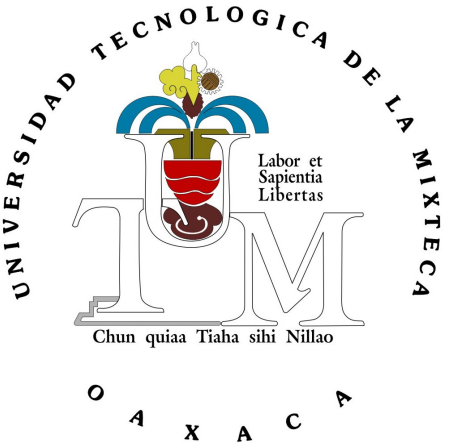


UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

"SISTEMA DE RECONOCIMIENTO MULTIMODAL PARA EL ESPAÑOL MEXICANO"

T E S I S

PARA OBTENER EL TÍTULO DE
MAESTRO EN ROBÓTICA

PRESENTA
FELIX EMILIO LUIS PÉREZ

DIRECTOR DE TESIS
DR. FELIPE DE JESÚS TRUJILLO ROMERO

ASESOR DE TESIS
DR. SANTIAGO OMAR CABALLERO MORALES

Huajuapan de León, Oaxaca, Abril 2013.

A mi Familia.

Por hacerme una persona responsable.

A Jean.

Por su apoyo, motivación y cariño.

Agradecimientos

Agradezco principalmente a mis directores de tesis, el Dr. Felipe de Jesús Trujillo Romero y el Dr. Santiago Omar Caballero Morales, por el tiempo y la paciencia dedicada a las revisiones de este trabajo, así como su apoyo y consejos en el mejoramiento del mismo.

Agradezco también a mis sinodales, la M.C. Verónica Rodríguez López, el Dr. José Aníbal Arias Aguilar y el Dr. Agustín Santiago Alvarado por su tiempo y disposición al revisar este trabajo, así como su valiosa contribución en el mejoramiento del mismo.

Al Dr. Raúl Cruz Barbosa por sus importantes observaciones durante el registro de este trabajo de tesis.

A la Universidad Tecnológica de la Mixteca, por permitirme cursar y concluir mis estudios de maestría en los diferentes espacios de la misma.

A mis profesores, en especial aquellos que hacían su trabajo con verdadero entusiasmo.

A mis padres, porque siempre han estado para apoyarme y me han impulsado en los momentos más difíciles de mi carrera, porque me enseñaron que el trabajo continuo genera mejores resultados, me enseñaron a ser una persona responsable, útil e insistente en mis objetivos.

A mis hermanos Fanny y Fabian, porque ellos han sido la parte distractora en mis estudios, esa parte que cuando estoy preocupado por algo, un simple ¡ash manito! o ¡vamos a jugar!, son suficientes para desviar mi atención y olvidar por un momento los pendientes.

A Jean, porque siempre me ha impulsado a alcanzar mis objetivos, porque cuando necesito que alguien me escuche sé que puedo contar con ella, porque gracias a su cariño y comprensión puedo darme valor para enfrentar nuevos retos, porque cuando decía, ¿ya fuiste a ver a Felipe?, y yo no sabía que contestar, ella sólo movía la cabeza para reprobar mis acciones y era suficiente para ponerme a trabajar de nuevo en la tesis.

A todas y cada una de las personas que se vieron involucradas, directa o indirectamente, con la realización de este trabajo, **gracias**.

Felix Emilio Luis Pérez.

Abril, 2013.

Índice general

Índice de figuras

Índice de cuadros

Prólogo

Durante las últimas décadas, los diferentes avances en procesamiento de señales, algoritmos, arquitecturas y plataformas de cómputo han provocado un inminente crecimiento en la investigación y desarrollo de sistemas de Interacción Humano-Computadora (HCI, por sus siglas en inglés). La Interacción Humano-Computadora, se refiere al diseño de sistemas computacionales que permiten a las personas llevar a cabo sus actividades de manera eficiente, segura y fácil (Preece y otros, 1994). Actualmente, no sólo aborda el tema relacionado con sistemas de oficina, donde fueron sus inicios, sino también temas relacionados con dispositivos móviles, servicios web, servicios de Internet, juegos, entre otros. Además, se agregan los temas relacionados con los robots autónomos y en consecuencia, la interacción humano-robot requerida para este tipo de sistemas (Kiesler y Hinds, 2004).

De acuerdo con la definición presentada por Goodrich y Schultz (2007), Interacción Humano-Robot (HRI) es un campo de estudio dedicado a entender, diseñar y evaluar sistemas robóticos para ser usados por los seres humanos y entre los seres humanos. El término interacción, indica la existencia de un tipo de comunicación entre dos o más entes, en este caso, el robot y el humano. Dicha comunicación se puede realizar de diferentes maneras, sin embargo, existe una gran influencia por el grado de proximidad entre el ser humano y el sistema robótico. Por tanto, la comunicación y en consecuencia la interacción entre estos dos entes, puede ser separada en dos categorías generales:

- Interacción a distancia. El ser humano y el robot se encuentran separados espacialmente e incluso temporalmente.
- Interacción de proximidad. El ser humano y el robot están localizados en el mismo espacio geográfico y temporal.

Dentro de esta división es posible distinguir categorías más específicas, tales como aplicaciones que requieren movimiento, manipulación o interacción social. Por ejemplo, la robótica móvil dentro de la interacción a distancia es comúnmente identificada como teleoperación o control a distancia y la interacción a distancia con manipulación es conocida como telemanipulación. Asimismo, la interacción de proximidad combinada con robótica móvil y manipulación dan lugar a un la robótica de servicio. Finalmente, la interacción social incluye

aspectos sociales, emocionales y cognitivos de interacción. En este caso, los humanos y los sistemas robóticos interactúan como compañeros, pudiendo ser este tipo de interacción a distancia o bien, de proximidad.

Como parte de los trabajos identificados en el área de HRI se encuentran los sistemas de reconocimiento automático, dentro de los cuales se pueden mencionar los enfocados al reconocimiento automático del habla, reconocimiento de rostros, formas, colores, gestos corporales, etc. Todos ellos con la intención de apoyar a las personas en sus actividades cotidianas, pero sobretodo, pretenden proveer de una interacción natural humano-robot.

En este sentido, el reciente desarrollo de la robótica fuera de aplicaciones industriales, se concentra cada vez más en la operación de robots de servicio con múltiples funciones, en un entorno dinámico y con la capacidad de interactuar con las personas y objetos del mismo entorno. Dicho desarrollo, se clasifica como una rama de la robótica de nombre "robótica de servicio", la cual se enfoca en desarrollar algoritmos, metodologías y robots que cumplan ciertos requisitos para que sean aceptados por la ideología del ser humano, además de que cubran la interacción humano-robot ideal (Breazeal, 2002).

De acuerdo con Rahimi y Karwowski (1992), uno de los problemas más importantes y complejos en el área de robótica de servicio es la interacción humano-robot, donde es el humano quién debe indicar al sistema robótico una tarea a realizar. En este contexto, la mejor manera de proveer dicha interacción es sin duda el reconocimiento del habla, rostros y gestos. El habla, por ser la comunicación natural entre las personas; los rostros y gestos porque son ellos quienes emiten gran parte de los mensajes en una conversación.

Es por ello, que el presente proyecto de tesis se enfoca en desarrollar un sistema de reconocimiento multimodal, capaz de interpretar señas y mensajes vocales emitidos por un usuario. El sistema debe ser capaz de interpretar los signos correspondientes al alfabeto del Lenguaje de Señas Mexicano (LSM) y comandos vocales pronunciados también en Español Mexicano. Ambos, enfocados y diseñados para controlar un sistema robótico de servicio.

Planteamiento del problema

Como se mencionó en párrafos anteriores, la interacción humano-robot es uno de los problemas más importantes y atacados en la robótica de servicio. Una posible forma de comunicación entre las personas y un robot, puede ser sin duda el uso de un teclado o botones de acción. Sin embargo, cuando se piensa en funciones como la marcación por voz de un teléfono celular, o la interacción con juegos por medio de un sensor tridimensional, resulta mucho más natural y amigable la comunicación mediante señas y voz, como si ésta fuese directamente con otra persona.

El presente trabajo de tesis, pretende proveer de un medio de interacción entre un ser humano y un sistema robótico, con el cual el usuario pueda dar instrucciones de movimiento

y manipulación al robot, pero sobre todo, sin utilizar una herramienta electrónica de comunicación como el teclado. Se plantea el desarrollo de un sistema de reconocimiento multimodal que incluya las dos formas de comunicación más usadas por los humanos, señas y voz. Por una parte, la identificación de signos correspondientes al alfabeto del Lenguaje de Señas Mexicano (LSM), proporcionando la interacción con el robot por medio de señas. En tal caso, el reconocimiento no dependerá de mecanismos electrónicos para modelar los símbolos del alfabeto. Por otro lado, el reconocimiento de comandos vocales, también para el español Mexicano, proporcionando la interacción mediante voz con el sistema robótico. En ambos casos, se interpretará un mensaje diseñado para controlar un sistema robótico de servicio.

Cabe mencionar que el lenguaje de señas es la comunicación natural de las personas con discapacidades auditivas y del habla. Además, tiene la característica principal de que su forma de transmisión es completamente visual y son parte de la cultura de las comunidades de sordomudos, en la cual, reflejan su visión del mundo. Por su parte, los comandos vocales son una representación abstracta de acciones o tareas que una persona quiere dar a entender por medio de la voz. Donde estos comandos, pueden incluir hasta un máximo de 10 palabras y son utilizados comúnmente para interactuar con sistemas electrónicos.

Justificación

El lenguaje de señas es utilizado principalmente por la comunidad de sordomudos. Sin embargo, las personas que no tienen dicha discapacidad también aprenden el lenguaje con el fin de comunicarse con aquellos que no pueden oir o hablar. Cabe mencionar que el uso de este lenguaje no está limitado a dicha comunidad, pues también es usado por aquellas personas que necesitan comunicarse en lugares donde las voces no se utilizan. Por ejemplo, los buzos o los usuarios de maquinaria ruidosa. Por otra parte, el hecho de utilizar un lenguaje de señas, facilita su aprendizaje ya que al ser un lenguaje, posee su propia estructura gramatical y lingüística como la semántica y la sintaxis.

En este proyecto de tesis, la utilización del LSM se basa en la gran cantidad de personas que pueden llegar a utilizar el lenguaje y la facilidad con que dichas personas puedan aprenderlo. Se resalta también, que son muy pocos los trabajos que se han desarrollado usando esta codificación, pues la mayoría de sistemas desarrollados a la fecha, utilizan el lenguaje de señas Norte-Americano, tal como se muestra en Munib y otros (2007). Cabe mencionar, que cada país posee su propia codificación para el alfabeto de señas (Munib y otros, 2007; Vargas y otros, 2010; Karami y otros, 2011; Paulraj y otros, 2010).

En conclusión, el reconocimiento de señas puede proveer un medio de comunicación humano-robot en dos situaciones básicas: con personas de discapacidades auditivas y del habla y con personas que dadas sus condiciones de trabajo, no puedan utilizar la voz para comunicarse. Por su parte, el reconocimiento de comandos vocales facilita la comunicación con aquellas personas que presenten alguna discapacidad física o motriz, personas de la ter-

cera edad y con la población en general. De esta manera, el sistema multimodal puede ser implementado como un sistema de interacción natural humano-robot, que independientemente de ser utilizado por personas con alguna discapacidad, ya sea física, auditiva o del habla, puede ser utilizado por la población en general.

Finalmente, el sistema de reconocimiento multimodal pueden servir como base para el desarrollo de sistemas de control robótico a distancia. Es decir, sistemas que faciliten el control de robots desde un sitio remoto o por medio de algún tipo de conectividad inalámbrica.

Hipótesis

Como se mencionó previamente, los sistemas de reconocimiento automático y en especial los sistemas de interacción humano-robot, representan uno de los problemas más importantes y complejos en el área de robótica (Rahimi y Karwowski, 1992). Por tal motivo, se desea desarrollar un sistema de interacción multimodal utilizando lenguaje natural como señas y voz. Dicho sistema deberá resultar más fácil y atractivo de usar que un sistema de comunicación por teclado.

En este sentido, es posible desarrollar el sistema multimodal con tres módulos principales: El primero, para reconocimiento de señas utilizando Redes Neuronales Artificiales; el segundo, enfocado al reconocimiento de comandos vocales utilizando Modelos Ocultos de Markov; y un tercer módulo utilizando ponderación estadística para unificar los resultados del reconocimiento de señas y comandos vocales. Los dos módulos de reconocimiento (señas y voz) deberán trabajar de manera independiente y proporcionar una interpretación del mensaje emitido por el usuario, no obstante sólo se tomará en cuenta el resultado emitido por el módulo final. Para ello, es necesario generar una base de datos de mensajes que el sistema sea capaz de reconocer y que dichos mensajes estén diseñados para controlar un sistema robótico de servicio.

Por su parte, la captura de los datos del usuario (imágenes y voz), se debe realizar por medio de un dispositivo *Kinect* (Microsoft, 2009), el cual es capaz de percibir datos visuales y sonoros de su entorno gracias a sus arreglos de cámaras y micrófonos montados en el mismo. En tal caso, se debe capturar primero la señal de voz, debido a su rapidez de procesamiento, para después capturar la señal visual representada por una imagen en color y una imagen en profundidad. De esta manera, el orden de ejecución debe ser el siguiente: módulo de voz, módulo de señas y al final, el módulo de ponderación estadística.

Objetivos

Objetivo general

Desarrollar un sistema de reconocimiento multimodal capaz de reconocer los símbolos del alfabeto del Lenguaje de Señas Mexicano y comandos vocales también en español Mexicano. Dicho sistema, deberá conjuntar interpretaciones de señas y voz para identificar un solo mensaje, el cual estará diseñado y enfocado a controlar un sistema robótico de servicio.

Objetivos específicos

- Implementar el reconocimiento del alfabeto del LSM utilizando Redes Neuronales Artificiales. Primero con imágenes bidimensionales y después con imágenes tridimensionales, en ambos casos, usando señas que no involucren movimiento.
- Implementar el reconocimiento de comandos vocales para el español Mexicano utilizando Modelos Ocultos de Markov.
- Utilizar técnicas de adaptación de voz, que permitan el funcionamiento multi-usuario del correspondiente módulo.
- Implementar un sistema de reconocimiento multimodal, usando señas y voz en español Mexicano.
- Validar el sistema de reconocimiento multimodal mediante el desarrollo de una aplicación en un entorno de simulación.

Organización de la tesis

El presente trabajo de tesis se encuentra organizado en cinco capítulos principales.

En el Capítulo 1, estado del arte, se muestra una serie de trabajos relacionados o similares al propuesto en el presente documento. Se hace mención de los diferentes proyectos e investigaciones que existen en relación al reconocimiento de señas y voz.

El Capítulo 2, muestra la teoria necesaria para comprender el funcionamiento del sistema multimodal. Se detallan las técnicas de procesamiento de patrones referente a Redes Neuronales y Modelos Ocultos de Markov, que son las dos técnicas utilizadas en el presente trabajo de tesis. El primero para identificar las señas y el segundo para procesar las muestras de voz.

El desarrollo de los módulo de señas, voz y de unificación, se detallan en Capítulo 3. Se muestran los dos casos de estudio realizados en el módulo de señas, ocupando una *webcam* y un

dispositivo Kinect. También se detalla la construcción del sistema de voz y su correspondiente adaptación de usuarios, así como la forma de unificar los resultados parciales de cada módulo de reconocimiento.

El Capítulo 4, presenta los resultados del sistema implementado. Por una parte los resultados parciales de cada módulo de reconocimiento y por la otra los resultados del sistema multimodal unificado.

Finalmente, el Capítulo 5 muestra las conclusiones y perspectivas de trabajo a futuro del presente proyecto de tesis.

Publicaciones derivadas

Como parte de este trabajo de tesis, se presentaron los siguientes artículos en conferencias nacionales e internacionales, los cuales son enlistados de manera cronológica.

1. Luis-Pérez, F. E., Martínez-Velazco, W. y Caballero-Morales, S. O. (2011). Towards the Development of a Speech Recognition Interface for Human – Robot Interaction. *III Congreso Nacional y II Congreso Internacional de Computación e Informática (CONACI 2011)*, páginas 139-147. ISBN: 978-607-782-615-6.

2. Luis-Pérez, F. E. y Trujillo-Romero, F. J. (2011). Reconocimiento del Lenguaje de Señas Mexicano para su Interpretación y uso en Sistemas Robóticos. *VII Semana Nacional de Ingeniería Electrónica (SENIE 2011)*, páginas 413-421. ISBN: 968-607-477-588-4.

3. Luis-Pérez, F. E., Trujillo-Romero, F. J. y Martínez-Velazco, W. (2011). Control of a Service Robot Using the Mexican Sign Language. *10th Mexican International Conference on Articial Intelligence (MICAI 2011). Advances in Soft Computing, Lecture Notes in Articial Intelligence*, Volumen 7095, páginas 419-430. ISBN: 978-3-642-25329-4. ISSN: 0302-9743.

4. Trujillo-Romero, F. J., Luis-Pérez, F. E. y Caballero-Morales, S. O. (2012). Multimodal interaction for service robot control. *22th International Conference on Electronics Communications and Computers (CONIELECOMP 2012)*, páginas 305-310. ISBN: 978-1-61284-1325-5. Ganador del premio internacional Rashid al mejor artículo de la conferencia.

5. Luis-Pérez, F. E., Trujillo-Romero, F. J. y Caballero-Morales, S. O. (2012). Comparativa del Desempeño de las Variantes de Backpropagation en una Tarea de Clasificación de Objetos. *VIII Semana Nacional de Ingeniería Electrónica (SENIE 2012)*, páginas 420-428. ISBN: 978-607-477-902-8.

6. Trujillo-Romero, F. J., Caballero-Morales, S. O. y Luis-Pérez, F. E. (2012). Sistema de reconocimiento del habla para identificación de usuario mediante el uso de codificación de predicción lineal y redes neuronales. *VIII Semana Nacional de Ingeniería Electrónica (SENIE 2012)*, páginas 364-373. ISBN: 978-607-477-902-8.

Capítulo 1

Estado del arte

1.1. Introducción

En el área de robótica, los sistemas de reconocimiento automático han tomado un papel muy importante en los procesos de interacción humano-robot. Por medio de éstos, se pretende proveer de un tipo de comunicación natural entre los seres humanos y los sistemas robóticos, ya sea utilizando características del habla, de rostros, formas, colores, gestos corporales, o cualquier otra seña utilizada en la interacción natural de las personas. Por este motivo, en el presente capítulo se presenta un resumen general de los trabajos relacionados al reconocimiento automático y sus correspondientes aplicaciones en robótica.

Se pueden mencionar por ejemplo, trabajos que utilizan las características de color y profundidad para detectar algún objeto. Un primer acercamiento a esta área es el reconocimiento de objetos o formas geométricas usando características de profundidad (Gadh y Prinz, 1992). Sin embargo, gran parte de las investigaciones se han enfocado en identificar figuras humanas, tomando en cuenta las variaciones de posición, vestimenta, condiciones de iluminación y fondo de imágenes (Dalal y otros, 2006; Ikemura y Fujiyoshi, 2010). También, se han utilizado sistemas estereoscópicos en la ubicación del cuerpo humano, tal es el caso del trabajo presentado por Muñoz y otros (2007), donde los autores utilizan un sistema de visión estereoscópica e información de profundidad para identificar y seguir el cuerpo de las personas. Un trabajo más reciente en esta área, fue presentado por Salas y Tomasi (2011), en este caso los autores combinan el color y la profundidad de las imágenes con la finalidad de detectar personas en ambientes cerrados.

Otros autores han incluido el uso del *Kinect* para obtener información tridimensional de los objetos. La rápida evolución de esta tecnología provee una alta calidad de sincronización entre imágenes de color y profundidad, aunado a una frecuencia de respuesta muy alta. Por ejemplo, utilizar el sensor de profundidad del *Kinect* y así implementar un sensor táctil sobre una mesa (Wilson, 2010). Además, Lu y otros (2011) utilizaron el *Kinect* para detectar la

silueta humana con información del sensor de profundidad y modelados 2D y 3D de la cabeza humana. Un ejemplo más, se observa en el trabajo presentado por Hernández-López y otros (2012), donde utilizan características de color y profundidad extraídas con dicho sensor para detectar objetos en el espacio de trabajo de un robot. Los autores utilizan éstas características y proponen una estrategia de movimiento procesada en tiempo real.

Otro tipo de reconocimiento es el que se relaciona con los rostros y las características del mismo. Dos problemas básicos e independientes que afrontan estos sistemas son: la detección del rostro en la escena o imagen, y el reconocimiento del rostro detectado. Actualmente, existe una gran cantidad de algoritmos de detección de rostros con desempeño variado y que depende de los escenarios considerados en cada caso (Yang y otros, 2002; Zhao y otros, 2003). Por su parte, el reconocimiento de rostros consiste en buscar un rostro dentro una base de datos de referencia para encontrar las coincidencias e identificar un rostro previamente dado (Lone y otros, 2011). En algunos casos este tipo de reconocimiento automático es utilizado en aplicaciones de análisis forense (Jain y otros, 2011).

Cabe mencionar que los sistemas de reconocimiento tienen una variedad de aplicaciones en robótica, desde implementación en sistemas militares, sistemas de exploración espacial, entretenimiento, o bien robots de asistencia y educación. En el ámbito militar, se pueden encontrar aplicaciones de robots controlados remotamente para abordar y revisar paquetes sospechosos (Wells y Deguire, 2005). Asimismo, implementación de robots con funciones de ayudantes o compañeros de los militares, trabajando como equipos de colaboración mixtos humano-robot (Bruemmer y Walton, 2003; Kennedy y otros, 2007).

En cuanto a exploración espacial, los robots se han utilizado comúnmente en exploración de la superficie lunar (Duke y otros, 2003), o superficies planetarias, principalmente la del planeta Marte, así como para dar mantenimiento a estaciones espaciales (Fong y Thorpe, 2001; Leger y otros, 2005).

Las aplicaciones de entretenimiento, son las más comercializadas para uso fuera de instituciones de investigación y desarrollo, en este caso el rol de los humanos es de observador y la interacción con los sistemas robóticos es mínima. El robot, generalmente reproduce un sonido pregrabado que sincroniza con sus formas de movimiento, por ejemplo robots bailadores (Kosuge y otros, 2003), o bien, mascotas robóticas (Fong y otros, 2003).

Finalmente, un área reciente de aplicaciones en robótica es la relacionada con robots de asistencia y educación, llamada comúnmente robótica de servicio, la cual es quizá una de las más demandantes en Interacción Humano-Robot (HRI). Como ejemplo de estas aplicaciones están aquellas que pretender aumentar el conjunto de tareas que una persona con discapacidad pueda llevar a cabo de forma independiente. Estas tareas incluyen asistencia de navegación en ambientes no estructurados (Kulyukin y otros, 2006), o facilidades de transportación (Yanco, 2001). Además, algunas investigaciones están explorando la mejor forma de utilizar los robots para promover la educación de los niños, tanto en el hogar como en las escuelas (Cooper y otros, 1999; Han y otros, 2005).

