VOLUMEN 28
Verano de la Ciencia XXIX
ISSN 2395-9797
www.jóvenesenlaciencia.ugto.mx

Control de un sistema robótico para la interacción multimodal en un entorno semiestructurado

Control of a robotic system for multimodal interaction in a semi-structured environment

Jesús Aldana Guzmán[1]
[1]Ingeniería en Mecatrónica - DICIS
j.aldanaguzman@ugto.mx[1]

Sergio Francisco Arce Vázquez[2]
[2]Ingeniería en Mecatrónica - DICIS
sf.arcevazquez@ugto.mx[2]

Alejandro Gutiérrez Castaño[3]
[3]Ingeniería en Mecatrónica - DICIS
a.gutierrezcastano@ugto.mx[3]

Miguel Ángel Pérez Sandoval[4]
[4]Ingeniería en Mecatrónica - DICIS
ma.perezsandoval@ugto.mx[4]

Christian Eduardo Rodríguez García[5]
[5]Ingeniería en Mecatrónica - DICIS
ce.rodriguez.garcia@ugto.mx[5]

Felipe Trujillo Romero[6]
[6]Depto. Ingeniería Electrónica - DICIS
fdj.trujillo@ugto.mx[6]

Resumen

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

Palabras clave: Robot de servicio; Lengua de Señas Mexicana; Reconocimiento de patrones; Modelos Ocultos de Markov; Lenguaje Natural; interacción Humano-Robot.

Introducción

En el área de robótica, los sistemas de reconocimiento automático han tomado un papel muy importante en los procesos de interacción humano-robot. Por medio de éstos, se pretende proveer de un tipo de comunicación natural entre los seres humanos y los sistemas robóticos, ya sea utilizando características del habla, de rostros, formas, colores, gestos corporales, o cualquier otra seña utilizada en la interacción natural de las personas. Por este motivo, se han desarrollado sistemas de control multimodal que permitan la interacción humano-robot mediante diferentes tipos de lenguaje natural como la voz y el lenguaje de señas.

Los robots móviles de control multimodal representan un avance significativo en área reciente de aplicaciones en robótica es la relacionada con robots de asistencia y educación, llamada comúnmente robótica de servicio, la cual es quizá una de las más demandantes en Interacción Humano-Robot (HRI). Como ejemplo de estas aplicaciones están aquellas que pretender aumentar el conjunto de tareas que una persona con discapacidad

pueda llevar a cabo de forma independiente. Estas tareas incluyen asistencia de navegación en ambientes no estructurados (Kulyukin et al., 2006), o facilidades de transportación (Yanco,2001). Según el censo de población y vivienda de 2020, en México residen aproximadamente 15 millones de personas mayores de 60 años, y más de 6 millones de personas presentan algún tipo de discapacidad (INEGI Censo población, 2020). Razón por la cual, este grupo de personas requiere asistencia y cuidados diarios, lo cual no siempre es fácilmente atendido, ya que, por diversas razones, estas personas a menudo carecen de la asistencia humana necesaria.

Dentro de esta área de aplicación, referente a la robótica de servicio, se encuentra ubicado el estudio del presente proyecto de investigación. Por esa razón, el proyecto desarrollado consiste en el diseño de un escenario de simulación que emule la estructura de una vivienda común. Una vez reconocido el entorno, el robot debe ser capaz de interpretar comandos proporcionados por el ser humano a través de dos modalidades distintas, visual y auditiva. Posteriormente, ejecutar las tareas correspondientes a cada instrucción dada por el usuario. Para los fines específicos de este proyecto, se ha seleccionado el robot Pioneer 3-DX, un robot de tipo diferencial, debido a su adaptabilidad a las características y necesidades del proyecto. En tal caso, es necesario utilizar dos tipos de reconocimiento automático: reconocimiento del lenguaje de señas y reconocimiento de comandos vocales. Es por ello por lo que en los siguientes párrafos se hará un bosquejo de los trabajos existentes en reconocimiento de señas y comandos vocales, así como las diferentes aplicaciones que se dan a este tipo de investigaciones.

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

No obstante, no todos los trabajos necesitan de una estructura o elemento mecánico para describir los movimientos y gestos de un usuario. En este sentido, los sistemas estereoscópicos han sido de gran utilidad en la captura de imágenes, así como en la extracción de información necesaria para reconocer algún gesto humano (Nickel y Stiefelhagen, 2007; Maldeni et al., 2011). El sensor Kinect, también ha servido como proveedor de datos y características en el reconocimiento de señas. Ejemplo de ello, se muestra en Trujillo-Romero y García-Bautista (2021), donde se realiza un reconocimiento de siete señas utilizando redes neuronales del tipo perceptrón multicapa.

