

**SISTEMA DE MANDO INALÁMBRICO DE UN MOTOR DC MEDIANTE EL USO
DE LA VOZ.**

ANTHONY CONEO MERCADO

JAIME PERNETT CASTILLO

WILMER NOT SUAREZ

CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR

FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA

CARTAGENA DE INDIAS

2000

**SISTEMA DE MANDO INALÁMBRICO DE UN MOTOR DC MEDIANTE EL USO
DE LA VOZ.**

ANTHONY CONEO MERCADO

JAIME PERNETT CASTILLO

WILMER NOT SUAREZ

Tesis de grado presentado como requisito para optar al título de Ingeniero
Electrónico

Director :

EDUARDO GÓMEZ VASQUEZ

Ingeniero Electricista y Magister en Ciencias Computacionales

CORPORACIÓN UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERIA ELÉCTRICA Y ELECTRÓNICA
CARTAGENA DE INDIAS

2000

Cartagena de Indias D.T y C. Abril 10 del 2000

Señores:

Comité Evaluador

Facultad de Ingeniería Eléctrica y Electrónica

Apreciados señores

Por medio de la presente me permito informarles que el proyecto de grado titulado **“SISTEMA DE MANDO INALÁMBRICO DE UN MOTOR DC MEDIANTE EL USO DE LA VOZ”** ha sido desarrollado de acuerdo a los objetivos establecidos.

Como autores del proyecto consideramos que el trabajo es satisfactorio y amerita ser presentado para su evaluación.

Atentamente,

Jaime E. Pernet Castillo.

Anthony E. Coneo Mercado.

Wilmer J. Not Suarez.

Cartagena de Indias D.T y C. Abril 10 del 2000

Señores:

Comité Evaluador

Facultad de Ingeniería Eléctrica y Electrónica.

Apreciados señores

Por medio de la presente me permito informarles que el proyecto de grado titulado **“SISTEMA DE MANDO INALÁMBRICO DE UN MOTOR DC MEDIANTE EL USO DE LA VOZ”** ha sido desarrollado de acuerdo a los objetivos establecidos.

Como director del proyecto considero que el trabajo es satisfactorio y amerita ser presentado para su evaluación.

Atentamente,

Ing. Eduardo Gómez Vásquez.

Nota de aceptación .

Jurado .

Jurado .

Cartagena de Indias D.T y C. Abril 10 del 2000

AGRADECIMIENTOS

Los autores del proyecto expresamos nuestros más sentidos agradecimientos a todas las personas que de una u otra forma ayudaron a conseguir los objetivos y a llevarlo a feliz termino. En especial agradecemos a nuestro director ING. Eduardo Gómez Vásquez, pilar fundamental en el desarrollo de la tesis y gran colaborador.

DEDICATORIA

A mis queridos padres y hermanas por su amor.

A mis familiares por su gran apoyo.

A mis amigos por su fidelidad.

Jaime Pernet Castillo

DEDICATORIA

Doy gracias a mis padres y familiares por darme todo el apoyo en los momentos que más lo necesitaba.

Anthony Coneo Mercado

DEDICATORIA

Agradezco al principal eje motor de mi vida, *"MIS PADRES"*. Gracias por haberme soportado y apoyado en todos mis altibajos.

Gracias doy a la existencia de mi hijo Jefferson Daniel Not por ser la personita que da alegría a mi vida, la persona que me hace entender que hay que seguir adelante, que esto no acaba aquí, que este es solo un comienzo.

Gracias a mis hermanos, familiares y amistades por su apoyo.

Wilmer Not Suarez.

CONTENIDO

	Pág.
1. INTRODUCCIÓN	20
2. OBJETIVOS DE LA INVESTIGACIÓN.....	23
2.1 OBJETIVO GENERAL:	23
2.2 OBJETIVOS ESPECÍFICOS:.....	23
3. MARCO TEÓRICO.....	24
3.1 PROCESAMIENTO DE VOZ.....	24
3.1.1.Almacenamiento de la señal de voz.	26
3.1.2 Discriminación voz-silencio.	28
3.1.2.1 Discriminación Robusta.....	28
3.1.2.2. Discriminación Fina.....	29
3.1.3.Reacondicionamiento de la Señal.	30
3.1.3.1 Preénfasis.....	30
3.1.4. Extracción de características.....	31
3.4.1.1Cepstral.	31
3.4.1.2 Melcepstral	32
3.2 REDES NEURONALES	33
3.2.1 Definición	33
3.2.2 Neurofisiología	35
3.2.3 Neurona artificial.	37
3.2.4.Modelo de Neurona Artificial.....	44
3.2.4.1.Funciones de Transferencia de Neurona Artificial.....	45

3.2.5. Entrenamiento de la Red Neuronal.....	47
3.2.5.1.Backpropagation.....	48
3.2.6.Salidas de la Red.....	51
3.3.CONDICIONES ADVERSAS DEL RECONOCIMIENTO DEL HABLA	51
3.3.1.Ruido:	51
3.3.2.Distorsión:.....	52
3.3.3.Efectos de la articulación.....	53
3.4 MÁQUINAS DE CORRIENTE CONTINUA	54
3.3.1 Las máquinas de corriente continua:	54
3.3.2. Principios generales:.....	56
3.5. CIRCUITO DE INVERSIÓN DE GIRO	57
3.5.1 Scr – transistor.	59
3.5.2 Reles.	61
3.6.PUERTO PARALELO	64
3.6.1. Registros del puerto paralelo	65
3.7. TRANSMISIÓN ANALÓGICA EN FM	66
Transmisión analógica	67
Transmisión Analógica en F.M	68
Transmisores de FM directos:	69
4.EXPERIMENTACIÓN	71
4.1. IMPLEMENTACIÓN DEL TRANSMISOR.....	73
4.1.1. Receptor (Descripción General).	74
4.1.2. Especificaciones Técnicas Sistema General.....	75
4.2.DISEÑO DEL ALGORITMO DE PROCESAMIENTO Y RECONOCIMIENTO DE VOZ.....	82
4.2.1.Pre-procesamiento.	82

4.2.1.1 Almacenamiento	83
4.2.1.2. Discriminación Voz-Silencio	84
Señal muestreada a 16000 Hz.....	85
4.2.1.3. Reacondicionamiento de la Señal.....	86
4.2.2. Procesamiento	87
4.2.2.1. Extracción de Características.....	87
4.2.3. Post-Procesamiento.....	88
4.2.3.1. Reconocimiento y validación.	88
4.2.3.2. Entrenamiento y Simulación de la Red Neuronal.....	89
4.2.3.2. Salidas de la Red Neuronal.....	92
4.3. SELECCIÓN DEL MOTOR	93
4.3.1 Fundamentación Teórica.	93
4.3.1.1. Características en vacío	93
4.3.1.2. Especificaciones técnicas del motor de excitación independiente	96
4.3.1.3. Extracción de la curva de velocidad (voltaje-corriente) del motor de c.c.	96
4.4. DISEÑO DEL CIRCUITO DE INVERSIÓN DE GIRO	98
5. ACCESORIOS DEL PROYECTO.....	103
5.1. DISEÑO DE UNA GRABADORA DE VOZ (EN DELPHI5.0).....	103
5.2.DISEÑO DE LOS ALGORITMOS DE INTERFASE MATLAB–PUERTO PARALELO (EN DELPHI5.0)	104
5.3. MANUAL DEL USUARIO.	105
5.3.1. Entrenamiento de la red.	108
5.3.2. Prueba	113
5.4.MANUAL DE INSTALACIÓN.....	116
5.4.1. Estructura física.	116
5.4.1.1. Bloque del Tx / Rx.	117
5.4.1.2. Bloque del PC.....	118

5.4.1.3 Bloque del motor (caja negra)	119
5.4.2. Software	120
6. ASPECTOS DE INFRAESTRUCTURA.....	124
6.1.RECURSOS HUMANOS	124
6.2.DESCRIPCIÓN DE EQUIPOS Y HERRAMIENTAS REQUERIDAS; DISPONIBLES EN LA CUTB:.....	124
6.3.LABORATORIOS	124
7.OBSERVACIONES Y CONCLUSIONES	125
8.GLOSARIO	128
9.BIBLIOGRAFÍA	140
10.ANEXO.....	132

INDICE DE FIGURAS.

	Pág.
Figura 1. Producción del habla	25
Figura 2. Diagrama típico de red.	34
Figura 3. Célula nerviosa.	35
Figura 4. Elemento de procesamiento de la red	38
Figura 6. Partes de un motor de corriente continua.	56
Figura 7. Puente H.	58
Figura 8. Inversor de giro.	59
Figura 9. Polaridad positiva	60
Figura 10. Polaridad negativa	61
Figura 11. Puente H con reles.	62
Figura 12. Polaridad positiva	62
Figura 13. Polaridad negativa.	63
Figura 14. Registros del puerto paralelo.	65
Figura 15. Bits de los registros del puerto paralelo.	66

Figura 16. Transmisor FM directo.	70
Figura 17. Diagrama de bloques del proyecto	72
Figura 18. Fases del proyecto	73
Figura 19. Transreceptor	74
Figura 20. Panel frontal.	76
Figura 21. Panel posterior	76
Figura 22. Micrófono	77
Figura 23. Conectores del Tx/Rx.	78
Figura 24. Comparador.	79
Figura 25. Espectro de las palabras	80
Figura 26. Diagrama de la etapa de entrenamiento	82
Figura 28. Representación grafica de la ecuación.	95
Figura 29. E_o vs I_e	95
Figura 30. Montaje del circuito de prueba.	96
Figura 31. Inversor de giro	99
Figura 32. Lógica de control.	100
Figura 33. Etapa amplificadora	100
Figura 34. Sentido positivo.	101
Figura 35. Sentido negativo.	102
Figura 36. Presentación	106

Figura 37. Menú principal.	107
Figura 38. Menú de entrenamiento	108
Figura 39. Sub-menú de grabación de sonidos.	108
Figura 40. Grabación de sonidos.	109
Figura 41. Palabras a entrenar	109
Figura 42. Grabadora de voz.	110
Figura 43. Sub-menú de entrenamiento.	111
Figura 44. Fase de prueba	113
Figura 45. Motor encendido.	114
Figura 46. Giro hacia izquierda o derecha.	114
Figura 47. Apagado del motor	114
Figura 48. Esquema del receptor.	117
Figura 49. Caja base	118
Figura 50. Conexión del cable del puerto.	119
Figura 51. Conexiones del motor.	120
Figura 52. Path browser.	121
Figura 53. Selección de add path.	122
Figura 54. Búsqueda de archivos raíz.	122

INDICE DE TABLAS

	Pág.
Tabla 1. Tabla técnica	82
Tabla 2. Voltaje excitación vs corriente I_e .	87
Tabla 3. Combinaciones de bits	89

INDICE DE ANEXOS.

	Pág.
ANEXO A. Estructura del Reconocedor.	132
ANEXO B. Listados de los principales programas.	139
ANEXO C. Programa del Grabador de Voz.	159
ANEXO D. Programas de Salidas al Puerto Paralelo.	162

1. INTRODUCCIÓN

El análisis de la voz ha sido uno de las preocupaciones del hombre en la actualidad, debido a que por medio de ella se pueden alcanzar innumerables propósitos. Uno de los más comunes es la codificación del sonido que se emite por medio del habla, existen otros como el manejo de algún proceso, ya sea industrial, de seguridad, didáctico, entre otros.

Un problema que presenta la voz es el medio de propagación que hace que el sonido se distorsione y se atenúe ya sea por el ruido mecánico o eléctrico, esto hace que la voz no sea fácil de identificar, lo que nos lleva a analizar otros parámetros de la voz que permitan identificarla independientemente del medio.

En los inicios del procesamiento de la voz, se tejieron muchas hipótesis y se plantearon muchas soluciones, desconociendo la complejidad que verdaderamente causaba el reconocimiento de la voz, los investigadores de la época se convencieron de la necesidad de utilizar mecanismos avanzados con el fin de lograr los objetivos en el modelamiento de la voz humana.

Los medios actuales de identificación de voz son diversos; cada uno especializado en determinados parámetros y con métodos que son aleatorios, los cuales nos

dan la posibilidad de identificarla y procesarla para luego emitir una respuesta al entorno.

Las empresas actuales, por lo general manejan todos sus procesos a través de interfaces con equipos computacionales, que manipulados por operadores ejercen mando sobre cada uno de los procesos. El monitoreo de una máquina de uso industrial debe ser por lo general constante, para que no se produzcan alteraciones que puedan afectar en un determinado momento la producción de la empresa, esto hace que el supervisor tenga gran responsabilidad y que en algún momento recaiga en él la culpabilidad de pérdidas monetarias debido a fallas en la máquina.

Resulta más confiable para el supervisor de un proceso industrial tener un mando directo sobre la maquinaria en caso de que se presente una falla o que simplemente se quieran alterar algunas variables inherentes del proceso, tales como: encendido y velocidad de motores, niveles de temperatura, entre otras. Por lo general, estas variables se modifican en cuartos de control en donde los operarios esperan la voz de mando del supervisor.

El grupo propone un sistema que ponga en comunicación directa supervisor con el proceso que éste maneja, donde la voz del supervisor sea mecanismo de mando directo de la maquinaria. Este será el estudio y desarrollo del proyecto que se

tratará en las siguientes páginas donde se verán los procesos y etapas que se deben seguir para conseguir la interacción entre el hombre y la máquina.

2. OBJETIVOS DE LA INVESTIGACIÓN.

2.1 OBJETIVO GENERAL:

Diseñar y Construir un sistema de mando inalámbrico controlado por la voz.

2.2 OBJETIVOS ESPECÍFICOS:

1. Implementar un equipo de transmisión y recepción de audio en radiofrecuencia, para garantizar la comunicación entre el usuario y la interface de mando.
2. Elaborar un programa en Matlab basado en redes neuronales, para el procesamiento y reconocimiento de voz dependiente del hablante de las palabras aisladas (encender, apagar, izquierda, derecha), en un ambiente de trabajo de relación señal a ruido mayor a 30 dB.
3. Diseñar y construir una interfaz entre el computador y el motor que permita realizar la acción de mando sobre el motor.

3. MARCO TEÓRICO.

3.1 PROCESAMIENTO DE VOZ

El ser humano cuenta con varias formas de comunicación con los demás seres de su misma especie o de diferente, uno de los más significativos medios de comunicación es la voz, esta forma de transmitir mensajes y emitir sensaciones depende de varios aspectos, y es característica de cada individuo lo cual permite identificar al uno del otro.

El habla por sí sola es el resultado de una complicada relación de músculos controlada por el centro del lenguaje en la región frontal del cerebro, generalmente en el lado izquierdo.

El conjunto del sistema puede ser comparado con un instrumento de viento. Los pulmones aportan el aire; la tráquea permite a éste llegar a la laringe y en ella se genera sonido por medio de las cuerdas vocales. La boca y las cavidades nasales sirven de caja de resonancia. El tono del sonido varía de acuerdo con el grado de tensión de las cuerdas vocales, mientras el timbre viene determinado por dicha

caja de resonancia, el volumen, la forma del pecho, la boca y las cavidades nasales (ver figura 1).

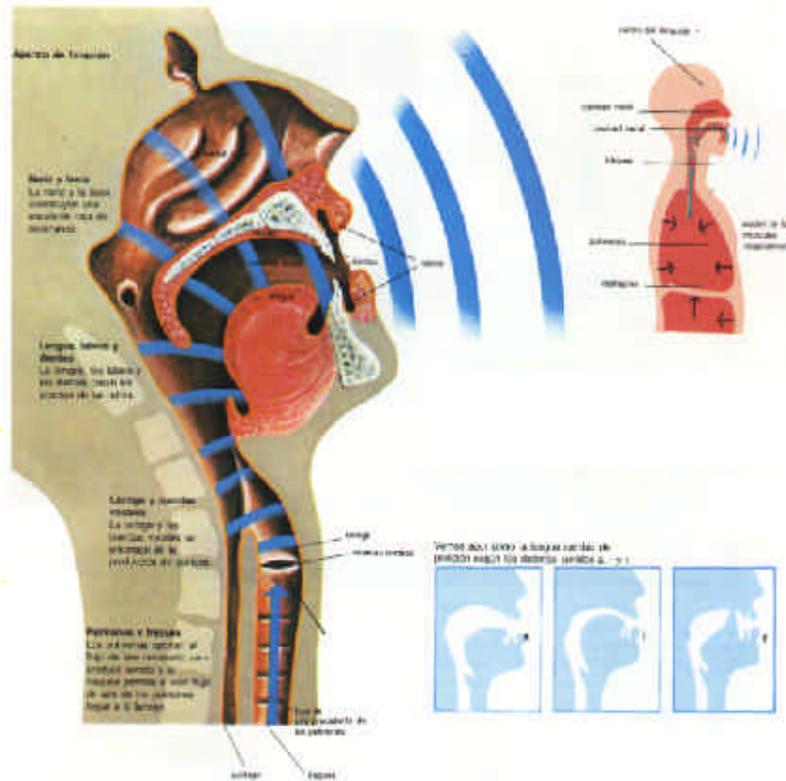


Figura 1. Producción del habla

La lengua, el paladar, los labios y los dientes son los responsables de la formación del sonido final.

La producción del habla puede ser resumida como una caja, en la cual sus elementos internos hacen que sea característica la voz para cada individuo, es decir, que podemos modelar como una función de transferencia dicha caja, la cual es propia del hablante y que sirve para poder diferenciar cada individuo.

La comunicación entre dos individuos que se encuentran distantes necesita más que la sola producción del habla, dependiendo de la distancia a que se encuentre, el individuo que envía el mensaje, así debe ser la cantidad de energía necesaria para que el receptor pueda recibir el mensaje, también dependiendo del medio donde se encuentre será bueno o malo para la comunicación. Para estos inconvenientes surgió la tecnología de comunicación a distancia, una de ellas es la radio-comunicación, la cual permite hacer más confiable la transmisión y recepción del mensaje.

Dentro del procesamiento de la voz podemos distinguir varios pasos que nos permiten analizar mejor la voz, ellos son los siguientes:

3.1.1. Almacenamiento de la señal de voz.

La señal de voz básicamente está constituida por ondas de presión producidas por el aparato fonador humano, una manera de capturar este tipo de señal es la realizada por un micrófono, el cual se encargará de convertir la onda de presión sonora en una señal eléctrica.

Los niveles de señal obtenidos por medio del micrófono no son los más adecuados para poder analizar y captar bien la voz, por lo que se hace necesario amplificar la señal como paso siguiente en el almacenamiento de voz. A partir de la señal analógica obtenida es necesario convertir la señal a formato digital para

poder procesarla en la computadora la cual realiza el proceso mediante los procesos: muestreo, retención y cuantificación. Estos procesos son conocidos como Modulación por Código de Pulsos (PCM). La señal vocal tiene componentes frecuenciales las cuales oscilan entre los 200 y 8 KHz, que pueden llegar a los 10 khz en algunos casos. Sin embargo la mayor parte de los sonidos vocales tienen energía espectral significativa hasta los 5khz. La frecuencia de muestreo dependerá del tipo de aplicación, para señales de voz se adopta un rango de 8 a 20 khz, la cual dependerá de la calidad del sonido que se desee hacer un muestreo. Otra consideración que se debe tener en cuenta es la cuantificación de la señal, la cual involucra la conversión de la amplitud de los valores muestreados en señales digitales, usando un número predeterminado de bits. El número de bits usados determinará la resolución que va a tener la muestra, y afectará la calidad de la voz muestreada, determinando así la cantidad de información a almacenar.

Para cada instante de muestreo, el convertidor analógico digital compara la señal muestreada con una serie de niveles de cuantificación predefinidos. El número de niveles N a usar determina la precisión del análisis y por tanto el número de bits necesarios. Cada bit adicional que se agrega contribuye en mejorar la relación señal a ruido en aproximadamente 6 dB. La señal de voz exhibe un rango dinámico de unos 50 a 60 dB. Por lo que resultaría suficiente una cuantificación de 8 o 9 bits para una buena calidad de voz.

3.1.2 Discriminación voz-silencio.

Una fase necesaria de dentro del procesamiento de la señal de voz es la detección de bordes, que no es más que aislar las palabras del silencio que las rodea, también conocida como detección de fronteras. Su importancia radica en la necesidad de predeterminar la palabra y poder aislarla de los ruidos introducidos por las perturbaciones.

La detección de fronteras suele abordarse en dos etapas diferenciales

- Discriminación Robusta.
- Discriminación fina.

3.1.2.1 Discriminación Robusta.

Un algoritmo simple y eficaz para la detección rasa de bordes, válido si el ruido no es excesivo, utiliza umbrales de tiempo y de energía (o amplitud) de la señal: La superación de estos umbrales es la muestra de la presencia de palabra. El umbral de tiempo resulta necesario debido a que no es suficiente afirmar que hay palabra cuando hay energía, es decir se pudiese tomar un ruido como palabra. Una palabra presenta siempre un mínimo de energía durante un mínimo de tiempo; y

viceversa, la ausencia de energía no implica fin de palabra, puesto que se terminaría al encontrar parte de un fonema explosivo. Una zona de silencio real tiene un mínimo de duración, por lo que es importante analizar los tiempos de muestreo de la señal. Por otra parte, es necesario, considerar como parte de la palabra el tramo de la señal inmediatamente anterior y posterior al momento del paso por el umbral de energía, pues, en caso contrario, podrían perderse los principios y/o finales de palabras constituidos por umbral de energía inferior al umbral. La detección rasa asegura que la palabra pronunciada quedará contenida en su totalidad en las fronteras detectadas, lo que resulta suficiente en muchos casos, conviene llevarla a cabo en combinación con el proceso de adquisición de la señal vocal.

3.1.2.2. Discriminación Fina

En un procedimiento más fino no solo tendríamos una mejora de los resultados de una eventual etapa subsiguiente de reconocimiento, sino a un indudable ahorro de memoria requerida para el almacenamiento de la señal y de tiempo a emplear en tratamientos posteriores. La detección fina se suele basar en los parámetros extraídos para los tratamientos posteriores (reconocimiento), aunque también se puede llevar a cabo mediante parámetros extraídos.

A partir de los resultados proporcionados por la detección rasa, el algoritmo de detección fina debe buscar de adelante hacia atrás, a partir del momento que se

cruzó el umbral de amplitud, el punto donde los parámetros extraídos indican la existencia de silencio. Para el fin de palabra, también de atrás hacia delante a partir del final raso, se procede de forma análoga hasta encontrar el momento en que deja de haber silencio. La existencia o no de silencio se determina a partir de los parámetros utilizados, y la manera de decidir dicha existencia depende obviamente de la naturaleza de éstos.

3.1.3.Reacondicionamiento de la Señal.

Las técnicas para reacondicionar señales permiten extraer la información acústica directamente a partir de la vocal emitida. Esto podríamos realizarlo por medio del preénfasis y la aplicación de la ventana de Hamming. Estos dos métodos nos permiten realizar un mejor modelaje de la señal y permitir la adecuación de la señal a fin de suavizarla asperezas que pueda tener la muestra obtenida.

3.1.3.1 Preénfasis.

Es utilizado con el propósito de suavizar el espectro y reducir las inestabilidades de cálculo asociadas con las operaciones aritméticas de precisión finita. Además

$$H(Z)=1 - a z^{-1}$$

se usa para compensar la caída de -6 dB. que experimenta la señal al pasar a través del tracto vocal. Se usa un filtro digital de primer orden cuya función de transferencia es:

3.1.4. Extracción de características.

3.4.1.1 Cepstral.

Los coeficientes cepstral se pueden definir como la transformada inversa del módulo del espectro de la señal de voz en escala logarítmica. También se puede definir como la transformada inversa del logaritmo del módulo de la transformada de la señal.

La técnica cepstral presenta las siguientes características:

- Las altas componentes cepstrales corresponden con variaciones rápidas del espectro y por lo tanto corresponden al rizado del espectro, el cual se relaciona estrechamente con la frecuencia fundamental y el carácter periódico de la excitación aplicado al tracto vocal.

- Las bajas componentes cepstrales corresponden con variaciones lentas de las componentes espectrales y por tanto contienen información de la envolvente del espectro, la cual se relaciona con la respuesta en frecuencia del filtro que modela el tracto vocal.

Lo que nos interesa generalmente son las características se utilizan las bajas componentes cepstrales para reconocer voz, de ahí que se usan los coeficientes cepstral para separar la información correspondiente a la excitación y la que corresponde al filtro resonador.

Los coeficientes cepstral ofrecen finalmente lo siguiente:

- a) Se usan como entrada a la red neuronal los primeros diez coeficientes.
- b) Tienen relación con la envolvente y con el tracto vocal. Por tanto puede ser bueno para distinguir entre vocales o distinguir vocales de algunas consonantes. Sin embargo, esta condición puede mejorar usando unos parámetros similares que son los del *Melcepstral*.

3.4.1.2 *Melcepstral*

Son unas componentes cepstral que se obtienen teniendo en cuenta características audio perceptuales. Pueden obtenerse como los cepstral de la

señal de voz filtrada. Este filtrado pretende simular las características de filtro que tiene el oído, buscando que el reconocimiento sea lo mas parecido al reconocimiento que realiza una persona.

Para reconocimiento de voces suele utilizarse los primeros diez o doce componentes cepstral de la señal de voz filtrada.

