imposible que el reconocimiento sea efectivo.
- La correcta localización de los puntos finales mantiene los cálculos computacionales del sistema al mínimo.

Una vez localizados los puntos finales de las palabras los conjuntos de coeficientes de autocorrelación de cada palabra recitada son guardados, es decir, el entrenamiento consiste de repeticiones iterativas de palabras y procesadas que conformarán el vocabulario.

Modo de Prueba.- EL método inicialmente procede de la misma forma que el modo de entrenamiento; se recita una palabra, un conjunto de características se cuantifican, y se localizan los puntos finales de las palabras. Seguido de la detección del punto final un análisis LPC completo se realiza sobre cada trama de la palabra para dar un patrón de prueba $T(n)$, $n=1, 2, \ldots NT$ para ser utilizado en el algoritmo DTW. Este patrón de prueba es óptimamente alineado con cada uno de los patrones de referencia, dando una distancia total para cada comparación D_i, $i = 1, 2, \ldots, V$. La regla de decisión ordena las distancias y provee un mejor candidato o un conjunto de candidatos basados en la reglas NN o KNN.

4.2. Resultados del reconocimiento de palabras aisladas

El sistema de reconocimiento de palabras aisladas fue usado en un variado y amplio campo de pruebas de evaluación y los resultados de esas pruebas se muestran en las siguientes tablas. Estas tablas dan el porcentaje de exactitud en el reconocimiento para un vocabulario que constó de los 10 dígitos del sistema numérico en el idioma Ingles. A primera vista las tablas muestran una la exactitud de reconocimiento que varia desde 61.8 hasta 99.5 por ciento, y algo muy importante que cabe destacar, la exactitud no es función del tamaño del vocabulario, sino de la complejidad del vocabulario. Así un vocabulario de tamaño pequeño que consista de varios sonidos similares es considerablemente más complejo que un vocabulario de 200 palabras polisilábicas extremadamente diferentes.

Experimento 1

		ENTRENAMIENTO									
PRUEBAS	Dígitos	0	1	2	3	4	5	6	7	8	9
	0	160									
	1		160								
	2			160							
	3				159	1					
	4					156					
	5		3				157				
	6							160			
	7			2					158		
	8									160	
	9		2								158

Experimento 1. Porcentaje de reconocimiento acertado 99.5%

En esta prueba primero se aplicó un filtrado de pre-énfasis (a=.95) en todas las palabras antes de que estas se sometieran al proceso de detección de inicio y fin de palabra, se procesó la señal en un sistema LPC. Después se fijó la restricción del paralelogramo (DTW), que consistió en una pendiente de $\frac{1}{2}$. Y al final de proceso la regla de decisión utilizada es la NN, que toma la distancia normalizada mínima resultante de entre todas las comparaciones con los patrones de referencia. La normalización consiste en dividir la distancia obtenida del proceso DTW entre el número de vectores LPC que representan al patrón de referencia, y esta será la distancia normalizada

Como muestra la tabla, el porcentaje de errores es pequeño y que estos pequeños errores muestran la importancia que existe en la aplicación del filtrado de pre-énfasis y la normalización de la distancia. En comparación, el experimento 5 muestra un reconocimiento sin la aplicación del pre-énfasis y sin normalización mostrando una gran diferencia en el porcentaje de reconocimiento alcanzado. Este experimento muestra el mejor porcentaje alcanzado durante todas las variantes aplicadas al sistema de reconocimiento.

Experimento 2

	ENTRENAMIENTO										
PRUEBAS		0	1	2	3	4	5	6	7	8	9
	0	160									
	1		160								
	2			160							
	3				157	3					
	4					160					
	5						160				
	6							160			
	7						4		156		
	8									160	
	9		3		1						154

Experimento 2. Porcentaje de reconocimiento acertado 99.3%

En esta tabla se aplicó de igual forma un filtrado de pre-énfasis (a=9.5) antes de pasar a la detección del inicio y fin de palabra. La restricción en la pendiente del algoritmo DTW fue de $\frac{1}{2}$. La única variación que hubo en esta prueba fue la toma de decisión, se empleó la regla KNN.

El empleo de esta regla en cierta forma desfavoreció el porcentaje de reconocimiento empleado. De los resultados analizados, los errores se debieron al establecer la lista de las primeras cinco distancias mínimas (normalizadas). Se escogió como palabra reconocida la palabra que apreció con el mayor número de repeticiones dentro de la tabla. En el caso de no haber ninguna repetición se tomo el que tuviera la menor distancia de ellas. Como se puede observar, no existe una diferencia muy grande entre los porcentajes de reconocimiento de este experimento y la del experimento 1.

Experimento 3

		ENTRENAMIENTO									
		0	1	2	3	4	5	6	7	8	9
PRUEBAS	0	159		1							
	1		158	2							
	2			160							
	3				160						
	4					156	1				
	5		3				156				1
	6		1					159			
	7			2					158		
	8									160	
	9		4	1	1		1	1			152

Experimento 3. Porcentaje de reconocimiento acertado 98.81%

En este experimento la regla de decisión NN fue empleada y se conservaron las mismas características empleadas para el proceso de los experimentos 1 y 2; sólo que ahora la distancia mínima no fue normalizada con respecto al número de vectores muestras del patrón de prueba. Lo cual arrojó un menor porcentaje de reconocimiento aunque todavía sin considerarse malo. Lo claro de este experimento es que la normalización de las distancias favorece a que el reconocimiento sea más confiable.

Experimento 4

		ENTRENAMIENTO									
		0	1	2	3	4	5	6	7	8	9
PRUEBAS	0	160									
	1		158								2
	2			160							
	3				159	1					
	4					160					
	5						158				2
	6							160			
	7			1			4		159		
	8									160	
	9		4				1				155

Experimento 4. Porcentaje de reconocimiento logrado 99%

En este experimento se aplicó las mismas características utilizadas en el experimento 3, excepto por la regla de decisión. En este experimento la regla KNN fue empleada, pero la lista de las distancias mínimas (con normalización) se incrementó a 8 integrantes, por supuesto que esto hizo que existieran más opciones de escoger de entre la lista de palabras más posibles

candidatos a ser reconocidos. De este experimento se puede observar que conforme se incrementó el número de integrantes de la lista habrá más errores, pero aún así el porcentaje de reconocimiento sigue siendo confiable.

Experimento 5

	ENTRENAMIENTO										
PRUEBAS		0	1	2	3	4	5	6	7	8	9
	0	55		10		50		1	44		
	1		104	3		51			1	1	
	2	1	19	78	1	30	1	2	27		1
	3	2	18	7	80	26	1	5	13	7	1
	4		3			153		1	3		
	5	4	15	12	2	29	83	1	10	3	1
	6	3	2	1				112	38	4	
	7	7	4	9		8		5	126	1	
	8	7	10	17	1	16		19	12	78	
	9	2	8	7	5	7	3	3	3	2	120

Experimento 5. Porcentaje de reconocimiento logrado 61.8%

En este experimento a manera ilustrativa se muestra como son determinantes los factores de la normalización, el filtro de pre-énfasis. Aquí nunca se aplicó un filtro de pre-énfasis lo cual hizo que muchos de los sonidos sordos de las palabras perdieran la oportunidad de ser realzados y cuantificados, es decir se escapo información valiosa en aquellas palabras que contienen sonidos sordos. Adjunto a esto, tampoco se le aplicó una normalización a las distancias, que como se ha visto en los experimentos anteriores ésta mejora la exactitud en el reconocimiento. Para este experimento se utilizó la regla de decisión NN, ya que la regla KNN en este caso arrojo aún resultados mucho más catastróficos.

4.3. Reconocimiento de palabras conectadas

Para resolver el problema del reconocimiento de palabras conectadas varios investigadores han propuesto como solución diferentes algoritmos. Los algoritmos analizados en este capitulo son los propuestos por Sakoe, Rabiner y Ney. Los métodos basados en reglas de segmentación o de segmentación por estadística requieren de un conocimiento detallado del vocabulario así como una clara dependencia a las reglas de segmentación o de un entrenamiento exhaustivo en el algoritmo de segmentación. Sakoe de forma independiente desarrolló un algoritmo de dos niveles para resolver la optimización de este problema. Sobre el primer nivel, todos los patrones de referencia son sistemáticamente comparados con todos las posibles subsecciones del patrón. La comparación es igual al empleado en el reconocimiento de palabras aisladas. Sobre el segundo nivel, usando la lista de las distancias acumuladas generadas, la estimación óptima de la secuencia desconocida de palabras es obtenida por la minimización de la distancia total de todas las secuencias posibles de palabras. Por otro lado, Myers y Rabiner derivaron otra posible solución, ellos hicieron uso de la comparación de todas las posibles secuencias de palabras que pueden ser realizadas por la sucesiva concatenación de los patrones de referencia. Así ellos obtuvieron lo que llamaron un algoritmo de niveles.

