SISTEMA DE RECONOCIMIENTO DE COMANDOS DE VOZ COMO INTERFAZ DE UNA HABITACIÓN CON TRES ATMÓSFERAS LUMÍNICAS

Proyecto de Grado

David Barragán Bermúdez

UNIVERSIDAD SANTO TOMAS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA

SISTEMA DE RECONOCIMIENTO DE COMANDOS DE VOZ COMO INTERFAZ DE UNA HABITACIÓN CON TRES ATMÓSFERAS LUMÍNICAS

David Barragán Bermúdez

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

Director
Ing. Carlos Enrique Montenegro Narváez

Codirector
Ing. Carlos Javier Mojica Casallas

Facultad de Ingeniería Electrónica

División de Ingenierías

Universidad Santo Tomás

Bogotá, D.C.

2022

Autoridades de la Universidad

RECTOR GENERAL

R.P. FRAY JOSE GABRIEL MESA ANGULO, O.P.

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL

R.P. FRAY LUIS FRANCISCO SASTOQUE POVEDA, O.P.

VICERRECTOR ACADEMICO GENERAL

R.P. FRAY EDUARDO GONZALEZ GIL, O.P.

SECRETARIO GENERAL

Dra. INGRID LORENA CAMPOS VARGAS

DECANO DIVISION DE INGENIERIAS

R.P. FRAY ERICO JUAN MACCI CÃ‰SPEDES, O.P.

SECRETARIA DE DIVISION

E.C. LUZ PATRICIA ROCHA CAICEDO

DECANO FACULTAD DE INGENIERIA ELECTRONICA

ING. CARLOS ENRIQUE MONTENEGRO NARVAEZ

Nota de aceptacion

_____

_____

_____

_____

_____

_____

_____

Firma del tutor

_____

Firma del jurado

_____

Firma del jurado

BOGOTA D.C. _____ DE 2022

ADVERTENCIA

La Universidad Santo Tomas no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velara por que no se publique nada contrario al dogma ni a la moral catolica y porque el trabajo no tenga ataques personales y unicamente se vea el anhelo de buscar la verdad cientÃfica.

Capitulo III Art. 46 del Reglamento de la Universidad Santo Tomas.

Índice general

Capítulo 1

Introducción

Actualmente, los desarrollos científicos han implementado herramientas, las cuales han permitido darles a personas con diferentes tipos de discapacidades, una mejor comunicación con el entorno donde se encuentren, o incluso generar comodidad a aquellas sin discapacidad alguna. La voz ha sido y seguirá siendo uno de los principales medios de comunicación que ha poseído la humanidad a lo largo de su historia, pues surge como un medio que, cuando se usa con un idioma en particular, puede distinguir culturas, países e incluso continentes [1].

En la comunicación entre dos individuos, siempre está un emisor y un receptor; por ejemplo, cuando uno de ellos llama a un asistente personal inteligente, envía una serie de datos por medio auditivo al segundo individuo, que sería el asistente de voz, el cual los recibe, procesa y reconoce, para de esta manera dar una respuesta y una ejecución dependiendo de la orden emitida [2]. A través de estas experiencias diarias y el compromiso de la ingeniería, se entiende que al integrar nuevas tecnologías, incluido el procesamiento de señales digitales, es posible crear herramientas que nos permitan ayudar a las personas en una variedad de situaciones.

En el campo de la medicina constituye un gran desafió lograr desarrollos tecnológicos con acceso para las personas con ciertas limitaciones [3]. La implementación de herramientas para enfrentar las dificultades cotidianas de estas personas, mejorando su calidad de vida, coadyuvando a hacerlas más productivas, y permitiéndoles potenciar su aporte al entorno social [4].

Colombia no es ajena a que parte de su población se encuentre con limitaciones o discapacidades de tipo motriz por diferentes circunstancias, por ende es importante que la sociedad

y sus instituciones genere proyectos y soluciones tecnológicas dirigidas a estas personas [5].

Una aplicación que crea una interacción más personalizada con el usuario en el reconocimiento de señales de voz, puede ser considerada como una de las principales herramientas de procesamiento digital de señales. Por ello, se decidió implementar un sistema que permita el reconocimiento de comandos de voz en tiempo real, el cual puede ser utilizado como una interfaz capaz de ajustar la condición de iluminación de un bombillo en la habitación; Esta interfaz electrónica luego contará con la implementación de algoritmos de procesamiento digital de señales basados en la investigación en un sistema de reconocimiento de señales de voz humana unidireccional, convirtiéndose en una de las herramientas clave que los usuarios usan para acceder a la tecnología.

1.1. Problema

Más de mil millones de personas en el mundo padecen alguna clase de limitación [5]. Como eje de este proyecto, se reconoce la situación vivida por un familiar cercano, de sexo femenino, y de tercera edad que sufre de un nivel avanzado de Osteoporosis en las piernas y cadera, lo cual no le permite tener una movilidad sin que sufra dolor. Debido a esto, para ella, acciones como prender o apagar la luz puede ejecutarlas solo con la ayuda de un tercero. Frecuentemente en los hogares, los interruptores son ubicados en lugares lejanos o inapropiados, lo que dificulta a las personas acceder a ellos y más a las que tienen alguna enfermedad que impida su normal motricidad. En este proyecto, se toma como referencia las personas con limitaciónes, en especial, aquellas en una situación de discapacidad motriz, pues esta es la población de mayor proporción en un 34 %, tanto en hombres como en mujeres, como se ve en la Figura 1-1.

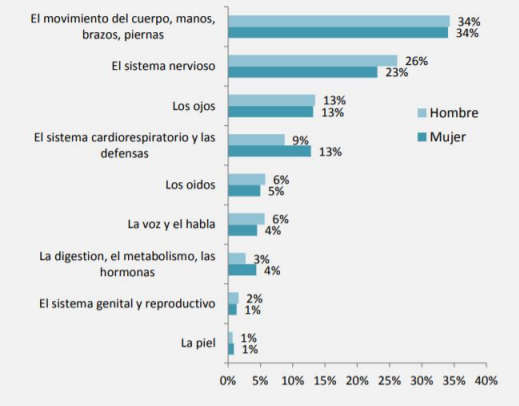

Figura 1-1: Personas con discapacidad según sexo y alteración que más le afecta. RLCPD. Tomado de Sala situacional de las Personas con Discapacidad (PCD); Ministerio de Salud y Protección Social Oficina de Promoción Social, [6].

Igualmente en dicha figura, se observa que la tercera parte de la población, hombres o mujeres, padecen de discapacidad motriz, manos, brazos o piernas. Basado en un estudio, de deduce que estos individuos tienen problemas como movimientos incontrolados, falta de coordinación, poco alcance y fuerza reducida, entre otras [7].

En el transcurso del tiempo, las discapacidades que afectan la movilidad, se presentan en un alto grado en personas de la tercera edad, al ser las más propensas a padecer algún

tipo de enfermedad motriz. El aumento y envejecimiento de la población mundial van a la par con la aparición de mayores enfermedades crónicas, las cuales, son las causantes de las discapacidades [5]. Quienes padecen una discapacidad motriz, tienen dificultades para realizar acciones cotidianas, que serian simples para una persona sana. [5].

En los viejos edificios, hospitales, centros comerciales y viviendas, existen innumerables barreras arquitectónicas y urbanísticas que carecen de tecnologías asistidas para la población con alguna limitación, como interruptores inteligentes, sillas de ruedas electrónicas, o escaleras mecánicas horizontales [9].

Cuando una persona tiene una enfermedad motriz, vive dificultades en sus actividades cotidianas, situación que se empeora por los obstáculos en los diseños de construcción y la ausencia de tecnologías de asistencia, tanto en su hogar como en el entorno social o lugares públicos. Lo que se podría concluir, es que que los problemas que afronta esta comunidad, no dependen solamente de sus condiciones físicas o biológicas, sino también en su interacción con un entorno desfavorable [17].

1.1.1. Pregunta problema

¿Cuál es el sistema de reconocimiento de comandos de voz como interfaz de una habitación con tres atmósferas lumínicas y el cual permita el acceso a tecnologías cotidianas para el uso de las personas que presentan o no alguna condición de discapacidad motriz?.

Preguntas Específicas

Para llegar a la resolución de la pregunta se responderán las siguientes preguntas específicas:

- ¿Cuáles son los pasos necesarios en el procesamiento digital de señales para realizar el reconocimiento de comandos de voz?
- ¿Como implementar un sistema de reconocimiento de comandos de voz en tiempo real para el control de un bombillo con tres atmósferas lumínicas?
- ¿Cuáles son las condiciones de funcionamiento que permiten conocer el porcentaje de aciertos y desaciertos en cada uno de los comandos de voz?

1.2. Justificación

La finalidad del desarrollo de un sistema de reconocimiento de comandos de voz, como interfaz de una habitación con tres atmósferas lumínicas, es facilitar la cotidianidad de la vida de una persona con o sin alguna condición de discapacidad motriz.

La famosa empresa Google Inc realiza su conferencia anual llamada Google I/O, donde miles de proyectos de desarrolladores son presentados, entre los cuales se ha hecho famoso el proyecto *Euphonia*, el cual usa la inteligencia artificial para usar la herramienta de reconocimiento de voz para personas con impedimentos del habla. Este proyecto llevado por Julie Cattiaua, una de las *Manager*, donde enuncia en ingles"quiere ser llevado para que sea un sistema que se adecua por si mismo a los diferentes patrones del habla de las personas por medio del reconocimiento de voz" [4]. También este proyecto, fue promovido por el nuevo servicio premium de Youtube, en la serie titulada "The Age of the AI" (La era de la inteligencia artificial) [10], en el que incentivan el uso de la inteligencia artificial, que implementa herramientas como el reconocimiento de voz.

En el año 2019, el gobierno Colombiano a través del MinTIC (Ministerio de Tecnologías de la Información y Comunicaciones) presentó proyectos dirigidos a la población en situación con discapacidad, durante el "Congreso internacional del Taller de los países miembros del Proyecto Mesoamericano de Integración y Desarrollo" [8]. Un proyecto que permita asistir a las personas con diferentes tipos de discapacidad, genera mas desarrollo en una sociedad; un ejemplo, es de los Gobiernos que buscan suministrar herramientas, y destinar la mayor cantidad de recursos posibles, brindando equitativamente mejor calidad de vida a la población, por ejemplo el proyecto ConverTic, un Software gratuito para rastrear los ciclos de alfabetización de personas con discapacidad visual, que permite a más de 1 millón de colombianos ciegos y con discapacidad visual usar computadoras, acceder a contenido digital y navegar por Internet. [14].

Por eso, exponiendo los antecedentes de proyectos anteriores que, con base al estudio de procesamiento digital de señales en diferentes dispositivos electrónicos de bajo costo, y su implementación en la tecnología asistida, justifican por qué desarrollar un sistema de reconocimiento de comandos de voz, que otorgue una interacción en tiempo real al usuario.

En este proyecto, se utiliza el reconocimiento de comandos voz, el cual le permitirá a personas con discapacidad motriz aprovechar su habla para accionar tecnologías en el hogar. Lo anterior busca brindar una solución al problema de las limitaciones de tipo motriz a la comunidad, ya que es una herramienta usada como interfaz a una aplicación de reconocimiento de voz.

1.3. Antecedentes

A continuación una lista de los antecedentes mas relevantes para este proyecto:

- La plataforma para fabricar la silla de ruedas actual se remonta a la antigüedad, aunque ha sufrido muchas mejoras a lo largo del tiempo, dejando importantes innovaciones y alternativas para el uso actual. La adaptación de este tipo particular de vehículo de movilidad para satisfacer las necesidades de sus usuarios ha dado lugar a importantes desarrollos, como las sillas de ruedas deportivas, de carreras y eléctricas, que permiten que las personas no tengan que hacer el esfuerzo de moverse, como en los vehículos tradicionales para sillas de ruedas, a menudo involucrando la fuerza del brazo ó con la ayuda de de un tercero. Ahora tienes tantas opciones que aquellos que ni siquiera pueden usar la opción eléctrica usarán sus voces en su lugar [11].

  Este modelo consta de dos partes, un micrófono como herramienta de captación de información de datos, el cual se encarga en tiempo real en tomar las señales sonoras que el usuario quiere que este reciba. El micrófono es conectado en el DSP TMS320C6711, luego usando la salida digital de este conectado a un PIC 16f876A para hacer accionar un servo motor que maneja una silla de ruedas mostrado en el artículo de M. T. Qadri & S. A. Ahmed en [11]. En este caso se evidencia la factibilidad que tiene el uso del procesador digital que contienen estos dispositivos secuenciales. En la Figura 1-2 se ilustra el modelo explicado:

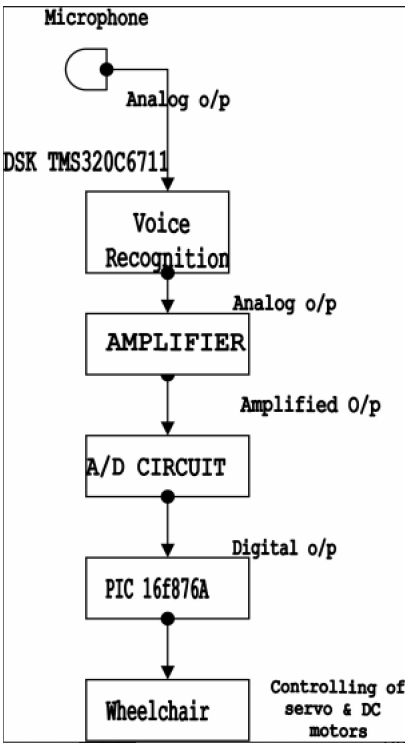

Figura 1-2: Modelo del sistema de silla de ruedas manejada por la voz. Tomado de M. T. Qadri and S. A. Ahmed, Voice Controlled Wheelchair Using DSK TMS320C6711 [11].

- El asistente virtual llamado “Siri”, el cual permite el uso exclusivo de la voz como principal interfaz entre el usuario y su uso con el *Smartphone*, como enviar un mensaje de texto o email, llamar, preguntar una ubicación, programar una alarma o recordatorio, entre otras de sus funciones. En específico, por medio de asistentes virtuales personales con inteligencia artificial como: “Siri” , “Cortana” y “Alexa” entre otros, contiene desarrollos incorporados a los textitSmartphones, Tablets, Laptops y más dispositivos con solo la voz [18].