La ventaja que ofrece el Melcepstral con respecto al cepstral es que es más específico para las características de la voz, por la capacidad que tiene de modelar los coeficientes que operan a bajas frecuencia.

3.2 REDES NEURONALES

3.2.1 Definición

Comenzaremos definiendo las estructuras de redes neuronales como colecciones de procesadores paralelos conectados entre sí en la forma de grafo dirigido, organizado de tal modo que la estructura de la red sea la adecuada para el problema que se esté considerando. Tomando la Figura 2 como referencia de un diagrama típico de red, se puede representar esquemáticamente cada elemento de procesamiento (o unidad) de la red como un nodo, indicando las conexiones

entre nodos mediante arcos. Se indicará la dirección del flujo de información dentro de la red mediante el uso de puntas de flecha en las conexiones.

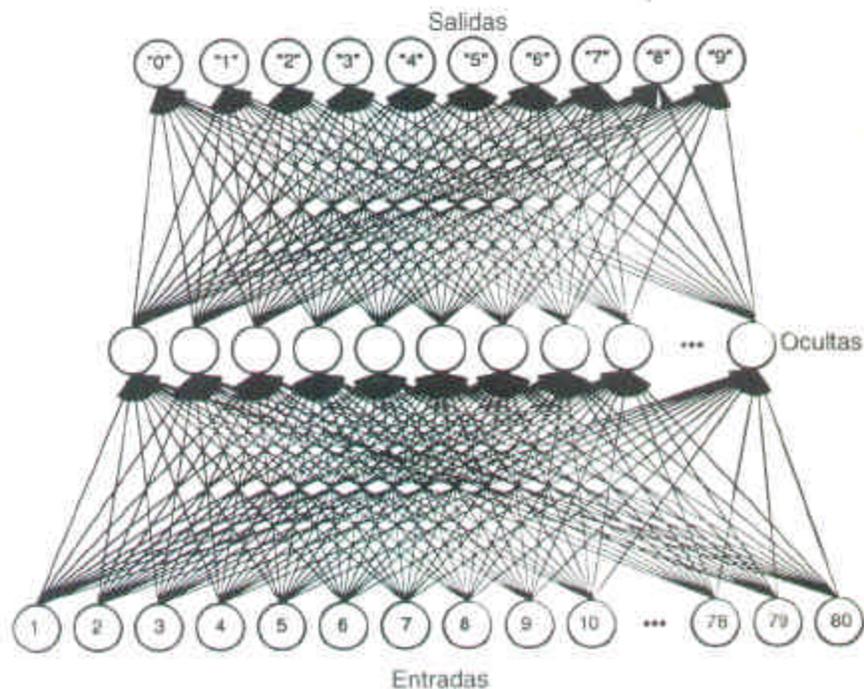


Figura 2. Diagrama típico de red.

Este esquema representa el problema del reconocimiento de cosas. La aplicación de una trama de entrada en la capa inferior de procesadores puede dar lugar a que se activen muchas de las unidades de la segunda capa (o capa oculta). La actividad de la capa oculta debería dar lugar entonces a que se activase una y sólo una de las unidades de salida: la que esté asociada a la trama que se esté identificando. También debería tenerse en cuenta el elevado número de conexiones que se necesitan para esta red; que es relativamente pequeña. Para poder entender mejor entraremos en el mundo de las neuronas.

3.2.2 Neurofisiología

Entre las estructuras fundamentales de una célula nerviosa típica se cuentan las dendritas, el cuerpo de la célula y un único axón. El axón de muchas neuronas está rodeado por una membrana que se denomina vaina de mielina. Los nodos de Ranvier interrumpen periódicamente la vaina de mielina a lo largo del axón. Las sinapsis conectan el axón de la neurona con distintas partes de otras neuronas.

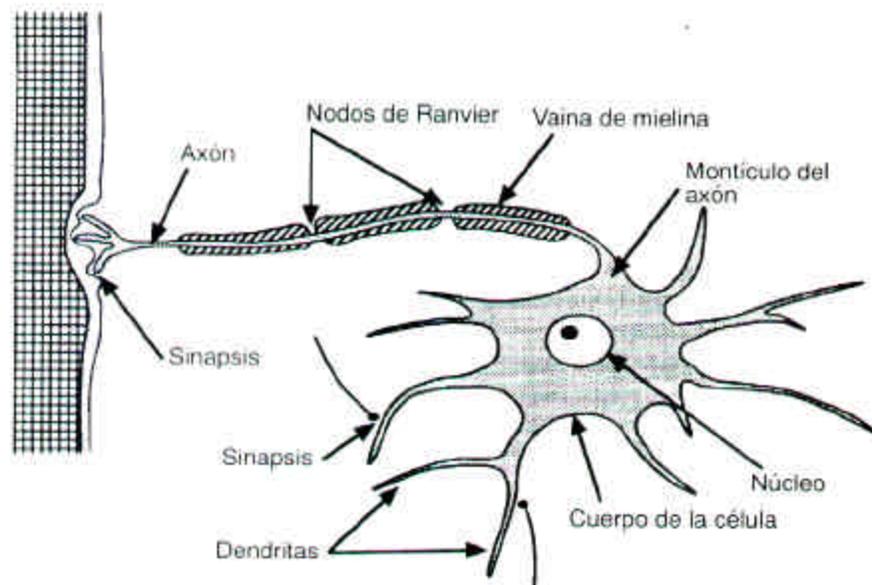


Figura 3. Célula nerviosa.

La Figura 3 representa los componentes principales de una célula nerviosa típica perteneciente al sistema nervioso central. La membrana de la neurona separa el plasma intracelular del fluido intersticial que se encuentra fuera de la célula. La membrana es permeable para ciertas especies iónicas, y actúa de tal forma que se mantenga una diferencia de potencial entre el fluido intracelular y el fluido

extracelular. Este efecto se consigue primordialmente mediante la acción de una bomba de sodio-potasio. También están presentes otras especies iónicas como son los iones cloruro e iones orgánicos negativos.

Todas las especies iónicas se pueden difundir a través de la membrana, con la excepción de los iones orgánicos, que son demasiado grandes. Dado que los iones orgánicos no pueden salir de la célula por difusión, su carga negativa neta dificulta la entrada en la célula de iones cloro por difusión; por tanto, habrá una concentración más alta de iones cloro fuera de la célula. La bomba de sodio-potasio determina una concentración más alta de potasio dentro de la célula y una concentración más alta de sodio fuera de ella.

La membrana celular es selectivamente más permeable para los iones de potasio que para los iones de sodio. El gradiente químico del potasio tiende a hacer que los iones de potasio salgan de la célula por difusión, pero la fuerte atracción de los iones orgánicos negativos tiende a mantener dentro el potasio. El resultado de estas fuerzas opuestas es que se alcanza un equilibrio en el cual hay más iones de sodio y cloro fuera de la célula, y más iones orgánicos y de potasio dentro de ella. Además, el equilibrio resultante produce una diferencia de potencial a través de la membrana de la célula de unos 70 a 100 milivoltios (mV), siendo el más negativo el fluido intracelular. Este potencial, que se denomina **potencial de reposo de la célula**.

3.2.3 Neurona artificial.

En esta sección se hace una transición de algunas ideas tomadas de la neurobiología a las estructuras idealizadas que constituyen la base de la mayoría de los modelos ANS. Primero describimos una neurona general artificial que posee la mayoría de las características que necesitaremos para futuras descripciones de modelos específicos. Más adelante dentro de esta sección examinaremos brevemente un ejemplo concreto de ANS denominado perceptrón. El perceptrón fue el resultado de un primer intento de simular la computación neuronal para llevar a cabo tareas complejas. Examinaremos concretamente algunas de las limitaciones de esta aproximación, y veremos la forma en que podrán superarse.

Los elementos individuales de cálculo que forman la mayoría de los modelos de sistemas neuronales artificiales no suelen llamarse *neuronas artificiales*; lo más frecuente es darles el nombre de nodos, unidades o elementos de procesamiento(PE). Todos estos términos se emplean de manera intercambiable.

Otro detalle que hay que tener en cuenta es que no siempre es correcto pensar que los elementos de procesamiento poseen una relación biunívoca con neuronas biológicas reales. A veces es mejor tomar los elementos de procesamiento

individuales como representantes de la actividad colectiva de un grupo de neuronas. Esta interpretación no sólo nos ayudará a evitar el error consistente en hablar como si nuestros sistemas fueran modelos reales del cerebro, sino que además hará más tratable el problema cuando realmente intentemos modelar la conducta de alguna estructura biológica.

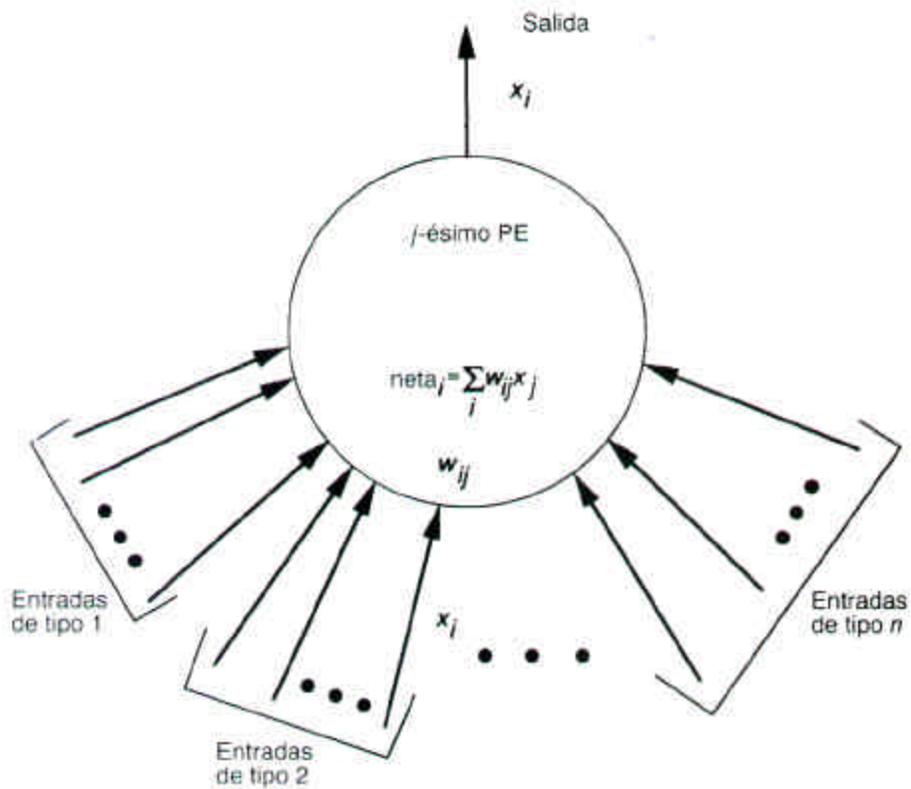


Figura 4. Elemento de procesamiento de la red

La figura 4 representa un único PE de una red. Las conexiones de entrada se representan en forma de flechas procedentes de otros elementos de procesamiento. Cada conexión de entrada tiene asociada una cantidad w_{ij} que se

denomina peso. Hay un único valor de salida, que se puede aplicar a otras unidades.

Este es nuestro modelo general de PE. Cada PE está numerado, siendo el i -ésimo el que aparece en la figura. Una vez advertido el lector acerca de la necesidad de no hacer demasiadas analogías biológicas, vamos a ignorar nuestra propia recomendación y haremos unas cuantas analogías. Por ejemplo, al igual que una neurona verdadera, el PE tiene muchas entradas pero tiene una sola salida, que se puede aplicar a muchos otros PEs de la red. La entrada que recibe el i -ésimo PE procedente del j -ésimo PE se indica en la forma X_j (obsérvese que este valor es también la salida del j -ésimo nodo, del mismo modo que la salida generada por el i -ésimo nodo se denota X_i). Cada conexión con el i -ésimo PE tiene asociada a él una magnitud llamada peso o intensidad de conexión. El peso de la conexión procedente del j -ésimo nodo y que llega al i -ésimo nodo se denota mediante w_{ij} . Todas estas cantidades tienen sus análogos en el modelo de una neurona estándar: la salida del PE se corresponde con la frecuencia de disparo de la neurona, y los pesos corresponden a la intensidad de las conexiones sinápticas entre neuronas. En nuestros modelos estas cantidades se van a representar mediante números reales.

Obsérvese que las entradas que llegan a un PE están desglosadas en varios tipos. Este desglose refleja el hecho consistente en que cada conexión de entrada puede tener uno de entre varios efectos. Una conexión de entrada puede ser

excitatoria o inhibitoria, por ejemplo. En nuestros modelos, las conexiones excitatorias tienen pesos positivos, y las conexiones inhibitorias tienen pesos negativos. También son posibles otros tipos. Los términos ganancia, amortiguamiento y disparo fortuito aluden a otras conexiones de propósitos especiales, que se describirán más adelante en el libro. Las conexiones excitatorias e inhibitorias suelen considerarse conjuntamente, y son las formas más comunes de entrada de los PE.

Cada PE determina un valor de entrada neto basándose en todas las conexiones de entrada. En ausencia de conexiones especiales, lo típico es calcular el valor de entrada neto sumando los valores de entrada, ponderados (multiplicados) mediante sus pesos correspondientes. En otras palabras, la entrada neta de la i -ésima unidad se puede escribir en la forma

$$neta_i = \sum_j X_j W_{ij} \quad (1.1)$$

En donde el índice j recorre todas las conexiones que posea el PE. Obsérvese que la excitación y la inhibición se tienen en cuenta automáticamente mediante el signo de sus pesos. Este cálculo de sumas de productos tiene un papel importante en las simulaciones de redes que se describirán más adelante. Dado que es frecuente que haya un número de interconexiones muy elevado en las redes, la velocidad con la que se puede llevar a cabo este cálculo suele ser determinante para el rendimiento de la simulación de cualquier red dada.

Una vez que la entrada neta ha sido calculada, se transforma en el valor de activación, o activación simplemente, para ese PE. Se puede escribir ese valor de activación en la forma

$$a_i(t) = F_i(a_i(t - a), neta_i(t)) \quad (1.2)$$

Para denotar que la activación es una función explícita de la entrada neta. Obsérvese que la activación actual puede depender del valor anterior de la activación, $a(t - 1)^3$. Incluimos esta dependencia en la definición por generalidad. En la mayoría de los casos, la activación y la entrada neta son idénticas, y los términos suelen emplearse de manera intercambiable. En algunas ocasiones, la activación y la entrada neta no son iguales, y es preciso prestar atención a la diferencia. En general, sin embargo, siempre podremos utilizar la activación para denotar la entrada neta, y viceversa.

Una vez que se ha calculado la activación del PE, se puede determinar el valor de salida aplicando la **función de salida**:

$$X_i = f_i(a_i) \quad (1.3)$$

Dado que normalmente es $a_i = neta_i$, esta función suele escribirse en la forma

$$X_i = f_i(neta_i) \quad (1.4)$$

Una de las razones por las cuales se estudia cuidadosamente el tema de la activación frente a la entrada neta es que el término función de activación se utiliza en algunas ocasiones para aludir a la función f , que transforma el valor de la entrada neta, $neta_i$, en el valor de salida del nodo, x_i . En este texto se empleará siempre el término función de salida para aludir a la $f_i()$ de las Ecuaciones (1.3) y (1.4). Téngase en cuenta, sin embargo, que la literatura no siempre sigue a rajatabla esta convención.

Cuando se están describiendo las bases matemáticas de los modelos de redes, suele resultar útil pensar que la red es un sistema dinámico, esto es, que es un sistema que evoluciona a lo largo del tiempo. Para describir una red de estas características, se escribirán ecuaciones diferenciales que describan el ritmo de cambio temporal de las salidas de los distintos PEs. Por ejemplo, $\dot{x}_i = g_i(x_i, neta_i)$ representa una ecuación diferencial general para la salida del i -ésimo PE, en donde el punto que hay encima de la x denota la derivación con respecto al tiempo. Dado que $neta_i$ depende de las salidas de muchas otras unidades, lo que se tiene realmente es un sistema de ecuaciones diferenciales acopladas.

Como ejemplo, examinemos la ecuación siguiente:

$$\dot{x}_i = -x_i + f_i(neta_i)$$

Que corresponde a la salida del i -ésimo elemento de procesamiento. Se aplican unos cuantos valores de entrada al PE para que $\text{net}_i > 0$. Si las entradas quedan estables durante un período de tiempo suficiente, el valor de la salida alcanzará un valor de equilibrio, cuando sea $\dot{x}_i = 0$, dado por

$$X_i = f_i(\text{net}_i)$$

que es idéntico a la ecuación (1.4). Frecuentemente, se puede suponer que los valores de entrada permanecen constantes hasta que se ha logrado el equilibrio. Una vez más la unidad tiene un valor de salida no nulo, la eliminación de las entradas dará lugar a que la salida se vuelva cero. Si $\text{net}_i=0$, entonces

$$\dot{x}_i = -x$$

lo cual significa que $\dot{x} \rightarrow 0$

También resulta útil ver la colección de valores de los pesos como un sistema dinámico. Recuérdese la descripción de la sección anterior, en la que se afirmaba que el aprendizaje es un resultado de la modificación de la fuerza de las uniones sinápticas entre neuronas. Se puede escribir un sistema de ecuaciones diferenciales para los pesos, $W_{ij} = G_i(w_{ij}, x_i, x_j, \dots)$ en donde G_i representa la ley de aprendizaje. El proceso de aprendizaje consiste en hallar los pesos que codifican ese conocimiento que deseamos que aprenda el sistema. Para la mayor parte de los sistemas reales, no es fácil determinar una solución en forma cerrada para

este sistema de ecuaciones. Existen técnicas, sin embargo, que dan lugar a una aproximación razonable de la solución. Probar la existencia de soluciones estables para estos tipos de sistemas es objeto de intensas investigaciones en la actualidad, y es probable que siga siéndolo durante algún tiempo.

3.2.4. Modelo de Neurona Artificial

Una neurona simple sin bias (b) es mostrada en la figura (5) de la izquierda.

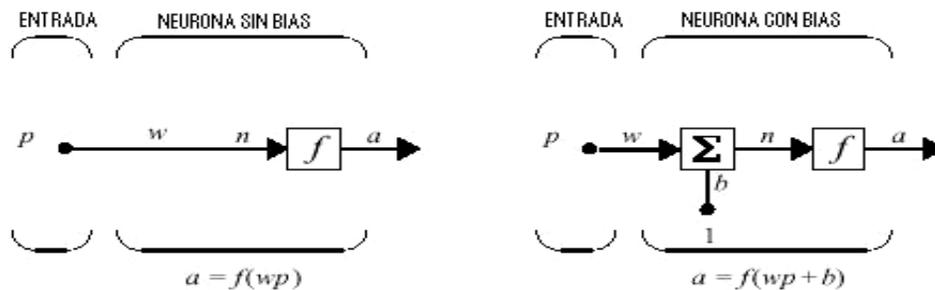


Figura 5. Modelo neurona artificial

La entrada del escalar p se transmite a través de una conexión que multiplica su fuerza por el peso del escalar w , formando el producto wp . Aquí wp es la entrada de la función de transferencia f que produce como respuesta el escalar a (los tipos de funciones de transferencia serán explicados en la siguiente sección). La neurona en la parte derecha de la figura anterior se encuentra alterada por el escalar, b . Se puede ver que b se agrega al producto wp , como muestra el sumando. El bias es similar a un peso, sólo que tiene una entrada constante ante cualquier entrada; que para este caso es de 1. La función de transferencia toma el

término w_p+b y genera el resultado de la neurona artificial.

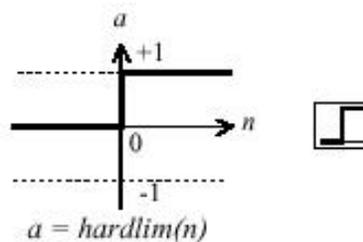
Note que w y b son ambos parámetros escalar ajustables de la neurona. La idea central de las redes neuronales es que tales parámetros pueden ser ajustables para que la red exhiba resultados deseados o una conducta interesante.

Así, nosotros podemos entrenar la red para hacer un trabajo particular ajustando el peso o el bias, o quizás la propia red ajustará estos parámetros (aprendizaje).

3.2.4.1. Funciones de Transferencia de Neurona Artificial

En la sección se estudiarán las funciones de transferencia que se mencionaron en el modelo de Neurona Artificial. Existe una gran variedad de funciones de Transferencia que se pueden asociar a la Neurona Artificial, las más usadas son: Hardlim, Purelin y Logsig.

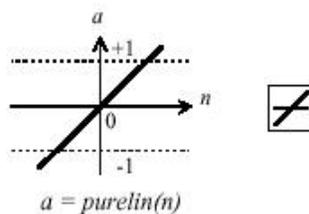
- Hardlim:



La Función de Transferencia Hardlim toma el dato de entrada y arroja un 1 si el

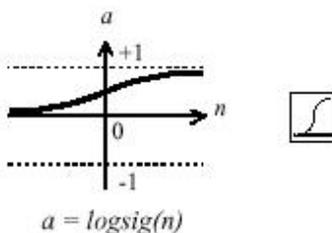
dato es positivo y cero cuando la entrada es negativa. La característica principal de este método es que solo existen dos posibles salidas, por tanto su uso está muy extendido cuando se necesita tomar decisiones y no se requiera de respuestas negativas.

- Purelin:



La Función de Transferencia Purelin genera respuestas que van desde -1 a $+1$, donde la variación se produce en forma lineal. Las neuronas con este tipo de función son muy usadas en los filtros lineales.

- Logsig:



La Función de Transferencia sigmoide toma la entrada que puede tener cualquier valor entre infinito y menos infinito, arroja una respuesta que varía de 0 a 1 .

Esta Función del normalmente se usa en redes de backpropagation, en parte porque es diferenciable.

El símbolo en el cuadrado al derecho de cada gráfico de Función de Transferencia mostrado anteriormente representa la función del traslado asociada.

3.2.5. Entrenamiento de la Red Neuronal

Como se dijo en el Modelo de un Neurona, cada neurona consta de una serie pesos (w_i) y bias (b_i) cuyos valores afectan la respuesta que nos proporcione la Red Neuronal ante una determinada entrada. El proceso de darle a la Red Neuronal los pesos y los bias adecuados para obtener las respuestas deseadas es lo que se conoce como entrenamiento de la Red Neuronal.

Se tienen pares de vectores de entrada y vectores de salida y se quiere calcular unos pesos (no necesariamente únicos) que produzcan unos vectores de salida lo más parecidos a los deseados para cada vector de entrada.

Todo entrenamiento parte de unos pesos iniciales (normalmente aleatorios y pequeños -respecto a los valores de las entradas-) y se van haciendo modificaciones (normalmente bastante pequeñas de forma que se asemeje a un movimiento bastante continuo y no a saltos) hasta alcanzar los pesos que se

buscan.

Dado que una Red Neuronal consta de una extensa cantidad de neuronas, resulta imposible ajustar todos los pesos y bias en forma manual. Para resolver este problema se han implementado una gran variedad de modelos matemáticos que buscan entrenar la Red Neuronal de tal manera que se produzca el menor error posible, estos modelos son: La Regla Perceptrón, El Entrenamiento Adaptativo, Backpropagation (Propagación hacia Atrás), Cuantificación del Vector de Aprendizaje, entre otros.

Según James A. Freeman en su libro sobre Redes Neuronales afirma que el modelo Backpropagation es el más indicado cuando se desea realizar un Reconocimiento de tramas, y dado que la voz después de procesada se convierte en tramas, esta afirmación nos inclina para el modelo (Backpropagation) será el que se aplique para el diseño de la Red Neuronal.

3.2.5.1. Backpropagation

El modelo de Backpropagation funciona de la siguiente forma:

La red aprende un conjunto predefinido de pares de entradas y salidas dados como ejemplo, empleando un ciclo propagación-adaptación de dos fases. Una vez que se ha aplicado una trama de entrada como estímulo para la primera capa de

unidades de la red, ésta se va propagando a través de todas las capas superiores hasta generar una salida. La señal de salida se compara entonces con la salida deseada, y se calcula una señal de error para cada unidad de salida. Las señales de error se transmiten entonces hacia atrás, partiendo de la capa de salida, hacia todos los nodos de la capa intermedia que contribuyan directamente a la salida. Sin embargo, las unidades de la capa intermedia sólo reciben una fracción de la señal total de error, basándose aproximadamente en la contribución relativa que haya aportado la unidad a la salida original. Este proceso se repite, capa por capa, hasta que todos los nodos de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, se actualizan los pesos de conexión de cada unidad, para hacer que la red converja hacia un estado que permita codificar todas las tramas de entrenamiento. La importancia de este proceso consiste en que, a medida que se entrena la red, los nodos de las capas intermedias se organizan a sí mismos de tal modo que los distintos nodos aprenden a reconocer distintas características del espacio total de entradas. Después del entrenamiento, cuando se les presente una trama arbitraria de entrada que contenga ruido o que esté incompleta, las unidades de las capas ocultas de la red responderán con una salida activa si la nueva entrada contiene una trama que se asemeje a aquella característica que las unidades individuales hayan aprendido a reconocer durante su entrenamiento. Y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir sus salidas si la trama de entrada no contiene la característica para reconocer la cual han sido entrenadas.