El método de Hermann Ney desarrollado en este trabajo, es esencialmente un tutorial, su propósito es de presentar una clara descripción del problema de reconocimiento de palabras conectadas y su solución en un sólo paso utilizando un algoritmo de un estado y compararlo con otros algoritmos. El algoritmo de un estado es básicamente derivado de la parametrización de la función de ajuste dinámico en el tiempo por un sencillo índice, y tratando el criterio de optimización directamente como función del la minimización de las distancias generadas entre las comparaciones. Esta aproximación es mucho más simple que las aproximaciones propuestas por Sakoe y Rabiner y computacionalmente resulta más eficiente el algoritmo de Ney. Las restricciones impuestas sobre el camino son descritas en términos de dos tipos de transiciones: reglas de transición para el interior de la palabra y para las fronteras entre cada palabra dentro de la cadena. Llevando acabo la optimización usando esas restricciones en el camino y la programación dinámica deja al algoritmo de un estado, para el cual no hay múltiples niveles de optimización como en las otras aproximaciones. En comparación con el método propuesto por Rabiner, el algoritmo de un estado no necesita de un número máximo preespecificado de palabras en la cadena de entrada. Lo que es más importante en el algoritmo de un estado es que no requiere más tiempo de computo que el correspondiente para el caso del reconocimiento de palabras conectadas sin ninguna restricción de ajuste. Los aspectos de la implementación de este algoritmo son descritos, y sus requerimientos de computo y de almacenamiento son comparados con el algoritmo de 2 niveles (Sakoe) y el de Rabiner.

Método de Rabiner

Para comprender como es que se resuelve el reconocimiento de una cadena de palabras, es necesario definir algunos términos. Se denota a la cadena desconocida de palabras que va ha ser reconocida como $T(m) = 1, 2, ..., M$ donde $T(m)$ representa una cadena de palabras de desconocida longitud. La cadena de prueba $T(m)$ se registra en el tiempo con una secuencia de L patrones de referencia $Rq_{(1)}(n), Rq_{(2)}(n), ..., Rq_{(L)}(n)$, donde cada $Rq_{(k)}$, $k= 1, 2, ..., L$ es uno de un conjunto de V patrones de referencia (las palabras del vocabulario) R_v, $v = 1, 2, ..., V$. la longitud del vth patrón de referencia se denota como N_v. Se define un super patrón de referencia $R_{q(1)q(2)}{}^{S}{}_{... q(L)}$ como la concatenación de los L patrones de referencia $R_{q(1)q(2) ... q(L)}$, ejemplo:

$$R^S = R_{q(1)} + R_{q(2)} + ... + R_{q(L)} \qquad 4.1$$

(4.1)

de la forma

$$R^S(n): \begin{cases} = R_{q(1)} + (n - \phi(0)) & 1+\phi(0) \le n \le \phi(1) \\ = R_{q(2)} + (n - \phi(1)) & 1+\phi(1) \le n \le \phi(2) \\ \\ = R_{q(L)} + (n - \phi(L-1)) & 1+\phi(L-1) \le n \le \phi(L) \end{cases} \qquad 4.2$$

donde la función de longitud $\phi(l)$ se define como

$$\phi(l) = \sum_{k=1}^{l} N_{q(k)} \qquad 4.3$$
$$\phi(0) = 0.$$

si se procura la comparación por medio del algoritmo DTW entre T(m) y R^S (n), y definiendo la distancia resultante al el final de la comparación como $D_{q(1)q(2)...q(L)}(M)$, entonces la solución ideal al problema será la minimización de

$$D^* = \min_{L} \left[\min_{q(1)q(2)...q(L)} \left[D_{q(1)q(2)...q(L)} \right] \right] \qquad 4.4$$

La cadena de palabras de referencia de longitud L, que minimiza la distancia acumulada es la mejor estimación de la cadena de prueba. Debería ser claro que, excepto para los casos triviales, la exhaustiva solución de la ecuación (4.18) no es práctica debido a los cálculos computacionales que se involucran. Por ejemplo para L=5, V=10 (un vocabulario con 5 palabras en una cadena), un total de NS = 105 (cadena de 5 palabras) + 104 (cadena de 4 palabras) +....+ 10 (cadena de una palabra) que tendrían que ser probadas. Como tal, han sido propuestas varias aproximaciones para resolver la ecuación. 4.4. Esas aproximaciones se enfatizan en reducir los cálculos computacionales resolviendo la minimización en una de estados (niveles) en que suficiente información es retenida así que unas series de cadenas de posibles candidatos es evaluada. Generalmente, la mejor cadena es retenida como una de los candidatos y la mayoría de las cadenas con una pobre distancia son descartados.

La figura 4 muestra el caso simple para obtener el super patrón de referencia $R^s=R_q(1)+R_q(2)+...+R_q(L)$ para índices fijos $q(1)q(2)...q(L)$. La restricción del algoritmo DTW es un paralelogramo en donde se comparan T y R^s, exactamente el mismo utilizado para el reconocimiento de palabras aisladas. Para encontrar el camino más óptimo dentro del paralelogramo, se llevaba acabo por líneas verticales. La figura 4(b) muestra un conjunto de líneas que han sido trazadas para diferentes fines de tramas en el interior de las referencias R. Para este caso los cálculos están dados en líneas verticales nuevamente, sin embargo, la línea horizontal formada por el fin de cada referencia forma una restricción sobre la región $G1$ cubierta del paralelogramo. La manera correcta para recoger los cálculos para el segundo patrón de referencia (región $G2$) los scores de las distancias acumuladas para todos los caminos que terminan al final de la primera línea horizontal deben ser retenidas, y usadas como condiciones iniciales sobres las distancias. De esta forma, los mismos cálculos, como son mostrados en la figura 4(a) pueden ser llevados acabo por niveles (palabras dentro de una secuencia de patrones de referencia) en una serie de cálculos.

La relevancia de los resultados de este algoritmo de niveles es que aproxima a encontrar el mejor camino dinámico (encontrando el mejor camino para cada patrón de referencia en la secuencia) puede ser extendido al caso de más de un patrón de referencia a cada nivel, calculando todas las distancias posibles al final de cada nivel y reteniendo la mejor distancia para cada índice m. Así, si definimos un rango comenzando por la variable $m_1(l)$ y terminando por la variable $m_2(l)$, entonces para resolver la mejor comparación con el mejor patrón, cada valor de m en el rango de $m_1(l)<=m<=m_2(l)$, al nivel l, debemos guardar rastro de las tres cantidades como sigue.

1) Distancia mínima acumulada, $D_l^B(m)=min_{1<=v<=V}[D_l^v(m)]$ donde $D_l^v(m)$ es la distancia acumulada para el vth patrón de referencia, al nivel l terminando a la trama m del patrón de referencia.
2) Mejor referencia, $W_l(m)=argmin_{1<=v<=V}[D_l^v(m)]$, el patrón de referencia dejando a la mínima distancia acumulada.
3) Rastreo del camino, $F_l^B(m)= F_l^{w(l(m)}(m)$, donde $F_l^v(m)$ es la trama del mejor patrón al nivel $l-1$ al cual el mejor camino para la trama de prueba m, al nivel l, para patrón de referencia R_v, terminada.

La figura 4c ilustra la operación del algoritmo para un caso simple en que se asume sólo hay dos patrones de referencia, denotados como A y B, los dos con la misma longitud. Esto es asumiendo

que la cadena es de longitud $L=4$ y que se conozca lo que va ser recitado. Al final del primer nivel, hay 6 posibles valores finales de m, y el patrón de referencia que tenga la distancia más chica se denota a lo largo de la línea horizontal al final del nivel. Similarmente en el nivel 2 y 3 los mejores caminos para cada posible final son denotados por la referencia, a ese nivel, queda la distancia mínima acumulada. Finalmente, al nivel 6, solo un único camino es retenido, este es el camino óptimo que minimiza la distancia total. Para determinar la mejor cadena de comparación, se debe rastrear la mejor secuencia desde $m=M$ hasta el inicio, queda como secuencia *AAABA* para los cuatro patrones de referencia comparados con el patrón de prueba. También en la figura 5 hay valores en la cadena de prueba que corresponden al final de cada referencia en la mejor secuencia de comparación. En principio; esos valores pueden ser usados como la mejor estimación de segmentación de las palabras dentro del patrón de prueba.