- También se analiza el trabajo de Fabian A. B. & Henry N. en el articulo [21] donde se demuestra cómo el interfaz que se quiere implementar entre un usuario con una máquina, es de un valor trascendental para los avances en esté sentido. Los dispositivos de hardware reconfigurable como la FPGA, DSP, ARDUINO, RASPBERRY O ESP32, logran ser herramientas de gran alcance para los estudiantes universitarios, en el artículo [21] se analiza que son dispositivos factibles para hacer este tipo de proyectos, aunque los resultados terminan siendo no un porcentaje mayor al 80 % o 90 % de acierto, cuando se quiere reconocer ciertas palabras, pues incluso hay algunas que no se lograron establecer como factibles para un mínimo reconocimiento. Como se puede ver en la Figura 1-3:

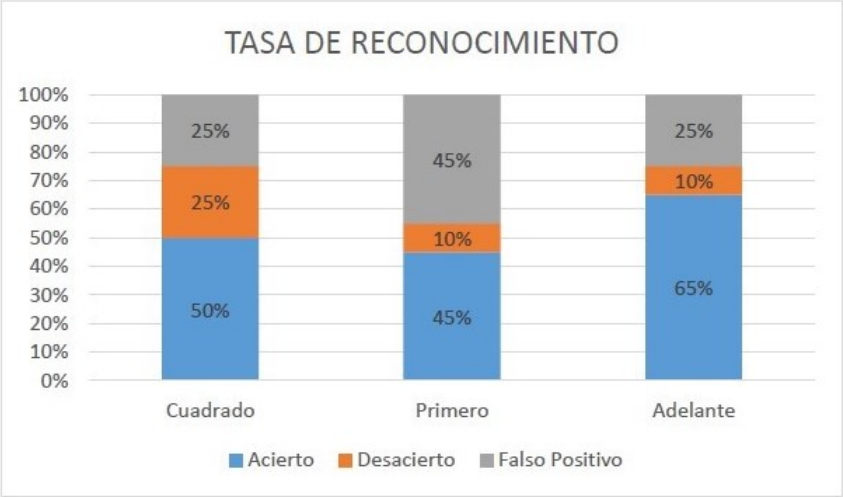

Figura 1-3: Tasa de reconocimiento del proyecto "Implementación de una Sistema de Reconocimiento de Voz en FPGA como Interfaz Hombre-Máquina en Aplicaciones de Robótica". Tomado de Fabian ALberto Bustos Goméz Y Henry Navy Pantevis Perilla, Implementación de un Sistema de Reconocimiento de Voz en FPGA como Interfaz Hombre-Máquina en Aplicaciones de Robótica [21].

La Figura 1-3 es un ejemplo de los resultados cuando se implementa la tecnología de reconocimiento de comandos de voz, en el que se observa para tres comandos la tasa de efectividad, pues en la implementación de estos proyectos abarcan el uso de las DSP combinados con proyectos de tecnologías de asistencia, como puede ser una silla de ruedas, entre otros.

En la Figura 1-4 se muestra el avance del reconocimiento de voz a lo largo de la historia, relacionando también con los previos documentos usados en esta sección.

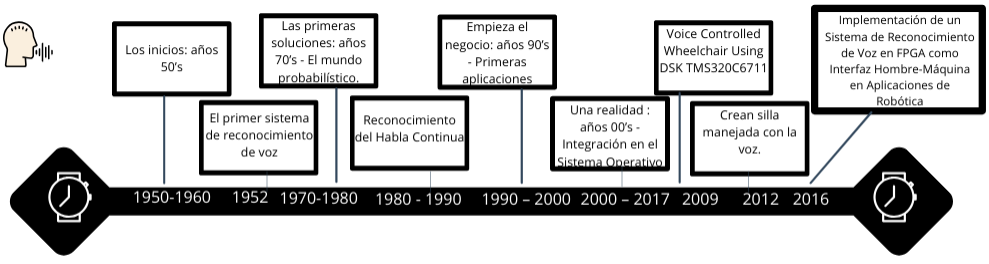

Figura 1-4: Historia de los Sistemas de Reconocimiento de Voz. Tomado de HISTORIA DE LOS SISTEMAS DE RECONOCIMIENTO DE VOZ [13].

Capítulo 2

Objetivos

General:

- Implementar un sistema de reconocimiento de comandos de voz como interfaz de una habitación con tres atmósferas lumínicas, para el uso en personas con o sin alguna condición de discapacidad motriz.

Específicos:

- Describir los pasos necesarios en el procesamiento digital de señales para realizar el reconocimiento de comandos de voz.
- Implementar un sistema de reconocimiento de comandos de voz en tiempo real para el control de un bombillo con tres atmósferas lumínicas.
- Evaluar las condiciones de funcionamiento que permitan conocer el porcentaje de aciertos y desaciertos en cada uno de los comandos elegidos.
- Crear un manual instructivo fácil y didáctico para la explicación y forma de uso del sistema de reconocimiento de comandos de voz, que indique el procedimiento necesario para crear la base de datos de las palabras ya establecidas.

Capítulo 3

Marco teórico

Para realizar un reconocimiento de voz, se requiere que el sistema que toma el tramo de datos de la grabación de voz en el intervalo de tiempo haga una extracción de las características de estos. El realizar está segmentación, permitirá analizar el entrelazamiento de los fonemas en las palabras, dividiendo en franjas el tramo obtenido, y así analizar y obtener las características de cada palabra captada, permitiendo finalmente comparar el comando recibido con las palabras que se quieren reconocer en el sistema. Este tipo de proyecto es un ejemplo de las tecnologías de asistencia combinadas con los dispositivos que usan procesamiento digital de señales [21]. En la Figura 3-1 se visualiza en bloques el proceso para realizar un reconocimiento de voz:

Figura 3-1: Figura en la que se ven los bloques de reconocimiento de voz. Tomado de Fabian ALberto Bustos Goméz Y Henry Navy Pantevis Perilla, Implementación de un Sistema de Reconocimiento de Voz en FPGA como Interfaz Hombre-Máquina en Aplicaciones de Robótica [21]

3.1. Bloques del sistema de reconocimiento de voz

Los bloques del sistema de reconocimiento de voz comprenden:

3.1.1. Captura de voz

Etapa donde se hace la obtención de los valores del comando emitido en grabación cuando el sistema esta en modo de captura de datos. Normalmente esta se hace por medio de un micrófono, pues toma la señal analógica del sonido de entrada y la transmite al sistema. Toda captura de voz se puede hacer en tiempo real donde el sistema esta siempre en modo de captura de voz ó también puede hacerse en intervalos de tiempo donde se la captura de datos se hace por un tiempo determinado.

3.1.2. Pre-procesamiento

Etapa donde se toman los datos obtenidos y se resaltan los de la voz de la persona. Esta etapa contiene varios procesos como el acortar señal de audio tomada y recortarla dejando solo el audio del comando emitido por el usuario. Todo audio recortado pasara por un proceso de filtrado, aislando el sonido no deseado por medio de un pre-énfasis, para obtener un tramo de datos que tenga solo el comando que se ha tomado en el intervalo de tiempo

3.1.3. Segmentación y Extracción de características

Etapa donde se realiza una segmentación que divide en pequeñas franjas los datos obtenidos diferenciando los fonemas de las palabras. Luego, creando matrices de valores que contendrán las características únicas del comando capturado.

3.1.4. Comparación

Ultima etapa donde se toma los datos obtenidos, para compararlos con la base de datos de los comandos ya establecidos, detectando el que se quiere reconocer.

3.2. Arduino UNO

El uso de un DSP (Digital Signal Processor) en las aplicaciones aporta una ventaja competitiva respecto al uso de sistemas analógicos [16]. La ventaja de la utilización del DSP respecto a los sistemas analógicos radica en su versatilidad en modificar valores en un

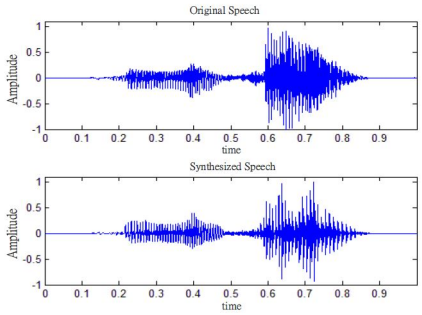

Figura 3-2: Oscilograma original de un discurso y uno sintetizado. Tomado de G. Huang and Z. Tian, 'An automatic speech recognition system on DSP board", [32].

programa, como en la implementación de algunas tareas que podrían ser difíciles o imposibles utilizando sistemas analógicos y con un coste menor. No obstante microcontroladores de bajo costo como son Arduino, Raspberry y ESP32, brindan herramientas sencillas para programación, convirtiéndose estos en dispositivos para un procesamiento digital de señales.

3.3. MATLAB

El software MATLAB, es una gran herramienta que permite reconocer palabras, las cuales que pueden ser grabadas, teniendo en cuenta que la duración de las palabras debe ser de 0.7 segundos como una medida de tiempo óptima para obtener los datos en el comando de MATLAB y luego las palabras se dividen en bloques iguales. Cada bloque de voz se procesa para funciones importantes como cruce por cero, desviación estándar y análisis de energía total [11].Por ejemplo en la Figura 3-2 se puede visualizar un audio sintetizado por medio del reconocimiento de voz.

La Etapa de MATLAB implementa toda la matemática necesaria, donde se hace el análisis y estudio de la voz, el articulo de F. Bustos Y H. Pantevis [21] da un ejemplo de como MATLAB es implementado en el desarrollo de un sistema de reconocimiento de voz, pues por medio de este se puede basar la matemática a implementar en placas de programación. En la Figura 3-3 se muestra cómo en este artículo es visualizada una de sus etapas de segmentación que es frecuentemente usada en los sistemas de reconocimiento de voz.

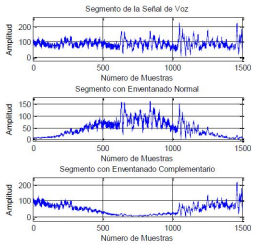

Figura 3-3: Segmentación para un tramo de la señal de voz. Pagina 65-110. Tomado de Fabian ALberto Bustos Goméz Y Henry Navy Pantevis Perilla, Implementación de un Sistema de Reconocimiento de Voz en FPGA como Interfaz Hombre-Máquina en Aplicaciones de Robótica [21].

En MATLAB se puede implementar diferentes procesos para el desarrollo el reconocimiento de voz, que permite analizar los datos y obtener sus características, entre ellos la *correlación*. Para diseñar un sistema de reconocimiento de voz se debe primero tener conocimientos en el proceso del habla, pues dependiendo del tipo con el que se quiera trabajar, se tiene en cuenta qué herramientas son las mejores para este, en este caso las características de la voz con las que se trabajan serán en un cuarto aislado para un usuario. Técnicas conocidas como FFT (*'Fast Fourier Transform*), *Windowing* (*"División de las señales de entrada en segmentos temporales"*) como se ve en la Figura 3-4 como ejemplo y STFT (*Short-time Fourier transform* son las principales técnicas usadas.

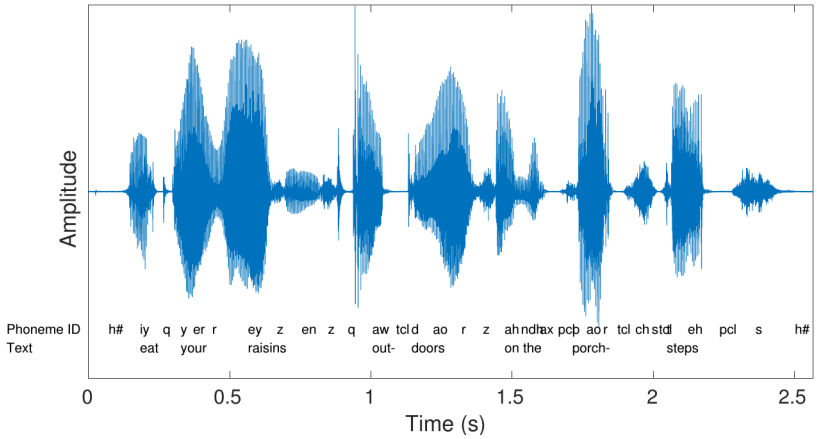

Figura 3-4: Ejemplo de técnica Windowing para un texto. Tomado de Tom Bäckström, Windowing, Basic representations and models, Introduction to speech processing [30]

Capítulo 4

Descripción de los pasos realizados en el procesamiento digital de señales para realizar el reconocimiento de comandos de voz

La comunicación entre el humano y las máquinas ha tenido un notable avance en los últimos 100 años. La interpretación natural y efectiva, para lograr que una máquina ejecute el comando enviado por el humano ha tenido muchos limitantes, los tecnológicos, entre otros. También, el desarrollo de la ingeniería, ha permitido adoptar a las máquinas nuevas capacidades de recibir y seguir los comandos que el humano implementa, llegando incluso a crear unos nuevos, basados en los avances, como por ejemplo, la inteligencia artificial.

El procesamiento digital de señales que permite la manipulación de una señal discreta entrante, brinda una transformación al lenguaje de la máquina, que brinda cumplir con tareas mas complejas. Esta transformación digital puede ser usada no solo en señales unidimensionales, sino incluso, el procesamiento de imágenes o videos, los cuales se como conocen como señales verticales. La voz del humano es una onda sonora, que reducida en los limites de frecuencia, permiten ser escuchadas por su sistema auditivo. Esos límites rondan entre los 20 Hz a 20k Hz. Se trabajará con una frecuencia de muestreo de 50k Hz, ya que brinda

una mayor tasa de efectividad, según estudios de Fabian Alberto Bustos Goméz & Henry Navy Pantevis Perilla [21] y David Reig Albiña [23].

4.1. La señal de voz humana

La señal de voz emitida por el ser humano se produce por diferentes mecanismos de los que constan su aparato fonador: el respiratorio, fonador, resonancia, y articulación [27], estos constituyen una obra de ingeniería biológica, que ha permitido estudios, análisis, y avances para la Ingeniería. En este caso de estudio, la voz humana será grabada en un espacio cerrado, que no permita un nivel de ruido alto, para fácilmente extraer las características de la voz con la cual se quiera hacer un reconocimiento de comandos de voz.

4.2. Características de la señal de voz

La señal acústica de la voz se puede convertir en una respuesta eléctrica analógica mediante un micrófono, y esta se visualiza en un par de ejes cartesianos en el tiempo. Por ejemplo, se muestra la forma de onda del comando "Encender bombillo" en la Figura 4-1. Es importante representar la señal del habla (voz) en función del tiempo, porque proporciona información sobre características importantes (como la energía y el cruce por cero), facilitando su procesamiento y análisis.

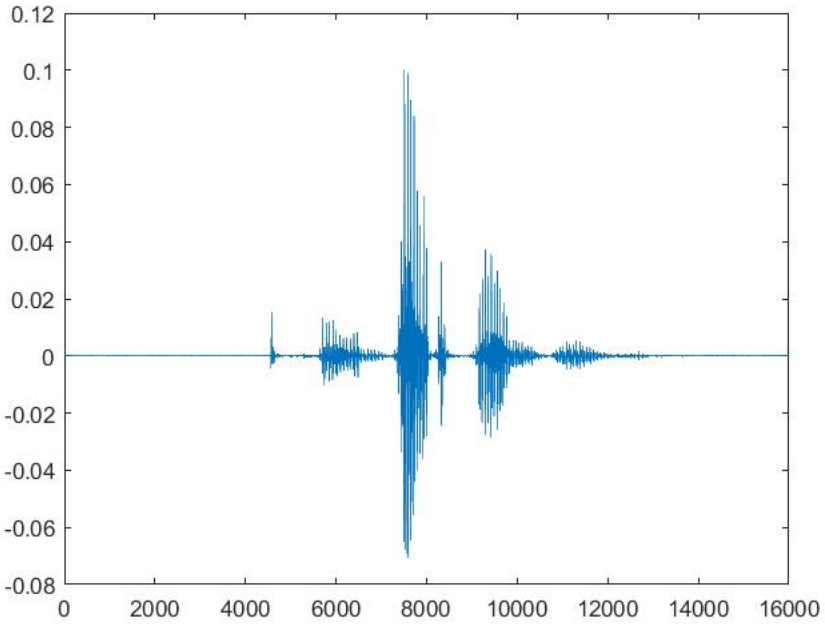

Figura 4-1: Gráfica del audio obtenido al decir "Encender bombillo'. Elaboración propia

4.2.1. Energía y cruce por cero

La energía es el primer parámetro que se puede medir en las señales de voz. Mediante ella, se da a conocer un valor en una parte del tiempo, que permite determinar si hay voz o no, reconociendo así la actividad básica de señal de voz, en el que se puede saber si es un tramo sonoro, definido como el momento en que entra una vibración, donde la energía va aumentando, y en los tramos sordos, donde la energía va disminuyendo. La ecuación 4.1 es la que se define para el calculo de la energía localizada:

$$E[m] = \sum_{n=0}^{N-1} x[n]^2 w[n-m]^2 \tag{4.1}$$

Donde: x representa a la señal de voz, m la muestra, w representa al perfil de la ventana de análisis y N es el tamaño de la ventana.

