



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS COMPUTACIONALES

MODELADO DE MOVIMIENTOS DE UNA MANO
ROBÓTICA PARA CONTROL POR MEDIO DE VOZ

Tesis que para obtener el grado de
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:
RICAÑO ZÚÑIGA RODOLFO

ASESOR: Dr. ALEJANDRO RANGEL HUERTA
CO-ASESOR: Dr. JUAN ALBERTO GUEVARA JARAMILLO

ENERO 2019



Contenido

1. INTRODUCCIÓN	3
2. ANTECEDENTES.....	6
3. METODOLOGÍA	8
3.1. Sistemas Difusos.....	9
3.1.1. Reglas Difusas IF-THEN	10
3.1.2. Sistema de Inferencia Difuso	11
3.1.2.1. Modelo Mamdani	12
3.1.2.2. Modelo Takagi-Sugeno.....	13
3.2. Redes Neuronales.....	14
3.2.1. Propagación hacia atrás.....	15
3.3. ANFIS.....	19
3.4. Extracción de Características.....	20
3.4.1. Coeficientes Ceptrales de Mel (MFCCs).....	21
3.4.2. Análisis de Componente Principales (PCA).....	21
3.4.3. Exponentes de Lyapunov.	23
3.5. Entrenamiento de la Red ANFIS con la extracción de características.	27
4. ETAPA DE MODELADO Y CONTROL.	29
4.1. Diseño de la Mano.....	29
4.2. Etapa de Control.....	31
5. RESULTADOS.....	34
6. CONCLUSIONES	41

Resumen: Se diseña y controla una mano robótica tanto para el uso de prótesis, en caso de que una persona pierda este miembro, como auxiliar en la industria, para el manejo de sustancias peligrosas para la salud de la persona. Esta mano funcionará por medio de un control de voz, en el cual nos permitirá moverla, escogiendo una de las 7 frases a utilizar, siendo esta asignada a un movimiento en específico. De estos comandos, se extraerán las características más importantes, permitiéndonos conservar los datos más relevantes del audio, ya que logra quitar el ruido y caracterizar la información para que así sea más fácil para el clasificador separar una de otra. Para esto último se compararon diferentes métodos, siendo estos el Análisis de Componentes Principales (PCA), Exponentes de Lyapunov o Coeficientes Ceptrales de Frecuencia de Mel (MFCC), métodos que ya sea quitando correlación de los datos, medir que tan altamente no lineal es una trayectoria, como basarse en el funcionamiento del habla, nos permite poder conservar solo los datos que son importantes. La razón por la que se está trabajando con voz y no con cualquiera de las otras bio-señales existentes en el cuerpo humano es debido a que la voz es más fácil de modular su magnitud, pudiendo tener más detalle a la hora de analizar el audio en comparación al analizar las otras bio-señales que normalmente son más pequeñas. Después de la extracción de características se usará un sistema neuro-difuso adaptativo (ANFIS), que se encargará de la clasificación de estos comandos que no es permitirá hacer el movimiento en relación a la orden que se le dé a la mano. Este clasificador, al ser un clasificador difuso, podrá permitirnos una mayor flexibilidad a la hora de discriminar cada uno de las frases utilizadas para el control de la mano. Se buscará también analizar los diferentes patrones comunes de estos movimientos para hacer una relación con respecto a los comandos de voz, esto es, decidir hasta que nivele cierra/abre la mano para poder tomar un objeto y así la persona pueda alzarlo.

1. INTRODUCCIÓN

Cesare Rossi et al. Propuso el modelo de una mano robótica con controladores para tendones con el objetivo de estudiar la factibilidad de construir una mano, en la cual el movimiento sea controlado con un simple actuador lineal. En este trabajo se propone que, para un mejor desempeño, la mano debe tener un alto número de grados de libertad, capacidad de diferentes tipos de agarre con fuerza y precisión variante, y la habilidad para ajustar la fuerza y velocidad de acuerdo a los diferentes objetos que se desean tomar [1].

En otro trabajo se hizo el análisis de las posturas óptimas de una mano para el agarre. De este trabajo se concluyó que el mejor agarre depende de la cinemática de la mano, de la tarea realizada y de las características del objeto a manipular [2]. Esto se puede lograr utilizando imágenes de las diferentes posturas de los dedos obtenidas por medio de resonancia magnética [3].

También se han realizados trabajos para analizar el uso de la mano en conjunto con los sensores de presión que permitirán un análisis más profundo de la dinámica del sistema, ya que permitiría un mejor desempeño en ambientes desconocidos como poder lidiar con incertidumbres significativas [4].

Otros autores usan como enfoque el control de postura y agarre de una mano artificial, usando un controlador integral de modos deslizantes [5]. El control por modos deslizantes consiste en forzar al estado del sistema a un comportamiento deseado, para el presente caso, llevar la dinámica del sistema a una superficie deslizante, siendo este método normalmente se escoge debido a su robustez y comportamiento ante modelos altamente no lineales [6].

Se ha realizado estudios en cuanto a la anatomía de una mano robótica para comprobar la hipótesis de que, con mayor número de dedos, se aumenta las combinaciones de las posiciones de los dedos al momento de tomar un objeto [7].

Por otro lado, se ha diseñado, usando enfoques similares, se han utilizado apoyos como los son un exoesqueleto de mano robótica controlado por electromiografía (EMG) para rehabilitación bilateral, donde se trata de mejorar la eficiencia del agarre usando la terapia de espejo, leyendo con los sensores EMG los movimientos de una mano para duplicarlo en la mano que este bajo rehabilitación [8] o el uso de dedos supernumerarios robóticos que sirven de apoyo a personas con limitaciones en la mano, buscando que haya una coordinación entre el sujeto y esta nueva prótesis [9].

En cuanto al reconocimiento por voz, las características de un hablante individual no son fácilmente identificables, pero son únicas debido a la fisiología del tracto supralaríngeo y los hábitos de articulación aprendidos. Las características ideales para hacer un correcto análisis de la señal deben mostrar una alta variabilidad entre hablantes y una baja variabilidad con un solo hablante, ser resistente al mimetismo, tener una alta frecuencia de ocurrencia en materiales relevantes, ser robusto en transmisión y ser relativamente fácil de extraer y medir [10].

Dong-Il Kim y Byung-Cheol Kim propusieron un método usando la extracción de Coeficientes Cepstrales en las Frecuencias de Mel, que me permitirán obtener la información relevante de la señal, para luego utilizar modelos escondidos de Markov para su posterior reconocimiento. Esto no sin antes hacer un pre-procesamiento de la señal y luego pasarla al dominio de las frecuencias por medio de la transformada rápida de Fourier [11].

Otra técnica que ayuda a la extracción de características es utilizando Wavelets y un PCA mejorado, el que permite quitar el ruido de la señal, apoyándose en la habilidad de PCA que tiene de de-correlacionar los coeficientes de los datos de prueba, en este caso mejorándose al utilizar wavelets que nos permite encontrar una relación entre ellos [12].

En otros trabajos, se ha hecho reconocimiento del habla por medio de modelos redes neuronales profundas y transductores pesados de estado finitos,

donde se clasifican las diferentes señales por medio de una red neuronal simple, para luego hacer su respectiva decodificación, usando transductores, siendo estas máquinas de estado finito con una entrada y salida por estado [13].

Estas redes neuronales se han mejorado utilizando algoritmos genéticos para el cálculo de los pesos que hay en las conexiones entre neuronas dada las ventajas que tienen al permitir salir de los mínimos locales que generan el entrenamiento por retroalimentación [14].

De este marco se derivan el ANFIS, siendo un Sistema de Inferencia Difuso implementado como una red adaptativa [15]. Esto quiere decir, que las reglas y conjuntos difusos serán construidos usando el entrenamiento que se aplica a las redes neuronales, por lo cual se utiliza un modelo Takagi-Sugeno, dado que su salida es una función, la cuál puede ser optimizada de forma similar a una red neuronal.

Por lo tanto, usando como punto de partida los trabajos ya antes mencionados, se hará un control para una mano robótica que nos permita hacer el agarre. Esto se hará, dado el comportamiento caótico que tienen las señales de voz, extrayendo las características más relevantes de 7 comandos de voz usando tanto los métodos de PCA, exponentes de Lyapunov y MFCC, como la combinación de PCA y Lyapunov o PCA y MFCC. Luego estas características se usarán para entrenar a la ANFIS de acuerdo a cada una de las frases o de las abreviaciones para ver cuál es más efectivo.

2. ANTECEDENTES

De los trabajos previamente comentados, es necesario enfocarnos en tres principalmente, dado la importancia que tiene con respecto a la investigación que se hizo, empezando por el artículo escrito por Georg Stillfried et al. ya que este analiza la cinemática de la mano humana con respecto a la fuerza de distribución durante el apretón como un poderoso agarre, reduciendo la contracción del tejido cuando se ejecutan las fuerzas a través de los dedos opuestos y optimizando las caras de contacto. Cuantificando esta optimalidad es la clave para construir manos robóticas, pero entendiendo que la moción exacta del dedo humano es también un importante punto. El objetivo del método presentado en este artículo es determinar la orientaciones precisas y posiciones de los ejes de rotación de las articulaciones de los dedos usando imágenes obtenidas por medio de resonancias magnéticas. Los huesos son segmentados de las imágenes y sus poses son calculadas con respecto a las posturas de referencia. Los ejes de orientación y posición son adaptados numéricamente para emparejar los movimientos de los huesos ya medidos [3].

Pero dado que el enfoque implica el manejo del audio, es necesario enfocarnos en el reconocimiento de voz, el cuál es usado ampliamente en entornos de aplicaciones móviles, aunque este resulta ser uno de los tópicos sin resultados significantes, esto debido a la dificultad al momento de extraer las características importantes. En este caso, los modelos ocultos de Markov y las Redes Neuronales son conocidos por ser efectivos para el procesamiento de la señal de la voz mientras que MFCC son de los métodos más populares para extraer características en el campo del reconocimiento de voz [11].

Otro método usado integra el análisis de componentes principales en el paquete de descomposición de wavelet. Se extiende la técnica de PCA para mejorar el habla, experimentado en datos corruptos por ruido gaussiano y no estacionario. Este método integra la habilidad de de-correlacionar los coeficientes por extraer una

relación lineal con el análisis de paquetes wavelet para derivar vectores de características utilizados para mejorar el habla. Esto nos permite operar con una conveniente función comprimida sobre esos nuevos coeficiente removiendo el ruido sin degradar el habla [12].

3. METODOLOGÍA

Se construye un sistema traductor de voz para manipular una mano robótica generando un sistema de control basado en la voz y la vista del usuario. Para el control de voz se utilizará un Sistema de Inferencia Difuso Neuro-Adaptativo (ANFIS), donde las entradas serán los comandos de voz y su salida los grados en los cuales se moverá el servomotor. En la figura 1 se muestra el diagrama de bloques donde se describe el funcionamiento del sistema. Como se puede observar, el humano hará la parte de agente reactivo, dará el comando de voz correspondiente para tomar un objeto, mientras que con la vista se ayudará para ajustarse a su forma y tamaño.

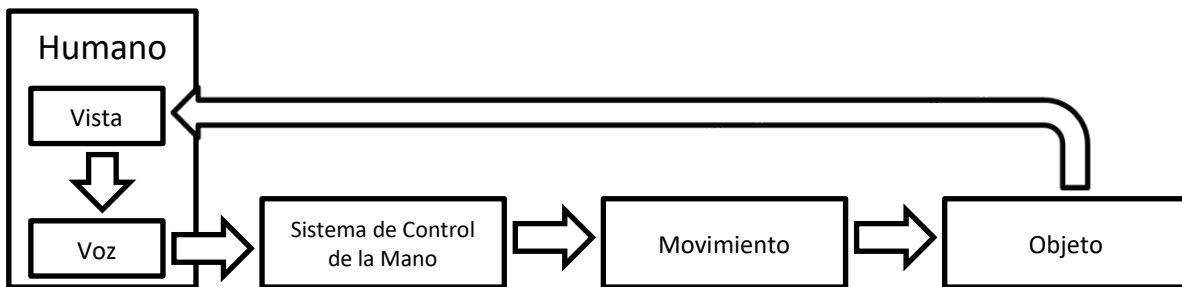


Figura 1. Esquema general del funcionamiento de la mano

En la figura 2 se muestra el diagrama de bloques de la metodología a seguir. Se empezará seleccionando 7 comandos a seguir, los cuales se muestran en la Tabla 1. Luego se graban estos comandos, siendo en este caso mezclados con ruido para crear un conjunto de entrenamiento viable.

