

Extracción de Características de Señales de Audio para el Reconocimiento de Comandos de Voz

A. Tovar Arriaga^{#1}, T. A. Cortes Aguilar^{#2}

[#] Tecnológico Nacional de México: Instituto Tecnológico José Mario Molina Pasquel y Henríquez, Unidad Académica Zapopan, Jalisco, México.

¹ adriana.tovar@zapopan.tecmm.edu.mx, ² teth.cortes@zapopan.tecmm.edu.mx,

Resumen— La extracción de características de señales acústicas en el reconocimiento de comandos de voz para el control de las acciones de un robot colaborativo, de manera ágil y eficaz, es un desafío para garantizar un entorno seguro para el trabajador. En este proyecto se realizaron grabaciones de señales de audio de diez comandos de voz en español, con entonación neutra. Bajo un enfoque de aprendizaje supervisado a través de un modelo de potenciación del gradiente basado en histogramas HGBT, se logra una métrica de exactitud del 75% para características extraídas de la señal en el dominio del tiempo, denominadas TEMPO, TZ, ZC y RMS. Para características extraídas del espectro de la señal en el dominio de la frecuencia denominadas como CHROMA y MFCC, se logra una mejor métrica de exactitud superior al 90% y con un tiempo de procesamiento menor a los 500ms.

Palabras clave— Reconocimiento de voz, Extracción de características, Espectrograma.

I. INTRODUCCIÓN

La digitalización de los sistemas de manufactura a través de las comunicaciones inalámbricas y la aplicación de la inteligencia artificial es lo que en actualidad está impulsando la cuarta revolución industrial. Y una de las tecnologías indispensables de esta transformación es la robótica colaborativa, que implica la incorporación de un tipo específico de robot, denominado COBOT, cuya principal característica es la interacción con humanos en entornos seguros para los trabajadores, reduciendo el riesgo por lesiones. Los robots colaborativos son máquinas flexibles que se adaptan fácilmente a múltiples aplicaciones, y reducen el tiempo necesario para su puesta en funcionamiento en las celdas de producción. Otra de las características importantes de la robótica colaborativa, es el uso de tecnologías que permiten a los trabajadores dar instrucciones al COBOT rápidamente y sin la necesidad de conocimientos avanzados de programación.

Los comandos de voz, utilizados para dar órdenes a los robots, pueden reemplazar las técnicas de programación tradicional y proporcionar un método más adecuado para asignar acciones y facilitar la colaboración humano máquina. No obstante, el principal desafío al dar instrucciones a un COBOT mediante comandos de voz radica en la manera en cómo se puede aplicar el lenguaje natural de forma sencilla a las acciones que debe realizar el robot. Una forma de alcanzar este objetivo, consiste en vincular directamente los comandos

de entrada a una única acción del robot. Otra forma consiste en usar modelos de aprendizaje para generalizar comandos de entrada. Para ambos casos, lo más importante es establecer la base de los comandos de voz en lenguaje natural que requiere un COBOT para realizar una determinada tarea.

Las señales de voz se caracterizan por su forma no estacionaria, con variaciones lentas en el dominio del tiempo, que el ser humano genera normalmente en segmentos de tiempo muy cortos de entre 5ms y 30ms, identificados como fonemas. El proceso de reconocimiento de comandos de voz consiste en extraer características de una señal acústica y después relacionarla a una acción del COBOT. La extracción de características es un punto fundamental del procesamiento del lenguaje natural y se refiere a la representación de datos informativos en forma vectorial que un sistema de cómputo incorpora a un modelo de clasificación para permitir que la máquina pueda comprender su significado [1]. Como referencia, el cerebro humano es capaz de reconocer el significado de una palabra entre 100ms y 500ms [2].

A. Enfoques en el reconocimiento de voz.

Existen tres enfoques para el reconocimiento de voz [3].

Acústico-Fonético. Se basa en encontrar sonidos asociados a una unidad fonética denominada fonema. En este enfoque la señal en el dominio del tiempo se divide en secciones que se asocian a un fonema. Después se determina la palabra en función de los fonemas encontrados en la señal. Su desventaja radica en que las propiedades acústicas características de cada fonema varían mucho en función de la articulación y entonación de la voz de la persona.

Reconocimiento de patrones. Consiste en el entrenamiento y comparación de patrones. El modelo estocástico HMM (*Hidden Markov Model*) es el más adecuado y más comúnmente usado en este enfoque, ya que permite gestionar información indeterminada o incompleta. Sin embargo, para mejorar su rendimiento se debe realizar un análisis de errores que suele ser complejo y requerir muchos recursos.

Inteligencia artificial. Es una combinación de los enfoques de detección de características acústicas y reconocimiento de patrones que se basan en la adquisición de conocimiento fonético utilizado para la clasificación del sonido. Se suele utilizar un modelo de aprendizaje supervisado que adquiere conocimiento de la relación de eventos fonéticos.