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

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

El reconocimiento automático del habla (ASR por sus siglas en inglés) es una técnica que permite convertir señales de voz en texto para reconocimiento y comprensión de ideas (Liu y Li, 2010). Posteriormente, surgen nuevos métodos de reconocimiento como Redes Neuronales y Modelos Ocultos de Markov, los cuales son usados normalmente como complemento uno del otro (Deshmukh, 2020). En las últimas décadas los sistemas de reconocimiento automático de voz han experimentado un notable progreso. En particular, se han dado numerosas investigaciones que utilizan un vocabulario extenso, reconocimiento en tiempo real y casos con independencia de usuario (Khanna, 2021). Al mismo tiempo, las aplicaciones comerciales de

reconocimiento de voz son más encontradas en el mercado. Por ejemplo, en la industria, las comunicaciones y teléfonos móviles, los sistemas eléctricos automotrices, la medicina, las casas inteligentes, etc.

Cabe mencionar que muchos de los trabajos desarrollados para reconocimiento del habla, están pensados en sistemas robóticos, ya sea para manipulación de objetos o movimiento de este. Deuerlein et al. (2021), desarrollaron una interfaz para el control de voz de un robot, permitiendo a los usuarios controlar los movimientos del robot a través de comandos vocales. En (Zemke et al., 2020) se explora la integración de robots en restaurantes de comida rápida, centrándose en las perspectivas de los clientes sobre el impacto social, la limpieza, la seguridad alimentaria y el reconocimiento de voz. Por su parte Bakouri et al. (2022), desarrollaron una aplicación móvil Android para controlar una silla de ruedas robótica mediante comandos de voz basado en redes neuronales convolucionales y un modelo de reconocimiento de voz; aunque el sistema se evaluó utilizando un corpus de habla en inglés mostrando resultados con una precisión de aproximadamente el 87,2%.

Métodos y materiales

En esta sección se abordará la descripción de las diferentes herramientas y métodos empleados en la realización de este trabajo. Se iniciará con una descripción referente a la Lengua de Señas Mexicana para situar el contexto. Para continuar con la introducción de la librería de Mediapipe usada para la creación del módulo de reconocimiento de gestos visuales. Posteriormente, se hablará de lo que son los Modelos ocultos de Markov y el espectograma Mel usados para capturar y crear la base de datos de muestras de voz. Finalmente se mencionará al programa CoppeliaSim, el cual es el software de simulación utilizado para evaluar el sistema de control multimodal implementado.

Lengua de Señas Mexicana

La Lengua de Señas Mexicana, se deriva del antiguo lenguaje de signos francés traído a México en 1869. Para ese tiempo, en México ya existían personas sordas que usaban señas para comunicarse, estas señas se fueron incorporando al nuevo lenguaje y se complementaron ampliamente con el sistema francés (Calvo, 2004). Como la mayoría de las lenguas de señas la LSM está compuesta de la dactilología e ideogramas (Serafín y González, 2011).

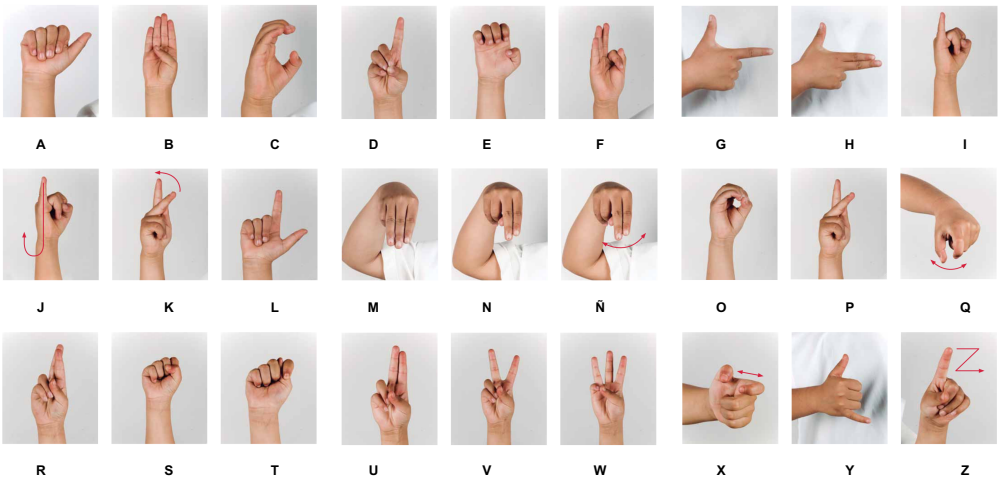


Figura 1. Alfabeto de la LSM (Serafín y González, 2011).

El alfabeto de la LSM es el que se muestra en la Figura 1 y está formado tanto por señas estáticas como por señas dinámicas. Las señas dinámicas son aquellas cuya trayectoria de movimiento es representada por una flecha color rojo (ver Figura 1). En la ejecución del alfabeto de la LSM se hace uso de una mano base y una mano dominante. Normalmente la mano dominante es la izquierda para personas zurdas y la mano derecha para personas diestras.

La LSM, corresponde al ISO 639-2 sgn-MX (Código estándar Mex-Esp, 2021). El ISO-639-2, es el estándar internacional para los códigos de idioma, cuyo objetivo es establecer los códigos reconocidos internacionalmente para la representación de lenguajes o familias de lenguajes.