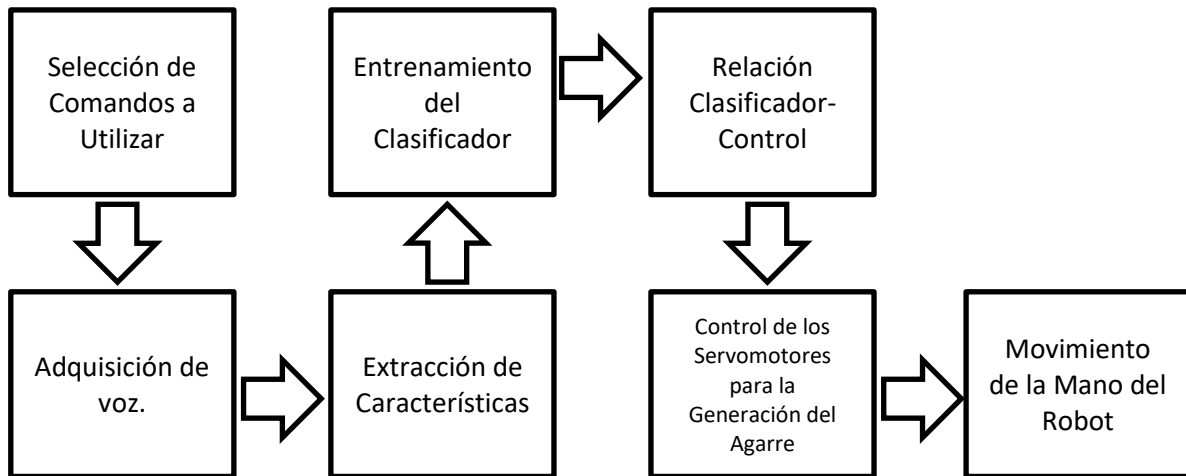


Figura 2. Metodología para el diseño de la mano.

Estos métodos se evalúan tanto en frases completas como en abreviaciones para saber la viabilidad de usar ya sea uno u otro, esto con la intención de también pensar en la comodidad de la persona que fuera a usar la mano.

Frases Completas	Abreviaciones
Mucho Menos	MUM
Poco Menos	POM
Menos	MEN
Agarre	NEU
Más	MAS
Poco Más	OMA
Mucho Más	UMA

Tabla 1. Frases utilizadas para el control de voz de la mano.

3.1. Sistemas Difusos

La lógica difusa fue planteada por Zadeh en 1965. Constituye una generalización de la lógica clásica, que es determinista: “verdadero” o “falso”. Sin

embargo, la lógica difusa, a diferencia de la lógica clásica, tiene fronteras imprecisas. La lógica difusa, al ser una forma de lógica multivaluada, puede manejar el razonamiento aproximado. Por esta razón, las variables lingüísticas se utilizan en la definición de conjuntos. Así, por ejemplo, una variable lingüística como podría ser la edad, puede tener valores tales como el de joven y viejo.

Otra ventaja importante es la transición de pertenencia de un conjunto a otro. Retomando el ejemplo de la variable de edad, consideremos dos conjuntos, uno A, jóvenes y otro B, viejos. Cada uno de estos tendrá una función que define su membresía del grupo y que será definida como $\mu(x)$, en la cual x representa el grado de membresía del conjunto. Por tanto, entre más cerca de uno (1) se encuentre el valor de x , mayor será la membresía de dicho conjunto [16].

3.1.1. Reglas Difusas IF-THEN

Las reglas difusas IF-THEN son expresiones de la forma SI A ENTONCES B, donde A y B son etiquetas de conjuntos difusos [15] caracterizadas por apropiadas funciones de membresía x y y [18]. Dicho de otra forma, estas reglas asumen la forma

$$\textit{if } x \textit{ is } A \textit{ then } y \textit{ is } B \quad (1)$$

A menudo “ x is A” es llamado antecedente o premisa, mientras que “ y is B” es llamado la consecuencia o conclusión. En esencia Ec. 1 describe una relación entre dos variables x y y ; esto sugiere que una regla difusa IF-THEN sería definida como una relación binaria difusa R en el espacio del producto $X \times Y$. Esta relación puede ser interpretada como A implica B, pudiendo ser escrita en 4 diferentes fórmulas [17].

- Implicación Material

$$R = A \rightarrow B = \neg A \cup B \quad (2)$$

- Calculo Proposicional

$$R = A \rightarrow B = \neg A \cup (A \cap B) \quad (3)$$

- Calculo Proposicional Extendido

$$R = A \rightarrow B = (\neg A \cap \neg B) \cup B \quad (4)$$

- Generalización del Modus Ponens

$$\mu_R(x, y) = \sup\{c | \mu_A(x) \bar{*} c \leq \mu_B(y) \text{ and } 0 \leq c \leq 1\} \quad (5)$$

Donde $R = A \rightarrow B$ y $\bar{*}$ es un operador T-Norma, siendo esta utilizada para calcular los valores de membresía de la *intersección* de dos o más conjuntos difusos.

A pesar de que las fórmulas previas se vean diferentes, todas son equivalentes a $\neg A \cup B$.

En la lógica clásica, si el antecedente es cierto, el consecuente también es cierto. En la lógica difusa, por otro lado, todas las reglas se ejecutan parcialmente, y el consecuente es cierto en cierto grado [17].

3.1.2. Sistema de Inferencia Difuso

Un Sistema de Inferencia Difuso (FIS) es una estructura computacional basado en los conceptos de teoría de conjuntos difusos, reglas difusas “IF-THEN” y razonamiento difuso. La estructura básica de un sistema de inferencia difuso consiste en tres componentes conceptuales: una base de reglas, cual contiene un base de reglas difusas, una base de datos, que define las funciones de membresía usadas en las reglas difusas, un mecanismo de razonamiento, el cual realiza el procedimiento de inferencia difuso sobre las reglas y los datos dados para obtener una razonable salida o conclusión, una interfaz de difuminación, que transforma los datos duros en grados de relación con los valores lingüísticos y una interfaz de des difuminación, que transforma los resultados difusos de la inferencia en salidas duras [18, 21].

Se mostrarán a continuación dos tipos de inferencia: el modelo de Mamdani y el de Takagi-Sugeno.

3.1.2.1. Modelo Mamdani

Es posiblemente el método más ampliamente utilizado, propuesto por Ebrahim Mamdani en 1975. El proceso se realiza en cuatro pasos [17]:

1. Fuzificación de las variables de entrada. El primer paso consiste en tomar los valores duros de las entradas y determinar el grado de pertenencia de estas entradas a los conjuntos difusos asociados.
2. Evaluación de las reglas. Tomamos las entradas anteriores y se aplican a los antecedentes de las reglas difusas. Si una regla tiene múltiples antecedentes, se utiliza el operador AND u OR para obtener un único número que represente el resultado de la evaluación. Este número (el valor de verdad) se aplica al consecuente.

Para evaluar el OR habitualmente se emplea la T-Conorma estándar (máximo), definida como $\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$, mientras que para el AND se emplea la T-norma min.

Finalmente, el resultado de la evaluación del antecedente se aplica al consecuente, aplicando un recorte o escalado según el valor de verdad del antecedente. El método más comúnmente utilizado es el recorte (clipping) que corta el consecuente con el valor de verdad del antecedente. El escalado proporciona un valor más preciso, preservando la forma original del conjunto difuso. Se obtiene multiplicando todos los valores por el valor de verdad del antecedente

3. Agregación de las salidas de las reglas. La agregación es el proceso de unificación de las salidas de todas las reglas; es decir, se combinan las funciones de pertenencia de todos los consecuentes previamente recortados o escalados, combinando para obtener un único conjunto difuso por cada variable de salida.

4. Defuzificación. El resultado final habitualmente es necesario expresarlo mediante un valor duro. En esta etapa se toma como entrada el conjunto difuso anteriormente obtenido para dar un valor de salida. Existen varios métodos de defuzificación, pero probablemente el más ampliamente usado es el centroide mostrado en la Ec. 6; que calcula el punto donde una línea vertical divide el conjunto en dos áreas con igual masa, siendo $\mu_A(x)$ la función de pertenencia de x en el conjunto A y x el valor a evaluar.

$$Centroide = \frac{\sum_{x=a}^b \mu_A(x)x}{\sum_{x=a}^b \mu_A(x)} \quad (6)$$

3.1.2.2. Modelo Takagi-Sugeno.

Como hemos visto, el modelo de inferencia de Mamdani requiere algún tipo de método para la defuzificación. En general, este método no es muy eficiente desde el punto de vista computacional, por lo que una opción para mejorar el tiempo de inferencia es emplear una función matemática en el consecuente, como se muestra en Ec. 7, siendo este el modelo Takagi-Sugeno [17].

$$if \ x \text{ is } A \text{ and } y \text{ is } B \text{ then } z = f(x, y) \quad (7)$$

Este modelo fue propuesto por Takagi, Sugeno y Kang [18,19], donde A y B son conjuntos difusos en el antecedente, donde $z = f(x, y)$ es una función dura en el consecuente. Usualmente $f(x, y)$ es un polinomio con respecto a las variables de entrada x y y , pero puede ser cualquier función tan grande como pueda describir apropiadamente la salida del modelo adentro de la región difusa especificada por el antecedente de la regla [18].

Este tipo de método proporciona mayor eficiencia, pero no presentan un marco tan natural para la representación del conocimiento humano. Un tipo habitual de representación del consecuente es un singleton (punta discreta), que toma valor uno en un valor puntual del universo del discurso y cero en cualquier otro punto [17].

3.2. Redes Neuronales.

Las actividades de investigación desarrolladas en torno al estudio de redes neuronales artificiales, simplemente rede neuronales o neuroredes, están motivadas en modelar la forma de procesamiento de la información en sistemas nerviosos biológicos. Especialmente, por la forma de funcionamiento del cerebro humano, que es completamente distinta al funcionamiento de un computador digital convencional. El cerebro humano corresponde al de un sistema altamente complejo, no-lineal y paralelo. En términos sencillos lo anterior equivale a decir que puede realizar muchas operaciones simultáneamente a diferencia de los computadores comunes que son de tipo secuencial [21].

Las Redes Neuronales Artificiales, ANN (Artificial Neural Networks) están inspiradas en las redes neuronales biológicas del cerebro humano. Están constituidas por elementos que se comportan de forma similar a la neurona biológica en sus funciones más comunes. Estos elementos están organizados de una forma parecida a la que presenta el cerebro humano. Las ANN al margen de "parecerse" al cerebro presentan una serie de características propias del cerebro. Por ejemplo las ANN aprenden de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos [22].

En las Redes Neuronales Artificiales, ANN, la unidad análoga a la neurona biológica es el elemento procesador, PE (process element). Un elemento procesador tiene varias entradas y las combina, normalmente con una suma básica. La suma de las entradas es modificada por una función de transferencia y el valor de la salida de esta función de transferencia se pasa directamente a la salida del elemento procesador. La salida del PE se puede conectar a las entradas de otras neuronas artificiales (PE) mediante conexiones ponderadas correspondientes a la eficacia de la sinapsis de las conexiones neuronales.

Los elementos básicos de una red neuronal normalmente son [23] [24]:

Las entradas, siendo estas capas las que reciben la información desde el exterior.

Los pesos que son coeficientes que nos proporciona la importancia de la entrada dentro de la función de agregación de la neurona. Estos pesos realizan la misma función que realizan las fuerzas sinápticas de las neuronas biológicas. En ambos casos, algunas entradas son más importantes que otras de manera que tienen mayor efecto sobre el procesamiento de la neurona al combinarse para producir la respuesta neuronal. Estos pesos pueden adaptarse dentro de la red determinando la intensidad de la señal de entrada registrada por la neurona artificial, siendo así la medida de la fuerza de una conexión de entrada. Estas fuerzas pueden ser modificadas en respuesta de los ejemplos de entrenamiento de acuerdo a la topología específica o debido a las reglas de entrenamiento.

SALIDAS: Cada elemento de procesamiento tiene permitido una única salida $y_i(t)$ que puede estar asociada con un número elevado de otras neuronas. Normalmente, la salida es directamente equivalente al valor resultante de la función de activación. En la Ec. 8 se muestra esta función de salida en relación con la función de activación.

$$y_i(t) = F_i(a_i(t)) = a_i(t) \quad (8)$$

3.2.1. Propagación hacia atrás.

El algoritmo de propagación hacia atrás busca por el mínimo de la función del error de los pesos usando el método de gradiente descendente. La combinación de pesos que minimizan la función de error es considerada ser una solución para el problema de aprendizaje. Desde que este método requiere calcular la gradiente del error por cada iteración, se debe garantizar que la función de error sea diferenciable y continua, por lo que se debe usar una función de activación en cada neurona que respete estas dos características [25].