A medida que las señales se propagan a través de las diferentes capas de la red, la trama de actividad que está presente en todas las capas superiores se puede ver como una trama con características que son reconocidas por las unidades de la capa subsiguiente. La trama de salida que se genera se puede ver como un mapa de características que ofrece una indicación de la presencia o ausencia de muchas combinaciones distintas de características dentro de la entrada. El resultado global de este comportamiento es que la Red Neuronal constituye un medio eficiente para permitir a un sistema de computadores examinar tramas de datos que pueden ser ruidosas, o estar incompletas, y reconocer tramas sutiles a partir de entradas parciales.

Hay varios investigadores que han demostrado que, durante el entrenamiento, El Backpropagation tiende a desarrollar relaciones internas entre nodos con el fin de organizar los datos de entrenamiento en clases de tramas[3]. Esta tendencia se puede extrapolar, para llegar a la hipótesis consistente en que todas las unidades de capas ocultas de La Red Neuronal por Backpropagation son asociadas de alguna manera a características específicas de la trama de entrada como consecuencia del entrenamiento. Lo que sea o no sea exactamente la asociación puede no resultar evidente para el observador humano. Lo importante es que la red ha encontrado una representación interna que le permite generar las salidas deseadas cuando se le dan las entradas de entrenamiento. Esta misma representación interna se puede aplicar a entradas que no fueran utilizadas

durante el entrenamiento. La Red Neuronal por Backpropagation clasificará estas entradas que no había visto hasta el momento según las características que compartan con los ejemplos de entrenamiento.

3.2.6.Salidas de la Red

Las salidas de la red están condicionadas por la cantidad de neuronas que tenga la capa de salida, estas neuronas tendrán diferentes valores los cuales les dan la importancia o peso que tiene en la respuesta, en la etapa de salida de la red se pueden utilizar algoritmos externos que permitan predeterminar a que es igual la salida o lo que es lo mismo detectar la palabra que fue reconocida después del procesamiento y postprocesamiento de la red.

3.3.CONDICIONES ADVERSAS DEL RECONOCIMIENTO DEL HABLA

Un reconocedor del habla encuentra a menudo tres condiciones adversas, que son: Ruido, Distorsión y los Efectos de la Articulación.

3.3.1.Ruido:

El ruido del ambiente acústico normalmente es considerado aditivo, significando que el signo grabado es una suma de la señal del habla y el ruido del ambiente.

Los niveles altos de ruido del ambiente son una de las preocupaciones primarias para un reconocedor de habla. Las fuentes de ruido del ambiente acústico son abundantes. Por ejemplo, en la industria, las fuentes de ruido incluyen la maquinaria, los discos duros de computadoras personales (PC), teléfono que este cerca y la conversación de fondo de otras personas. Éstas son las fuentes del ruido proporcionan a una degradación de la respuesta de un reconocedor del discurso. El nivel de presión legítimo (SPL) en una oficina personal normal está alrededor de 45 a 50 dB (criterio del ruido NC 40 ~ 45). Dentro de un fábrica el nivel del ruido acústico considerablemente más alto.

Otros tipos de ruido como el ruido eléctrico y ruido del quantization que por supuesto están presentes en cualquier sistema de reconocimiento de habla electrónico, están generalmente a un nivel debajo del umbral de preocupación. No obstante, ruido debido a la transmisión de potencia 60Hz que a veces pueden ser un factor que afecta al reconocedor.

3.3.2. Distorsión:

Aparte de la contaminación aditivo debido al ruido, el señal de habla también sufre una serie de distorsiones espectrales inevitables antes de almacenarse y procesarse para iniciar el reconocimiento del habla. Hay dos tipos de distorsión: lineal y no lineal. Aunque la mayoría de nuestra discusión se concentrará en distorsiones lineales.

El transductor del micrófono, dependiendo de su tipo y posición, puede deformar significativamente el espectro del habla. Cuando la configuración del transductor uso de prueba es diferente al usado durante entrenamiento de los modelos de la referencia, la desigualdad en distorsión espectral se vuelve uno de los mayores problemas. Por ejemplo, un sistema de reconocimiento del habla de vocabulario grande con una exactitud de 85% en una condición del transductor emparejada (micrófono Sennheiser HMD224 para entrenamiento) sólo podría lograr menos de 19% la palabra exactitud cuando un micrófono diferente (PZM&fs) durante la prueba de las palabras.

Cuando en un reconocedor automático del habla se utiliza un micrófono de ancho de banda muy restringido (menor de 8Khz), se pueden causar distorsión extensa del espectro a analizar, dado que hay un cierto rango de frecuencias que fueron atenuadas, es decir, se pierde parte de la información.

3.3.3.Efectos de la articulación

Muchos factores afectan la manera de hablar de cada hablador individual. Incluso el acto psicológico de comunicarse con una máquina de reconocimiento de habla podría hacer al hablador producir una diferencia notable en su acento legítimo y modelo rítmico. Cambios característicos en articulación debido a la influencia del medio ambiente, conocido como el efecto Lombardo, también puede ser dramático. Cuando un hablador habla en un ambiente con un ruido enmascarando

de 90 dB esta demostrado es que el primer formante de una vocal aumenta a menudo mientras el segundo formante disminuye, produciendo un cambio potencial en el espacio vocal [8]. Estos cambios característicos afectan dramáticamente la actuación de un reconocedor de habla automático.

3.4 MÁQUINAS DE CORRIENTE CONTINUA

3.3.1 Las máquinas de corriente continua:

Las máquinas de corriente continua se clasifican en generadores y motores. La estructura física de la máquina consta de dos partes el estator o parte estacionaria y el rotor o parte giratoria. La parte estacionaria de la máquina consta de *carcaza*, que provee el soporte físico y las *piezas polares* que se proyectan hacia adentro y suministran un trayecto para el flujo magnético de la máquina. Los extremos de las piezas polares que están cerca del rotor se extienden sobre la superficie de éste para distribuir igualmente el flujo sobre la superficie. Estos extremos se denominan *zapatitas del polo*. La superficie expuesta de la zapata del polo se llama *cara del polo* y la distancia entre la cara del polo y el rotor se llama intervalo de aire o *entrehierro*.

Hay dos embobinados principales en una máquina de c.c. los embobinados del inducido y los de los inductores. *Los embobinados del inducido* se definen como aquellos en los que se induce un voltaje y *los embobinados inductores* se definen como los que producen el flujo magnético principal en la máquina. En una máquina de c.c. normal. Los embobinados del inducido se localizan en el rotor y los embobinados inductores se localizan en el estator. Por la circunstancia de que los embobinados del inducido están localizados en el rotor. Un rotor de máquina de c.c. de por sí, algunas veces es llamado armadura.

En el rotor de una máquina de c.c. se distingue el *colector*, el cual se caracteriza porque se construye de láminas de cobre aisladas por un tipo de material similar a la mica. La función del colector es la de recoger la energía de las bobinas del inducido, o recibir la energía externa suministrada a los mismos. Sobre el colector se colocan las escobillas, las cuales se hacen de carbón, grafito, grafito metálico o una mezcla de carbón y grafito. Tienen una alta conductividad para reducir las

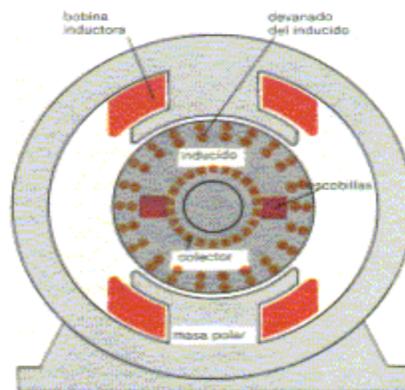


Figura 6. Partes de un motor de corriente continua

pérdidas eléctricas y un bajo coeficiente de fricción para reducir el desgaste excesivo. En la figura 6 se puede apreciar una máquina de c.c. con algunas de las partes anteriormente señaladas.

3.3.2. Principios generales:

Inducción electromagnética. Un campo magnético está representado por líneas de flujo continuas que se consideran saliendo de un polo norte y entrando a un polo sur de un imán. Cuando cambia el número de esas líneas eslabonadas por una bobina, “como en la figura”, se induce un voltaje en la bobina igual a 1 V por un cambio de 10^8 eslabonamientos (Mxls) por cada vuelta de la bobina, es decir,

$$E = (DfT \cdot 10^{-8}) / t \text{ V.}$$

Si las líneas de flujo se deforman por el movimiento del conductor de la bobina antes de romperse, la dirección del voltaje del inducido se considera hacia dentro del conductor si se muestra que las flechas por el flujo distorsionado, apuntan en el sentido del giro de las manecillas del reloj,

Si un conductor lleva una corriente alrededor de él se forman espiras de flujo. La dirección del flujo es en el sentido de las manecillas del reloj si la corriente es tal que se aleja del observador y hacia el conductor y es en sentido contrario al giro de las manecillas del reloj si la corriente del conductor sale del papel y se dirige al observador.

Si este conductor está en presencia de un campo magnético la combinación del flujo del campo y el flujo que genera el conductor puede considerarse que produce una concentración de flujo en el lado del conductor en donde los flujos son aditivos, y una disminución en el lado donde se oponen. El resultado es una fuerza sobre el conductor, que tiende a moverlo hacia el lado que tiene flujo reducido. Esta es la acción del motor.

3.5. CIRCUITO DE INVERSIÓN DE GIRO

Los circuitos inversores de giro son generalmente circuitos asistidos, por elementos que se conjugan en diferentes progresiones con en el fin de trabajar para cada uno de los ciclos.

Una de las configuraciones más comunes y la cual mencionaremos en este parte del proyecto es la configuración tipo H la cual es llamada así por la forma en que se encuentran dispuestos sus elementos, Ver figura (7), esta configuración es la que más común en el diseño de inversores de giro y tiene varias opciones como veremos a continuación:

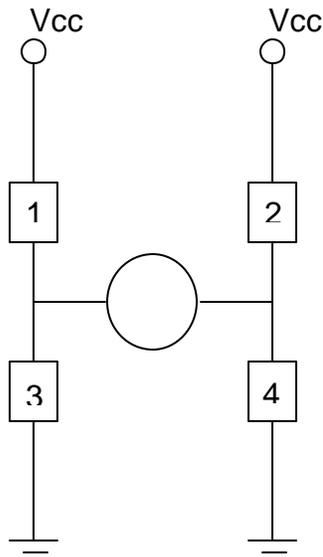


Figura 7. Puente H.

En este tipo de configuración las pareja 1 – 3 se combina para producir la polaridad en un sentido, mientras que la pareja 2 y 4 se conjugan para producirlas en el otro sentido , teniendo como referencia los elementos de conjugación, podemos utilizar varios elementos de la electronica de potencia para suministrar la corriente necesaria en la circulacion del dispositivo. Dentro de esto elementos estan los SCR, transistores de potencia, FET de potencia y los reles, estos ultimos que son los elementos que en ultima instancia evaluaremos para el diseño de la etapa de potencia del circuito, asi que de manera superficial mencionaremos las características generales de las mas comunes

3.5.1 Scr – transistor.

En la figura 8 se puede observar claramente la combinación que existe entre dos elementos de potencia para producir la inversión de giro en este caso las parejas son SCR Y Transistor.

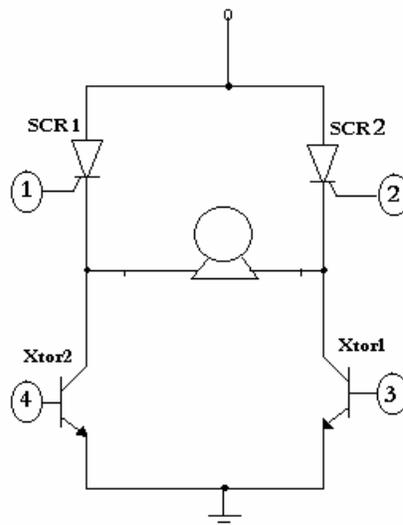


Figura 8. Inversor de giro.

El circuito trabaja con el encendido y apagado de los tiristores, este proceso se hace por pareja las cuales están reseñadas en el dibujo (SCR1, Xtor1), el tiristor 1 se enciende con una pulso de activación en el Gate, mientras que la base del transistor 1 al recibir igualmente una corriente entra en conducción en el proceso de encendido o giro a la derecha, el cual le imprime una polaridad positiva al motor figura 9, el apagado de la secuencia se produce cuando se deja de alimentar la base del transistor, esto produce que el transistor no conduzca y por lo tanto corta

la circulación de corriente en el sentido positivo, el SCR queda si corriente, al mismo tiempo que su gate deja de recibir pulso.

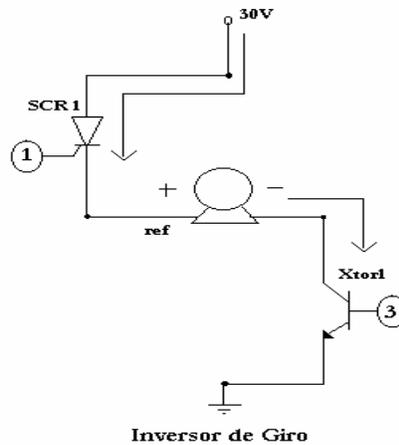


Figura 9. Polaridad positiva

En un proceso similar se produce la secuencia negativa en donde la otra parte del sistema, o sea la segunda pareja opera conjuntamente cambiando la polaridad del motor que pasa de tener en la referencia un voltaje positivo a tener uno negativo ver figura 10, y de igual manera se realiza el proceso de apagado.

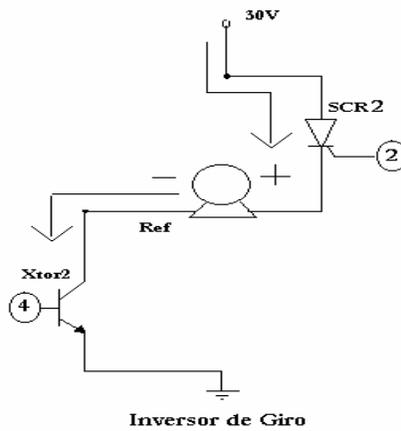


Figura 10. Polaridad negativa

3.5.2 Relés.

Es la configuración más sencilla y por lo tanto la que usaremos en el diseño las combinaciones de este dispositivo son gemelas porque utilizan 4 relés agrupados en parejas el circuito trabaja igual que el anterior para la polaridad positiva las parejas de relés se encuentran sus contactos normalmente abierto como se puede ver en la figura 11.

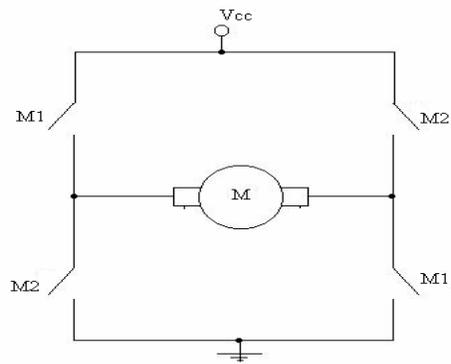


Figura 11. Puente H con reles.

Al activar los reles M1 que corresponden a la pareja 1 – 3, cierran sus contactos y el motor queda con la referencia positiva como se ve en la figura (12).

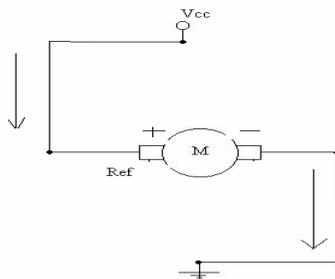


Figura 12. Polaridad positiva

Para el caso de la inversión de polaridad o de giro se utiliza la combinación 2 – 4 cerrando sus contactos y quedando la referencia del motor negativa como indica la figura 13.

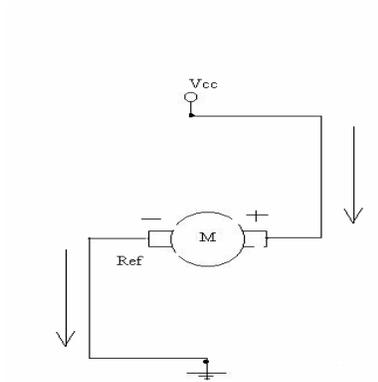


Figura 13. Polaridad negativa.

Cada configuración presenta su cierto grado de dificultad, en el caso de la ultima configuración, en el tiempo en que bs relés conmutan de un estado al otro, este tiempo de retardo que puedan tener puede causar un corto circuito con la fuente de manera que el motor sufra daños, para ello hay que regular los tiempos de conmutación de manera adecuada, los otros tipos de configuraciones son de mayor dificultad y obedecen a situaciones de polarización y despolarización en el caso de transistores, apagado y encendido en los SCR, y los FET que son dispositivos de manejo de campo donde la polarización es muy importante y segura, presentan dificultad con su manipulación debido al problema de campo externo que puede afectar el rendimiento del dispositivo.

3.6.PUERTO PARALELO

El puerto paralelo se utiliza generalmente en el PC para conectar la impresora; no obstante, esta interfaz es muy flexible y puede ser utilizada para una gran variedad de aplicaciones distintas, entre las que se encuentran la conexión de un circuito de Interfaz de potencia.

Como sabe, la particularidad del puerto paralelo es que esta pensado para transmitir y recibir los datos de 8 en 8 bits (de byte en byte), lo que se traduce fundamentalmente en una mayor velocidad de transferencia de información. Una de las diferencias del puerto paralelo respecto al puerto serie es que las señales eléctricas utilizadas para representar los estados lógicos 0 y 1 son de 0 y +5 voltios, respectivamente, en vez de las tensiones negativas (entre -3 y -15 voltios) y positivas (entre +3 y +15 voltios) utilizadas por el puerto serie. Una diferencia menor la constituye el hecho de que en el lado del PC el puerto paralelo dispone de un conector hembra, mientras que el puerto serie dispone de un conector macho.

CONECTOR DEL PUERTO PARALELO				
CONECTOR PC	SEÑAL	TIPO SEÑAL	CONECTOR IMPRESORA	DESCRIPCIÓN
1	/STR	Control	1	Activación de transferencia de datos
2	D0	Dato	2	Bit de dato 0
3	D1	Dato	3	Bit de dato 1
4	D2	Dato	4	Bit de dato 2
5	D3	Dato	5	Bit de dato 3
6	D4	Dato	6	Bit de dato 4
7	D5	Dato	7	Bit de dato 5
8	D6	Dato	8	Bit de dato 6
9	D7	Dato	9	Bit de dato 7
10	/ACK	Estado	10	Transferencia de datos correcta
11	BSY	Estado	11	Impresora ocupada
12	PAP	Estado	12	Sin papel
13	OFON	Estado	13	Indicador de impresora en línea
14	/ALF	Control	14	Alimentación de línea automática
15	/FEH	Estado	32	Error
16	/INI	Control	31	Inicialización de impresora
17	/DSL	Control	36	Selección de impresora
18-25	Tierra		19-30,33	Tierra 0 V.
-	0V		16	-
-	Chasis		17	Tierra protegida
-	+5V		18	
-	-		34,35	Sin usar

Figura 14. Registros del puerto paralelo.

3.6.1. Registros del puerto paralelo

El puerto paralelo dispone de tres registros a través de los cuales se intercambian tanto los datos de información como las señales de control. Estos registros son el registro de datos, registro de estado y registro de control. El **registro de datos** contiene la información que va a ser transferida a la interfaz o que ha sido recibida de la misma. El **registro de estado** nos ofrece información del estado del dispositivo conectado al puerto paralelo (si está conectado o no, si recibe los datos correctamente, etc.). Por último, el **registro de control** controla el comportamiento del dispositivo así como la generación de interrupciones hardware. El registro de

datos y el registro de control son registros bidireccionales, mientras que el registro de estado es un registro de sólo lectura.

REGISTRO	OFFSET		7	6	5	4	3	2	1	0
Registro de datos	00h	Señal	D7	D6	D5	D4	D3	D2	D1	D0
		Patilla	9	8	7	6	5	4	3	2
Registro de estado	01h	Señal	/BSY	/ACK	PAP	OFON	/FEH	-	-	-
		Patilla	11	10	12	13	15	-	-	-
Registro de control	02h	Señal	-	-	-	IRQ	DSL	/INI	ALF	STR
		Patilla	-	-	-	-	17	16	14	1

Figura 15. Bits de los registros del puerto paralelo.

3.7. TRANSMISIÓN ANALÓGICA EN FM

La radio comunicación es un sistema que utiliza como soporte la información transmitida mediante ondas hertzianas de frecuencias comprendidas entre unos 10 khz y algunos centenares de GHz. La información puede consistir en una señal acústica, a cuyas características de amplitud y frecuencia va ligado algún parámetro de la onda llamada portadora, en cuyo caso se llama comunicación radiofónica.

De acuerdo con los tipos de señales que se existen podemos transmitir de dos formas ya sea analógicamente o digitalmente, cabe mencionar que los fenómenos

naturales (dentro de ellos el habla), esta representados señales analógicas, que son continuas en el tiempo y a las cuales se les debe dar un tratamiento como tal.

Transmisión analógica

La transmisión analógica basa su importancia en el tratamiento de la señal, en su forma propia, o sea la analógica, la información debe ser transportada entre un transmisor y un receptor. Dependiendo de la relación entre el transmisor y el receptor, podemos tener tres modos de transmisión, estos son: SIMPLEX, HALF-DUPLEX Y FULL DUPLEX.

-Simplex (SX). : la transmisión solo va en un dirección, son llamados también de un solo sentido, solo para recibir o solo para transmitir. Una ubicación puede ser un transmisor o un receptor, pero no ambos. Un ejemplo de la transmisión simplex es la radiodifusión de la radio comercial o de la televisión.

-Half-duplex (HDX).: Con una operación half-duplex la transmisión se puede realizar en ambos sentidos pero no al mismo tiempo. También se les conoce como sistemas con alternativa de dos sentidos, cualquier sentido o cambio y fuera. Un ejemplo de half duplex son los radios de *botón para hablar* como los policíacos.

-Full-duplex (FDX). : La operación Full-duplex permite transmitir y recibir al mismo tiempo es la mas completa de la transmisiones, se le llama líneas simultaneas de

doble sentido, duplex o de ambos sentidos. Una ubicación puede transmitir y recibir simultáneamente. Un sistema telefónico estándar es el ejemplo claro de la comunicación Full-duplex.

Los mensajes para ser enviados deben ser modulados, de forma tal que la Información sea recibida lo mejor posible, y que sea descifrada por el receptor. Para ello reconocemos tres formas de modulación, las cuales van de acuerdo a tres parámetros fundamentales de la señal, como son la amplitud, frecuencia o fase, característicos de cada señal.

Transmisión Analógica en F.M

La modulación en frecuencia a diferencia de la de amplitud, trabaja una de las características más importante de la señal como lo es la frecuencia, se realiza variando la frecuencia de la portadora de amplitud constante directamente proporcional, a la amplitud de la señal modulante, con una relación igual a la frecuencia de la señal modulante, es decir a la variación de la señal modulante se le asignara variaciones de frecuencia de acuerdo a una referencia o portadora.

Dentro de las ventajas de la transmisión en frecuencia con respecto a la de amplitud es que la potencia necesaria para la transmisión va a ser menor debido a que ahora no hay que amplificar la señal, solo se necesita una repetidora la cual proporcione la estabilidad y mantenimiento de las frecuencias.

La modulación en frecuencia se puede hacer de modo indirecto o de modo directo. La *FM directa* es la modulación angular en el cual la frecuencia de la portadora varía (desviada), directamente por la señal modulante. Con la FM directa, la variación de frecuencia instantánea es directamente proporcional a la amplitud de la señal modulante. Podemos encontrar varios tipos de moduladores directos entre ellos los simples, donde la señal depende de un capacitor de un micrófono el cual actúa como transductor de energía acústica a mecánica, el modulador de diodo varactor, el modulador de reactancia de FM y moduladores de FM directos de circuito integrado lineal entre otros.

La *FM indirecta* es una modulación angular en la cual la frecuencia de la portadora se desvía indirectamente por la señal modulante. La FM indirecta se logra cambiando directamente la fase de la portadora y es, por lo tanto, una forma de modulación en fase directa. La fase instantánea de la portadora es directamente proporcional a la señal modulante. Una de las ventajas del FM indirecto es que usa un oscilador de cristal con búfer para la fuente de la señal portadora, por lo que la señal es más estable en frecuencia, a diferencia de los directos. La desventaja que presenta es que las características de capacitancia contra voltaje no son propiamente lineales para su configuración.

Transmisores de FM directos: Los transmisores de FM directos producen una forma de onda de salida, en la cual la desviación de frecuencia es directamente proporcional a la señal modulante. Consecuentemente, el oscilador de la

portadora debe desviarse directamente. El transmisor consta de los bloques mostrados en la figura 16. El primer bloque es un modulador de frecuencia y oscilador de muestreo el cual proporciona una frecuencia la cual va a servir de referencia luego vemos los multiplicadores de frecuencia dependiendo de la portadora piloto por la cual se va a transmitir en la antena. y un circuito de realimentación conocido como control de frecuencia automática (AFC), el cual compara la frecuencia de la portadora del oscilador sin cristal con un oscilador de cristal de referencia (cabe mencionar la estabilidad que proporciona el oscilador de cristal), y entonces, produce un voltaje de corrección proporcional a la diferencia entre las dos frecuencias. El voltaje de corrección se regresa al oscilador de la portadora para comparar automáticamente cualquier movimiento que pueda haber ocurrido.