Hay que tener en cuenta que una cantidad significativa de personas sordas son mayormente monolingües en la LSM. Esto significa que ni el LSM y ni el español son adecuados para una completa comunicación entre la comunidad de sordos de México en ninguna forma, sea por video, por escrito, o por contacto personal.

MediaPipe

MediaPipe es un framework multiplataforma de código abierto desarrollado por la empresa Google que aplica herramientas de ML que permiten desarrollar aplicaciones en dispositivos móviles, ordenadores e incluso a través de la web. Este framework contiene modelos de ML para detección de rostros, seguimiento de manos, segmentación de cabello, detección y seguimiento 2D y 3D de objetos y detección y seguimiento de posturas humanas entre muchos otros (Lugaresi et al., 2019).

MediaPipe está desarrollado para que pueda trabajar en Android, IOS, Python, C++, y Javascript. Se podría pensar que la utilización de modelos de aprendizaje maquina en dispositivos con Android o IOS, que mayormente son smartphones o tabletas, conllevaría un procesamiento muy lento, sin embargo, esto no es así. Gracias a esto último, este framework es muy utilizado en aplicaciones sociales como lo es ”Instagram” o ”Snapchat” para la realización de sus filtros en tiempo real (MediaPipe, 2020).

Además, existe MediaPipe Solutions el cual ofrece bibliotecas y herramientas para implementar rápidamente técnicas de inteligencia artificial y aprendizaje automático en el desarrollo de aplicaciones. Y al formar parte del proyecto de código abierto de MediaPipe, estas se pueden conectar, personalizar y utilizar en varias plataformas de desarrollo sin costo alguno.

Modelo Oculto de Markov (HMM)

Este modelo tiene como objetivo modelar la reproducción de voz, siendo capaz de crear una estructura para parametrizar las características que componen el habla y, con base en ellas, generar probabilidades estadísticas para la identificación del hablante. Estas estadísticas se obtienen a partir de probabilidades de ocurrencia de un conjunto de datos sonoros extraídos de una base de datos, lo cual permite evitar el trabajo directo con audio.

El HMM comienza con un conjunto de cadenas de Markov, donde se encuentra una serie de datos con una probabilidad de transición entre ellos y con posibilidades observables de resultado de cada uno, con una determinada probabilidad de salida. Esto significa que se busca maximizar la probabilidad de ocurrencia tras buscar la sucesión de observables (conjunto de datos característicos extraídos de cada muestra de audio).

Para obtener estas probabilidades estadísticas, es necesario analizar una base de datos que contenga los audios con sus respectivas transcripciones. A partir de estos audios, se tomarán muestras pequeñas y se agruparán por fonemas. Cada fonema tendrá su sucesión de estados, donde cada estado estará formado por una serie de sucesiones de tramas y cada trama se asociará con una característica. Una vez analizada la base de datos, se podrán establecer probabilidades de transición entre estados y salidas para cada característica.

Como se puede inferir, el modelo HMM ha sido empleado en este proyecto para el desarrollo del sistema de reconocimiento de voz, debido a su capacidad de predicción basada en el análisis probabilístico de series de datos. Para implementar esto, se requirió la generación de una base de datos eficiente, compuesta por un conjunto de muestras de audio para cada comando (clase), con un mínimo de 15 muestras por comando. El audio de prueba utilizado es aquel grabado en tiempo real por el usuario.

Espectrograma de Mel

Es una representación usada para el procesamiento de señales de audio, sobre todo para el reconocimiento de voz y su análisis, tomando como base la escala Mel, la cual a su vez se encuentra sustentada en la frecuencia perceptual diseñada para aproximar la forma en que oímos los humanos. Dado que la percepción

humana de la frecuencia no es lineal, somos más sensibles a variaciones en tonos bajos que en tonos altos. Por esta razón, la escala Mel es lineal en frecuencias bajas y logarítmica en frecuencias altas.

La obtención del espectro de Mel tiene como objetivo representar señales de audio como series de datos en el tiempo, específicamente las frecuencias de la señal de sonido, para su posterior análisis mediante el modelo oculto de Markov. Esto permite realizar un análisis probabilístico y la subsiguiente predicción en la clasificación del audio. Por lo tanto, la obtención del espectro de Mel requiere un proceso específico que debe seguirse rigurosamente para asegurar que el conjunto de datos sea óptimo y eficiente. A continuación, se describe dicho proceso:

1. Captura de sonido: El sonido se captura en forma de ondas.
2. Segmentación en frames: El audio se fragmenta en segmentos cortos, llamados frames.
3. Suavización de la transición: Se crea una transición suave entre cada muestra.
4. Transformada de Fourier de corto plazo (STFT): Se aplica una STFT a cada ventana para obtener una representación en frecuencia.
5. Mapeo a escala Mel: El resultado de la STFT se mapea en la escala Mel, lineal para frecuencias menores a 1 kHz y logarítmica para frecuencias superiores.
6. Cálculo de la potencia espectral: Se eleva al cuadrado la transformada de Fourier para obtener la potencia espectral de cada ventana.
7. Generación de la representación visual: Se generan coeficientes de energía para cada ventana y se crea una representación visual que muestra la intensidad de la frecuencia en la escala Mel a lo largo del tiempo.