En la figura 3 se muestra la estructura de una red neuronal, donde la salida es la capa L-ésima y tiene m_L neuronas donde L es la profundidad de la red mientras que la entrada es la capa 0, y tiene m_0 neuronas.

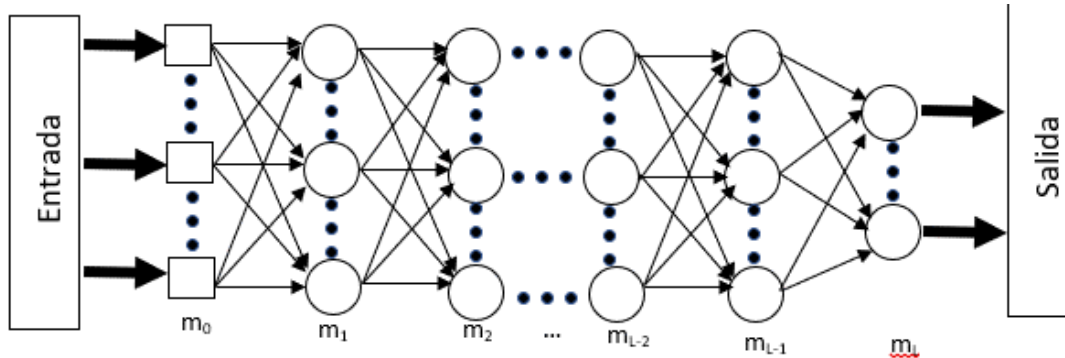


Figura 3. Red Multicapa

En el funcionamiento de la ANN, nos encontraremos dos tipos de señales: *señales de función* y *señales de error*. Las señales de función son los estímulos que entran en la capa 0, y pasa hacia adelante, capa por capa del nodo tradicional, hasta la capa L, en donde se genera la señal de salida, mientras que las señales de error son las correcciones de las sinapsis haciendo retro propagación hacia atrás [21].

Estas correcciones son el objetivo del aprendizaje de la ANN [26], ya que es necesario corregir los parámetros de la red (pesos y umbrales) con el fin de que las entradas presentadas produzcan las salidas deseadas, es decir, minimizar la función error E mostrada en Ec. 9.

$$E = \frac{1}{N} \sum_{n=1}^N e(N) \quad (9)$$

Donde N es el número de patrones o muestras y $e(n)$ es el error cometido por la red para el patrón n , dado por Ec. 10:

$$e(n) = \frac{1}{2} \sum_{i=1}^{n_c} (s_i(n) - y_i(n))^2 \quad (10)$$

Siendo $Y(n) = (y_1(n), \dots, y_{n_c}(n))$ los vectores de salida de la red y $S(n) = (s_1(n), \dots, s_{n_c}(n))$ los vectores de salidas deseadas para el patrón n , respectivamente.

Ahora supongamos que $\{(X(n), S(n)), n = 1, \dots, N\}$ el conjunto de muestras o patrones que representan el problema a resolver, donde $X(n) = (x_1(n), \dots, x_{n_1}(n))$ son los patrones de entrada a la red, $S(n) = (s_1(n), \dots, s_{n_c}(n))$ son las salidas deseadas para dichas entradas y N el número de patrones disponibles, donde el aprendizaje será de la siguiente manera:

1. Se inicializa los pesos y umbrales de la red. Generalmente esta inicialización es aleatoria y con valores alrededor del cero.
2. Se toma un patrón n del conjunto de entrenamiento, $(X(n), S(n))$, y se propaga hacia la salida de la red el vector de entrada $X(n)$ utilizando Ec. 11, siendo esta la fórmula de activación de las neuronas de la capa de entrada; Ec. 12, que es la activación de la capa oculta donde f es la función de activación de las neuronas y a_j^{c-1} son las activaciones de las neuronas de la capa $c - 1$, y Ec. 13, que es la activación de la capa de salida, donde $Y = (y_1, y_2, \dots, y_{n_c})$ es el vector de salida de la red.

$$a_i^1 = x_i \text{ para } i = 1, 2, \dots, n_1 \quad (11)$$

$$a_i^c = f(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c) \text{ para } i = 1, 2, \dots, n_c \quad (12)$$

$$y \text{ } c = 2, 3, \dots, C - 1$$

$$y_i = a_i^c = f(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c) \text{ para } i = 1, 2, \dots, n_c \quad (13)$$

De esta forma se obtiene la respuesta de la red para dicho vector de entrada, $Y(n)$

3. Se evalúa el error cuadrático cometido por la red para el patrón n utilizando Ec. 10.
4. Se aplica la regla delta generalizada para modificar los pesos y umbrales de la red. Para ello se siguen los siguientes pasos:

- a. Se calculan los valores δ para todas las neuronas de la capa de salida utilizando Ec. 14

$$\delta_i^c(n) = -(s_i(n) - y_i(n))f'(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c) \quad (14)$$

- b. Se calculan los valores δ para el resto de las neuronas de la red utilizando Ec. 15, empezando desde la última capa oculta y retropropagando dichos valores hacia la capa de entrada.

$$\delta_j^{c+1}(n) = f'(\sum_{k=1}^{n_c} w_{kj}^c a_k^c + u_j^c) \sum_{i=1}^{n_{c+1}} \delta_i^{c+2}(n) w_{ji}^c \quad (15)$$

- c. Se modifican pesos y umbrales de la red siguiendo Ec. 16 y Ec. 17 para los pesos y umbrales de la capa de salida y Ec. 18 y Ec.19 para el resto de los parámetros de la red.

$$w_{ji}^{c-1}(n) = w_{ji}^{c-1}(n-1) + \alpha \delta_i^c(n) a_j^{c-1}(n) \quad (16)$$

$$\text{para } j = 1, 2, \dots, n_{c-1} \text{ } i = 1, 2, \dots, n_c$$

$$u_i^c(n) = u_i^c(n-1) + \alpha \delta_i^c(n) \text{ para } i = 1, 2, \dots, n_c \quad (17)$$

$$w_{kj}^c(n) = w_{kj}^c(n-1) + \alpha \delta_j^{c+1}(n) a_k^c(n) \quad (18)$$

$$\text{para } k = 1, 2, \dots, n_c, j = 1, 2, \dots, n_{c+1} \text{ } c = 1, 2, \dots, C-2$$

$$u_j^{c+1}(n) = u_j^{c+1}(n-1) + \alpha \delta_j^{c+1}(n) \quad (19)$$

$$\text{para } j = 1, 2, \dots, n_{c+1} \text{ } c = 1, 2, \dots, C-2$$

5. Se repiten los pasos 2, 3 y 4 para todos los patrones de entrenamiento, completando así una iteración o ciclo de aprendizaje.
6. Se evalúa el error total E mostrado en Ec. 9 cometido por la red. Dicho error también recibe el nombre de error de entrenamiento, pues se calcula utilizando los patrones de entrenamiento.

7. Se repiten los pasos 2, 3, 4, 5 y 6 hasta alcanzar un mínimo del error de entrenamiento, para lo cual se realizan m ciclos de aprendizaje.

3.3. ANFIS

Los ANFIS son un tipo de sistemas que integra a las redes neuronales artificiales con la lógica difusa, heredando las características de ambos. Debido a las ventajas que poseen este tipo de sistemas se han implementado en entornos muy variados, tales como pronóstico de desempeño de plantas hidráulicas, estimación de la resistencia a la compresión del concreto, supresión de ruido impulsivo en señales, entre otros [27].

En concreto [22], la colaboración entre las Redes Neuronales Artificiales y la Lógica Difusa puede ser utilizado para mejorar o resolver algunas de las limitaciones que presentan cada una de ellas; estos nuevos sistemas híbridos, llamados sistemas neuro-difusos, desarrollan las propiedades y ventajas propias de cada tecnología en beneficio de la otra tecnología complementaria, obteniendo una mejora importante en el comportamiento global del sistema.

En los sistemas de control basados en lógica difusa no es imprescindible partir de un modelo matemático del mismo, sino que como se ha indicado anteriormente un conocimiento básico de cómo funciona el sistema puede ser suficiente. No obstante, la tarea de explicitar este conocimiento mediante un conjunto de reglas tipo IF-THEN no siempre resulta sencilla, ya que la definición precisa de las funciones de pertenencia y de las relaciones entre las variables del sistema conllevan cierta dificultad.

Las Redes Neuronales Artificiales por su propia naturaleza y comportamiento son incapaces de expresar de forma explícita el conocimiento adquirido; pero por el contrario presentan la capacidad de aprender y relacionar las variables del sistema a partir de datos obtenidos en experiencias anteriores.

La cooperación de ambas tecnologías se realiza en los siguientes pasos:

1. A partir del conocimiento que tienen los expertos del sistema a controlar se infieren las funciones de pertenencia y las reglas borrosas que definen el modelo borroso del sistema objeto de estudio.
2. Se establecen las conexiones y el valor inicial de los pesos de la red neuronal de acuerdo con el modelo borroso.
3. Se aplica dicha Red Neuronal al sistema objeto de estudio.
4. La red neuronal es entrenada con los datos obtenidos para mejorar su precisión.
5. Después del entrenamiento, las conexiones y los pesos de la red neuronal son interpretados como funciones de pertenencia y reglas difusas. De esta manera queda explicitado el conocimiento adquirido por la red neuronal y el modelo borroso del sistema representa con mayor precisión al sistema real.

Dado a que entre más datos de entrada sean procesados en la ANFIS, más reglas de inferencia se crearan, haciendo que el tiempo de entrenamiento sea del orden $O(nm)$, por lo cual se utilizara un método extracción de características, buscando con esto reducir las características de los objetos del conjunto de entrenamiento conservando las más relevantes, reduciendo así el entrenamiento además de hacerlo más eficiente.

3.4. Extracción de Características.

La base de conocimiento de este sistema difuso es obtenida por medio del set de entrenamiento, generado al momento de clasificar los diferentes audios, luego de extraer las características más significativas de cada comando. Los

métodos que se utilizaron para la extracción de características fueron MFCC, PCA, Exponentes de Lyapunov.

3.4.1. Coeficientes Ceptrales de Mel (MFCCs)

Los coeficientes MFCC son un conjunto de parámetros des correlacionados los cuales son calculados a través de una compresión logarítmica de las energías de salida filtradas, esto derivado a través de un banco de filtros triangulares con separación perceptual [28].

Los MFCCs son una característica ampliamente usada en el reconocimiento automático del discurso o el locutor y fueron introducidos por Davis y Mermelstein en los años 80 y han sido el estado del arte desde entonces.

La frecuencia ceptral de Mel [29] es altamente efectiva en reconocimiento de audio y en modelar el tono subjetivo y la frecuencia contenida en señales de audi. La escala de Mel es calculada por Ec. 20

$$Mel(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right) \quad (20)$$

Donde $Mel(f)$ es la escala logarítmica de la escala normal de frecuencia. La escala de Mel tiene un intervalo constante de frecuencia de mel, y abarca el rango de frecuencia de 0 a 2050 hertz. Los MFCCs son calculados desde los coeficientes de potencia de la Transformada Rápida de Fourier (FFT) los cuales son filtrados por una banda triangular pasabajos. Los MFCCs son calculados por Ec. 21

$$C_n = \sqrt{\frac{2}{k}} \sum_{k=1}^K (\log S_k) \cos[n(k - 0.5)\pi/k], \quad n = 1, 2, \dots, N \quad (21)$$

Donde $S_k (k = 1, 2, \dots, K)$ es la salida de los bancos de filtros y N es el número total de muestras en cada unidad de 20 milisegundos de audio.

3.4.2. Análisis de Componente Principales (PCA)

PCA es una técnica que reduce la dimensionalidad de los datos en la cual se busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. Esta convierte un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables sin correlación lineal llamadas componentes principales [30].

Se comienza con una matriz ajustada de datos, X , cuál consiste en n observaciones (filas) con p variables (columnas) [31]. La ecuación básica de PCA esta, en notación matricial, dada por Ec. 22.

$$Y = W'X \quad (22)$$

Donde W es una matriz de coeficientes que es determinado por PCA, siendo sus factores el promedio ponderado de las variables originales. Estos pesos son contruidos de tal forma que la varianza Y es maximizada, donde cada uno de sus componentes y_i y y_j tienen entre ellas una covarianza igual a cero.