B. Escala de MEL.

Los humanos no percibimos las frecuencias en una escala lineal, detectamos mejor las diferencias en frecuencias más bajas que en frecuencias más altas. En 1937 se propuso la escala MEL (1) para una unidad tal que distancias iguales en el tono sonaran igualmente distantes para el oyente [4].

$$f_{MEL} = 2595 \log_{10} \left(1 + f/700 \right) [dB] \quad (1)$$

La extracción de características a partir de los coeficientes MFCC (*Mel frequency Cepstral Coefficient*) es una de las técnicas más usadas en el reconocimiento automático de voz mediante inteligencia artificial según Abdul y Al-Talabani [5]. Los coeficientes de Mel aportan características típicas de las diferentes frecuencias presentes en una señal y se calculan realizando cinco procesos consecutivos: 1) segmentar la señal, 2) calcular el espectro de potencia, 3) aplicar un banco de filtros Mel a los espectros de potencia, 4) calcular los valores logarítmicos de todos los bancos de filtros y finalmente, 5) aplicar la transformada discreta de Fourier DFT (*Discrete Fourier Transform*). En el presente trabajo la extracción de características se realizó con la biblioteca Librosa [6], que calcula el espectro de potencia o espectrograma mediante la transformada de Fourier de corto plazo STFT (*Short-Term Fourier Transform*).

C. Transformada de Fourier de corta duración STFT

A diferencia de la transformada discreta de Fourier DFT que se obtiene sumando la función en el tiempo de $x(n)$ desde n igual a 0 hasta $N - 1$, como se muestra en (2), la transformada de Fourier de corta duración STFT considera solo un segmento o marco de tiempo de corta duración de una señal más larga y calcula su transformada de Fourier. Esto significa que DFT describe los componentes de frecuencia en la señal promediada a lo largo de todo el tiempo en N cantidad de muestras, mientras que STFT multiplica una función de tiempo por una función de ventana $w(n)$ de duración finita con valores distintos de cero desde n igual a 0 hasta $N - 1$, como se muestra en (3). En la ecuación (2) para calcular la DFT, N es el número de muestras, n es la muestra actual con valores reales de $n = 0, \dots, N - 1$, k es la frecuencia actual discreta, $x(n)$ es el valor de la señal en la muestra n y $X(k)$ es el valor de la transformada discreta de Fourier en la muestra n .

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-i 2\pi \frac{k}{N} n} \quad (2)$$

En la ecuación (3) para calcular STFT, $x = 0, 1, \dots, L - 1$ es una señal de tiempo discreto de valor real de longitud L obtenida mediante muestreo equidistante con respecto a una frecuencia de muestreo fija F_s expresada en Hz. $w = 0, 1, \dots, N - 1$ es la ventana de muestreo, para el caso de una ventana rectangular $w(n) = 1$ para $n = 0, 1, \dots, N - 1$. El parámetro de longitud N determina la duración del segmento considerado, que se calcula como N/F_s segundos. El

parámetro de salto H se especifica en cantidad de muestras y determina el tamaño del paso en el que se desplazará la ventana a través de la señal. Con $m = 0, 1, \dots, M$ y $k = 0, 1, \dots, K$ el número $M = (L - N)/H$ es el índice de cuadro máximo tal que el rango de tiempo de la ventana está completamente contenido en el rango de tiempo de la señal. El número $K = N/2$ para valores pares de N corresponde a la frecuencia de Nyquist. El número complejo $X(m, k)$ denota el k -ésimo coeficiente de Fourier para el m -ésimo marco de tiempo. Para cada marco de tiempo fijo m , se obtiene un vector de espectro de tamaño $K + 1$.

$$X(m, k) = \sum_{n=0}^{N-1} x(n + mH) w(n) e^{-i 2\pi \frac{k}{N} n} \quad (3)$$

El **espectrograma** $y(m, k)$ es una representación bidimensional de la magnitud al cuadrado de STFT $X(m, k)$, como se describe en (4). La función **librosa.stft**, regresa una matriz de valores complejos y el espectrograma se obtiene a partir de la función del valor absoluto.

$$y(m, k) = |X(m, k)|^2 \quad (4)$$

D. Ejemplo de espectrograma para una señal.

Supongamos una señal (5) construida a partir de la suma de tres señales senoidales de 0.5kHz, 1kHz y 1.5kHz con amplitudes de 0.4, 0.8 y 0.5 respectivamente, como se muestra en la Figura 1.

$$x(n) = 0.4 \sin(2\pi 500t) + 0.8 \sin(2\pi 1000t) + 0.5 \sin(2\pi 1500t) \quad (5)$$

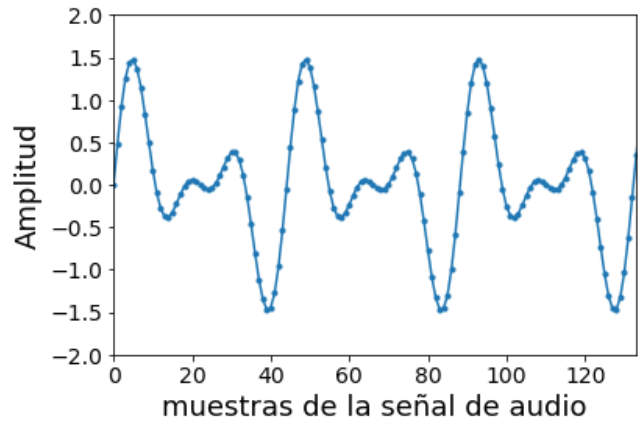


Fig. 1. Señal construida a partir de la suma de tres señales senoidales.