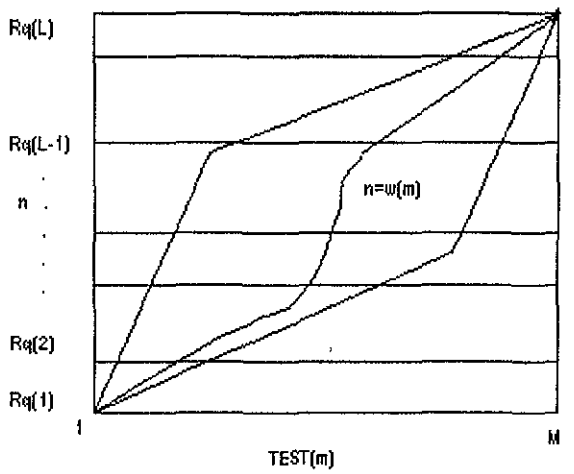

Figura 4a Se muestra la restricción de fin de palabra del patrón de referencia con una frase

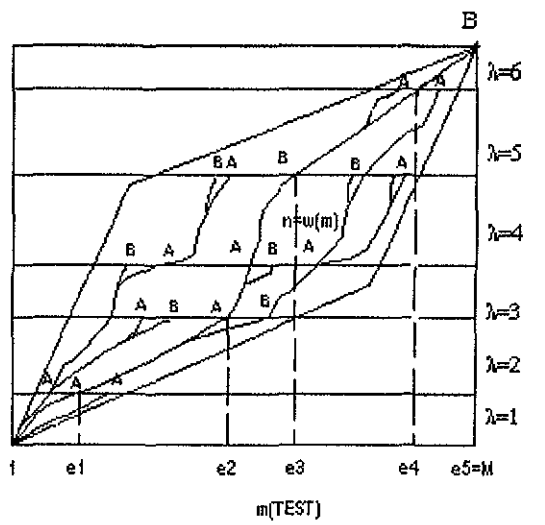

Figura 4c Ilustración del mejor camino DTW en una comparación de cuatro palabras

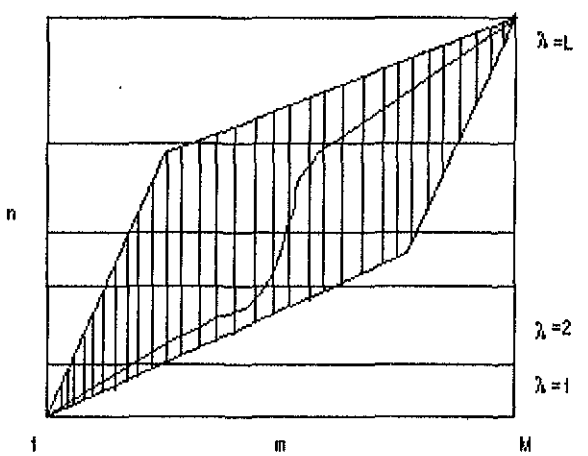

Figura 4b Implementación del algoritmo de fin de palabra DTW

Método de Sakoe

Para el algoritmo propuesto tenemos *1,2,...,n,...,N* representa un vocabulario *N*. Los patrones de referencia de la palabra n es representada como

$$B^n=b^n{}_1,b^n{}_2,...,b^n{}_j,...,b^n{}_jn. \qquad 4.5$$

tenemos,

$$C=c_1,c_2,...,c_i,...,c_I \qquad 4.6$$

Que es patrón desconocido de voz a la entrada. Esta entrada puede ser una sola palabra o múltiples palabras. De aquí llamada palabras conectadas o una frase. El concepto básico del principio de reconocimiento está mostrado en la figura 4c. El operador "+" es empleado (significando concatenación de dos patrones) como, por ejemplo

$$B^m+B^n=b^m{}_1,b^m{}_2,...,b^m{}_j{}^m,b^n{}_1,b^n{}_2,...,b^n{}_j{}^n \qquad 4.7$$

Una frase de patrones de referencia B de palabras *n(1),n(2),...,n(k)* es sintetizada concatenando sus patrones de referencia como

$$B=B^{n(1)}+B^{n(2)}+...+B^{n(k)} \qquad 4.8$$

La comparación entre los patrones es hecha entre el patrón de entrada *C* desconocido y el patrón de referencia sintetizado *B*, dando una distancia *D(C,B)*. Esos procesos son repetidos, cambiando el número de palabras *k* y los índices *n(1), n(2),...,n(k)*. Cuando este proceso es llevado acabo hasta todas las permutaciones repetidas de los índices, los parámetros óptimos *k=k'* y *n(x)=n'(x), x=1,2,...k'* son determinados, y que dan una distancia mínima *D(C,B)*. Entonces, la decisión es hecha que el patrón de entrada *C* comprende *k'* palabras *n'(1),n'(2',...,n'(k'))*. Matemáticamente, este principio es formulado en la minimización del siguiente problema.

$$T=mink_{,n(x)}[D(C,B^{n(1)}+B^{n(2)}+...+B^{n(x)}+...B^{n(k)})] \qquad 4.9$$

Este esquema de reconocimiento no necesita de una segmentación preliminar porque la comparación entre los patrones es hecha sobre una frase completa. Así, la posibilidad de reconocimientos errados causados por la inexactitud de la segmentación es completamente excluida. La minimizacion de la ecuación (4.4), no es un problema fácil de resolver por el método

de exhaustivas comparaciones. Este realiza muchas operaciones debido a todas las posibles concatenaciones de los patrones de referencia que deben ser probados.

La minimización de la ecuación 4.9 es llevado acabo en dos pasos: uno para la unidad de palabra nivel y la otra para el completo voz conectada nivel (o nivel frase). Un patrón parcial *C(l,m)* para un patrón de entrada *C* es definido como

$$C(l,m)=c_{l+1},c_{l+2},\ldots,c_m \qquad 4.10$$

(k-1) las fronteras de las palabras *l(1), l(2),...,l(k-1)* son asumidas sobre el eje del tiempo del patrón de entrada *C* llevando a éste a dividirse en *k* patrones parciales.

$$C=C(l(0),l(1)+C(l(1)l(2))+\ldots+c(l(k-1),l(k)) \qquad 4.11$$

Donde *l(0)=0* y *l(k)=I*. La forma asimétrica para calcular la distancia mantiene las siguientes propiedades relacionadas al patrón parcial

$$D(C,B^m+B^n)=min[D(C(0,l),B^m)+D(C(l,l),B^n)] \qquad 4.12$$

Poniendo (4.12) en (4.9) y aplicando repetidamente la relación en 11, se obtiene

$$Tmin_{k,l(x)}=\left[\sum_{x=1}^{k}\min_{n(x)}[D(l(x-1),l(x),n(x))]\right] \qquad 4.13$$

Donde la notación *D(l,m,n)* es una abreviación de *D(C(l,m),Bn)*, que es una distancia entre el patrón parcial *C(l,m)* y el patrón de referencia *Bn*. Hay dos problemas de minimización involucrados en 4.13. Ellos son resueltos por siguiente algoritmo.

1) Comparación por nivel-palabra: Calcula y memoriza la Distancia parcial

$$D'(l,m)=\min_n[D(l,m,n)] \qquad 4.14$$

Decisión parcial

$$N'(l,m)=argmin_n[D(l,m,n)] \qquad 4.15$$

Para cada combinación de *l* y *m*, donde $0 \le l \le m \le I$. (El operador "argmin" significa encontrar el parámetro óptimo *n*.)

2) Comparación por nivel-frase. Resuelve la minización

$$T_k = \min_{l(k)} \left[\sum D'(l(x-1), l(x)) \right] \qquad 4.16$$

Y entonces,

$$T = \min_k [Tk] \qquad 4.17$$

Llevando a los parámetros óptimos $k=k'$ y $l(x)=l'(x)$, donde $x=1,2,...,k'$. Así, la minimización de la ecuación4.16 se lleva acabo en dos pasos. Resultando los parámetros óptimos k' y $l'(k)$, con las decisiones parciales $N'(l,m)$, da el siguiente resultado de reconocimiento.

Decisión haciendo proceso:

$$n'(x) = N'(l'(x-1), l'(x)) \qquad 4.18$$

donde

$$x=1,2,...,k'$$

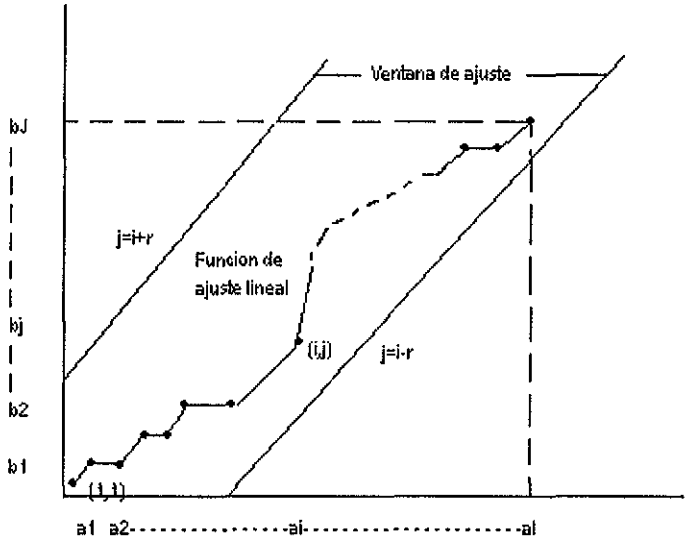

Figura 4d Principio de comparación

Es imposible de discutir en detalle las diferentes aproximaciones de los algoritmos DTW para el reconocimiento de palabras conectadas. Sin embargo en el siguiente punto se ilustra la sofisticación de esos métodos, así como una breve discusión de la aproximación utilizada en este trabajo.