La matriz de pesos, W , es calculada de la matriz de varianza-covarianza S . Esta matriz es calculada usando Ec. 23.

$$s_{ij} = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{n-1} \quad (23)$$

La descomposición de valores singulares de S provee la solución al problema de PCA. Esta es definida en Ec. 24 como:

$$U'SU = L \quad (24)$$

Donde L es una matriz diagonal de eigenvalores de S y U es la matriz de eigenvectores de S . W es calculado de L y U , usando la relación de Ec. 25.

$$W = UL^{-\frac{1}{2}} \quad (25)$$

La correlación entre un factor i -esimo de W y la j -esima variable original de X se obtiene utilizando Ec. 26.

$$r_{ij} = \frac{u_{ji}\sqrt{l_i}}{s_{jj}}$$

Donde u_{ij} es un elemento de U , l_i es un elemento de la diagonal de L , y s_{jj} es un elemento de la diagonal de S .

Cuando la matriz de correlación, R , es usado en lugar de la matriz de covarianzas, S , la ecuación para Y debe ser modificada por Ec. 27.

$$Y = W'D^{-\frac{1}{2}}X \quad (27)$$

Donde D es una matriz diagonal hecha con los elementos diagonales de S .

3.4.3. Exponentes de Lyapunov.

En cambio, Exponentes de Lyapunov, nos permite medir el grado de separación de dos trayectorias infinitesimalmente cercanas, con lo que nos es posible considerar el vecindario de una trayectoria regular [32].

De acuerdo a Luis Lara et al. [33], supongamos un espacio de fases asociado a una cierta dinámica autónoma, donde cada punto de este está en correspondencia con una única trayectoria la cual está parametrizada en el tiempo t . Al cabo de un incremento δt , la nueva trayectoria pudo haberse acercado o alejado a la inicial, lo cual da una medida del comportamiento de la estabilidad local. Por simplicidad consideremos el sistema Ec. 28.

$$\frac{d}{dt}y = f(y) \quad (28)$$

La condición inicial $y_0 = y(t_0)$ define la curva γ_0 , lo que es igual a $y(t, p_0)$. En el tiempo t_0 producimos la variación δy_0 y la nueva curva solución γ la expresamos como $y = y(t, p_0) + \delta y$, donde la variación esta definida por Ec. 29

$$\frac{d}{dt} \delta y = \left(\frac{\partial f}{\partial y}\right)_{\gamma_0} \delta y \quad (29)$$

Integrando Ec. 29 se obtiene Ec. 30

$$\frac{\delta y(t)}{\delta y(t_0)} = \exp \int_{t_0}^t \frac{\partial f}{\partial y} dt'' \quad (30)$$

Y cuando los valores de t son próximos a t_0 la variación se aproxima a Ec.

31

$$\frac{\delta y(t)}{\delta y(t_0)} \equiv \exp\left(\frac{\partial f}{\partial y}\right)_{t_0} \delta t \quad (31)$$

Cuando $\left(\frac{\partial f}{\partial y}\right)_{t_0} > 0$, decimos que la solución $y(t, p_0)$ es localmente inestable en p_0 . Al contrario, si $\left(\frac{\partial f}{\partial y}\right)_{t_0} < 0$ es negativa es localmente estable. Para determinar en forma elemental el exponente de Lyapunov, consideremos nuevamente Ec. 1, por lo que tenemos:

$$\lambda = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \left| \frac{\delta y(t)}{\delta y(t_0)} \right| \quad (32)$$

Un método utilizado para evaluar computacionalmente este exponente de forma eficiente por medio de autovalores. Este método consiste en realizar cambios locales de coordenadas de manera que se pueda diagonalizar localmente las ecuaciones de las variaciones. Los autovalores de estas últimas están directamente vinculados a la estimación de los exponentes. Cuando la transformación puntual se resuelve explícitamente, en muchos casos los signos del espectro de coeficientes quedan determinados evitándose el cálculo numérico. Entonces consideremos el sistema

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, \dots, x_m), \\ &\dots \dots \dots \\ \dot{x}_m &= f_m(x_1, \dots, x_m) \end{aligned} \quad (33)$$

Donde las funciones f_k satisfacen las condiciones de existencia de solución del sistema. Las correspondientes variaciones de x_k , $k: 1, \dots, m$ son determinadas por Ec. 34

$$\delta \dot{x}_k = \sum_{j=1}^m \frac{\partial f_k}{\partial x_j} \delta x_j, k = 1, \dots, m \quad (34)$$

Que en Ec. 35 se representa su notación vectorial

$$\delta \dot{x} = J \delta x \quad (35)$$

Donde J es la matriz Jacobiana del sistema Ec. 33. Dado que rara vez se obtienen una solución explícita, es necesario recurrir a métodos de integración numérica para determinar Ec. 33 y Ec. 35. Considerando una variable t, la cual es discretizada de forma $t_{i+1} = t_i + \varepsilon, i = 0, \dots, n$; siendo n el número de los i-tervalos $I_i : (t_i, t_{i+1})$. Con el fin de poder determinar en forma explícita las fluctuaciones, en cada intervalo I_i , se lineariza Ec. 35 aproximando a $J(t)$ con $J_i = J(t_i)$, por lo que quedaría aproximada como

$$\delta \dot{x} = J_i \delta x \quad (36)$$

Para determinar cada i variación, se diagonaliza la matriz J_i por medio de las matrices P_i y P_i^{-1} , construidas a partir de los autovectores columna de J_i , por lo que se obtiene la matriz diagonalizada

$$D_i = P_i^{-1} J_i P_i \equiv \delta \dot{y} = D_i \delta y \quad (37)$$

Por ser D_i las componentes de la variación δy_k se determinan mediante

$$\delta \dot{y}_k = \lambda_{ik} \delta y_k \quad k: 1, \dots, m \quad (36)$$

Donde $\lambda_{ki}, k = 1, \dots, m$ son los autovalores de J_i . Debido a la diagonalización de la matriz Jacobiana las variaciones $\delta y_k k = 1, \dots, m$ corresponden a las m direcciones de los autovectores en el espacio y_1, \dots, y_m .

Integrando la Ec. 36 con valores iniciales en t_i , determinados la solución en t_{i+1} , por tanto

$$\frac{\delta y_{ki+1}}{\delta y_{ki}} = \exp \lambda_{ik} \varepsilon, \quad k: 1, \dots, m \quad (37)$$

Donde los autovalores λ_{ik} dan una medida de estabilidad local en el i -ésimo intervalo y la k -ésima dirección. Cuando la dinámica es lineal, el cálculo se simplifica ya que la matriz Jacobiana es la matriz del sistema.

Entonces, conociendo el espectro de autovalores de la matriz J_i , se puede estudiar cualitativamente las propiedades de estabilidad a lo largo de a curva de solución. Definimos las siguientes medias aritméticas

$$\Lambda_k^n = \frac{1}{n} \sum_{i=1}^n Re(\lambda_{ik}) \quad k: 1, \dots, m \quad (38)$$

Pero dado que la estabilidad de la solución asintótica es lo que nos interesa, definimos el espectro Λ_k como

$$\Lambda_k = \lim_{n \rightarrow \infty} \Lambda_k^n \quad k: 1, \dots, m \quad (39)$$

Los cuales, en general son función del punto en el espacio x_1, \dots, x_m dado que las transformaciones locales se hacen sobre una determinada trayectoria de Ec. 33.

Ec. 38 y Ec. 39 determinan los nuevos coeficientes Λ_k , donde los índices k dan un promedio del comportamiento de la inestabilidad local a lo largo de la curva solución elegida, más específicamente sobre la solución asintótica.

Cuando n tiende a infinito y ε tiende a cero, Ec. 38 y Ec. 39 se reescribe como Ec. 40 y Ec. 41 respectivamente.

$$\Lambda_k(t) = \frac{1}{t} \int_0^t Re(\lambda_k(t')) dt' \quad (40)$$

$$\Lambda_k = \lim_{n \rightarrow \infty} \Lambda_k(t) \quad k: 1, \dots, m \quad (41)$$

Las definiciones dadas por Ec. 39, 39, 40 y 41 son una alternativa a los exponentes de Lyapunov y esencialmente dan una medida de la estabilidad a lo largo de una trayectoria. Si todos los coeficientes Λ_k son negativos, entonces la órbita es asintóticamente estable, de lo contrario, la órbita es inestable.

3.5. Entrenamiento de la Red ANFIS con la extracción de características.

Por lo que la metodología a seguir será la extracción de características utilizando y comparando los métodos anteriores, además de mezclarlos entre ellos. Como los valores que se generan, aún con la extracción de características sigue siendo demasiado grande para que la ANFIS los procese todos, por lo tanto, se escogerán tres valores, que sería el valor máximo, la media y el valor mínimo, dada su importancia a la hora de representar datos estadísticos. Esta información se utilizará para entrenar las diferentes ANFIS (se usará una por cada servomotor), siendo su salida los grados en que cada servomotor debe girar para adoptar la postura que se busca.

En la figura 4 se muestra el diseño de la ANFIS, donde se puede observar como el mínimo, la media y la máxima es puestas como entrada de los valores que van a ser clasificados, para luego ser evaluados por medio de los 7 conjuntos difusos puestos para cada uno de sus respectivos comandos, los cuales serán entrenados en base a el conjunto de entrenamiento obtenido por medio del preprocesamiento, dándonos así una salida que es la cual nos dirá cuál es la clase a la que pertenece.

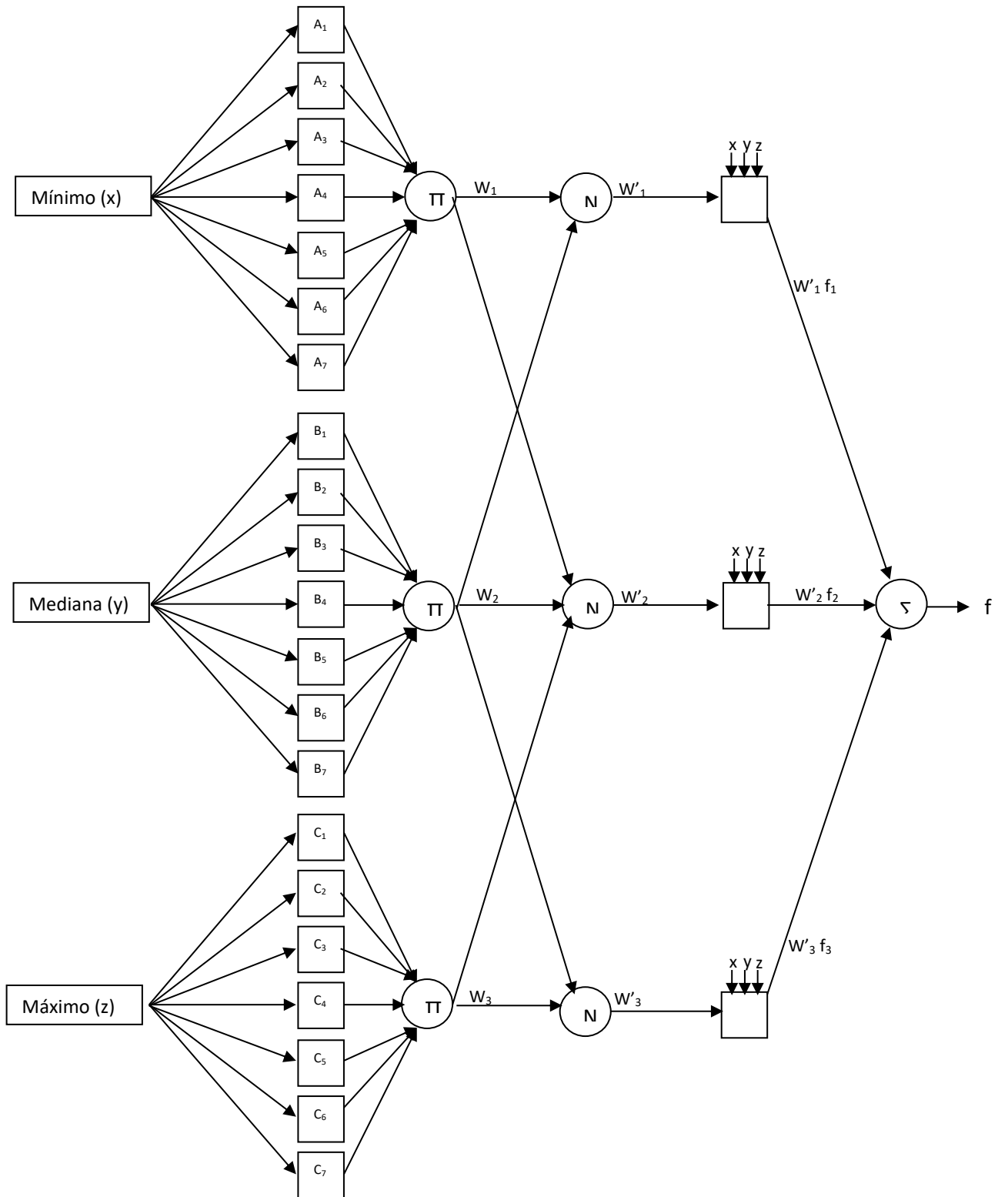


Figura 4. ANFIS utilizada para el reconocimiento de los comandos de voz.