Dentro de esta última área de aplicación, referente a la robotica de servicio, se encuentra ubicado el estudio de la presente tesis. En tal caso, es necesario utilizar dos tipos de reconocimiento automático: reconocimiento del lenguaje de señas y reconocimiento de comandos vocales. Es por ello que en las siguientes secciones se hará un bosquejo de los trabajos existentes en reconocimiento de señas y comandos vocales, así como las diferentes aplicaciones que se dan a este tipo de investigaciones.

1.2. Sistemas de señas

Como se puede observar, el área de aplicación de los sistemas de reconocimiento automático es muy amplia. Sin embargo, en la robótica de servicio uno de los retos más importantes es el reconocimiento de gestos para controlar el sistema. Para ello, se requiere capturar e interpretar movimientos de la cabeza, los ojos, cara, manos, brazos o cuerpo entero. Ejemplo de ello, se puede observar en Hashimoto y otros (2009), donde los autores realizan un seguimiento de los movimientos oculares y con ello controlar una silla de ruedas. Por su parte, Yan y otros (2012) utilizan una estructura montada sobre la persona para percibir todos los movimientos que realiza ésta última y reproducirlos en un sistema robótico.

Entre las diferentes partes del cuerpo, la mano es la herramienta más eficaz para interacción por su amplia funcionalidad en cuanto a comunicación y manipulación. Se han analizado diferentes estilos de interacción que permitan una comunicación natural e intuitiva entre un ser humano y un sistema robotizado. En este contexto, se han desarrollado algunos sistemas que permitan interactuar por medio de las señas, inicialmente con la ayuda de guantes electrónicos o acelerómetros para modelar la mano humana. Por ejemplo, usando la codificación del lenguaje de señas Norte-Americano (Waldron y Kim, 1995), Australiano (Kadous, 1996) o Mexicano (Villa-Angulo y Hidalgo-Silva, 2005). Estos últimos, no sólo identificaron las señas, sino también realizaron la traducción de éstas a su correspondiente mensaje de voz y texto.

No obstante, no todos los trabajos necesitan de una estructura o elemento mecánico para describir los movimientos y gestos de un usuario. En este sentido, los sistemas estereoscópicos han sido de gran utilidad en la captura de imágenes, así como en la extracción de información necesaria para reconocer algún gesto humano (Nickel y Stiefelhagen, 2007; Maldeni y otros, 2011). Un ejemplo de ello, se muestra en el trabajo presentado por Kollorz y otros (2008), donde se utilizan características de profundidad con el objetivo de reconocer un conjunto de 12 señas aplicadas a sistemas automotrices. El *Kinect* (Microsoft, 2009), también ha servido como proveedor de datos y características en el reconocimiento de señas. Ejemplo de ello, se muestra en López-Monroy y Leal-Meléndrez (2011), donde se realiza un reconocimiento de siete señas utilizando Modelos Ocultos de Markov.

Otros trabajos se han apoyado de un modelo tridimensional de la mano para hacer el reconocimiento de señas. Wu y Huang (2001) realizan el modelado de la forma, estructura

cinemática y movimientos de la mano, que sirven en el reconocimiento de algunos símbolos del alfabeto de señas Norte-Americano. En estos casos, la estructura o modelo tridimensional es montado sobre la mano del usuario, generalmente identificada mediante segmentaciones usando el color de la piel (O'Hagan y otros, 2002; Mo y otros, 2005), o extracción de fondos uniformes (Malik y Laszlo, 2004). Posteriormente, se hace un seguimiento de los movimientos de la mano con el objetivo de conocer el estado actual de ésta y extraer las características necesarias para identificar una seña determinada (Erol y otros, 2007).

En relación al seguimiento de los movimientos de la mano, se han utilizado diferentes técnicas con dicho propósito. Desde el uso de secuencias de video para construir la trayectoria que sigue la punta de un solo dedo (Yuan-Hsiang y Chen-Ming, 2010), hasta el uso de elementos de programación genética y Redes Bayesianas Dinámicas (El-Sawah y otros, 2007), las cuales permiten el seguimiento de toda la estructura de la mano. También se han usado los métodos comunes de seguimiento y predicción como el filtro de Kalman y el filtro de partículas. Un ejemplo de este último, es aplicado al control de aparatos electrodomésticos (Bretzner y otros, 2002). Por su parte, Ho-Sub y otros (2001) utilizan Modelos Ocultos de Markov para hacer el seguimiento de la palma de la mano y reconocer 12 elementos gráficos y 36 caracteres alfanuméricos, cada uno de los cuales debe ser trazado con la mano, y así poder reconocerlo.

Cabe mencionar que la mayoría de investigaciones define su propia codificación de señas. Esto, en dependencia de la aplicación que se dará al sistema de reconocimiento. Por ejemplo en Posada-Gomez y otros (2007), los autores solamente identifican la mano extendida y su desplazamiento para poder mover una silla de ruedas. La dirección de movimiento de la mano define la dirección de movimiento de la silla (arriba= adelante, abajo=atrás, derecha=giro a la derecha, izquierda=giro a la izquierda). Por su parte, Trigo y Pellegrino (2010) definen un alfabeto de seis símbolos para probar su reconocedor de señas. En este caso, los autores utilizan como señas: la mano abierta, la mano formando una "v" con los dedos índice y medio, apuntando con el dedo índice y pulgar extendidos, apuntando sólo con el dedo índice, levantando el pulgar y la mano cerrada. Además, aseguran que el reconocedor es invariante a cambios de traslación, escala y rotación.

De manera similar, en Nguyen y otros (2011) los autores proponen un sistema de señas como forma de controlar un sistema robótico. Se proponen diferentes señas para definir cinco mensajes de comunicación con el robot, las cuales hacen referencia al llamado del robot, posturas de acuerdo y desacuerdo con el mismo, señalarle objetos y detener su movimiento. Asimismo, Nagi y otros (2011) definen seis señas con las cuales controlan un robot móvil de uso domestico. Dichas señas representan los números del cero al cinco, donde el cero es la mano cerrada y el cinco la mano abierta. Sin embargo, los autores definen un guante de color específico para hacer la separación de la mano con el resto del escenario.

En lo que respecta al uso de lenguaje de sordomudos, la mayor parte de las investigaciones se ha realizado con la codificación del lenguaje de señas Norte-Americano, así se muestra en Binh y Ejima (2005), donde los autores utilizaron técnicas de lógica difusa combinadas con

Redes Neuronales en la clasificación de dichos símbolos. Poco después, Munib y otros (2007) utilizaron la transformada de Hough y Redes Neuronales para clasificar las señas del mismo alfabeto. Por su parte, Chen y otros (2008) utilizaron solamente cuatro símbolos de este alfabeto para probar su sistema de reconocimiento. Utilizaron los símbolos correspondientes a las señas a, b, i y u, además de utilizar métodos estadísticos basados en características *Haar-like* (Viola y Jones, 2001) y el algoritmo de aprendizaje *AdaBoost* (Freund y Schapire, 1997). De forma similar, en Ayala-Ramirez y otros (2011) se utilizaron sólo seis símbolos del mismo alfabeto, en este caso, los autores se basaron en características geométricas y de color con la finalidad de reconocer la mano y así poder controlar un robot móvil.

Pocos han sido los trabajos que se enfocan en utilizar codificaciones diferentes, por ejemplo Al-Jarrah y Halawani (2001) decidieron utilizar el lenguaje de señas Árabe con sistemas neuro-difusos; Maung (2009) ocupa Redes Neuronales para reconocer las señas del alfabeto de Birmania; Dias y otros (2009) también usan modelos neuro-difusos en el reconocimiento del alfabeto brasileño; Karami y otros (2011) desarrollaron un sistema de reconocimiento para el lenguaje de señas Persa. Con respecto al lenguaje de señas Mexicano, se puede mencionar el trabajo desarrollado por Villa-Angulo y Hidalgo-Silva (2005), donde los autores utilizaron un guante electrónico para capturar las señas; y el trabajo mostrado en Luis-Pérez y otros (2011) donde se hace el reconocimiento de las señas usando Redes Neuronales Artificiales.

1.3. Sistemas de voz

El habla es la forma más natural y eficiente de comunicación entre las personas, pues resulta un mecanismo sencillo para la transmisión de ideas. Por ello, surge uno de los grandes retos en la robótica de nuestros días, la comunicación por voz entre humanos y sistemas robotizados, permitiendo una interacción humano-robot más acorde a lo acostumbrado por las personas.

El reconocimiento automático del habla (ASR por sus siglas en inglés) es una técnica que permite convertir señales de voz en texto para reconocimiento y comprensión de ideas (Liu y Li, 2010). Un primer acercamiento a este tipo de reconocedores se dio a principios de los años 80's cuando Burton y otros (1983) hacen uso de cuantización de vectores para hacer un reconocedor de palabras aisladas. Posteriormente, surgen nuevos métodos de reconocimiento como Redes Neuronales y Modelos Ocultos de Markov, los cuales son usados normalmente como complemento uno del otro (Tretin y Gori, 2001). Sin embargo, Waibel y otros (1989) utiliza solamente las Redes Neuronales para diseñar un reconocedor del habla.

En las últimas décadas los sistemas de reconocimiento automático de voz han experimentado un notable progreso. En particular, se han dado numerosas investigaciones que utilizan un vocabulario extenso, reconocimiento en tiempo real y casos con independencia de usuario (Young, 1996; Zweig y Picheny, 2004). Al mismo tiempo, las aplicaciones comerciales de reconocimiento de voz son más encontradas en el mercado. Por ejemplo en la industria,

las comunicaciones y teléfonos móviles, los sistemas eléctricos automotrices, la medicina, las casas inteligentes, etc.

En lo que respecta a sistemas de reconocimiento con vocabulario amplio se puede mencionar el trabajo presentado por Woodland y otros (1994), donde los autores utilizan Modelos Ocultos de Markov y la herramienta HTK (Hidden Markov Models Toolkit) para diseñar un reconocedor de voz con frases largas. Un trabajo similar es mostrado en Gales y Young (1996), donde se reconocen comandos de longitud media y se ataca el problema de interferencia producida por el ruido externo al momento de tomar las señales de audio.

Cabe mencionar que muchos de los trabajos desarrollados para reconocimiento del habla, están pensados en sistemas robóticos, ya sea para manipulación de objetos o movimiento del mismo. Schuller y otros (2008) implementan un reconocedor de emociones por medio de la voz para controlar un brazo robótico de asistencia y manipular objetos médicos. Zhang y otros (2008) desarrollan un sistema de interacción para un robot de servicio especializado en el cuidado de la salud de personas enfermas, el cual puede ser utilizado en entornos caseros. Por su parte Atrash y otros (2009) realizan una interfaz de dialogo humano-robot para comunicarse con una silla eléctrica y poder desplazarla en entornos cerrados.

El reconocimiento de emociones e identificación de usuarios es una de las posibles aplicaciones del procesamiento de voz, sobre todo porque los humanos tendemos a expresar nuestro estado emocional a través de la voz. Muchas de las investigaciones realizadas a la fecha se centran en determinar el estado de animo de un usuario, por ejemplo Luengo y otros (2005), donde se utilizan parámetros prosódicos para determinar las emociones de un usuario en específico. Otro ejemplo se presenta en Solís-Villarreal (2011), donde el autor clasifica una serie de emociones a partir de señales de voz usando soporte vectorial y memorias asociativas. En lo que respecta a la identificación de usuarios se tiene el trabajo desarrollado por Cruz-Beltrán y Acevedo-Mosqueda (2008); ellos realizaron la identificación de usuarios mediante Redes Neuronales y Coeficientes de Predicción Lineal (LPC), al igual que en Trujillo-Romero y otros (2012).

En lo que respecta al español mexicano, varias han sido las investigaciones realizadas, muchas de ellas utilizando la herramienta HTK desarrollada por la universidad de Cambridge (Young y Woodland, 2006). Además, gran parte de esas investigaciones se ha implementado utilizando sílabas y fonemas como unidades básicas para el modelado de la voz (Oropeza-Rodríguez y Suárez-Guerra, 2006). Se puede mencionar el trabajo de Flores y otros (2001), donde se realiza una síntesis de voz utilizando métodos de selección de unidades de longitud variable. Por otro lado, se puede hacer énfasis en el proyecto de la Universidad Nacional Autónoma de México, quienes desean implementar una interacción natural con el robot GOLEM por medio de diálogos cognitivos con el sistema (Golem, 2012; Avilés y otros, 2009).

Capítulo 2

Marco Teórico

2.1. Introducción

Uno de los últimos objetivos en la interacción humano-robot consiste en conseguir una comunicación natural y sin esfuerzo. Las nuevas tecnologías permiten que, más allá de la interacción existente a través del teclado y del ratón, surjan nuevas modalidades de comunicación entre un sistema robótico y el usuario que lo maneja. Dos de los canales más usados para este tipo de comunicación son las expresiones visuales obtenidas a partir de una imagen o un video y las expresiones léxico-fonéticas obtenidas de un discurso. Para que el sistema robótico consiga establecer una interacción adecuada, éste debe ser capaz de procesar y analizar dichas expresiones por medio de técnicas vinculadas al reconocimiento automático de patrones.

Siguiendo la definición de Watanabe (1985), un patrón es una entidad a la que se le puede dar un nombre y que está representada por un conjunto de propiedades, medidas y las relaciones entre ellas (vector de características). Por ejemplo, un patrón puede ser una señal sonora y su vector de características, el conjunto de coeficientes espectrales extraídos de ella (espectrograma). Otro ejemplo podría ser la imagen de una cara humana, de la cual se extrae el vector de características formado por un conjunto de valores numéricos, calculados a partir de la misma. El reconocimiento automático, descripción, clasificación y agrupamiento de patrones son actividades importantes en una gran variedad de disciplinas científicas, tales como biología, psicología, medicina, visión por computador, inteligencia artificial, teledetección, etc.

Un sistema de reconocimiento de patrones tiene uno de dos objetivos principales. Identificar un patrón como miembro de una clase ya definida, también llamada clasificación supervisada. O bien, asignar un patrón a una clase que aún no ha sido definida, comúnmente llamada clasificación no supervisada, agrupamiento o *clustering*.

Recordemos que el objetivo del presente trabajo de tesis consiste en desarrollar un sistema de reconocimiento multimodal. Capaz de reconocer expresiones visuales (señas) y fonéticas (comandos vocales), con el fin de controlar un sistema robótico de servicio. Debe identificar las señas emitidas por el usuario, haciendo uso de Redes Neuronales Artificiales y utilizar Modelos Ocultos de Markov para reconocer los comandos de voz. Ambas técnicas, identificadas como métodos de reconocimiento de patrones con clasificación supervisada.

En este capítulo daremos referencias al estado actual de las técnicas utilizadas para el reconocimiento de señas y comandos vocales que pretendemos abordar en la tesis, con el objetivo de situar al lector en el contexto adecuado.

2.2. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) son métodos diseñados para el procesado de datos y la organización del conocimiento basado en la imitación del funcionamiento de los sistemas nerviosos biológicos. Una Red neuronal no se basa en un modelo algebraico explícito, sino en un conjunto de unidades de activación, denominados también "nodos" o "neuronas artificiales" conectadas unas con otras. Estas conexiones tienen una gran semejanza con las dendritas y los axones en los sistemas nerviosos biológicos (Hagan y otros, 1995).

Como se muestra en la Figura 2.1, las neuronas biológicas tiene cuatro componentes básicos: D endritas, Soma, Axón, y Sinapsis. A través de las Dendritas, las neuronas reciben señales provenientes de estímulos externos o de otras neuronas, el Soma las combinas mediante una operación no lineal, y finalmente genera un resultado que se comunica a otras neuronas vecinas a través del Axón y las Sinapsis. Aunque el mecanismo de funcionamiento es mucho más complejo, lo expuesto es suficiente para entender el origen de las RNA.

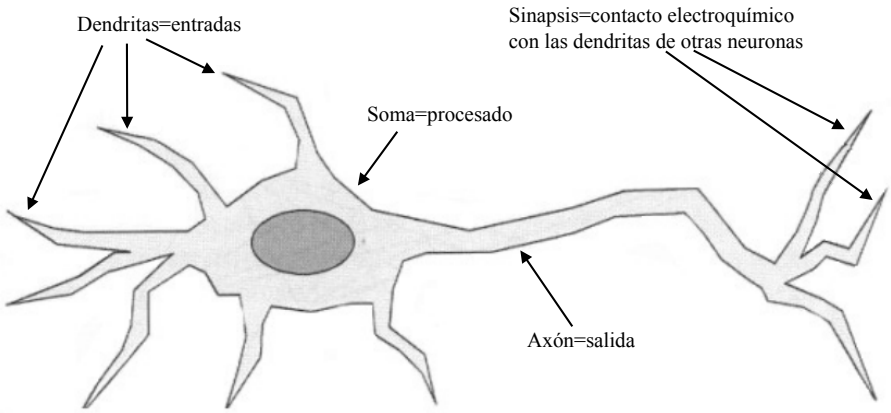


Figura 2.1: Esquema de una neurona biológica mostrando sus cuatro componentes básicos.

Las RNA al margen de "parecerse" al cerebro presentan una serie de características propias del mismo. Por ejemplo, las RNA aprenden de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos. La neurona artificial, simula las funciones básicas de la neurona natural. Ésta contiene dos algoritmos, uno de ellos calcula la suma ponderada de los valores que llegan por las conexiones de entrada (función de propagación), el otro denominado "función de transferencia", genera una respuesta o salida que se envía a otras neuronas. La red de neuronas que se forma es capaz de aprender, principalmente mediante el ajuste de "peso" en las conexiones entre neuronas, hasta que la red en su conjunto proporciona predicciones con una precisión satisfactoria. En la Figura 2.2, se muestra un esquema de la neurona artificial mostrando las similitudes con la neurona biológica.

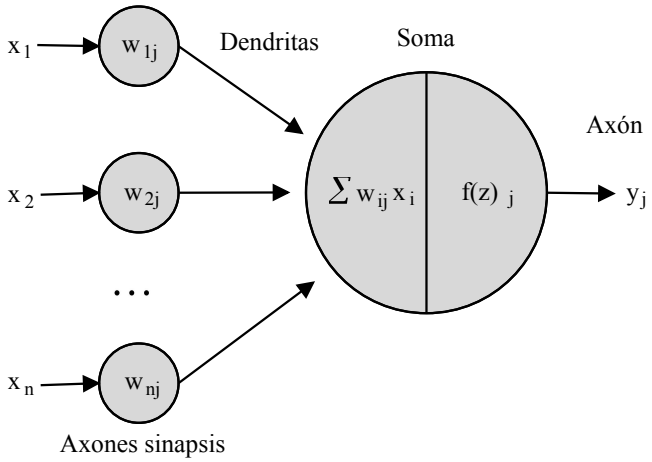


Figura 2.2: Esquema de una neurona artificial.

Según Haykin (1999), una red neuronal artificial puede definirse como un procesador distribuido masivamente, que tiene una propensión natural para almacenar conocimiento experimental y hacerlo disponible para su uso. Además, el conocimiento es adquirido mediante un proceso de aprendizaje y las fuerzas de conexión entre neuronas, conocidas como "pesos sinápticos", son utilizados para almacenar conocimiento.

Cronológicamente, Frank Rosenblatt fue uno de los pioneros en el área de las RNA, pues en 1957 publicó un elemento llamado "Perceptrón" (Rossenblatt, 1957). Dicho perceptrón era un sistema clasificador que podía identificar patrones geométricos y abstractos. El primer perceptrón era capaz de aprender algo y era robusto, de forma que su comportamiento variaba sólo si resultaban dañados los componentes del sistema. Además, presentaba la característica de ser flexible y comportarse correctamente después de que algunas celdas fueran destruidas.

2.2.1. Red neuronal multicapa

Actualmente, las Redes Neuronales se caracterizan porque tiene todas sus neuronas agrupadas en distintos niveles llamados capas. Existen dos capas con conexiones hacia los datos del mundo exterior. Una capa de entrada, donde se presentan los datos a la red, y una capa de salida que emite la respuesta de la red a una entrada. El resto de las capas reciben el nombre de capas ocultas. La salida de cada neurona se propaga por igual por estas conexiones hasta las neuronas destino. Cada conexión tiene un peso asociado que pondera el valor numérico de la señal que viaja por ésta. Así pues, una red de neuronas artificial puede verse como un grafo cuyos nodos tienen funcionamiento similar, los cuales propagan la información a través de las distintas conexiones. En la Figura 2.3, se muestra el aspecto de una red neuronal artificial compuesta de tres capas (red neuronal multicapa).

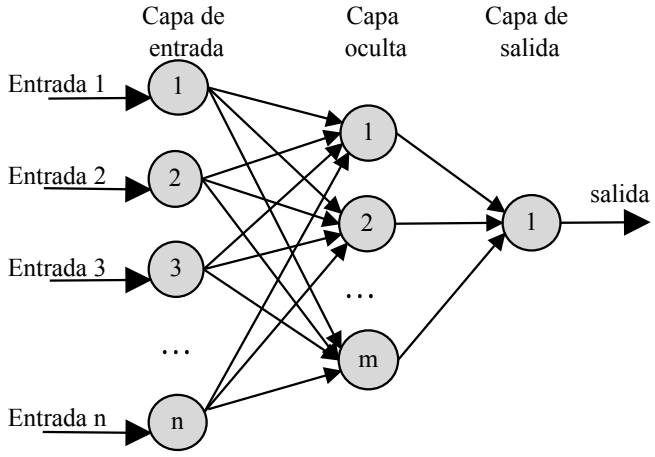


Figura 2.3: Representación de una red neuronal multicapa.

Una red neuronal multicapa o perceptrón multicapa presenta las conexiones de sus neuronas hacia adelante (RNA *feed-fordward*). Generalmente todas las neuronas de un nivel se conectan con todas las neuronas de la capa inmediatamente posterior. En algunas ocasiones, dependiendo de la red, se encuentran conexiones de neuronas que no están en niveles consecutivos, o alguna de las conexiones entre dos neuronas de niveles consecutivos no existe, es decir, el peso asociado a dicha conexión es constante e igual a cero. Además, todas las neuronas de la red tienen un valor umbral asociado. Se suele tratar como una entrada cuyo valor es constante e igual a uno, y lo único que varía es el peso asociado a dicha conexión.

La función no lineal que se aplica a la salida de cada neurona se conoce como función de trasferencia y debe cumplir únicamente con dos condiciones: ser continua y diferenciable. Las funciones más utilizadas para este propósito son:

- Signo. Con ella se planteó el modelo del perceptrón de Rossenblatt (Rossenblatt, 1957).