Entorno de simulación: CoppeliaSim

CoppeliaSim, anteriormente conocido como V-REP, es un simulador de robots utilizado en la industria, la educación y la investigación (Rohmer, Singh, y Freese, 2013). Originalmente fue desarrollado dentro del departamento de I+D de Toshiba y actualmente está siendo desarrollado y mantenido activamente por Coppelia Robotics AG, una pequeña empresa ubicada en Zúrich, Suiza. En la figura 2 se observa la pantalla principal del entorno de simulación de CoppeliaSim.

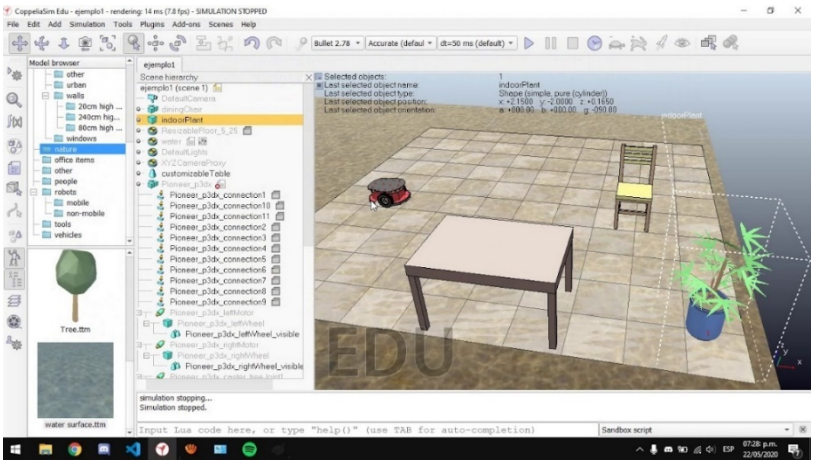


Figura 2. Pantalla principal del simulador de CoppeliaSim.

Está construido sobre una arquitectura de control distribuida que tiene scripts de Python y Lua, o complementos de C/C++ que actúan como controladores síncronos individuales. Los controladores asíncronos adicionales pueden ejecutarse en otro proceso, hilo o máquina a través de varias soluciones de middleware con lenguajes de programación como C/C++, Python, Java y Matlab.

CoppeliaSim utiliza un motor cinemático para cálculos cinemáticos directos e inversos, y varias bibliotecas de simulación física para realizar simulaciones de cuerpos rígidos. Los modelos y las escenas se construyen ensamblando varios objetos como mallas, articulaciones, sensores, nubes de puntos, etc., en una estructura

jerárquica. Las funciones adicionales, proporcionadas por complementos, incluyen: planificación de movimiento, visión artificial y procesamiento de imágenes, detección de colisiones, cálculo de distancia mínima, interfaces gráficas de usuario personalizadas y visualización de datos (por ejemplo, a través de gráficos). Los principales campos de aplicación de CoppeliaSim son la investigación robótica (Jiménez et al., 2020) y la educación (Camargo et al., 2021).

Implementación

Con la finalidad de poder llevar a cabo de manera satisfactoria el sistema propuesto se realizaron cuatro diferentes módulos. Estos módulos son los que se enumeran a continuación:

1. Modulo de procesamiento de LSM.
2. Modulo de procesamiento de Voz.
3. Módulo de simulación.
4. Integración de modular.

Módulo de procesamiento de LSM

Este módulo se encarga del procesamiento y análisis de los símbolos del LSM en tiempo real, de modo que el robot pueda identificarlas y convertirlas en instrucciones. Se comienza este módulo con el desarrollo del sistema que nos permitirá realizar el reconocimiento de imagen. Para lo cual, principalmente, se hace uso de la librería MediaPipe. Mediante esta librería se realiza, entre otras cosas, la detección y la extracción de gestos de la mano, siguiendo la secuencia que se describe a continuación:

- Conversión de los fotogramas de video de formato BGR a RGB.
- Detección de las manos con MediaPipe.
- Identificación de puntos clave de la mano.
- Extracción de coordenadas de puntos clave de la mano.
- Identificación de dedos levantados.
- Cálculo de distancias entre los puntos clave de la mano.

Se comienza con la conversión de los fotogramas de video de formato BGR a RGB, debido a que se requiere la información de RGB para el procesamiento de la información para poder detectar de manera eficiente las manos mediante la función de “HandProcessing” que posee Mediapipe. Una vez detectada la mano del usuario se realiza un mapeado de la misma para obtener los puntos de interés dentro de la mano, identificándolos y al mismo tiempo asignando cada uno de ellos a variables para que puedan ser utilizados por el módulo desarrollado para este proyecto.