La tasa de cruce por ceros mide las veces que la señal pasa por el nivel cero el segmento de análisis. Lo que ayuda a proporcionar una idea general de la distribución en frecuencia de la señal [28].

4.3. Captura de voz

La captura de datos de la voz humana como señal se hace usualmente por medio de un micrófono, el cual otorga un valor analógico en el periodo de tiempo establecido, para así guardar los datos y poder ser representados en forma digital; estos datos son analizados, transformados y manipulados por medio de ecuaciones y funciones en MATLAB que permiten extraer las características necesarias que permita al usuario reconocer el comando emitido.

En las primeras pruebas realizadas se hizo uso del micrófono integrado en las computadoras. Se analizaron los métodos de manipulación de datos de audio usando el Software de MATLAB, tales como el audio entrante o uno ya grabado, obteniendo los archivos de audio que serán los datos base para análisis en el manejo del microcontrolador. Al inicio, se implementan comandos como *Audiorecorder*, *Getaudiodate* y *fprintf*, para grabar y manejar los archivos de audio que se obtengan por medio de un micrófono. La señal de voz se recopila con un período de muestreo de 50K Hz y la duración promedio de cada palabra es de 950 ms (milisegundos).

Por medio del comando *Audiorecorder* se obtiene el primer tramo de datos de la voz del usuario por un tiempo personalizado, para hacer la base de datos de los comandos que se desea usar en el primer reconocimiento de voz. *Getaudiodata* y *fprintf* serán los comandos con los cuales se puede obtener los datos en el archivo de voz grabado por el comando *Audiorecorder*, y guardarlo en un archivo de texto dando inicio el proceso de detección de extremos, segmentación, extracción de características y comparación. En la Figura 4-1 se muestra el resultado del audio obtenido al recibir el comando “Encender Bombillo” con el código en el Anexo 10.2 se muestra la primera prueba.

4.4. Pre-procesamiento

En la etapa de pre-procesamiento se manipulará la señal de voz capturada para obtener una señal que contenga solamente la palabra emitida por el usuario, por medio de una eliminación que otorga una señal de extremos basada en el estudio de la energía de la trama de datos, y finalmente, para compensar la atenuación en la señal será usado un filtrado o

etapa llamada pre-énfasis. Algunos ejemplos de uso, se pueden visualizar en el articulo [21].

En primer lugar, se declara en MATLAB la longitud del audio que se estará tomando en tiempo real, en este caso de 3 segundos, 50K Hz como la frecuencia de muestreo, una resolución de 24 bits, y un audio de tipo mono estéreo, y finalmente, una ganancia y nivel de *Offset* que harán la señal de audio en unos valores mas fáciles de manejar.

La longitud del vector que se usará como buffer estático para esta señal de audio de 3 segundos con los parámetros mencionados se calcula en la siguiente fórmula:

$$\text{Fs} = 50000 \text{ Hz}$$
$$\text{Tiempo de audio} = 3 \text{ segundos}$$
$$\text{C} = \text{Cantidad de muestras}$$

$$Fs * T = C \tag{4.2}$$

$$\text{3seg x 50000} = 150000 \text{ muestras}$$

4.4.1. Detector de extremos

En el paso de detección máxima, se realiza un análisis del búfer obtenido del sonido generado por el usuario, siendo el principio y el final de la palabra almacenada mucho más cortos que el primero. Los comandos seleccionados tienen en promedio 950 ms de longituda, por lo cual el búfer de estos, será mucho mas corto que uno de 3 segundos.

La detección de puntos extremos se basa en el análisis de la evolución de la potencia y las intersecciones de cruces por cero. Sin embargo, existen alternativas como el coeficiente COPER, que combina estas dos técnicas, lo que permite evaluar el crecimiento de una señal mediante un único coeficiente [23]. Luego, en este paso se detecta el inicio de la palabra y se almacenan las tramas hasta que se detecta el final de la instrucción. La siguiente Figura 4-2 muestra un resumen esquemático de este paso.

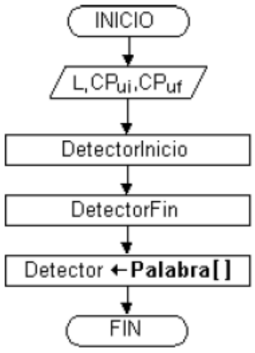

Figura 4-2: Diagrama de flujo de la sub-rutina detector automático de extremos. Tomado de David Reig Albiña, Implementación de algoritmos para la extracción de patrones característicos en Sistemas de Reconocimiento De Voz en Matlab y de Fernando Peralta, Reyes Aníbal Cotrina Atencio, Asesor: Guillermo Tejada Muñoz. RECONOCEDOR Y ANALIZADOR DE VOZ, [23] y [24]

- COPER

  El parámetro COPER [24] permite analizar la evolución temporal de las tramas, y su fórmula se encuentra a continuación, utilizando la ecuación 4.1 de cero-intersecciones, pero multiplicada por el análisis energético de la muestra, de tal manera que la intensidad acumulada depende no sólo en el signo de cambio, sino también en su amplitud para que el ruido de fondo no alcance una gran acumulación de densidad.

$$COPER = \sum_{L}^{m=0} |y[m] \cdot |y[m]| - y[m-1] \cdot |y[m-1]|| \tag{4.3}$$

  L es tamaño de la trama

- Detección Inicio y Fin

  Al detectar el umbral de inicio usando el parámetro de COPER, se comienza a almacenar los datos de las tramas en un buffer dinámico que se ira almacenando en un nuevo vector *Ssr[]*, hasta que sea detectado el fin del comando emitido. Establecido previamente un valor de umbral de energía para las tramas, y asignado un número de tramas que no superen el umbral, se podrá detectar un silencio, esto también sirve

para comandos con letras que causarían problemas, como los que tienen la letra m o n, que en la pronunciación parecen un silencio [23].

Al terminar el proceso de detección de fin del comando, y sus datos guardados en un nuevo vector, la función entregará al programa principal un vector únicamente con los valores del comando pronunciado y ya recortado, pero en este caso la longitud de este vector no será de 150000 muestras como lo fue del audio original, si no será mucho mas corta, como por ejemplo para la palabra "Cálido", del cual se obtuvo un promedio de tiempo de pronunciación de 420 ms , lo que equivale a un vector con longitud de 21001 datos, en la Figura 4-3 se muestra la diferencia del audio de 3 segundos comparado con la palabra "Cálido" con la misma, pero ya detectada de inicio a fin.

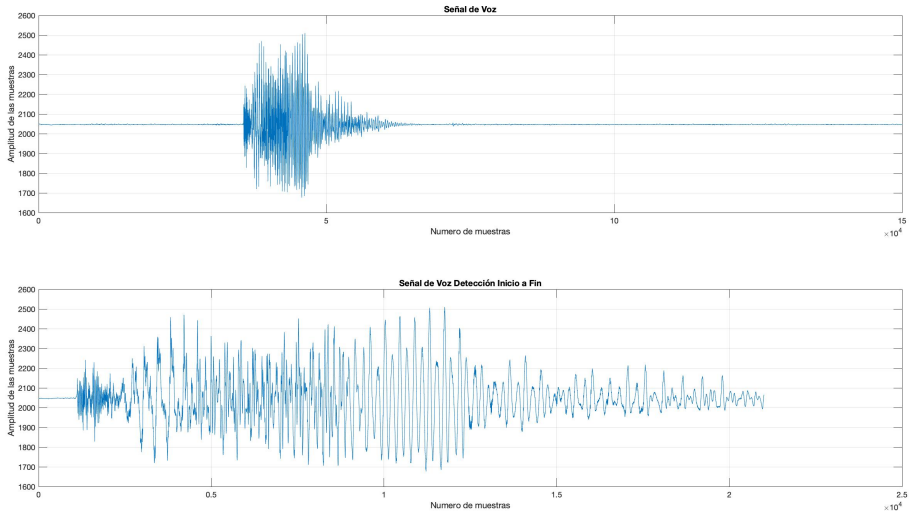

Figura 4-3: Gráficas de audio para la palabra Cálido. Y gráfica de esta misma palabra detectada de inicio y fin. Elaboración propia

4.4.2. Pre-Énfasis

Se ha implementado un filtro FIR de primer orden, como se muestra en la ecuación 4.4, este filtro para compensar la atenuación que se produce debido a las características fisiológicas del sistema del habla humano, es también de 6 dB/octava [26]. Para reducir el ruido en la grabación y aumentar las características acústicas.

$$S_{pp}(n) = S_{sr}(n) - \alpha S_{sr}(n-1) \tag{4.4}$$

Donde α es una constante arbitraria establecida en *0.9609375* [25], $S_{sr}(n)$ es la señal sin ruido de la detección anterior y el $S_{pp}(n)$ final es la salida del filtro. La longitud del vector de salida será la misma que la señal recortada sin ruido, como se muestra en la Figura 4-4, la palabra "Cálido" con el filtro de primer orden aplicado, en comparación con la palabra recortada y sin recortar.

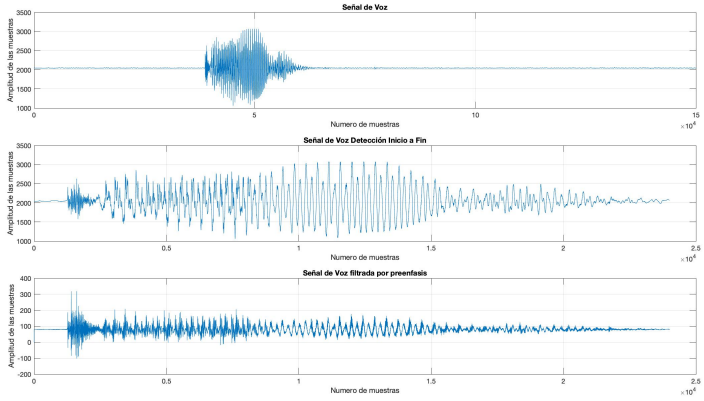

Figura 4-4: Gráficas de audio tomado para la palabra Cálido original, detectada de inicio y fin, y ya filtrada por pre-énfasis. Elaboración propia

4.5. Segmentación

En esta etapa, para las señales de voz se aplica en esta etapa una segmentación de 1500 datos por cada bloque, para poder obtener esta cantidad se dividiría con redondeo la longitud de la señal ya recortada y filtrada sobre 1500. A continuación, en un proceso llamado creación de ventanas, cada intervalo se multiplica por la ventana de Hamming mostrada en la ecuación 4.5 . Las ventanas se superponen entre sí en un 50 % para reducir la discontinuidad y realzar el formante del fonema. La ventana inicial se denomina ventana normal y ventana complementaria superpuesta. Las ecuaciones 4.6 y 4.7 corresponden a los segmentos de la ventana normal Swc y la ventana adicional Swcv, respectivamente [21], recorriendo el vector que contiene la señal de voz:

$$H_w = \begin{cases} 0.54 - 0.46cos\left(\frac{2\pi n}{N-1}\right) & Si 0 \leqslant n \leqslant N-1 \\ 0 & Otro\ caso \end{cases} \tag{4.5}$$

$$S_{wc} = S_{pp} \cdot H_{wc} \tag{4.6}$$

$$S_{wc} = S_{pp} \cdot (1.08 - H_w) \tag{4.7}$$

Se obtendrá dos vectores que se usarán para el proceso de autocorrelación, pero esta etapa será por cada 1500 datos del vector original, con el que se está haciendo la segmentación, por lo que se haría dentro de un ciclo repetitivo del valor de la longitud del vector del audio sobre 1500.

4.5.1. Extracción de características

Esta sección está dividida en la Autocorrelación como primer etapa; y Reflexión y Codificación predictiva lineal de Coeficientes (LPC) como segunda. Previamente la etapa de extracción de características facilitó la obtención de un vector característico de m coeficientes para cada segmento que se obtuvo del proceso de segmentación. En esta etapa se hace una reconstrucción del método recursivo de Levinson-Durbin, cuyo objetivo es reducir la complejidad de la implementación del hardware [20]. La ecuación 4.8 muestra la etapa de extracción de características, teoría aplicada para obtener la matriz de los valores de LPC del comando de audio, que permitirá terminar el proceso de la extracción:

$$\begin{gathered} for(i = 2; i \leq m; i++) \\ \{ \\ K_i = \frac{r_{xx}(i) - r_{xxb}[i-1]K_{i-1}}{r_0(i) - r_{xx}[i-1]K_{i-1}} \\ a[i] = \begin{bmatrix} a[i-1] \\ 0 \end{bmatrix} + K_i \begin{bmatrix} -a_b[i-1] \\ 0 \end{bmatrix} \\ \} \end{gathered} \tag{4.8}$$

Donde los coeficientes K_i representa la reflexión y K_{i-1} la de la iteración anterior, r_{xx} el

vector de coeficientes de autocorrelación, r_{xxb} el vector inverso de r_{xx} y a_b el vector inverso de LPC. En la extracción de características, la autocorrelación devuelve un vector de $m+1$ coeficientes por cada segmento, mientras que la extracción de características entrega un vector de m coeficientes.

- Autocorrelación

  La etapa de extracción de características se divide en dos partes, que entregan los coeficientes necesarios para hacer una comparación que reconocería que comando de voz ha emitido el usuario. Para obtener los coeficientes de autocorrelación, se usará la ecuación 4.9(hasta m+1 coeficientes), esta sumatoria se aplicará en un ciclo repetitivo recorriendo las muestras de cada segmento y al mismo tiempo guardando en otro ciclo los $m+1$ coeficientes de autocorrelación, almacenados para las ventanas normales y complementarias en un vector único.

$$r_{xx}(i) = \sum_{n=0}^{N-1} S_n \cdot S_{n-i}; 0 \leq i \leq m \tag{4.9}$$

  Donde S_n es la muestra de entrada, Sw es el vector normal, Swc es el vector complemento, N es el número de muestras y m es el número de coeficientes.

  La Figura 4-5 de [21] muestra un diagrama de bloques de autocorrelación.