La función es definida por:

$$sign(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x \leq 0 \end{cases} \tag{2.1}$$

Sin embargo, la función no es diferenciable en 0. Por tanto, se dejo de lado y de ella derivaron las funciones sigmoide y tangente hiperbólica.

- Sigmoide. Toma valores entre 0 y +1 para una variación de la variable independiente x entre $+\infty$ y $-\infty$. Su expresión esta dada por:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

- Tangente hiperbólica. De manera similar a lo que ocurre con la función sigmoide, la tangente hiperbólica toma valores entre -1 y $+1$ para una variación de la variable independiente entre $+\infty$ y $-\infty$. En este caso, -1 codifica la mínima actividad de la neurona y $+1$ la máxima actividad de la misma. Esta dada por la siguiente expresión:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{2.3}$$

- Función lineal a tramos. En este caso, se utilizan tres funciones lineales para formar una función no lineal. Lo más habitual es que la función tome valores entre -1 y $+1$ como se muestra en la ecuación 2.4. No obstante, puede considerarse una función que varíe entre 0 y $+1$ como lo expresado en la ecuación 2.5.

$$f(x) = \begin{cases} -1 & \text{si } x < -1 \\ x & \text{si } -1 < x < 1 \\ 1 & \text{si } x > 1 \end{cases} \tag{2.4}$$

$$f(x) = \begin{cases} 0 & \text{si } x < -1 \\ 0.5 + 0.5x & \text{si } -1 < x < 1 \\ 1 & \text{si } x > 1 \end{cases} \tag{2.5}$$

En cuanto a la forma de disponer las neuronas, se tiene un gran número de posibilidades. El número de neuronas que forman las capas de entrada y salida están determinadas por el problema, mientras que el número de capas ocultas y neuronas en cada una de ellas no puede ser determinada por ninguna regla teórica, por lo que el diseñador es quién determina esta arquitectura. Únicamente, está demostrado que dado un conjunto de datos conexo, con una sola capa oculta es posible establecer una relación entre el conjunto de datos, aunque no está especificado el número de neuronas necesarias. En otro caso, si el conjunto de datos no es conexo, deben haber al menos dos capas ocultas (Kolmogorov, 1957).

2.2.2. Proceso de aprendizaje

Asociado a cada tipo de red existe al menos un algoritmo de aprendizaje. Dicho algoritmo, consiste en un método sistemático para encontrar un valor adecuado de los pesos sinápticos de la red. En términos generales, se basan en definir una función objetivo implícita o explícita, que represente de forma global el estado de la red. A partir de ella, los pesos asignados inicialmente van evolucionando a unos valores que llevan dicha función a un mínimo (estado estable de la red). El aprendizaje, por tanto, consiste en hallar los valores precisos de dichos pesos para resolver un problema en específico.

El objetivo del aprendizaje o entrenamiento de una RNA es conseguir que una aplicación determinada, para un conjunto de entradas, produzca el conjunto de salidas deseadas o por lo menos consistentes. El proceso de entrenamiento consiste en la aplicación secuencial de diferentes conjuntos o vectores de entrada para que se ajusten los pesos de las interconexiones según un procedimiento predeterminado. Durante la sesión de entrenamiento los pesos convergen gradualmente hacia los valores que hacen que cada entrada produzca el vector de salida deseado.

Los algoritmos de entrenamiento o los procedimientos de ajuste de los valores de las conexiones de las RNA se pueden clasificar en dos grupos: Supervisado y No Supervisado.

- Aprendizaje Supervisado. Estos algoritmos requieren el emparejamiento de cada vector de entrada con su correspondiente vector de salida. El entrenamiento consiste en presentar un vector de entrada a la red, calcular la salida de la red, compararla con la salida deseada, y el error o diferencia resultante se utiliza para realimentar la red y cambiar los pesos de acuerdo con un algoritmo que tiende a minimizar el error.
- Aprendizaje No Supervisado. El conjunto de vectores de entrenamiento consiste únicamente en vectores de entrada. El algoritmo de entrenamiento modifica los pesos de la red de forma que produzca vectores de salida consistentes. El proceso de entrenamiento extrae las propiedades estadísticas del conjunto de vectores de entrenamiento y agrupa en clases los vectores similares.

Existen diferentes algoritmos de aprendizaje que optimizan las conexiones entre las neuronas según el error que esté cometiendo la red, entendiendo por error la diferencia que existe entre la salida ofrecida por la red y la salida deseada. Los más utilizados son los algoritmos por descenso de gradiente que se basan en la minimización o maximización de una determinada función. Generalmente, se minimiza una función monótona creciente del error, por ejemplo, el valor absoluto del error o el error cuadrático medio. Dicha función a minimizar se denomina función de coste.

La función de coste más utilizada es la correspondiente al error cuadrático medio (Bishop, 1996), la cual es definida como sigue:

$$J = \frac{1}{2M} \sum_{i=1}^{M} \sum_{j=1}^{N} e_j^2(i) = \frac{1}{2M} \sum_{i=1}^{M} \sum_{j=1}^{N} (d_j(i) - y_j(i))^2 \tag{2.6}$$

Donde M es el número de patrones usados para entrenar la red, N el número de neuronas en la capa de salida, $d_j(i)$ es la j-ésima salida deseada de la red para el *i-ésimo* patrón de entrenamiento y $y_j(i)$ es la j-ésima salida ofrecida por la red para el *i-ésimo* patrón de entrenamiento. Sin embargo, existen también la función de coste entrópica y la usada en la norma de Minkowski, ambas descritas en Bishop (1996).

Una vez definida la función de coste, se debe aplicar el procedimiento para minimizar dicha función, este proceso recibe el nombre de aprendizaje o entrenamiento de la red. De ellos, existen dos tipos:

- *On-Line*. El aprendizaje se realiza patrón a patrón. Durante todo el entrenamiento, se proporciona a la red cada una de las entradas con su salida deseada. Se mide el error y en función de éste se adaptan los pesos sinápticos mediante el algoritmo de aprendizaje seleccionado.
- *Off-Line*. El aprendizaje se realiza por épocas. Una época supone el paso de todos los patrones de entrenamiento de la red. Se mandan a la red todos los patrones de entrenamiento, se obtiene el error total cometido y se adaptan los pesos en función de este valor promediado según el número de patrones (Haykin, 1999).

En cualquier caso, la red aprende utilizando ejemplos, pero lo realmente atractivo de estos sistemas es la capacidad de generalización. Esto se refiere a la calidad de la respuesta ante entradas que no han sido utilizadas en su entrenamiento. Por tanto, cabe distinguir dos modos o fases de funcionamiento de una RNA: "Entrenamiento" y "Reconocimiento". Así pues, una vez fijados los pesos en la fase de entrenamiento, la red pasa a la fase de reconocimiento, donde procesa entradas correspondientes a la aplicación real.

2.2.3. Entrenamiento *Backpropagation*

El primer desarrollo para entrenamiento de una red neuronal multicapa fue descrita en la tesis de Paul Werbos en 1974 (Werbos, 1974). En esta tesis se presentaba un algoritmo de entrenamiento para Redes Neuronales de contexto general. Sin embargo, fue hasta mediados de los años 80's cuando Rumelhart y otros (1986), entre otros investigadores (Parker, 1985; LeCun, 1985), retomaron el algoritmo y comenzaron a popularizarlo en diferentes publicaciones con el nombre de *backpropagation*.

El algoritmo de aprendizaje *backpropagation* es un algoritmo de descenso por gradiente que retropropaga las señales desde la capa de salida hasta la capa de entrada, optimizando los

valores de los pesos sinápticos mediante un proceso iterativo que se basa en la minimización de la función de coste. Por ello, el algoritmo puede dividirse en dos fases.

1. Propagación hacia adelante. El vector de entrada o patrón de entrada es presentado a los nodos de la capa de entrada. Las señales se propagan desde la capa de entrada hasta la capa de salida, capa por capa. Se determina la salida de la red y el error cometido al comparar ésta con el valor deseado en la salida de la red neuronal.

2. Propagación hacia atrás. En función de los errores cometidos en la capa de salida, el algoritmo se encarga de optimizar los valores de los pesos sinápticos desde la capa de salida hasta la capa de entrada. Se retropropaga el error de la capa de salida a la capa de entrada a través de las capas ocultas sucesivas.

En la Figura 2.4 se representa la última capa oculta y la capa de salida de una red neuronal multicapa. En ella, se identifican los dos tipos de señales y su propagación dentro de la red (Parker, 1985).

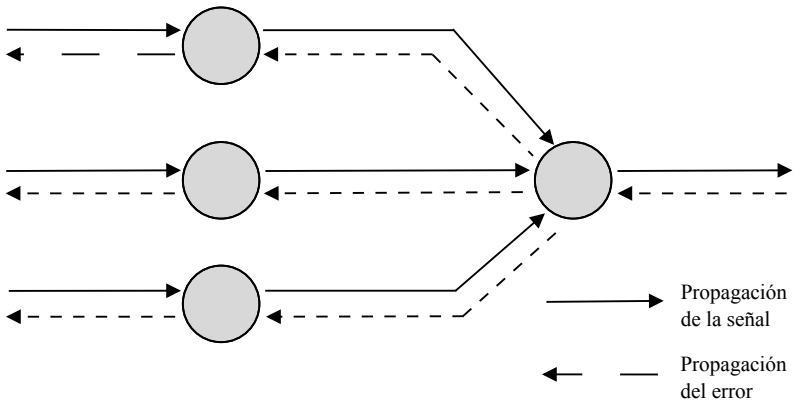


Figura 2.4: Propagación de las señales del algoritmo *backpropagation*.

Dado que se desea minimizar la función de coste, el problema se vuelve no lineal y como tal, el problema de minimización de la función error se resuelve por técnicas de optimización no lineales que se basan en ajustar los parámetros siguiendo una determinada dirección. En este método, la dirección elegida es la negativa al gradiente de la función error.

A partir de aquí existen dos opciones. Se pueden cambiar los parámetros cada vez que se introduce un patrón de entrenamiento (*on-line*), o solamente cambiarlos cuando se hayan introducido todos los parámetros de entrenamiento por cada época (*off-line*). De acuerdo con Haykin (2009), el algoritmo *backpropagation* presenta un mejor desempeño de entrenamiento actualizando los pesos por medio del método *on-line*. Para este modo de operación y siguiendo el desarrollo presentado en Haykin (1999, 2009), los ciclos del algoritmo dado el conjunto de entrenamiento $\{x(m), d(m)\}_{m=1}^{M}$, se pueden resumir en las siguientes cinco fases:

1. Inicialización. Asumiendo que no existe información a priori disponible, se establecen los pesos sinápticos con valores aleatorios entre 0 y +1, o entre −1 y +1.

2. Presentación de las muestras de entrenamiento. Se presenta a la red neuronal el conjunto completo de vectores o patrones de entrenamiento. Por cada vector de entrenamiento, siguiendo un orden específico, se realiza el cálculo de propagación hacia adelante y hacia atrás como se muestra en las fases tres y cuatro respectivamente.

3. Cálculo de propagación hacia adelante. Dado el conjunto de entrenamiento $\{x(m), d(m)\}$, donde $x(m)$ es el vector de entrada o vector de entrenamiento de la red neuronal y $d(m)$ la salida deseada de la red. Se calculan los valores de propagación y transferencia de la red capa por capa con dirección hacia adelante. La función de propagación o suma ponderada de las entradas $v_j^{(l)}(m)$ para la neurona j de la capa l es definida como:

$$v_j^{(l)}(m) = \sum_{i=0}^{P} w_{ji}^{(l)}(m) y_i^{(l-1)}(m) \tag{2.7}$$

Donde P es el número de neuronas en la capa anterior $l-1$, $y_i^{(l-1)}(m)$ es la señal de salida de la neurona i en la capa $l-1$ para el vector de entrenamiento m y $w_{ji}^{(l)}(m)$ es el peso sináptico que conecta las neuronas j de la capa l e i de la capa $l-1$. Para $i=0$, se tiene que $y_0^{(l-1)}(m) = +1$ y $w_{j0}^{(l)}(m) = b_j^{(l)}$, donde éste último es el umbral aplicado a la neurona j en la capa l.

Asumiendo el uso de una función sigmoidal como función de transferencia, la señal de salida de la neurona j en la capa l es definida por:

$$y_j^{(l)}(m) = \frac{1}{1 + e^{-v_j^{(l)}(m)}} \tag{2.8}$$

Si la neurona j está en la primera capa de la red, entonces:

$$y_j^{(0)}(m) = x_j(m) \tag{2.9}$$

Donde $x_j(m)$ es el *j-ésimo* elemento del vector de entrada $x(m)$. Por otra parte, si la neurona j está en la capa de salida, entonces:

$$y_j^{(L)}(m) = o_j(m) \tag{2.10}$$

En tal caso, $o_j(m)$ es el *j-ésimo* elemento del vector de salida proporcionado por la red. Con ello, se puede calcular el error entre el vector deseado y el vector obtenido como:

$$e_j(m) = d_j(m) - o_j(m) \tag{2.11}$$

en el cual, $d_j(m)$ es el *j-ésimo* elemento del vector de respuestas deseado $d(m)$.

4. Cálculo de propagación hacia atrás. En esta fase se realiza el cálculo de los gradientes de la red y se calcula capa por capa. Si la neurona j está en la capa de salida L, el gradiente se define como:

$$\delta_j^{(L)}(m) = e_j^{(L)}(m) o_j(m)[1 - o_j(m)] \tag{2.12}$$

En otro caso, si la neurona j se encuentra en una capa oculta l, el gradiente es calculado como:

$$\delta_j^{(l)}(m) = y_j^{(l)}(m)[1 - y_j^{(l)}(m)] \sum_{k=1}^{P} \delta_k^{(l+1)}(m) w_{kj}^{(l+1)}(m) \tag{2.13}$$

Donde P es el número de neuronas en la capa $l + 1$. El ajuste de los pesos sinápticos de la red en la capa l se realiza de acuerdo a la regla delta como sigue:

$$w_{ji}^{(l)}(m + 1) = w_{ji}^{(l)}(m) + \eta \delta_j^{(l)}(m) y_i^{(l-1)}(m) \tag{2.14}$$

Donde η es el índice de aprendizaje de la red neuronal.

5. Iteraciones. Las iteraciones se realizan siguiendo las fases tres y cuatro con nuevos patrones de entrenamiento hasta que los parámetros libres o pesos sinápticos se estabilizan en un punto donde la función de coste (ecuación 2.6) alcanza un valor pequeño aceptable, generalmente definido por el investigador.

Como se mencionaba con anterioridad, estas cinco fases realizan un entrenamiento *on-line* de la red neuronal, pues por cada patrón de entrenamiento se actualizan todos los pesos sinápticos. No obstante, la generalización a entrenamiento tipo *off-line* es prácticamente inmediata. En la Figura 2.5, se muestran las variables involucradas en el algoritmo *backpropagation* y su relación con el resto de los elementos de la red neuronal.

2.2.4. Variantes del algoritmo *Backpropagation*

El algoritmo de aprendizaje *Backpropagation* es el más conocido para el entrenamiento de Redes Neuronales. Además, es un algoritmo de máximo descenso que busca minimizar la función de coste J, dada por un error $e_j(i)$. Sin embargo, en la práctica su uso se ve muy limitado debido a algunos inconvenientes que presenta.

- Baja velocidad de convergencia. Cuando la naturaleza del error es no lineal y sólo se encuentra disponible la información del gradiente. El coeficiente de aprendizaje η debe mantenerse bastante pequeño para asegurar una convergencia estable. Esto, en consecuencia, aumenta el tiempo del proceso de aprendizaje.
- Mínimos locales. El parámetro del error en algún momento puede presentar mínimos locales, los cuales provocarán que el algoritmo de aprendizaje, por ser descendiente, llegue a detenerse.

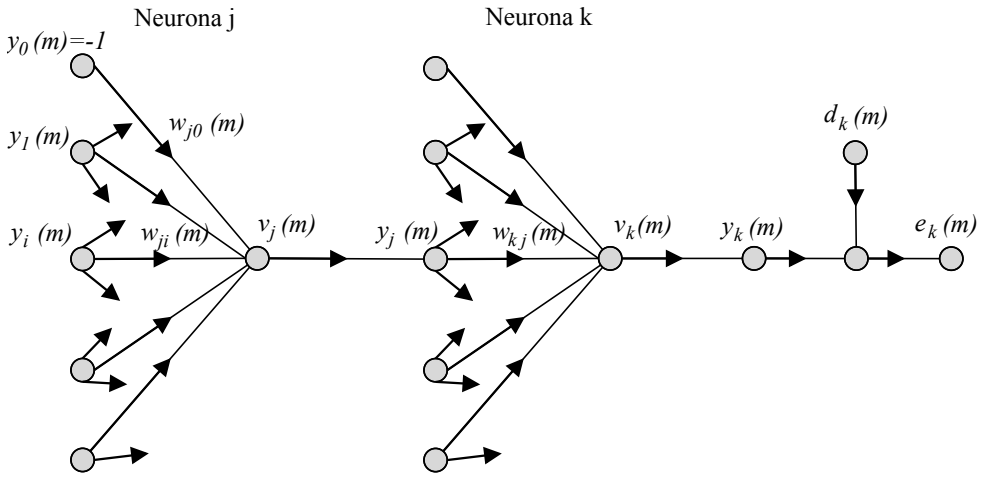


Figura 2.5: Variables involucradas en el algoritmo *backpropagation*.

- Oscilaciones. En un caso extremo donde el error presenta un comportamiento en forma de "onda" inclinandose suavemente hacia el mínimo real, el algoritmo de máximo descenso puede entrar en una situación oscilatoria, saltando de un lado a otro en vez de ir progresivamente hacia un mínimo.
- El tamaño del paso. Dado un coeficiente de aprendizaje fijo, el tamaño de paso o cambio en los pesos, es directamente proporcional a la magnitud del gradiente. Entonces, el paso es largo cuando la pendiente del error es grande y corto cuando la pendiente del error es pequeña. Pero cuando el error presenta altiplanicies y declives que cambian rápidamente, un coeficiente de aprendizaje fijo conlleva a un bajo rendimiento en el aprendizaje de la red.

Debido a esto, se han introducido mejoras del algoritmo para acelerar la convergencia del mismo. En esta sección describiremos brevemente las más importantes, las cuales se pueden clasificar en modificaciones heurísticas de *backpropagation* y técnicas de optimización numérica. Las primeras incluyen el uso de momento y razón de aprendizaje variable. Las segundas, la implementación de gradiente conjugado y el algoritmo de Levenberg-Marquardt.

Momento

La regla delta generalizada se basa en la búsqueda del mínimo de la función error mediante el descenso del gradiente de la misma. Esto puede llevar a un mínimo local de la función error, donde el gradiente vale cero, y por lo tanto los pesos no se ven modificados, pero el error cometido por la red es significativo. Esta variante es muy similar al *backpropagation* clásico

ya que el incremento de pesos es igual al gradiente de la función de error con signo negativo, pero además se le añade un termino que es el incremento de pesos anterior, es decir:

$$w_{ji}^{(l)}(m+1) = w_{ji}^{(l)}(m) - \eta\delta_j^{(l)}(m)y_i^{(l-1)}(m) + \alpha[w_{ji}^{(l)}(m-1)] \tag{2.15}$$

Donde α es la constante de momento, que puede tomar valores en el intervalo $(0, 1)$. La constante de momento es la encargada del nuevo incremento en el valor de w_{ij} en relación al incremento previo del mismo peso. Este nuevo término controla la velocidad de acercamiento al mínimo, acelerandola cuando se está lejos y alentandola cuando se está cerca.

Razón de aprendizaje variable

La tasa o razón de aprendizaje η juega un papel muy importante en el comportamiento de los algoritmos de aprendizaje. Si es pequeño, la magnitud del cambio de los pesos sinápticos será pequeña y por lo tanto tardará mucho en converger. Si es demasiado grande el algoritmo oscilará y dificilmente encontrará un mínimo de la función error. En algunos casos se ha demostrado que el valor óptimo de la tasa de aprendizaje, para una convergencia rápida es el valor inverso del mayor autovalor de la matriz Hessiana H. Pero, computacionalmente este proceso es ineficiente, ya que para obtener la matriz H es necesario evaluar las segundas derivadas de la función error o función de coste. Por ello, se emplean técnicas heurísticas que van variando el valor de la tasa de aprendizaje en cada iteración. Algunas de éstas son:

- Incrementar la tasa de aprendizaje cuando el gradiente $\delta_j(i-1)$ es próximo al gradiente $\delta_j(i)$, así como disminuirla en caso contrario.
- Multiplicar la tasa de aprendizaje por un valor mayor a uno si los gradientes actual y previo tienen el mismo signo, o por un valor entre cero y uno en caso contrario.
- Multiplicar la tasa de aprendizaje por una cantidad mayor que uno cuando haya decrecido la función error, con el fin de avanzar más rápidamente, y multiplicarla por una cantidad menor que uno en caso contrario. Cabe mencionar que este heurístico es el que utiliza Matlab en el entrenamiento de tipo "traingda".

Gradiente conjugado

El método del gradiente conjugado utiliza un vector de dirección, el cual es una combinación lineal de vectores dirección previstos y el actual vector gradiente negativo. Pretende reducir el comportamiento oscilatorio y reforzar el ajuste de los pesos sinápticos en concordancia con el camino de los vectores de dirección previos.

Se $p(m)$, el vector dirección de la iteración m del algoritmo. El vector de pesos sinápticos es actualizado conforme a la regla delta:

$$w(m+1) = w(m) + \eta(m)p(m) \tag{2.16}$$

Donde $\eta(m)$ es el coeficiente de aprendizaje en la iteración m. El vector inicial de la dirección es igual al negativo del vector gradiente en el punto $m = 0$, esto es:

$$p(0) = -g(0) \tag{2.17}$$

Los vectores sucesivos de dirección son calculados como una combinación lineal entre el actual vector gradiente y el vector dirección anterior.

$$p(m+1) = -g(m+1) + \beta(m)p(m) \tag{2.18}$$

Donde $\beta(m)$ es un parámetro que varía con el tiempo y es calculado a partir de $g(m)$ y $g(m+1)$. Su dirección es obtenida a partir de la formula de *Fletcher-Reeves* (Fletcher y Reeves, 1964) como se muestra a continuación:

$$\beta(m) = \frac{g^T(m+1)g(m+1)}{g^T(m)g(m)} \tag{2.19}$$

Algunos autores demuestran que el aprendizaje basado en el método del gradiente conjugado requiere menos épocas para converger que el algoritmo estándar, aunque es computacionalmente más complejo (Johansson y otros, 1991).

Levenberg-Marquardt

Cuando la función de coste tiene la forma de suma de cuadrados, como lo es el caso del error cuadrático medio (ecuación 2.6), la matriz Hessiana puede ser aproximada como:

$$H = J^T J \tag{2.20}$$

y el gradiente puede ser expresado como:

$$g = J^T e \tag{2.21}$$

Donde J es la matriz Jacobiana, que contiene las primeras derivadas de los errores de la red con respecto a los pesos y el bias, e es el vector de errores de la red. En este sentido,

la matriz Jacobiana puede ser obtenida con el método *backpropagation* y además es menos compleja que la matriz Hessiana (Hagan y Menhaj, 1994).