En la Figura 2 se muestra la señal en el dominio de la frecuencia, calculada a partir de la transformada discreta de Fourier. En la Figura 3 se muestra el espectrograma obtenido a partir de la magnitud del cuadrado de STFT como se definió en la ecuación (3) en la escala de MEL.

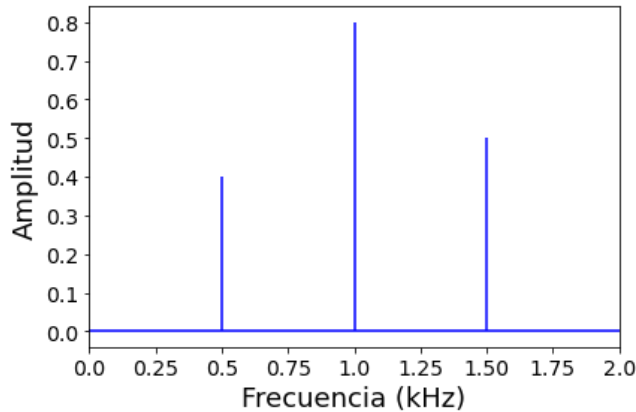


Fig. 2. Transformada discreta de Fourier DFT de la ecuación (4).

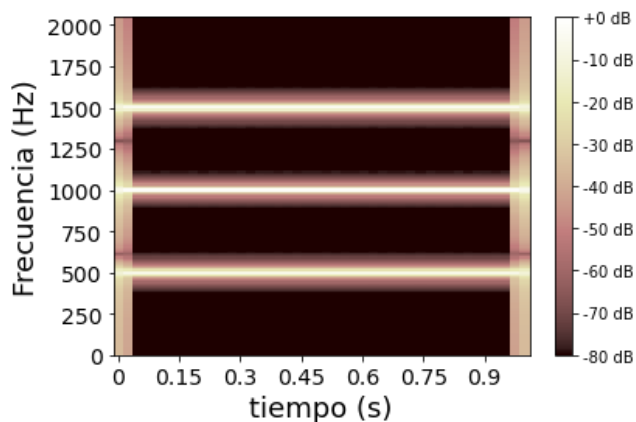


Fig. 3. Espectrograma de la señal de la ecuación (4).

El contenido del presente artículo se dividió de la siguiente manera, en la sección II se hace una revisión del estado del arte en literatura científica sobre el reconocimiento de comandos de voz para el control de robots. En la sección III se describen cada una de las características extraídas de las señales de audio en el dominio del tiempo y en el dominio de la frecuencia. En la sección IV se presentan los resultados sobre el rendimiento modelo de aprendizaje y los tiempos de extracción de datos informativos en función del número de características. Finalmente, en la sección V presentan las conclusiones.

II. REVISION DEL ESTADO DEL ARTE

En esta sección del documento se describen algunos sistemas de reconocimiento de voz para el control de robots.

En Tenorio Gonzalez *et al.* [7] se usaron 20 comandos de voz en español, organizados en dos vocabularios de frases cortas y palabras para controlar el movimiento de un robot móvil. Los 10 comandos de frases cortas usados fueron: hacia adelante, hacia atrás, gira a la izquierda, ve a tu derecha, para ahí, sigue así, por ahí no, muy bien, así no, hasta ahí. Y los 10 comandos de palabras usados fueron: avanzar, regresar, izquierda, derecha, fin, bien, mal, excelente, terrible y objetivo. Usaron el sistema de reconocimiento Sphinx3 con

un nivel de modelos acústicos T22 para la transcripción básica de fonemas del español, con tres niveles de etiquetas: fonético, fonemas y palabras. El español hablado en México es un lenguaje de 22 fonemas definido por 17 consonantes y 5 vocales. Una palabra se conforma por fonemas, y a su vez un fonema se conforma por un conjunto de rasgos fonéticos propios del lenguaje. Para el proceso de etiquetado se obtiene el espectrograma, oscilograma y el tono (*pitch*) de la señal acústica. Las transcripciones del habla son analizadas por un intérprete diseñado para buscar palabras de interés según la etapa del proceso de aprendizaje. Una vez identificadas, el intérprete modifica el comportamiento del algoritmo de aprendizaje según su significado. Si las palabras son calificadoras, la función de recompensa se modifica añadiendo un valor según su significado. Si se comprende una orden de acción, se envía la acción correspondiente al robot, modificando también el proceso de aprendizaje reforzado.

En Vicente Samper *et al.* [8] controlan un manipulador robótico UR3 del fabricante Universal Robot mediante la interacción de lenguaje natural a través de un dispositivo Alexa de Amazon. Utilizan la base de datos DynamoDB que pertenece a los servicios en la nube que ofrece AWS. El controlador del robot revisa constantemente las variables definidas en la base de DynamoDB y cuando algunas de estas variables modifica su valor a través de un comando de voz, el robot realiza una acción previamente programada. Las variables modificadas en la base de datos se vinculan directamente a las coordenadas cartesianas del robot o la ejecución de operaciones de ensamble previamente definidas.