4. ETAPA DE MODELADO Y CONTROL.

4.1. Diseño de la Mano

En la figura 5 se muestra el agarre de una esfera y botella (cilindro) siendo vistas de forma superior y transversal, siendo estas las formas más comunes que implican el agarre de mano [4]. Esta será la base para poder diseñar la mano artificial que se utilizará como prótesis, esto con la idea de poder imitar este tipo de movimientos de forma exacta.

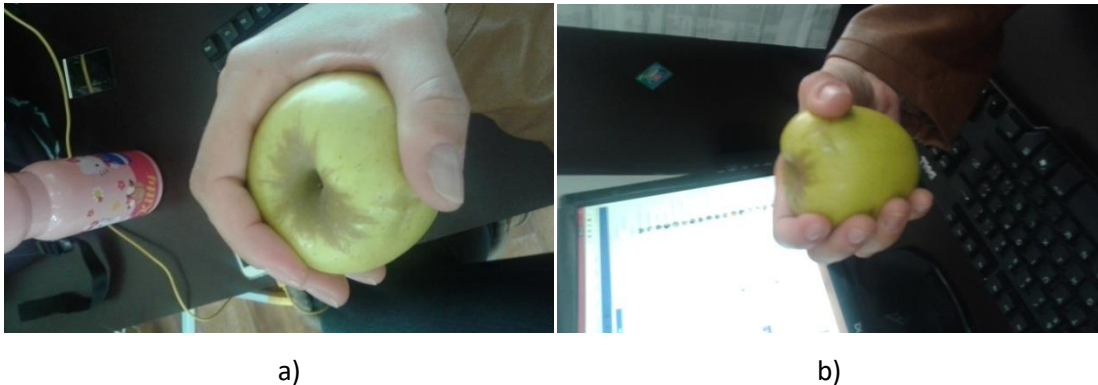


Figura 5. Manzana tomada con una mano mostrada en vista a) superior b) transversal.

La mano constará de un diseño hecho en un diseño 3D como se muestra en la figura 6. Como se puede notar, se sustituirá el meñique por un segundo pulgar oponible, permitiéndonos tener un mayor apoyo a la hora de tomar un objeto. También constará de 13 servomotores en cada articulación, siendo su soporte las piezas mostradas en color azul en la figura 6.

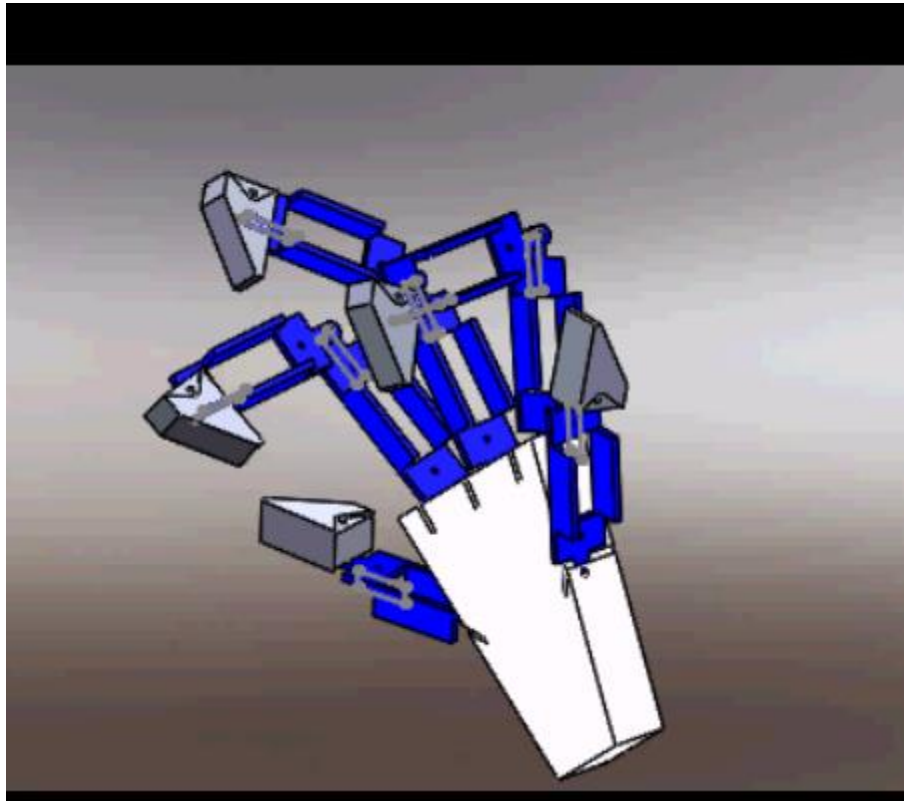
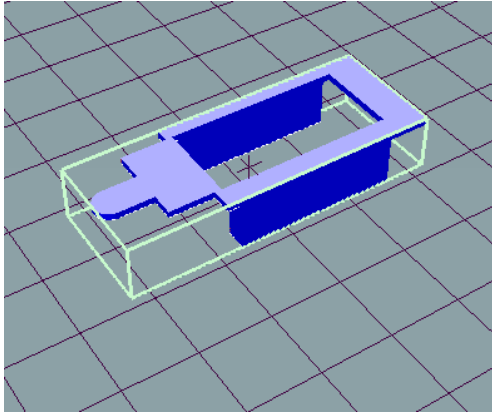
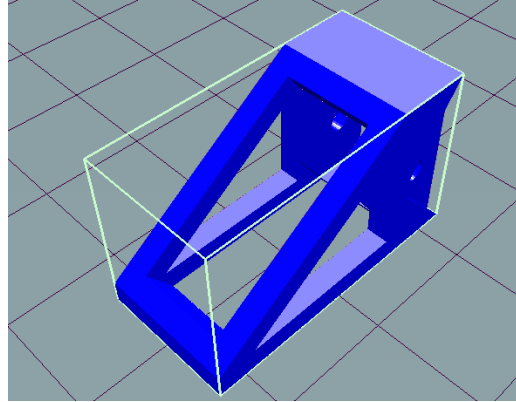


Figura 6. Modelo de la mano a utilizar en el proyecto creado en Solid Works.

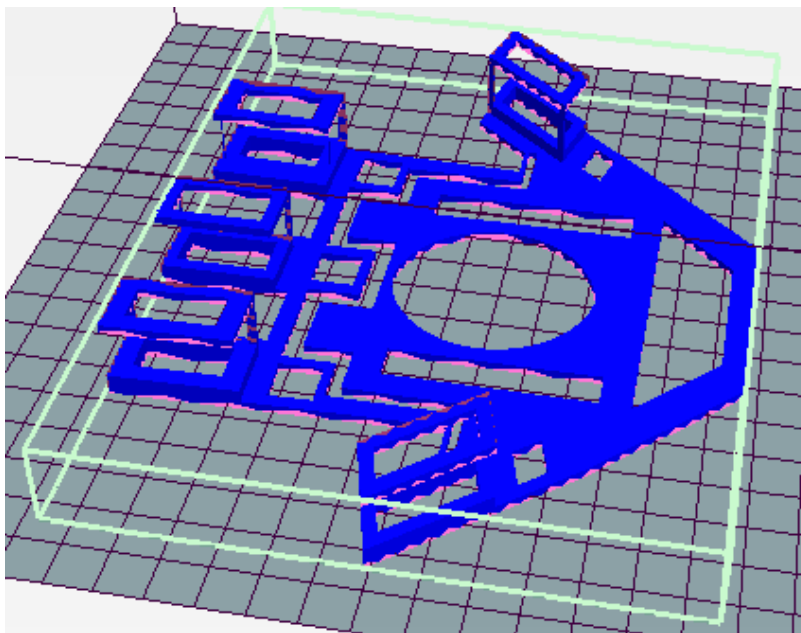
El diseño fue generado en Solid Works, un software que se utiliza para modelado mecánico en 2D y 3D. Luego de generado en este software se transporta a un archivo STL para luego abrirlo en un programa denominado Repetier-Host, el cual nos permite imprimir estos modelos con una impresora 3D. En la figura 7 se muestra en Repetier-Host el modelo de la palma, el soporte del servomotor y la punta del dedo.



a)



b)



c)

Figura 7. Piezas utilizadas para la construcción de la mano. a) Soporte para el servomotor, siendo el equivalente a las articulaciones de la mano. b) Punta del dedo c) Palma de la mano.

4.2. Etapa de Control

Luego de analizados los diferentes patrones, estos se relacionarán con los diferentes comandos de voz que se le darán al sistema. Para hacer estas pruebas, por el momento se utiliza la comunicación serial entre la computadora y la mano usando Matlab y una tarjeta Arduino UNO. Se planea en el futuro poder tener

embebido en la Arduino tanto los diferentes FIS como también una entrada para la voz.

Aunque en este caso el sistema se utilizara en una prótesis en personas que han perdido este miembro, o algún otro dependiendo de la prótesis, se busca además que pueda realizar la mayoría de sus actividades cotidianas de una forma más cómoda. El sistema también es aplicable a sistemas de manipulación de residuos tóxicos o peligrosos para una persona donde utilizando un sistema de visión y este sistema de control por voz se pueden llevar a cabo tareas que no involucren destrezas manuales para manipular el control robótico. Para obtener los grados correctos, los cuales serán las salidas del ANFIS, se medirán las posiciones de los servomotores de la mano mientras agarra múltiples objetos.

En la figura 8 se muestra un diagrama de comunicación del sistema, donde se puede observar que los datos obtenidos usando los métodos de extracción de características sobre los audios y escogiendo el valor máximo, mínimo y la media de estos, para luego enviarlos a las entradas de cada uno de las tres FIS, dado que son movimientos generalizados por las tres articulaciones presentes en cada dedo (dos en el caso del pulgar). Para hacer esto, las salidas de las ANFIS programadas en MATLAB se enviarán por puerto serial (USB) a una tarjeta arduino, la cual a su vez en cada una de sus salidas digitales estará conectado cada servomotor. Estos se moverán dependiendo de los grados de movimiento que se obtienen de las salidas de las diferentes ANFIS, las cuales a su vez se ajustarán por medio de una lectura hecha por el arduino el cual nos indicará la posición que se encuentren los servos dependiendo de cómo se debe cerrar o abrir la mano en cada caso.

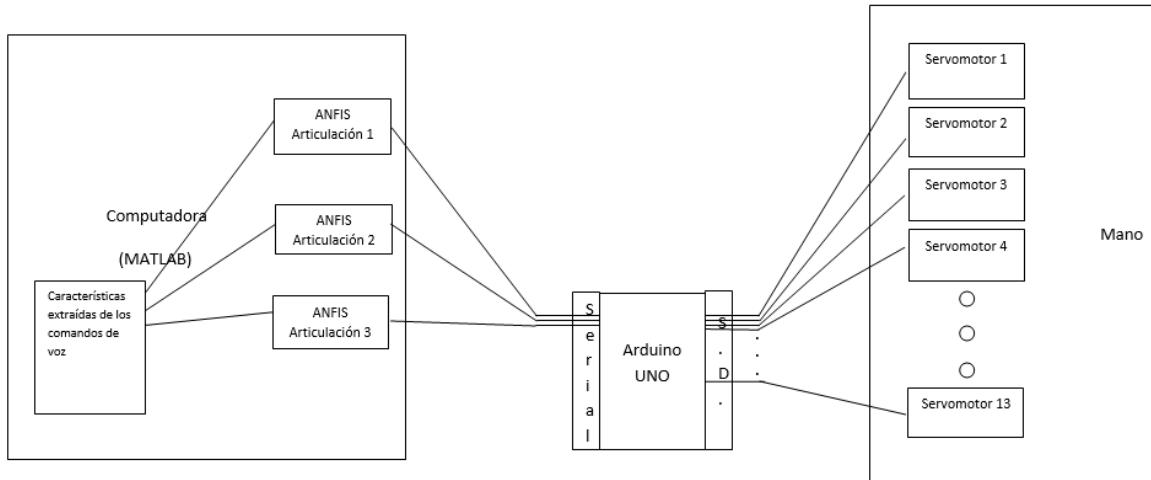


Figura 8. Diagrama de comunicación entre la computadora y la mano.