Método de Hermann Ney

La técnica para el reconocimiento de palabras conectadas consiste básicamente en el mismo empleado para el reconocimiento de palabras aisladas, claro que con unas variantes.

Una de las mejores soluciones para el reconocimiento de palabras y que funciona bien es la técnica de la programación dinámica. Para el reconocimiento de palabras aisladas, varios autores han mostrado que el problema del alineamiento no lineal entre los patrones de voz que poseen una escala arbitraria, puede ser resuelto efectivamente por la programación dinámica. El método que se ha elegido para el reconocimiento de palabras conectadas es una optimización del mismo problema presentado para el reconocimiento de palabras aisladas. Una de las características que hacen más atractiva esta formulación es la pequeña cantidad de información que se necesita saber de ante mano: tan sólo es necesario conocer los patrones de referencia, éstos estarán presentados en palabras aisladas. Otra ventaja es que las tres primordiales operaciones que ocurren en el reconocimiento de palabras conectadas suceden de manera simultanea. Es decir, la segmentación de cada una de las palabras que están dentro de la cadena, el alineamiento no lineal y el reconocimiento son realizados al mismo tiempo. De esta forma, los errores en el reconocimiento debidos a los errores en las fronteras entre palabras o el alineamiento en el tiempo no son posibles.

El algoritmo es forzado a comparar las palabras completas, y como resultado de esto, las fronteras de las palabras son determinadas automáticamente.

En el reconocimiento de palabras conectadas la entrada al sistema será una secuencia de palabras de un vocabulario específico, y el reconocimiento se basa en la comparación de los patrones de referencia, en este trabajo se realizó con un vocabulario que contiene el conjunto de los 10 dígitos de (0 a 9). Para desarrollar los reconocimientos dentro de las cadenas se utilizó una base de datos conteniendo 1000 patrones de referencia, 100 patrones para cada dígito; por otro lado las cadenas de 2,3,4,5,7 dígitos constaron cada una de 150 palabras conectadas.

Un diagrama de bloques de un reconocimiento de patrones aplicado a la solución de este problema se muestra en la Fig 4d. Este diagrama de bloques es casi idéntico al utilizado en el reconocimiento de palabras aisladas, con una gran excepción. Esta excepción es la retro alimentación que esta en un ciclo continuo mostrada a la salida del bloque que contiene la regla de decisión, en el final de cada reconocimiento (nivel), alimenta de nuevo al algoritmo DTW con un conjunto de estimaciones de donde han sido acoplados los finales de cada palabra de la cadena. De esta manera, el algoritmo DTW puede progresivamente construir un conjunto de comparaciones de la cadena de prueba, y al final de cada búsqueda, determinar una lista ordenada de las distancias acumuladas que hubo. Los demás componentes para el reconocimiento de palabras conectadas como, el analizador acústico, los patrones de referencia, el algoritmo DTW,

y la regla de decisión son esencialmente las mismas usadas para el reconocimiento de palabras aisladas. Así, un punto clave acerca de este tipo de reconocimiento es que los patrones de referencia son los patrones de la mismas palabras usadas en el vocabulario pero aisladas. El otro punto que hay que hacer notar es que el algoritmo DTW no puede ser restringido, es decir que la detección del punto final de las palabras no tiene fronteras.

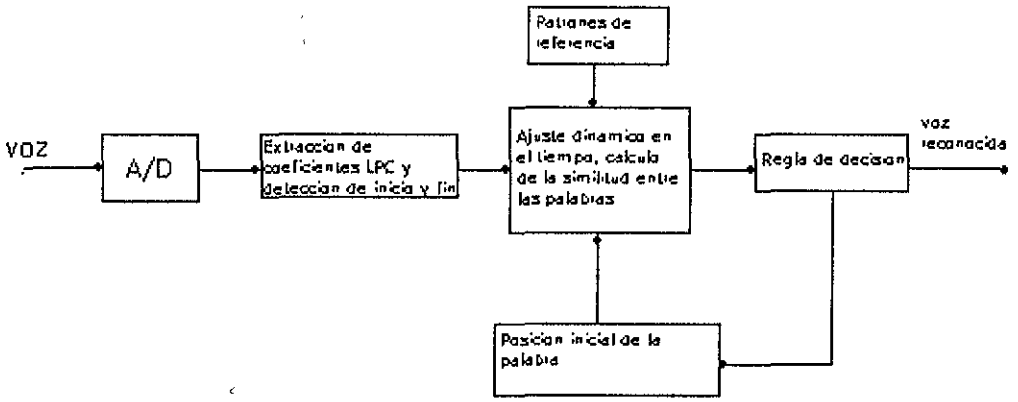

Fig. 4d Diagrama de bloques para un sistema de reconocimiento de palabras conectadas

4.4. La formulación del problema de la unión de los patrones

Aquí se presenta una simple aproximación del problema de comparar los patrones en cadenas de palabras para el reconocimiento de palabras conectadas.

Asumimos una entrada desconocida o un patrón de prueba que consiste de $i = 1, ..., N$ tramas, donde una trama se representa por un vector de características. Del patrón de entrada se conoce previamente que está conformado por palabras individuales que son escogidas de un vocabulario previamente conocido y especificado. Las palabras del vocabulario corresponden a un conjunto de k patrones o modelos de referencia obtenidos de palabras recitadas aisladamente. Éstos modelos de referencia se distinguen por el índice $k = 1, ..., K$. Las tramas de los modelos de referencia se denotan como $j = 1, ..., j(k)$, donde $j(k)$ es la longitud del modelo k.

La última meta del reconocimiento de palabras conectadas es determinar que secuencia $q(1), ...,q\ (R)$ de modelos de prueba que se aproxime al patrón de entrada. Donde el criterio de comparación necesita de más especificaciones. La concatenación de los patrones $q(1), ...,q(R)$ se refiere a un super patrón de referencia que puede ser manejado como un solo patrón pronunciado, el proceso de comparaciones es el mismo usado para el caso del reconocimiento de palabras aisladas. Pero lo que se propone para el procedimiento de comparación se encuentra detallado en la Fig. 4a. La idea básica de ésta figura es que las tramas i del patrón de entrada y de las tramas j de cada modelo de referencia k definen un conjunto de puntos en el espacio de tres dimensiones (i, j, k). Cada coordenada $(i\ j,\ k)$ en el plano es asociado con una cuantificación de disimilitud local de esa coordenada $d(i, j, k)$ entre los correspondiente eventos acústicos. El problema del este tipo de reconocimiento puede ser visto como encontrar el camino a través del conjunto de coordenadas $(i\ j,\ k)$ que conforman en espacio y que provea la mejor comparación entre el patrón de prueba y la secuencia desconocida de los modelos de prueba. Los tres parámetros $i\ j,\ k$ tienen diferentes caracteres: Los parámetros del tiempo i y j tienden a cambiar más o menos uniformemente en orden ascendente, mientras el número del modelo k es constante para largas subsecuencias del camino óptimo y que puede sólo cambiar después de que el camino haya pasado la frontera del mismo modelo con $j = J(k)$. Sin embargo, para desglosar bien el funcionamiento del algoritmo, es crucial el tratar estos tres parámetros matemáticamente equivalentes. Formalmente el camino W se determina como una secuencia de coordenadas

$$W = (w(1),w(2), ..., w(1), ..., w(L)) \qquad 4.19$$

Donde $w(l) = (i(l), j(l), k(l))$ y l es el parámetro del camino para indexar el conjunto ordenado de las coordenadas del camino. La distancia global es el criterio para hacer la comparación de similitud. Es decir, la suma de todas las distancias locales a lo largo del camino. El problema del reconocimiento de cadenas de palabras puede ser estandarizado a la minimización del problema

$$\min_{w} \sum d(w(l)) \qquad 4.20$$

Minimizar la distancia global con respecto a todos los caminos permitidos. Del mejor camino, la secuencia asociada de los modelos puede ser únicamente recobrada como está claramente ilustrada en la Fig. 4e