Los puntos clave de la mano se encuentran principalmente dentro de la palma y en 3 puntos de los dedos que son las articulaciones y la punta de cada uno de los dedos. Dando un total de 21 puntos de interés que quedan registrados y servirán para la extracción de coordenadas de puntos clave de la mano. Una vez conseguidas y almacenadas las coordenadas de los puntos de interés será la fuente principal de información por la que conseguiremos conocer que postura del LSM se está realizando

Con estas coordenadas se almacenan en matrices para su posterior utilización ya que se deberán de cumplir para cada comando que busquemos, en nuestro caso nos basaremos principalmente en la punta de los dedos para la seña correspondiente, entonces, cuando las coordenadas que nos entregue el *frame* actual coincidan con la matriz de coordenadas propia del comando solicitado, este ejecutará la orden.

Para garantizar que la seña que estamos realizando solamente corresponda al comando que deseamos debemos marcar limites dentro de las distancias de los puntos clave de la mano, para esto debemos señalar distancias mínimas y máximas para que la coincidencia necesaria de la seña dada con la seña correspondiente al comando solicitado sea alta evitando así un desperfecto en la comunicación

Para este trabajo se han definido 6 comandos a interpretar, cada uno con una tarea asignada dentro del hogar. La forma en la que el robot ha de dar interpretación a dichos comandos depende de la modalidad de control que esté siendo atendida; en el caso de voz, el comando será simplemente la palabra con el nombre

del comando, mientras que, en imagen, cada comando estará directamente relacionado con el LSM, lo cual además resulta idóneo para personas con alguna discapacidad del habla. En la Tabla 1, se presentan las descripciones de las relaciones entre comandos, tareas y letras del LSM.

Tabla 1. *Relación de comandos de voz y símbolos LSM*

Comando de voz	Tarea a realizar	Letra LSM
Entrada	Abrir la puerta de la entrada principal	I
Estufa	Apagar la estufa ubicada en la cocina	E
Patio	Abrir la puerta del patio trasero	P
Televisor	Apagar el televisor ubicado en la sala	T
Foco	Apagar la luz de la lampara de la recamara principal	F
Ropa	Recoger la ropa sucia del baño y llevarla al patio	R

Módulo de procesamiento de Voz

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

Al igual que el módulo de señas, el módulo de reconocimiento de voz se puede dividir en diferentes etapas o componentes más pequeños que faciliten el proceso asociado al tratamiento de señales acústicas. En la Figura 3, se presentan dichos componentes. Este módulo de reconocimiento se realiza mediante un modelado acústico de palabras a nivel fonema, ya que así se modela a partir del espectrograma o espectro de Mel.

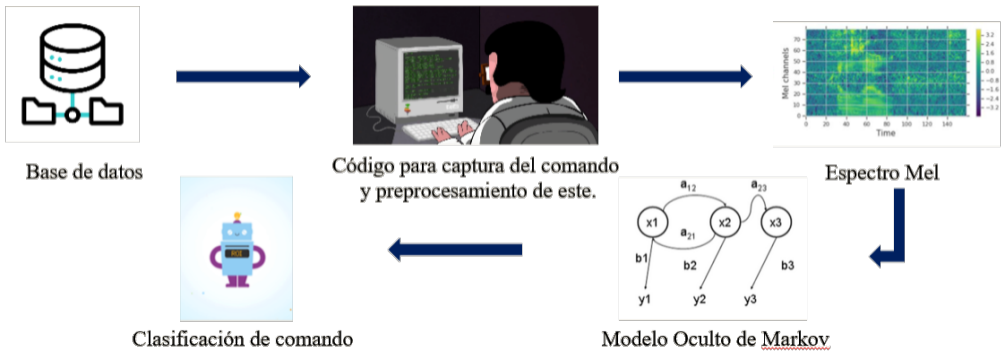


Figura 3. Secuencia de módulo de reconocimiento de voz.

De acuerdo con el esquema de desarrollo, como primer paso es necesario construir un corpus de entrenamiento, formado a su vez, por un corpus textual y un corpus oral. Dicho corpus de entrenamiento, es utilizado para la construcción y entrenamiento supervisado de los modelos acústicos, los cuales son representados mediante los Modelos Ocultos de Markov.

Por otro lado, se debe de generar tanto el modelo del lenguaje como el diccionario fonético a partir del corpus textual. En ese caso es relativamente simple ya que solo se usaron 6 palabras. Una vez construidos los modelos acústicos, el modelo del lenguaje y el diccionario fonético, los tres elementos son utilizados por el algoritmo de búsqueda para estimar la palabra dada una muestra de voz. El algoritmo de búsqueda compara una señal acústica de voz con los patrones de los modelos acústicos y como resultado genera una secuencia

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

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

Módulo de simulación

El fin de este proyecto va enfocado al control multimodal de un robot que en este caso será un Pioneer 3-DX, para el trabajo y modelado del funcionamiento del robot lo llevaremos primero a un entorno virtual en el que se le podrá integrar la parte codificada con la que actuará el robot y se mantendrá bajo un espacio controlado y virtual donde para esto utilizaremos el entorno de CoppeliaSim. CoppeliaSim nos brindará las facilidades del uso de un robot comercial utilizable en varios campos y al cual también nos permitirá ver su comportamiento bajo el tratamiento con código para la realización de distintas tareas como las que se presentan en la Tabla 1. Estas tareas se le serán asignadas para poder cumplir nuestro valor social agregado dentro del proyecto, el apoyo a personas mayores, con capacidades diferentes o en entornos poco favorables para su desempeño.