En Kaczmarek *et al.* [9] se usó un sistema de comandos por señas visuales y voz para controlar las tareas de un robot industrial. Los comandos de voz en inglés, fueron *plus x*, *minus x*, *plus y*, *minus y*, *pluz z*, *minus z*, *stop*. Los investigadores programaron en Visual Studio una interfaz y usaron la librería System.Speech.Recognition de Microsoft. Para mediciones en una estación real, reportan un retraso de la reacción del robot al comando de voz entre 1.47s y 2.07s, y para una estación virtual de 2s a 2.68s. Los investigadores atribuyen esta diferencia entre el entorno real y el virtual a la capacidad de cómputo y al protocolo de comunicación. Los comandos de voz fueron grabados con una emoción neutra y una intensidad de 70 dB con 60 dB de ruido, reportan una tasa de reconocimiento del 91.43%.

En Telkes *et al.* [10] se realizó el control de un robot mediante comandos de voz en inglés, como: *start robot*, *stop robot*, *open*, *close*, *home*, *position up*, *position down*, *position left*, *position right*, *rotate*, *home*. Y comandos basados en aprendizaje de lenguaje de alto nivel como: *finish*, *task*, *repeat*, *pick*, *place*, *stack*, *push*. Los investigadores hacen una comparación entre las dos formas de dar instrucciones a robots mediante comandos de voz, ver Tabla I. Para el reconocimiento de lenguaje natural utilizan la aplicación VOSK, filtrando palabras innecesarias para la operación del

robot. Para señales de audio sin ruido reportan resultados experimentales en el reconocimiento de comandos de voz sin errores. Con ruido y en el peor de los casos, la aplicación falló en 3 ocasiones para la misma instrucción dada 10 veces. Para comandos basados en aprendizaje de lenguaje de alto nivel, los investigadores reportan la operación de ensamble de un sistema de engranes helicoidales con seis pasos y un sistema de engranes planetarios con nueve pasos. En ambos casos, las instrucciones de voz guiaron el proceso y en el caso del engranaje planetario, se incluyeron tres tareas manuales y colaborativas con el robot.

TABLA I

COMPARACIÓN ENTRE FORMAS DE DAR INSTRUCCIONES A ROBOTS MEDIANTE COMANDOS DE VOZ

	Basado en comandos	Basado en aprendizaje
Descripción	El robot vincula cada palabra directamente a una acción de control o movimiento	El robot utiliza instrucciones de lenguaje de alto nivel para aprender acciones
Comandos vocabulario	Bien definidos	Poco definidos
Acciones	Limitado	Amplio
Respuesta	Corto plazo	Mediano o largo plazo
	Rápida	Lenta

III. MÉTODO

En el presente trabajo se construyó un *dataset* con 10 comandos de voz para controlar el movimiento de un manipulador robótico, descritos en la Figura 4. Se hicieron 20 grabaciones de cada uno de los comandos con diferentes personas y usando una entonación neutra. El proceso de extracción de características se implementó utilizando la biblioteca Librosa para Phyton en un equipo de cómputo ASUS TUF F15 con procesador i7 de INTEL, 16 GB de RAM y GPU RTX4050 de NVIDIA.



Fig. 4. Descripción de los comandos de voz de C1 a C10 propuestos para el control de un manipulador robótico colaborativo UR3 [11].

E. Características en el dominio de la frecuencia.

Para obtener el espectrograma, se calculó la transformada de Fourier de corto plazo STFT en la señal de voz con una longitud de ventana de 1024 y una longitud de salto de 512. Los coeficientes MFCC se obtuvieron por medio de la librería **librosa.feature.mfcc** aplicando filtros Mel, tomando la magnitud logarítmica y aplicando una transformada discreta

del coseno DCT a los espectrogramas. En las líneas de código 14 y 16 de la Figura 5 se declaró en 40 el número de coeficientes cepstrales de frecuencia Mel.

La característica extraída por medio de la librería **librosa.feature.chroma_stft** representa todo el espectro proyectado en 12 contenedores, es decir, indica la intensidad de cada uno de los 12 semitonos distintos o CROMA de la octava en cada intervalo de tiempo. La escala cromática es una escala de 12 sonidos que se compone de una sucesión de semitonos, que en el sistema inglés se identifican como C, D, E, F, G, A, B, C#, D#, E#, F#, G#, A#, y B#. Una octava se define como un intervalo de frecuencias entre dos sonidos cuya frecuencia central del segundo es el doble del primero, como se muestra en la tabla II para el semitono de A0 a A8 [12].

TABLA II

FRECUENCIAS CENTRALES DEL SEMITONO A0-A8

8va	A0	A1	A2	A3	A4
f(Hz)	27.5	55	110	220	440
8va	A5	A6	A7	A8	
f(Hz)	880	1760	3520	7040	

F. Características en el dominio del tiempo.

La característica extraída mediante **librosa.feature.tempo** estima el ritmo de señal de audio o TEMPO calculado en pulsaciones por minuto (PPM), se asocia a la rapidez con la que se pronuncia un comando de voz. El ritmo se extrae de la envolvente de la señal de audio y muestra los cambios significativos de su amplitud, sin importar el tono o frecuencia de la misma [13].