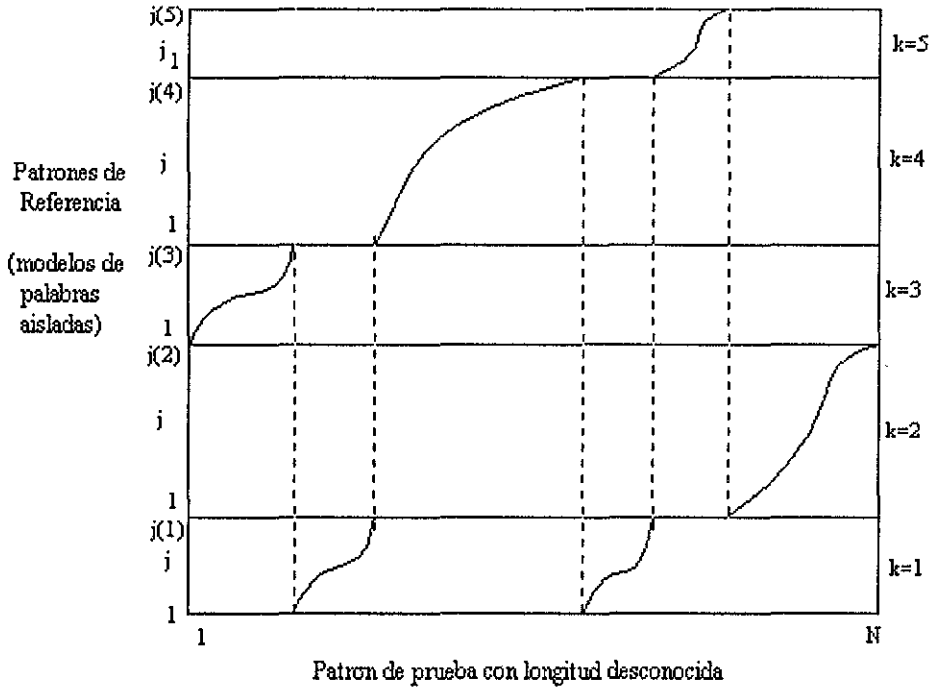

Fig. 4e. El problema del reconocimiento de palabras conectadas.
El camino óptimo provee la secuencia desconocida de palabras
así también como el alineamiento no lineal entre las correspondientes
Secuencias de los modelos de referencia y el patrón de entrada.

En adición a la minimización de la distancia global, el ajuste dinámico en el tiempo requiere de obedecer ciertas restricciones de continuidad que están implicadas en la propia naturaleza física de los patrones a ser comparados. Las restricciones surgen de la necesidad de preservar el orden en el tiempo y en todo lo largo de los ejes del tiempo, así como el de que exista una continuidad implicando que ningún evento acústico sea omitido en la secuencia *i(1), ..., i(l), ..., i(L).* La restricción de continuidad determina las posibles coordenadas sucesoras *(i, j, k).* Una posible desventaja de la definición de la distancia global es que ella depende de la longitud del camino, de esta manera, los caminos más chicos serán favorecidos. Debido a la concatenación de las palabras modelo (aisladas) para formar el super patrón de referencia. Es conveniente distinguir los dos tipos de regla de transición: la regla de transición del DTW en el interior de

la palabra modelo y el patrón de prueba y la regla de transición entre las fronteras de los modelos de referencia. Para la regla de transición en el interior del modelo se siguen las mismas restricciones utilizadas en el reconocimiento de palabras aisladas, estas mantienen la siguiente relación entre dos coordenadas consecutivas,

$$Si\ w(l) = (i, j, k), j > 1$$

$$W(l-1) \in \{(i-1,j,k),(i-1,j-1,k),(i,j-1,k)\}$$

El punto *(i, j, k)* solo puede ser alcanzado de uno de los *puntos (i-1, j, k), (i-1, j-1, k), (i j-1, k)* y para la regla de fronteras donde la concatenación de las coordenadas de la palabra que ha sido reconocida y las coordenadas de los siguientes modelos que son candidatos. La trama inicial de la palabra modelo k, debe ser alcanzada por la trama final de cualquier palabra modelo k* incluyendo la misma k. Esto depende fuertemente en como las palabras aisladas pueden ser combinadas y como deben ser cambiadas en caso que se tengan restricciones del tipo sintáctico. El problema de la coarticulación en las fronteras es vencido uniendo las partes interiores de las palabras de forma que esto se haga en el proceso automáticamente. Finalmente, hay restricciones en la detección de los puntos finales que requieren que el camino del algoritmo DTW comience en la primera trama de cualquier palabra modelo y que termine al final de cualquier palabra modelo.

4.5. Adaptación del algoritmo DTW

El objetivo final del algoritmo es determinar la secuencia desconocida de palabras. Para lograr esta meta, es suficiente saber en que trama *i* del patrón de prueba el mejor camino ha comenzado dado un punto final de una palabra modelo *k* dada.

Los detalles del mejor camino dentro de los modelos de referencia no son de mucha importancia para el problema del reconocimiento. Para ejecutar la recursión del ajuste no lineal para una trama *i*, sólo una pequeña porción del arreglo completo $D(i, j, k)$ de las distancias acumuladas se necesita, normalmente los elementos correspondientes a la trama sucesora *i*: $\{D(i-1, j, k): k = 1, ..., J(k)\}$. La coordenada asociada con esos elementos de un corte vertical a través del plano de la Fig. 4e. Esta columna de almacenaje se refiere simplemente a una columna con un arreglo de distancias acumuladas denotada como $D(j, k)$. Así, usando este tipo de arreglo la recursión puede ser llevada acabo procediendo a lo largo del eje del tiempo del patrón de prueba y actualizando la columna de almacenamiento punto por punto. Donde todo lo que necesitamos es la mínima distancia total como un score de comparaciones entre los modelos de referencia y el patrón de prueba. La diferencia esencial de este tipo de reconocimiento y el de palabras aisladas es que alguna forma el rastreo debe ser añadido para permitir al algoritmo recobrar la secuencia desconocida de palabras. Para el camino puntero de la coordenada (i, j, k), hay una única coordenada que comienza por la línea $j = 1$ para la misma palabra modelo *k*. Entonces, para cada coordenada, una coordenada sucesora $B(i, j, k)$ puede ser definida como el valor de la trama inicial de la palabra sucesora. La Fig. 4f muestra el concepto básico del rastreo para las tres coordenadas sucesoras del punto (i, j, k). Originalmente el arreglo de los caminos de rastreo depende de un índice triple (i, j, k) de la misma manera como el arreglo $D(i, j, k)$ de las distancias acumuladas.

Sólo se está interesado en el rastreo de los puntos $(i, j(k), k)$ en donde se hayan alcanzado los limites de frontera de las palabras modelo. Se puede reducir el arreglo de los puntos de rastreo $B(i, j, k)$ en un arreglo del tipo $B(j, k)$ y actualizarlo punto por punto sabiendo de donde ha venido el mejor camino, de este arreglo podemos rastrear las coordenadas potenciales en el eje del tiempo del patrón de prueba que han del mejor camino. En otras palabras, es necesario recabar un arreglo de distancias $D(i, j, k)$, y a la vez tener en el mismo arreglo sus correspondientes coordenadas que describen en donde empezó $B(i)$ y donde termino la frontera de la palabra modelo con la coordenada del patrón de prueba en su eje del tiempo $B(i)$. Con este arreglo se selecciona los primeros elementos del arreglo que están caracterizados por tener una distancia acumulada pequeña, entonces en este punto, se aplica una regla de decisión para escoger que palabra se ha reconocido para concatenarla en el super patrón, para esto se necesita hacer un arreglo de índices *k* que se actualiza inmediatamente después de haber extraído el índice *k* de la palabra reconocida. Después de tener la decisión, se selecciona del mismo arreglo de distancias acumuladas aquellas que tengan el mismo índice *k* y de ahí se sacan todos los índices $B(i)$ (estos indican en donde se acoplo la frontera de la palabra modelo con el eje del tiempo del patrón de prueba) se cuentan y se saca una estadística de cuantas veces se repitieron los índices $B(i)$. Al lograr esto, se consigue obtener un conjunto de coordenadas que potencialmente nos dirán donde ha terminado y donde ha comenzado una palabra dentro del patrón de prueba. La decisión de que punto sobre el eje del tiempo del patrón de prueba debe

ser considerada como el fin y el inicio de una palabra debería caer sobre el que tenga el mayor número de repeticiones (que casi siempre este índice coincide con el que tiene la distancia mínima acumulada), ya que estadísticamente refleja menos posibilidades de error en la decisión de inicio y fin de alguna palabra dentro del patrón de prueba. Con este índice determinado la recursión del algoritmo DTW comienza, así sucesivamente hasta que se terminen de rastrear todos los inicios y fin de palabras que implícitamente se va haciendo la concatenación de las palabras reconocidas debido al arreglo de los índices k que se hayan recabado que obviamente contiene la secuencia óptima de palabras.

El diagrama 1 muestra el esquema del algoritmo usado para el reconocimiento de palabras conectadas.

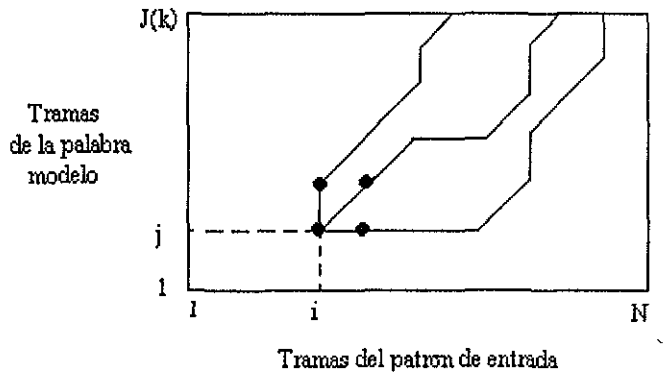

Fig. 4f Rastreo de las tres sucesoras coordenadas (i, j, k) hacia sus correspondientes tramas finales.