- Reflexión y LPC

  El coeficiente de reflectancia se obtiene de la ecuación 4.10. Los coeficientes LPC de cada segmento son entonces el resultado de un cálculo utilizando la ecuación 4.11, donde cada segmento se deriva recursivamente de los coeficientes anteriores, $a_{[i]}$ comienza con dos líneas, pero aumenta de tamaño con cada nuevo factor calculado, por lo que a tendrá m líneas.

$$K_i = \frac{r_{xx}(i) - r_{xxb}[i-1]K_{i-1}}{r_{xx}(0) - r_{xx}[i-1]K_{i-1}} \tag{4.10}$$

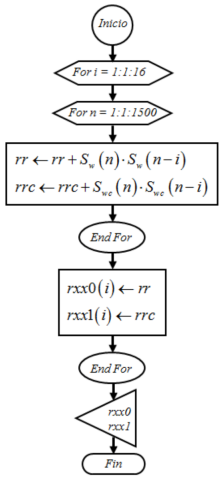

Figura 4-5: Diagrama de flujo del bloque de autocorrelación. Tomado de Fabian ALberto Bustos Goméz Y Henry Navy Pantevis Perilla, Implementación de un Sistema de Reconocimiento de Voz en FPGA como Interfaz Hombre-Máquina en Aplicaciones de Robótica [21]

$$a[i] = \begin{bmatrix} a[i-1] \\ 0 \end{bmatrix} + K_i \begin{bmatrix} -a_b[i-1] \\ 1 \end{bmatrix} \tag{4.11}$$

A continuación, se calcula como $a[1]$ y se asigna a K_i .El primer bucle hecho con el comando for repite este proceso para cada coeficiente, en este caso $m = 15$. El segundo bucle de for realiza un número apropiado de iteraciones de acuerdo con el valor de i, y calcula de forma recursiva el i-*ésimo* coeficiente LPC. Dado que el valor de K_i varía de 0 a 1, multiplicar por 8 en el cálculo de $a[1]$ y multiplicar por 4096 en el cálculo de K_i es suficiente para evitar el uso de números de coma flotante. Por otro lado, al calcular a_j, se divide en 4096 partes para preservar la dinámica de la ecuación. La elección de las constantes 8 y 4096 se hace para lograr un truncamiento de 12 bits, esto para limitar el valor de los coeficientes de LPC.

4.6. Comparación

Debido a factores como la duración de la señal de voz efectiva, el tono de la pronunciación y el ruido de fondo, el comando de voz cambiará drásticamente, incluso si la pronuncia la misma persona. Por lo tanto, la comparación de la instrucción de referencia y la palabra

recién aprendida debe hacerse al azar.

El coeficiente de correlación de Pearson es el resultado de la covarianza y la desviación estándar de los dos vectores de datos, donde x y y son los vectores de datos a comparar, $cov(X,Y)$ es la covarianza de X,Y y σ la desviación estándar. Como se muestra en las ecuaciones 4.12, 4.13 y 4.14. Luego, utilizando las definiciones de covarianza y desviación estándar del cálculo del coeficiente de Pearson, la ecuación final se representará como se muestra en la formula 4.15.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{4.12}$$

$$cov(X,Y) = \frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) \tag{4.13}$$

$$\sigma_X = \sqrt{\sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1}} \tag{4.14}$$

$$\rho_{XY} = \frac{\frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{n-1} \cdot \sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^{n} (y_i - \bar{y})^2}} \tag{4.15}$$

El proceso de comparación se basa en tomar un nuevo vector para obtener el coeficiente LPC del comando recién pronunciado, y compararlo con el coeficiente LPC de referencia obtenido al realizar la extracción de características de todos los comandos.

Calculando con cada vector de referencia de los audios ya pre-grabados y el nuevo comando, se comienza con un ciclo for, donde se usará el valor medio de los dos coeficientes LPC y una sumatoria respectiva según la Ec. 4.12. En la misma secuencia se obtiene la varianza de ambos vectores, y al final, enel ciclo for se calcula la desviación estándar, expresada por la raíz cuadra de la varianza. Para esta desviación estandar se realiza la división que dará como resultado el CCP (coeficientes de correlación de Pearson) por cada iteración del ciclo for.

Los coeficientes de correlación de Pearson (CCP) oscilarán en valores de 14000 a 15000, pero se toma 15291 como el valor que establece un positivo en la comparación de los coeficientes de LPC de ambos audios, así por medio de una nueva variable C se cuentan los

coeficientes de correlación de Pearson, que superaron este umbral seleccionado. El comando adquirido es considerado válido si la totalidad supera el umbral en un 80 % sobre la cantidad de comparaciones. Con cada nuevo comando emitido se hace comparación con toda la base de datos de referencia, hasta encontrar el que alcance mas del 80 %. La elección de umbrales para las comparaciones de CCP se describe en detalle en la siguiente sección junto con los resultados de la implementación del software.

4.7. Implementación

La implementación del sistema completo comienza por el uso de un ordenador portátil, pues por medio de este se conectarían todos los periféricos, así como el bombillo que está conectado a una placa de Arduino UNO. Se optó por usar el Arduino pues permite directamente usar el código creado en MATLAB como el software que manejara la placa y ejecutar el comando dependiendo del resultado del reconocimiento de voz. Luego de tener conectado el Arduino al computador y preparado el codigo en MATLAB, se conecta a la placa un módulo de Relé que va conectado con el bombillo de tres atmósferas lumínicas.

4.7.1. HARDWARE

Después de implementarse en el software y comprender el proceso de reconocimiento de voz, el sistema se ejecutó en la placa de microcontrolador de código abierto Arduino Uno [22] utilizando el paquete de soporte de MATLAB para la escritura en estas tarjetas. Esta sección muestra un diagrama general, donde el Anexo 10.4 contiene el código implementado en MATLAB para procesar el sistema y conectarlo a la placa Arduino UNO.

4.7.2. Adquisición de la señal de voz

La adquisición de la señal de voz se realizará usando el micrófono integrado del computador, con la que la demostración en tiempo real de reconocimiento de los comandos permitirá saber cuál se detectó. Las ventajas de este sistema, es que tanto se puede escoger la resolución en bits, como máximo valor 24, para darle la mejor calidad de audio al sistema a trabajar, como escoger el tipo de canal, que para este caso se usará en Mono-estéreo.

Capítulo 5

Implementación de un sistema de reconocimiento de comandos de voz en tiempo real para el control de un bombillo con tres atmósferas lumínicas.

5.1. Grabación y análisis de base de datos

Para la grabación de la base de datos que se usó en el sistema, se grabó usando un simple comando de voz y su respectiva gráfica, como se puede ver en el código del Anexo 10.3 y la Figura 5-3 donde se puede visualizar cada audio.

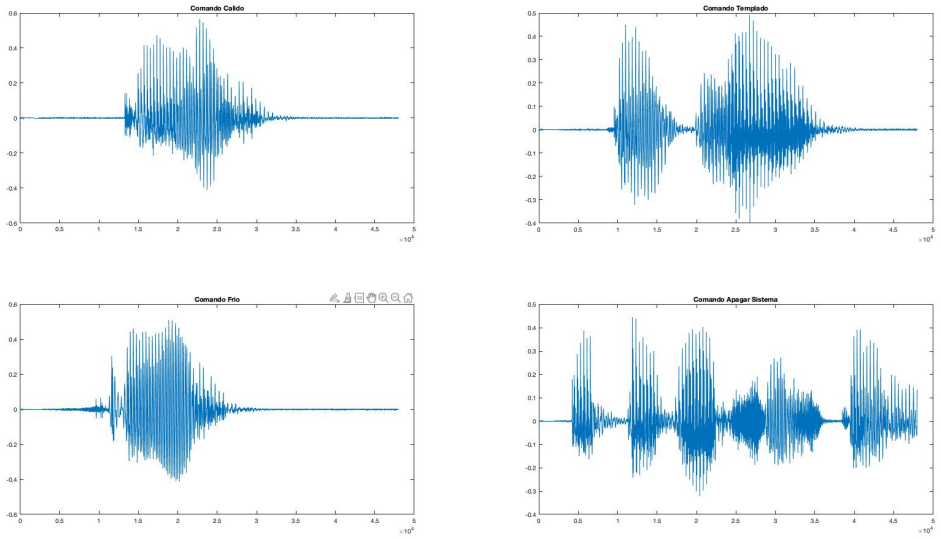

Figura 5-1: Gráfica de cada audio comando del sistema. Elaboración propia.

Luego a cada uno de los comandos de audio se aplicó: la detección de extremos, pre énfasis, segmentación, auto correlación, y finalmente, se guardó los coeficiente de LPC en un archivo de texto. Para comprobar el funcionamiento de cada etapa, se usó comandos como *Plot* para ver en gráfica que los coeficientes se mostraran correctamente, este código se puede analizar en el Anexo 10.3.

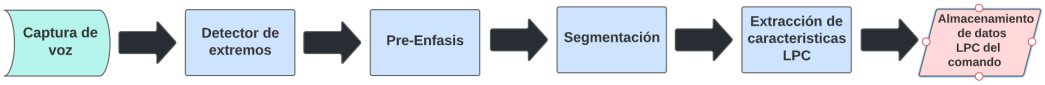

Figura 5-2: Diagrama de bloques para captura de audio y extracción de características - LPC. Elaboración propia.

Con esto se comprobó que cada audio se recorta como se observó en la Figura 5-3 y por consiguiente el pre énfasis que es un filtro donde muestra un audio más limpio, como se mostró en la Figura 5-4, y finalmente, que la longitud de los coeficientes de LPC depende de la longitud de cada comando, como por ejemplo, los coeficiente de LPC de “Frío” serian mas cortos que los de “Apagar Sistema”. En la figura 5-3 y 5-4 se hace gráfica de los coeficientes del comando “Frío” y “Apagar Sistema”:

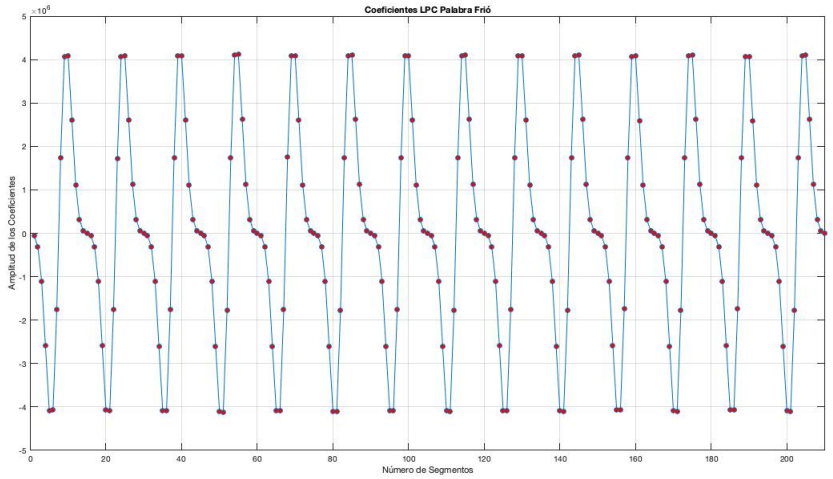

Figura 5-3: Gráfica Coeficiente LPC de Comando Frío. Elaboración propia.

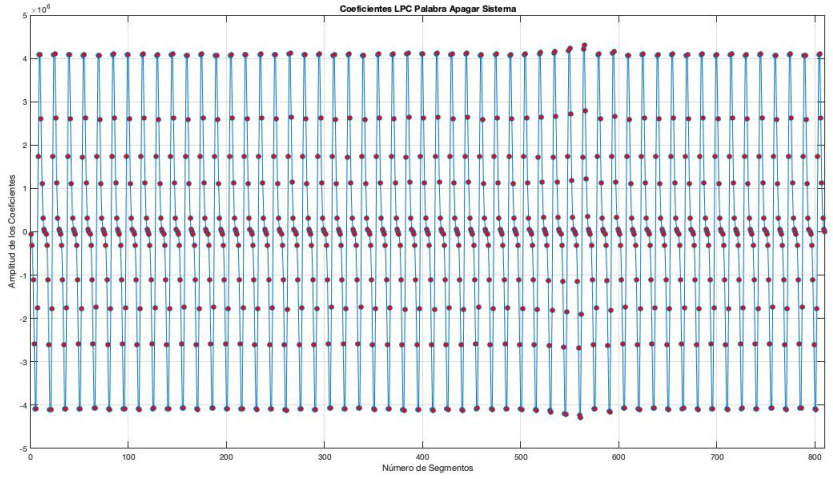

Figura 5-4: Gráfica Coeficiente LPC de Comando Apagar Sistema. Elaboración propia.

5.2. Control

Con las condiciones de operación evaluadas para una aplicación, se determinó la interfaz apropiada y encargada para la comunicación entre este y la de luminiscencia del cuarto.

Como etapa de activación para el bombillo, se maneja mediante la voz la luminiscencia del cuarto con una etapa de control formado por la placa de Arduino UNO y un módulo de Relé. Para hacer cambio de la intensidad lumínica de este bombillo, su control se basa en la cantidad de encendidos que recibe, pues es acumulativo. Teniendo solo tres intensidades: Cálida, Neutra y Fría. Así finalmente se obtuvo un sistema de reconocimiento de voz que interactúa con el usuario en tiempo real y ejecuta los comandos que previamente se tienen guardados, estos para que el usuario maneje la luminiscencia del cuarto.

También se dispondrá de un comando de apagado para cuando el usuario lo deseé, así el sistema completo se apagará sin importar en que estado de luminiscencia esté, y esperará cualquiera de los tres comandos de intensidad por parte del usuario, si el sistema recibe un comando, el bombillo inicia su encendido en el nivel de luminiscencia deseado. Para la habitación con 3 atmósferas lumínicas se usará un bombillo Led Scene Switch, Anexo 10.1, de la marca Philips.

Se ambientó un espacio donde el usuario por medio del Arduino UNO conectado al bombillo con un Modulo Relé (HW-383) Figura 5-5, cambiara por medio de los comandos de voz la intensidad de la luz, en el que se tiene 3 ambientes disponibles. También en la Figura 5-6 se puede visualizar la conexión que tiene el Arduino Uno usando el Pin 13 a este módulo de Relé con el Bombillo.

5.3. Sistema completo

El sistema completo se resume en tres conexiones principales: El computador donde se ejecuta todo el código creado y donde se recibe los comandos de voz por medio de su micrófono; la placa de Arduino que obedece al resultado obtenido por el código en Matlab; y la etapa de control que será manejada por la placa de Arduino. La siguiente Figura 5-7 hace ilustración del sistema completo y en funcionamiento.

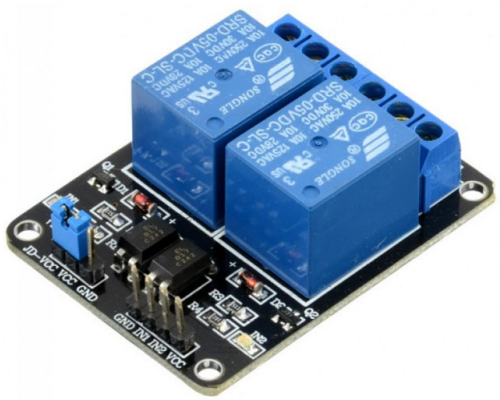

Figura 5-5: Modulo Relé HW-383. Tomado de 5V Dual-Channel Relay Module HW-383 [34].