Por tanto, el algoritmo de Levenberg-Marquardt utiliza la aproximación de la matriz Hessiana para actualizar los pesos sinápticos, de acuerdo a la siguiente expresión:

$$w(m+1) = w(m) - [J^T J + \mu I]^{-1} J^T e \tag{2.22}$$

Donde μ es un escalar que se decrementa con cada paso exitoso del algoritmo y se incrementa cuando un paso puede incrementar el valor de la función error. I es la matriz identidad.

2.3. Modelos Ocultos de Markov

Un Modelo Oculto de Markov (Hidden Markov Model, HMM) es un proceso estocástico que consta de un proceso de Markov no observado (oculto) $\mathbf{q} = \{q_t\}_{t \in N}$ y un proceso observado $\mathbf{O} = \{o_t\}_{t \in N}$, cuyos estados son dependientes estocásticamente de los estados ocultos, es decir, es un proceso bivariado (**q,O**). Estadísticamente, se entiende como proceso estocástico al concepto matemático que sirve para caracterizar una sucesión de variables aleatorias, también llamadas estocásticas, que evolucionan en función de otra variable, generalmente en el tiempo. Cada una de las variables aleatorias del proceso tiene su propia función de distribución de probabilidad y, entre ellas, pueden estar correlacionadas o no. Los HMM's se pueden considerar también como sistemas generativos estocásticos, los cuales se emplean en la modelación de series de tiempo. Estos, se introdujeron inicialmente a finales de la década de 1960 y principios de los años 1970. Los métodos estadísticos de los Modelos Ocultos de Markov, se han vuelto más populares en los últimos años debido a dos razones principales (Rabiner, 1989):

- Los modelos son muy ricos en estructura matemática y pueden formarse las bases teóricas para usarse en un amplio rango de aplicaciones.
- Los modelos al implementarse apropiadamente para diversas aplicaciones, trabajan muy bien en la práctica.

2.3.1. Cadenas de Markov

Las cadenas de Markov se utilizan normalmente para modelar aquellos procesos aleatorios que requieren de cierta memoria. Sea $S_1, S_2, \ldots, S_N$ una secuencia de variables aleatorias cuyos valores se representan mediante un alfabeto de símbolos finito $\chi = \{1, 2, \ldots, c\}$, y aplicando la fórmula de Bayes

$$P(S_1, S_2, \ldots, S_N) = \prod_{i=1}^{N} P(S_i|S_1, S_2, \ldots, S_{i-1}) \tag{2.23}$$

Se dice que las variables aleatorias forman una cadena de Markov si:

$$P(S_i|S_1, S_2, \ldots, S_{i-1}) = P(S_i|S_{i-1}) \quad \forall \quad i \tag{2.24}$$

Y como consecuencia, en una cadena de Markov:

$$P(S_1, S_2, \ldots, S_N) = \prod_{i=1}^{N} P(S_i|S_{i-1}) \tag{2.25}$$

En este caso, los procesos aleatorios tienen una capacidad de memoria tan limitada que el valor de la variable en el instante de tiempo t depende únicamente del valor inmediatamente anterior y de ningún otro. Las cadenas de Markov serán invariantes en el tiempo si a pesar del valor del índice temporal i se cumple:

$$P(S_i = s'|S_{i-1} = s) = P(s'|s) \quad \forall \quad s, s' \in \chi \tag{2.26}$$

donde $p(s'|s)$ resulta ser la función de transferencia que puede tomar forma de matriz $c \times c$. Además, $p(s'|s)$ debe satisfacer las condiciones típicas para todo $s \in \chi$.

$$\sum_{s' \in \chi} p(s'|s) = 1, \qquad p(s'|s) \leq 1, \quad s' \in \chi \tag{2.27}$$

Suponiendo que S_i son los estados de una cadena de Markov, ésta resultaría en un proceso de estados finitos cuyas transiciones entre los estados serían definidas por la función de transferencia $P(s'|s)$. Un ejemplo de cadena de Markov de 2 estados (N=2) es la que se muestra en la Figura 2.6.

2.3.2. Definición de Modelos Ocultos de Markov

Un Modelo Oculto de Markov es una cadena de estados s junto con un proceso estocástico que toma valores en un alfabeto χ y el cual depende de s. Se puede definir un HMM como un autómata de estados finitos y estocástico caracterizado por los siguientes parámetros (Rabiner, 1989):

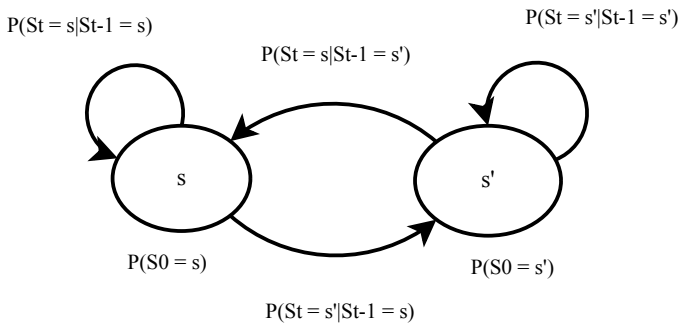


Figura 2.6: Cadena de Markov de dos estados.

1. Número de estados del modelo (N). Se refiere a la secuencia de estados que conforman el modelo como $S = (S_1, S_2, \ldots, S_N)$, y al estado en el instante de tiempo t como q_t. Estos estados normalmente permanecen ocultos pero tienen asociada alguna magnitud o característica física.

2. Número de símbolos distintos por estado $\chi = \{v_1, v_2, \ldots v_M\}$, o número de fuentes gaussianas que participan en función de la densidad de probabilidad conjunta.

3. Matriz de probabilidad de transición entre los estados $A = \{a_{ij}\}$, de tamaño $N \times N$, y que define la probabilidad que existe de encontrarse en el estado i en el instante de tiempo $t-1$, para pasar al estado j en t.

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \qquad 1 \leq i, j \leq N \tag{2.28}$$

4. Probabilidades iniciales de los estados $\pi = \{\pi_i\}$, donde se determinan las probabilidades de cualquiera de los estados en el instante inicial:

$$\pi_i = P(q_1 = S_1), \qquad 1 \leq i \leq N \tag{2.29}$$

5. Probabilidades de emisión de símbolos $B = \{b_j(O_t)\}$, uno por cada estado, donde $b_j = (b_{j1}, b_{j2}, \ldots, b_{jM})$ es la probabilidad de emisión del símbolo v_k del alfabeto en el estado j.

Para una completa especificación de los HMM's es necesario definir los dos parámetros que determinan la geometría del modelo como N y M, además de las tres probabilidades A, B, π. Estas últimas representan los parámetros del modelo, todas ellas mediante la notación:

$$\lambda = A, B, \pi \tag{2.30}$$

Otro de los aspectos importantes de esta teoría es la referente a las restricciones que se derivan de un modelo estocástico, es decir:

$$\sum_{i=1}^{N} \pi_i = 1$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad 1 \leq i \leq N$$

$$\sum_{m=1}^{M} b_{jm}(O_t) = 1, \qquad 1 \leq i \leq M \tag{2.31}$$

Una vez definidos los HMM's, se plantean los tres problemas inmediatos que pueden ser resueltos con esta teoría como sigue:

1. Dada la secuencia observada $\mathbf{O} = O_1, O_2, \ldots, O_T$ y el modelo λ, ¿cómo calcular la probabilidad resultante de dicha secuencia $P(\mathbf{O}|\lambda)$, dado el modelo anterior? Este problema se conoce comúnmente con el nombre de *inferencia*.

2. Dada la secuencia de observaciones $\mathbf{O} = O_1, O_2, \ldots, O_T$ y el modelo λ, ¿cómo encontrar la secuencia de estados (ocultos) $Q = q_1, q_2, \ldots, q_T$ que mejor explique la secuencia observada? Este problema se resuelve mediante el algoritmo de Viterbi (Viterbi, 1967).

3. ¿Cómo ajustar los parámetros del modelo $\lambda = \{A, b, \pi\}$ para maximizar $P(\mathbf{O}|\lambda)$? Este problema se conoce con el nombre de *aprendizaje*.

2.3.3. Tipos de HMM's

Un HMM puede ser representado como un grafo dirigido de transiciones/emisiones. La siguiente clasificación no se debe exclusivamente a alguna de sus características. Sin embargo, los dos primeros tipos cuya clasificación depende de los valores de las matrices de probabilidad de transición, son excluyentes entre ellos pero no con el tercer tipo. Así, es posible tener HMM's que sean no ergódicos y autoregresivos o bien ergódicos y autoregresivos al mismo tiempo.

Ergódicos o completamente conectados

Cuando un HMM tiene una matriz de probabilidad de transición de estados completa (es decir, que no es cero para ningún a_{ij}) se dice que el HMM es ergódico. En este tipo de HMM's todos los estados están interconectados, por lo que todos los estados son alcanzables

de manera directa entre ellos. En procesamiento de voz, estos Modelos Ocultos son usados para modelar palabras. La Figura 2.7 muestra un ejemplo de este tipo de HMM.

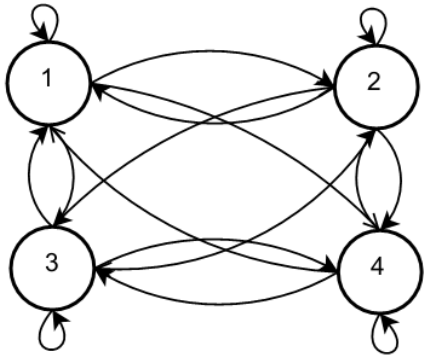


Figura 2.7: HMM Ergódico con 4 estados.

No Ergódicos

En los casos en los que las matrices de transición del HMM pueden tener algunos valores "0", se dice que dichos modelos son "no ergódicos". Por ejemplo, si se tiene una matriz triangular superior, se tendría un HMM como el mostrado en la Figura 2.8. A estos modelos se les conoce también como modelos "izquierda-derecha", pues la secuencia de estados producida por la secuencia de observaciones, siempre deberá proceder desde el estado más a la izquierda, hasta el que esté más a la derecha. Estos HMM's imponen un orden temporal, pues los estados con número menor, generan observaciones que ocurrieron antes que las generadas por los estados con índices mayores. En reconocimiento del habla estas arquitecturas modelan bien los aspectos lineales de las secuencias y son utilizados para modelar fonemas (en especial la estructura izquierda-derecha con tres estados emisores.

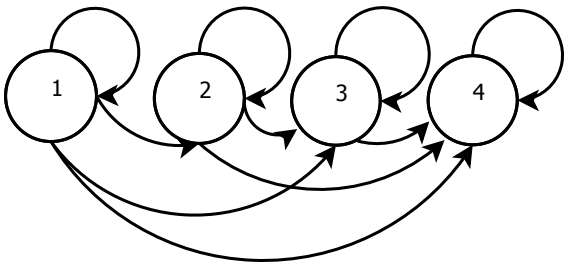


Figura 2.8: HMM No Ergódico con 4 estados.

Autoregresivos

Los HMM autoregresivos, tienen casos especiales del parámetro "B". Cuando los símbolos observables de un HMM son vectores continuos (no son un conjunto discreto como un alfabeto), la función de distribución de probabilidad $b_{jm}(O_t)$, es remplazada por la función continua $b_j(O_t), \quad 1 \leq j \leq N$ donde $b_j(O_t)dt =$ probabilidad de que el vector de observación O se encuentre entre t y $t + dt$. A continuación se muestran las formas especiales de $b_j(O_t)$ que han sido propuestas.

- Gausianos con mezcla de densidades.

$$b_j(O_t) = \sum_{k=1}^{M} c_{jk} N(x, \mu_{jk}, U_{jk}) \tag{2.32}$$

  donde c_{jk} es el peso de la mezcla, N es la distribución normal y μ_{jk}, U_{jk} son los vectores de medias y covarianzas asociados con el estado j y la mezcla k.

- Gausianos autoregresivos con mezcla de densidades

$$b_j(O_t) = \sum_{k=1}^{M} c_{jk} b_{jk}(O_t) \tag{2.33}$$

  donde

$$b_{jk} = \frac{e^{\delta(x;a_{jk})/2}}{(2\pi)^{k/2}}$$

  y

$$\delta(x; a_{jk}) = r_a(0) r_x(0) + 2 \sum_{j=1}^{p} r_a(i) r_x(i)$$

  $\delta(x; a)$ es la distancia LPC estándar entre el vector x (de dimensión K) con autocorrelación r_x y un vector LPC a (de dimensión p) con autocorrelación r_a.

2.3.4. Aprendizaje en los Modelos Ocultos de Markov

El problema más difícil de los HMM's es determinar un método para ajustar los parámetros (A, B, π) del modelo, con la finalidad de satisfacer los criterios de optimización. No se conoce una forma analítica para fijar dichos parámetros, y que además maximicen la probabilidad de la secuencia de observación. Sin embargo, son varios los métodos de entrenamiento no supervisado que existen en la literatura, los cuales se pueden clasificar en dos grupos: *algoritmos de optimización o búsqueda ascendente* (del inglés *hill-climbing*, por ejemplo, EM, k-medias segmentado y búsqueda del gradiente), y *algoritmos de búsqueda global*, tales como algoritmos genéticos y *simulated annealing*.

Los algoritmos de búsqueda ascendente dependen en gran medida de la manera en que se inicialice el modelo, de forma que, en la práctica y si los parámetros iniciales no han sido los óptimos, la búsqueda puede conducir a un modelo sub-óptimo. Para evitar este problema se proponen una serie de técnicas mostradas en Juang y Rabiner (1990), aunque éstas impliquen una mayor carga computacional. Por otra parte, los algoritmos de búsqueda global no dependen en exceso de la inicialización del modelo, precisamente por su capacidad global para encontrar el óptimo.

A continuación se muestran aquellos algoritmos de entrenamiento que se consideran más relevantes:

Algoritmo de Baum-Welch

El algoritmo EM (*Expectation-Maximization*) (Dempster y otros, 1977) es un método general que se utiliza para estimar los parámetros del modelo, de forma que se maximice la probabilidad (*maximum-likelihood*, ML) de una distribución, la cual es generada a partir de un conjunto incompleto de datos observados, es decir, existen datos no conocidos por algún motivo.

Se pueden definir dos aplicaciones principales del algoritmo EM: la primera, cuando efectivamente el conjunto de datos observados resulta incompleto; la segunda, cuando la optimización de la función de probabilidad es demasiado compleja y se necesitan asumir ciertas simplificaciones (similares a la pérdida de información), para resolver el problema de optimización. Esta última aplicación se utiliza normalmente en tareas de reconocimiento de patrones.

El problema de maximización de la función de probabilidad planteado es como sigue: sea $p(x|\Theta)$ la función de densidad de probabilidad parametrizada por el conjunto de valores definidos en Θ; por ejemplo, p podría ser un conjunto de fuentes gaussianas y Θ los valores de las medias y las varianzas que las definen; y sea el conjunto de datos de tamaño N generados por la distribución anterior $\chi = \{x_1, x_2, \ldots, x_N\}$. Asumiendo la independencia de las observaciones, la función de densidad de probabilidad de los datos observados resulta en función de probabilidad que depende de los parámetros en Θ:

$$p(\chi|\Theta) = \prod_{i=1}^{N} p(x_i|\Theta) = L(\Theta|\chi) \tag{2.34}$$

El objetivo del algoritmo EM consiste en encontrar los valores de Θ que maximicen la función L.

$$\Theta^* = \underset{\Theta}{argmax} \quad L(\Theta|\chi) \tag{2.35}$$

Normalmente, y para simplificar los cálculos y evitar problemas de *overflow*, lo que se hace es maximizar su equivalente logarítmico $logL(\Theta|\chi)$. El desarrollo del algoritmo EM se divide en dos pasos claramente diferenciados: primero, se realiza una estimación (expectation, E) de los parámetros de la función de probabilidad, suponiendo que el conjunto de datos estuviera completo; en segundo lugar, se maximiza la función con los valores de los parámetros supuestos en el paso Anterior (maximization, M) (McLachlan y Krishnan, 1997). Esta secuencia se repite, haciendo que el valor de la probabilidad logarítmica aumente en cada iteración, hasta que se encuentre el máximo local de la función de probabilidad. La aplicación del algoritmo EM sobre los modelos de Markov se da a conocer también como el algoritmo de *Baum-Welch* (Dempster y otros, 1977).

Búsqueda del gradiente

En este caso y a diferencia del EM, el algoritmo de búsqueda del gradiente trabaja de manera on-line, directamente sobre las muestras obtenidas, convergiendo mucho más rápidamente hacia el máximo. Esto es debido a que, mientras la convergencia del EM no tomaba el camino de mayor gradiente sobre la superficie de la función de probabilidad, necesitando de un número elevado de iteraciones hasta llegar al máximo, para el algoritmo de búsqueda del gradiente se define un ratio de aprendizaje que le permite converger hacia el máximo de manera más rápida. Además, computacionalmente hablando, las ecuaciones que definen el gradiente negativo sobre la función de probabilidad logarítmica $logP(\mathbf{O}|\lambda)$ se pueden derivar directamente, en vez de tener que aplicar sobre ellas el operador de Lagrange (tal y como se hace con el algoritmo EM). De esta manera, se obtiene una sencilla regla para el gradiente de la función que permite un aprendizaje on-line sobre los datos de entrada.

K-medias segmentado (Viterbi)

Con este método se tratan de ajustar los parámetros del modelo $\lambda = \{A, B, \pi\}$ para maximizar $P(\mathbf{O}, I|\lambda)$ donde I es la secuencia óptima de estados calculada mediante el algoritmo de Vierbi (Viterbi, 1967).

Asumiendo que se tiene un conjunto de entrenamiento formado por w secuencias de T observaciones, el algoritmo de entrenamiento k-medias segmentado se puede resumir en los siguientes pasos:

1. Se seleccionan y se extraen aleatoriamente N símbolos de las observaciones, que definen N *clusters* o grupos a los que se asocia mediante mínima distancia euclídea cada uno de los $w \times T$ vectores de observaciones del conjunto. A los *clusters* obtenidos se les denomina estados del modelo.

2. A partir de los estados obtenidos y de las secuencias observadas, se actualizan los coeficientes a_{ij} (matriz de transición entre estados).

3. A continuación se calculan los parámetros que definirán las probabilidades de emisión de cada estado:

   $\hat{c_{jm}}$ = porcentaje de vectores del estado j que han sido clasificados en el *cluster* m.

   $\hat{\mu_{jm}}$ = media de los vectores del estado j clasificados en el *cluster* m.

   $\hat{U_{jm}}$ = matriz de covarianzas de los vectores del estado j clasificados en el *cluster* m.

4. En seguida se utiliza el algoritmo de Viterbi para calcular la secuencia de estado ocultos que mejor explica cada una de las secuencias de entrenamiento mediante: $\hat{\lambda_i} = \{\hat{A}_i, \hat{B}_i, \hat{\pi}_i\}$.

5. Finalmente, si alguno de los vectores es reasignado a un nuevo estado en el Paso 4, se utiliza la nueva distribución de los *clusters* para repetir los pasos 2 a 5. Si esto no sucede, el algoritmo termina.

Capítulo 3

Sistema multimodal

3.1. Introducción

La comunicación entre humanos y máquinas siempre ha sido una tarea difícil y tediosa. La falta de recursos y capacidad en las máquinas para dicha comunicación ha obligado a sus diseñadores a hacer que el usuario sea quien se adapte al funcionamiento de los equipos en vez de ser al revés. Es por ello que en las últimas décadas ha surgido una nueva generación de sistemas multimodales que intentan resolver muchos de esos problemas de una forma flexible, adaptable, robusta y tolerante a fallos. Todo, desde una perspectiva de sistemas unimodales que se complementan entre si.

Se puede definir un sistema multimodal o interfaz multimodal como aquella que intenta resolver el problema de la adaptación de la máquina al usuario en vez de que el usuario sea quien se adapte a la máquina. En el proceso se realiza una combinación de distintas técnicas de entrada y salida de información, junto con avances en cuanto a interfaces tangibles cuyo objetivo es convertir los elementos de nuestro entorno, en elementos de interacción digital (Oviatt y otros, 2004). En el presente trabajo de tesis, se utilizan dos formas de entrada de información: Por una parte, imágenes que proporcionan información visual de señas, y por la otra, señales acústicas para determinar comandos vocales de control y movimiento.

En el presente capítulo se detalla la construcción de los módulos de reconocimiento de señas, reconocimiento de comandos vocales y la unificación de ambos reconocedores para forma el sistema multimodal. El módulo de señas, toma en cuenta el uso de imágenes bidimensionales (tomadas por una *webcam*) e imágenes tridimensionales tomadas por el dispositivo Kinect. En el módulo de voz, se muestra el desarrollo del respectivo reconocedor, así como el proceso de adaptación de nuevos usuarios. Finalmente, el módulo de unificación detalla los pasos a seguir para unir los resultados parciales y construir el sistema multimodal.

3.2. Módulo de señas

Las señas, representan un medio de comunicación no oral y no escrita que permite la comunicación de las personas sobre todo en situaciones donde la voz no puede ser utilizada, por ejemplo: comunidades de personas con discapacidades auditivas o del habla, ambientes donde el ruido es tan fuerte que la voz no es escuchada, debajo del agua en el caso de los buzos, o simplemente porque una seña puede transmitir una serie de ideas de manera natural y sencilla. Debido a esto, se desarrollo un módulo de reconocimiento de señas para proveer una interacción natural entre un humano y un sistema robótico. En las siguientes secciones se muestra a detalle la manera de construcción para el módulo mencionado.

3.2.1. Alfabeto usado

Como se menciona en secciones anteriores, el alfabeto utilizado durante el proyecto de tesis, es el correspondiente al Lenguaje de Señas Mexicano. Sin embargo, no se utilizan todos los símbolos del alfabeto, pues se descartan aquellos que involucran algún desplazamiento o movimiento de la mano. Por ejemplo, el símbolo correspondiente a la letra "j", el cual es similar al símbolo de la letra "i" pero con un movimiento que traza la trayectoria de una jota sin perder la seña original (véase Figura 3.1).

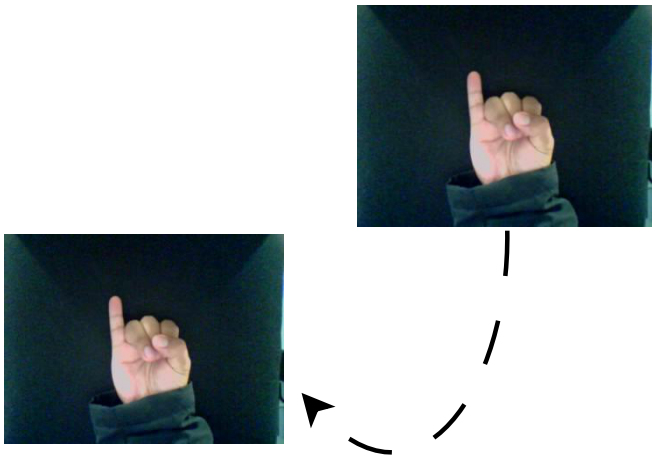


Figura 3.1: Seña J del alfabeto de LSM.