Para documentar los métodos aplicados en el diseño e implementación de este proyecto, se ha decidido dividir el informe en dos componentes principales. El primero se enfoca en el diseño del escenario de simulación, mientras que el segundo abarca la programación y validación del funcionamiento. De esta manera, se explicará detalladamente el método aplicado.

La generación del escenario de simulación se realizó mediante un proceso que requirió de investigación y acerca de los posibles escenarios a diseñar, así como de los algoritmos de control de un robot móvil para planificar una ruta entre un punto inicial hasta la meta deseada.

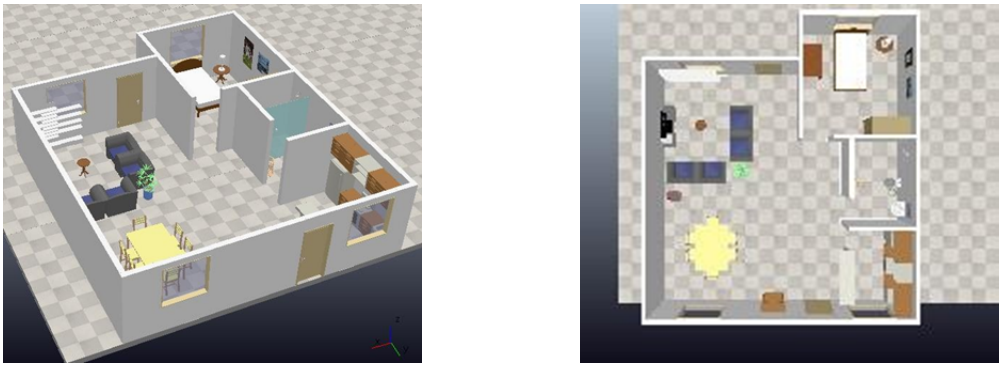
En primer lugar, para su diseño, se utilizó el software simulador de robótica CoppeliaSim. Este simulador permite al usuario utilizar elementos y modelos específicos del software, como paredes, sillas, mesas, sofás, etc. El proceso comenzó con la búsqueda y análisis de modelos 3D de casas y planos. Esto permitió crear un boceto del escenario deseado para el proyecto, incluyendo medidas preliminares que se utilizarían más adelante para la adecuación del entorno. Posteriormente, realizando una revisión de los modelos 3D disponibles en el simulador de CoppeliaSim, y estableciendo aquellos que serían utilizados, se realizó una búsqueda de los modelos restantes a partir del boceto ideado. Esta búsqueda se realizó en plataformas de dominio público para el acceso a modelos 3D, como lo son GrabCAD y Thingeverse.

Finalmente, la importación de modelos en conjunto con los modelos propios de CoppeliaSim, y la reasignación de posiciones mediante coordenadas, permiten obtener el escenario de simulación final. Cabe destacar que cada uno de los modelos u objetos añadidos deben ser modificados para que cuenten con las características de medibles, detectables y colisionables y, de esta manera, interactúen con el robot generando un entorno más realista.

Como resultado del diseño y construcción del escenario de simulación (Ver Figura 4), se cuenta con una estructura actual de vivienda que abarca aproximadamente 75.4 metros cuadrados. Este espacio comprende una habitación de 3.5 por 4 metros, un baño de 2 por 3 metros, una cocina de 2.5 por 3 metros y una estancia que combina sala y comedor de 4.8 por 8.5 metros. Estas dimensiones también incluyen el espacio ocupado por las paredes internas de la vivienda.

El escenario se ha diseñado con la estructura general de una casa, que incluye paredes, puertas, ventanas y espacios claramente definidos y distribuidos. Además, ahora cuenta con elementos específicos en la sala, comedor, baño y cocina para cada uno de los seis comandos de interpretación: “entrada", “estufa”, "patio", “televisor”, "foco" y "ropa". De esta manera, el robot tiene una acción asociada para cada área de la casa, permitiendo una interacción más específica y controlada en el entorno doméstico. Para que el robot se pueda desplazar en este escenario se implementó el algoritmo A*, el cual permite encontrar la trayectoria más corta desde un punto de inicio dado hacia un punto final predefinido, utilizando valores heurísticos para determinar

la ruta más eficiente hacia el objetivo, considerando el costo necesario para desplazarse entre los nodos de un grafo.



a b

Figura 4. Dos vistas diferentes del escenario construido.

Integración Modular

Para realizar la integración de los módulos se planteó la implementación de considerar una ponderación lineal a partir de cada una de las respuestas de los módulos de voz y de señas. Esta ponderación se realizó mediante la combinación lineal de las salidas de ambos módulos. El objetivo es determinar cuál de ellos tiene prioridad en caso de que cada uno de ellos entregue un comando de salida diferente. Por lo cual se procederá a explicar como se realizó dicha combinación lineal.