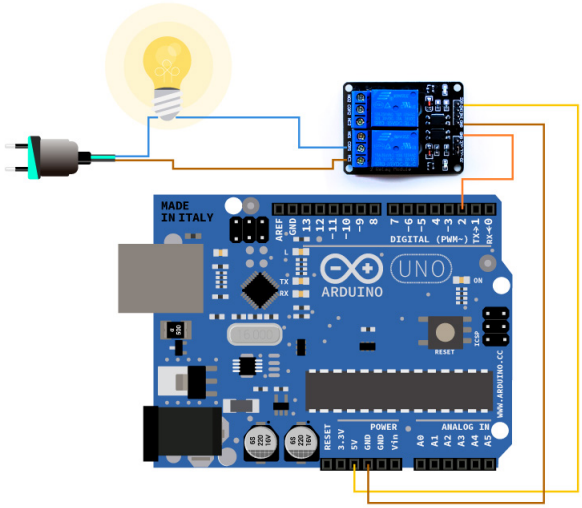

Figura 5-6: Conexión Arduino Uno al Modulo Relé. Tomado de Utiliza un relé con tu placa Arduino. Alberto Valero [35]

También la Figura 5-8 muestra un diagrama de flujo del sistema completo, pero más simplificado que será usado en el Manual Instructivo.

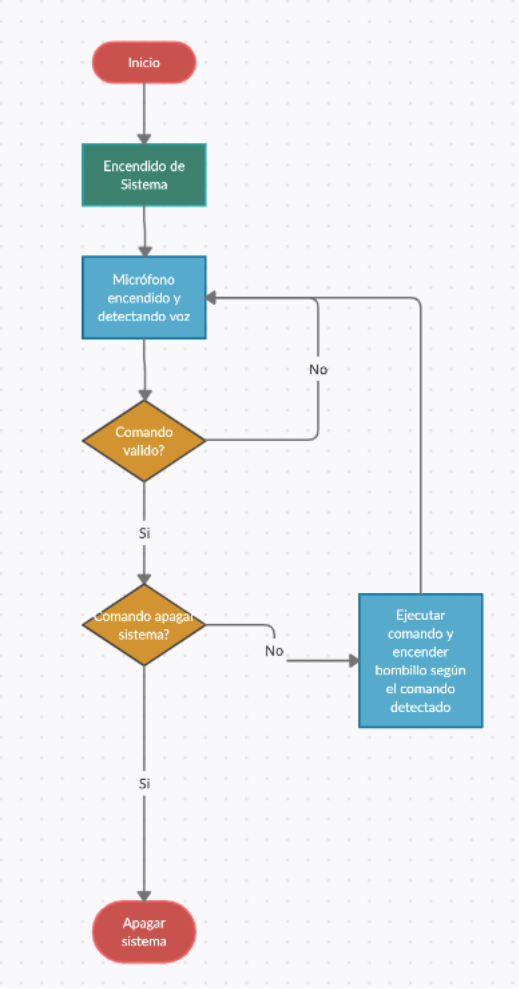

Figura 5-7: Diagrama de flujo para el sistema completo. Elaboración propia.

Finalmente en las Figuras 5-9 y 5-10 muestran la conexión entre el Arduino Uno con el módulo de Relé y una foto tomada del sistema completo y en funcionamiento:

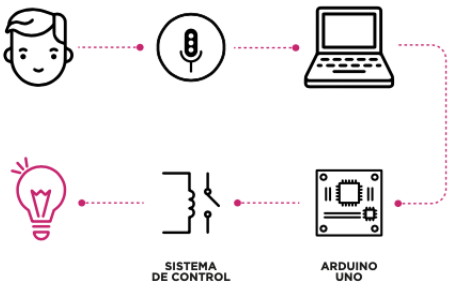

Figura 5-8: Diagrama de flujo simplificado para el sistema completo. Elaboración propia.

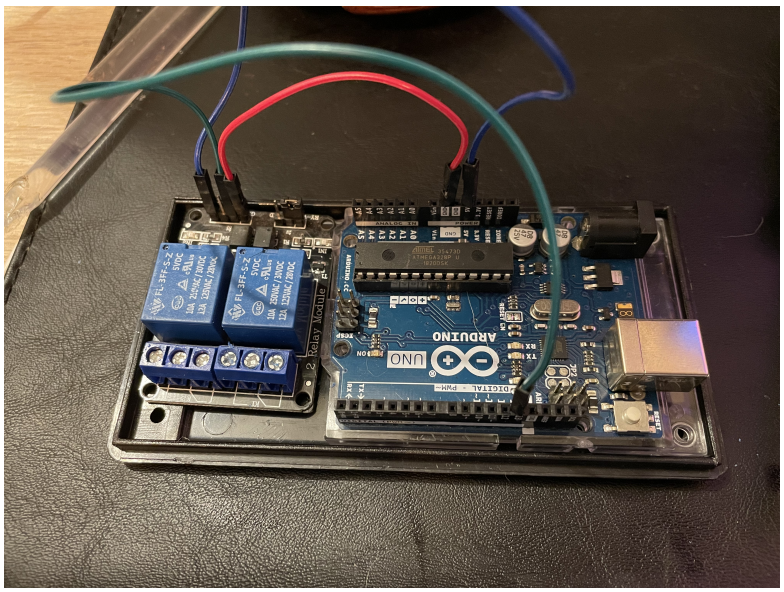

Figura 5-9: Conexión Arduino Uno - Módulo Relé. Elaboración propia.

5.4. Prueba experimental

Se dispuso de un adulto familiar de sexo femenino y de avanzada edad (90 años), con serias dificultades en su capacidad motriz, pero con suficientes niveles de entendimiento y lucidez mental para hacer una prueba experimental del sistema completo. Antes de hacer la simulación se le dieron indicaciones de comó pronunciar claramente cada comando de voz.

Figura 5-10: Conexión final del sistema completo: Computador - Arduino Uno - Módulo Relé - Bombillo. Elaboración propia.

Capítulo 6

Evaluación de las condiciones de funcionamiento

6.1. Fonemas

Uno de los grandes retos para cualquier sistema de reconocimiento de voz, es el estudio y análisis de los fonemas. La voz humana en su mayoría tiene una frecuencia menor a 10.000 Hz y su sonido cambia debido al contexto-coarticulación, el habla más coloquial y la velocidad con que se habla, todas estas hacen variar el sonido, lo que representa factores para tener en cuenta cuando se crea la matemática para el sistema de reconocimiento de comandos de voz [36]. Un novedad presentada es el llamado "borrado" de fonemas en ciertas palabras, como la palabra "Templado", que tiene una "m" y esta letra como la "n" también, representan un sonido muy bajo.

6.2. Contaminación auditiva

Los niveles sonoros de la contaminación auditiva también representan un reto para cualquier sistema de reconocimiento de voz y ciertamente en la noche hay un menor nivel que el que hay en el día. Los mayores factores que disminuyen el silencio en la ciudad o una área urbana es el tránsito de vehículos, industrias cercanas, y obras en construcción o reparación. El nivel promedio de decibelios (dB) en una casa urbana, oscila entre 40 a 50 dB, valores

propios de una habitación en silencio [37]. Para hacer las pruebas realizadas en el siguiente capítulo, se encontró un mejor resultado al hacerse de noche, por lo que para la creación de base de datos de todos los comandos, se optó por hacer en horas nocturnas.

Capítulo 7

Resultados

7.1. Comandos

Hay tres tipos de resultados para los comando de voz:“Falso", donde el resultado no es reconocido para ninguno de los 4 comandos de referencia en el sistema; el “Falso positivo", que es cuando la palabra pronunciada es reconocida por el sistema pero como otro comando, y por ultimo el “Acierto”, que es cuando el comando es correctamente reconocido con la base de datos. Como la obtención de los datos se verían afectados por el ruido del ambiente, se optó por realizar las pruebas en horario nocturno ,donde hay menor riesgo de interferencia. En la siguiente Figura 7-1, se gráfico la tasa de reconocimiento de 50 pruebas hechas para cada comando de voz:

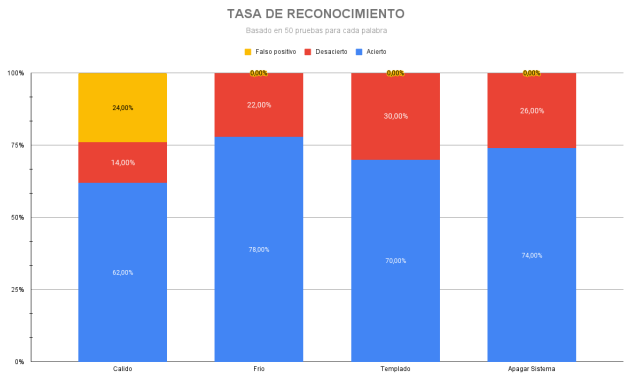

Figura 7-1: Gráfica para tasa de reconocimiento para los 4 comandos de voz basado en 50 pruebas. Elaboración propia.

7.2. Tratamiento de resultados de reconocimiento

Las tasas de reconocimiento hicieron saber qué tan acertado es el sistema con cada comando. El resultado del reconocimiento de voz se enfoca mas en qué tan efectivo es el sistema, analizando la voz en un lugar de poco ruido, recibiendo en tiempo real los datos para que este accione la etapa de control y cambiar al antojo del usuario la luz que le rodea en este lugar. La palabra "Cálido" será entre los comandos la que en porcentaje tiene más casos de falsos positivos; mientras que la palabra "Templado", en porcentaje tiene más probabilidad de tener casos de desaciertos; y los comandos "Frío" y "Apagar sistema" son los que presentan un porcentaje de acierto mayor.

Los resultados de cada comando se presentan en las siguientes tres Figuras 7-2, 7-3 y 7-4

Figura 7-2: Sistema en funcionamiento, comando "Cálido" como el comando detectado y bombillo iluminando en este nivel. Elaboración propia.

Figura 7-3: Sistema en funcionamiento, comando “Frío” como el comando detectado y bombillo iluminando en este nivel. Elaboración propia.

Figura 7-4: Sistema en funcionamiento, comando “Templado” como el comando detectado y bombillo iluminando en este nivel. Elaboración propia.

7.3. Prueba experimental

Los resultados obtenidos en la Figura 7-1 muestran que todos los comandos tuvieron un porcentaje de acierto mayor al 60 %, en especifico: Cálido obtuvo 62 %, Frío 78 %, Templado 70 % y Apagar Sistema 74 %. Por lo que sistema cuenta con un rango promedio de acierto mayor al 70 %.

El diseño experimental para la prueba con el adulto mayor contó con 4 opciones de palabras, que son los cuatro comandos de voz que se pusieron aprueba.Para ver su ejecución se hizo la grabación del experimento. La Tabla 7.1 muestra el diseño experimental hecho para esta prueba. Como variable independiente se tomo el comando de voz emitido: Cálido, Frío, Templado ó Apagar Sistema, que el usuario decide emitir y para ver qué efecto tiene sobre la variable dependiente. Como variable dependiente se tomo el comando de voz detectado por el sistema: Acierto, Desacierto ó Falso Positivo, que cambiara el nivel lumínico del cuarto. Como variables controladas: Hora de prueba experimental (Día - cuando hay mas ruido ambiental, ó de Noche cuando hay menos ruido en el ambiente), el volumen para el comando de voz emitido, la claridad y correcta pronunciación del comando, el intervalo de tiempo usado para pronunciar el comando de voz.

Diseño Experimental ¿Que tan acertado fue el reconocimiento de voz en 20 pruebas para cada comando?	
Variable independiente	Comando de voz emitido: Cálido, Frío, Templado ó Apagar Sistema
Variable dependiente	El Comando de Voz detectado (Acierto, Desacierto ó Falso Positivo) que cambiara el nivel lumínico del cuarto
Variables controladas	- Hora de prueba experimental (Día - cuando hay mas ruido ambiental, ó de Noche cuando hay menos ruido en el ambiente) - Volumen para el comando de voz. - Claridad y correcta pronunciación del comando - Intervalo de tiempo usado para pronunciar el comando de voz

Tabla 7.1: Tabla de Diseño Experimental y sus variables para la prueba experimental con adulto mayor. Elaboración propia.

Luego de las pruebas, se obtuvieron los siguientes resultados de la Tabla 7.2, y se gráfico

la tasa de reconocimiento en la Figura 7-5:

Comando	Cantidad de Pruebas	Acierto	Desacierto	Falso Positivo
Calido	20	11	6	3
Frio	20	14	3	3
Templado	20	11	9	0
Apagar Sistema	20	8	9	3

Tabla 7.2: Tabla de resultados de prueba experimental para los 4 comandos de voz basado en 20 pruebas. Elaboración propia.

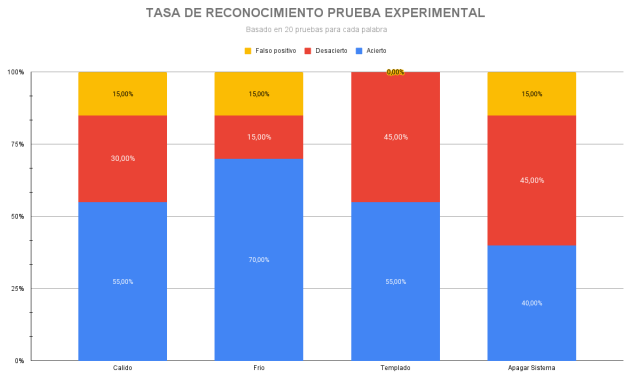

Figura 7-5: Gráfica para tasa de reconocimiento para los 4 comandos de voz basado en 20 pruebas. Elaboración propia.

Esta prueba experimental tomo alrededor de una hora y media para ser realizada, también se debe tener en cuenta que se hizo a las 2 de la tarde,por lo que el factor de ruido ambiental afectaría la cantidad de aciertos a comparación de las pruebas hechas de la Figura 7-1 que se realizaron en la noche. Fue necesario realizar múltiples correcciones y una guía acompañada al adulto mayor para hacer que los comandos de voz fuera pronunciado correctamente,, claro y no muy fuerte o muy bajo en el lapso de tiempo que el sistema esta en modo de captura de audio. En los resultados se obtuvo que el sistema contó con un rango promedio de acierto cercano al 60 %. Entre los comandos “Cálido”, “Templado”, y “Frío” tuvieron un promedio mayor del 60 % de acierto; mientras que el comando “Apagar sistema” fue el que presentó un 40 % de aciertos.

Capítulo 8

Conclusiones del trabajo y futuras líneas de investigación

En este último capítulo se exponen las principales conclusiones obtenidas durante el desarrollo de la tesis y el direccionamiento de futuras líneas de investigación.

8.1. Conclusiones y recomendaciones de la Tesis

- Los pasos necesarios para el desarrollo de este sistema de reconocimiento de voz se basaron en una matemática de almacenamiento de datos por arreglos, los cuales luego de ser filtrados, recortados, segmentados y extraídas sus características, se ejecuta una comparación de sus coeficientes por diferentes métodos que hicieron posible un sistema, estas son: la autocorrelación, reflexión, LPC y correlación de Pearson. Un sistema que use un Hardware independiente del PC es una futura oportunidad de aplicación del sistema para mostrar.

- Ya que los fonemas de los usuarios y la hora del día, son factores determinantes, se concluyó que para un alto porcentaje de aciertos (buen funcionamiento) de reconocimiento de voz, es necesario implementar un algoritmo que se adecúe al nivel de ruido, mejorando el enfoque del sistema en la detección de los comandos, y de los diferentes intervalos de tiempo para la pronunciación de la palabra, independientemente de lo

cortos o lo largos que sean; mejorando así la aplicación de la detección de sus extremos.