La característica extraída mediante **librosa.feature.tonnetz** estima el centroide del timbre TZ, calculando la distancia euclidiana entre los cuadros de análisis $n+1$ y $n-1$ para desarrollar una medida de cambio armónico para el cuadro n . Un pico en la función de detección indica una transición de una región armónicamente estable a otra. Se utiliza para detectar cambios en el contenido armónico de la señal asociado al timbre de la voz, que depende de la anatomía vocal, interpretación, intensidad y forma de expulsar el aire. Esta característica puede distinguir una voz femenina fuerte y clara, de una voz masculina grave y suave, así como distinguir melodías o instrumentos [14].

La característica extraída mediante la librería **librosa.feature.zero_crossing_rate** calcula la tasa de cruce por cero ZC de una señal de audio en el tiempo y representa los cambios de signo de la señal durante el cuadro. Es decir, la cantidad de veces que la señal cambia de valor, de positivo a negativo y viceversa, dividida por la longitud del cuadro. La tasa de cruce por cero se suele utilizar como un algoritmo básico de detección de tono, también se utiliza para la detección de actividad de voz con el propósito de determinar si hay presencia de voz humana en un segmento de audio. Su valor también se usa ampliamente para clasificar sonidos de

percusión, tanto en el reconocimiento de voz como en la extracción de información musical.

La característica extraída por medio de la librería **librosa.feature.rms** calcula el valor cuadrático medio RMS (*Root Mean Square*) para cada cuadro, ya sea a partir de las muestras de audio o de un espectrograma. Es más rápido obtener el valor RMS a partir de muestras de la señal de audio, ya que no requiere calcular STFT. Sin embargo, es preferible obtener el valor RMS a partir de un espectrograma porque brinda una representación más precisa de la energía a lo largo del tiempo debido a que el cuadro se divide en ventanas de muestreo más cortas.

En el código programado para la extracción de características de las señales de audio de los comandos de voz, se asignó un valor de 1000 a la variable FM, como se muestra en la línea 3 de la Figura 5, que se utilizó como un factor de multiplicación de los resultados de la media μ y la desviación estándar σ de CROMA, MFCC, RMS, ZC y TZ; el resultado de TEMPO simplemente se agregó al vector de resultados.

```

No.  Código Python
1  def extractFeature1(fileName, mfcc, chroma, rms, zero_crossing_rate, tonnetz, tempo):
2  with soundFile.SoundFile(fileName) as soundFile:
3      FM=1000 # factor de multiplicación
4      X = soundFile.read(dtype='float32')
5      sampleRate = soundFile.samplerate
6      result=nup.array([])
7      stft= nup.abs(librosa.stft(X))
8      if chroma:
9          chroma1=FM*nup.mean(librosa.feature.chroma_stft(S=stft, sr=sampleRate),T,axis=0)
10         result=nup.hstack((result, chroma1))
11         chroma2=FM*nup.std(librosa.feature.chroma_stft(S=stft, sr=sampleRate),T,axis=0)
12         result=nup.hstack((result, chroma2))
13     if mfcc:
14         mfccs1=FM*nup.mean(librosa.feature.mfcc(y=X, sr=sampleRate, n_mfcc=40),T,axis=0)
15         results=nup.hstack((result, mfccs1))
16         mfccs2=FM*nup.std(librosa.feature.mfcc(y=X, sr=sampleRate, n_mfcc=40),T,axis=0)
17         results=nup.hstack((result, mfccs2))
18     if rms:
19         RMS1=FM*nup.mean(librosa.feature.rms(y=stft))
20         result=nup.hstack((result, RMS1))
21         RMS2=FM*nup.std(librosa.feature.rms(y=stft))
22         result=nup.hstack((result, RMS2))
23     if zero_crossing_rate:
24         ZCR1=FM*nup.mean(librosa.feature.zero_crossing_rate(y=X))
25         result=nup.hstack((result, ZCR1))
26         ZCR2=FM*nup.std(librosa.feature.zero_crossing_rate(y=X))
27         result=nup.hstack((result, ZCR2))
28     if tonnetz:
29         TZ1 =FM*nup.mean(librosa.feature.tonnetz(y=X, sr=sampleRate))
30         result=nup.hstack((result, TZ1))
31         TZ2 =FM*nup.std(librosa.feature.tonnetz(y=X, sr=sampleRate))
32         result=nup.hstack((result, TZ2))
33     if tempo:
34         TMP, beat_frames = librosa.beat.beat_track(y=X, sr=sampleRate)
35         result=nup.hstack((result, TMP))
36     return result

```

Fig. 5. Código para la extracción de características de la señal de audio.

G. Características extraídas de los comandos C4 y C10

En esta sección se muestran las características extraídas, RMS, ZC, TEMPO y espectrograma para las señales acústicas de los comandos C4 y C10. En la Figura 6 se muestra el oscilograma de la señal de audio de la grabación del comando de voz de C4, donde se le ordena al manipulador robótico: muévete hacia atrás, y en la Figura 7 se muestra su espectrograma. En la Figura 8 se muestra el oscilograma de la señal de audio de la grabación de voz del comando C10, donde se le ordena al manipulador robótico: para ahí y en la Figura 9 se muestra su espectrograma.