En la figura 9 se muestra el diseño final de la mano, donde se puede ver los diferentes servomotores empotrados en la estructura de la mano y a la vez conectados en la tarjeta Arduino MEGA.

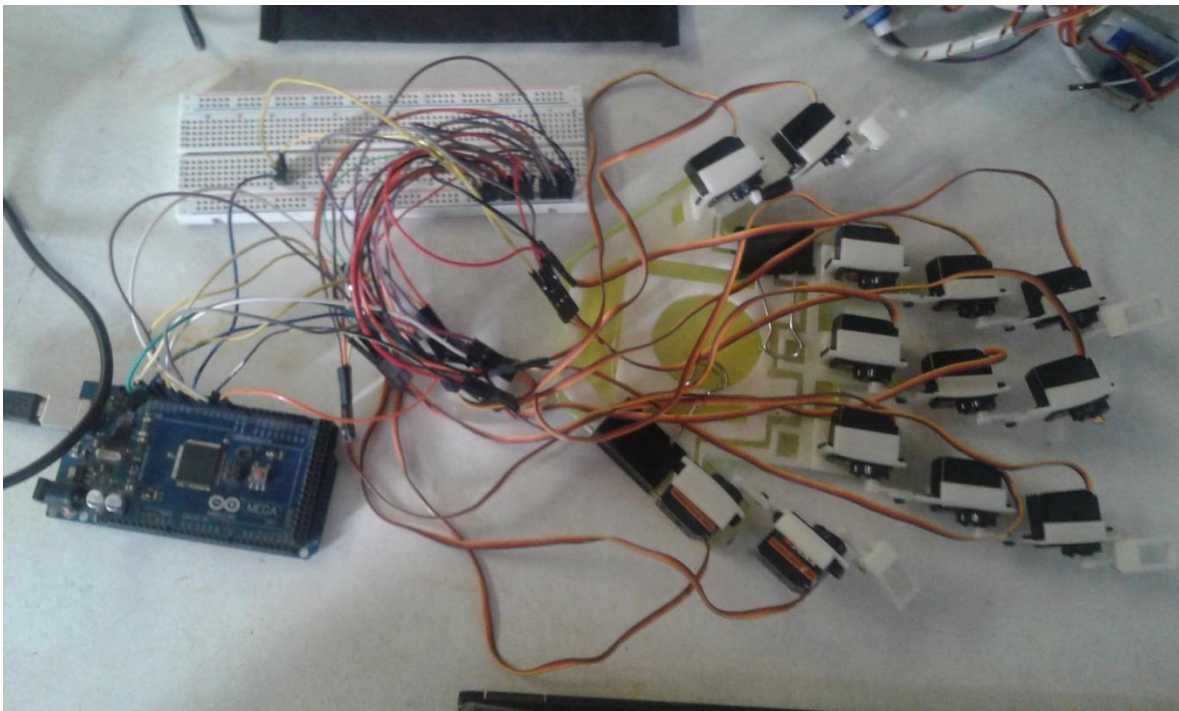


Figura 9. Diseño completo de la mano robótica, donde se pueden observar sus conexiones tanto a corriente como a los pines de la tarjeta Arduino MEGA.

5. RESULTADOS

Se probó el sistema de control usando dos sets de comandos, utilizando uno con frases completas y otra con abreviaciones. En cada una de ellas se mezcló cada uno de los comandos con diferentes tipos de ruido, resultando en 56 muestras en cada set. En la figura 10 se muestran los conjuntos difusos generados al entrenar la ANFIS utilizando un conjunto de entrenamiento al que se le aplicó PCA.

En la figura 11 se muestran los conjuntos que se generaron usando Exponentes de Lyapunov.

En la figura 12 se muestran utilizando MFCC.

También se mezclaron estos métodos para observar el desempeño de cada uno, siendo estos métodos MFCC más PCA (figura 13) y Lyapunov más PCA (figura 14). Cabe aclarar que dada la dificultad que tiene la ANFIS para entrenarse con varias entradas, se tomaron tres valores siendo el valor mínimo, la media y el valor máximo de cada uno de los sets de entrenamiento.

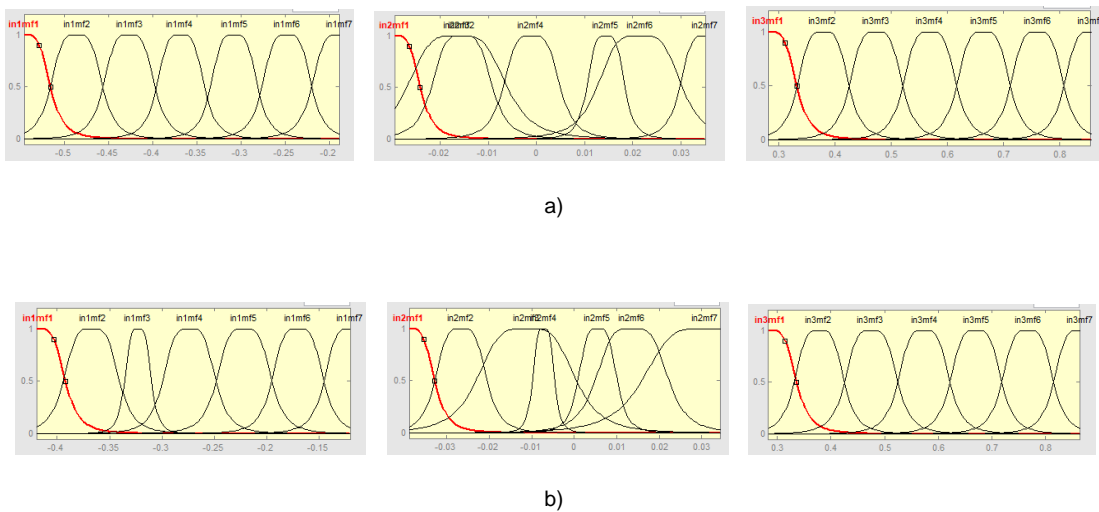
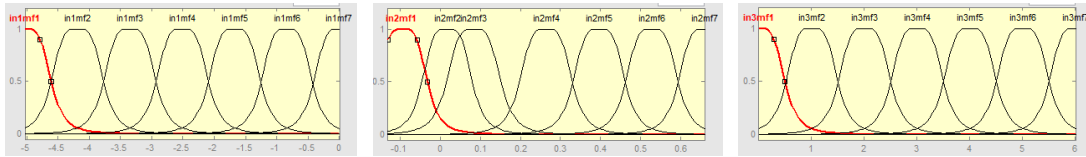
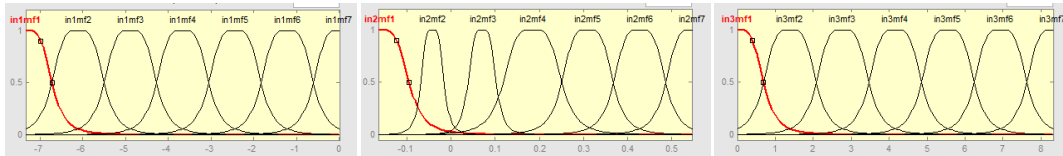


Figura 10. Conjuntos Difusos Generados al entrenar la ANFIS con PCA utilizando a) frases completas y b) abreviaciones.

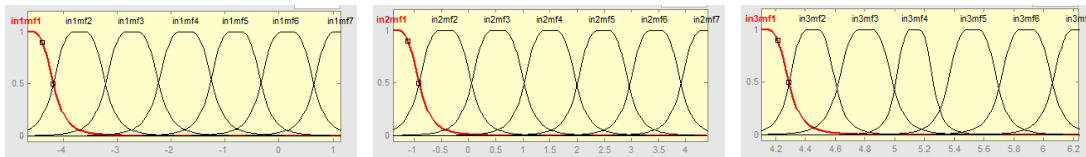


a)

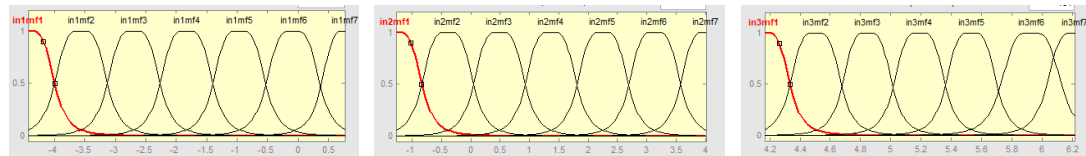


b)

Figura 11. Conjuntos Difusos Generados al entrenar la ANFIS con Exponentes de Lyapunov utilizando a) frases completas y b) abreviaciones.

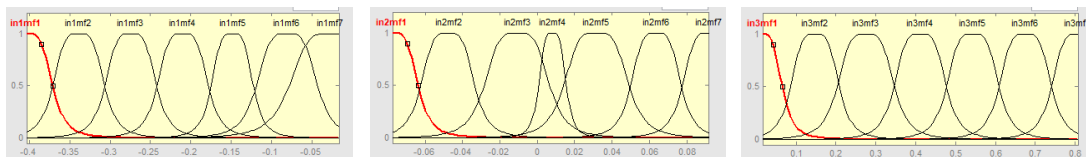


a)

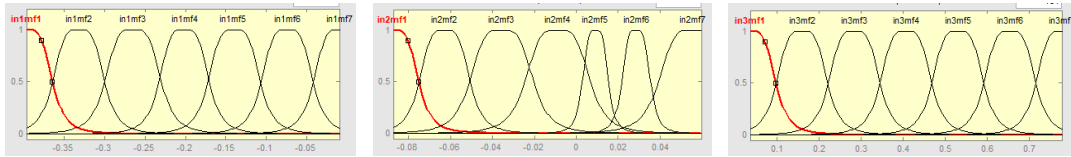


b)

Figura 12. Conjuntos Difusos Generados al entrenar la ANFIS con MFCC utilizando a) frases completas y b) abreviaciones.

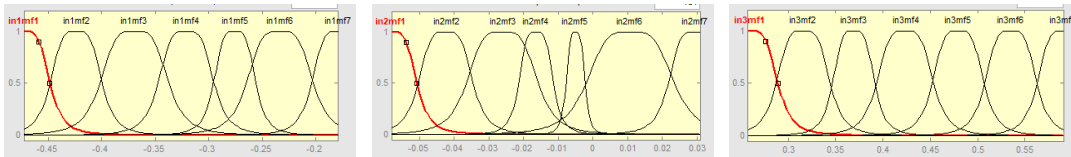


a)

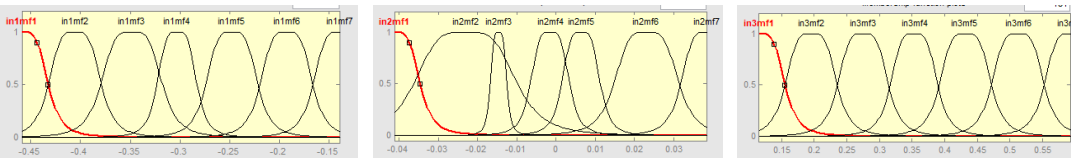


b)

Figura 13. Conjuntos Difusos Generados al entrenar la ANFIS con Lyapunov más PCA utilizando a) frases completas y b) abreviaciones.