- Las tres(3) diferentes atmósferas lumínicas permiten al usuario ambientes como cálido, neutro y frío, sin embargo, actualmente existen diversos tipos de bombillos inteligentes en el mercado que hace posible más tipos de proyectos que combinen funciones del reconocimiento de voz con el control de lugares dependiendo de su ambiente lumínico.

- El Manual Instructivo (Brochure) permite una rápida y simple explicación del funcionamiento y forma de uso del sistema completo, esto para acceder brevemente a cualquier atmósfera lumínica deseada.

- Cualquier persona con o sin discapacidad motriz puede tener acceso al sistema. Quien padece una limitación o una discapacidad motriz tiene la facilidad de no hacer esfuerzo alguno, sólo usando la voz, al contar con el sistema de reconocimiento de comando de voz, que le permite controlar la luminiscencia de una habitación. Después de realizar mínimos ajustes al comando de voz,con la orientación del estudiante de ingeniería, y mediante pocos intentos, el sujeto de la prueba experimental, una adulta de tercera edad, y sexo femenino, logró que el sistema respondiera eficazmente a los objetivos trazados, de controlar el encendido y apagado de la luz en la habitación.

8.2. Futuras líneas de investigación

Como línea de investigación se pueden proyectar diferentes métodos para usar un sistema de reconocimiento de voz en otras tarjetas y placas programables. Sobre lo anterior, para ser completamente integrado o embebido el sistema en una tarjeta como el Arduino, que tiene una memoria Flash de máximo 4 MB, se requeriría una memoria mínima de 33 MB para la matemática implementada en MATLAB, aparte del espacio de memoria para cada comando de audio nuevo que dicta el usuario, el cual ronda en un promedio de 1 MB incluyendo los archivo de texto para guardar sus coeficientes. Para disminuir el tamaño de memoria que ocupa este código y la base de datos, ciertamente la mejor manera es reducir la frecuencia de muestreo, pero al disminuir esta variable, se reduce dramáticamente la probabilidad de aciertos para todos los comandos de audio en todo el sistema, y como la frecuencia actual

tiene un promedio de 70 % haciendo un sistema muy poco factible.

Capítulo 9

Manual Instructivo

Para crear el Manual Instructivo ("Brochure") se utilizó el programa Adobe Illustrator, el cual es un editor de gráficos vectoriales, desarrollado y comercializado por Adobe Systems [38]. Los puntos desarrollados en el Manual muestran un resumen del funcionamiento, la interpretación de los resultados, y finalmente, las conclusiones y aplicaciones del sistema de reconocimiento de comandos de voz. En la siguiente Figura 9-1 se muestra una imagen del Manual Instructivo.

Figura 9-1: Imagen completa del manual instructivo. Elaboración propia.

Capítulo 10

Anexos

10.1. Bombillo

Bombillo Led Scene Switch, de la marca Philips:

Figura 10-1: Bombillo Led Scene Switch
. Tomado de [33].

10.2. Matlab - Código simple para grabación de comando de voz y ser guardado en archivo de texto como base de datos

```
recorder = audiorecorder(8000,16,1)%8k como frecuencia, 16 como cantidad de bits,
y 1 como canal mono
recordblocking(recorder,2) % cantidad de tiempo a grabar: 2 segundos
audioarray = getaudiodata(recorder)
play(recorder)
plot(audioarray)
fileID =  fopen('test.txt','w');%se crear archivo de texto con formato de escritura
fprintf(fileID, 'test\n\n');%se determina que cada dato se organizar en matriz 1
columna y N filas
fprintf(fileID, '%d', audioarray);%se escribe el comando de voz en el archivo de texto
fclose(fileID);
```

10.3. Matlab - Código para la captura de comandos de voz y su almacenaje en la base de datos

Las funciones son las mismas que del código principal y serán anexadas en el siguiente anexo.

```
clc
clear
close all
clear all


t_end = 3;
Fsc = 50000;
nBits = 16;
Nchannel = 1;
```

```
Gain = 1024;
nm = Fsc*30e-3;
Noise_level = 1500;
nbits = 2^14;
Offset = 2047;


recObj = audiorecorder(Fsc, nBits, Nchannel);%comando para declarar la
grabacion de voz
recObj.StartFcn = 'disp('' Start speaking '')';%mensaje para a grabar voz


input('Press to record voice');%input para empezar grabar voz
recordblocking(recObj, t_end);%comando para grabacion
disp('End of recording. Playing back ...');%mensaje para terminar grabacion
Audio = Gain*(getaudiodata(recObj)) + Offset;%comando para terminar
 grabacion
fileID =  fopen('Prueba.txt','w');% comando para crear archivo de texto
de formato de escritura
fprintf(fileID, '%d\n', Audio);%se sobre escribe en el archivo txt los datos
 de audio obtenidos, se usa el %d para que cada dato ocupe una fila
fclose(fileID);%se cierrar la escritura en el archivo de texto
%play(recObj)% reproduccion del audio obtendio
%plot(Audio)%graficacion del audio
%input('continuar')
load Prueba.txt %cargar el archivo de texto

jj = 0;
S = zeros(1,Fsc*t_end); %Un vector que almacena la señal de voz de entrada
for p = 1:1:Fsc*t_end
```

```
    S(p) = Prueba(p,:);%creacion vector que tendra los archivos de audio
end

N = length(S);

%[Ns, Ssr, Eav] = Eliminacion(S, N, nm);%funcion de elminiacion

[Ssr,longPal] = detectorExtremos(S);

input('Reproducir palabra de limitada');%input para empezar grabar voz


Ns = length(Ssr);
soundsc(Ssr,Fsc);



%regla 3 con tiempo de grabación
%Fs = 50000; %Frecuencia de muestreo a utilizar (voz en este caso).
To = N/Fsc; %Grabación de tiempo-> Señal de entrada
Tg = Ns*To/N; %Tiempo de grabación-> corte de señal

[Spp] = Pre_enfasis(Ssr,Ns);

scrsz = get(0,'ScreenSize');
figure('Name','Señal de Voz completa y Recortada','Position',
[0 420 scrsz(3) scrsz(4)/1.205])
subplot(3,1,1),plot(S),title('Señal de Voz'),xlabel('Numero de muestras'),
ylabel('Amplitud de las muestras'),grid
subplot(3,1,2),plot(Ssr),title('Señal de Voz Detección Inicio a Fin'),
```

```
xlabel('Numero de muestras'),ylabel('Amplitud de las muestras'),grid
subplot(3,1,3),plot(Spp),title('Señal de Voz filtrada por preenfasis'),
xlabel('Numero de muestras'),ylabel('Amplitud de las muestras'),grid

x = fix(Ns/nm);
y = 15;
a = zeros(2*x,y);
aj = zeros(1,y);
aj1 = zeros(1,y);

%aj1=0;
for j=1:x

 [Sw,Swc,Seg] = Segmentacion(j,Spp,nm, nbits);
 [rxx0, rxx1, Swn, Swcn, auto] = Autocorrelacion(nm, Sw, Swc);
 m = length(rxx0);%será la longitud del vector que contiene
 coeficiente de autocorrelación de ventana normal
 b = (2*j)-1;
 v = (2*j);

 rxx0b = rxx0(m:-1:1);
 a(b,1) = fix(rxx0(2)/(rxx0(1)/8));
 Ki = fix(rxx0(2)/(rxx0(1)/8));

 f1=4096;

 rxx1b = rxx1(m:-1:1);
 a(v,1) = fix(rxx1(2)/(rxx1(1)/8));
 Ki1 = fix(rxx1(2)/(rxx1(1)/8));
 for i = 2:m-1
```

```
     Ni = rxx0(i+1)-(rxx0b(i))'*Ki;
     Di = rxx0(1)-(rxx0(i))'*Ki;
     Ki = ceil(Ni/floor(Di/f1));

     Ni1 = rxx1(i+1)-(rxx1b(i))'*Ki1;
     Di1 = rxx1(1)-(rxx1(i))'*Ki1;
     Ki1 = ceil(Ni1/floor(Di1/f1));
     for h = 1:i-1
         aaj = floor((Ki*a(b,i-h))/f1);
         aj(h) = (a(b,h))-aaj;
         aj(i) = Ki;
         aaj1 = floor((Ki1*a(v,i-h))/f1);
         aj1(h) = (a(v,h))-aaj1;
         aj1(i) = Ki1;
     end
     a(b,:) = aj;
     a(v,:) = aj1;
 end
end

l=0;

lpc=zeros(1,2*x*y);
for i = 1:1:2*x
    for k = 1:1:y
        l=l+1;
        lpc(l) = a(i,k);
    end
end
```

```
[m,n] = size(a);

%Codigo para guadrar los coeficientes de LPC en un archivo de TXT

fileID =  fopen('LPC_Prueba.txt','w');% comando para crear archivo
 de texto de formato de escritura
fprintf(fileID, '%d\n', a);%se sobre escribe en el archivo txt los datos
 de audio obtenidos, se usa el %d para que cada dato ocupe una fila
fclose(fileID);

fileID =  fopen('Filas_Prueba.txt','w');% comando para crear archivo
de texto de formato de escritura
fprintf(fileID, '%d', m);%se sobre escribe en el archivo txt los datos de
audio obtenidos, se usa el %d para que cada dato ocupe una fila
fclose(fileID);


scrsz = get(0,'ScreenSize');
figure('Name','Coeficientes','Position',[0 420 scrsz(3) scrsz(4)/1.205])
plot(lpc,'-o','MarkerFaceColor','r'),xlim([0 2*x*y]),title('Coeficientes
 LPC Palabra Apagar Sistema'),xlabel('Número de Segmentos'),
ylabel('Amplitud de los Coeficientes'),grid


% figure('Name','LPC')
% plot(lpc,'-o','MarkerFaceColor','r'),xlim([0 15]),title('Coeficientes LPC'),
xlabel('Número de Coeficientes'),ylabel('Amplitud de los Coeficientes'),grid
```

10.4. Matlab - Código final del SISTEMA DE RECONOCIMIENTO DE COMANDOS DE VOZ COMO INTERFAZ DE UNA HABITACIÓN CON TRES ATMÓSFERAS LUMÍNICAS

```
clc
close all

x=instrhwinfo('serial');%se declara una variable para leer las entrada
seriales al computador
x.SerialPorts%se visualiza todas las entradas seriales
a = arduino('/dev/cu.usbmodem141401');%se crea la variable para usar
el paquete de arduino ya instalado

PinLed='D13';%se declara la variable para el pin 13 del arduino que
estara conectado al modulo relé
b = 0;%indicador en que estado esta el bombillo
writeDigitalPin(a,PinLed,0);%se inicializa el pin en bajo para no
encender el bombillo

n = 10;
while n > 1
    disp('Programa de reconocimiento de voz')

    [aaC , aaF , aaT , aaA] = Audio();%funcion para traer toda la base
     de datos para todos los comandos de audio
    [aa] = CapturaDeAudio();%se captura el nuevo audio y se obtiene
    los coefeciente de LPC
    [comando, x1] = Comparacion(aaC , aaF , aaT , aaA , aa);%funcion de
```

```
comparacion del comando detectado
disp (comando)%se visualiza el comando detectado

pause(3)
%si fue detectado el comando calido
if x1 == 1 && b == 1
    disp('El sistema ya esta encendido y esta en nivel Calido')

elseif x1 == 1 && b == 0
    writeDigitalPin(a,PinLed,1);
    b = 1;

elseif x1 == 1 && b == 2
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);
    pause(1);
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);

    b = 1;

elseif x1 == 1 && b == 3
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);

    b = 1;
```

```
%si fue detectado el comando templado
elseif x1 == 2 && b == 0
    writeDigitalPin(a,PinLed,1);
    pause(1);
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);
    b = 2;

elseif x1 == 2 && b == 1
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);
    b = 2;

elseif x1 == 2 && b == 2
    disp('El sistema ya esta encendido y esta en nivel Templado')

elseif x1 == 2 && b == 3
    writeDigitalPin(a,PinLed,0)
    pause(1);
    writeDigitalPin(a,PinLed,1);
    pause(1);
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);

    b = 2;

%si fue detectado el comando frio
```

```
elseif x1 == 3 && b == 0
    writeDigitalPin(a,PinLed,1);
    pause(1);
    writeDigitalPin(a,PinLed,0)
    pause(1);
    writeDigitalPin(a,PinLed,1);
    pause(1);
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);

    b = 3;

 elseif x1 == 3 && b == 1
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);
    pause(1);
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);
    b = 3;

 elseif x1 == 3 && b == 2
    writeDigitalPin(a,PinLed,0);
    pause(1);
    writeDigitalPin(a,PinLed,1);
    b = 3;
```

```
 elseif x1 == 3 && b == 3
    disp('El sistema ya esta encendido y esta en nivel Frio')

%si fue detectado el comando apagar sistema
elseif x1 == 4
    writeDigitalPin(a,PinLed,0);
    b = 0;



end

end
```

Función Audio: Para cargar base de datos de los 5 comandos de audio

```
function [aaC , aaF , aaT, aaA] = Audio()
clc
columnas = 15;

        fileID_a = fopen('LPC_Calido.txt'); %cargar el archivo de texto de LPC
        filas = load ('Filas_Calido.txt');
        formatSpec = '%f';
        sizeA = [filas columnas];
        aaC = fscanf(fileID_a,formatSpec,sizeA);
        fclose(fileID_a);

        fileID_a = fopen('LPC_Frio.txt'); %cargar el archivo de texto de LPC
        filas = load ('Filas_Frio.txt');
        formatSpec = '%f';
        sizeA = [filas columnas];
```

```
        aaF = fscanf(fileID_a,formatSpec,sizeA);
        fclose(fileID_a);

        fileID_a = fopen('LPC_Templado.txt'); %cargar el archivo de texto
     de LPC
        filas = load ('Filas_Templado.txt');
        formatSpec = '%f';
        sizeA = [filas columnas];
        aaT = fscanf(fileID_a,formatSpec,sizeA);
        fclose(fileID_a);

        fileID_a = fopen('LPC_ApagarSistema.txt'); %cargar el archivo de
    texto de LPC
        filas = load ('Filas_ApagarSistema.txt');
        formatSpec = '%f';
        sizeA = [filas columnas];
        aaA = fscanf(fileID_a,formatSpec,sizeA);
        fclose(fileID_a);
end
```