Por tanto, de los 27 símbolos que contiene el alfabeto del LSM se utilizaron solo 23, descartando aquellos con movimiento y los que fuesen similares entre ellos. En la Figura 3.2 se muestran los 23 símbolos utilizados para el estudio.

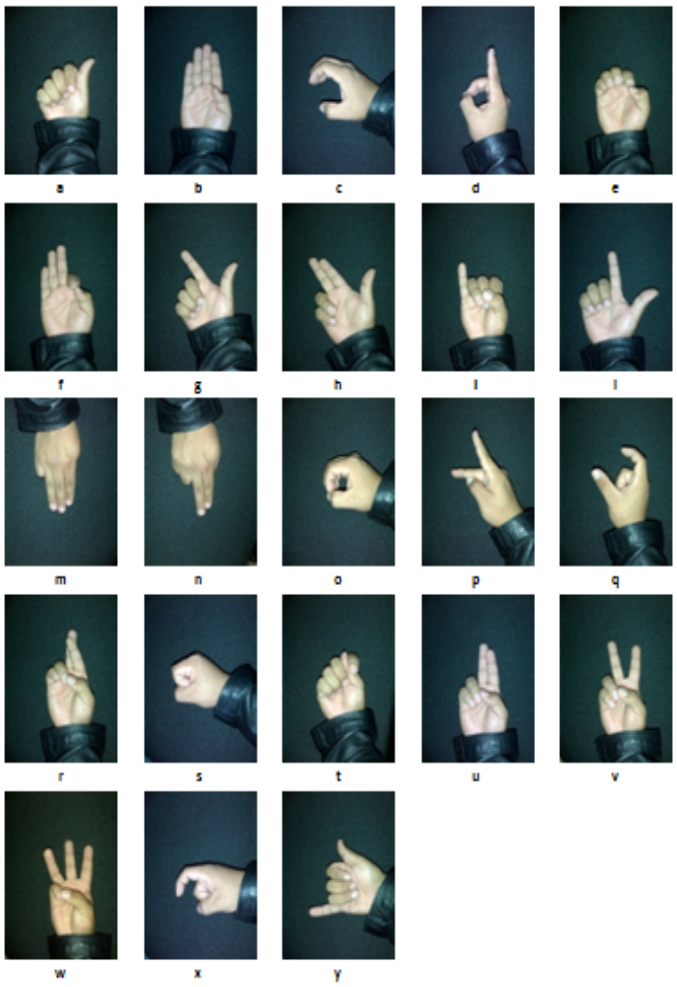


Figura 3.2: Símbolos utilizados del alfabeto del LSM.

3.2.2. Casos de estudio

Con los 23 símbolos del alfabeto del Lenguaje de Señas Mexicano, se formaron dos casos de estudio: El primero de ellos, realizando las capturas de imágenes con la *webcam* de una laptop, usando sólo imágenes bidimensionales; el segundo, utilizando el dispositivo Kinect con las cámaras RGB y de profundidad con que cuenta el mismo sensor, utilizando de esa manera, imágenes tridimensionales. A continuación, se detallan las características de cada caso de estudio.

Caso: imágenes bidimensionales

Para este caso de estudio, las capturas se realizaron utilizando la *webcam* de una laptop. Sin embargo, era necesario controlar el entorno de trabajo, es decir, las imágenes debían tomarse usando un fondo uniforme y teniendo en cuenta que la persona que hacia los signos usaba una prenda que sólo dejaba descubierta la mano. Esto con la intención de evitar la interferencia de objetos en la muñeca del usuario, por ejemplo: reloj, pulsera, etc. Además, de esta forma se aseguraba que la región de interés obtenida en la fase de segmentación pertenencia solamente a la mano. Un ejemplo de las imágenes utilizadas en este caso se muestra en la Figura 3.3, se observa que la mayor parte de la imagen es de color negro excepto la mano. Esta característica es útil al momento de realizar la segmentación y obtener solo la región de interés: la mano.

Figura 3.3: Imagen capturada con una *webcam*.

De estas imágenes, se tomaron cuatro conjuntos similares variando solamente el fondo de la imagen, es decir, cuatro imágenes por seña (véase Figura 3.4). Adicional a los cuatro conjuntos tomados con la *webcam*, por cada conjunto, se obtuvieron dos conjuntos más utilizando un filtro de variación fotométrica, haciendo un total de 12 conjuntos (12 imágenes por seña). El filtro utilizado fue el correspondiente a la función de potencia que se muestra en la ecuación 3.1.

$$f(x) = ce^x \quad 0{,}9 \leq x \leq 1{,}1 \tag{3.1}$$

Donde c es una constante y x es el parámetro que se desea variar para obtener diferentes intensidades en las imágenes tomadas por la *webcam*. El parámetro x se varió de manera aleatoria para obtener una imagen más obscura ($0{,}9 \leq x \leq 1$) y una imagen mas brillante ($1 \leq x \leq 1{,}1$) que la del conjunto original. Cabe mencionar que todos los conjuntos utilizados para este caso de estudio fueron adquiridas de un mismo usuario.

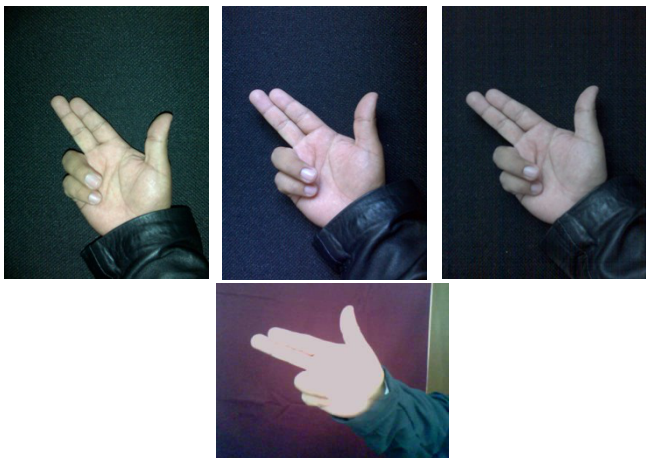


Figura 3.4: Conjuntos tomados por la *webcam*.

Los 12 conjuntos obtenidos, sirvieron para crear la base de datos de imágenes utilizadas para el entrenamiento del sistema, cada uno de los 12 conjuntos utilizados tenía una variación en la iluminación de la escena. Dicha variación permitió incorporar un espectro más amplio de iluminación en las imágenes de prueba y sobre todo discriminar la iluminación al momento de realizar las pruebas con datos capturados en tiempo real.

Caso: imágenes tridimensionales

En el segundo caso de estudio, las imágenes fueron tomadas por medio de un dispositivo Kinect. Con esto, la restricción de fondo uniforme y prendas que cubran todo el brazo no era necesaria en las imágenes capturadas. Para evitar dicha restricción se definió una zona de profundidad donde debían realizarse las señas, esta zona fue definida de 20cm (entre los 50 y 70 centímetros a partir del Kinect). De esa manera, todos los objetos que se encontraban en el rango definido, eran capturados por la cámara de profundidad del Kinect y el resto de objetos eran despreciados. Por su parte, la cámara RGB capturaba toda la escena como lo hace una cámara común. En la Figura 3.5 se puede observar un ejemplo de las imágenes tomadas por el dispositivo Kinect.

En este caso, se tomaron diez conjuntos del alfabeto de señas como el mostrado en la Figura 3.6. Cada uno con un usuario diferente, incluyendo cinco hombres y cinco mujeres. El objetivo de tomar la misma seña con diferentes usuarios fue para que el sistema aprendiera con datos reales y tomara en cuenta las variaciones en tamaño de los dedos, facilidad con que realiza cada usuario la seña, y las correspondientes variaciones de rotación en cada una de las señas.

(a) Imagen RGB (b) Imagen de profundidad

Figura 3.5: Ejemplo de imagen tomada por el Kinect.

Los diez conjuntos fueron tomados en el mismo espacio de trabajo y con el mismo fondo de escena, variando solamente el usuario que realizaba las señas de entrenamiento. Para cada usuario, se tomó la muestra de las imágenes proporcionadas por los sensores de profundidad y cámara RGB, haciendo un total de 23 imágenes RGB y 23 imágenes de profundidad por usuario (una imagen por seña) y 10 imágenes RGB y de profundidad por seña (una por usuario).

3.2.3. Esquema general de desarrollo

Con la finalidad de facilitar el trabajo de reconocimiento de señas, se dividió el problema original en pequeñas tareas que pudieran evaluarse de manera independiente. Se determinaron una serie de tareas a realizar en cada caso de estudio, las cuales incluyen captura de imágenes, segmentación, extracción de características y el aprendizaje de la red neuronal. Cada una de las tareas realizadas es mostradas en la Figura 3.7. En color gris claro se muestran las técnicas utilizadas en el sistema de señas con imágenes bidimensionales y en gris obscuro, las técnicas usadas con las imágenes de profundidad y RGB.

Se observa que como primer paso se realizo la captura de las imágenes, ya sea con una *webcam* o con un dispositivo Kinect. Posteriormente, esas imágenes fueron segmentadas para obtener la región de interés, en nuestro caso, la mano del usuario. Una vez segmentada la mano, fue necesario encontrar el vector de características de la seña, dicho vector necesitaba reflejar rasgos específicos para cada una de las señas del alfabeto. Esas características o rasgos de cada una de las señas fueron utilizadas como datos de entrada para la red neuronal multicapa. En caso de que el sistema se encontrara en fase de entrenamiento se capturaba la siguiente seña para completar el aprendizaje de la red. De lo contrario, se hacia el reconoci-

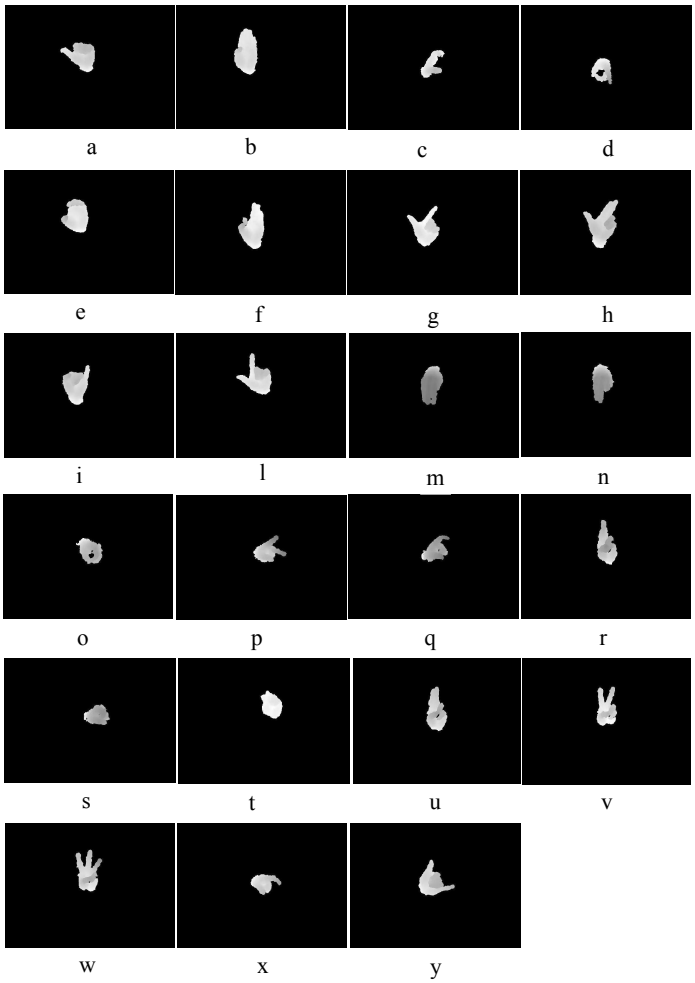


Figura 3.6: Alfabeto del LSM tomado por el Kinect.

miento de la seña proporcionada por el usuario y se interpretaba la seña como una posible acción para el sistema robótico. A continuación se detallan las diferentes fases de desarrollo para el correcto funcionamiento del reconocedor de señas.

3.2.4. Segmentación

El primer paso para la realización del módulo de señas fue la segmentación de imágenes. En el caso de las capturas tomadas por la *webcam* se utilizaron contornos activos (Blake y Isard, 1998). Este tipo de segmentadores, separa los objetos dentro de una imagen del fondo de la misma y permite la extracción del contorno de objetos basado en modelos que usan información a priori sobre la forma de los objetos. Estas métodos son los más robustos ante

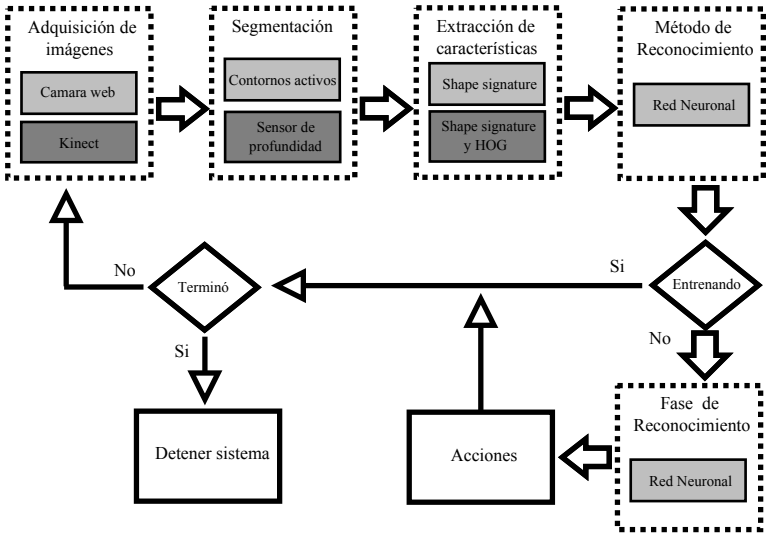


Figura 3.7: Esquema general de desarrollo para el sistema de reconocimiento de señas.

la presencia de ruido y por tanto permite una segmentación ideal en imágenes complejas. Los contornos activos se pueden clasificar en *snakes* (serpientes), patrones deformables y contornos dinámicos. En este trabajo se utilizaron *snakes* para llevar a cabo la segmentación (Lankton y Tannenbaum, 2008).

Un *snake* se puede definir como una curva *spline* minimizadora de energía, guiada por fuerzas restrictivas externas e influenciada por fuerzas de la imagen, que tiende a localizarse en características de ésta como líneas y bordes. Es, por tanto, un contorno activo que evoluciona de forma dinámica hacia los contornos relevantes de la imagen. El *snake* también posee una serie de fuerzas internas que sirven para imponer restricciones de suavidad, es decir, para regularizar la solución. Las fuerzas de la imagen empujan al *snake* hacia características de la imagen como líneas, bordes y contornos subjetivos, mientras que las fuerzas restrictivas externas añaden información de alto nivel para hacer que el *snake* se vaya hacia el mínimo local deseado.

En la Figura 3.8 se muestra el progreso de la segmentación utilizando el *snake* con la seña correspondiente a la letra "y" del LSM. La primera imagen muestra el estado inicial del segmentador y la última imagen muestra el contorno final de la seña.

Por su parte, las imágenes tridimensionales fueron segmentadas por el sensor de profundidad del Kinect. Como se menciona en párrafos anteriores se definió un rango de captura para el sensor y sólo se tomaban los objetos localizados en dicho rango. Así, el resto de los objetos en la escena eran despreciados por el mismo sensor, quedándose solamente con la información de la mano tal como se muestra en la Figura 3.5.

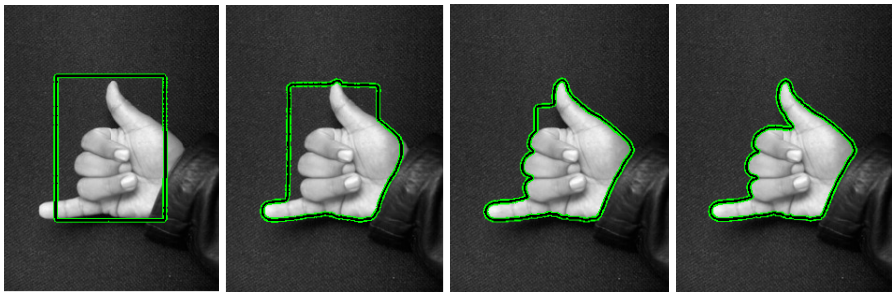


Figura 3.8: Diferentes transiciones del snake hasta el contorno final.

3.2.5. Extracción de características

Una vez obtenida la segmentación de las imágenes se procedió a obtener los descriptores de forma. Existe gran cantidad de descriptores que utilizan los contornos como información para generar el descriptor correspondiente. Dos descriptores de este tipo son *shape context* (Belongie y otros, 2002) el cual explota los contornos obtenidos a partir de un operador de gradiente y *shape signature* (Fujimura y Sako, 1999) que utiliza solo el contorno exterior de los objetos a clasificar.

Dado que tanto el *snake* como el sensor de profundidad del Kinect proporcionan como resultado el borde del objeto se utilizaron los métodos de *shape signature* e Histograma de Gradientes Orientados (HOG) como descriptores de forma.

En este caso el contorno de los objetos fue la mano del usuario con la forma del signo correspondiente a un símbolo del alfabeto LSM (véase Figura 3.9). Cabe mencionar que en el caso de las imágenes bidimensionales sólo se utilizó el descriptor de *Shape signature*, mientras que en el caso de imágenes tridimensionales se utilizaron ambos métodos para la extracción de características.

Shape signature

De manera general, la obtención de la firma del objeto se realiza mediante el cálculo de las distancias desde el centro de gravedad del objeto hacia cada uno de los puntos que forman el contorno. Esto nos genera un histograma de distancias como el de la Figura 3.10 el cual es el histograma para el contorno del símbolo mostrado en la Figura 3.9a. Dicho histograma consiste de 360 valores y fue utilizado como vector de entrenamiento de la Red Neuronal en el caso de imágenes tomadas por una *webcam*.

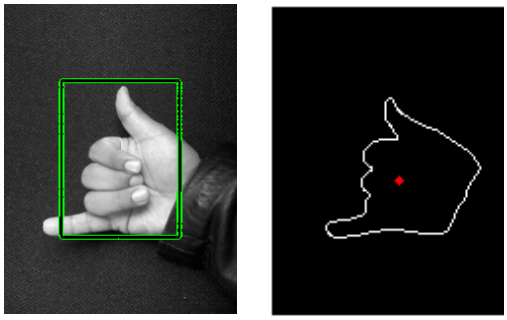


Figura 3.9: (a) Imagen original, (b) bordes de la seña.

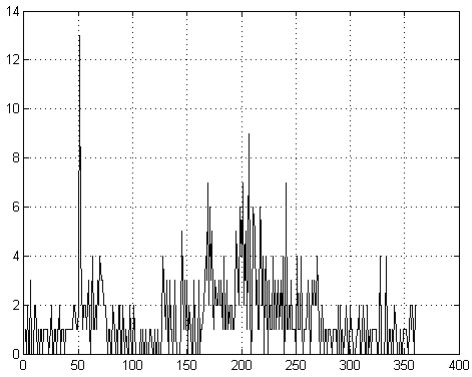


Figura 3.10: Histograma de distancias para el símbolo que representa la letra “y”.

Histograma de Gradientes Orientados

El algoritmo correspondiente al Histograma de Gradientes Orientados ha sido utilizado como uno de los principales descriptores de formas en el proceso del reconocimiento de objetos. Además, su uso se ha extendido al reconocimiento y seguimiento de personas o figuras humanas (Dalal y otros, 2006). Para el desarrollo del presente trabajo se utilizó una ventana de 3x3, con las orientaciones mostradas en el Cuadro 3.1.

El uno al centro de la ventana es el pixel que representa el borde de la seña. Por cada pixel de tipo borde, se obtiene la orientación del gradiente en esa sección de la imagen, definido como el vector normal al borde en dicho punto. Con todas las orientaciones se forma un histograma de ocho valores, el número total de gradientes en cada dirección de la ventana. Este último vector fue concatenado con el histograma de la firma y en conjunto formaron el vector de características para el caso de estudio con las imágenes del Kinect.

Cuadro 3.1: Orientaciones usadas en el algoritmo HOG.

135°	90°	45°
180°	1	0°
225°	270°	315°

3.2.6. Entrenamiento de la Red Neuronal

Con la obtención de los histogramas que forma la descripción de los diferentes símbolos del alfabeto LSM se obtuvieron los datos de entrada para poder entrenar la Red Neuronal. En ambos casos de estudio, se crea una Red Neuronal de tres capas. En el primer caso, donde se utilizan imágenes de una *webcam*, el vector de entrada se construye con los 360 valores proporcionados por la implementación de *Shape signature*. Por tanto, la red neuronal se construye con 360 neuronas en la capa de entrada, 23 neuronas en la capa oculta y sólo una neurona en la capa de salida como se muestra en la Figura 3.11. La única neurona en la capa de salida era capaz de tomar valores entre 0 y 23, dependiendo de la seña reconocida.

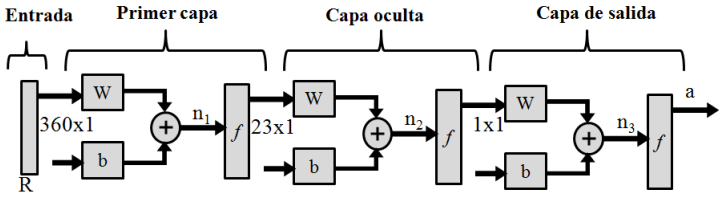


Figura 3.11: Estructura de la Red Neuronal empleada con imágenes bidimensionales.

En el segundo caso de estudio, usando imágenes de profundidad del Kinect, el vector de entrada se construye con la concatenación de los 360 valores de la firma del objeto y las ocho orientaciones de los gradientes. Con ello se forma un vector de entrada de 368 valores, creando una Red Neuronal de 368 neuronas en la capa de entrada, 184 neuronas en la capa oculta y 23 neuronas en la capa de salida. Donde cada neurona de la salida representa una seña del alfabeto del LSM, habilitándose solamente la neurona que correspondía a la seña reconocida.

Para el entrenamiento de las Redes Neuronales se usó el algoritmo de *backpropagation* y sus variantes: Momento, Razón de aprendizaje variable, Gradiente conjugado y Levenberg-Marquardt. La ventaja que presentan dichos algoritmos es la rapidez de convergencia y robustez respecto a otro tipo de entrenamiento (Haykin, 2009).

3.3. Módulo de voz

Como se menciona en capítulos anteriores, el habla es la forma más natural de comunicación entre los humanos. Por medio de ella se pueden transmitir ideas con la menor pérdida de información. Por ello, el presente trabajo de tesis plantea el uso de la voz para controlar los movimientos y manipulaciones que realiza un sistema robótico.