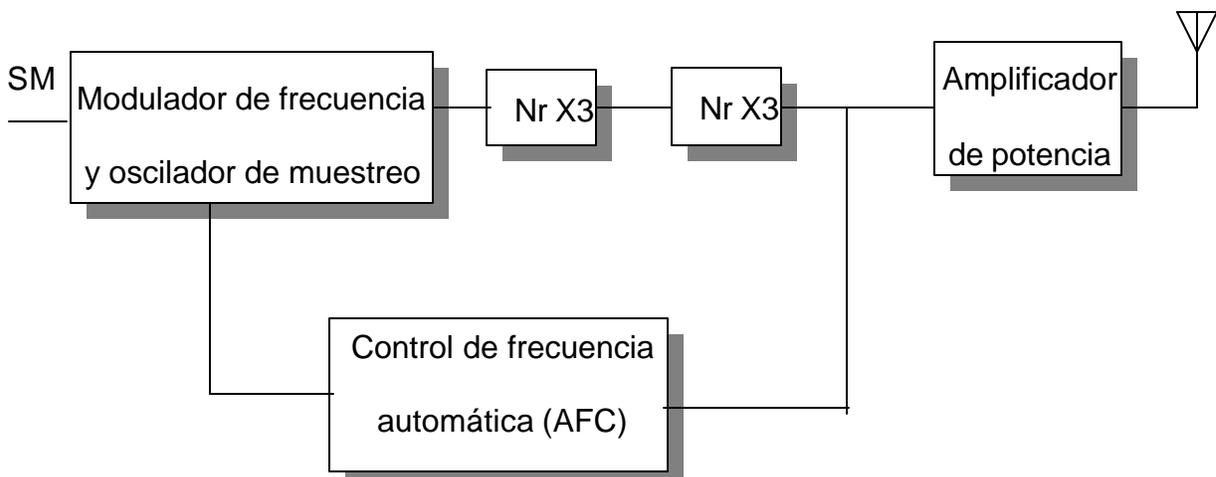


Figura 16. Transmisor FM directo.

4.EXPERIMENTACIÓN

En esta sección se explicará detalladamente cada uno de los pasos a seguir para el diseño y montaje del Sistema de Mando Inalámbrico de un Motor DC Mediante el Uso de la Voz.

La primera parte es definir cuál es el elemento sobre el cual se va a ejercer mando, y después se diseña el mando de dicho elemento.

Dado que el objetivo del proyecto es demostrar la potencialidad de las redes neuronales, el elemento sobre el cual se va ejercer mando queda a criterio del experimentador. Nosotros para realizar nuestras pruebas decidimos seleccionar un motor DC, ya que se amolda a las palabras del reconocedor (encender, apagar, izquierda, derecha). En primera instancia se seleccionó un motor DC de 30 V, el cual era de tipo didáctico y que por tener baja potencia era de fácil manejo, a parte de eso el prototipo inicial era un solo encendido y apagado del motor, con lo que cumplíamos con las especificaciones de mando del motor. Pero por sugerencias cambiamos el motor y utilizamos dos ordenes nuevas como son el giro a la izquierda y giro a la derecha.

Trabajando con los recursos de la universidad localizamos un motor de mayor potencia y al cual se le podía hacer inversión de giro, este motor será motivo de análisis mas adelante.

Ahora definimos un esquema general de lo que desarrollaremos:

1. Tenemos que implementar una interfaz inalámbrica que permita a usuario desplazarse en un cierto entorno, que haga más eficiente el uso del sistema de mando por voz.
2. Diseñar una interfaz entre la salida del computador y el motor que permita el control o sea encendido, apagado e inversión de giro, y manejo de potencia del motor en el proceso de arranque y trabajo.
3. Definir la forma de cómo se va a interpretar la información del usuario (voz), que para nuestro caso es el sistema de Reconocimiento de Voz.

En un diagrama de bloque podemos definirlo así:

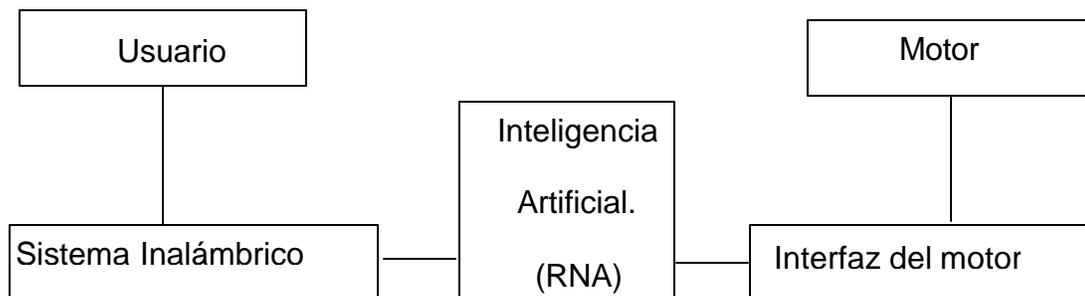


Figura 17. Diagrama de bloques del proyecto

La representación gráfica sería siguiente:

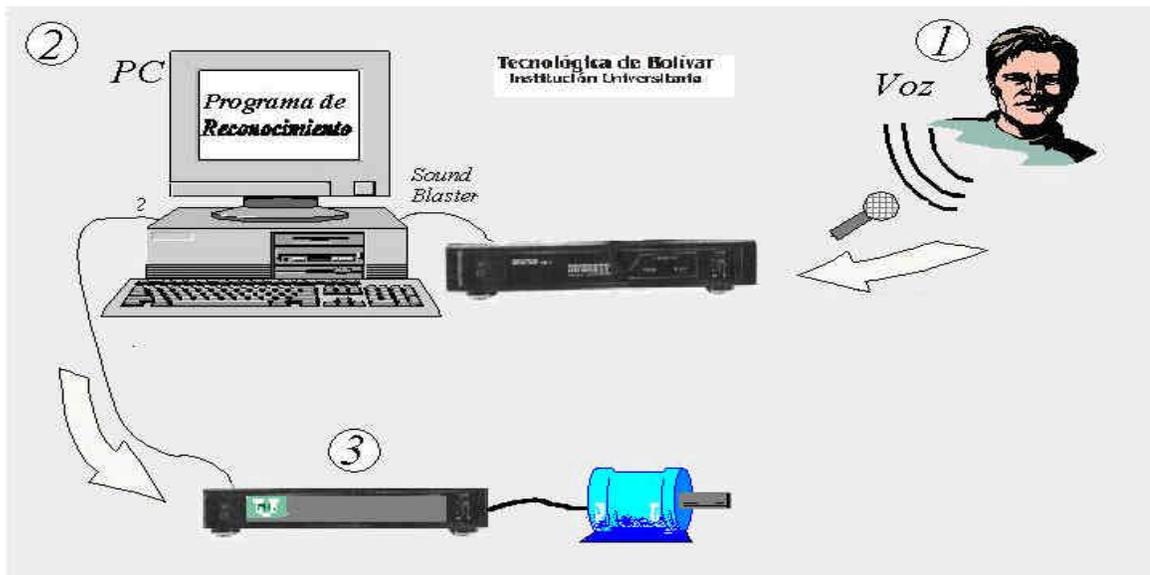


Figura18. Fases del proyecto

Ya establecidas las tres fases del proyecto podemos explicar en detalle cada una de éstas.

4.1. IMPLEMENTACIÓN DEL TRANSMISOR

Para la correcta selección del transmisor se partió de los siguientes parámetros:

- a. Distancia entre el transmisor y receptor debe ser de 50mts(sin obstáculos).
- b. La relación señal a ruido debe ser de 30dB.
- c. Ancho de Banda de 8Khz.

Entendiendo los parámetros y las facilidades que debería tener el transmisor se procedió a la búsqueda del mismo.

El sistema de transmisión aparte de servir para transmitir, era necesario tener una señal de alarma que sirviese como punto de información para automatizar el acceso al proceso de captación de la voz, fase primaria en el proceso de reconocimiento de voz.

El dispositivo seleccionado fue el WMS / 10 H (ver figura 19)



Figura 19. Transreceptor

4.1.1. Receptor (Descripción General).

Este micrófono trabaja en la banda alta de VHF. Tanto el micrófono inalámbrico como el receptor, son elementos de alta fidelidad. El sistema inalámbrico incluye una unidad receptora y un micrófono inalámbrico de mano. El sistema opera en la banda alta de VHF, dentro de las frecuencias de 160 a 250 MHz, controladas por cuarzo.

4.1.2. Especificaciones Técnicas Sistema General

Frecuencia portadora	160 a 250 MHz.
Estabilidad de frecuencia	± 0.005 % controlado por cristal de cuarzo.
Sistema de modulación	FM.
Rango máximo de desviación	± 15 KHz, con compresor limitador.
Respuesta de frecuencia	20Hz a 15KHz ± 2 dB.
Relación señal / ruido	> 100 dB
Distorsión (T.H.D)	$< 0.5\%$
Rango dinámico de audio	> 100 dB y > 118 dB con limitador.
Distancia efectiva entre mic. y receptor	50 m.
Rango de temperatura de operación	-10°C a 55°C

ESPECIFICACIONES TÉCNICAS DEL RECEPTOR

Sistema de recepción	Un canal con frecuencia fija controlada por cuarzo.
Sensibilidad de recepción	25 dB/uV, para una relación señal / ruido de 60 dB.
Relación señal / ruido	Mayor a 108 dB, a desviación de 15KHz y entrada de antena de 60 dB/uV
De-énfasis	50 μS .
Rechazo a la señal de imagen	Mínimo 60 dB.

Nivel de salida de audio	No balanceada: 0-0.5 V, a 15 KHz de desviación y carga de 600Ω.
Conector de salida	No balanceado: jack tipo phono de 6.3 mm de diámetro.
Tensión de alimentación	115 V o 230 V AC
Dimensiones (an x al x pr)	430x44x200 mm.
Peso	1.8 Kg.

PANEL FRONTAL

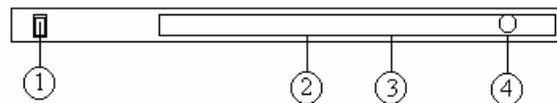


Figura 20. Panel frontal.

1. Conmutador de encendido / apagado.
2. Señal de radiofrecuencia. Indica de receptor de la señal del micrófono.
3. Señal de audio frecuencia. Indica el nivel de amplitud de la señal de entrada de audio del micrófono.
4. Control de volumen.

PANEL POSTERIOR

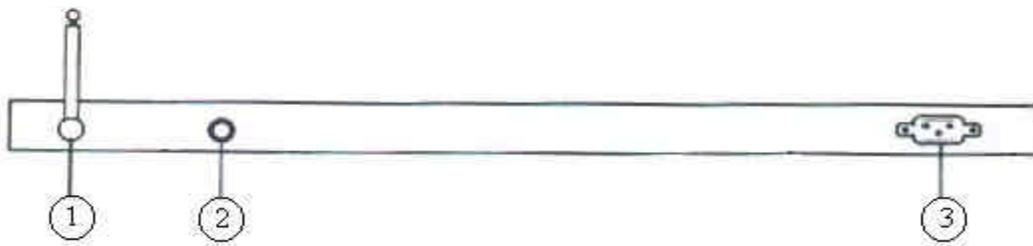


Figura 21. Panel posterior

1. Antena B: 75 Ohm.
2. Salida de audio balanceada.
3. Conector para el cable de red de alimentación de CA.

MICRÓFONO DE MANO.

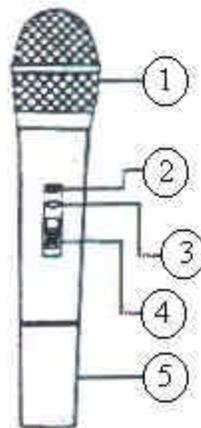


Figura 22. Micrófono

1. Cabeza protectora del micrófono.
2. Indicador de batería agotada.
3. Indicador de encendido / apagado.
4. Conmutador.
5. Cubierta de batería.

ESPECIFICACIONES TÉCNICAS DEL MICROFONO

Potencia de salida de RF	50 mW (máx).
Emisión de interferencia	Mínimo 40 dB por debajo de la Portadota.
Antena	Incorporada.
Pre-énfasis	50 μ S.
Elemento del micrófono	Dinámico unidireccional.
Batería	Standard de 9 V (006P).
Consumo de corriente	< 250 mA
Duración de la pila	Mayor a 4 horas de operación continua. Con batería alcalina, 8h.
Dimensiones Peso	Largo 240 mm, diámetro 55 mm.
Peso	280 g.

Finalmente la señal de indicación es por medio de un led, el cual se le hará una adecuación de señal para que pueda ser captado por el puerto paralelo del computador, que tendrá la función de estar leyendo información de entrada para activar el sistema de captación de la voz.

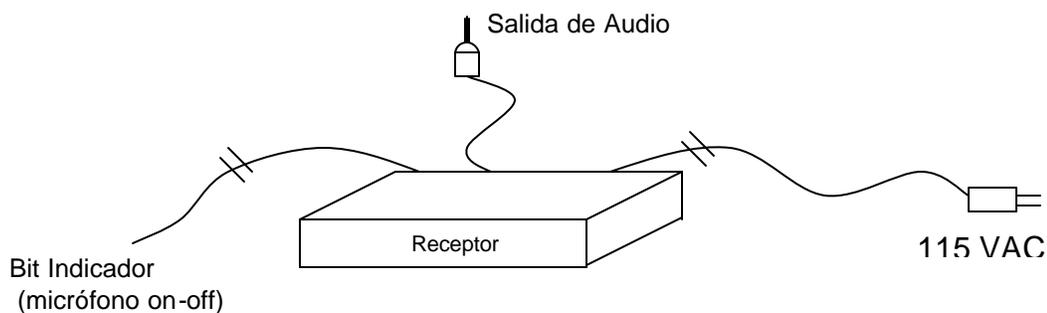


Figura 23. Conectores del Tx/Rx.

El sistema de adecuación de la señal de activación es un esquema sencillo, que consta de un comparador (741) con un voltaje de referencia establecido por circuito detector de micrófono. Este comparador discrimina el punto en que el micrófono está activado o no, esto se ilustra en la figura 24:

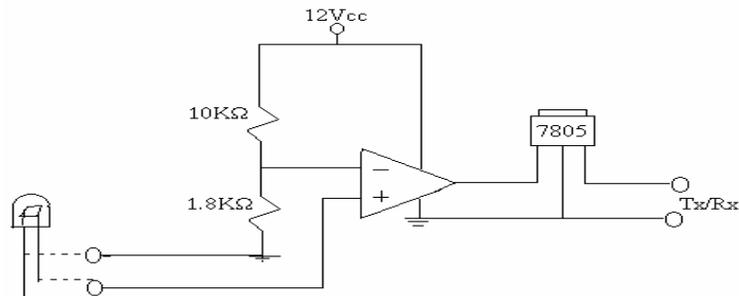


Figura 24. Comparador.

En la primera etapa, la entrada es la voz y la salida es, la voz y una señal de activación como se anoto anteriormente.

La segunda etapa tiene como variables de entrada la voz y un bit, que le indicara en que momento se realiza el proceso de recolección de la voz, y de salida 2 bits donde se representan las palabras reconocidas por el programa.

Ahora debemos predeterminar cuales son las cuatro ordenes que le daremos al motor para ello se seleccionaran cuatro palabras distintas que tengan diferentes niveles de energía y cruces por cero, para que el dispositivo pueda identificar claramente la palabras. Para ello sometimos las palabras a un editor de voz, que nos permitía observar el espectro de las señales de voz, y por medio del cual

seleccionamos cuatro en particular ENCENDER, APAGAR, IZQUIERDA Y DERECHA.

Las cuales arrojaron los siguientes señales de voz :

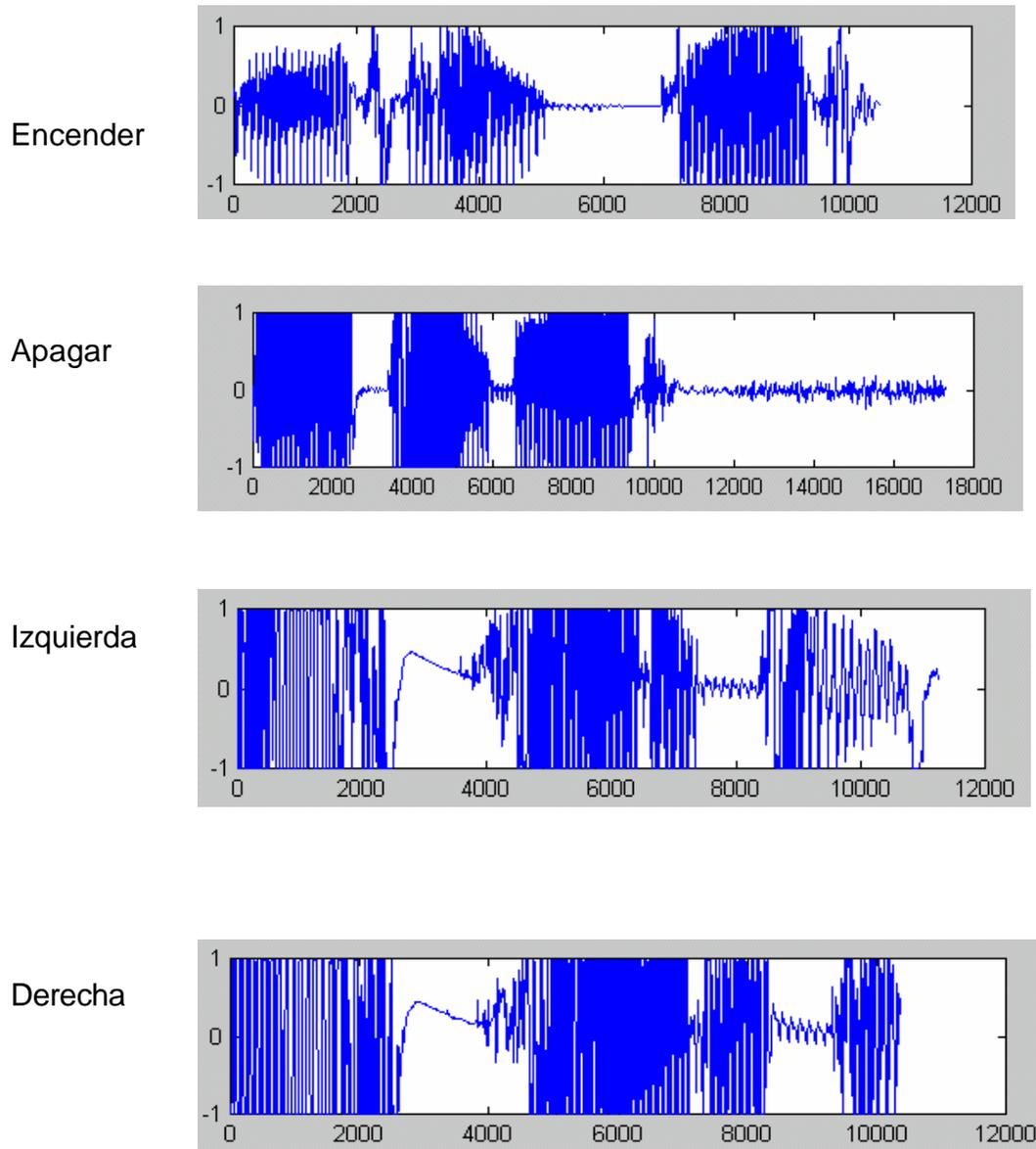


Figura 25. Espectro de las palabras

Seleccionadas las palabras procedemos al diseño e implementación del algoritmo de la red neuronal.

Inicialmente partimos de que las redes neuronales podemos detectar tres grandes procesos o fases dentro del entrenamiento estas son:

- Preprocesamiento.
- Procesamiento.
- Postprocesamiento.

Cada una con sus diferentes pasos los cuales seguiremos en la elaboración del algoritmo.

4.2.DISEÑO DEL ALGORITMO DE PROCESAMIENTO Y RECONOCIMIENTO DE VOZ

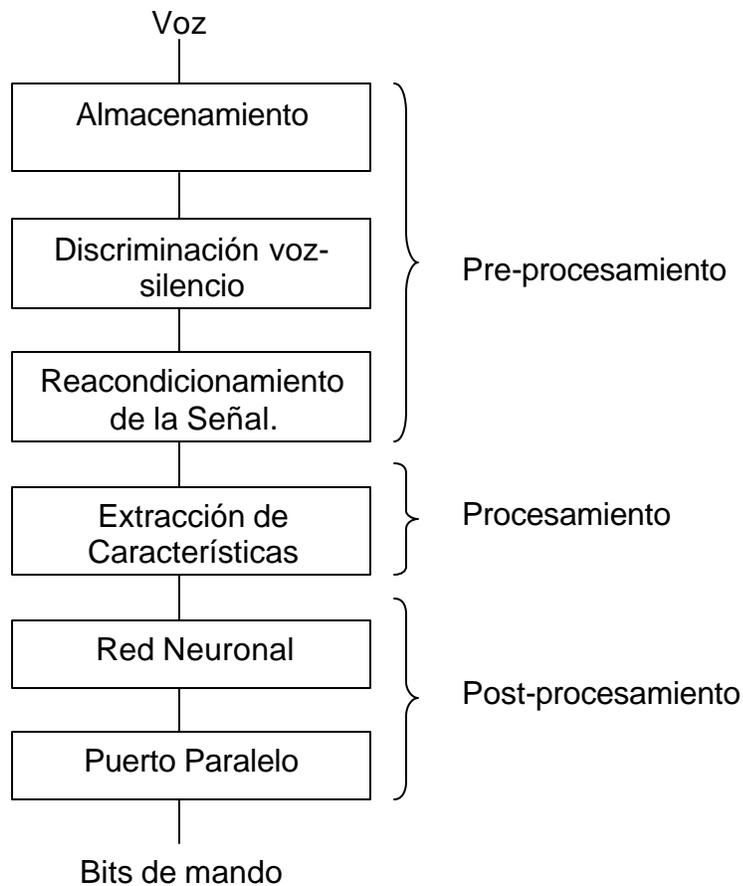


Figura 26. Diagrama de la etapa de entrenamiento

4.2.1.Pre-procesamiento.

El programa a realizar tiene fines didácticos por lo cual en cada fase se observaran ciertos items dentro de los procesos de reconocimiento de voz.

En la fase de entrenamiento se deben hacer estudios estadísticos de cuanto debe ser la cantidad de palabras necesarias para poder tener una referencia en el entrenamiento de voz.

Como en nuestro caso son cuatro palabras las que entrenaremos, tomaremos 10 muestras de cada una con el fin de tener la suficiente base de datos para que el reconocedor pueda entrenar la red.

En total tendremos 40 palabras que serán nuestra referencia de la voz. Se podrán pronunciar la misma palabra pero en diferentes tonos, con el fin de que la red pueda extraer las características particulares de cada palabra.

Dentro de la red neuronal es importante que las palabras sean grabadas en un mismo ambiente con el fin de que se pueda extraer ruidos característicos del medio donde estamos realizando la prueba.

4.2.1.1 Almacenamiento

El sistema es automático, por lo que la grabadora debe ser automática. Esta se desarrolló en el programa Delphi 5.0 (accesorios del proyecto), en un archivo .EXE, porque Matlab tiene la posibilidad de llamar archivos extensión ejecutable. El programa de la grabadora, debe estar sensando un bit de entrada del puerto paralelo (bit error 32), y al recibir una señal de alto, iniciara el proceso de

grabación de la voz, al retornar el bit a cero la grabadora se apagará instantáneamente. En el proceso de grabación tenemos que seleccionar una frecuencia de muestreo capaz de contener todas las componentes de frecuencia de la palabra en el proceso. Para ello utilizamos el teorema de Nyquist, el cual nos plantea que debemos hacer un muestreo al doble de la frecuencia máxima de la voz. Generalmente la mayoría de componentes de frecuencia voz se encuentra entre los 300hz y los 8 Khz, por lo que decidimos tomar de frecuencia de muestreo a 16Khz, que es una buena frecuencia para muestrear la voz.

Una vez tomado la muestra de voz es almacenada en un archivo .WAV, que nos permite escuchar la voz en un proceso de verificación, este archivo es leído desde Matlab por medio de una instrucción que realiza el proceso de lectura de los archivos con la extensión mencionada.

Como las cuatro palabras son diferentes, tenemos que realizar un proceso de detección de palabras, para poder encontrar el inicio y el final de cada una.

4.2.1.2. Discriminación Voz-Silencio

Para poder detectar las palabras se diseña un algoritmo que hace el proceso de detección partiendo de los datos que tenemos.

Debemos anotar que la red tomará por defecto una trama, en la cual el programa va a analizar el ruido del ambiente en un pequeño instante de tiempo. Por la

información recopilada ese instante de tiempo oscila entre los 100ms y los 500ms, en nuestro caso tomamos 200ms, después de varios procesos de pruebas. Después de transcurrido ese tiempo el programa debe proceder a detectar el comienzo y el final de la voz.