Función Captura de audio: Para capturar comando de audio dictado por el usuario

```
function [aa] = CapturaDeAudio()
clc

t_end = 3;
Fsc = 50000;
nBits = 24;
Nchannel = 1;
Gain = 1024;
```

```
nm = Fsc*30e-3;
nbits = 2^14;
Offset = 2047;


recObj = audiorecorder(Fsc, nBits, Nchannel);%comando para declarar
la grabacion de voz
recObj.StartFcn = 'disp('' Start speaking '')';%mensaje para a grabar voz


input('Press to record voice');%input para empezar grabar voz
recordblocking(recObj, t_end);%comando para grabacion

%para hacer en tiempo real el sistema, poner en comentario el anterior comando, input

disp('End of recording. Playing back ...');%mensaje para terminar grabacion
y = Gain*(getaudiodata(recObj)) + Offset;%comando para terminar
 grabacion
fileID =  fopen('Audio.txt','w');% comando para crear archivo de texto
 de formato de escritura
fprintf(fileID, '%d\n', y);%se sobre escribe en el archivo txt los datos de
 audio obtenidos, se usa el %d para que cada dato ocupe una fila
fclose(fileID);%se cierrar la escritura en el archivo de texto
%play(recObj)% reproduccion del audio obtendio

load Audio.txt %cargar el archivo de texto

S = zeros(1,Fsc*t_end); %Vector donde se guarda la Señal de Voz de entrada
for p = 1:1:Fsc*t_end
    S(p) = Prueba(p,:);%creacion vector que tendra los archivos de audio
```

```
end

N = length(S);

%[Ns, Ssr, Eav] = Eliminacion(S, N, nm);%funcion de elminiacion

[Ssr, longPal] = detectorExtremos(S);

Ns = length(Ssr);

%soundsc(Ssr,Fsc);


%regla 3 con tiempo de grabación
To = N/Fsc; %Grabación de tiempo-> Señal de entrada
Tg = Ns*To/N; %Tiempo de grabación-> corte de señal

[Spp] = Pre_enfasis(Ssr,Ns);

x = fix(Ns/nm);
y = 15;
a = zeros(2*x,y);
aj = zeros(1,y);
aj1 = zeros(1,y);

%aj1=0;
for j=1:x

 [Sw,Swc,Seg] = Segmentacion(j,Spp,nm, nbits);
 [rxx0, rxx1, Swn, Swcn, auto] = Autocorrelacion(nm, Sw, Swc);
```

```
m = length(rxx0);%será la longitud del vector que contiene
coeficiente de autocorrelación de ventana normal

b = (2*j)-1;
v = (2*j);

rxx0b = rxx0(m:-1:1);
a(b,1) = fix(rxx0(2)/(rxx0(1)/8));
Ki = fix(rxx0(2)/(rxx0(1)/8));

f1=4096;

rxx1b = rxx1(m:-1:1);
a(v,1) = fix(rxx1(2)/(rxx1(1)/8));
Ki1 = fix(rxx1(2)/(rxx1(1)/8));
for i = 2:m-1
    Ni = rxx0(i+1)-(rxx0b(i))'*Ki;
    Di = rxx0(1)-(rxx0(i))'*Ki;
    Ki = ceil(Ni/floor(Di/f1));

    Ni1 = rxx1(i+1)-(rxx1b(i))'*Ki1;
    Di1 = rxx1(1)-(rxx1(i))'*Ki1;
    Ki1 = ceil(Ni1/floor(Di1/f1));
    for h = 1:i-1
        aaj = floor((Ki*a(b,i-h))/f1);
        aj(h) = (a(b,h))-aaj;
        aj(i) = Ki;
        aaj1 = floor((Ki1*a(v,i-h))/f1);
        aj1(h) = (a(v,h))-aaj1;
        aj1(i) = Ki1;
```

```
     end
     a(b,:) = aj;
     a(v,:) = aj1;
 end
end

l=0;

lpc=zeros(1,2*x*y);
for i = 1:1:2*x
    for k = 1:1:y
        l=l+1;
        lpc(l) = a(i,k);
    end
end

aa = a;

end
```

Funciones llamadas dentro de la función "Captura de audio"

- **Función "detectorExtremos" para la detección de extremos y recortar el audio.**

```
function [Ssr,longPal] = detectorExtremos(S)
[~, columnas]=size(S);
i=floor(columnas/1500);%cuantas veces hay que calcular
umbral (numero de las tramas)
m=2; %el coeficiente de índice inicial comienza con 2 porque
necesitamos datos (x-1)
n=1502;% relación de estadísticas final
```

```
%esto contara las ventanas consecutivas sin info. para detectar
final pronunciación.

longPal=1;
t=0;

for j=1:1:i %esto dividira la señal
    umbral=0;
        for k=m:1:n-2 %bucle iterar sobre muestras n a m S calcular
    el umbral del marco
            umbral=umbral+abs(S(k)*abs(S(k))-S(k-1)*abs(S(k-1)));
        end

            %evolucionUmbral(n)=umbral; %el valor se almacena
        cada 1500 entradas

            %=DETECTOR INICIO=> si cruza el umbral, comenzará
 a guardar fotogramas


        if (umbral>5e+06) %comprobamos si se detecta inicio, si supera
            Umbral Inicio=1.25
            t=0;
            longPal=longPal+1500;

            if n>150000
                n = 150000;
                Ssr(longPal-1500:longPal-2)=S(m:n);
                break;
            end
```

```
            Ssr(longPal-1500:longPal)=S(m:n); %guardar trama


        end
        %=DETECTOR FIN=> deteccion de final si paso 15
        sin info (t=15)
        if (umbral<5e+06) %se analiza si incremento t,
            if (t<10 && longPal~=1)
            t=t+1;
%                   if t == 10
%                       break;
%                   end
            end
            %longPal=longPal+1500;
            %palabradelimitada(longPal-1500:longPal-2)=S(m:n-2);
        %almacenamos tramas si t<15
        end
   m=m+1500;
   n=n+1500;
end
end
```

- **Función "Pre_enfasis" para la detección el filtro del pre énfasis y obtención de un audio mas claro**

```
function [Spp] = Pre_enfasis(Ssr, Ns)
Spp = zeros(1,Ns); % Vector
Sn = 0;
Sn1 = 0;
%o = 0;
for k = 1:1:Ns
    Sn = Ssr(k);
```

```
    Spp(k) = Sn-0.9609375*Sn1;
    Sn1 = Ssr(k);
end
Spp(1) = 0;
end
```

- **Función "Segmentacion" para la segmentación con metodo de ventana de Hamming y invertida de Hamming.**

```
function [Sw, Swc, Seg] = Segmentacion(j, Spp, nm, nbits)

Seg = zeros(1,nm); % muestras de cada seg.
l = (j*nm+1)-nm; % el numero de la muestra para iniciar los seg.
for i = l:(j*nm)
Seg((i+1)-l) = Spp(i); %se establece que Seg (i-l+1) comience en 1
%*Normally-Windowed*
Sw = floor((Seg.*round(nbits.*hamming(nm)'))/nbits); % Enventanado
(Hamming) para cada Segmento
%*Complementarily-Windowed*
Swc = floor((Seg.*round(nbits.*(1.08-hamming(nm)')))/nbits);
% Enventanado (Hamming) Complemetaria para cada Segmento

%temp = Seg.*hamming(1500)';
end

end
```

- **Función "Autocorrelacion" para la autocorrelación de los vectores con de la función anteriror.**

```
function [ rxx0, rxx1, Swn, Swcn, auto] = Autocorrelacion( nm, Sw, Swc )
%rx(i) = sum,n=0,N-1[S(n).S(n-i)]
```

```
ncoef = 16;
Swn = zeros(1,nm+(ncoef-1));%vector de 1515
Swcn = zeros(1,nm+(ncoef-1));%vector de 1515
auto = zeros(1, nm);
% Swn(1:12)= 9;
% Swcn(1:12) = 9;
    for k = ncoef:1:nm+(ncoef-1) %16 hasta 1515
        Swn(k) = Sw(k-(ncoef-1)); %1 hasta 1500, haciendo zeros los
    primeros 15
        Swcn(k) = Swc(k-(ncoef-1));%1 hasta 1500,  haciendo zeros los
        primeros 15
        %Se crean segmentos ficticios para calcular la %autocorrelación
        con todas las muestras
    end
rxx0 = zeros(1,ncoef); % autocorrelacion
    rxx1 = zeros(1,ncoef); %autocorrelacion
    for i=1:1:ncoef %hasta 16
        rr = 0; %Variable temporal que almacena cada elemento de
    autocorrelación
        rrc = 0; %Variable temporal que almacena cada elemento de
        autocorrelación
        for n = ncoef:1:nm+(ncoef-1)%16 hasta 1515
            mult = (Swn(n)*Swn(n-i+1)); % desde la 16,17,18 hasta 1515
        por 16,15,14,13 hasta 1
            rr = rr + mult;%acumula el valos de mult

            mult1 = (Swcn(n)*Swcn(n-i+1));
            rrc = rrc + mult1;%acumula el valos de mult1

            if i==1 %el primer valor de auto sera el primer valor mult
```

```
            auto(n-(ncoef-1)) = rr;
            end
        end
        rxx0(i) = (rr/256); % valor acumulado de rr divido sobre 256
        para truncar los 8 bits mas bajos
        rxx1(i) = (rrc/256);
    end
end
```

Función "Comparacion" para la comparación de los coeficientes de LPC del comando capturado con la base de datos.

```
function [comando, x1] = Comparacion(aaC , aaF , aaT  , aaA , aa)
clc
close all
[a1,~] = size(aa);


[porcenC] = PearsonC(aa,aaC);

[porcenF] = PearsonF(aa,aaF);

[porcenT] = PearsonT(aa,aaT);

%[porcenE] = PearsonE(aa,aaE);

[porcenA] = PearsonA(aa,aaA);

if porcenC > 80 && a1>=29 && a1<=35
    comando = 'Usted dijo Calido';
```

```
    x1 = 1;

elseif porcenF > 80 && a1>=21 && a1<=27
    comando = 'Usted dijo Frio';
    x1 = 3;

elseif porcenT > 80 && a1>=39 && a1<=45
    comando = 'Usted dijo Templado';
    x1 = 2;

elseif porcenA > 80 && a1>=69 && a1<=75
    comando = 'Usted dijo ApagarSistema';
    x1 = 4;

else
    comando = 'Comando no Valido';
    x1 = 5;

end



end
```

Funciones llamadas dentro de la función "Comparacion"

- **Función "PearsonC", "PearsonF", "PearsonT", y "PearsonA", para la detección de extremos y recortar el audio.**

```
function [porcenC] = PearsonC(aa,aaC)

[m,n] = size(aa);
```

```
[m1,n1] = size(aaC);%referencia
nbits = 2^14;

y = 15;

bb = aaC;

if m1 > m
    for i=m+1:1:m1
        for k=1:1:n1
           aa(i,k) = 0;
        end
    end
end

Ccp = zeros(1,m1);%se usa el tamaño de filas del audio como
referencia, por tanto es el de aa
x1 = zeros(1,n);
y1 = zeros(1,n);

for i=1:m1
    for j=1:y
    x1(j) = bb(i,j);%cada fila de valores
    y1(j) = aa(i,j);%cada fila de valores
    end

xm1=(mean(x1)); %xm1=fix(mean(bb(i,:))); Hace el valor promedio
    de cada fila para cada arregloe aa y bb que se pasaron a x1 y y1
ym1=(mean(y1)); %ym1=fix(mean(aa(i,:))); Hace el valor promedio
de cada fila para cada arregloe aa y bb que se pasaron a x1 y y1
```

```
% Covarianza y Desviación Estándar
Sxy1 = 0;
vx1 = 0;
vy1 = 0;

    for k = 1:1:y
        xiXm = x1(k)-xm1;
        yiYm = y1(k)-ym1;
        Sxy1 =Sxy1+(floor(xiXm/y)*yiYm);%Ecuanción de Covarianza
        vx1 = vx1 + xiXm^2; %Ecuación de Varianza
        vy1 = vy1 + yiYm^2;
    end
dx1 = fix(sqrt(vx1)/(y-1)); %desviacionEstándar
dy1 = fix(sqrt(vy1));       %desviacionEstándar
dxdy1 = dx1*dy1;
Ccp(i) =fix((Sxy1)/(dxdy1/nbits));
    %for q = 1:1:y
    %    Xm(i,q)=floor((x1(q)-xm1)/y); %Resta para el cálculo de la
    covarianza entre elc omando end
    %end
D(i)=floor(sqrt(vx1)/(y-1));
end
c=0; %cantidaddecoeficientesquesuperanelumbraldeigualdad
for u = 1:m1
    if Ccp(u) >= 15291
        c=c+1;
    end
end
```

```
porcenC = c*100/(m1);
 %numero de Coeficientes de Pearson que sobrepasaron
 el umbral


end


function [porcenF] = PearsonF(aa,aaF)

[m,n] = size(aa);
[m1,n1] = size(aaF);%referencia
nbits = 2^14;
y = 15;

bb = aaF;

if m1 > m
    for i=m+1:1:m1
        for k=1:1:n1
           aa(i,k) = 0;
        end
    end
end

Ccp = zeros(1,m1);%se usa el tamaño de filas del audio
como referencia, por tanto es el de aa
x1 = zeros(1,n);
y1 = zeros(1,n);
```

```
for i=1:m1
    for j=1:y
    x1(j) = bb(i,j);%cada fila de valores
    y1(j) = aa(i,j);%cada fila de valores
    end

xm1=(mean(x1)); %xm1=fix(mean(bb(i,:))); Hace el valor promedio
     de cada fila para cada arregloe aa y bb que se pasaron a x1 y y1
ym1=(mean(y1)); %ym1=fix(mean(aa(i,:))); Hace el valor promedio
 de cada fila para cada arregloe aa y bb que se pasaron a x1 y y1

% Covarianza y Desviación Estándar
Sxy1 = 0;
vx1 = 0;
vy1 = 0;

    for k = 1:1:y
        xiXm = x1(k)-xm1;
        yiYm = y1(k)-ym1;
        Sxy1 =Sxy1+(floor(xiXm/y)*yiYm);%Covarianza
        vx1 = vx1 + xiXm^2; %Varianza
        vy1 = vy1 + yiYm^2;
    end
dx1 = fix(sqrt(vx1)/(y-1)); %DesviaciónEstándar
dy1 = fix(sqrt(vy1));       %DesviaciónEstándar
dxdy1 = dx1*dy1;
Ccp(i) =fix((Sxy1)/(dxdy1/nbits));
    %for q = 1:1:y
    %    Xm(i,q)=floor((x1(q)-xm1)/y); %Resta para el cálculo de
    la covarianza entre elc omando end
```

```
    %end
D(i)=floor(sqrt(vx1)/(y-1)); %Desviaciónestandar
end
c=0; %cantidaddecoeficientesquesuperanelumbraldeigualdad
for u = 1:m1
    if Ccp(u) >= 15291
        c=c+1;
    end
end

porcenF = c*100/(m1);
 %PorcentajedeCoeficientesdePearsonquesuperanelumbraldecomparación
% if c >= m1-1
%     comando = 'Usted dijo Frio';
% else
%     comando = 'Usted no dijo Frio';
% end

end

function [porcenT] = PearsonT(aa,aaT)

[m,n] = size(aa);
[m1,n1] = size(aaT);%referencia
nbits = 2^14;
y = 15;

bb = aaT;

if m1 > m
```

```
    for i=m+1:1:m1
        for k=1:1:n1
           aa(i,k) = 0;
        end
    end
end

Ccp = zeros(1,m1);%se usa el tamaño de filas del audio
como referencia, por tanto es el de aa
x1 = zeros(1,n);
y1 = zeros(1,n);

for i=1:m1
    for j=1:y
    x1(j) = bb(i,j);%cada fila de valores
    y1(j) = aa(i,j);%cada fila de valores
    end

xm1=(mean(x1)); %xm1=fix(mean(bb(i,:))); Hace el valor
    promedio de cada fila para cada arregloe aa y bb que se pasaron a x1 y y1
ym1=(mean(y1)); %ym1=fix(mean(aa(i,:))); Hace el valor
promedio de cada fila para cada arregloe aa y bb que se pasaron a x1 y y1

% Covarianza y Desviación Estándar
Sxy1 = 0;
vx1 = 0;
vy1 = 0;

    for k = 1:1:y
        xiXm = x1(k)-xm1;
```

```
        yiYm = y1(k)-ym1;
        Sxy1 =Sxy1+(floor(xiXm/y)*yiYm);%Covarianza
        vx1 = vx1 + xiXm^2; %Varianza
        vy1 = vy1 + yiYm^2;
    end
dx1 = fix(sqrt(vx1)/(y-1)); %DesviaciónEstándar
dy1 = fix(sqrt(vy1));       %DesviaciónEstándar
dxdy1 = dx1*dy1;
Ccp(i) =fix((Sxy1)/(dxdy1/nbits));
    %for q = 1:1:y
    %    Xm(i,q)=floor((x1(q)-xm1)/y); %Resta para el
    cálculo de la covarianza entre elc omando end
    %end
D(i)=floor(sqrt(vx1)/(y-1)); %Desviaciónestandar
end
c=0; %cantidaddecoeficientesquesuperanelumbraldeigualdad
for u = 1:m1
    if Ccp(u) >= 15291
        c=c+1;
    end
end

porcenT = c*100/(m1);
 %PorcentajedeCoeficientesdePearsonquesuperanelumbraldecomparación
% if c >= m1-1
%     comando = 'Usted dijo Templado';
% else
%     comando = 'Usted no dijo Templado';
% end
```

```
end

function [porcenA] = PearsonA(aa,aaA)

[m,n] = size(aa);
[m1,n1] = size(aaA);%referencia
nbits = 2^14;
y = 15;

bb = aaA;

if m1 > m
    for i=m+1:1:m1
        for k=1:1:n1
           aa(i,k) = 0;
        end
    end
end

Ccp = zeros(1,m1);%se usa el tamaño de filas del audio como
referencia, por tanto es el de aa
x1 = zeros(1,n);
y1 = zeros(1,n);

for i=1:m1
    for j=1:y
    x1(j) = bb(i,j);%cada fila de valores
    y1(j) = aa(i,j);%cada fila de valores
    end
```

```
xm1=(mean(x1)); %xm1=fix(mean(bb(i,:))); Hace el valor
    promedio de cada fila para cada arregloe aa y bb que se pasaron a x1 y y1
ym1=(mean(y1)); %ym1=fix(mean(aa(i,:))); Hace el valor
promedio de cada fila para cada arregloe aa y bb que se pasaron a x1 y y1

% Covarianza y Desviación Estándar
Sxy1 = 0;
vx1 = 0;
vy1 = 0;

    for k = 1:1:y
        xiXm = x1(k)-xm1;
        yiYm = y1(k)-ym1;
        Sxy1 =Sxy1+(floor(xiXm/y)*yiYm);%Covarianza
        vx1 = vx1 + xiXm^2; %Varianza
        vy1 = vy1 + yiYm^2;
    end
dx1 = fix(sqrt(vx1)/(y-1)); %DesviaciónEstándar
dy1 = fix(sqrt(vy1));       %DesviaciónEstándar
dxdy1 = dx1*dy1;
Ccp(i) =fix((Sxy1)/(dxdy1/nbits));
    %for q = 1:1:y
    %    Xm(i,q)=floor((x1(q)-xm1)/y); %Resta para el cálculo
    de la covarianza entre elc omando end
    %end
D(i)=floor(sqrt(vx1)/(y-1)); %Desviaciónestandar
end
c=0; %cantidaddecoeficientesquesuperanelumbraldeigualdad
for u = 1:m1
    if Ccp(u) >= 15291
```

```
        c=c+1;
    end
end

porcenA = c*100/(m1);
 %PorcentajedeCoeficientesdePearsonquesuperanelumbraldecomparación
% if c >= m1-1
%     comando = 'Usted dijo Apagar';
% else
%     comando = 'Usted no dijo Apagar';
% end

end
```