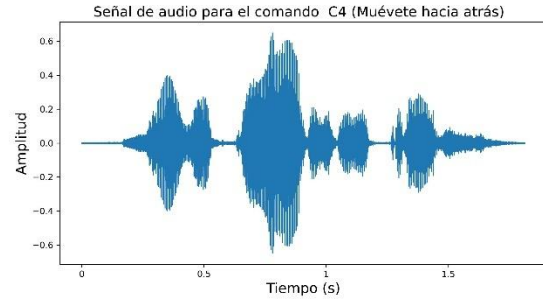


Fig. 6. Señal de audio de la grabación de un comando de voz C4.

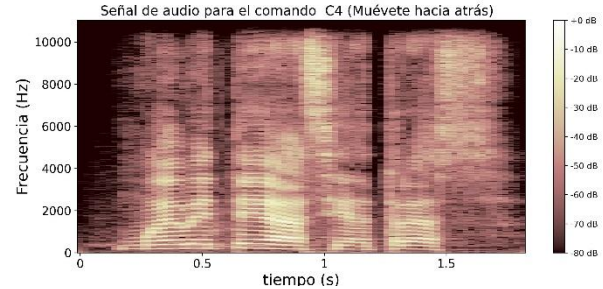


Fig. 7. Espectrograma de la grabación de un comando de voz C4.

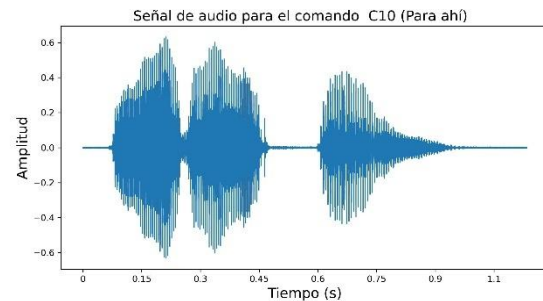


Fig. 8. Señal de audio de la grabación de un comando de voz C10.

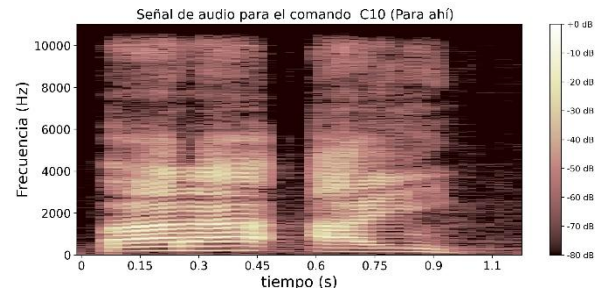


Fig. 9. Espectrograma de la grabación de un comando de voz C10.

Para 20 grabaciones de audio de los comandos C4 y C10, en la Figura 10 se muestra la gráfica de dispersión de la desviación estándar σ contra la media μ del valor cuadrático RMS. En la Figura 11 se muestra la gráfica de dispersión de la desviación estándar σ contra la media μ de la tasa de cruce por cero ZC. En la Figura 12 se muestra la gráfica de dispersión de la desviación estándar σ del valor cuadrático RMS contra las pulsaciones por minuto.

Se eligieron los comandos C4 y C10 para mostrar de que es posible identificar sus diferencias a partir de las

características extraídas del oscilograma y del espectrograma de las señales acústicas. Y en consecuencia la viabilidad de aplicar un modelo como potenciación del gradiente basado en histogramas HGBT (*Histogram-based Gradient Boosting Classification Tree*) que permita reconocer los comandos de voz vinculados directamente a una acción manipulador robótico. El modelo HGBT se implementó mediante la biblioteca scikit-learn [14].

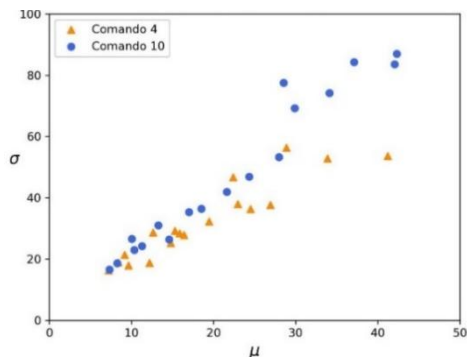


Fig. 10. Gráfica de dispersión de la desviación estándar σ contra la media μ del valor cuadrático medio RMS de las grabaciones de C4 y C10.

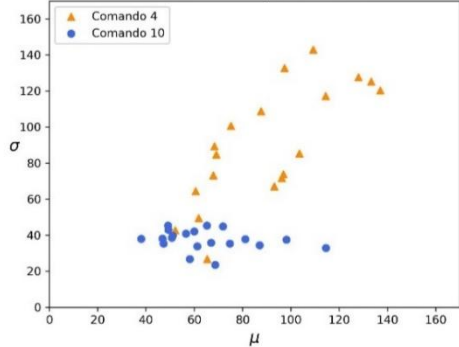


Fig. 11. Gráfica de dispersión de la desviación estándar σ contra la media μ de la tasa de cruce por cero ZC de las grabaciones de C4 y C10.