Para los cálculos del detector de límites de la palabra utilizamos los siguientes datos:

Señal muestreada a 16000 Hz

Número de bits 8

Tiempo para cada frame: $(1/16000)*(256)=16$ ms

Numero de frames para (200ms): $200/16 = 12$ frames

Frames traslapados: $(12*2-1)= 23$ frames traslapados

Numero de muestras de ruido: $256*12=3072$ muestras

Para el tope derecho: (Palabra más corta)

Longitud palabra medida en frames: $8000/256 \approx 32$ $32*2-1=63$

$(23+1)+63=87$ entonces $\text{tope}_i=24$ y $\text{tope}_d=113$.

Con estos datos el detector comienza a leer la información después del ruido ambiente, a partir de los frames traslapados, y luego analiza el comienzo de una palabra y el tope máximo que ella tiene. El problema que presentaba el programa es que al ocurrir una zona de silencio en la palabra, el programa lo detectaba como un comienzo por lo que tuvo que hacersele un ajuste al programa, este lo

realizamos con el fin de que en los silencios de la palabra el detector no produzca un corte debido a la ausencia de información y así se pierda la palabra, para ello en Matlab existen rutinas de detección de energía y cruces por cero, las cuales permiten que se detecte si en verdad sigue la palabra o finaliza completamente, el proceso de muestreo se realizo por prueba y error, se llego a la conclusión que se debía realizar cada 5 frames, comprendiendo que en este lapso debía por lo menos presentarse otro pico de energía de la palabra, si esta se estaba detectando en el momento.

4.2.1.3. Reacondicionamiento de la Señal

Terminado el proceso de detección de las palabras procedemos a realizar un Reacondicionamiento de la señal, este se hace por medio de un filtro digital de primer orden el cual se enmarca dentro de la etapa de pre-énfasis la estructura de la ecuación matemática que describe este proceso es:

$$H(Z)=1 - a z^{-1}$$

El valor de **a** es de 0,95 que es un valor típico para este tipo de tratamiento de señales, después del proceso de filtrado se debe diseñar el algoritmo de estandarización de la longitud, en este programa se analiza el tamaño de la palabra, por medio del numero de elementos que tiene el vector, se predetermina una longitud promedio de todas las palabras y se hacen comparaciones de si es

mayor o menor procediendo a redondear por medio de un comando (ROUND) que tiene Matlab. Después de terminado el proceso de muestreo se procede a remuestrear con el fin de poder establecer si la nueva estandarización es la adecuada, de no ser así el programa vuelve y revisa hasta que acomoda las palabras a una misma longitud promedio.

4.2.2. Procesamiento

El procesamiento de la voz es una de las etapas más importantes en el proceso de reconocimiento de voz, en él recaen las funciones de extracción de características y la fase de entrenamiento y validación del reconocedor. En esta etapa del reconocedor utilizamos varias herramientas de Matlab como son los toolbox de redes neuronales.

4.2.2.1. Extracción de Características

El método de extracción de características que se va a usar es el Melcepstral, que es conseguir los coeficientes melcepstral del vector de voz. Para la realización de este algoritmo utilizamos el comando melcepst, mediante el cual se extraen los coeficientes. los parámetros que debemos predeterminar son :

Y = Palabra vectorizada.

Tm = Frecuencia de muestreo.

Ncmel = número de Coeficientes Melcepstral.

WW = vector de características (Generación)

De donde la palabra vectorizada es la que ha sido pre-procesada con anterioridad, la frecuencia de muestreo es la que hemos venido trabajando en los items anteriores y que corresponde a la rata de muestreo del grabador de voz, el Ncmel son la cantidad de coeficientes Melcepstral que se usan en el método, los coeficientes son doce, el comando *melcepst* esta acompañado filtros y de funciones de ventaneo los cuales suavizan la forma de la señal de manera que la extracción de características es mas eficiente (ver anexo).

Terminado el proceso de extracción de las características procedemos a la fase de reconocimiento y validación.

4.2.3. Post-Procesamiento

4.2.3.1. Reconocimiento y validación.

En esta fase de reconocimiento se hace necesario una sub-fase de validación que nos permita establecer el logro de la red. Para ello validamos 40 palabras mas, en cuatro bloques de 10, cada uno con el fin de comparar con las palabras grabadas inicialmente este paso es importante por que le permite al usuario tener una

estadística del alcance de la red y además saber que tipo de errores se tuvo en la fase de diseño.

4.2.3.2. Entrenamiento y Simulación de la Red Neuronal

Una vez captada y adecuada la señal de voz; y también haber extraído sus características, estas últimas son guardarlas en el vector de características P.

El vector P va a constituir la entrada a la red neuronal para el entrenamiento, este vector se normaliza antes de introducirlo a la red neuronal con el fin de optimizar el proceso.

Para el entrenamiento de la Red Neuronal, Matlab cuenta con un comando llamado `trainbpx`, este comando tiene como función tomar las palabras que se grabaron en la etapa de entrenamiento y modificar los pesos de la red hasta que está última entregue el vector de salida esperado por el entrenador.

Dado que no existe un entrenamiento perfecto de la red, solo se puede “jugar” con una serie de parámetros propios del algoritmo de aprendizaje para así lograr que los resultados sean lo mas preciso posible, estos parámetros son:

Normalización de los parámetros de entrada y de prueba (df)

Máximo número de iteraciones (me)

Error máximo global (eg)

Tasa de apredizaje (lr)

Coefficiente de momento (momentum)

Tasa de error (err_ratio)

df=100;

me=1000;

eg = 0.1;

lr = 0.001;

momentum=0.95;

err_ratio= 1.04;

Los valores que presentan los parámetros anteriores son producto de un ensayo de prueba y error hasta lograr que la Red Neuronal haga un reconocimiento satisfactorio (mayor al 80% de efectividad). Los resultados de esta prueba pueden mejorar teniendo en cuenta las voces de validación y cambiando los parámetros mencionados anteriormente, para el caso de nuestra experimentación los resultados se pueden ver en la figura 27



Figura 27. Resultados de la fase de validación

En la figura anterior podemos observar la cantidad de aciertos que se obtuvieron en cada palabra, lo que nos hace pensar en que debemos mejorar las voces iniciales y los parámetros de entrenamiento, con el fin de obtener un reconocimiento de voz mucho más óptimo.

Finalmente en el tabla 1 podemos ver la hoja técnica de nuestra experiencia en el entrenamiento de la Red Neuronal.

Después de haber entrenado la Red Neuronal, sigue la fase de simulación, que será realizada por el comando simuff. En esta fase se introducirá el vector P ya normalizado en la red entrenada para obtener un vector de salida de 4 posiciones (una posición para cada palabra reconocida); que será analizado por otro

algoritmo que me asocie cada salida a cada una de las palabras que fueron entrenadas.

TÉCNICA DE PROCESAMIENTO	MelCepstral
TIPO DE RED	Multicapas
NUMERO DE CAPAS	3
NEURONAS DE LA CAPA DE ENTRADA	1224
NEURONAS DE CAPA OCULTA	3
NEURONAS DE LA CAPA DE SALIDA	4
NORMALIZACIÓN DE PATRONES	100
TASA DE APRENDIZAJE	0.01
MÁXIMO NÚMERO DE ITERACIONES	1000
ERROR MÁXIMO GLOBAL	0.1
COEFICIENTE DE MOMENTO	0.95
TASA DE ERROR	1.04
ACIERTOS “ENCENDER”	7
ACIERTOS “APAGAR”	7
ACIERTOS “IZQUIERDA”	9
ACIERTOS “DERECHA”	9

Tabla 1. Tabla técnica

4.2.3.2. Salidas de la Red Neuronal

Las salidas de la red neuronal están representadas por un vector de cuatro posiciones; cada una de una las posiciones tiene un valor que oscila dentro de cierto rango, en donde si se produjo un buen reconocimiento de la palabra solo una de las posiciones tendrá un valor elevado (cercano a 1), mientras que las demás posiciones tenderán a cero. El programa de identificación de la palabra reconocida opera basándose en los valores que entrega la capa de salida de la

red, y también ejecuta cada uno de los archivos que habilitan los bits de salida del puerto paralelo del PC.

Los listados de los principales archivos utilizados en el programa de reconocimiento de voz son se encuentran en el ANEXO B

4.3. SELECCIÓN DEL MOTOR

4.3.1 Fundamentación Teórica.

4.3.1.1. Características en vacío

Por características en vacío, se entiende a una curva que representa la f.e.m. generada en vacío, en función de la corriente de excitación, siendo nula la carga eléctrica del generador y la velocidad constante. Esta característica se puede deducir teóricamente de la ecuación de la f.e.m. inducida en una máquina de C.C, así:

En máquinas sabemos que: $E_o = k f_o n$ voltios (1).

Donde:

E_o f.e.m. inducida en vacío, debida al campo magnético principal.

f_o flujo magnético creado por el devanado de excitación. (V-S)

- n** velocidad del rotor de la máquina. (r.p.m.)
- k** Constante que depende de las características propias de cada máquina y de la unidades utilizadas en la ecuación (1), así:

$$k = \frac{r}{a} \cdot \frac{N'}{60} \quad (2). \quad \text{Donde:}$$

- r** número de polos.
- a** número de circuitos en paralelo.
- N'** número de conductores del inducido.

Si por definición la velocidad ha ser constante se deduce que la variación de E_o será proporcional a la variación del flujo, pero:

$$I_e N_{exc} = \sum H_{fe} + \sum H_{ad} \quad (3)$$

La ecuación (3) representa la f.m.m. o amperios-vueltas creados por la corriente de excitación, además el flujo es proporcional a la inducción y este a la intensidad de campo H. En la ecuación (3) la primera sumatoria equivale a la f.m.m. requerida para enviar a través de los elementos de hierro o acero el flujo ϕ y la segunda sumatoria es la requerida para que el mismo flujo atravesase él entre hierro o aire. Por consiguiente la ecuación (1) se puede representar así:

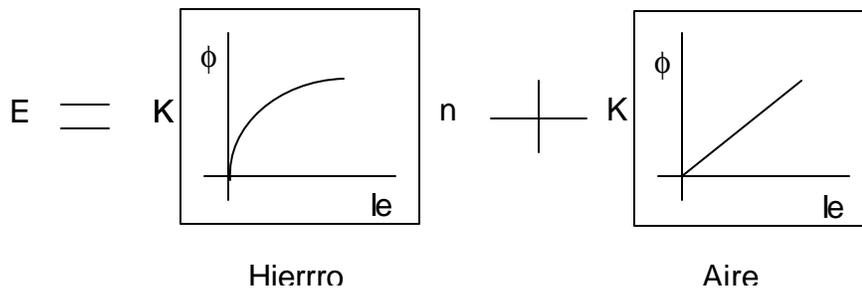


Figura 28. Representación grafica de la ecuación.

Con lo anterior se pretende señalar que la característica de vacío corresponde a la suma de dos características, una línea que corresponde al aire y la otra semejante a las curvas magnéticas B vs H que corresponde al hierro, tal como se representa en la figura 29. La tensión inducida e_r , cuando la corriente de excitación es cero se denomina tensión remanente y se debe a la remanencia magnética, la recta que coincide con la parte lineal de la característica se denomina Recta de entre-hierro.

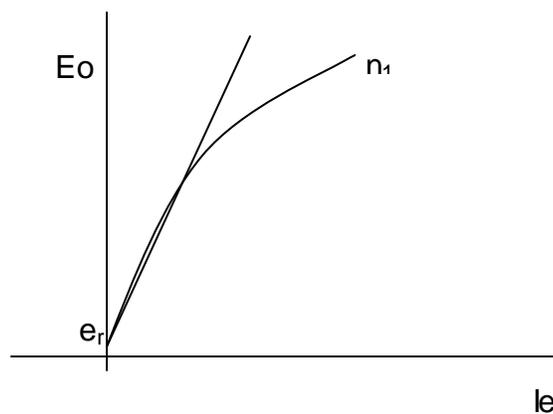


Figura 29. E_o vs le

4.3.1.2. Especificaciones técnicas del motor de excitación independiente

Potencia	:1HP
Voltaje de C.C.	:230V DC
Resistencia del Inducido	:6Ω
Resistencia de la bobina de excitación R_f	:1220Ω

4.3.1.3. Extracción de la curva de velocidad (voltaje-corriente) del motor de c.c.

Para su obtención se monta el circuito de la figura 30. Haciendo girar el generador a velocidad constante y como se puede observar, se excita con una resistencia R_e que permita regular la corriente de excitación de mínimo a máximo. Antes de energizar el circuito de excitación y con el generador girando a velocidad deseada, se debe medir una tensión, la cual corresponde al flujo remanente e_r . Luego energizamos el circuito de excitación con R_e en máximo. Se toman valores de tensión inducida y corriente de excitación; posteriormente se van tomando los mismos valores para diferentes corrientes de excitación, esto siempre en forma ascendente y sin devolver el reostato R_e , pues de lo contrario se forman ciclos de histéresis.

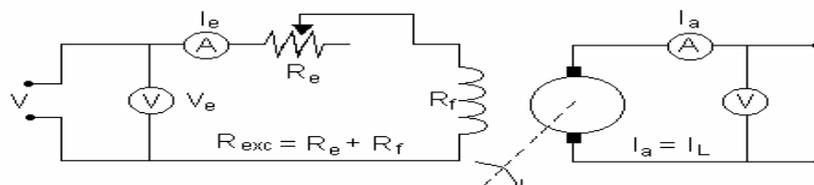


Figura 30. Montaje del circuito de prueba.

Los datos obtenidos están consignados en la tabla 2 que se presenta a continuación.

Ve(V)	Va(V)	Ie(mA)
160	9.0	100
160	9.3	117
160	9.5	118.2
160	9.65	118.5
160	9.85	118.9
160	10.35	119.2
160	10.5	119.2
160	10.7	119.2

Tabla 2. Voltaje excitación vs corriente Ie.

4.4. DISEÑO DEL CIRCUITO DE INVERSIÓN DE GIRO

El diseño del circuito inversor de giro se hizo a partir del motor seleccionado y remitiendo nos a las sugerencias que se hicieron en el anteproyecto en donde se pedía un motor de mayor potencia, para el cual se dimensionaron los elementos utilizados.

Potencia del motor = 1 Hp = 745.6W

Voltaje del motor = 230 V

Inominal = 3.24 A

Voltaje de trabajo= 45 V

La siguiente fase fue seleccionar el inversor de giro que se va a utilizar, llegamos a la conclusión de que él mas adecuado seria el de puente H con reles como se ilustra en la figura 31 y que procederemos a explicar a continuación.

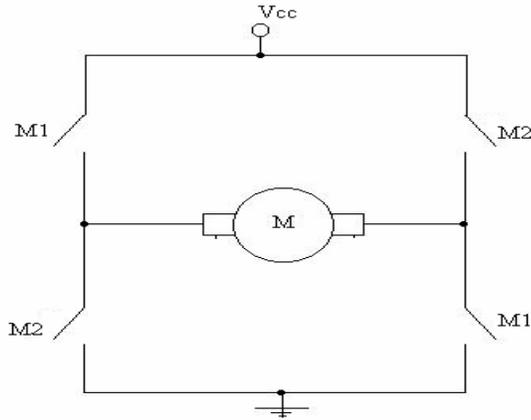


Figura 31. Inversor de giro

Para poder definir como es la activación de los relés en la etapa de potencia tenemos que remitirnos a las señales de activación procedentes del circuito de control el cual vamos a reseñar como fue su diseño.

Primero partimos de que las señales de salida del computador están representadas en las combinaciones de los dos bits del puerto paralelo, osea en las combinaciones mostradas en la tabla 3.

BIT (D1- D0)	RESPUESTA
0 0	APAGAR
0 1	DERECHA-ENCENDER
1 0	IZQUIERDA
1 1	NO VALIDO (APAGAR)

Tabla 3. Combinaciones de bits

Estas combinaciones lógicas deben producir solo una respuesta como se indica para poder activar las parejas dependiendo la orden. El circuito que desarrollamos es uno con compuertas lógicas como se puede ver en la figura 32, este provee la respuesta que necesitamos para manejar la inversión de giro.

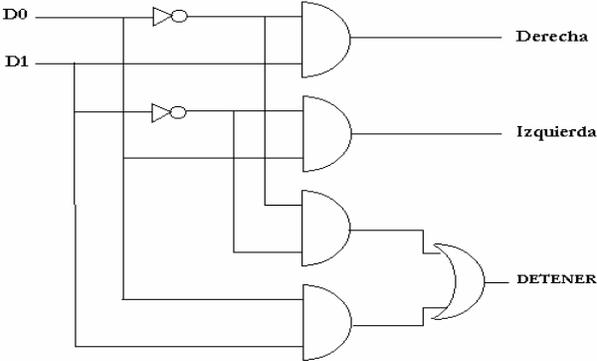


Figura 32. Lógica de control.

Esta señal proveniente del circuito es muy baja por lo que se hizo una etapa amplificadora con transistores de potencia el esquema representativo del diseño lo podemos ver en la figura 33.

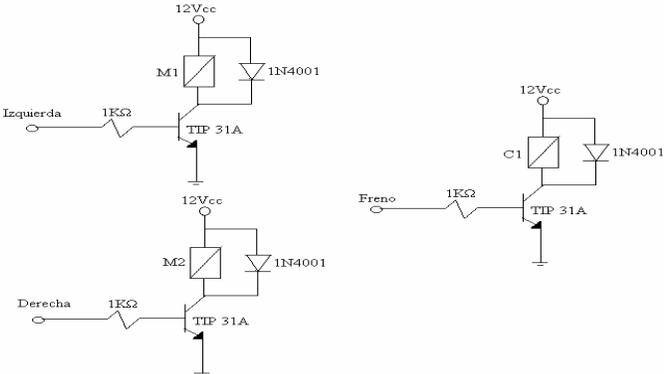


Figura 33. Etapa amplificadora

La señal que provenía del circuito lógico es capaz de excitar la base de los transistores y producir una corriente tal que la bobina de los relés sea excitada y por lo tanto se haga el cambio de estado en los contactos. Los contactos de los relés se encuentran en la etapa de inversión de giro figura 31, ahora procedemos a explicar el proceso de inversión de giro.

Al recibir los transistores una señal por ejemplo derecha, la pareja numero (M1) que conforman los contactos de los relés M1, pasan de un estado normalmente abierto a un estado normalmente cerrado, permitiendo la referencia positiva como se indica en la figura 34.

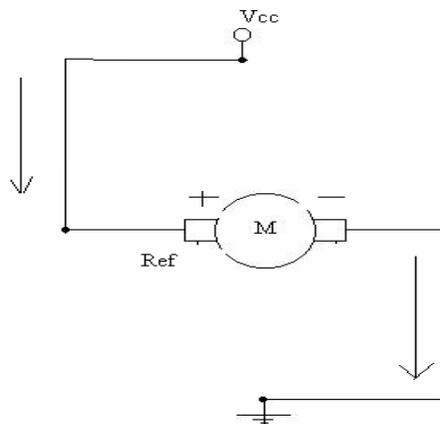


Figura 34. Sentido positivo.

Caso contrario ocurre en el proceso de activación de izquierda donde la señal activa la pareja M2, cerrándose sus contactos lo que permitía que se efectuara un cambio de polaridad en la referencia y por lo tanto una inversión de giro, en la figura 35, se puede observar el efecto del cambio de referencia.

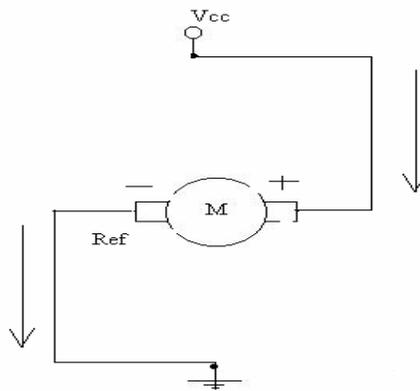


Figura 35. sentido negativo.

El circuito de frenado es igual solo ahora los contactos de los reles están sirviendo como interruptor en la cantonera permitiéndole el uso solo en el caso de que la bobina presente excitación que sería el caso de que se envíe una señal del circuito lógico.

5. ACCESORIOS DEL PROYECTO.

Esta parte del proyecto la dedicamos a lo que corresponde a la parte de detalles, en él haremos alusión a los elementos externos a la plataforma de Matlab y elementos físicos que se usan en el proyecto por ello mencionaremos:

- Programa de grabación de voz.
- Programa de salidas al puerto.
- Manual del usuario.
- Manual de instalación.

5.1. DISEÑO DE UNA GRABADORA DE VOZ (EN DELPHI5.0)

El diseño de la grabadora de voz se hace en Delphi5.0 utilizando las herramientas visuales que nos presentan en esta versión, el diseño de la grabadora se hace partiendo de dos consideraciones como son la de uso manual y uso automático, como nuestro proyecto es de tipo automático hicimos énfasis en que la señal de activación debe ser de un agente externo por lo que se tomo un bit del puerto

paralelo que nos permita sensor una señal de activación, el bit seleccionado fue el de error por ser un bit altamente confiable y generalmente usado por el puerto, en el listado del programa se creo una rutina que permanentemente estuviese detectando a un tiempo de muestreo de 100ms, esta información esta condicionando un proceso, si el bit es de encendido se llamaba la grabadora que tiene el programa, que es una función ya predeterminada dentro de Delphi5.0 lo que inicia un proceso de grabado y de almacenado en una dirección ya establecida. Si no es así el programa debe mantenerse sensando hasta que detecte un cambio en el puerto. El listado del programa se encuentra en el ANEXO C.

5.2.DISEÑO DE LOS ALGORITMOS DE INTERFASE MATLAB–PUERTO PARALELO (EN DELPHI5.0)

Los algoritmos de la interfase de Matlab con el puerto paralelo parten de la posibilidad de comunicarse desde Matlab con un archivo de extensión ejecutable, teniendo en cuenta esto utilizando nuevamente Delphi5.0 se hacen algoritmos los cuales permitan enviar la señal de actuación al motor por el puerto paralelo, este programa solo debe colocar un 0, 1, 2 o 3 en binario por medio de los dos bits que tiene, osea la combinación de D0 y D1, y finalmente debe mantener un mensaje de reconocimiento dependiendo de la palabra que haya detectado, el tiempo de el mensaje se predetermino en 3 segundos, para la facilidad del usuario,

permaneciendo la información en los bits de salida debido a la capacidad de almacenamiento del latch de datos. Los listados de los programas pueden verse en el ANEXO D

5.3. MANUAL DEL USUARIO.

El manual que presentaremos a continuación es una extensión del trabajo, con la finalidad de facilitar al usuario el manejo del software de reconocimiento de voz.

El proyecto de reconocimiento del habla lo podemos dividir en dos grandes secciones:

- Fase de entrenamiento.
- Fase de prueba.

El manejo del programa se hace bajo el ambiente Matlab, se sugiere que sea en la versión Matlab 5.2. la cual tiene los toolbox de Neural Network, que nos facilitaran el desarrollo del programa, además esta versión tiene otras opciones que son bien importantes en el desarrollo del programa.

El nombre del programa es RECVOZ1.0 el cual nos presenta un pantallazo de ambientación y de presentación del mismo, como se ve en la figura 36.



Figura 36. Presentación

La cual consta de los nombres de los integrantes de la tesis, además de tres opciones que son:

Ayuda: Opción diseñada para dar proveer al usuario del manual y dar breve información de las redes neuronales, del contenido del programa y generales de todo el desarrollo de la tesis.

Salir: Es la opción prevista para cuando se quiere retornar a la plataforma Matlab dándole final al uso del programa.

Siguiente: Es el enlace a los dos principales partes del proceso de reconocimiento de voz, en él podemos encontrar las dos secciones principales como se muestra en la figura 37.



Figura 37. Menú principal.

La primera sección es el entrenamiento el cual se hace en fases progresivas, la segunda sección es la fase de ejecución del sistema en general, el botón de salida es un retorno al formato de presentación del programa.

A continuación en el manual del usuario veremos los dos grandes procesos o secciones como son el entrenamiento y prueba, dando las respectivas recomendaciones para cada uno.

5.3.1. Entrenamiento de la red.

Al presionar el botón de entrenamiento nos encontraremos con un menú de entrenamiento como se ilustra en la figura 38.

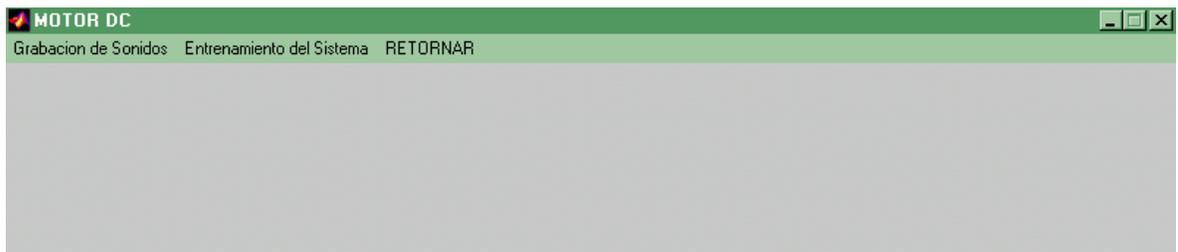


Figura 38. Menú de entrenamiento

En el se observan dos fases claves en el reconocimiento de voz como lo son el grabación de sonidos y el entrenamiento del sistema.

Grabación de sonidos: En el cual se hace la fase de captación de la voz y la adecuación de la misma.

La grabación sonido tiene tres opciones ver figura 39.



Figura 39. sub-menú de grabación de sonidos.