Los sistemas de reconocimiento del habla consideran que la señal de voz es la realización de algún mensaje codificado como una secuencia de uno o más símbolos. Éstos, puede ser clasificado de dos formas: Dependiente de usuario, es decir, se construye tomando en cuenta las características particulares de un solo usuario y sólo él puede usar el sistema; o independiente de usuario, el cual se construye a partir de características de voz de muchos usuarios, para posteriormente personalizar el sistema mediante técnicas de adaptación de usuario.

En este módulo, se desarrolla un sistema de reconocimiento del habla independiente de usuario, el cuál utiliza como técnica de reconocimiento a los Modelos Ocultos de Markov y se apoya de la herramienta de desarrollo HTK, implementada por la Universidad de Cambridge (Young y Woodland, 2006).

Es las siguientes secciones, se muestra de forma detallada el desarrollo del módulo de reconocimiento del habla. Se explican los procesos para la creación del corpus de entrenamiento, el modelado acústico, así como la implementación y evaluación del sistema de reconocimiento automático del habla. Se muestra también el proceso para realizar la adaptación de voz de nuevos usuarios del sistema, esto con la finalidad de que el sistema pueda ser utilizado por cualquier usuario.

3.3.1. Esquema general de desarrollo

Al igual que el módulo de señas, el reconocimiento del habla se puede dividir en componentes más pequeños que facilitan el proceso asociado al tratamiento de señales acústicas. En la Figura 3.12 se presentan dichos componentes, los cuales incluyen la adaptación de nuevo usuario. Este módulo de reconocimiento se realiza mediante un modelado acústico de palabras a nivel fonema, ya que así se modela mejor la coarticulación de las palabras.

De acuerdo con el esquema de desarrollo, como primer paso es necesario construir un corpus de entrenamiento, formado a su vez, por un corpus textual y un corpus oral. Dicho corpus de entrenamiento, es utilizado para la construcción y entrenamiento supervisado de los modelos acústicos, los cuales son representados mediante los Modelos Ocultos de Markov. Por su parte, tanto el modelo del lenguaje como el diccionario fonético son construidos a partir del corpus textual. Una vez construidos los modelos acústicos, el modelo del lenguaje y el diccionario fonético, los tres elementos son utilizados por el algoritmo de búsqueda para estimar una palabra ($\hat{W}$) dada una muestra de voz ($\mathbf{O}$). El algoritmo de búsqueda compara una señal acústica de voz ($\mathbf{O}$) con los patrones de los modelos acústicos (HMM's) y como

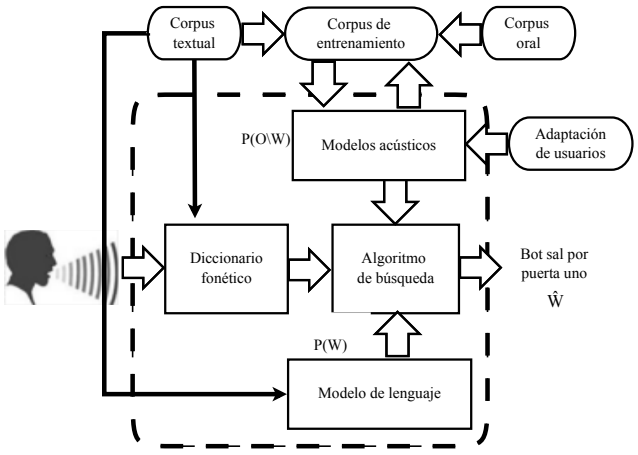


Figura 3.12: Esquema general de desarrollo para el sistema de reconocimiento del habla.

resultado genera una secuencia de modelos acústicos que representan los fonemas que mejor describen la señal de voz con máxima probabilidad.

De manera general, los modelos acústicos son aquellos que proveen de la probabilidad de observar una señal acústica de voz ($\mathbf{O}$), dada una palabra o frase (W). El modelo del lenguaje es el conjunto de reglas gramaticales, vocabulario y probabilidades a priori de las frases incluidas en el sistema. El diccionario fonético, proporciona todo el conjunto de palabras que se desea reconocer, además de su correspondiente descomposición fonética, y el algoritmo de búsqueda permite estimar la palabra o frase reconocida por el sistema.

Adicionalmente, se cuenta con el componente de adaptación de usuario que sirve para la construcción de los nuevo modelos acústicos, los cuales se crean con las características especiales de cada usuario del sistema. Este componente incluye también el uso de un corpus textual de adaptación el cual es leído por el nuevo usuario al comenzar a usar el sistema.

3.3.2. Corpus de entrenamiento

Un corpus de entrenamiento se define como una base de datos o colección de archivos de voz (audio) y transcripciones textuales de los mismos en un formato que puede ser usado para la creación y refinación de modelos acústicos para los sistemas de reconocimiento del habla. Dentro de estos corpus, se pueden diferenciar dos tipos:

- Textual. Consiste en una colección de textos representativos de un lenguaje.
- Oral. Representado por una colección de archivos de audio (voz), generalmente obtenidos por la lectura del texto representativo.

Cuadro 3.2: Frases de entrenamiento del sistema (corpus textual).

No.	Frase	No.	Frase
1	Abrir la puerta principal.	26	Buscar llaves del segundo piso.
2	Encender todas las luces.	27	Checar llantas del auto.
3	Encender luces de la sala.	28	Jalar llaves del auto.
4	Servir copa extra de vino.	29	Guardar la faja de mama.
5	Servir vaso con refresco.	30	Guardar el ferrocarril de los niños.
6	Cocinar pollo para la comida.	31	Llenar mochila de chocolates.
7	Sacar gato por la mañana.	32	Bañar a los niños chicos.
8	Jalar puerta del baño.	33	Llamar a la niñera extraña.
9	Preparar comida para el perro.	34	Cambiar texto de bienvenida.
10	Evitar entrada a extraños.	35	Prender chimenea de noche.
11	Encender alarma de casa.	36	Quitar chicles del tapete favorito.
12	Introducir pollo frio al refrigerador.	37	Felicitar a la niñera del día.
13	Jalar correa del gato.	38	Poner yeso extra a la figura.
14	Sacar pájaro de jaula.	39	Formar examen especial para niños.
15	Extraer características de visitantes.	40	Evitar fumar en la cocina.
16	Leer mensajes de contestadora.	41	Exponer fachada a la luz.
17	Llamar perro desde el barco.	42	Extender falda de la niña.
18	Cuidar niños de extraños.	43	Extraer llaves de oficina.
19	Correr los pájaros extraños.	44	Llevar mucho chocolate al niño.
20	Guardar juego del niño.	45	Extraer cigarros de sala.
21	Lavar peluches de perro y gato.	46	Extender jícara de gato y perro.
22	Encender lavadora por la mañana.	47	Forjar juego de chapas.
23	Encender estéreo con música fuerte.	48	Llamar ferrocarril de chatarra.
24	Apagar televisor después de luces.	49	Extender chaleco para el perro.
25	Subir casa del gato.	50	Mojar mochila en la lluvia.

Corpus textual

Debido a las limitaciones en existencia de corpus de entrenamiento para el español Mexicano, se construyó un corpus textual de entrenamiento con 50 frases. Dichas frases se diseñaron de tal manera que incluyeran al menos cinco muestras de voz por cada fonema del español Mexicano, tal como se muestra en Green y otros (2003), donde los autores muestran que con cinco muestras de voz se pueden obtener altos porcentajes de reconocimiento, aún con personas que presentan discapacidades de pronunciación en el habla. Cabe mencionar que los fonemas utilizados por el sistema se definieron en base al alfabeto *Mexbet* presentado por Cuétara (2004). El cuadro 3.2 muestra las 50 frases con las que se formó el corpus textual de entrenamiento.

Corpus oral

Una vez generadas las frases de entrenamiento, se procede a realizar la lectura de dichas frases, con lo cual se forma el corpus oral de entrenamiento. Dicho corpus se genera con las muestras de voz de 10 usuarios diferentes, cinco mujeres y cinco hombres, con esto se asegura que las muestras de voz incluyen diferentes variaciones en tono, intensidad de voz y velocidad de pronunciación. De esta manera, la adaptación de voz no presentará ninguna dificultad.

La grabación de las frases se realiza con la ayuda de un micrófono y usando un script de Matlab. Cada uno de los archivos de audio es almacenado y codificado para obtener un muestreo de la voz del usuario. La codificación utilizada es la MFCC (*Mel Frequency Cepstral Coefficients*).

La voz para su reconocimiento debe ser codificada en un formato que represente sus características espectrales más importantes. Para esto, se han propuesto diferentes métodos, entre ellos se puede mencionar a los Coeficientes de Predicción Lineal (*Linear Predictive Coefficients*, LPCs) y los Coeficientes Cepstrales de Frecuencia Mel (MFCCs). En particular, los MFCCs han demostrado desempeño superior en cuanto a reconocimiento e identificación de emociones por voz (Jurafsky y Martin, 2009; Young y Woodland, 2006). Los MFCCs se basan en la variación que se conoce de la percepción que tiene el oído humano para diferentes frecuencias: el oído humano percibe el sonido en bandas de frecuencia de amplitud variante (Davis y Mermelstein, 1980).

La herramienta HTK contiene un módulo denominado HCopy especialmente diseñado para codificación de voz en varios formatos como LPC y MFCCs. La ejecución de HCopy se presenta a continuación:

HCopy –C config0.scp –S codifica.scp

En donde config0.scp es un archivo de texto que contiene las especificaciones de la codificación MFCC y codifica.scp un archivo de texto que contiene una lista de los archivos de sonido a codificar. En la Figura 3.13 se muestra el archivo de configuración config0.txt.

En el archivo se indica con la linea SOURCEFORMAT=WAV el formato de origen de los archivos de voz a codificar (en este caso, WAV). TARGETKIND=MFCC_0_D_A indica el formato destino que consiste en MFCCs con coeficientes de energía (0), delta (D) y aceleración (A). Con la línea TARGETRATE=100000.0 se especifica que la señal se muestree cada 10 milisegundos. Con las líneas WINDOWSIZE=250000.0 y USEHAMMING=T se especifica que se use una ventana Hamming de 25 milisegundos para la codificación. NUMCEPS=12 indica que se usarán 12 MFCCs con un coeficiente adicional para representar la energía de la señal. Con esto se tendrán en total 13 coeficientes los cuales, al añadírseles coeficientes delta (D) y de aceleración (A) darán una codificación de 39 coeficientes.

```
SOURCEFORMAT=WAV
TARGETKIND=MFCC_0_D_A
TARGETRATE=100000.0
SAVECOMPRESSED=T
SAVEWITHCRC=T
WINDOWSIZE=250000.0
USEHAMMING=T
PREEMCOEF=0.97
NUMCHANS=26
CEPLIFTER=22
NUMCEPS=12
ENORMALISE=F
```

Figura 3.13: Archivo de configuración para codificación MFCC.

3.3.3. Modelos acústicos y diccionario fonético

El modelado acústico consiste en el proceso de establecer representaciones estadísticas para las características espectrales de la señal de voz. En nuestro caso, los Modelos Ocultos de Markov fueron utilizados para modelar las características acústicas de las voces de los usuarios. Dichos modelos fueron generados con tres estados y con una arquitectura de tipo izquierda-derecha. Además de utilizar ocho componentes gaussianos por estado.

Como se menciona anteriormente, se realiza el modelado a nivel fonético en lugar de a nivel palabra, pues el fonema es la unidad básica con la que se puede formar una o más palabras, por ejemplo, la palabra CASA se forma por la secuencia de fonemas /k/ /a/ /s/ /a/. En este sentido, el diccionario fonético es usado para establecer la secuencia de fonemas que define cada una de las palabras en el vocabulario utilizado por el reconocedor. En este caso, se definieron las secuencias de fonemas para cada palabra de control asociada a los 23 símbolos del alfabeto del LSM (véase Cuadro 3.4).

3.3.4. Modelo del lenguaje

También llamado gramática debido a que representa un conjunto de reglas o probabilidades que determinan las secuencias de palabras permisibles en un lenguaje. Esta característica incrementa el desempeño de los reconocedores del habla ya que guía el proceso de reconocimiento mediante ciertas restricciones en cuanto a secuencias fonéticas que son estadísticamente más probables que otras.

Para el caso de estudio mostrado en este trabajo de tesis, se definen dos tipos de sentencias de control, la primera de ellas para manipulación de objetos y la segunda para controlar el movimiento del sistema robótico. Las sentencias o frases de manipulación deben seguir la siguiente estructura:

Dispositivo + *Tarea* + *Objeto*

Aquí, *Dispositivo* define el nombre o identificador del sistema robótico, por ejemplo robot, bot, móvil, brazo, manipulador, cube etc. *Tarea* identifica el tipo de acción que se realiza sobre un *Objeto* en específico (véase Figura 3.14).

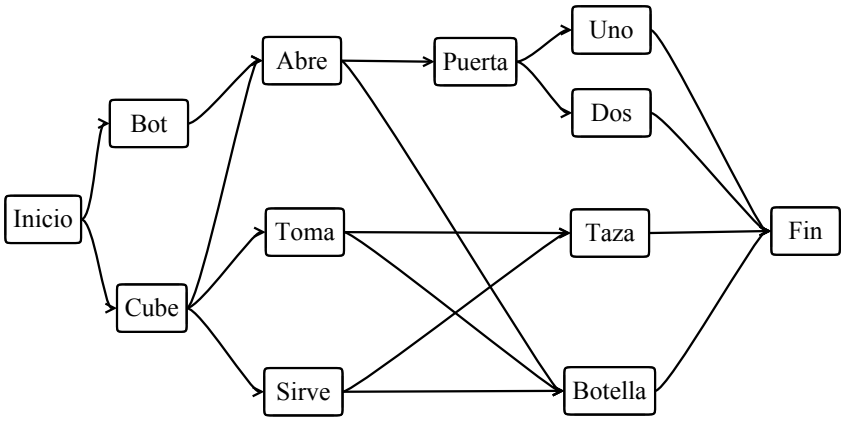


Figura 3.14: Frases para tareas de manipulación.

Por su parte, las frases de control de movimiento deben presentar la siguiente estructura:

Dispositivo + *Tarea* + *Configuración*

donde *Dispositivo* define, nuevamente, el nombre o identificador del sistema robótico. *Tarea* identifica el tipo de acción que realizara el sistema robótico y *Configuración* proporciona los detalles de la tarea que se desea realizar (véase Figura 3.15).

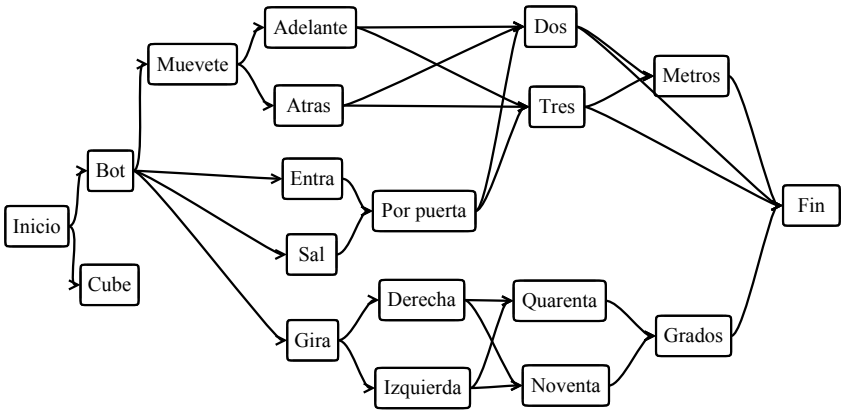


Figura 3.15: Frases para tareas de movimiento.

De este modo, la gramática permite reconocer comandos simples y de mediana complejidad. Es importante mencionar que tanto el modelado acústico como en el modelo del lenguaje fue implementado utilizando la herramienta HTK. Lo mismo sucedió con los algoritmos de búsqueda para la evaluación del presente módulo.

3.3.5. Adaptación de usuarios

Cuando las muestras de voz para entrenamiento se encuentran disponibles, los parámetros de los HMM's para reconocimiento se pueden estimar de manera eficiente, favoreciendo que el algoritmo de Viterbi produzca buenos resultados. Sin embargo, este desempeño depende de las muestras de voz utilizadas para el entrenamiento, y su desempeño puede ser deficiente con usuarios distintos cuyas voces no se usaron para entrenar el sistema.

En este caso, las técnicas de adaptación de usuario conocidas como *Maximum Likelihood Linear Regression* (MLLR) (Leggetter y Woodland, 1995) y *Maximum A-Posteriori* (MAP) (Young y Woodland, 2006) se han desarrollado para ajustar los parámetros de los HMM's de un sistema Independiente de Usuario, o Dependiente de Usuario, a las características acústicas de un usuario en particular. Estas técnicas normalmente requieren de algunas muestras de voz del usuario (datos de adaptación) para estimar "transformaciones" que ajusten los parámetros de los HMM's a su voz. La adaptación es supervisada cuando hay conocimiento de las palabras pronunciadas por el usuario, y es no supervisada cuando no se tiene dicha información.

MLLR es la técnica de adaptación de usuario que más se utiliza en el ámbito del reconocimiento de voz. En particular, MLLR es superior a otras técnicas como MAP cuando la cantidad de material de voz para adaptatión es limitada (Mak y otros, 2006). Es por ello, que en el presente trabajo se utiliza dicha técnica para implementar la adaptación de nuevos usuarios en el sistema multimodal.

MLLR se basa en el supuesto de que un conjunto de transformaciones lineales se puede usar para reducir la diferencia entre los modelos acústicos de un reconocedor de voz y los datos de adaptación. Estas transformaciones son aplicadas sobre la media y varianza de las mixturas de gaussianas de los HMM del sistema base, teniendo el efecto de ajustar dichos parámetros de tal manera que aumente la probabilidad de que los HMM's del sistema generen los datos de adaptación.

En el Cuadro 3.3 se muestran las frases utilizadas para realizar la adaptación de nuevos usuarios en el módulo de voz.

Cuadro 3.3: Frases de adaptación para nuevos usuarios.

No.	Frase	No.	Frase
1	El extraño niño está llorando mucho.	9	Así el barco avanza rápido.
2	El ratón jalo la azúcar.	10	Mi familia vivió en México.
3	El futbol llanero mueve mucha afición.	11	Mi mama cumple años extra mañana.
4	La faja talla extra esta al revés.	12	El tío comió pollo chino.
5	El pájaro ya está en la jaula.	13	Según ellos la silaba es correcta.
6	El gato gruño muy fuerte.	14	La pieza exhumada es única y característica.
7	El elefante es más grande en África.	15	Algún día volveré y venceré.
8	El chango es pequeño en América.	16	Aquí llovió mucho desde anoche.

3.4. Módulo de unificación

Como se mencionó al inicio de esta tesis, el desarrollo del sistema multimodal se basa en los dos módulos de reconocimiento, ambos descritos en las secciones anteriores. El módulo de señas, que interpreta una seña realizada con las manos del usuario, y el módulo de voz, que reconoce un comando vocal pronunciado por el mismo usuario. Cada uno de los módulos de reconocimiento proporciona una interpretación independiente de los datos proporcionados por el usuario del sistema, por tanto, dichas interpretaciones necesitan ser integradas en un solo comando de control que será enviado al sistema robótico. En esta sección, se presenta el último módulo del sistema multimodal, el cual consiste en la unificación de resultados independientes de señas y voz.

3.4.1. Esquema general de unificación

En la Figura 3.16 se muestra el diagrama a bloques del sistema multimodal realizado. Se observa que cada módulo de reconocimiento recibe un tipo de datos de entrada, el cual es procesado de manera independiente por el correspondiente módulo de reconocimiento.

El módulo de voz toma como entrada una secuencia de audio, esta es procesada por medio de los Modelos Ocultos de Markov implementados en HTK y genera una frase de control para el robot. Dicha frase tiene asociada una probabilidad de ocurrencia, es decir, se conoce estadísticamente el comando pronunciado por el interlocutor.

Por su parte, el módulo de señas toma como entrada una imagen tridimensional del Kinect que es procesada mediante Redes Neuronales Artificiales. En este caso, se obtiene como salida un vector binario de 23 valores, donde cada posición del vector representa una seña del alfabeto. De esta manera, si se enciende la primer posición del vector, significa que

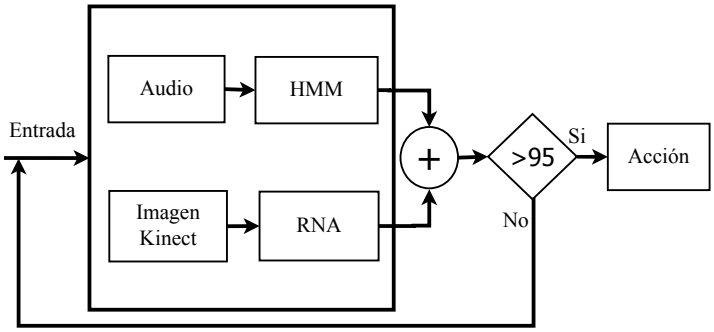


Figura 3.16: Estructura del sistema de reconocimiento multimodal.

se está reconociendo la primer seña del alfabeto (la letra "a"), de igual manera, si el único valor diferente de cero está localizado en la última posición del vector, significa que se ha reconocido el último signo del alfabeto. En este caso, la salida también es interpretada con una probabilidad de ocurrencia mediante un muestreo de datos. Así, si el valor de salida de la red neuronal se acerca a uno, se tiene una mayor probabilidad de ocurrencia para la seña reconocida, en caso contrario, la probabilidad de que la seña reconocida sea correcta, disminuye.

Para poder utilizar ambos módulos de reconocimiento, se definieron 23 comandos de control de movimiento y manipulación de objetos por parte del sistema robótico, esto con la finalidad de ocupar los 23 símbolos del alfabeto del LSM. En el Cuadro 3.4 se muestran los 23 comandos utilizados para probar el sistema, se muestra la relación entre las señas y los comandos de voz utilizados. Dichos comandos están diseñados para utilizarse por personas con discapacidades o personas de la tercera edad.

3.4.2. Modo de operación

Las salidas proporcionadas por los módulos de reconocimiento, pueden complementarse entre ellas cuando ambas reconocen el mismo comando, o bien, interferir entre ellos cuando reconocen comandos diferentes. Inicialmente, todos los comandos que puede reconocer el sistema tienen la misma probabilidad de ocurrencia (Figura 3.17 (a)).

En caso de que los módulos de reconocimiento asocien sus datos de manera diferente (Figura 3.17 (b)), donde el sistema de señas indica que el comando reconocido es la letra "e" con una probabilidad de 0,5 y el sistema de voz reconoce el comando "l" con una probabilidad de 0.8, difícilmente se alcanzará el umbral del sistema, ya que el comando para la letra "e" influye de manera negativa en el comando "l". Por tanto, el sistema debe actualizar los porcentajes de reconocimiento de cada seña (Figura 3.17 (c)) y solicitar nuevamente ambos

Cuadro 3.4: Frases de control utilizadas en el sistema multimodal.