Desde una perspectiva puramente matemática proporcionada por el algebra lineal, se concibe una combinación lineal de la siguiente manera: *"Sean $u_1, \ldots, u_n$, elementos del espacio vectorial U. Se dice que v en U es combinación lineal de estos vectores si existen escalares $c_1, \ldots, c_n$ en K tales que: $v = c_1u_1 + \ldots + c_nu_n$"* .

En otras palabras, una combinación lineal de dos o más vectores se define como el vector resultante de la suma de esos vectores, cada uno multiplicado por un escalar correspondiente. En el contexto de un plano, cualquier vector puede ser generado mediante una combinación lineal de dos vectores no colineales. Este concepto también se extiende a escalares, donde una combinación lineal de escalares implica un escalar multiplicado por otro escalar denominado coeficiente.

En este proyecto, se emplea una combinación lineal de escalares para cuantificar los porcentajes de identificación del sistema de imagen y del sistema de voz. A partir de los porcentajes de identificación proporcionados por ambos sistemas, se generan dos combinaciones lineales según las ecuaciones (1) y (2). Aquí, **A** representa el porcentaje de identificación del sistema de voz, **B** el del sistema de imagen, y α es un coeficiente fijo que facilita las combinaciones.

$$C_1 = \alpha A + (1 - \alpha)B \quad (1)$$

$$C_2 = \alpha B + (1 - \alpha)A \quad (2)$$

Estas combinaciones se utilizan para la toma de decisiones sobre las instrucciones a ejecutar, basándose en la combinación de mayor valor. Este valor depende de los porcentajes de identificación y del coeficiente α. Aunque α normalmente tiene un valor fijo de 0.5, en este caso se ajustó a 0.4 para ponderar la mayor precisión del sistema de reconocimiento de imagen en comparación con el de voz.

Una vez determinado el comando reconocido y por consecuencia la acción a realizar se requiere de ejecutar el control sobre el robot. Para ello, se utilizó el algoritmo A* es una técnica de búsqueda informada que busca

el camino más corto desde un estado inicial hasta un estado objetivo utilizando una heurística óptima. Para su ejecución, se necesitan la posición inicial del agente, el punto de destino y las coordenadas de los obstáculos presentes en el entorno. Este algoritmo combina la búsqueda de coste uniforme con una heurística que guía la búsqueda hacia el objetivo de manera eficiente. En este caso, la heurística utilizada corresponde a la distancia euclidiana entre un punto A y un punto B (A* reference).

Es posible usar el algoritmo A* ya que, en este tipo de problemas, como el descrito por este proyecto, se tiene información sobre el estado inicial, el estado final (meta) y un conjunto de reglas que permiten desplazamientos en un grafo formado por nodos. A medida que el agente o robot se mueve a través del grafo, siguiendo estas reglas, eventualmente alcanzará el nodo correspondiente al estado final. Esto concluye la búsqueda y proporciona el conjunto de desplazamientos necesarios para llevar al agente de un punto a otro. El algoritmo A* basa su funcionamiento en la siguiente ecuación:

$$f(n) = h(n) + g(n) \tag{3}$$

Donde:

f(n) es el costo total estimado para alcanzar el objetivo pasando por el nodo n.
g(n) es el costo real del camino desde el nodo inicial hasta el nodo n.
h(n) es el costo heurístico estimado desde el nodo n hasta el nodo objetivo.

Finalmente, se implementa un sistema de control proporcional para el permitir al robot seguir la ruta calculada mediante el algoritmo de planificación de trayectorias A*. Comencemos por definir a un sistema dinámico como una entidad que recibe acciones externas o variables de entrada y responde a estas acciones a través de las denominadas variables de salida. Estos sistemas requieren un elemento de control de proceso, siendo el controlador PID uno de los más simples y frecuentemente efectivos. El controlador PID intenta corregir el error entre una variable de proceso medida y el punto de ajuste deseado calculando la diferencia y aplicando una acción correctiva para ajustar el proceso en consecuencia. Este tipo de controlador opera a través de tres parámetros: Proporcional (P), Integral (I) y Derivada (D), que pueden ponderarse o ajustarse para modificar su efecto en el proceso.

Este tipo de controladores son útiles en la programación del control de la velocidad de las ruedas del robot. En este proyecto, se emplea un controlador tipo P, que utiliza únicamente ganancias proporcionales. El método de retroalimentación mide el error de posición y orientación, es decir, la diferencia entre la posición y orientación actuales y las deseadas. Basándose en dos ganancias proporcionales, el controlador calcula las velocidades lineales (v) y angular (ω) requeridas para el robot. Posteriormente, a través de las ecuaciones 4 y 5, se determinan las velocidades de las ruedas del robot. En estas ecuaciones, r representa el radio de las ruedas del robot y L la distancia entre estas.

$$u_r = \frac{v}{r} + \frac{L\omega}{2r} \tag{4}$$

$$u_l = \frac{v}{r} - \frac{L\omega}{2r} \tag{5}$$

Resultados

En las siguientes secciones se presentan los resultados obtenidos a partir de las diferentes pruebas realizadas a los módulos implementados en el desarrollo de este trabajo de investigación.