Voces iniciales: Son las voces que grabamos en la fase de inicial del proceso, estas voces nos servirán como base de datos para el entrenamiento.

Voces de validación: Son las voces de prueba las cuales nos sirven de punto de comparación con las voces iniciales y que son parte previa al proceso de adecuación de voces.

Tanto en las voces de validación como en las voces iniciales encontramos submenú iguales como en la figura 40 donde se hallan las palabras a entrenar.

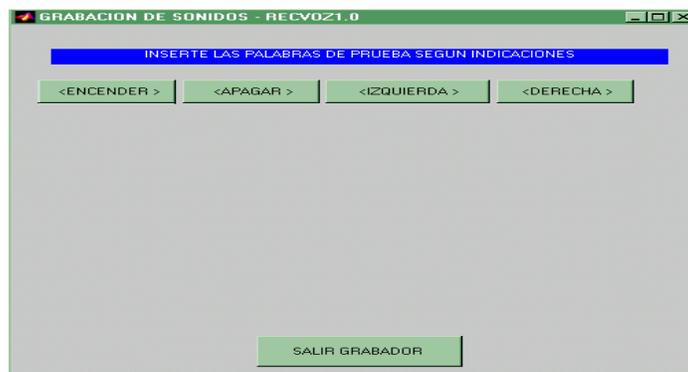


Figura 40. Grabación de sonidos.

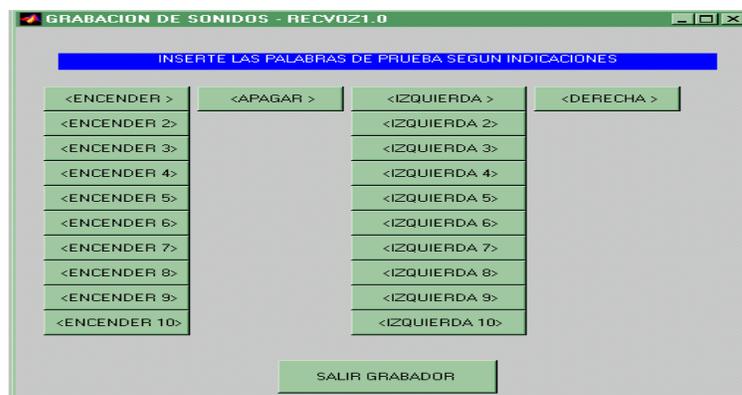


Figura 41. Palabras a entrenar

Al seleccionar alguna de las palabras se extiende una columna de palabras donde se indica la cantidad de palabras de la misma que se deben grabar para poder entrenar a la red. Figura 41.

Al presionar el botón indicador de cada palabra se dará comienzo al proceso de grabación de voz, llamando al programa grabador de voz desarrollado en Delphi 5.0 que tiene un activador manual como se ve en la figura 42, y otro automático, que dependerá de la señal de encendido del micrófono recibida por el puerto paralelo y que inicia el proceso de captación de voz.



Figura 42. Grabadora de voz.

Terminado el proceso de grabación de voces se le recomienda al usuario se verifique el directorio raíz **“C: Matlab/recvoz/voces1”** y **“C:Matlab/recvoz/voces2”** donde se almacenan las palabras en formato **.wav**, si se encuentran todas almacenadas, si no es así vuelva al entrenamiento y fase de grabación e intente de nuevo con la palabra. Si no se hace lo anterior el programa arrojará resultados fallidos en el proceso de reconocimiento de la voz.

Luego de la verificación del proceso regrese al programa a la fase entrenamiento y presione la opción **Edición de voces**.

Edición de voces: Es donde se realiza el proceso de edición de voces, en el se realizan la detección de bordes, la delimitación de los vectores, el relleno de ceros en los vectores, y el proceso de adecuación en general.

El programa por medio de mensajes le informara cuando el programa de edición halla terminado, el proceso de reconocimiento es lento por lo que se recomienda esperar los mensajes de continuación. Una vez recibido el mensaje se procede con la segunda fase el procesamiento de la voz.

Entrenamiento del sistema: En esta parte se hace la extracción, reconocimiento y entrenamiento de la red neuronal. En este identificaremos dos opciones procesamiento y reconocimiento, ver figura 43.

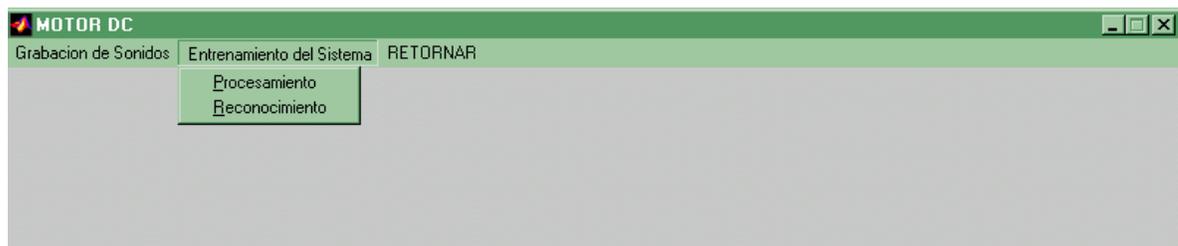


Figura 43. sub-menú de entrenamiento.

Procesamiento: es la parte que comprende la actualización de datos obtenidos en la adecuación de voz y la extracción de características que para nuestro caso es el algoritmo Melcepstral, que es el extractor de coeficientes del reconocedor.

Reconocimiento: a pesar de ser parte del postprocesamiento aquí lo tendremos en cuenta el procesamiento para fines didácticos. Este es la función que entrena la red y que permite establecer los parámetros finales con los cuales la red va a crear su propio red de reconocimiento. La selección de los parámetros de entrenamiento esta predeterminada en el software por lo que el usuario solo tiene la posibilidad de activar el entrenamiento, las capas y la cantidad de neuronas deben ser programadas desde software.

Una vez se han realizado los dos procesos el programa indicaras la finalización de la fase de entrenamiento, en caso tal debe regresar a la fase de prueba por medio de la opción retornar, que se observa en la figura 38, la cual fue mencionada con anterioridad.

5.3.2. Prueba

Al seleccionar la opción prueba en el menú principal entramos otro menú el cual tiene una extensión a dos opciones como se puede ver en la figura 44.

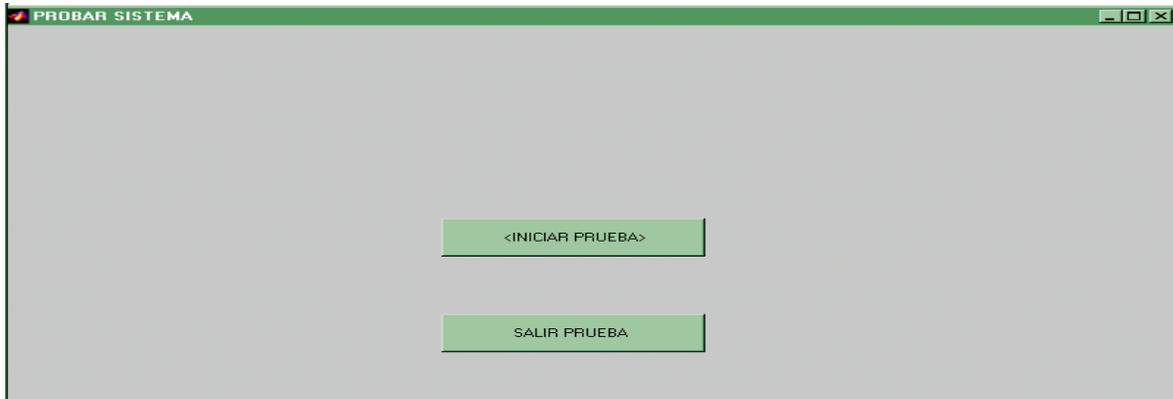


Figura 44. Fase de prueba

Salir de prueba es la opción de regreso al menú anterior, mientras que **prueba** inicia un proceso de activación.

El proceso de prueba es automático y es el trabajo del sistema en general, en el ocurrirán procesos continuos los cuales realizan el reconocimiento de voz, teniendo en cuenta en un primer paso la recepción de la voz por medio de la grabador la cual aparece en el momento de encender el micrófono.

En un proceso continuo se realiza la edición de las palabras, luego la extracción de características para finalizar con el reconocimiento de voz del cual se habla en el trabajo de grado, todos estos procesos son automáticos por lo que el proceso

de verificación se hará por ventana hechas en Delphi5.0 y en la parte física por el motor. Las cuatro ventanas son:

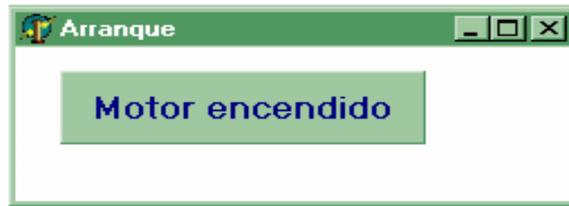


Figura 45. Motor encendido.

Para indicar que se dio inicio al proceso de activación del motor.

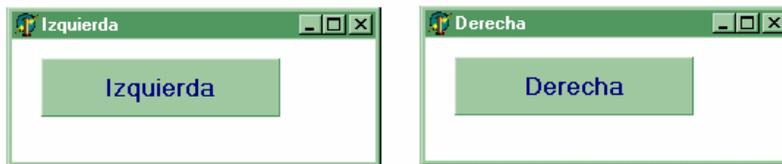


Figura 46. Giro hacia izquierda o derecha.

Para indicar hacia adonde gira el motor en el proceso de trabajo

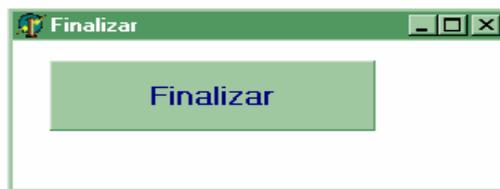


Figura 47. Apagado del motor

Como la finalización del proceso o apagado del motor.

Finalmente a pesar de todas las ventajas de las redes neuronales puede existir la posibilidad de que la palabra no sea reconocida, ya sea por Interferencia producto del medio o la pronunciación no adecuada. Esto esta dentro del rango de precisión de la red, para el caso de que la red no reconozca, esta emitirá un mensaje.

El cual es una invitación a volver a probar con otra palabra o con la misma si así se desea. Aunque las redes neuronales se ha avanzado aun el proceso de simulación neuronal no esta en un 100%, pero se siguen haciendo estudios para mejorar el proceso de reconocimiento.

Finalmente recordar al usuario que el indebido uso del manual y la no correcta realización de pasos es la posible causa de errores que se presenten en el reconocedor de voz (**RECVOZ1.0**).

5.4.MANUAL DE INSTALACIÓN.

El proyecto consta de dos partes en su instalación una es la parte física y otra la instalación del programa, en el manual que veremos a continuación se hacen las respectivas explicaciones de cómo es el mecanismo para poder instalar ambas partes.

5.4.1. Estructura física.

La estructura física la podemos describir en tres bloques asociados entre sí.

- Bloque de transmisión: el cual consta de las entradas y salidas de Tx/Rx que utilizamos en el proyecto.

- Bloque de PC donde se activan la estradas y salidas del computador en nuestro caso el puerto paralelo y la tarjeta de sonido.

- Bloque del motor en donde se asignas las respectivas conecciones del motor.

5.4.1.1. Bloque del Tx / Rx.

Un grafico simplificado del sistema de transmisión es el que mostramos a continuación:

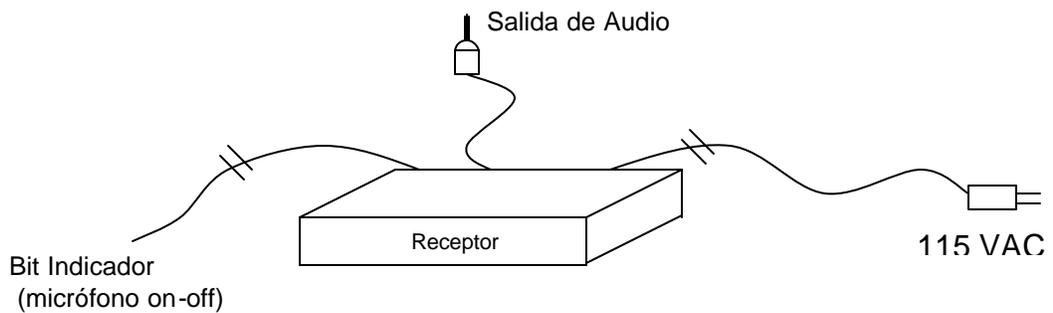


Figura 48. Esquema del receptor.

En él indicamos los tres conectores que tiene el receptor y que van dirigidos a distintas partes.

El cable 115 VAC es un cable común de par trenzado y el cual va conectado a la fuente de alimentación de 115 VAC como lo indica la figura, este voltaje puede ser tomado de cualquier toma disponible en el laboratorio.

El cable de salida el Rx es un cable estereo el cual tiene en su terminal un adaptador a mono con la finalidad que pueda ser compatible con la entrada de la tarjeta de sonido elemento que será finalmente el punto de conexión.

El cable del bit identificado es un típico RJ11 conocido como conector de teléfono a el cual fue acoplado al trasmisor, esta terminal va conectada a la caja base donde se encuentran los circuitos electrónicos en la entrada indicada.

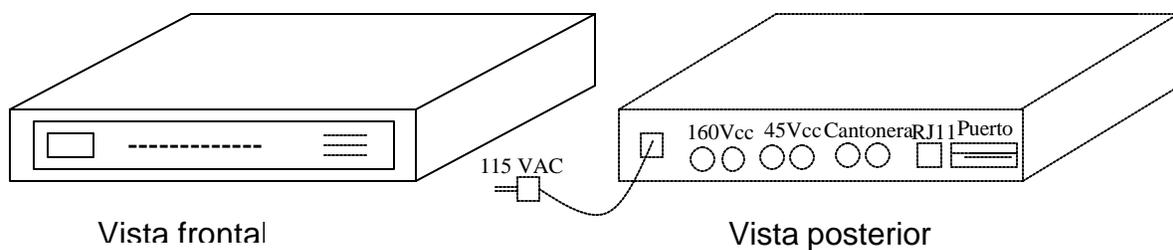


Figura 49. Caja base

5.4.1.2. Bloque del PC.

El bloque del computador es más sencillo en el solo tenemos en cuenta dos cosas, el puerto paralelo y la entrada de la tarjeta de sonido esta ultima va conectado donde sea indicado por el computador en este caso no particularizamos la conexión porque dependiendo de cual sea la unidad, esta tendrá su entrada de audio o multimedia en diferentes parte y lo mismo pasa con el cable del puerto de salida que esta colocado en cierta parte dependiendo de la unidad en uso. Ahora la salida del cable del puerto esta conectado a la entrada que es encuentra en la caja base como se indica en la figura 50.

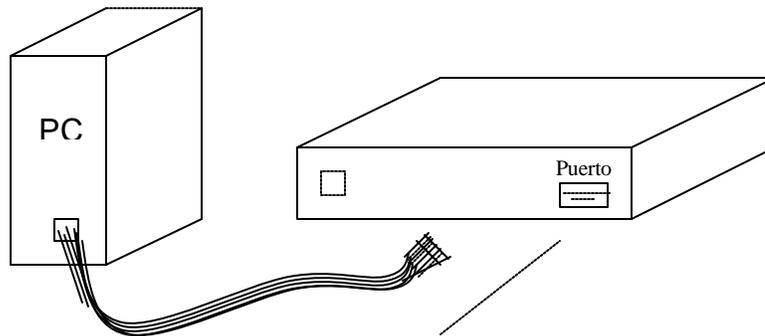


Figura 50. Conexión del cable del puerto.

Este cable contiene las informaciones tanto de la entrada del bit indicador del transmisor como la salida de los bits de control en el proceso de apagado, encendido y giro del motor.

5.4.1.3 Bloque del motor (caja negra).

Este bloque cuenta con una entrada y tres salidas, la entrada es la que se menciona como salida del computador y que se maniobra internamente, las salidas son los dispositivos de alimentación para el motor. Ellos son los 160 Vdc del estator, 45 voltios del rotor y el voltaje de activación para la cantonera.

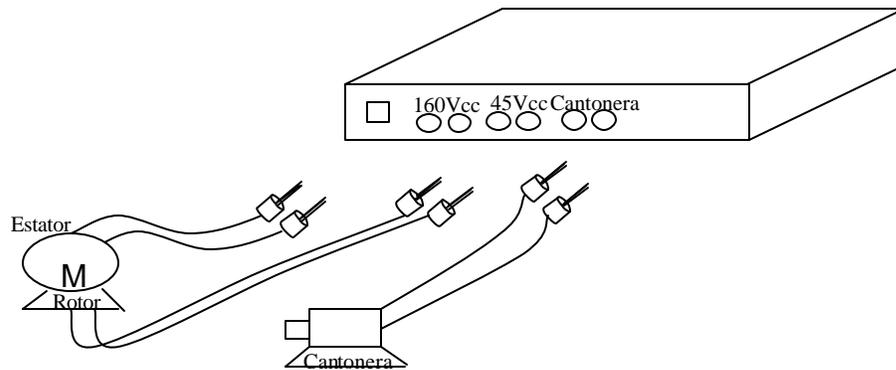


Figura 51. Conexiones del motor.

Las conexiones se hacen tal como lo indica la caja negra en la grafica 51.

5.4.2. Software

Para la instalación del programa deben estar configurados la plataforma Matlab5.2

- En primera instancia deben estar instalados los toolbox de “Signal Processing” y “Neural Networks”.
- Es necesario indicar que los siguientes directorios deben previamente crearse desde el explorador de Windows:

C:\Matlab\recvoz : grabar los programas fuentes.

C:\Matlab\recvoz\patrones\vocesinicialaes : grabar el conjunto de entrenamiento.

C:\Matlab\recvoz\patrones\vocesvalidacion : grabar el conjunto de validación.

C:\Matlab\recvoz\patrones\voces1editadas : grabar el conjunto de entrenamiento editado.

C:\Matlab\recvoz\patrones\voces2editadas : grabar el conjunto de validación editado.

C:\Matlab\recvoz\patrones\signal : grabar la palabra a reconocer

Para hacer esto se recurre a Matlab en el path browser como indica la figura 51.

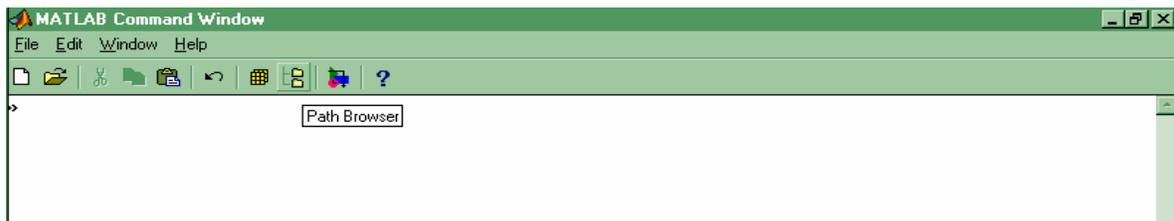


Figura 52. Path browser.

Se posiciona en la opción de path y se selecciona el add to path... con el cual se busca la rutina deseada como se muestra en la figura 53.

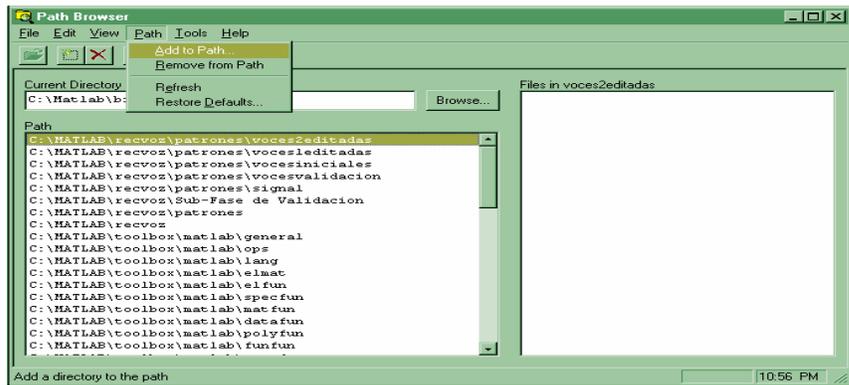


Figura 53. selección de add path.

Al abrir add to path se abre una ventana figura 54. la cual comunica con un browser de donde se puede localizar la dirección estipulada como raíz para crear las rutinas de los directorios.

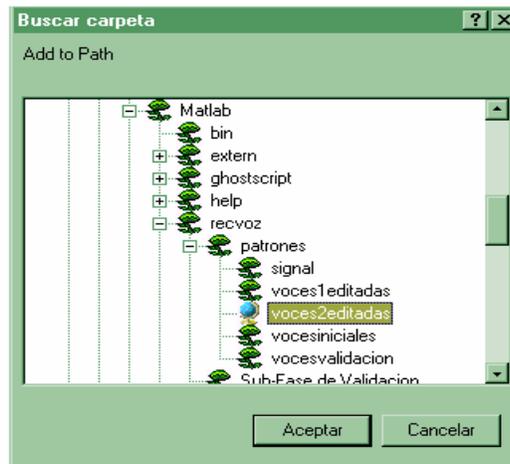


Figura 54. Búsqueda de archivos raíz.

Terminado este proceso podemos decir que la fase de instalación del software ha terminado, y con ello el proceso de instalación del proyecto en general, ahora

podemos proceder a trabajar utilizando el comando **RECVOZ1.m** que debe ser llamado bajo la plataforma Matlab.

6. ASPECTOS DE INFRAESTRUCTURA.

6.1.RECURSOS HUMANOS

Director : Eduardo Gómez. Ingeniero Electricista.

Asistente : Margarita Upegui. Ingeniera Sistemas.

6.2.DESCRIPCIÓN DE EQUIPOS Y HERRAMIENTAS REQUERIDAS;

DISPONIBLES EN LA CUTB:

- ✓ Computador Pentium 233Mhz
- ✓ Osciloscopio 20Mhz.
- ✓ Multímetros.
- ✓ Generadores de señales.
- ✓ Matlab 5.2 con toolboxes de procesamiento de voz y de redes neuronales.

6.3.LABORATORIOS

- ✓ Laboratorio de electrónica Corporación Universitaria Tecnológica de Bolívar
Campus de Ternera.
- ✓ Laboratorio de informática Corporación Universitaria Tecnológica de Bolívar
Campus de Manga.

7.OBSERVACIONES Y CONCLUSIONES

Después de la realización del presente proyecto de investigación el grupo ha concluido lo siguiente:

- Hemos llevado a cabo un sistema de reconocimiento de palabras aisladas, monolocator, basándonos en el entrenamiento de una Red Neuronal que simula el proceso de captación del habla.
- La adquisición de muestras se realiza utilizando una tarjeta de sonido estándar, programando para ello una grabadora de voz que se activa con un bit en el puerto paralelo, permitiendo así que la grabadora pueda ser activada de forma inalámbrica.
- Para lograr la captación de la voz se requiere de un sistema de radiofrecuencia de baja potencia, de tal forma que la persona realice el mando se encuentre alejada a 50mts del motor y así el mando pueda ser a distancia. El receptor de RF debe tener un conector que permita llevar la información de audio directamente a la tarjeta de sonido.

- Utilizando el modelo Melcepstral, se hizo posible extraer el vector de características de la señal de voz, tomando como parámetros los 10 primeros coeficientes del modelo.

- Para evitar utilizar información no representativa en el proceso de reconocimiento (silencios), se han limitado los datos del registro de reconocimiento a la palabra pronunciada, utilizando para ello un discriminador palabra/silencio implementado en base a medidas temporales (energía y cruces por cero).

- Para analizar la respuesta que nos entrega la Red Neuronal se utiliza el método de la distancia, en donde se toma el vector de salida de la red y la distancia que posee el mayor valor con respecto a los otros tres me indica que palabra ha reconocido la Red Neuronal

- Tras un proceso el de entrenamiento del sistema y sintonización de los pesos, se procede a la comparación de las características de los patrones de referencia con los parámetros de la palabra a reconocer. Hemos utilizando para ello el algoritmo Backpropagation.

- El sistema de reconocimiento permite enviar comandos en función de la palabra reconocida a través del puerto de comunicación paralelo, ejerciendo mando sobre el Motor. La señal de salida del puerto es llevada una interfaz de potencia, en donde se activan y/o desactivan una serie de relés que hacen posible que el motor ejecute las acciones izquierda, derecha, encender y apagar.

- Se pueden cambiar las palabras de mando. Por ejemplo, la palabra encender se puede cambiar por arrancar, pero para ello habría que cambiar los parámetros iniciales de la Red Neuronal para que el entrenamiento pueda adaptarse a los cambios realizados.

- Tras evaluar el sistema de reconocimiento, en el que se consiguieron índices de error muy bajos, podemos concluir que el objetivo marcado al comienzo de este trabajo se ha conseguido de forma satisfactoria.

8.GLOSARIO

Se presenta una breve explicación de algunos términos técnicos:

AMPLITUD: Es la distancia existente desde el punto de repaso a un punto de máximo alejamiento de una partícula en vibración.