Seña	Frase	Seña	Frase
a	Bot retrocede lento.	o	Bot sal por la puerta uno.
b	Bot detente.	p	Bot sal por la puerta dos.
c	Cube posición inicio.	q	Cube gira la perilla.
d	Bot retrocede rápido.	r	Cube desplaza la ventana.
e	Bot avanza rápido.	s	Bot entra por la puerta dos.
f	Bot avanza rápido tres metros.	t	Bot entra por la puerta uno.
g	Bot retrocede lento dos metros.	u	Cube sirve la botella.
h	Bot avanza rápido dos metros.	v	Cube toma el vaso.
i	Bot retrocede rápido.	w	Bot requiero tu atención.
l	Bot gira noventa grados a la izquierda.	x	Cube jala la puerta.
m	Bot gira noventa grados a la derecha.	y	Bot sirve el vaso.
n	Sistema termina ejecución.		

datos (seña y voz). Lo mismo sucede si alguno de los datos no es proporcionado, el ciclo se mantendrá hasta que un comando sobrepase el umbral del 95 %.

Por otra parte, cuando se soliciten los datos al usuario y éstos asocien la información a un mismo comando, las probabilidades de ocurrencia de ambos sistemas se suman y de esa manera pueden superar el umbral del sistema.

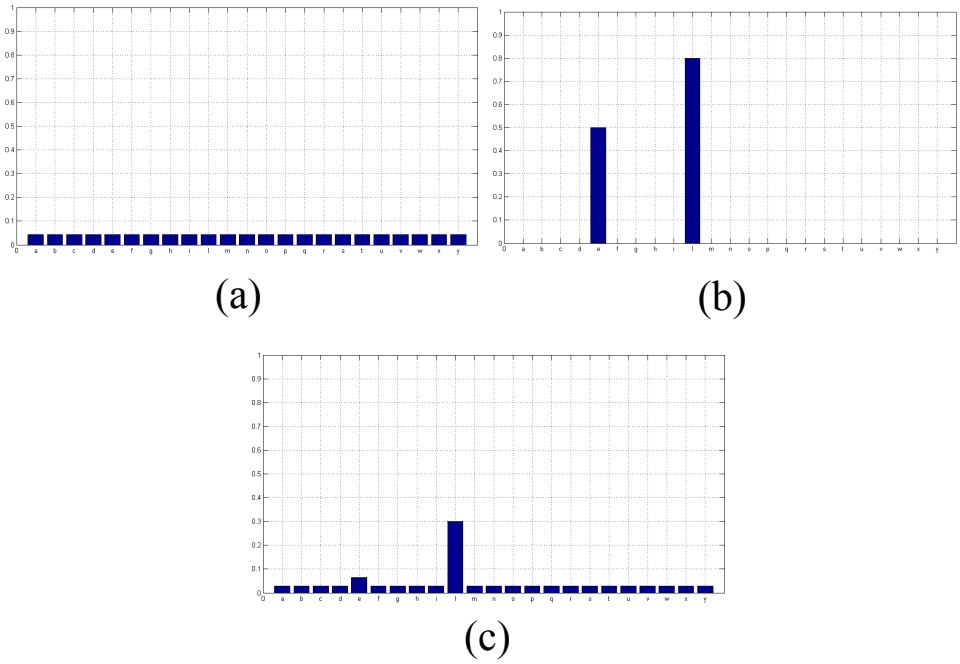


Figura 3.17: Probabilidades de ocurrencia para los comandos “e” y “l” en t_0 (a), y t_1 (c).

Capítulo 4

Resultados

Como se describe en el capítulo tres, la metodología a seguir para el sistema multimodal fue la siguiente. Se implementaron redes neuronales para reconocer un conjunto de 23 señas del alfabeto del LSM, en tal caso se presentaron dos formas de obtener y segmentar las imágenes, una utilizando una *webcam* y la otra mediante las cámaras de profundidad y RGB de un sensor Kinect. Posteriormente, se desarrollo el sistema de reconocimiento del habla por medio de Modelos Ocultos de Markov y su correspondiente adaptación de voz mediante la técnica MLLR. Finalmente, se implemento el sistema multimodal para unificar los resultados de cada uno de los reconocedores y mandar una sola señal al sistema robótico.

En el presente capítulo se muestran los resultados de cada una de las etapas mencionadas y los resultados obtenidos al desarrollar una aplicación en un entorno de simulación.

4.1. Resultados experimentales

4.1.1. Módulo de señas

Como primer paso en el reconocimiento de señas, se realizó una comparación de los algoritmos de entrenamiento *backpropagation* y sus variantes. Dicha comparación se realizó usando la base de datos COIL-20 (Nene y otros, 1996). En tal caso, se utilizó como característica de clasificación un histograma de intensidades como el mostrado en la Figura 4.1.

De esta comparación se obtuvo que el algoritmo que mejor clasifica objetos es el que corresponde a la variante de gradiente conjugado ya que el algoritmo de Levenberg-Marquardt fue muy lento y no termino de entrenarse con el vector de entrada de 256 valores, sólo realizó 59 iteraciones de entrenamiento. El cuadro 4.1 muestra los porcentajes de generalización obtenidos en por cada algoritmo.

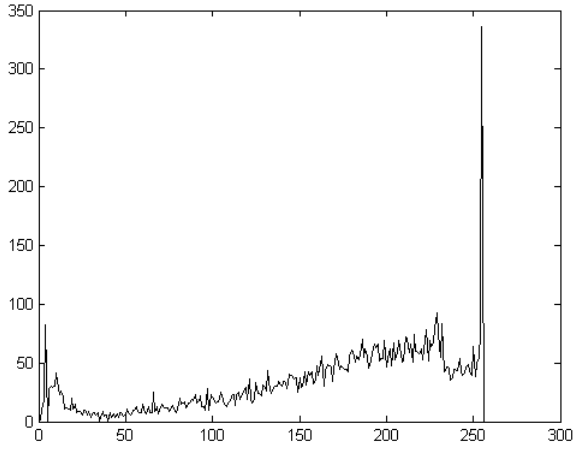


Figura 4.1: Histograma de intensidades.

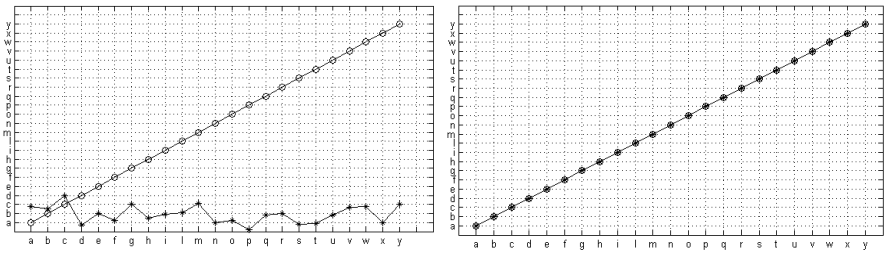
Cuadro 4.1: Porcentaje de reconocimiento de los algoritmos *backpropagation*.

Algoritmo	% de reconocimiento
Backpropagation Base	65.5 %
Momentum	65.5 %
Razón de aprendizaje variable	65.5 %
Gradiente Conjugado	69.0 %
Levenberg-Marquard	44.7 %

Una vez obtenidos los resultados de la comparación se decidió utilizar el algoritmo de entrenamiento de Gradiente Conjugado para el resto de las implementaciones, se probó la red neuronal antes y después del entrenamiento para observar la variación al realizar el reconocimiento de las 23 señas. En ambos casos, usando imágenes bidimensionales y tridimensionales, la red neuronal se entreno de manera correcta clasificando correctamente todas las imágenes usadas en el entrenamiento, tal como se muestra en la Figura 4.2.

Para evaluar los sistemas de reconocimiento, se realizaron pruebas con nuevas imágenes las cuales eran capturadas en tiempo real. Por una parte, la *webcam* y por la otra el sensor Kinect. Debemos recordar que en el primer caso solo se uso la firma de los objetos y en el segundo se agrego el Histograma de Gradientes Orientados, con ello se obtuvieron los porcentajes de reconocimiento mostrados en el Cuadro 4.2.

Con estos porcentajes de reconocimiento se decidió utilizar la implementación del Kinect para ser usado por el sistema multimodal. En ambos casos, las señas que más problemas presentaron al momento de realizar las pruebas fueron las correspondientes a los símbolos: b, d, f, g, m y s.



(a) (b)

Figura 4.2: Salida de la red neuronal (a) antes del entrenamiento y (b) después del entrenamiento.

Cuadro 4.2: Porcentaje de generalización en los sistemas de señas.

caso	% de reconocimiento
webcam	90 %
Kinect	95 %

4.1.2. Módulo de voz

Inicialmente, el sistema fue probado con el mismo corpus de entrenamiento para verificar la exactitud en el modelado de fonemas y la estabilidad del sistema en línea base. La métrica para medir el rendimiento del sistema fue el porcentaje de exactitud en las palabras reconocidas, el cual es obtenido mediante la siguiente expresión.

$$\%WAcc = (N - D - S - I)/N \tag{4.1}$$

Donde N es el número de palabras de referencia (frases correctas), S, D e I, son el número de palabras Sustituidas, Eliminadas e Insertadas en la frase reconocida. Por lo tanto, esta medida considera los diferentes tipos de errores que pueden cambiar el significado de una frase reconocida.

Los resultados obtenidos con las mismas frases de entrenamiento son mostradas en el Cuadro 4.3, los cuales muestran un porcentaje de reconocimiento del 98.9 %.

Posteriormente, se probo el sistema con nuevos usuarios, es decir, personas diferentes a los 10 que apoyaron en el corpus de entrenamiento, para ello los usuarios debieron leer las 16 frases de adaptación mostradas en el Cuadro 3.3 y así poder utilizar el sistema adaptado. Así pues, el módulo de señas fue probado con 6 usuarios más, tres mujeres y tres hombres.

Cuadro 4.3: Desempeño del reconocedor con las frases de entrenamiento.

N	D	S	I	% de reconocimiento
1418	5	8	2	98.9

Cuadro 4.4: Porcentaje de generalización en el sistema de voz.

Usuario	fallas/total	% de reconocimiento
Mujeres	8/138	94.2 %
Hombres	5/138	96.3 %

En cada caso, el usuario debía leer las 23 frases utilizadas por el sistema obteniendo un total de 138 muestras de voz para prueba. En este caso, ya no se utilizo el porcentaje de palabras reconocidas como medida de desempeño, sino el número de frases reconocidas de manera correcta. Con ello, se obtuvieron los resultados mostrados en el Cuadro 4.4, mostrando un porcentaje de reconocimiento del 95.25 %.

4.1.3. Sistema multimodal

Como se mencionó en secciones anteriores, se utilizaron 23 comandos de control para probar el sistema multimodal. Donde cada una de las frases de control fue asociada a una seña del alfabeto del LSM. El sistema completo se desarrolló en Matlab 2009a © donde se utilizo el toolbox de Redes Neuronales y la herramienta HTK para procesamiento de voz. Para la captura de imágenes del sensor kinect se utilizo el grapp de OpenNi para Matlab en su versión 1.0.

Se realizaron las pruebas con los mismos usuarios que entrenaron el sistema, las cinco mujeres y los cinco hombres que apoyaron en las señas y frases de entrenamiento. Cada usuario realizó las 23 señas del LSM y pronunció cada una de las frases de control, de tal manera que se tuvo la misma cantidad de muestras por cada comando de control, para probar el sistema. Con dichas pruebas se generó la base de datos de prueba, estos datos permitieron ejecutar el sistema de manera iterativa capturando muestras de voz y señas hasta alcanzar el valor máximo de convergencia definido en 95 %.

En el Cuadro 4.5, se muestran las veces que necesitaron proporcionar los datos los usuarios (filas) hasta reconocer el respectivo comando (columnas). Cabe mencionar que con esta forma de trabajo, el sistema asegura un porcentaje de reconocimiento del 95 % en las señas y comandos de voz proporcionados. Además, dicho porcentaje puede ser alcanzado en un máximo de dos iteraciones (valor dentro del cuadro).

Se observa que los comandos que más se repitieron fueron los correspondientes a las señas

Cuadro 4.5: Resultados con los usuarios de entrenamiento.

usu	Frases																					
	a	b	c	d	e	f	g	h	i	l	m	n	o	p	q	r	s	t	u	v	w	x	y
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	2	1	1	1	1	1	1	2	1	1	2	1	1	1	1	1	1	1	1	1	2	1
3	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1
4	1	2	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	2	1	1	1
6	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1
7	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	2	1	1	1	1	2	2	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
9	1	1	1	1	1	1	2	1	1	1	1	2	1	1	2	1	1	1	1	1	1	1	1
10	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

b, d, f, g, m y s, todas ellas a causa de errores en el módulo de reconocimiento de señas, pues en varios casos se confundía la seña del símbolo “s”, con la seña del símbolo “o”. Lo mismo sucedió con las señas “m” - “n”, “g” - “h” y “b” - “f”.

En cada caso de prueba, el máximo de iteraciones realizadas por el sistema para determinar el comando proporcionado es de dos veces, lo cual nos asegura que siempre se determina el comando y que a lo mas se realizan dos veces una misma seña y dos veces se pronuncia un mismo comando.

En la Figura 4.3, se muestra la interfaz gráfica del sistema multimodal. La interfaz fue realizada en Matlab y es capaz de mostrar en cada iteración la seña reconocida y el comando de voz interpretado. Además, se muestra el número de iteraciones que se han realizado antes de alcanzar el mínimo de porcentaje de reconocimiento (95 %).

4.1.4. Aplicación

La implementación del sistema de interacción multimodal fue probada en el software de simulación Roboworks©, diseñado por la Universidad de Texas. En dicho simulador fueron probados los 23 comandos de control listados en el Cuadro 3.4. Cabe mencionar que algunos de los comandos pueden incluir el uso de un brazo manipulador y de una plataforma móvil. Por ejemplo, el comando correspondiente a la letra “y” (Bot sirve vaso), el cual puede necesitar de un movimiento por parte de la plataforma y el correspondiente movimiento del brazo mecánico. Sin embargo, la mayoría de comandos requieren solo el movimiento de una plataforma, tal como se observa en la Figura 4.4, donde se muestra el desarrollo de la tarea asociada a la letra “v” (cube toma vaso).

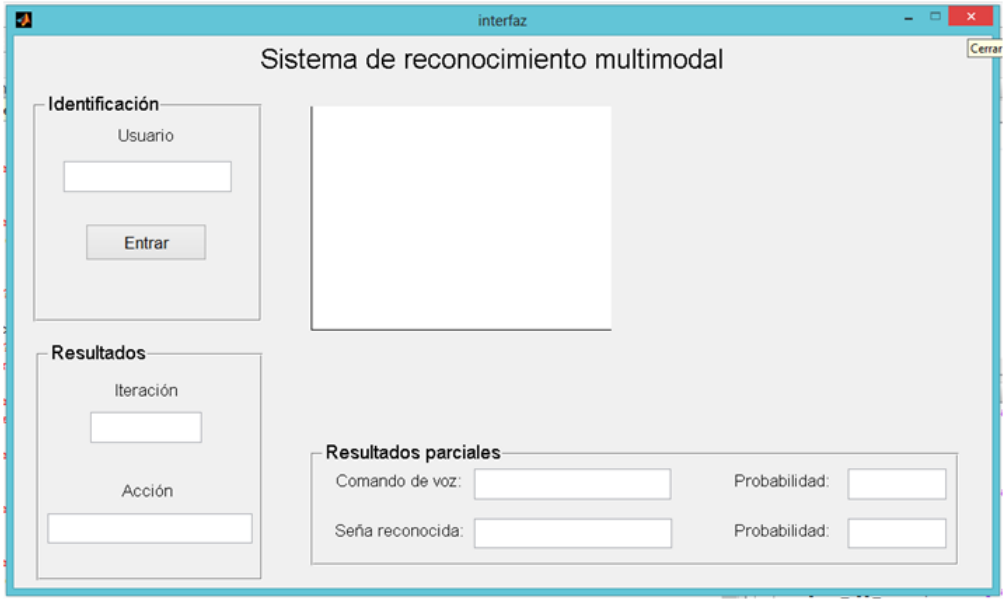


Figura 4.3: Interfaz del sistema de reconocimiento multimodal.

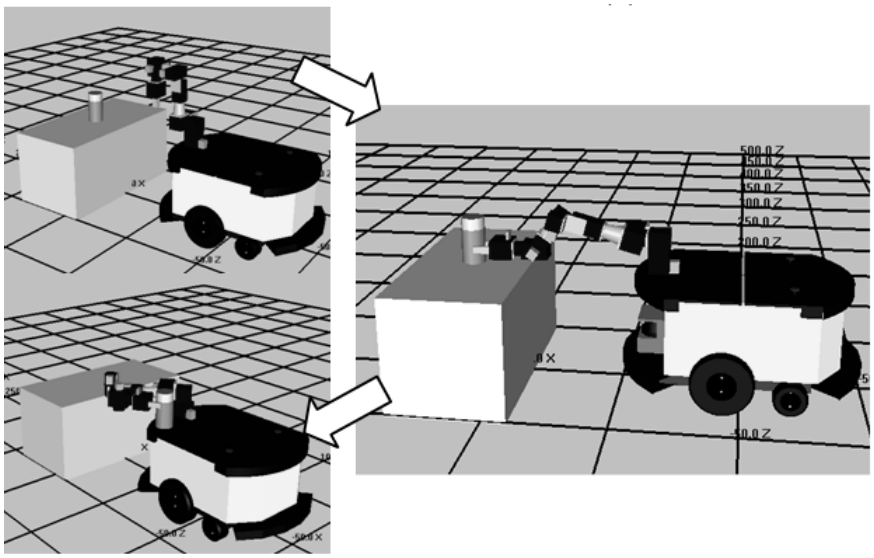


Figura 4.4: Ejemplo de simulación.

Capítulo 5

Conclusiones y Perspectivas

5.1. Conclusiones

El desarrollo del presente trabajo de tesis permitió obtener un sistema de reconocimiento multimodal capaz de reconocer e interpretar un conjunto de 23 frases de control pronunciadas en español mexicano, así como 23 símbolos pertenecientes al alfabeto del Lenguaje de Señas Mexicano. Cada una de las frases pronunciadas por el interlocutor se asociaba de manera directa con un símbolo del LSM, por ello, es necesario que para utilizar el sistema se proporcionen ambos datos de entrada. Por una parte, una imagen tridimensional capturada por un dispositivo Kinect (RGB y profundidad), y por la otra, una muestra de voz tomada por un micrófono conectado a la pc utilizada.

El sistema desarrollado permite realizar una comunicación casi natural entre una persona y un sistema robótico de servicio, pues no es necesario ningún dispositivo especial para obtener los datos de entrada del sistema. Las señas son realizadas con la mano del usuario y los comandos de voz son proporcionados por el mismo, sin necesidad de un dispositivo electrónico intermedio. El único dispositivo que se necesita para utilizar el sistema multimodal, es un sistema Kinect, el cuál comparado con cámaras de alto rendimiento es por mucho más barato y accesible al público en general.

Con este sistema se pretende llegar a la población en general pero sobre todo a personas con discapacidades motrices, auditivas, visuales o personas de la tercera edad. La finalidad de dicho sistema es proveer a estas personas un medio de interacción natural e intuitiva con sistemas mecánicos o robóticos, los cuales son completamente independientes del sistema desarrollado en esta tesis.

Con las técnicas utilizadas a lo largo del trabajo fue posible alcanzar un porcentaje de reconocimiento del 95 % en una o máximo dos iteraciones del sistema, es decir, se deben repetir a lo mas dos veces las señas y los comandos de voz para poder comunicarse con el

sistema robótico. Eso considerando las confusiones que generaba el módulo de señas, pues en repetidas ocasiones se confundían los símbolos de las señas b, d, f, g, m y s; muy probablemente por la similitud que existe entre algunos símbolos del alfabeto usado. Por su parte, el módulo de voz no mostró inconvenientes, pues las frases se reconocían sin problema alguno.

5.2. Perspectivas

Con los resultados obtenidos en este trabajo, se puede pensar en la implementación del reconocedor en un sistema robótico real para verificar el funcionamiento del mismo y observar el tiempo de reacción del sistema robótico. Aunado a esta perspectiva, será necesario hacer uso de una red de sensores para agilizar el procesamiento de las señales, sobre todo el proceso asociado al reconocimiento de señas, ya que la extracción de características resulta un tanto costosa computacionalmente. Al implementar una red de sensores, se puede realizar procesamiento en paralelo y agilizar el tiempo de respuesta del sistema real.

Como se menciona en el capítulo de señas, solamente se utilizan los símbolos que no incluyen movimiento. Por tal motivo, un posible trabajo a futuro es la inclusión de dichas señas para tener un espectro más amplio de señas reconocidas, abarcando las 27 señas del alfabeto del LSM. Esto permite un rango más amplio de aplicaciones y tareas que puede ejecutar el sistema robótico.

Finalmente, una extensión del presente trabajo pudiera ser la implementación de un sistema traductor de señas a voz y viceversa, proporcionando así, un medio de comunicación entre personas con discapacidades auditivas, visuales y personas que no padecen alguna de las discapacidades mencionadas.

Bibliografía

Al-Jarrah, O. y Halawani, A.: «Recognition of gestures in Arabic sign language using neuro-fuzzy systems». *Artificial Intelligence*, 2001, **133(1-2)**, pp. 117–138.

Atrash, A.; Kaplow, R.; Villemure, J.; West, R.; Yamani, H. y Pineau, J.: «Development and validation of a robust speech interface for improved human-robot interaction». *International Journal of Social Robotics*, 2009, **1(14)**, pp. 345–356.

Avilés, H.; Sucar, E.; Vargas, B.; Sanchez, J. y Corona, E.: «Markovito: A Flexible and General Service Robot». En: D. Liu; L. Wang y K.C. Tan (Eds.), *Studies in Computational Intelligence*, volumen 177, p. 401–423. Springer, Heidelberg, 2009.

Ayala-Ramirez, V.; Mota-Gutierrez, S. A.; Hernandez-Belmonte, U. H. y Sanchez-Yanez, R. E.: «A Hand Gesture Recognition System Based on Geometric Features and Color Information for Human Computer Interaction Tasks». Robotics Summer Meeting, 2011. University of Veracruz. Department of Artificial Intelligence.

Belongie, S.; Malik, J. y Puzicha, J.: «Shape Matching and Object Recognition Using Shape Contexts». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, **(24)**, p. 509–521.

Binh, N. D. y Ejima, T.: «Hand gesture recognition using fuzzy neural network». En: *Proceedings of the International Conference on Graphics, Vision and Image Processing*, pp. 1–6, 2005.

Bishop, C. M.: *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1996. ISBN 0198538642.