Bibliografía

[1] Las Nuevas Tecnologías al servicio de la discapacidad. David Miranda, responsable de marketing habla de tecnologías inclusivas. Disponible en: https://fundacionadecco.org/azimut/las-nuevas-tecnologias-al-servicio-de-la-discapacidad/.

[2] Asistente de voz. Glosario. AT INTERNET a Piano Company. Disponible en: https://www.atinternet.com/es/glosario/asistente-de-voz/

[3] Posibilidades y limitaciones en el desarrollo humano desde la influencia de las tecnologías de la información y la comunicación en la salud: el caso latinoamericano. Olga Cecilia Wilches-Flórez & Ángela María Wilches-Flórez. 2017. Universidad de la Sabana. Disponible en: https://personaybioetica.unisabana.edu.co/index.php/personaybioetica/article/view/7115/pdf

[4] Sanación gracias a la I.A. | The Age of A.I. , Capitulo 2, Temporada 1. Fecha de consulta: Febrero 202. Disponible en: https://www.youtube.com/watch?time_continue=7&v=V5aZjsWM2wo&feature=emb_logo; Diciembre 2018-2019.

[5] Discapacidad y rehabilitación; Informe mundial sobre la discapacidad, Fecha de consulta: Enero de 2020. Recuperado de: http://www.who.int/disabilities/world_report/2011/es; 2011.

[6] Sala situacional de las Personas con Discapacidad (PCD); Ministerio de Salud y Protección Social Oficina de Promoción Social, Paginas: 37, Noviembre 2017. Fecha de consulta: Febrero 2020. Available: https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/DE/PES/presentacion-sala-situacional-discapacidad-2017.pdf; Pagina 21.

[7] Todo lo que necesitas saber sobre discapacidad motriz, Fecha de consulta: Enero de 2020. Available: https://www.incluyeme.com/todo-lo-que-necesitas-saber-sobre-discapacidad-motriz; 2016.

[8] Camilo Jaimes. Latinoamerica, COLOMBIA PRESENTÓ PROYECTOS TIC PARA PERSONAS CON DISCAPACIDAD.. Fecha de consulta: Febrero 2020. Recuperado de: https://consultorsalud.com/colombia-presento-proyectos-tic-para-personas-con-discapacidad/; 2019.

[9] REGISTRO PARA LA LOCALIZACION Y CARACTERIZACION DE PERSONAS CON DISCAPACIDAD, Fecha de consulta: Enero de 2020. Available: http://www.discapacidadcolombia.com/index.php/estadisticas/154-estadisticas-en-discapacidad; 2015.

[10] EL nuevo proyecto de Google, Euphonia, tiene como objetivo mejorr el reconocimiento de voz para las personas con impedimentos del habla, Fecha de consulta: Febrero 2020. Recuperado de: https://afrotech.com/googles-new-project-euphonia-aims-to-improve-voice-recognition-for-people-with-speech-impairments; 2019.

[11] M. T. Qadri and S. A. Ahmed, Voice Controlled Wheelchair Using DSK TMS320C6711, 2009 International Conference on Signal Acquisition and Processing, Kuala Lumpur, 2009, pp. 217-220. doi: 10.1109/ICSAP.2009.48 , Avialable: http://ieeexplore.ieee.org.crai-ustadigital.usantotomas.edu.co/stamp/stamp.jsp?tp=&arnumber=5163858&isnumber=5163805.

[12] Crean silla manejada con la voz. VOS. Fecha de consulta: Febrero 2020. URL: https://www.vosmagazine.com/2018/08/crean-silla-ruedas-manejada-con-voz.html; 2006.

[13] HISTORIA DE LOS SISTEMAS DE RECONOCIMIENTO DE VOZ, Fecha de consulta: Junio 2020. Disponible: https://www.timetoast.com/timelines/historia-de-los-sistemas-de-reconocimiento-de-voz; 2017.

[14] Creación de programas en Colombia para personas con discapacidad, Available: https://www.incluyeme.com/creacion-de-programas-en-colombia-para-personas-con-discapacidad/; 2017.

[15] Informe mundial sobre la discapacidad, Fecha de consulta: Enero de 2020. Recuperado de: http://www1.paho.org/arg/images/Gallery/Informe_spa.pdf.

[16] Programación de FPGAs y DSP, Soluciones Electronicas de Cantabria. Fecha de consulta: Junio 2020. Disponible en: http://selcanelectronica.com/programacion-vhdl.php, 2013.

[17] Necesidades educativas especiales asociadas a discapacidad motora,Primera edicion, Santiago de Chile, Available: http://especial.mineduc.cl/wp-content/uploads/sites/31/2016/08/GuiaMotora.pdf; Diciembre, 2007.

[18] Asistentes virtuales más famosos: Alexa, Cortana y Siri, Por Redacción España, el 29/04/2021. Disponible en: https://agenciab12.com/noticia/asistentes-virtuales-famosos-alexa-cortana-siri.

[19] DSPs, Electronicasi. Fecha de consulta: Junio 2020. Pag 91. Disponible en: http://www.electronicasi.com/wp-content/uploads/2013/04/dspElectronica-avanzada.pdf.

[20] ALGORITMO DE LEVINSON ?DURBIN, Angel Paul. Septiembre 8, 2017. Disponible en: https://kupdf.net/download/algoritmo-de-levinson-durbin_59b18c94dc0d604330568edb_pdf

[21] Fabian ALberto Bustos Goméz Y Henry Navy Pantevis Perilla, Implementación de un Sistema de Reconocimiento de Voz en FPGA como Interfaz Hombre-Máquina en Aplicaciones de Robótica, Universidad Pedagógica y Tecnológica de Colombia, Seccional Sogamoso, Ingenieria Electrónica, Semestre II de 2016, Available: https://repositorio.uptc.edu.co/bitstream/001/1869/1/TGT-434.pdf.

[22] © 2022 Arduino, Arduino UNO R3. Disponible en: https://docs.arduino.cc/hardware/uno-rev3.

[23] David Reig Albiña, Implementación de algoritmos para la extracción de patrones característicos en Sistemas de Reconocimiento De Voz en Matlab, Universidad Politecnica de Valencia, Escuela Politecnica Superior de Gandia, I.T. Telecomunicaciones (Imagen y Sonido), Gandia, 2014, Disponible en: https://riunet.upv.es/bitstream/handle/10251/59092/TFC_DAVID_REIG_ALBI%C3%91ANA.pdf?sequence=1.

[24] Fernando Peralta, Reyes Aníbal Cotrina Atencio, Asesor: Guillermo Tejada Muñoz. RECONOCEDOR Y ANALIZADOR DE VOZ, Recuperado de UNMSM (Universidad Nacional Mayor de San Marcos). Facultad de Ingeniería Electrónica, 2015. Available: https://www.researchgate.net/publication/238089777_RECONOCEDOR_Y_ANALIZADOR_DE_VOZ.

[25] Li, J., An, D., Lang, L., & Yang, D. (2012). Embedded Speaker Recognition System Design and Implementation Based on FPGA. Procedia Engineering, 29, 2633-2637. Recuperado de: https://www.researchgate.net/publication/271608384_Embedded_Speaker_Recognition_System_Design_and_Implementation_Based_on_FPGA.

[26] Ignacio Cobeta, Fautismo Nuñez, Secundino Fernández: Patología de la Voz. Ponencia oficial. Sociedad Española de Otorrinolaringología y Patología Cérvico-Facial 2013. Pagina 85. Recuperado de : https://seorl.net/PDF/ponencias%20oficiales/2013%20Patolog%C3%ADa%20de%20la%20voz.pdf.

[27] Aparato fonador humano: qué es, partes y funciones, Samuel Antonio Sánchez Amador, Fecha de consulta: Julio de 2021. Available: https://psicologiaymente.com/salud/aparato-fonador-humano; 2021.

[28] Herramientas software para la docencia de la señal de voz en Ingeniería, Técnica de Telecomunicaciones, S. Bleda ; J. Francés ; S. Marini ; J.J. Martínez, Departamento de Física Ingeniería de Sistemas y Teoría de la Señal, Universidad de Alicante, San Vicente del Raspeig Ap. 99, E-03080, Alicante (España), Instituto Universitario de Física Aplicada a las Ciencias y Tecnologías, Universidad de Alicante, San Vicente del Raspeig Ap. 99, E-03080, Alicante (España), 2012, Disponible en: https://web.ua.es/es/ice/jornadas-redes-2012/documentos/posters/246141.pdf.

[29] C5535/C5545 eZdsp USB Stick Development Kit, TMDX5535EZDSP. TEXAS INSTRUMENTS. 2019. Available: http://www.ti.com/tool/TMDX5535EZDSP.

[30] Tom Bäckström, Windowing, Basic representations and models, Introduction to speech processing. 2019. Available: https://wiki.aalto.fi/display/ITSP/Windowing.

[31] TI Designs Speech Recognition Reference Design on the C5535 eZdsp TEXAS INSTRUMENTS. 2019. Available: http://www.ti.com/lit/ug/tidubj5a/tidubj5a.pdf.

[32] G. Huang and Z. Tian, ´´An automatic speech recognition system on DSP board", 2016 International Automatic Control Conference (CACS), Taichung, 2016, pp. 224-226.doi: 10.1109/CACS.2016.7973913, Available: https://ieeexplore-ieee-org.crai-ustadigital.usantotomas.edu.co/document/7973913.

[33] Philips Scene Switch LED Bulb 9W 3 Step Brightness Change E27. Luminance systems. Available: https://www.luminancesys.com/products/philips-scene-switch-led-bulb-9w-3-step-brightness-change-e27.

[34] 5V Dual-Channel Relay Module HW-383. Enero 2021. Available: https://components101.com/switches/5v-dual-channel-relay-module-pinout-features-applications-working-datasheet.

[35] Utiliza un relé con tu placa Arduino. Alberto Valero. 2016 Disponible en: http://diwo.bq.com/utilizar-rele-arduino-zum-core/.

[36] Eduardo Morales, Enrique Sucar. INAOE. 2017. Reconocimiento de Voz .Disponible en: https://ccc.inaoep.mx/~esucar/Clases-ia/Laminas2017/recvoz.pdf

[37] Ester Rodriguez. Audioprotesista en su centro Salesa ? Pau Claris. ¿CUÁLES DEBEN SER LOS NIVELES SONOROS EN NUESTRO ENTORNO COTIDIANO?. 21 MAYO 2019. Disponible en: https://www.salesa.es/es/noticias/cuales-deben-ser-los-niveles-sonoros-en-nuestro-entorno-cotidiano/_noticia:167/

[38] Adobe Illustrator. Adobe Creative Cloud. Disponible en: https://www.adobe.com/co/products/illustrator.html?sdid=KQPQL&

mv=search&ef_id=CjwKCAjws8yUBhA1EiwAi_tpEdG6USb1E1LKWK_
suSb-8NSt7eA4RA49WiXkF1NRSYbb0chUSeuDdRoCMRgQAvD_BwE:G:s&
s_kwcid=AL!3085!3!459896392480!e!!g!!adobe%20illustrator!
9499870682978134137 98&gclid=CjwKCAjws8yUBhA1EiwAi_tpEdG6USb1E1LKWK_
suSb-8NSt7eA4RA49WiXkF1NRSYbb0chUSeuDdRoCMRgQAvD_BwE