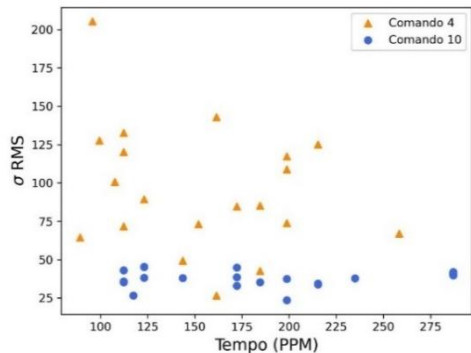


Fig. 12. Gráfica de dispersión de la desviación estándar σ del valor cuadrático medio RMS contra TEMPO de las grabaciones de C4 y C10

IV. RESULTADOS

Para el conjunto de datos de entrenamiento, con una tasa de aprendizaje de $LR=0.02$ y un número máximo de hojas por

nodo en 10, después de 50 iteraciones el modelo HGBT alcanzó una métrica de exactitud (*accuracy*) del 98.9%, en otros experimentos, el modelo HGBT alcanzó una métrica de exactitud del 99.4% con 30, 20, 15 y 12 iteraciones para LR de 0.04, 0.06, 0.08 y 0.10 respectivamente, como se muestra en la Figura 13 y en la Tabla III.

TABLA III
EXACTITUD DE HGBT CONTRA EL NÚMERO DE CARACTERÍSTICAS EXTRAÍDAS, PARA DIFERENTES PARÁMETROS DE LR.

						accuracy (%)					
TEMPO	TZ	ZC	RMS	CROMA	MFCC	Características extraídas	LR=0.02	LR=0.04	LR=0.06	LR=0.08	LR=0.10
•						1	27.2	27.7	27.7	27.7	27.7
•	•					3	51.6	56.6	56.1	55.5	57.2
•	•	•				5	69.4	71.1	72.7	72.2	70.5
•	•	•	•			7	68.8	73.3	73.8	71.6	75.0
•	•	•	•	•		31	91.6	94.4	95.0	93.8	93.8
•	•	•	•	•	•	111	98.9	99.4	99.4	99.4	99.4

En la Figura 14 se muestra el tiempo para extraer las características de la señal de audio. La extracción de 1 característica como TEMPO requiere una $\mu=0.823s$, la extracción de 3 características, TEMPO y TZ requiere una $\mu=13.575s$, la extracción de 5 características TEMPO, TZ y ZC requiere una $\mu=13.804s$, la extracción de TEMPO, TZ, ZC y RMS requiere una $\mu=13.622s$. La extracción de características mediante CROMA y MFCC que implica el cálculo de STFT, CROMA eleva el número de características a 31 datos y requiere una $\mu=16.116s$, finalmente MFCC eleva la el número de características a 111 datos y requiere una $\mu=17.024s$.

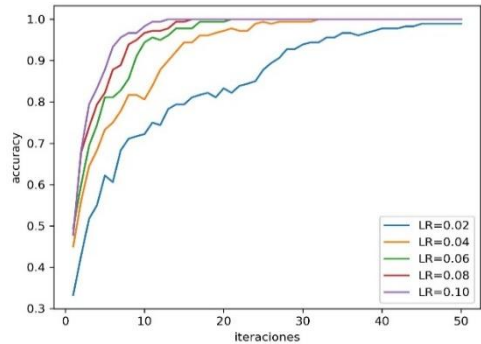


Fig. 13. Gráfica de accuracy contra el número de iteraciones.

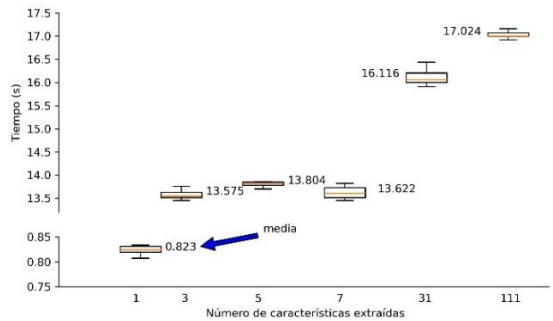


Fig. 14. Tiempo de extracción contra el número de características.

Para 10 pruebas de la extracción de 111 características de una grabación de audio para el reconocimiento de un comando de voz C10 se necesitó una $\mu=0.163s$ con una $\sigma=0.035s$, y para una grabación de audio del comando C4 se necesitó una $\mu=0.174s$ con una $\sigma=0.033s$.

V. CONCLUSIONES

La cantidad de datos obtenidos a partir de las características extraídas de la señal acústica influye significativamente en el rendimiento del modelo de aprendizaje. Para datos de entrenamiento, se logró una métrica de exactitud mayor al 90% con el modelo HGBT, para las características espectrales obtenidas por CROMA y MFCC. La suma de las características obtenidas mediante TEMPO, TZ, ZC y RMS a partir de la señal en el dominio del tiempo muestran que el modelo HGBT tiene un rendimiento menor, no mayor al 75%.