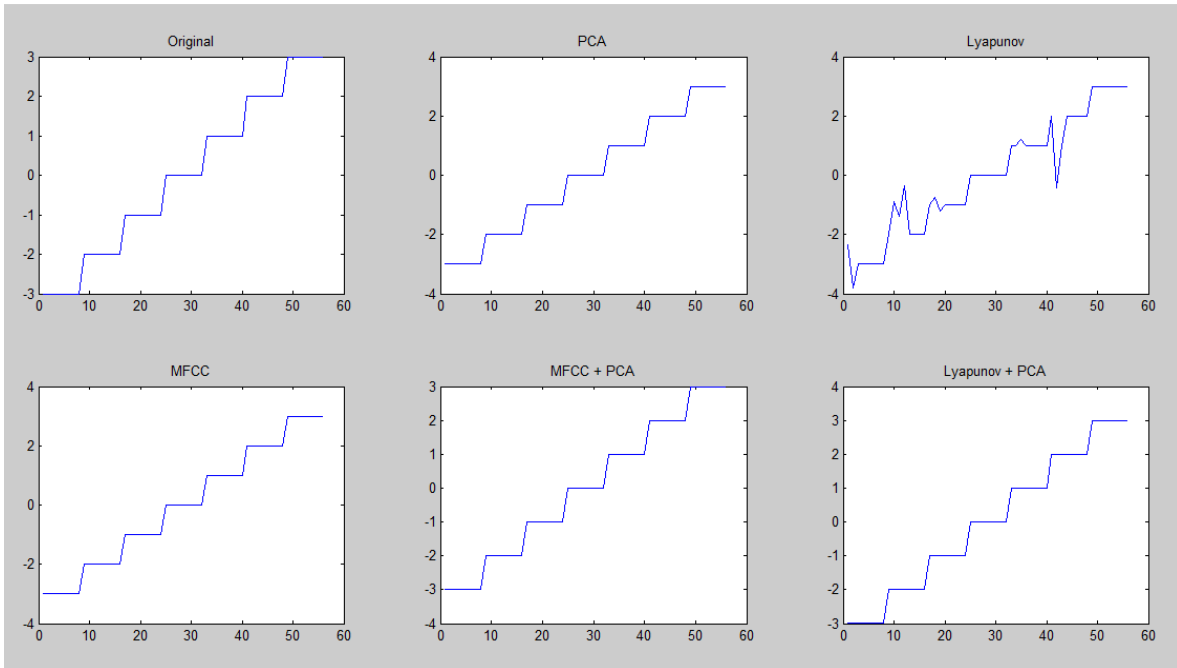
a)



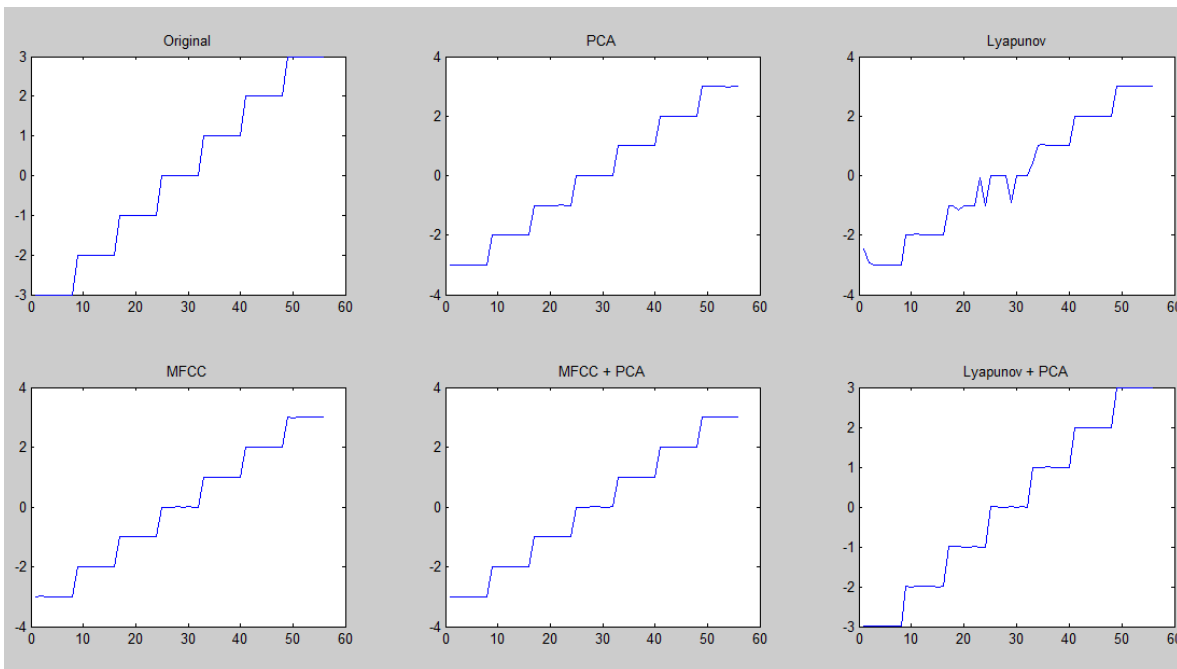
b)

Figura 14. Conjuntos Difusos Generados al entrenar la ANFIS con MFCC más PCA utilizando a) frases completas y b) abreviaciones.

En la figura 15 se muestran las clases de datos del conjunto original y las clases generadas por cada una de las ANFIS utilizando los métodos previamente mencionados para la extracción de características, observándose que Exponentes de Lyapunov tiene un mal desempeño en comparación del resto de los métodos. No obstante mejora considerablemente cuando se mezcla con PCA. En la figura 16 se puede observar esto de forma más clara, ya que se muestra el grado de error de cada uno de los métodos, notándose que mientras MFCC produce un error pequeño en comparación de los demás métodos en a), en b) se observa que hay un menor error en el método de Lyapunov más PCA.

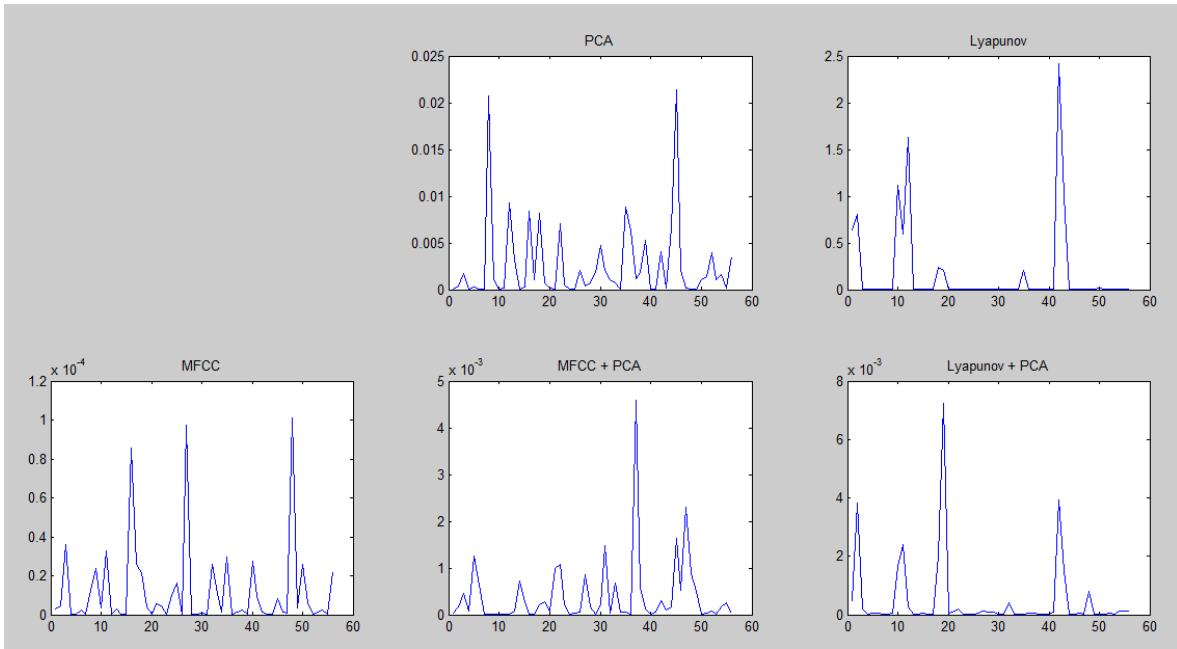


a)

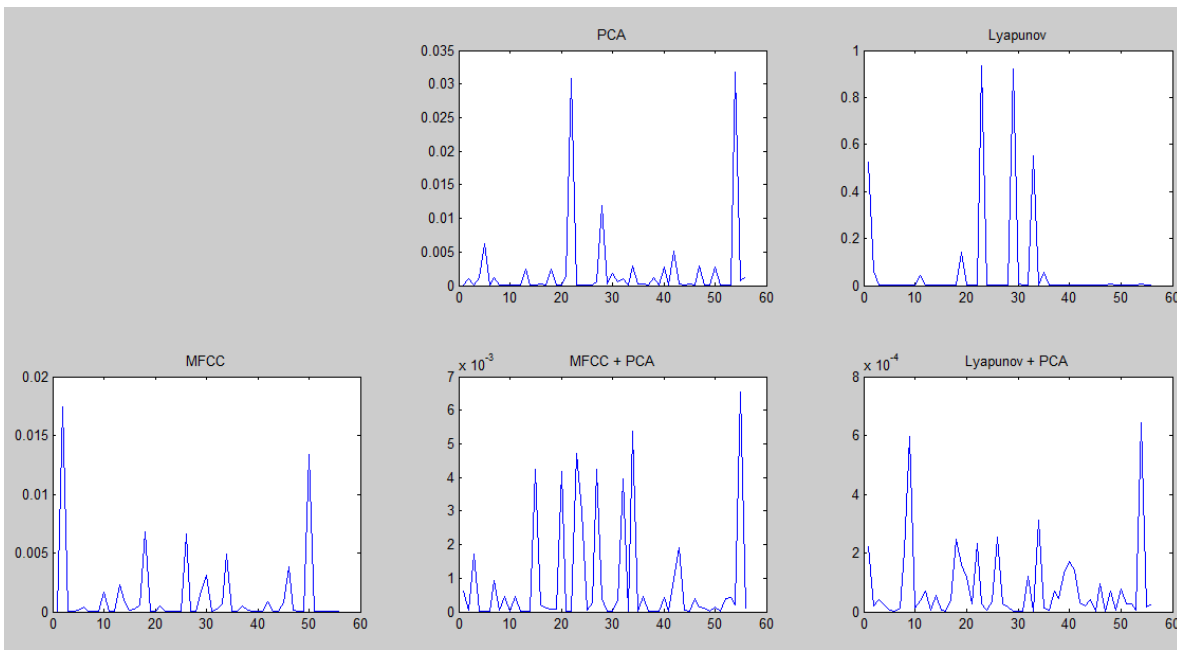


b)

Figura 15. Resultados de cada uno de los métodos de extracción de características aplicado en cada ANFIS utilizando a) frases completas y b) abreviaciones



a)

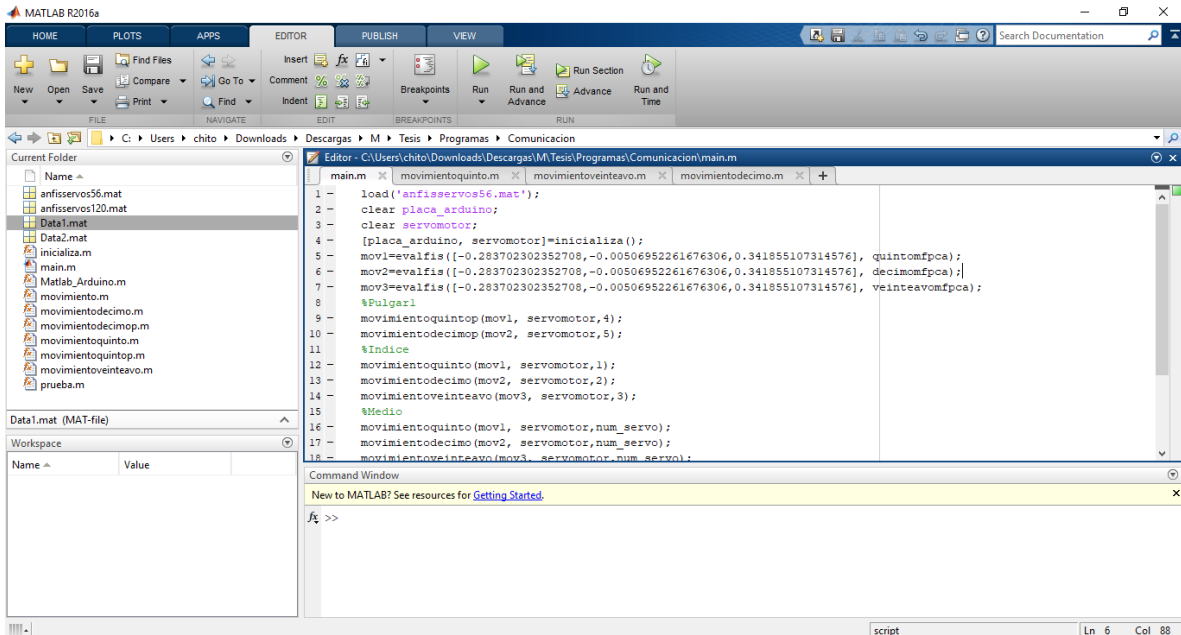


b)

Figura 16. Error obtenido en cada uno de los métodos aplicados al conjunto de entrenamiento para la ANFIS utilizando a) frases completas y b) abreviaciones.