ANCHO DE BANDA: Es la gama de frecuencias a las que responde con efectividad un resonador o un filtro, es decir, el margen de frecuencias dentro del cual cada componente presenta una amplitud equivalente al menos al 70.7 por 100 de la que presenta el componente con mayor amplitud. La extensión de la frecuencia efectiva de un resonador se denomina ancho de banda. Se calcula convencionalmente considerando sus límites entre aquellos armónicos cuya amplitud a la salida es el 70.7% de la amplitud de salida de la frecuencia resonante.

APARATO FONADOR: Es el sistema que produce el habla y está formado por tres elementos: Un generador de energía (constituido por los pulmones), un sistema vibrante (constituido por la laringe y las cuerdas vocales) y una cavidad resonante (constituida por el conducto vocal).

ARMÓNICO: También denominado hipertono, un armónico es una frecuencia componente de la onda sonora que es el múltiplo de la frecuencia fundamental. Por ejemplo, si una onda tiene una frecuencia fundamental de 200 Hz, los componentes de 400 y 600 Hz se conocen como segundo y tercer armónico.

ARTICULACIÓN: Conjunto de movimientos realizados por los órganos articuladores para interrumpir o modificar la salida de la corriente de aire al exterior durante la emisión de cada sonido. También suele denominarse articulación la posición adoptada conjuntamente por los órganos en el momento de la producción del sonido.

BANDAS FORMÁNTICAS: Son zonas espectrales en la que se concentra la energía acústica.

DASE DE DATOS DE ENTRENAMIENTO: Compuesta por varias pronunciaciones de las unidades a modelar.

CANCELADOR DE ECOS: Proceso de atenuación en la voz entrante generado cuando el usuario comienza a hablar mientras desde el terminal remoto se le está mandando un mensaje hablado.

CAVIDADES SUPRALARINGEAL: Zona de la laringe por donde pasa la corriente de aire, vibrando o no, según haya sido la actuación de las cuerdas vocales.

DEPENDENCIA O INDEPENDENCIA DEL LOCUTOR: Dependencia del locutor supone incorporar patrones de unidades lingüísticas adaptados a un locutor determinado, independencia del locutor supone patrones validos para cualquier hablante.

DETECTOR DE ACTIVIDAD: Determinación automática de los instantes de comienzo y final de la voz de entrada al sistema.

ENERGÍA: Computa la energía de presión del sonido expresada en dB en un rango de los datos del oscilograma.

ENTRENAMIENTO DISCRIMINATIVO: Proceso de entrenamiento orientado a reducir directamente el número de errores de reconocimiento, en lugar de buscar un modelado estadístico de los datos.

ESPECTRO DEL SONIDO: Diagrama que muestra las amplitudes relativas de la diferentes frecuencias componentes de un sonido. Los espectros pueden ser de línea o continuos, dependiendo del tipo de onda, periódica o aperiódica respectivamente que represente.

ESPECTRO FAST FOURIER TRANSFORM (FFT): Genera un discreto espectro de energía, usando el algoritmo de transformada de Fourier (FFT), tomando los datos desde una región especificad en la fuente del oscilograma.

ESPECTRO DE ENERGÍA PROMEDIO DEL TERMINO LARGO (LTA): Genera un espectro de energía promedio largo, usando el FFT, en los datos del oscilograma en una región especificada.

ESPECTROGRAFÍA (U SONOGRAFÍA): Ver definición de espectrógrafo.

ESPECTRÓGRAFO (U SONÓGRAFO): Instrumento que realiza la descomposición automática de la onda sonora compleja en cada uno de sus componentes y suministrando todos los datos de interés.

FILTRO ACÚSTICO: Resonador con capacidad selectiva respecto a las frecuencias, empleado para transmitir o pasar un sonido.

FONACIÓN: Proceso de producción de una onda sonora mediante la acción de una fuente de sonido. Esta fuente puede ser Glotal, como en el caso de los sonidos sonoros, o de ruido, como sucede en los sordos.

FRAME: Se mide en milisegundos (5, 10, 15, 20, ...mseg), para el análisis automáticamente se convierte al número valores de datos en una muestra dentro de un frame de datos capturados.

FRECUENCIA: Número de ciclos realizados en la unidad de tiempo, convencionalmente el segundo. La unida de frecuencia es el Hertzios (Hz) o ciclos por segundos.

FRECUENCIA EFECTIVA: Ver frecuencia fundamental.

FRECUENCIA FUNDAMENTAL: Es la frecuencia de repetición de la onda compleja y el máximo común divisor de todas sus frecuencias componentes. La frecuencia fundamental o la de cualquier armónico, se mide en Hz. Desde el punto de vista lingüístico, la función contrastiva de la frecuencia fundamental a nivel de la palabra se denomina to. La función de la frecuencia fundamental a nivel de oración es la entonación. En el análisis de una onda compuesta periódica, la frecuencia de cada una de estas ondas sinusoidales integrantes es múltiplo de la frecuencia fundamental (la más baja).

GRAMÁTICA DE RECONOCIMIENTO: Procedimiento destinado a reducir el número de palabras susceptibles de ser reconocidas en cada momento, basado en restricciones sintácticas, semánticas, etc.

HABLA CONECTADA: El usuario pronuncia de forma fluida un mensaje utilizando un vocabulario muy restringido.

HABLA CONTINUA: Pronunciación natural para un vocabulario amplio de palabras.

INTELIGENCIA ARTIFICIAL: Es el campo de la ciencia de la computación dedicado a analizar y desarrollar sistemas que reproduzcan e imiten los procesos de pensamiento y razonamiento del hombre.

INTENSIDAD: Potencia acústica transmitida a través de una onda, medida en una superficie de un cm^2 perpendicular a la dirección de propagación de dicha onda. La intensidad es proporcional al cuadrado de la onda.

INVERSOR DE GIRO: Entendiéndose por este el circuito encargado de hacer girar el motor de C.C en ambos sentidos.

LENGTH (LEN): Representa la duración del frame, informando el número de puntos de datos en la muestra de cada frame.

MEL-CEPSTRUM: Parámetros que representan la envolvente espectral de la voz, utilizando un espaciamiento o resolución en frecuencia semejante a la que utiliza el oído.

MODULACIÓN DE FRECUENCIA (FM): Consiste en variar la frecuencia de la portadora según el valor algebraico de la amplitud de la señal, se aumenta o disminuye proporcionalmente a la variación de la amplitud.

NASAL: Consonante articulada con el paso a la cavidad nasal abierto, merced al descanso de la úvula y una oclusión en alguna zona de la cavidad oral. El aire sale al exterior a través de las fosas nasales.

NASALIZADO (U ORONASAL): Sonido en el que el flujo de aire sale al exterior libremente a través de la cavidad oral y de la cavidad nasal (a la que puede acceder por hallarse la úvula bajada).

ONDA APERIÓDICA: Onda en la que no se repite el mismo perfil de un ciclo a lo largo del tiempo. En las ondas aperiódicas existen componentes de todas las frecuencias.

ONDA COMPLEJA: Onda resultante de la adición de un número determinado de ondas simples.

ONDA PERIÓDICA: Onda que repite el perfil de un ciclo a intervalos regulares de tiempo.

ONDA SONORA: Propagación de una perturbación a través de un medio material como es el aire, en forma de una serie de compresiones y rarefacciones alternas que afectan a cada una de las partículas componentes de ese medio.

ONDA SINUSOIDAL (O SINOSOIDE): Movimiento oscilatorio simple que se corresponde con tonos puros.

OSCILOGRAMA: Diagrama de la señal acústica.

PALABRAS AISLADAS: El usuario pronuncia una sola palabra o comando.

PARAMETRIZACIÓN DE LA VOZ: Extracción de rasgos característicos de voz.

PARÁMETROS DIFERENCIALES (DELTA-ENERGÍA, DELTA-CEPSTRUM, ETC): Contienen información que se relaciona con la pendiente asociada a la evolución de los parámetros estáticos correspondientes (energía, cepstrum).

PERÍODO: Es el tiempo que se requiere en completar un ciclo completo.

PEAK: El máximo de amplitud de los datos del oscilograma, es calculado por cada frame. Este valor caracteriza el comportamiento de la señal como cambio en el tiempo. El máximo de amplitud de onda, es producto de que las partículas del aire se ponen en movimiento por una fuerza externa que actúa sobre ellas alcanzando

un máximo desplazamiento respecto al punto de origen y continua cada partícula ejerciendo de nuevo una fuerza sobre las que se encuentran en su vecindad.

PUERTO PARALELO: Interfaz por medio de la cual los datos entran en forma paralela. En un PC generalmente el dispositivo conectado al puerto paralelo es una impresora.

RECHAZO: Reconocimiento de sonidos indeseados en la voz entrante para evitar su confusión con palabras del vocabulario a reconocer.

RECONOCIMIENTO DE PATRONES: Proceso de clasificación utilizando un conjunto de referencias obtenidas en una fase de entrenamiento del reconocedor.

RECONOCIMIENTO DEL HABLA: Proceso de conversión de un mensaje hablado en texto.

RECONOCIMIENTO EN CONTEXTO O “WORD SPOTTING”: Técnica especialmente utilizada en reconocimiento de palabras aisladas, encaminada a detectar la presencia de palabras del vocabulario a reconocer en el contexto de otras palabras o pronunciaciones.

RECONOCIMIENTO FONÉTICO: Es el basado en el uso de patrones asociados a sonidos elementales del habla.

REDES NEURONALES (NEURALS NETWORKS): Son programas de inteligencia artificial capaz de simular algunas de las funciones del ser humano. Sin reglas convencionales, una red neuronal, obtiene experiencia analizando automática y sistemáticamente una cantidad de datos, para determinar reglas de comportamiento. En base a estas reglas, puede realizar predicciones sobre nuevos casos. Estas técnicas se aplican a problemas de clasificación y series de tiempo. Y ofrecen el potencial de identificar conexiones que otras técnicas no pueden, porque utiliza relaciones lineales y no-lineales entre los datos, puede trabajar con cualquier tipo de distribución (no solamente distribución normal) y maneja datos con redundancia y/o inconsistencia en la información.

RELÉS: Dispositivo consistente en un electroimán que al ser excitado trae una armadura sobre la cual se encuentran montadas dos o más (hasta cientos) láminas aisladas. Actúa como contactos que pueden estar normalmente cerrados o normalmente abiertos.

RESONANCIA: Fenómeno por el cual un cuerpo, denominado resonador y que posee una tendencia natural a vibrar a determinada frecuencia, experimentará vibraciones de mayor amplitud cuando es puesto en movimiento por otro cuerpo vibrante a una frecuencia similar.

RESONADOR: Cualquier cavidad del aparato fonador que entre en vibración al recibir las ondas acústicas producidas por la vibración de las cuerdas vocales, amplificando o reforzando ciertas frecuencias.

RN: Clasificación mediante Redes Neuronales Artificiales.

RUIDO: En fonética acústica, se conoce como ruido la sensación perceptiva originada por ondas sonoras complejas aperiódicas.

TARJETA DE SONIDO: Dispositivo que funciona como una interfaz de entrada de audio al PC.

TASA DE ERROR DE PALABRA: Número de palabras erróneamente reconocidas sobre el número total de palabras ensayadas, expresadas en tantos por cien. Análogamente se habla de tasas de error de frase y fonema.

TIRISTOR: Corresponde a una familia de los transistores de potencia. Está constituido por tres o más uniones del tipo *pnp*. Se utilizan principalmente para rectificar corriente alterna que alimentan los motores de los vehículos de tracción eléctrica.

TONO: Impresión auditiva de la frecuencia fundamental de una onda sonora.

TONO LARINGEO: Vibración producida en la glotis mediante la abertura u cierres sucesivos y continuados de las cuerdas vocales.

TRACTO VOCAL: Conjunto formado por las cavidades infraglólicas, glólicas y supraglólicas.

TRANSISTOR: Dispositivo electrónico de estado sólido, realizado sobre un soporte de material semiconductor. Los transistores pueden realizar todas las funciones de las válvulas electrónicas; pueden por tanto funcionar como amplificadores, detectores, rectificadores, generadores, etc.

TRANSMISOR : Dispositivo que permite efectuar un enlace o comunicación entre dos puntos alejados. En nuestro caso el enlace es por FM.

9.BIBLIOGRAFÍA

[1] ALLEY CHARLES. Ingeniería Eléctrica Asistida por Computador. México, Limusa 1979-1987.

[2] BLATTNER, Meera M. y DANNENBERG, Roger B. Multimedia interface design. New York, Estados Unidos: Addison- Wesley Publishing Company, 1992.

[3] HILERA, José R. y MARTINEZ, Víctor. Redes Neuronales Artificiales: Fundamentos, Modelos y Aplicaciones. Delaware, Estados Unidos: Addison- Wesley.Iberoamericana. 1995.

[3] MANUALES Y CATALOGOS DE LAS HERRAMIENTAS DE MATLAB. USA :Mathworks Inc.1985

[4] MILLMAN, Jacob. Analog and Digital Circuits and Transistors. Tokio: Mc Graw Hill 1972.

[5] NAKAMURA, Shoichiro, Análisis Numérico y Visualización gráfica con Matlab. México : Prentice-Hall, 1997.

[6] OPPENHEIM, Alan y SCHAFER, Ronald. Discrete-Time Signal Processing. New Jersey, Estados Unidos. Prentice hall. 1989.

[7] RABINER, Lawrence R, y SCHAFER, Ronald W. Digital Processing of Speech Signals.

[8] RABINER, Lawrence y JUANG Biing-Hwang. Fundamental of speech Recognition. USA: Prentice Hall, 1993.

[9] TOMASI, Wayne. Sistemas de Comunicaciones Electrónicas. México: Prentice Hall Hispanoamericana,S.A. 1996.

ANEXO A.

ESTRUCTURA DEL RECONOCEDOR

Para activar el programa es necesario haber instalado previamente la plataforma MATLAB 5.2, adicionándole los “toolboxes” de Señales (Signals Processing Toolbox) y Redes neuronales artificiales (Neural Networks Toolbox).

El control de las diversas partes del reconocedor se realiza a través de una interfaz gráfica para usuario, la cual fue mencionada en el manual del usuario del cual se ha reseñado con anterioridad.

El programa se subdivide en 2 secciones, 3 etapas, 5 fases.

Las secciones son enumeradas a continuación:

- **Sección 1:** Entrenamiento del sistema.
- **Sección 2:** Prueba del sistema.

Cada sección se subdivide en 3 etapas, denominadas así:

- **Etapa 1** : Preprocesamiento de la voz.
- **Etapa 2** : Reconocimiento de los parámetros de la voz.
- **Etapa 3** : Postprocesamiento de los resultados de la red neuronal.

A su vez se tienen 5 fases distribuidas entre las 3 etapas enunciadas anteriormente, cuyos nombres son:

Fase 1. Almacenamiento de la voz.

Fase 2. Edición y adecuación de la voz.

Fase 3. Extracción de características de la voz.

Fase 4. Reconocimiento de la voz.

Fase 5. Salida del reconocedor.

El diagrama de bloque de la estructura del reconocedor puede verlo a continuación.

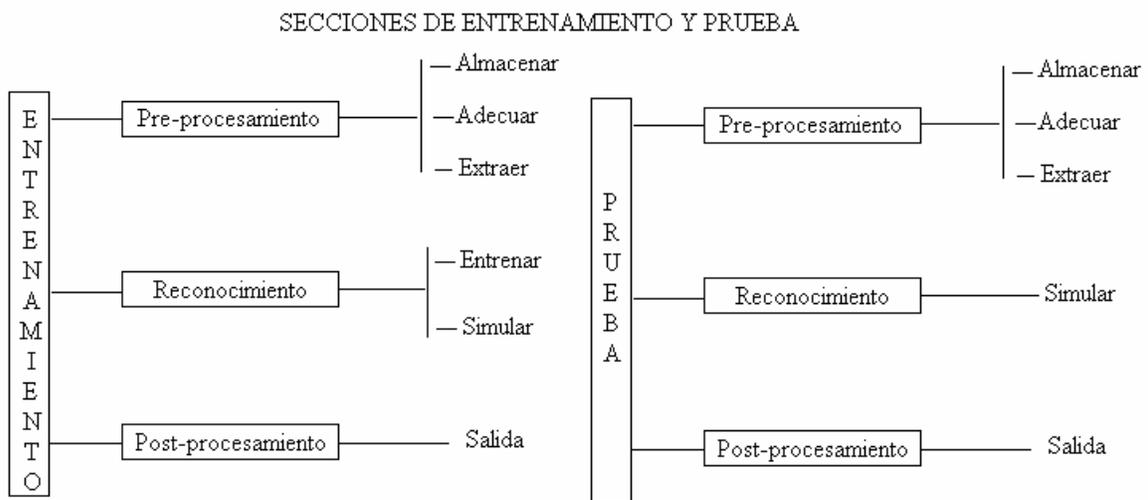


Figura 55. diagrama de bloque del programa reconocedor de voz

SECCION 1:ENTRENAMIENTO.

Los principales archivos que usamos en el reconocedor serán consignados a continuación dando una breve información de cual es su función.

Almacenamiento

Grabar.m : se utiliza para hacer un llamado a la grabadora de la voz la cual ha sido diseñada delphi5.0.

Grabar.cfg: subrutina utilizada para poder llamar al archivo de la grabadora en ejecutable desde la plataforma MATLAB5.2

Recorder.exe: es el archivo que contiene la grabadora con la especificaciones necesarias para el proceso de almacenamiento de la voz

Entrenamiento.m: se utiliza para ingresar al proceso de entrenamiento la cual tiene los tres de reconocimiento.

Voces1.m : es el encargado de administrar la captación de las voces iniciales en el proceso de almacenamiento.

Voces2.m : es el encargado de administrar la captación de las voces de validación en el proceso de almacenamiento.

Edición.m: Edición de las palabras del conjunto de entrenamiento y validación para discriminar los silencios iniciales y finales de las palabras.

Editor1.m: Detecta el principio y final de las palabras del conjunto de las iniciales.

Editor2.m: Detecta el principio y final de las palabras del conjunto de validación.

Discriminador.m: Calcula los principios y finales de las palabras basadas en el estudio de energía y cruces por cero de las tramas de una señal de voz.

Adecuación.

Lectura.m: este archivo es el encargado de llamar los archivos de lectura y extracción de características.

leav1editadas.m. Es el archivo encargado de hacer el proceso de lectura de las voces iniciales que han sido editadas en la etapa previa, también realiza un proceso de preénfasis y eliminación de ceros en el archivo de voz.

Leav2editadas.m: es el archivo encargado de hacer el proceso de lectura de las voces de validación que han sido editadas en la etapa previa. también realiza un proceso de preénfasis y eliminación de ceros en el archivo de voz

preénfasis.m: se utiliza para suavizar la señal en el proceso de extracción de características, es un filtro digital.

Características1.m: Ejecuta la extracción de características de la palabras del conjunto de voces iniciales.

Características2.m: Ejecuta la extracción de características de la palabras del conjunto de voces validación.

Extracción.m: llama al Melcepstral que es nuestro método extracción de características.

Melcepstral.m: es el archivo donde se encuentra aplicado la función melcepst, que es propia de los toolbox de Matlab5.2 y que determina los coeficientes melcepst.

Funresamplex.m: Remuestrea una palabra discretizada para estandarizar la longitud de las palabras.

Resamplex.m: Estandariza las longitudes de las palabras remuestreando la señal digitalizada mediante las operaciones de interpolación.

Simulación y salida.

RNA.m: Captura la información del número de neuronas de la capa oculta para la sub-fase de entrenamiento y la sub-fase de validación.

Neuronal.m: Realiza la sub-fase de entrenamiento y la sub-fase de validación, Realiza la etapa de postprocesamiento (Fase de Salida), es el programa estructural del reconocedor de voz y donde se encuentra la fase de resultados matlab5.2.

Normalizar.m: Calcula el patrón de normalización para cada característica del vector de características.

SECCION 2: PRUEBA.

Almacenamiento.

Reconocer.m: llama las secuencias de grabación, detección y método que se le va a aplicar para detectar la palabra reconocida.

Grabar1.m : se utiliza para hacer un llamado a la grabadora de la voz la cual ha sido diseñada delphi5.0.

Adecuación

Leaprueba.m: Es el archivo encargado de hacer el proceso de lectura de las voz de prueba que han sido editadas en la etapa previa, también realiza un proceso de preénfasis y eliminación de ceros en el archivo de voz.

Extracción

Caracteristicasprueba.m.: Ejecuta la extracción de características de la palabras del conjunto de voces iniciales.

Simulación.

rnaprueba.m: Realiza fase de validación, Realiza la etapa de postprocesamiento(Fase de Salida) ,es el programa estructural del reconocedor de voz en la fase de prueba.

Salida

metododistancia.m: este archivo contiene el programa que se usa para averiguar cual palabra es la reconocida y poder darle una asignación como salida al usuario.

ANEXO B

LISTADOS DE LOS PRINCIPALES PROGRAMAS (MATLAB 5.2).

Entrenamiento.m

```
close;
clear all;
clc;
global h1
h1=figure('Position',[0, 0, 800, 560],...
    'NumberTitle','off', ...
    'Name','MOTOR DC',...
    'Resize','off',...
    'Menubar','None');

f1=uimenu(h1, 'Label','Grabacion de Sonidos', ...
    'Position', 1);
f2=uimenu(h1, 'Label','Entrenamiento del Sistema', ...
    'Position', 2);
f3=uimenu(h1, 'Label','RETORNAR',...
    'Position', 3,'Callback','RETORNAR');

% Grabacion de datos

f1o1=uimenu(f1,'Label','&voces iniciales',...
    'Callback','Voces1');

f1o2=uimenu(f1,'Label','&voces de validacion',...
    'Callback','Voces2');

f1o3=uimenu(f1,'Label','&Edicion de voces',...
    'Callback','Edicion(''1'')');

% Entrenamiento del sistema

f2o1=uimenu(f2,'Label','&Procesamiento',...
    'Callback','lectura(''1'')');

f2o2=uimenu(f2,'Label','&Reconocimiento','Callback',['algor=3;', 'save
c:\matlab\recvoz\algoritmo algor;', 'mf2o2']);
```

voces1.m

```
figure(2);

set(gcf,'Position',[30 40 500 450],...
    'NumberTitle','off', ...
    'Name','GRABACION DE SONIDOS - RECVOZ1.0',...
    'Resize','on',...
    'Menubar','None');

uicontrol(gcf,'Style','text','Position',[30 400 450 20],...
    'BackgroundColor','b',...
    'HorizontalAlignment','center',...
    'ForegroundColor','w',...
    'String','INSERTE LAS PALABRAS DE PRUEBA SEGUN
INDICACIONES');
%-----
%ENCENDER
%-----

uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String',' <ENCENDER >', ...
    'Position',[20 350 100 30],...
    'Callback','ENCENDERgrabar')

%-----
%APAGAR
%-----

uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String',' <APAGAR >', ...
    'Position',[125 350 100 30],...
    'Callback','APAGARgrabar')

%-----
%IZQUIERDA
%-----

uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String',' <IZQUIERDA >', ...
    'Position',[230 350 120 30],...
    'Callback','IZQUIERDAgrabar')

%-----
%DERECHA
%-----

uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String',' <DERECHA >', ...
    'Position',[355 350 100 30],...
    'Callback','DERECHAgrabar')

%-----
% SALIR
%-----

uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String','SALIR GRABADOR ', ...
    'Position',[180 10 150 40],...
    'Callback','close')
```

voces2.m

```
figure(2);
set(gcf,'Position',[30 40 500 450],...
    'NumberTitle','off', ...
    'Name','PRUEBA DE SONIDOS',...
    'Resize','on',...
    'Menubar','None');
uicontrol(gcf,'Style','text','Position',[30 400 450 20],...
    'BackgroundColor','m',...
    'HorizontalAlignment','center',...
    'ForegroundColor','w',...
    'String','INSERTE LAS PALABRAS DE PRUEBA SEGUN
INDICACIONES');

%-----
%ARRANCAR
%-----

testdat='vocesvalidacion'; %Directorio para grabar voces de entrenamiento
uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String',' <ENCENDER >', ...
    'Position',[20 350 100 30],...
    'Callback','ENCENDERprobar')

%-----
%DETENER
%-----
uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String',' <APAGAR >', ...
    'Position',[125 350 100 30],...
    'Callback','APAGARprobar')

%-----
%IZQUIERDA
%-----
uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String',' <IZQUIERDA >', ...
    'Position',[230 350 120 30],...
    'Callback','IZQUIERDAprobar')

%-----
%DERECHA
%-----
uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String',' <DERECHA >', ...
    'Position',[355 350 100 30],...
    'Callback','DERECHAprobar')

%-----
% SALIR
%-----

uicontrol('Style','PushButton','HorizontalAlign','center', ...
    'String','SALIR GRABADOR ', ...
    'Position',[180 10 150 40],...
```

```
'Callback','close')
```