La extracción de características espectrales de los datos de entrenamiento necesita un tiempo de procesamiento mayor en comparación a la extracción de características en el dominio del tiempo. Las pruebas de extracción de 111 características para el reconocimiento de los comandos C10 y C4 muestran que el tiempo de procesamiento es menor a los 500ms, de forma conveniente para el control de las acciones de un COBOT en tiempo real.

AGRADECIMIENTOS

Agradecemos al Ingeniero en electrónica Raúl Romero Valenzuela por su apoyo durante las grabaciones de los comandos de voz, así como a la U. A. Zapopan del Instituto Tecnológico José Mario Molina Pasquel y Henríquez, al Tecnológico Nacional de México TecNM y a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación SECIHTI del gobierno de México por el apoyo recibido para la divulgación del presente trabajo.

REFERENCIAS

- [1] X. Lv, M. Zhang, and H. Li, "Robot control based on voice command", in *Proc. IEEE Int. Conf. Automat. Logistics*, Qingdao, China, pp. 2490-2494, 2008. DOI: <https://doi.org/10.1109/ICAL.2008.4636587>.
- [2] K. Pammer, P. C. Hansen, M. L. Kringelbach, I. Holliday, G. Barnes, A. Hillebrand, K. D. Singh, and P. L. Cornelissen, "Visual word recognition: the first half second". *Neuroimage*, vol. 22, no. 4, pp. 1819-1825, 2004, DOI: <https://doi.org/10.1016/j.neuroimage.2004.05.004>.
- [3] N. Desai, K. Dhameliya, and V. Desai, "Feature extraction and classification techniques for speech recognition: A review". *International Journal of Emerging Technology and Advanced Engineering*, vol. 3 no. 12, pp. 367-371, 2013.
- [4] S. S. Stevens, J. Volkman, and E. B. Newman. "A scale for the measurement of the psychological magnitude pitch". *Journal of the Acoustical Society of America*, vol. 8, pp.185–190, 1937. DOI: <https://doi.org/10.1121/1.1915893>.
- [5] Z. K. Abdul and A. K. Al-Talabani, "Mel Frequency Cepstral Coefficient and its Applications: A Review" in *IEEE Access*, vol. 10, pp. 122136-122158, 2022. DOI: <https://doi.org/10.1109/ACCESS.2022.3223444>.
- [6] B. McFee, C. Raffel, D. Liang, D.P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. "Librosa: Audio and music signal analysis in python". In *Proceedings of the 14th Python in science conference*, Austin, TX, USA, pp. 18–25, 2015. DOI: <https://doi.org/10.25080/Majora-7b98e3ed-003>.
- [7] A. C. Tenorio Gonzalez, E. F. Morales, L. Villaseñor Pineda. "Teaching a Robot to Perform Tasks with Voice Commands". In: G. Sidorov, A. Hernández Aguirre, C. A. Reyes García Eds. *Advances in Artificial Intelligence. MICAI 2010. Lecture Notes in Computer Science*, vol. 6437. Springer, Berlin, Heidelberg, 2010. DOI: https://doi.org/10.1007/978-3-642-16761-4_10.
- [8] J. M. Vicente Samper, I. Campos, B. Sanz Worrell, A. Rodriguez, J. M. Oñate Tevar, and J. M. Sabater Navarro, "Ejemplo de integración de Alexa con un robot UR", en *XL Jornadas de Automática: libro de actas, Ferrol*. pp. 360-365, 2019. DOI: <https://doi.org/10.17979/spudc.9788497497169.360>.
- [9] W. Kaczmarek, J. Panasiuk, S. Borys, and P. Banach, "Industrial Robot Control by Means of Gestures and Voice Commands in Off-Line and On-Line Mode". *Sensors*, vol. 20, no. 21, 6358, 2020. DOI: <https://doi.org/10.3390/s20216358>.
- [10] P. Telkes, A. Anglauraud, and R. Pieters, "Instructing Hierarchical Tasks to Robots by Verbal Commands", in *2024 IEEE/SICE International Symposium on System Integration (SII)*, Ha Long, Vietnam, pp. 1139-1145, 2024. DOI: <https://doi.org/10.1109/SII58957.2024.10417491>.
- [11] J. Pelegrí. "La integración de las pinzas FESTO en los COBOTS de UR, cuestión de minutos gracias a los URCAPS". Universal Robots, 2021. [En línea]. Disponible en: <https://www.universal-robots.com/es/blog/integracion-pinzas-festo-brazos-roboticos/>.
- [12] D. Byrd, "Una tabla de tonos musicales", Facultad de Informática y Escuela de Música Universidad de Indiana, 2007. [En línea]. Disponible en: <https://homes.luddy.indiana.edu/donbyrd/Teach/MusicalPitchTable.htm>.
- [13] L. F. Romero Morales y J. M. Flores Albino. "Algoritmo para el cálculo del ritmo en una señal de audio digital". *Research in Computing Science*, vol. 120, pp. 113-122, 2016.
- [14] T. Łukaszewicz and D. Kania, "Trajectory of Fifths Based on Chroma Subbands Extraction—A New Approach to Music Representation, Analysis, and Classification," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 3, pp. 2157-2169, 2025. DOI: <https://doi.org/10.1109/TPAMI.2024.3519420>.
- [15] F. Pedregosa, et al. "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830., 2011.