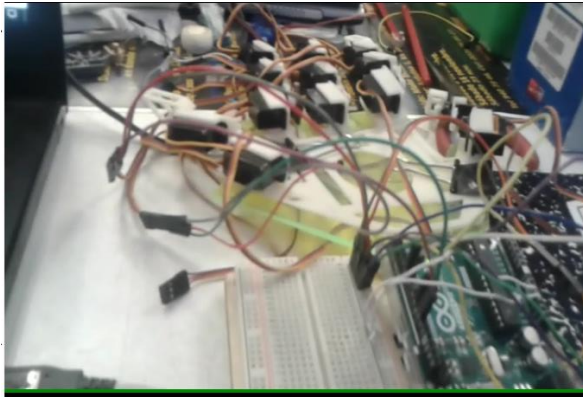
De acuerdo a los resultados mostrados previamente se escogerá el método de MFCC más PCA, ya que es el único que presenta el mismo error al utilizar tanto abreviaciones como palabras completas.

En la figura 17 se muestra el programa principal utilizado para la ejecución de los movimientos de la mano, en donde se utilizan las cadena donde se guardan las características extraídas de los comandos de voz para hacer el movimiento correspondiente de agarre. En la imagen 18 se muestra el abrir y cerrar de la mano aplicando el comando mucho menos, donde se puede observar como se cierran los dedos sin forzar los servomotores, permitiendo poder hacer el agarre sin dañar la mano.

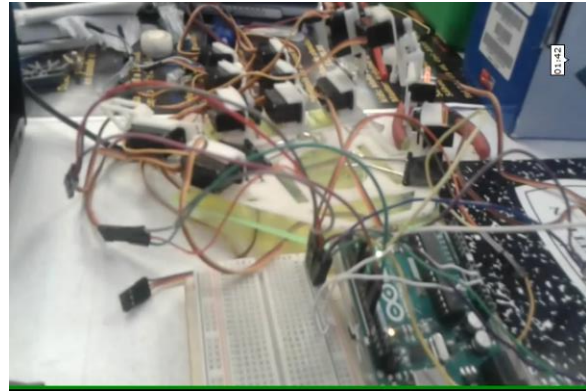


```
main.m
1 - load('anfiservos56.mat');
2 - clear placa_arduino;
3 - clear servomotor;
4 - [placa_arduino, servomotor]=inicializa();
5 - mov1=evalfis([-0.283702302352708,-0.00506952261676306,0.341855107314576], quintomfcca);
6 - mov2=evalfis([-0.283702302352708,-0.00506952261676306,0.341855107314576], decimomfcca);
7 - mov3=evalfis([-0.283702302352708,-0.00506952261676306,0.341855107314576], veinteavomfcca);
8
9 - movimientoquintop(mov1, servomotor,4);
10 - movimientodecimop(mov2, servomotor,5);
11 - %Indice
12 - movimientoquinto(mov1, servomotor,1);
13 - movimientodecimo(mov2, servomotor,2);
14 - movimientoveinteavo(mov3, servomotor,3);
15 - %Medio
16 - movimientoquinto(mov1, servomotor,num_servo);
17 - movimientodecimo(mov2, servomotor,num_servo);
18 - movimientoveinteavo(mov3, servomotor,num_servo);
```

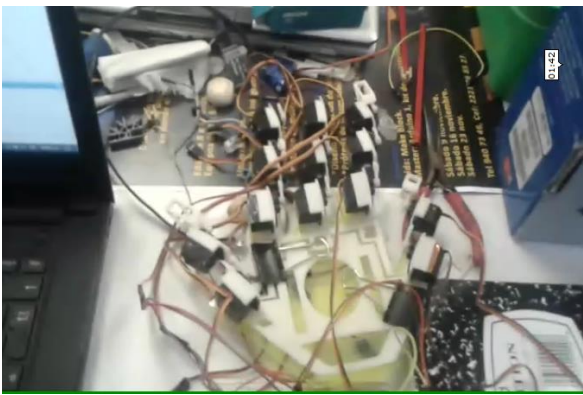
Figura 17. Programa principal para el control de la mano, el cuál esta codificado en MATLAB.



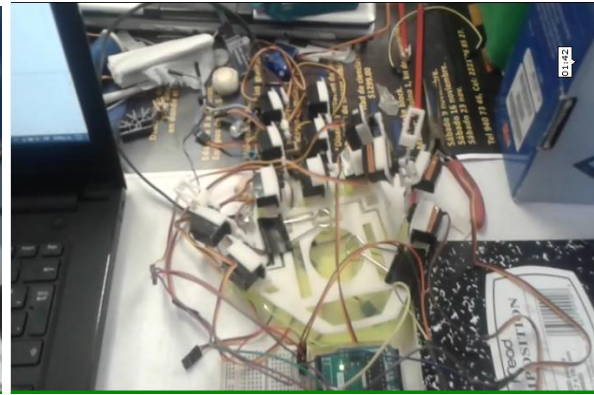
a)



b)



c)



d)

Figura 18. a) Posición neutral mano vista transversal b) Posición de agarre vista transversal c) Posición neutral vista superior d) Posición de agarre vista superior.

6. CONCLUSIONES

En este documento se puede observar la viabilidad de usar comandos de voz para poder hacer el movimiento de una mano. Se compararon diferentes métodos de extracción de características, recomendándose usar MFCC más PCA por su estabilidad, ya que en ambos casos mantiene el mismo grado de error. También se pudo observar el movimiento de la mano, reaccionando por cada comando que se ejecutaba. En efecto, en este proyecto se logró implementar algoritmos de comandos de voz para controlar los movimientos de una mano robótica, basado en el entrenamiento de un ANFIS implementado en Matlab.

Para trabajo a futuro es necesario grabar y clasificar más voces para el entrenamiento de la ANFIS, permitiéndonos manejar con mayor exactitud el control de la mano, además de modificar el diseño de la mano a un diseño más resistente y más ergonómico. Además de todo esto, es necesario usar un sistema embebido, ya que el uso de Matlab fue solo para métodos de prueba para comprobar la viabilidad del sistema, por lo que se recomienda, para comodidad de la persona que usará la mano, incluir todo en la tarjeta Arduino.

Bibliografía

- [1] Cesare Rossi, Sergio Savino, Vincenzo Niola and Stefano Troncone. A study of a robotic hand with tendon driven fingers. *Robotica* (2015) volume 33, pp. 1034–1048.
- [2] Francesca Cordella, Loredana Zollo, Antonino Salerno, Dino Accoto, Eugenio Guglielmelli and Bruno Siciliano. Human Hand Motion Analysis and Synthesis of Optimal Power Grasps for a Robotic Hand. *International Journal of Advanced Robotic Systems*. 2014
- [3] Georg Stillfried, Ulrich Hillenbrand, Marcus Settles and Patrick van der Smagt. MRI-Based Skeletal Hand Movement Model. *The Human Hand as an Inspiration for Robot Hand Development*, pp. 49-75. 2014.
- [4] AhmedM. Almassri, W. Z. Wan Hasan, S. A. Ahmad, A. J. Ishak, A. M. Ghazali, D. N. Talib, and Chikamune Wada. Review Article. Pressure Sensor: State of the Art, Design, and Application for Robotic Hand. *Hindawi Publishing Corporation Journal of Sensors*. Vol. 2015. 2014
- [5] Guido Herrmann, Jamaludin Jalani, et al. Robotic hand posture and compliant grasping control using operational space and integral sliding mode control. *Robotica* volume 34, pp. 2163–2185. 2016.
- [6] Pablo J. Prieto, Nohe R. Cazarez-Castro, Danelis García, Selene L. Cardenas-Maciel. Estabilidad para un control borroso en modo deslizante aplicado a un robot paralelo neumático. *Revista Iberoamericana de Automática e Informática industrial* 12, pp. 488–496. 2015.
- [7] Abdeldjalil Naceri, Alessandro Moscatelli, Marco Santello, and Marc O. Ernst. Multi-digit Position and Force Coordination in Three- and Four-Digit Grasping. *Haptics: Neuroscience, Devices, Modeling, and Applications* pp. 101-108. 2014.
- [8] Daniele Leonardis, Michele Barsotti, et al. An EMG-Controlled Robotic Hand Exoskeleton for Bilateral Rehabilitation. *IEEE TRANSACTIONS ON HAPTICS*, VOL. 8, NO. 2, 2015
- [9] Wu, Faye Y. and Harry Asada. "Bio-Artificial Synergies for Grasp Posture Control of Supernumerary Robotic Fingers." in *Robotics: Science and Systems X*, July 12-16, 2014.
- [10] John H.L. Hansen and Taufiq Hasan. Speaker Recognition by Machines and Humans. A tutorial Review. *IEEE SIGNAL PROCESSING MAGAZINE*. Vol. 74. 2015
- [11] Dong-Il Kim and Byung-Cheol Kim. Speech Recognition using Hidden Markov Models in Embedded Platform. *Indian Journal of Science and Technology*, Vol 8(34), 2015.
- [12] Mohamed anouar Ben messaoud, Aïcha Bouzid, Nouredine Ellouze. Speech enhancement based on wavelet packet of an improved principal component analysis. *Computer Speech and Language* Vol. 35 pp. 58–72, 2016.

- [13] Yajie Miao, Mohammad Gowayed, Florian Metze. EESEN: End-to-End Speech Recognition Using Deep RNN Models and WFST-Based Decoding. Automatic Speech Recognition and Understanding (ASRU). 2015
- [14] Hitesh Gupta, Deepinder Singh Wadhwa. Speech Feature Extraction and Recognition Using Genetic Algorithm. International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 1. 2014
- [15] L. A. Zadeh. Fuzzy Sets. Information and Control. 8:338-353, 1965.
- [16] Ruvalcaba Coyaso, Francisco Javier & Vermonden, Anäis (2015). Lógica difusa para la toma de decisiones y la selección de personal. Universidad & Empresa, 17(29), 239-256.
- [17] Carlos Gonzales Morcillo. Lógica Difusa. Una introducción Práctica
- [18] Jyh-Shing Roger Jang, Chuen-Tsai Sun, Eiji Mizutani. Neuro-Fuzzy and Soft Computing: A computational Approach to Learning and Machine Intelligence.
- [19] E. Mizutani, H. Takagi, and D. M. Auslander. A cooperative system based on soft computing methods to realize higher precision of computer color recipe prediction. In Proceedings of Applications and Science of Artificial Neural Networks, part of SPIE's International Symposium on OE/Aerospace Sensing and Dual Use Photonics. Pages. 303-314, April 1995.
- [20] J.A. Nelder and R. Mead. A simplex method for function minimization. The Computer Journal, 7:308-313, 1965.
- [21] Laurene Fausett. Fundamentals of Neural Networks. Prentice-Hall, New Jersey, USA, 1994.
- [22] Xabier Basogain Olabe. Redes Neuronales Artificiales y sus Aplicaciones. Escuela Superior de Ingeniería de Bilbao, EHU, pp. 5,71
- [23] Ivan Martínez Ortiz, Introducción a la Redes Neuronales, Universidad Complutense de Madrid, Facultad de Informática.
- [24] Francisco José Palacios Burgos, Herramientas en GNU/Linux para estudiantes universitarios, La Neurona Artificial, 2003.
- [25] R. Rojas. Neural Networks: A Systematic Introduction, Springer, 1996, pp. 151,152.
- [26] Pedro Isasi Viñuela, Inés M. Galván León. Redes de Neuronas Artificiales: Un Enfoque Práctico. Pearson Educación, 2004, pp. 63,64.
- [27] Jyh-Shing Roger Jang. ANFIS : Adaptive-Network-Based Fuzzy Inference System . IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. 23, NO. 3, MAY/JUNE 1993

- [28] M. A. Hossan, S. Memon and M. A. Gregory. A novel approach for MFCC feature extraction. 2010 4th International Conference on Signal Processing and Communication Systems, Gold Coast, QLD. 2010, pp. 1-5.
- [29] Min Xu et al. (2004). «HMM-based audio keyword generation». En Kiyoharu Aizawa, Yuichi Nakamura, Shin'ichi Satoh. Advances in Multimedia Information Processing - PCM 2004: 5th Pacific Rim Conference on Multimedia. Springer.
- [30] Steven M. Holland. PRINCIPAL COMPONENTS ANALYSIS (PCA). Department of Geology, University of Georgia, Athens, GA 30602-2501.
- [31] NCSS Documentation: Principal Components Analysis. NCSS, LLC. Pp. 425-1,2.
- [32] Anatole Katok. Lyapunov exponents, entropy and periodic orbits for diffeomorphisms. Publications mathématiques de l'I.H.É.S., tome 51. 1980, pp. 137-173.
- [33] Luis Lara, Cesar Stoico, Rodrigo Machado, Mario Castagnino. Estimación de los Exponentes de Lyapunov. ENIEF 2003 - XIII Congreso sobre Métodos Numéricos y sus Aplicaciones, pp. 1441- 1451.