ENCENDERprobrar.m

```
%-----  
%ENCENDER  
%-----  
traindat='vocesiniciales'; %Directorio para grabar voces de entrenamiento  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 1>', ...  
          'Position',[20 350 100 30],...  
          'Callback','grabar(''palabral.wav'',traindat)')  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 2>', ...  
          'Position',[20 320 100 30],...  
          'Callback','grabar(''palabra2.wav'',traindat)')  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 3>', ...  
          'Position',[20 290 100 30],...  
          'Callback','grabar(''palabra3.wav'',traindat)')  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 4>', ...  
          'Position',[20 260 100 30],...  
          'Callback','grabar(''palabra4.wav'',traindat)')  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 5>', ...  
          'Position',[20 230 100 30],...  
          'Callback','grabar(''palabra5.wav'',traindat)')  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 6>', ...  
          'Position',[20 200 100 30],...  
          'Callback','grabar(''palabra6.wav'',traindat)')  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 7>', ...  
          'Position',[20 170 100 30],...  
          'Callback','grabar(''palabra7.wav'',traindat)')  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 8>', ...  
          'Position',[20 140 100 30],...  
          'Callback','grabar(''palabra8.wav'',traindat)')  
  
uicontrol('Style','PushButton','HorizontalAlignment','center', ...  
          'String',' <ENCENDER 9>', ...  
          'Position',[20 110 100 30],...  
          'Callback','grabar(''palabra9.wav'',traindat)')
```

```

uicontrol('Style', 'PushButton', 'HorizontalAlignment','center', ...
    'String', ' <ENCENDER 10>', ...
    'Position', [20 80 100 30],...
    'Callback', 'grabar(''palabra10.wav'',traindat)')

```

edición.m

```

function Edicion(xwyz)
h1=1
uicontrol(h1,'Style','text','Position',[330 300 150 120],...
    'HorizontalAlignment','center',...
    'ForegroundColor','b',...
    'String','INICIANDO EDICIÓN DE VOCES')
pause(1)
editor1(1);
uicontrol(h1,'Style','text','Position',[330 340 150 40],...
    'HorizontalAlignment','center',...
    'ForegroundColor','b',...
    'String','Finalizó la Edición de voces de entrenamiento')
pause(1)
editor2(1); uicontrol(h1,'Style','text','Position',[330 310 150 40],...
    'HorizontalAlignment','center',...
    'ForegroundColor','b',...
    'String','Finalizó la Edición de voces de validación')

```

editor1.m

```

function editor1(x)
Npal=4;
Nej=10
filas=Npal*Nej;

numpalabra=1:1:filas;
for n=1:filas,
    cd c:\matlab\recvoz\patrones\vocesiniciales
    [b,fs,nbits]=wavread(strcat('palabra',num2str(numpalabra(n))));
    [principio,final]=discriminador(b,fs);

    cb=b(principio:final);

    cd c:\matlab\recvoz\patrones\Voces1editadas
    wavwrite(cb,fs,nbits,strcat('palabra',num2str(numpalabra(n))));
end

```

discriminador.m

```

principio=0;
final=0;

```

```

switch fs
    case 11025,
        tope_i=34;
        tope_d=79;
    case 16000,
        tope_i=50;
        tope_d=113;
    case 22050,
        tope_i=68;
        tope_d=145;
    otherwise,
        sprintf('Error en tasa de muestreo.discriminador')
end

lpal=length(b);
sig=enframe(b,256,128);

for i=1:tope_i-1
    ener(i)=(sum((sig(i,:)).^2));
    crossp(i)=cruce(sig(i,:));
end

en=max(ener);
crn=max(crossp);

s_tam=size(sig);

det=0;
for i=tope_i:s_tam(1)-5,
    sener=(sum((sig(i,:)).^2));
    sener2=(sum((sig(i+5,:)).^2));
    crsig=cruce(sig(i,:));
    crsig2=cruce(sig(i+5,:));
    if ((sener>en) & (crsig>crn)) & ((sener2>en) & (crsig2>crn))
        det=i;
        break;
    end
end

fdet=0;
for i=s_tam(1):-1:tope_d
    sener=(sum((sig(i,:)).^2));
    sener2=(sum((sig(i-5,:)).^2));
    crsig=cruce(sig(i,:));
    crsig2=cruce(sig(i-5,:));
    if ((sener>en) & (crsig>crn)) & ((sener2>en) & (crsig2>crn))
        fdet=i;
        break;
    end
end
end

```

```

principio=(det+1)*256/2;
final=(fdet+1)*256/2;
falla=final-principio;
palfal=3500;
switch fs,
    case 11025,
        palpro=4352;
    case 16000,
        palpro=6400;
    case 22050,
        palpro=8704;
end

if (falla<0 | falla<palfal),
    final=lpal;
    principio=palpro;
end

```

enframe.m

```

function f=enframe(x,win,inc)
nx=length(x);
nwin=length(win);
if (nwin == 1)
    len = win;
else
    len = nwin;
end
if (nargin < 3)
    inc = len;
end
nf = fix((nx-len+inc)/inc);
f=zeros(nf,len);
indf= inc*(0:(nf-1)).';
inds = (1:len);
f(:) = x(indf(:,ones(1,len))+inds(ones(nf,1),:)));
if (nwin > 1)
    w = win(:)';
    f = f .* w(ones(nf,1),:);
end

```

cruce.m

```

function [crux]=cruce(raw)

tam =length(raw);
cruces=raw(2:tam).*raw(1:tam-1);
enc=find(cruces<0);

```

```
crux=length(enc);
```

lectura.m

```
function lectura(x)

%LECTURA DE ARCHIVOS
clear;
hl=1
uicontrol(hl,'Style','text','Position',[330 300 150 120],...
          'HorizontalAlignment','center',...
          'ForegroundColor','b',...
          'String','INICIANDO LECTURA Y EXTRACCIÓN DE
CARACTERÍSTICAS')
pause(1)
leavleditadas(1);

leav2editadas(1);
uicontrol(hl,'Style','text','Position',[330 300 150 120],...
          'HorizontalAlignment','center',...
          'ForegroundColor','b',...
          'String','FINALIZÓ LECTURA')

pause(1)
pp='1';
extraccion;
```

leavleditadas.m

```
function leavleditadas(x)

%ENTRADA
cd c:\Matlab\recvoz\patrones\vocesleditadas;

limM=25000;
Npal=4;
Nej=10;
filas=Npal*Nej;

%LECTURA

M=[];
M=zeros(filas,limM);
numpalabra=1:1:filas;
for n=1:filas,
    y=wavread(strcat('palabra',num2str(numpalabra(n))));
    cerosy=find(y==0);
    y(cerosy)=[];
    y =preenfasis(y);
```

```

        Z(n)=length(y);
        M(n,1:Z(n))=y';
    end

%IGUALACION

mz=ceil(mean(Z));
for n=1:filas,
    N(n,:)=resamplex(M(n,1:Z(n)),mz);
end

%PRODUCTO DE SALIDA

save c:\matlab\recvoz\word-E    N
save c:\matlab\recvoz\maximos  mz
[voz,tm,nbits]=wavread('palabra1');
save c:\matlab\recvoz\frecuencia  tm
save c:\matlab\recvoz\dimensiones Npal Nej

```

extracción.m

```

clc;
sprintf('Inicio la Extracción de Características')
pp=str2num(pp);
save c:\matlab\recvoz\ppvoz  pp

caracteristicas1(1);
caracteristicas2(1);
clear;
h1=1
uicontrol(h1,'Style','text','Position',[330 340 150 40],...
          'HorizontalAlignment','center',...
          'ForegroundColor','b',...
          'String','Finalizó la extracción de características')

uicontrol(h1,'Style','text','Position',[330 310 150 40],...
          'HorizontalAlignment','center',...
          'ForegroundColor','b',...
          'String','Proceder el algoritmo de aprendizaje')

```

preénfasis.m

```

function [preenfasis]=pp_preenfasis(s)

w=length(s);
preenfasis=s(2:w)-0.95*s(1:w-1);

```

caracteristicas1.m

```
function caracteristicas1(pp)

%ENTRADAS
load c:\matlab\recvoz\word-E;
load c:\matlab\recvoz\frecuencia;

Npal=4;
Nej=10;
filas=Npal*Nej;

for n=1:filas
    E(n,:)=caracter(N(n,:),tm,pp);
end
Q=E';

n=0;
M1=[];
for i=1:Npal,
    for j=1:Nej,
        n=n+1;
        M1(i,j)=n;
    end
end
Mpal=M1(:)';
P=Q(:,Mpal);

%SALIDA
D=size(P);
fid = fopen('c:\matlab\recvoz\train.dat','wb');
fwrite(fid,D,'float')
fwrite(fid,P,'float')
fclose(fid)
```

carácter.m

```
function [ww]=caracter(y,tm,pp)

ww=[];    % Vector de características
k2=1;    % Contador del vector de características
ncmel=12; % Numero de coeficientes mel-cepstral

                % METODO DE EXTRACCION: Melcepstral
pp==1

[ww,k2]=melcepstral(y,tm,ncmel,ww,k2);
```

Melcepstral.m

```
function[ww,k2]=melcepstral(y,tm,ncmel,ww,k2)

    y1=melcepst(y,tm,'Mt',ncmel);
    yx=y1';
    melc=yx(:)';
    ww=[ww,melc];
    lonmel=length(melc);
    k2=k2+lonmel;
```

melcepst.m

```
function c=melcepst(s,fs,w,nc,p,n,inc,fl,fh)

if nargin<2 fs=11025; end
if nargin<3 w='M'; end
if nargin<4 nc=12; end
if nargin<5 p=floor(3*log(fs)); end
if nargin<6 n=pow2(floor(log2(0.03*fs))); end
if nargin<9
    fh=0.5;
    if nargin<8
        fl=0;
        if nargin<7
            inc=floor(n/2);
        end
    end
end
end

if any(w=='R')
    z=enframe(s,n,inc);
elseif any (w=='N')
    z=enframe(s,hanning(n),inc);
else
    z=enframe(s,hamming(n),inc);
end
f=rfft(z. ');
[m,a,b]=melbankm(p,n,fs,fl,fh,w);
pw=f(a:b,:).*conj(f(a:b,:));
pth=max(pw(:))*1E-6;
if any(w=='p')
    y=log(max(m*pw,pth));
else
    ath=sqrt(pth);
    y=log(max(m*abs(f(a:b,:)),ath));
end
c=rdct(y).';
if ~any(w=='0')
    c(:,1)=[];
end
```

```

if any(w=='e')
    c=[log(sum(pw)).' c];
end
nf=size(c,1);
if p>nc
    c(:,nc+1:size(c,2))=[];
elseif p<nc
    c=[c zeros(nf,nc-p)];
end

% calculate derivative

if any(w=='D')
    vf=(4:-1:-4)/60;
    af=(1:-1:-1)/2;
    ww=ones(5,1);
    cx=[c(ww,:); c; c(nf*ww,:)];
    vx=reshape(filter(vf,1,cx(:)),nf+10,nc);
    vx(1:8,:)=[];
    ax=reshape(filter(af,1,vx(:)),nf+2,nc);
    ax(1:2,:)=[];
    vx([1 nf+2],:)=[];
    if any(w=='d')
        c=[c vx ax];
    else
        c=[c ax];
    end
elseif any(w=='d')
    vf=(4:-1:-4)/60;
    ww=ones(4,1);
    cx=[c(ww,:); c; c(nf*ww,:)];
    vx=reshape(filter(vf,1,cx(:)),nf+8,nc);
    vx(1:8,:)=[];
    c=[c vx];
end

if nargout<1
    [nf,nc]=size(c);
    t=((0:nf-1)*inc+(n-1)/2)/fs;
    ci=(1:nc)-any(w=='0')-any(w=='e');
    imh = imagesc(t,ci,c. ');
    axis('xy');
    xlabel('Time (s)');
    ylabel('Mel-cepstrum coefficient');
    map = (0:63)'/63;
    colormap([map map map]);
    colorbar;
end

```

RNA.m

```
clc;
diary c:\matlab\recvoz\salida1.txt on %Bitacora de la sesión de MATLAB.
h2=figure(2);

set(h2,'Position',[180,30, 350, 370],...
    'Color',[0,0.5, 0.5],...
    'NumberTitle','off',...
    'resize','off',...
    'selected','on',...
    'WindowStyle','modal',...
    'Name','Entrenando la Red Neuronal:');

ed0 = uicontrol(h2, 'Style','text', ...
    'Position', [70,160, 110,20],...
    'String',...
    ' Neuronas ocultas : ');
%'Position', [100,120, 110,40],...
ed2 = uicontrol(h2, 'Style','edit', ...
    'Position', [180,160, 58,20],...
    'String','3',...
    'CallBack',['inp_txt=get(ed2, 'String');',...
    'clf;neuronal(str2num(inp_txt))']);
```

neuronal.m

```
function neuronal(neuro)

nntwarn off

%ENTRADAS

S1=neuro;
load c:\matlab\recvoz\dimensiones
load c:\matlab\recvoz\algoritmo
aprendizaje=algor;

fid = fopen('c:\matlab\recvoz\train.dat','r');
[D,count] = fread(fid,[1,2],'float');
[P,count] = fread(fid,[D(1),D(2)],'float');
fclose(fid);

%NORMALIZACION

load c:\matlab\recvoz\ppvoz
norma=max(max(abs(P)));
normac=ceil(norma);
divi=normac+1;
P=P./divi;
```

```

minmaxP=[];
lonP=size(P);
for fp=1:lonP(1),
    minP=min(P(fp,:));
    maxP=max(P(fp,:));
    if minP==maxP,
        maxP=minP+0.00001;
    end
    otromm=[minP,maxP];
    minmaxP=[minmaxP;otromm];
end

% VECTORES OBJETIVO
T=[];
for n=1:Nej,
    T2=eye(Npal);
    T=[T,T2];
end

%INICIALIZACION DE PESOS
[w1,b1,w2,b2] = initff(minmaxP,S1,'tansig',4,'logsig');

%=====
%ENTRENAMIENTO
%PARAMETROS
df = 100;
me = 800;
eg = 0.2 ;
lr = 0.01;
momentum=0.95;
err_ratio= 1.04;

%=====

tp = [df me eg lr lri lrd momentum err_ratio];
[w1,b1,w2,b2,ep,tr] = trainbpx(w1,b1,'tansig',w2,b2,'logsig',P,T,tp);

save c:\matlab\recvoz\pesos w1 b1 w2 b2 divi

fid = fopen('c:\matlab\recvoz\test.dat','r');
[D2,count] = fread(fid,[1,2],'float');
[P2,count] = fread(fid,[D2(1),D2(2)],'float');
fclose(fid);

%NORMALIZACION
tamP2=size(P2);
P2=P2./divi;

%Inicializacion de variables de aciertos
m=Npal*Nej;

```

```

cor=0;
enc=0;
apa=0;
izq=0;
der=0;

% SIMULACION

a=simuff(P2,w1,b1,'tansig',w2,b2,'logsig');
b=a;
save c:\matlab\recvoz\salidas b

d=0;
des=0;
for i=1:m
    aa=sort(a(:,i));
    tam=size(aa);
    if aa(tam(1))~=aa(tam(1)-1),
        [v,p]=max(a(:,i));
        a(:,i)=zeros;
        a(p,i)=1;
        if a(:,i)==T(:,i),
            cor=cor+1;
            indice=0;
            indice=find(a(:,i)==1);
            switch indice
                case 1,
                    enc=enc+1;
                case 2,
                    apa=apa+1;
                case 3,
                    izq=izq+1;
                case 4,
                    der=der+1;
            end
        else
            d=d+1;
            des(d)=i;
        end
    end
end;

close
h2=figure(2);

set(h2,'Position',[180,30, 350, 370],...
'Color',[0,0.5, 0.5],...
'NumberTitle','off',...
'resize','off',...
'selected','on',...
'WindowStyle','modal',...
'Name','Resultados de la Red Neuronal:');

uicontrol(h2, 'Style','text', ...
'Position', [70,250, 150,20],...
'String',...

```

```

        ' total aciertos : ');

uicontrol(h2, 'Style','edit', ...
    'Position', [220,250, 58,20],...
    'String',cor)

uicontrol(h2, 'Style','text', ...
    'Position', [70,220, 150,20],...
    'String',...
    ' Porcentaje de aciertos : ');

uicontrol(h2, 'Style','edit', ...
    'Position', [220,220, 58,20],...
    'String',(cor/m)*100)

uicontrol(h2, 'Style','text', ...
    'Position', [70,190, 150,20],...
    'String',...
    'Numero de iteracciones');

uicontrol(h2, 'Style','edit', ...
    'Position', [220,190, 58,20],...
    'String',ep)

uicontrol(h2, 'Style','text', ...
    'Position', [70,160, 150,20],...
    'String',...
    'Aciertos encender');

uicontrol(h2, 'Style','edit', ...
    'Position', [220,160, 58,20],...
    'String',enc)

uicontrol(h2, 'Style','text', ...
    'Position', [70,130, 150,20],...
    'String',...
    'Aciertos apagar');

uicontrol(h2, 'Style','edit', ...
    'Position', [220,130, 58,20],...
    'String',apa)

uicontrol(h2, 'Style','text', ...
    'Position', [70,100, 150,20],...
    'String',...
    'Aciertos izquierda');

uicontrol(h2, 'Style','edit', ...
    'Position', [220,100, 58,20],...
    'String',izq)

uicontrol(h2, 'Style','text', ...
    'Position', [70,70, 150,20],...
    'String',...
    'Aciertos derecha');

```

```
uicontrol(h2, 'Style','edit', ...
    'Position', [220,70, 58,20],...
    'String',der)
```

diary off

reconocer.m

```
function reconocer(wxyz)
grabarl('onesignal.wav','signal');
detector(1);
leaprueba(1);
caracteristicasprueba(1);
rnaprueba(1);
metododistancia(1);
```

detector.m

```
function detector(x)
cd c:\matlab\recvoz\patrones\signal
[b,fs,nbits] = wavread('onesignal');
[principio,final]=discriminador(b,fs);
cb=b(principio:final);
cd c:\matlab\recvoz\patrones\signal
wavwrite(cb,fs,nbits,'onesignal-C');
clc;
```

leaprueba.m

```
function leaprueba(x)
%ENTRADA
cd c:\Matlab\recvoz\patrones\signal;
load c:\matlab\recvoz\maximos;
```

```

y7 = wavread('onesignal-C');
cerosy7=find(y7==0);
y7(cerosy7)=[];
y7 = preenfasis(y7);
p = length(y7);

V_one=resamplex(y7',mz);
save c:\matlab\recvoz\oneword V_one

```

caracteristicasprueba.m

```

function caracteristicasprueba(x)

clc;

%ENTRADA
load c:\matlab\recvoz\oneword
load c:\matlab\recvoz\frecuencia
load c:\matlab\recvoz\ppvoz

y=V_one;
ww=[];
[ww]=caracter(y,tm,pp);
P3=ww';

save c:\matlab\recvoz\onetest P3

```

rnaprueba.

```

function rnaprueba(x)

clc;
nntwarn off

%ENTRADA
load c:\matlab\recvoz\onetest
load c:\matlab\recvoz\pesos
load c:\matlab\recvoz\ppvoz;

%NORMALIZACION
tamP3=size(P3);
P3=P3./divi;

%SIMULACION
a=[];
a=simuff(P3,w1,b1,'tansig',w2,b2,'logsig');
b_one=a;

```

```
%SALIDA
save c:\matlab\recvoz\out b_one
```

metododistancia.m

```
function [producto]=metododistancia(x)

%ENTRADAS
load c:\matlab\recvoz\out
producto=[];
a=b_one;

%SELECCION
[v1,p1]=max(a);
b=a;
a(p1)=-9999;
[v2,p2]=max(a);

if (v1-v2)>0.15,
    indice = find(b==b(p1));
    switch indice
    case 1,
        producto=['ENCENDER'];
        encender
    case 2,
        producto=['APAGAR'];
        apagar
    case 3,
        producto=['IZQUIERDA'];
        izquierda
    case 4,
        producto=['DERECHA'];
        derecha
    end

else
    producto=['?'];
    error(dlg('Palabra no Reconocida'));
end
```

resamplex.m

```
function [Y]=resamplex(vox,N)

y1=vox;
L=length(y1);
if L==N,
    Y=y1;
```

```

    return
end

if L>N,
    Y=y1;
    I=L/(L-N);
    E=round([I:I:L]);
    Y(E)=[];
    L2=length(Y);
    if L2<N,
        Y=[Y,Y(L2)];
    elseif L2>N,
        Y(L2)=[];
    end
end
end
if L<N,
    Nx=2*L-1;
    if N<=Nx,
        I=L/(N-L+1);
        [Y]=funresamplex(y1,N,I,L);
    else
        Lx=floor((N+1)/2);
        yx=y1;
        L1=L;
        while L1<N,
            if L1<=Lx,
                Nc=2*L1-1;
            else
                Nc=N;
            end
            I1=L1/(Nc-L1+1);
            [Y]=funresamplex(yx,Nc,I1,L1);
            yx=Y;
            L1=length(yx);
        end
    end
end
end
end

```

ANEXO C.

PROGRAMA DEL GRABADOR DE VOZ.

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  ExtCtrls, MPlayer, StdCtrls, Buttons;

type
  TForm1 = class(TForm)
    MediaPlayer1: TMediaPlayer;
    Timer1: TTimer;
    Panel2: TPanel;
    Panel5: TPanel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton8: TSpeedButton;
    SaveDialog1: TSaveDialog;
    Timer2: TTimer;
    Panel1: TPanel;
    SpeedButton3: TSpeedButton;
    RadioGroup1: TRadioGroup;

    procedure Timer1Timer(Sender: TObject);
    procedure SpeedButton8Click(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure Timer2Timer(Sender: TObject);
    procedure RadioGroup1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}
var
```

```

fast:integer;
  a,b,c,e,f,h:integer;
  g:string;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  asm
MOV DX,379h ;
in al,dx;
mov fast,al
end;

  if (h=1) and (e=1) then form1.close;

  if (fast=135) and (a<>1)then
begin
  mediaplayer1.wait:=true;
  if f<>2 then
  mediaplayer1.filename:='c:\recorder\recorder final\sail16.wav';
mediaplayer1.open;
mediaplayer1.startrecording;
mediaplayer1.wait:=true;
a:=1;
e:=2;
c:=0;
h:=1;

  end;

if (fast=143) and(a=1) then
begin
mediaplayer1.filename:='c:\matlab\recvoz\patrones\signal\onesignal.wav';
  if b=3 then mediaplayer1.filename:=g;
mediaplayer1.save;
a:=2 ;
e:=1 ;
b:=0;
mediaplayer1.close;
  end;
end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
if savedialog1.execute then
begin
g:=savedialog1.filename;
savedialog1.filename:='';
b:=3;
F:=5;

end;
end;

```

```

procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
timer1.enabled:=false ;
mediaplayer1.wait:=true;
if f<>2 then
mediaplayer1.filename:='c:\recorder\recorder final\sail16.wav';
mediaplayer1.open;
mediaplayer1.startrecording;
mediaplayer1.wait:=true;
a:=1;
e:=2;
c:=0;
end;

procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
if e=2 then
begin
mediaplayer1.filename:='c:\matlab\recvoz\patrones\signal\onesignal.wav';
if b=3 then mediaplayer1.filename:=g;
mediaplayer1.save;
a:=1 ;
mediaplayer1.close;
timer1.enabled:=false;
e:=1;
c:=0;
b:=0
end;
end;

procedure TForm1.Timer2Timer(Sender: TObject);
begin
if (e<>1) and(e=2)then
begin
c:=15+c;
speedbutton3.left:=(c);
if c>86 then c:=-15;
end;
if e=1 then
begin
speedbutton3.left:=(0);
end;
end;
procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
if radiogroup1.ItemIndex=0 then
mediaplayer1.filename:='c:\recorder\recorder final\sail16.wav');
if radiogroup1.ItemIndex=1 then
mediaplayer1.filename:='c:\recorder\recorder final\sail22.wav');
if radiogroup1.ItemIndex=2 then
mediaplayer1.filename:='c:\recorder\recorder final\sail44.wav');
if radiogroup1.ItemIndex=3 then
mediaplayer1.filename:='c:\recorder\recorder final\sail96.wav');
f:=2
end;
end.

```

ANEXO D.

PROGRAMAS DE SALIDAS AL PUERTO PARALELO.

ENCENDER

```
unit start;
interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  ExtCtrls;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Panel1: TPanel;
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;

implementation

{$R *.DFM}
var
  a:integer;
procedure TForm1.FormCreate(Sender: TObject);
begin
  asm
MOV DX,378h ;
mov al,1;
out dx,al;
mov a,al
  end;
  timer1.enabled:=true;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  form1.close;
end;
end.
```

APAGAR

```
unit stop;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  ExtCtrls;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Panell: TPanel;
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}
var
  a:integer;
procedure TForm1.FormCreate(Sender: TObject);
begin
  asm
  MOV DX,378h ;
  mov al,0;
  out dx,al;
  mov a,al
  end;
  timer1.enabled:=true;

end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  form1.close;
end;

end.
```

DERECHA

```
unit Right;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  ExtCtrls;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Panell: TPanel;
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}
var
  a:integer;
procedure TForm1.FormCreate(Sender: TObject);
begin
  asm
  MOV DX,378h ;
  mov al,2;
  out dx,al;
  mov a,al
  end;
  timer1.enabled:=true;

end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  form1.close;
end;

end.
```

IZQUIERDA

```
unit left;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  Dialogs,
  ExtCtrls;

type
  TForm1 = class(TForm)
    Timer1: TTimer;
    Panel1: TPanel;
    procedure FormCreate(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}
var
  a:integer;
procedure TForm1.FormCreate(Sender: TObject);
begin
  asm
  MOV DX,378h ;
  mov al,2;
  out dx,al;
  mov a,al
  end;
  timer1.enabled:=true;

end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  form1.close;
end;

end.
```