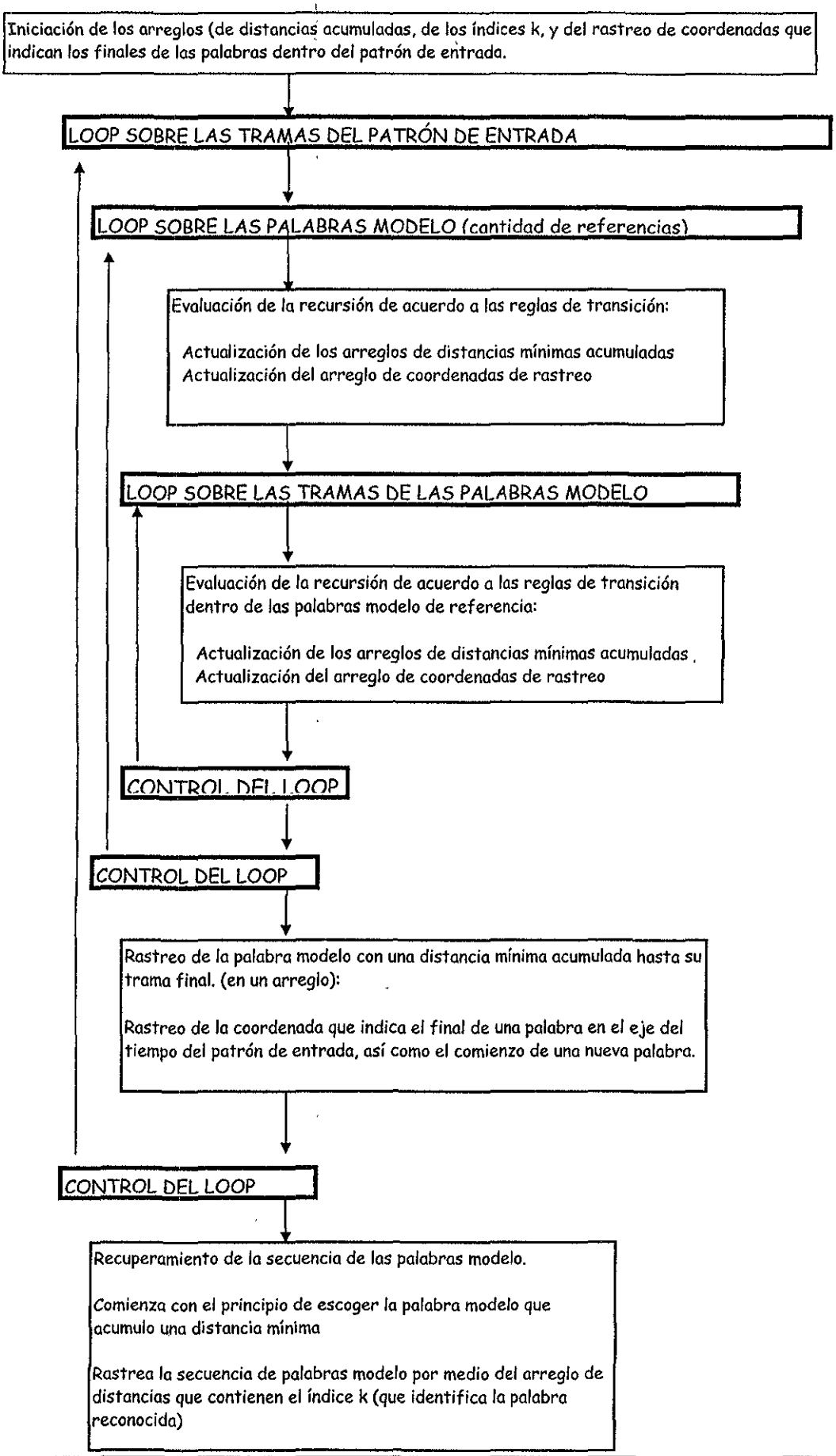

Diagrama 1,Esquematiza el algoritmo DTW empleado para el reconocimiento de palabras conectadas

4.6. Resultados del reconocimiento de palabras conectadas

Para estos experimentos se empleó una base de datos de referencia que contiene 10 repeticiones de cada dígito en inglés, los parlantes de los patrones de referencia son totalmente independientes de los parlantes de palabras conectadas, es decir, que ningún parlante que haya recitado un conjunto de palabras conectadas aparecerá como parlante en los patrones de referencia.

Hay un punto que es importante destacar, es el que no se haya considerado seleccionar algún tipo de ruido como patrón de referencia. En algunos otros sistemas se propone el uso de un patrón de referencia que este constituido solamente por ruido de fondo, ya que con esto se podría identificar los espacios de ruido (silencios) que existen entre palabra y palabra.

En el primer experimento la propuesta para resolver estos problemas fue tratar de hacer una segmentación óptima tratando de encontrar los inicios y finales de cada palabra conforme el algoritmo DTW fuera identificando una palabra reconocida; en otras palabras, una vez detectada una palabra el algoritmo nos dice cual es el fin de la palabra y a partir de esa coordenada el algoritmo comienza a buscar hacia delante y hacia atrás el inicio de la siguiente palabra, en donde se rebase un umbral de cruces por cero (representativo del ruido), justo en esta coordenada el algoritmo busca reconocer otra palabra en una vecindad de coordenadas alrededor de ella. La vecindad de coordenadas se toman como posibles inicios de palabras y en cada coordenada se hará el reconocimiento y si el reconocimiento arroja como identificada dos veces la misma palabra entonces esa será la palabra reconocida y el inicio de palabra queda identificada por los puntos de la vecindad.

Por otro lado, la base de datos de prueba contiene cadenas de palabras desde dos hasta cinco dígitos. El algoritmo empleado ha sido el que previamente se analizó propuesto por H. Ney, por ser efectivo y fácil de implementar. Es muy importante destacar que el algoritmo DTW no tiene ningún tipo de restricción en la pendiente ni uso del paralelogramo, esto se debe a que las palabras que están dentro de la cadena tienen una duración de tiempo mucho más corta que las palabras que están en la base de referencia. El favorecer una pendiente sería arbitrario y sin justificación; aunque si se hicieron los experimentos con restricciones en la pendiente y los resultados fueron muy desfavorables ya que los resultados obtenidos fueron casi de un 100% de error en la cadena de palabras reconocida.

Otro punto que es necesario resaltar, es el uso de una base de datos de referencia que es rica en parlantes y no el de una base de datos con muchas repeticiones de la misma palabra. Se hicieron experimentos con la base de datos completa y con una base de datos pequeña. La base completa constó de 10 parlantes que recitaron 10 repeticiones de cada dígito, un total de 1000 palabras de referencia. La base pequeña constó de igual forma los mismos 10 parlantes pero ahora solo recitaron una vez cada dígito, un total de 100 palabras de referencia. Los resultados obtenidos demostraron tener la misma efectividad el usar una base más pequeña y por lo tanto el tiempo se reduce resultando más ventajoso para el sistema.

Los siguientes puntos muestran los factores determinantes del sistema de reconocimiento de palabras conectadas.

- Es necesario eliminar el ruido que hay al comienzo de la primera palabra dentro de la cadena
- La cadena se pasa a través del proceso LPC con las mismas características con las que fueron procesadas las palabras que componen la base de referencia
- Se utiliza el mismo algoritmo DTW utilizado para el reconocimiento de palabras aisladas excepto por:
  - No hay restricción en las pendientes, dentro del algoritmo DTW
  - No existe ningún tipo de paralelogramo dentro del algoritmo DTW
- No es necesario conocer la cantidad de palabras que están dentro de la cadena
- No es necesario hacer previamente una segmentación a la cadena, el algoritmo lo hace automático

El factor determinante para un buen reconocimiento dentro de una cadena de palabras, es la buena segmentación de cada una de las palabras que integran a dicha cadena. Las siguientes tablas que muestran los resultados de los experimentos en donde la segmentación fue determinada de forma automática por el mismo algoritmo conforme fuese detectando una palabra reconocida.

Primer Experimento

Experimento para cadenas de 2 dígitos

Longitud de la cadena	0 errores	1 error	2 errores
2	140	10	0
Expresado en porcentajes	93.3%	6.7%	0%

En este experimento fue lógico que se hayan obtenido los mejores resultados de entre todos los demás experimentos. Porque en estas cadenas es fácil segmentar cada una de las palabras, puesto que existe un espacio de tiempo entre una palabra y la otra. Los pequeños errores se han debido a que definitivamente el sistema de reconocimiento se equivocó, pero de aquí se puede ver que la segmentación juega un factor determinante para que el reconocimiento sea favorable. En los siguientes resultados se observa que conforme tenga más palabras la cadena el reconocimiento será más inexacto.