Blake, A. y Isard, M.: *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer, Cambridge, 1998. ISBN 9783540762171.

Breazeal, C.: *Designing Sociable Robots (Intelligent Robotics and Autonomous Agents)*. The MIT Press, Cambridge, Massachusetts, 2002. ISBN 978-0262025102.

Bretzner, L.; Laptev, I. y Lindeberg, T.: «Hand Gesture Recognition using Multi-Scale Colour Features, Hierarchical Models and Particle Filtering». En: *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, p. 423, 2002.

Bruemmer, D. J. y Walton, M.: «Collaborative tools for mixed teams of humans and robots». Multi-Robot Systems, 2003. Washington DC.

Burton, D. K.; Shore, J. E. y Buck, J. T.: «A generalization of isolated word recognition using vector quatization». *IEEE Transactions ASSP*, 1983, **8**, pp. 1021–1024.

Chen, Q.; Georganas, N. D. y Petriu, E. M.: «Hand Gesture Recognition Using Haar-Like Features and a Stochastic Context-Free Grammar». *IEEE Transactions on Instrumentation and Measurement*, 2008, pp. 1562–1571.

Cooper, M.; Keating, D.; Harwin, W. y Dautenhans, K.: «Robots in the classroom – tools for accessible education». En: *Fifth European Conference of the Advancement of Assistive Technology*, , 1999.

Cruz-Beltrán, A. y Acevedo-Mosqueda, M. A.: «Reconocimiento de Voz Usando Redes Neuronales Artificiales Backpropagation y Coeficientes LPC». *Congreso Internacional de Cómputo en Optimización y Software*, 2008, **6**, pp. 89–99.

Cuétara, J.: «Fonética de la Ciudad de México: Aportaciones desde las Tecnologías del Habla». MSc. Dissertation, 2004. Universidad Nacional Autónoma de México.

Dalal, N.; Triggs, B. y Schmid, C.: «Human detection using oriented histograms of flow and appearance». En: *Lecture Notes in Computer Science, European Conference on Computer Vision*, volumen 3952, pp. 428–441. Graz, Austria, 2006.

Davis, S. B y Mermelstein, P.: «Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences». *IEE Trans. Acoustics, Speech and Signal Processing*, 1980, **(28)**, p. 357–366.

Dempster, A. P.; Laird, N. M. y Rubin, D. B.: «Maximum-likelihood from incomplete data via the em algorithm». *Journal of Royal Statistics Society*, 1977, **39(1)**, p. 1–38.

Dias, D. B.; Madeo, R. C. B.; Rocha, T.; Biscaro, H. H. y Peres, S. M.: «Hand movement recognition for Brazilian Sign Language: A study using distance-based neural networks». En: *International Joint Conference on Neural Networks*, pp. 697–704, 2009.

Duke, M.; Hoffman, S. y Snook, K.: «Lunar Surface Reference Missions: A Description of Human and Robotic Surface Activities». NASA Lyndon B. Johnson Space Center, 2003. Houston, TX, USA.

El-Sawah, A.; Joslin, C.; Georganas, N. D. y Petriu, E. M.: «A Framework for 3D Hand Tracking and Gesture Recognition using Elements of Genetic Programming». En: *Fourth Canadian Conference on Computer and Robot Vision*, pp. 495–502, 2007.

Erol, A.; Bebis, G.; Nicolescu, M.; Boyle, R. D. y Twombly, X.: «Vision-based hand pose estimation: A review». *Computer Vision and Image Understanding*, 2007, **108(1-2)**, pp. 52–73.

Fletcher, R. y Reeves, C. M.: «Function minimization by conjugate gradients». *The computer journal*, 1964, **7(2)**, pp. 149–154.

Flores, L.; Vargas, A.; Olivier, A.; Kirschning, I. y Cervantes, O.: «Síntesis en Español Mexicano con el Método de Selección de Unidades de Longitud Variable». Ingeniería en Sistemas Computacionales, Universidad de las Américas de Puebla-México, 2001.

Fong, T.; Nourbakhsh, I. y Dautenhahn, K.: «A survey of socially interactive robots». *Robotics and Autonomous Systems*, 2003, **42(3-4)**, p. 143–166.

Fong, T. y Thorpe, C.: «Vehicle teleoperation interfaces». *Autonomous Robots*, 2001, **11(1)**, p. 9–18.

Freund, Y. y Schapire, R. E.: «A desicion-theoretic generalization of on-line learning and an application to boosting». *Journal of Computer and System Sciences*, 1997, **55(1)**, p. 119–139.

Fujimura, K. y Sako, Y.: «Shape Signature by Deformation». En: IEEE Computer Society (Ed.), *Shape Modeling International*, volumen 4. Aizu, Japan, 1999.

Gadh, R. y Prinz, F. B.: «Recognition of geometric forms using the differential depth filter». *Computer-Aided Design*, 1992, **24(11)**, pp. 583–598.

Gales, M. J. y Young, S. J.: «Robust continuous speech recognition using parallel model combination». En: *IEEE Transactions on Speech and Audio Processing*, volumen 4, pp. 352–359, 1996.

Golem: «Universidad Autónoma de México». http://leibniz.iimas.unam.mx/ luis/golem/, 2012. Último acceso, 27 de enero.

Goodrich, M. A. y Schultz, A. C.: «Human-Robot Interaction: A Survey». *Foundations and Trends in Human-Computer Interaction*, 2007, **1(3)**, pp. 203–275.

Green, P.; Hawley, M.; Enderby, P.; Brownsell, S.; Hatzis, A.; Cunningham, S.; Parker, M.; Carmichael, J.; Palmer, R. y O'Neill, P.: «STARDUST Speech Training and Recognition for Dysarthric Users of Assistive Technology». En: *Proc. of Association for the Advancement of Assistive Technology in Europe (AAATE)*, pp. 959–963, 2003.

Hagan, M. T.; Demuth, H. B. y Beale, M. H.: *Neural Network Design*. PWS Pub, 1995. ISBN 0-9717321-0-8.

Hagan, M. T. y Menhaj, M.B.: «Training feedforward networks with the Marquardt algorithm». *IEEE Transactions on Neural Networks*, 1994, **5(6)**, pp. 989–993.

Han, J.; Jo, M.; Park, S. y Kim, S.: «The educational use of home robots for children». En: *IEEE International Workshop on Robots and Human Interactive Communication*, pp. 378–383, 2005.

Hashimoto, M.; Takahashi, K. y Shimada, M.: «Wheelchair control using an EOG- and EMG-based gesture interface». En: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, pp. 1212 – 1217, 2009.

Haykin, S.: *Neural networks a Comprehensive Foundation.* Pearson Prentice Hall, 1999. ISBN 0132733501.

——: *Neural networks and learning machines.* Prentice Hall, 2009. ISBN 0131471392.

Hernández-López, J. J.; Quintanilla-Olvera, A. L.; López-Ramírez, J. L.; Rangel-Butanda, F. J.; Ibarra-Manzano, M. A. y Almanza-Ojeda, D. L.: «Detecting objects using color and depth segmentation with Kinect sensor». En: *The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science*, volumen 3, p. 196–204, 2012.

Ho-Sub, Y.; Jung, S.; Younglae, J. B. y Hyun, S. Y.: «Hand gesture recognition using combined features of location, angle and velocity». *Pattern Recognition*, 2001, **34(7)**, pp. 1491–1501.

Ikemura, S y Fujiyoshi, H.: «Real-Time Human Detection using Relational Depth Similarity Features». En: *Lecture Notes in Computer Science, Computer Vision*, volumen 6495, pp. 25–38, 2010.

Jain, A. K.; Klare, B. y Park, U.: «Face recognition: Some challenges in forensics». En: *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, pp. 726–733, 2011.

Johansson, E. M.; Dowla, F. U. y Goodman, D. M.: «Backpropagation learning for multilayer feed-forward neural networks using the conjugate gradient method». *International Journal of Neural Systems*, 1991, **2(4)**, pp. 291–301.

Juang, B. H. y Rabiner, L. R.: «The segmental k-means algorithm for estimating the parameters of hidden markov models». *IEEE Transaction on Accoustic, Speech and Signal Processing*, 1990, **38(9)**, p. 1639–1641.

Jurafsky, D. y Martin, J. H.: *Speech and Language Processing.* Pearson: Prentice Hall, 2009. ISBN 978-0131873216.

KADOUS, M. W.: «Machine Recognition of Auslan Signs Using PowerGloves: Towards Large-Lexicon Recognition of Sign Language». En: *Proceedings of the Workshop on the Integration of Gesture in Language and Speech*, pp. 165–174, 1996.

KARAMI, A.; ZANJ, B. y SARKALEH, A. K.: «Persian sign language (PSL) recognition using wavelet transform and neural networks». *Expert Systems with Applications*, 2011, **38(3)**, pp. 2661–2667. NY, USA.

KENNEDY, W. G.; BUGAJSKA, M.; MARGE, M.; ADAMS, W.; FRANSEN, B. R.; PERZANOWSKI, D.; SCHULTZ, A. C. y TRAFTON, J. G.: «Spatial representation and reasoning for human-robot collaboration». En: *Proceedings of the AAAI national Conference on Artificial Intelligence*, , 2007.

KIESLER, S. y HINDS, P.: *Human-robot Interaction: A Special Double Issue of human-computer Interaction*. Human-computer interaction. CRC Press, Mahwah, New Jersey, 2004. ISBN 978-0805895537.

KOLLORZ, E.; PENNE, J.; HORNEGGER, J. y BARKE, A.: «Gesture recognition with a Time-Of-Flight camera». *International Journal of Intelligent Systems Technologies and Applications*, 2008, **5(3)**, pp. 334–343.

KOLMOGOROV, A. N.: «On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition». Doklady Akademiia Nauk, 1957. SSRR 114, 953-956.

KOSUGE, K.; HAYASHI, T.; HIRATA, Y. y TOBIYAMA, R.: «Dance partner robots – Ms. DancerR». En: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volumen 4, pp. 3459–3464, 2003.

KULYUKIN, V.; GHARPURE, C.; NICHOLSON, J. y OSBORNE, G.: «Robot-assisted wayfinding for the visually impaired in structured indoor environments». *Autonomous Robots*, 2006, **21(1)**, p. 29–41.

LANKTON, S. y TANNENBAUM, A.: «Localizing Region Based Active Contours». *IEEE Transaction on Image Processing*, 2008, **17(11)**, pp. 2029–2039.

LE-CUN, Y.: «Une procedure d'apprentissage pour reseau a seuil assymetrique». *Cognitiva*, 1985, **85**, pp. 599–604.

LEGER, P. C.; TREBI-OLLENNU, A.; WRIGHT, J. R.; MAXWELL, S. A.; BONITZ, R. G.; BIESIADECKI, J. J.; HARTMAN, F. R.; COOPER, B. K.; BAUMGARTNER, E. T. y MAIMONE, M. W.: «Mars exploration rover surface operations: Driving spirit at gusev crater». En: *IEEE International Conference on Systems, Man, and Cybernetics*, volumen 2, pp. 1815–1822, 2005.

Leggetter, C. J. y Woodland, P. C.: «Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models». *Computer Speech and Language*, 1995, **9(2)**, p. 171–185.

Liu, H. y Li, X.: «A Selection Method of Speech Vocabulary for Human-Robot Speech Interaction». *IEEE International Conference on Systems Man and Cybernetics*, 2010, pp. 2243–2248.

Lone, M. A.; Zakariya, S. M. y Ali, R.: «Automatic Face Recognition System by Combining Four Individual Algorithms». En: *International Conference on Computational Intelligence and Communication Networks*, pp. 222–226, 2011.

López-Monroy, A. P. y Leal-Meléndrez, J. A.: «Reconocimiento de Gestos basado en Modelos Ocultos de Markov utilizando el Kinect». Instituto Nacional de Astrofísica,Óptica y Electrónica, 2011. Puebla, México.

Lu, X.; Chia-Chih, C. y Aggarwal, J. K.: «Human Detection Using Depth Information by Kinect». En: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 15–22, 2011.

Luengo, I.; Navas, E.; Hernáez, I. y Sánchez, J.: «Reconocimiento automático de emociones utilizando parámetros prosódicos». *Procesamiento del lenguaje natural*, 2005, **35**, pp. 13–20.

Luis-Pérez, F. E.; Trujillo-Romero, F. J. y Martínez-Velazco, W.: «Control of a Service Robot Using the Mexican Sign Language». En: Ildar Batyrshin y Grigori Sidorov (Eds.), *Proceedings of Advances in Soft Computing (LNAI)*, volumen 7095, pp. 419–430. Puebla, México, 2011.

Mak, M.; Hsiao, R. y Mak, B.: «A comparison of various adaptation methods for speaker verification with limited enrollment data». En: *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, pp. 929–932, 2006.

Maldeni, k.; Wijesundera, L.; Morris, J.; Jawed, K.; Saz, O. y Lleida, E.: «Gesture Recognition using High Resolution Stereo». Department of Electrical and Computer Engineering, 2011. Auckland, New Zealand.

Malik, S. y Laszlo, J.: «Visual touchpad: a two-handed gestural input device». En: *Proceedings of the 6th international conference on Multimodal interfaces*, pp. 289–296, 2004.

Maung, T. H. H.: «Real-time hand tracking and gesture recognition system using neural networks». *World Academy of Science, Engineering and Technology*, 2009, p. 466–470.

McLachlan, G. J. y Krishnan, T.: *The EM algorithm and extensions.* John Wiley & Sons, 1997.

MICROSOFT: «Xbox 360 + Kinect». http://www.xbox.com/en-us/kinect, 2009. Último acceso, 28 de Agosto.

MO, Z.; LEWIS, J. P. y NEUMANN, U.: «SmartCanvas: a gesture-driven intelligent drawing desk system». En: *Proceedings of the 10th international conference on Intelligent user interfaces*, pp. 239–243, 2005.

MUÑOZ, R.; AGUIRRE, E. y GARCÍA, M.: «People Detection and Tracking using Stereo Vision and Color». *Image and Vision Computing*, 2007, **25(6)**, pp. 995–1007.

MUNIB, Q.; HABEEB, M.; TAKRURI, B. y AL-MALIK, H. A.: «American sign language (ASL) recognition based on Hough transform and neural networks». *Expert Systems with Applications*, 2007, **32(1)**, pp. 24–37.

NAGI, J.; DUCATELLE, F.; DI-CARO, G. A.; CIRESAN, D.; MEIER, U.; GIUSTI, A.; NAGI, F.; SCHMIDHUBER, J. y GAMBARDELLA, L. M.: «Max-pooling convolutional neural networks for vision-based hand gesture recognition». En: *IEEE International Conference on Signal and Image Processing Applications*, pp. 342–347, 2011.

NENE, S. A.; NAYAR, S. K. y MURASE, H.: «Columbia Object Image Library (COIL-20)». Columbia University, 1996. Technical Report CUCS-006-96.

NGUYEN, T. T. M.; PHAM, N. H.; DONG, V. T. y NGUYEN, T. T. H., V. S.AND TRAN: «A fully automatic hand gesture recognition system for human-robot interaction». En: *Proceedings of the Second Symposium on Information and Communication Technology*, pp. 112–119, 2011.

NICKEL, K. y STIEFELHAGEN, R.: «Visual recognition of pointing gestures for human–robot interaction». *Image and Vision Computing*, 2007, **25(12)**, p. 1875–1884.

O'HAGAN, R. G.; ZELINSKY, A. y ROUGEAUX, S.: «Visual Gesture Interfaces for Virtual Environments». *Interacting with Computers*, 2002, **14**, pp. 231–250.

OROPEZA-RODRÍGUEZ, J. L. y SUÁREZ-GUERRA, S.: «Algoritmos y métodos para el Reconocimiento de Voz en Español Mediante Silabas». *Computación y Sistemas*, 2006, **9(3)**, pp. 270–286.

OVIATT, S.; DARREL, T. y FLICKNER, M: «Multimodal Interfaces that Flex Adapt and Persist». *Communications of the ACM*, 2004, **47(1)**.

PARKER, D. B.: «Learning-Logic: Casting the cortex of the human brain in silicon». Center for Computational Research in Economics and Management Science, MIT, Cambridge, MA, 1985. Technical Report TR-47.

PAULRAJ, M. P.; YAACOB, S.; BIN ZANAR-AZALAN, M. S. y PALANIAPPAN, R.: «A phoneme based sign language recognition system using skin color segmentation». En: *6th International Colloquium on Signal Processing and Its Applications*, pp. 1–5, 2010.

Posada-Gomez, R.; Sanchez-Medel, L. H.; Hernandez, G. A.; Martinez-Sibaja, A.; Aguilar-Laserre, A. y Leija-Salas, L.: «A Hands Gesture System Of Control For An Intelligent Wheelchair». En: *4th International Conference on Electrical and Electronics Engineering*, pp. 68–71, 2007.

Preece, J.; Carey, T.; Rogers, Y.; Holland, S.; Sharp, H. y Benyon, D.: *Human-Computer Interaction*. Ics Series. Addison-Wesley Publishing Company, 1994. ISBN 9780201627695.

Rabiner, L. R.: «A Tutorial on Hiden Markov Models and Selected Applications in Speech Recognition». En: *Proceedings on IEEE*, volumen 77, pp. 257–286, 1989.

Rahimi, M. y Karwowski, W.: *Human-Robot Interaction*. Taylor & Francis Inc, Bristol, PA, USA, 1992.

Rossenblatt, F.: «The perceptron, a perceiving and recognizing automation». Cornell Aeronautical Laboratory, 1957. Report No. 85-460-1.

Rumelhart, D. E.; Hinton, G. E. y Williams, R. J.: «Learning representation by Back-propagating errors». *Nature*, 1986, **323**, pp. 533–536.

Salas, J. y Tomasi, C.: «People detection using color and depth images». En: *Lecture Notes in Computer Science, Pattern Recognition*, volumen 6718, pp. 127–135, 2011.

Schuller, B.; Rigoll, G.; Can, S. y Feussner, H.: «Emotion sensitive speech control for human-robot interaction in minimal invasive surgery». *Robot and Human Interactive Communication*, 2008, pp. 453–458.

Solís-Villarreal, J. F.: «Modelo de procesamiento de voz para la clasificación de estados». Instituto Politécnico Nacional, 2011. Centro de Investigación en Computación.

Tretin, E. y Gori, M.: «A survey of hybrid ANN/HMM models for automatic speech recognition». *Neurocomputing-37*, 2001, pp. 91–126.

Trigo, T. R. y Pellegrino, S. R. M.: «An analisys of features for hand-gesture classification». En: *Proceedings of International Conference on Systems, Signals and Image Processing*, p. 412–415, 2010.

Trujillo-Romero, F. J.; Caballero-Morales, S. O. y Luis-Pérez, F. E.: «Sistema de reconocimiento del habla para identificación de usuario mediante el uso de codificación de predicción lineal y redes neuronales». *VIII Semana Nacional de Ingeniería Electrónica*, 2012, pp. 386–395.

Vargas, L. P.; Barbara-Jiménez, L. y Mattos, L.: «Sistema de identificación de Lenguaje de Señas usando Redes Neuronales Artificiales». *Revista Colombiana de Física*, 2010, **42(2)**.

VILLA-ANGULO, R. y HIDALGO-SILVA, H.: «A wearable neural interface for real time translation of spanish deaf sign language to voice and writing». *Journal of Applied Research and Technology*, 2005, **3(3)**, pp. 169–186.

VIOLA, P. y JONES, M: «Robust real-time object detection». Cambridge Res. Lab, Cambridge, MA, 2001. Tech. Rep. CRL2001/01.

VITERBI, A. J.: «Error bounds for convolutional codes and an asymptotical optimal decoding algorithm». *IEEE Transaction on Information Theory, IT*, 1967, **(13)**, p. 260–269.

WAIBEL, A.; HANAZAWA, T.; HINTON, G.; SHIKANO, K. y LANG, K. J.: «Phoneme recognition using time-delay neural networks». *IEEE transaction on Acoustics, Speech and Signal Processing*, 1989, **37(3)**, pp. 328–339.

WALDRON, M. B. y KIM, S.: «Isolated ASL sign recognition system for deaf persons». *IEEE transaction on Rehabilitation Engineering*, 1995, **3(3)**, pp. 261–217.

WATANABE, S.: *Pattern Recognition: Human and Mechanical.* Wiley, New York, 1985.

WELLS, P. y DEGUIRE, D.: «TALON: A universal unmanned ground vehicle platform, enabling the mission to be the focus». En: *Proceedings of SPIE—Unmanned Ground Vehicle Technology*, volumen 7, 2005.

WERBOS, P.: «Beyond Regression: New tools for prediction and analisys in the behavioral sciences». Harvard University, Cambridge, MA, 1974. Ph.D Thessis.

WILSON, A. D.: «Using a depth camera as a touch sensor». En: *Proceeding of ACM International Conference on Interactive Tabletops and Surfaces*, volumen 133, pp. 69–72, 2010.

WOODLAND, P. C.; ODELL, J. J.; VALTCHEV, V. y YOUNG, S. J.: «Large vocabulary continuous speech recognition using HTK». En: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volumen 2, 1994.

WU, Y. y HUANG, T. S.: «Hand modeling, analysis and recognition». *IEEE Signal Processing Magazine*, 2001, **18(3)**, pp. 51–60.

YAN, R.; TEE, K. P.; CHUA, Y.; LI, H. y TANG, H: «Gesture Recognition Based on Localist Attractor Networks with Application to Robot Control». *IEEE Computational Intelligence Magazine*, 2012, **7(1)**, pp. 64–74.

YANCO, H. A.: «Development and testing of a robotic wheelchair system for outdoor navigation». En: *Proceedings of the 2001 Conference of the Rehabilitation Engineering and Assistive Technology Society of North America*, RESNA Press, 2001.

YANG, M. H.; KRIEGMAN, D. J. y AHUJA, N.: «Detecting faces in images: A survey». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, **24(1)**, pp. 34–58.

YOUNG, S.: «A Review of large-vocabulary continuous-speech recognition». En: *IEEE signal processing magazine*, pp. 45–57, 1996.

YOUNG, S. y WOODLAND, P.: *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, 2006.

YUAN-HSIANG, C. y CHEN-MING, C: «Automatic Hand-Pose Trajectory Tracking System Using Video Sequences». User Interfaces, Rita Matrai (Ed.), 2010. ISBN: 978-953-307-084-1.

ZHANG, T.; ZHU, B.; LEE, L. y KABER, D.: «Service robot anthropomorphism and interface design for emotion in human-robot interaction». En: *IEEE International Conference on In Automation Science and Engineering*, pp. 674–679, 2008.

ZHAO, W.; CHELLAPPA, R. y PHILLIPS, P. J.: «Face recognition: A literature survey». *ACM Computing Surveys*, 2003, **35(4)**, pp. 399–458.

ZWEIG, G. y PICHENY, M.: «Advances in Large Vocabulary Continuous Speech Recognition». En: *Advances in Computers, Elsevier Science*, , 2004. ISBN-10:0-12-012160-3.