Experimento para cadenas de 3 dígitos

Longitud de la cadena	0 errores	1 error	2 errores	3 errores
3	134	10	5	1
Expresado en porcentajes	89.3%	6.6%%	3.3%	0.6%

Como se puede observar en estos resultados, ya empieza a haber mayores errores en el reconocimiento y en su gran mayoría se debe a una mala segmentación. La detección del inicio y fin de una palabra se comienza a complicar. Por ejemplo, en el caso de que el final de una palabra tenga un sonido sordo y el comienzo de la palabra que le sigue también contenga un sonido sordo, esto propiciara a que el inicio y final de las correspondientes palabras se traslapen y por consecuente sean mal identificadas. A partir de las cadenas de 3 dígitos los espacios de tiempo entre palabra y palabra se reducen.

Experimento para cadenas de 4 dígitos

Longitud de la cadena	0 errores	1 error	2 errores	3 errores	4 errores
4	115	9	9	12	5
Expresado en Porcentajes	76.6%	6%	6%	8%	3.3%

Ya en este experimento el porcentaje de reconocimiento de cero errores se alejó bastante de lo que sería el 100%. En casi todas las representaciones gráficas en el tiempo de las palabras conectadas se observa que cada parlante tiene la tendencia a decir las cadenas de palabras en grupos de dos, es decir en el caso de 4 dígitos, primero recita dos dígitos sin que haya casi nada de silencio entre ellas, después hay un silencio para continuar recitando los dos últimos dígitos restantes. El hecho de que no haya espacio de tiempo suficiente entre las palabras, hace que la segmentación corra el riesgo de hacerse mal y este error desencadena errores en las sucesivas palabras a ser reconocidas. A fin de cuentas el encontrar el punto correcto que indica el inicio de una palabra hará que el reconocimiento sea mucho más acertado.

Experimento para cadenas de 5 dígitos

Longitud de la cadena	0 errores	1 error	2 errores	3 errores	4 errores	5 errores
5	90	25	8	12	10	5
Expresado en porcentajes	60%	16.6%	5.3%	8%	6.6%	3.3%

Para este experimento ha sido demostrado que conforme se incremente el número de palabras que integran la cadena mayor será el error en el reconocimiento de la cadena. Los resultados muestran de manera drástica como este sistema no ha funcionado positivamente, para poder alcanzar un mejor reconocimiento en este sistema se debe de reconocer que el principal problema en estas cadenas fue que el espacio de tiempo que existe entre las palabras es mínimo y que la duración en tiempo de cada una de las palabras que conforman la cadena es muy pequeño en comparación con la duración que tienen los patrones de referencia. A pesar de todo, los resultados para de este experimento no han sido tan malos si consideramos que la base de referencia es independiente de la base de prueba; además de que la segmentación que se implemento demostró ser no muy buena para las cadenas más largas de 3 dígitos.

Segundo Experimento

Las siguientes tablas muestran los experimentos hechos para cada una de las cadenas de palabras. El único factor que se varió fue la técnica usada para la segmentación.

En la técnica usada en el primer experimento se puede deducir que el hecho de considerar la vecindad de coordenadas que están alrededor de un fin de palabra se utilicen como los posibles inicios de la siguiente palabra y que para cada una de esas coordenadas se haga un reconocimiento, se pierde en teoría mucho tiempo, es por eso que para este experimento no se siguió la misma técnica, lo que se empleó, fue considerar justamente a la coordenada final de la palabra reconocida como el único posible inicio de la siguiente palabra. Con esto se gana mucho tiempo en reconocimiento para el mejor de los casos, pero tiene la gran desventaja es el confundir el buen inicio de la palabra que sigue. Tanto puede comenzar mucho antes tanto puede comenzar mucho después el inicio de la palabra y obviamente esto orilla a que el reconocimiento sea erróneo. De igual forma, donde existan espacios de silencio, ellos se considerarán como parte de la palabra a reconocer y hasta aveces cuando el silencio es muy grande este será considerado como una palabra. Es por eso que los resultados de las siguientes tablas decaen mucho en comparación con los del primer experimento.

Por todo esto se puede concluir que el éxito de esta técnica queda sujeta a que se reconozcan cadenas de palabras que cumplan una característica que engloba una buena segmentación. Es el hecho de que hayan sido recitadas con un mínimo de silencio entre cada palabra, sólo un instante entre cada palabra bastará para que exista una segmentación correcta y tenga un mayor éxito el reconocimiento. Como ya se mencionó previamente, si se recitan en conjuntos de palabras todas las palabras que constituyen a la cadena habrá espacios de silencio que obliga al algoritmo a hacer un reconocimiento malo.

Experimento para cadenas de 2 dígitos

Longitud de la cadena	0 errores	1 error	2 errores
2	129	21	0
Expresado en porcentajes	86%	14%	0%

Lo que se observó en este experimento, las cadenas de 2 dígitos contienen un gran espacio de silencio entre cada palabra, este factor fue determinante para el mal reconocimiento como ya se ha mencionado. Es obvio que el porcentaje de reconocimiento es alto en comparación con los siguientes experimentos ya que sólo hay dos dígitos en la cadena pero cuando hay más como se ve más adelante el reconocimiento decae.

Experimento para cadenas de 3 dígitos

Longitud de la cadena	0 errores	1 error	2 errores	3 errores
3	123	8	13	3
Expresado en porcentajes	82%	5.3%	8.6%	2%

En este experimento se mantuvo casi con el mismo porcentaje de reconocimiento efectivo que el del experimento para dos dígitos, porque no existe la tendencia de recitar los dígitos en pequeño grupos, sino que en su mayoría son recitados en un solo conjunto. Y se puede observar de la tabla que no hay mucha diferencia entre los índices de error entre las columnas de 1 error y de 2 errores, estos errores son propios de tomar como definitivo el inicio de palabra la coordenada donde el algoritmo DTW haya determinado el final de la palabra que ha sido reconocida previamente.

Experimento para cadenas de 4 dígitos

Longitud de la cadena	0 errores	1 error	2 errores	3 errores	4 errores
4	93	10	24	21	5
Expresado en Porcentajes	62%	6.6%	16%	14%	3.3%

De igual forma, en este experimento el índice de mayor error ocurre en la columna de 2 errores porque las cadenas fueron recitadas en conjuntos de dos palabras existiendo un gran silencio entre cada conjunto.

Experimento para cadenas de 5 dígitos

Longitud de la cadena	0 errores	1 error	2 errores	3 errores	4 errores	5 errores
5	79	8	18	25	15	5
Expresado en porcentajes	52.6%	5.3%	12%	16.6%	10%	3.3%

De este experimento se observó que las cadenas de 5 dígitos fueron recitadas primero en un conjunto de 2 dígitos y después en un conjunto de 3 dígitos. Observando la tabla, en la columna de 3 errores destaca el mayor índice de errores, esto se debe a la manera en que fue recitada la cadena de dígitos, entre la tercera palabra y la segunda palabra existe en su mayoría de las cadenas un silencio.

5. CONCLUSIONES

El principal objetivo de este trabajo fue el desarrollar en particular un sistema de reconocimiento de palabras aisladas y otro para el reconocimiento de palabras conectadas. Principalmente este sistema está enfocado en aseverar experimentalmente la efectividad de la distancia de Itakura. Los dos sistemas emplean esta distancia por su rapidez en sus cálculos, claro esto no podría ser tan rápido sin la caracterización de señal en pequeños vectores de coeficientes LPC. Esta decisión demostró ser muy eficiente en el empleo de ambos sistemas.

El sistema propuesto en este trabajo para el reconocimiento de palabras aisladas ha sido el que Rabiner junto con Itakura desarrollaron, en principio se siguieron las mismas restricciones y metodología propuestas por ellos. Un punto primordial que debe destacarse en este experimento es que no se impusieron restricciones en las pendientes locales que hay dentro del paralelogramo del algoritmo DTW. Otro punto, determinante es el empleo del paralelogramo porque este impone una región de validez forzando a que siempre exista un mapeo del inicio de la palabra hasta el final de la palabra.

Haciendo una revisión de las ideas que se proponen en este sistema, veremos que hay un número de factores que deben ser identificados en la implementación de un sistema de reconocimiento, estos factores son:

Los locutores determinan las características con las que el sistema operara, es decir, es muy importante que el vocabulario de referencia y el vocabulario a ser probado sea realizado por el mismo conjunto de parlantes. Esto incrementara de manera determinante el porcentaje correcto de palabras reconocidas. De otra manera, si los parlantes que grabaron la base de referencia no son los mismos que los que grabaron la base que será probada, existirá una reducción en el porcentaje correcto de palabras aisladas.

La complejidad del vocabulario es un factor que afecta en la exactitud del reconocimiento. Es decir, si se tienen dentro de todas las palabras del vocabulario mismos sonidos en el interior de todas las palabras del vocabulario.

El sistema de senso juega otro factor importante, ya que el hablar muy cerca al micrófono producirá mucho ruido, para esto se necesita de algún tipo de cancelación de ruido. Estos factores serán muy importantes para la localización del principio y fin de la palabra. Prácticamente sin la localización correcta de estos puntos el reconocimiento será catastrófico.

La detección correcta del inicio y fin de las palabras fue llevada acabo en principio por el algoritmo propuesto por Rabiner- Los experimentos que se realizaron al principio indicaron que era necesario hacer una buena detección de estos puntos ya que si la palabra tenía mucho ruido antes de iniciar la palabra generaría malos reconocimientos, los puntos básicos que se debieron calibrar fueron los umbrales de energía y cruces por cero.

Por lo tanto, los resultados obtenidos demuestran que la confianza que puede ser depositada en este algoritmo es grande, ya que funciona acertadamente, rápido en sus cálculos y que no importa en que lenguajes se utilice este algoritmo siempre funcionará correctamente.

Para el experimento de palabras conectadas, se experimento con un algoritmo recursivo que funciona en una sola etapa, este algoritmo básicamente es el mismo que se empleó en los experimentos de palabras aisladas. Se utiliza el algoritmo DTW y la distancia de Itakura. La modificación que existe en este sistema, es que al momento de que se identifique la primera palabra dentro de la cadena, automáticamente se localiza su punto final y este punto final representa el inicio de la siguiente palabra, por lo tanto, esta coordenada vuelve a alimentar al algoritmo DTW como si fuera el inicio de una palabra que está recitada aisladamente y así se sigue sucesivamente hasta terminar de recorrer toda la cadena. De manera implícita al reconocer cada palabra estas se van concatenando con las palabras que ya hayan sido reconocidas, de esta forma es resuelto el problema de saber cual fue la secuencia con la que fueron recitadas las palabras.

La revisión de los principales factores que influyen en el buen funcionamiento del sistema son:

- Para las palabras conectadas que serán reconocidas, el algoritmo debe ser independiente del número de palabras que contenga la cadena de palabras, es decir, que no debe existir ningún máximo ni mínimo de palabras en la cadena y que el algoritmo sepa identificar cuantas palabras fueron recitadas.

- El algoritmo de ajuste dinámico del tiempo demostró se una herramienta valiosa para comparar de forma no lineal los conjuntos de vectores LPC que caracterizan a los patrones de referencia con los vectores LPC que caracterizan a las palabras de prueba, para poder después determinar así una medida de similitud entre dichas palabras.

- El algoritmo implementado para el reconocimiento de palabras conectadas requiere de generar excesivas distorsiones en el ajuste lineal. Para garantizar que haya una correcta alineación entre los patrones, el ajuste debe ser realizado libremente sin forzarlo a estar dentro de una región. Él porque de esta razón, es que no debe existir restricción alguna del tipo paralelogramo en el proceso del DTW, esto es porque no se puede garantizar que el área que delimita el paralelogramo sea válido para el reconocimiento. El paralelogramo tiene como restricciones que la señal de prueba no pueda ser mayor ni menor que dos veces el tamaño del patrón de referencia, es decir que sus pendientes deben de ser de $\frac{1}{2}$. Al momento de que se rompan estas restricciones, en el paralelogramo ocurrirá un colapso y por lo tanto no se podrá hacer un reconocimiento. Esto ocurre en este tipo de reconocimiento porque la longitud de una palabra que se encuentra dentro de la cadena tiene una duración en promedio de menos de la mitad de la que tienen las palabras de referencia. El dejar sin restricciones de frontera el algoritmo DTW, podría hacernos pensar que nunca encontraría el punto final de la palabra en prueba, ya que el paralelogramo obliga a que desde los puntos iniciales hasta los puntos finales sean evaluados y sin el paralelogramo esto no sucedería; además de que para poder emplear el paralelogramo primero se necesita conocer previamente los puntos finales de cada una de

las palabras que conforman la cadena y después evaluar si tienen la longitud mínima requerida para poderse comparar dentro de un paralelogramo con la correspondiente palabra de referencia. El empleo de esta técnica demostró que automáticamente en la mayoría de los casos el algoritmo por si sólo encuentra el final de una palabra, ya sea mal o bien reconocida, pero nos indica que ahí ha ocurrido un cambio importante. Se ha podido observar que existen diversos casos en que esta técnica tampoco es óptima, ésta presento errores cuando dos palabras o hasta cuatro fueron recitadas de manera muy rápida y que por ello no existen lapsos de tiempo entre ellas y el algoritmo se confundió tomando toda la cadena como si fuese una sola palabra.

- Sin duda otro factor que mejoraría el reconocimiento, sería que la base de referencia debería ser entrenada con la misma rapidez con la que usualmente se dictan los dígitos, ya que con esto, se podría guardar una mucho mejor una relación entre las longitudes de duración de las palabras de las respectivas bases. Al momento de que se tenga una palabra de corta duración dentro de la cadena, esta será favorecida a ser confundida con otra, porque al momento de tener menos duración implícitamente tendrá un pequeño conjunto de vectores LPC que la representen, y al momento de entrar al algoritmo DTW habrá menos evaluaciones entre este pequeño conjunto de vectores LPC y el conjunto más extenso de vectores LPC de la palabra de referencia. Entonces en los casos que sucede esto, la distancia mínima acumulada será pequeña porque el ajuste lineal en el tiempo fue bastante deficiente, casi una línea vertical.

- Las restricciones de las pendientes locales, también resultó mejor dejarlas a un lado. Los resultados fueron mucho más acertados sin estas restricciones; los tres posibles caminos de avanzar en el algoritmo DTW quedan sin favorecer a alguno en especial, simplemente se escoge la coordenada que posea la distancia mínima acumulada en ese momento.

- Quizás el factor más importante y determinante para la obtención de un sistema con un alto porcentaje de reconocimiento correcto, es la correcta segmentación de las palabras que componen la cadena. En otras palabras, él poder detectar con un cierto margen de tolerancia en donde comienza y donde termina cada una de las palabras, garantizará un más acertado índice de reconocimiento de palabras conectadas. El método empleado en este trabajo no funcionó del todo bien cuando las cadenas de palabras son mayores a 3 dígitos. La aseveración de este punto fue comprobada de forma experimental, de tal forma que la segmentación se realizó de manera manual. Se segmentaron muchas cadenas de palabras y en casi todos los casos el reconocimiento de toda la cadena resultó ser correcta. Es por eso, que si se tiene un buen programa que haga una correcta segmentación, el reconocimiento acertado de las palabras conectadas estaría por arriba del 90% sin importar el número de palabras que conformen las cadenas.

Como última observación. En todos los experimentos para palabras conectadas siempre existió dentro de la tabla de resultados un índice que muestra los casos cuando no se reconoció correctamente ninguna de los dígitos que componían a la cadena. El principal motivo de que sucedieran estos casos fue a una mala detección de inicio de cadena, es decir que la primera palabra de la cadena tenía demasiado silencio antes de empezar su recitación o el caso contrario, que se haya determinado su inicio mucho después de haber ocurrido su inicio correcto.

El reconocimiento de voz resulta ser uno de los últimos límites de nuestra habilidad humana para poder proporcionar un lenguaje natural hablado entre la máquina y el hombre. Yo creo que el perfeccionamiento de este tópico puede ser resuelto de forma viable en un futuro no muy distante. Hasta el momento, la valiosa meta de una comunicación sin restricciones parece estar fuera de nuestro alcance.

- ***BIBLIOGRAFÍA***

Libros:

1. Proakis, J. and Manolakis, D., Digital Signal Processing, Principles, Algorithms and Applications, Macmillan Publishing, 1992 USA

2. Rabiner, Lawrence R. and Schafer, R.W. Digital Processing of Speech Signals Prentice-Hall, 1978 USA

3. Wheddon, C. And Linggard, R., Speech and Language Processing Chapman and Hall, 1990 USA

4. Saito, Shuzo and Nakato, Kazuo, Fundamentals of Speech Signal Processing Academic Press Inc. USA

5. J.D. Markel and A.H. Gray, Jr. Linear Prediction of Speech Berlin Heidelberg New York 1976 USA

Artículos:

6. Herrera, V.R. Algazi and D. Irvine, An acoustic Approacch for Isolated Speech Recognition, Proceedings of the International Conferenca on Signal Processing Applications and Technology, ICSPAT 94, Vol. 2

7. Herrera, V.R. Algazi, V. Brown and D. Irvine, Subword Segmentation Alternatives for Isolated and Connected Words Recognition, Proceeding VII European Signal Proceesing Conference, EUPSICO, 94

8. L. Rabiner and S.E. Levinson, "Isolated and Connected Word Recognition-Theory and Selected Applications" IEEE 1981

9. H. Sakoe, "Two-Level DP-Matching-A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition," IEEE 1979

10. H. Ney, "The Use of a One-Satge Dynamic Programming Algorithm for Connected Word Recognition,"
IEEE 1984