Sistema de reconocimiento de imagen

Tras la programación y validación, primero del sistema de reconocimiento de imagen de manera individual, y después del proyecto en conjunto, se obtuvo un resultado satisfactorio en cuanto a identificación. Dado que el código utiliza técnicas de umbralización, se estableció un valor de umbral del 90% para afirmar que un comando ha sido identificado correctamente. Con este umbral, el porcentaje de identificación obtenido fue más que aceptable, lo que justifica dar mayor relevancia a este sistema en comparación con el sistema de reconocimiento de voz.

a b

Figura 5. Identificación de dos comandos diferentes: a) 'Televisor', y b) 'Patio'

Se realizaron entre 30 pruebas con los 6 comandos y sus respectivas letras del Lenguaje de Señas Mexicano (LSM), obteniendo una clasificación correcta en prácticamente todas las pruebas, con una tasa de reconocimiento del 94.5%. Las pocas excepciones se debieron principalmente a dos factores: 1) errores del usuario al realizar las señas y 2) el cambio de iluminación en la captura de la imagen. En la Figura 5, se pueden observar dos capturas del reconocimiento de dos de los seis símbolos del alfabeto de LSM los cuales se relacionan con la acción de ir a la encender la televisión Figura 5.a. Mientras que en la Figura 5.b representa el momento en el cual se reconoce la letra "P" de LSM que se asoció a la acción de ir al patio.

Sistema de reconocimiento de voz

Por otro lado, el sistema de reconocimiento de voz presentó resultados menos favorables. Si bien fue capaz de identificar correctamente al menos 4 de las 6 clases, tuvo problemas con las clases 'patio' y 'televisor', identificándolas incorrectamente en otras clases. Además, hubo casos en que las predicciones fueron erróneas para muestras de audio específicas de las otras 4 clases. Los resultados de esta área pueden observarse en el video de presentación de resultados. En este caso se obtuvo una tasa de reconocimiento de 68.5%. Esta tasa tan baja se debió, principalmente a ruido presente durante la captura de audio al momento de realizar las pruebas del sistema de reconocimiento de la voz. También, influyó el hecho de tener pocas muestras de cada palabra en el corpus lo cual no permitió al modelo estadístico inferir de manera adecuada.

Debido a estos problemas, se decidió dar menor ponderación al sistema de reconocimiento de voz en las combinaciones lineales, ya que su fiabilidad es variable. Tras un análisis, se concluyó que este problema podría estar relacionado con la base de datos, ya sea por el entorno en el que fue generada o por sus características de creación y transferencia. Para mejorar los resultados en esta área, se recomienda generar una nueva base de datos en un entorno controlado y con un conjunto diverso de personas, para lograr un entrenamiento más robusto del sistema.

Planificación de trayectorias y movimiento del robot

Finalmente, esta área de programación y validación del proyecto la cual generó los resultados más satisfactorios. Se realizaron pruebas de repetibilidad al algoritmo A* para la planificación de trayectorias con el fin de evaluar si el algoritmo era capaz de producir consistentemente la misma rutina ante los mismos parámetros de entrada. Se llevaron a cabo un total de 150 pruebas, es decir, 25 para cada comando, obteniendo un porcentaje de repetibilidad del 100%. En otras palabras, el algoritmo siempre generó la misma rutina cuando los parámetros de entrada permanecían invariables. En cuanto al movimiento del robot, no se

presentaron problemas durante las pruebas. El robot cumplió con sus desplazamientos, tiempos de espera y ajustes de velocidades sin ningún contratiempo.

En la Figura 6, se presentan seis capturas de la trayectoria ejecutada una vez que el sistema reconoce el comando de ir al patio. La trayectoria de ejecutas desde el punto de reposo del robot hasta la puerta que daría el acceso al patio trasero de la casa. Una vez terminada la tarea asignada el robot regresa a su punto de reposo inicial, esto si no existe alguna otra tarea a llevar a cabo.

Figura 6. Robot cumpliendo la rutina del comando 'Patio', siendo a) punto inicial y f) punto final pasando.

Conclusiones

En este proyecto, se logró replicar el desplazamiento de un robot Pioneer en CoppeliaSim, obedeciendo órdenes de voz y del lenguaje de señas mexicano hacia destinos particulares como televisión, lámpara, jardín, entrada, cocina y armario. El robot mostró una actuación sólida en cuanto a su capacidad de navegación, combinando de manera eficiente la detección de órdenes en audio y en imagen. La comunicación entre los sistemas de reconocimiento y el control del robot fue satisfactoria, demostrando la factibilidad de emplear inputs multimodales en el manejo de robots.

Los retos clave fueron la exactitud en la identificación de señas y la comprensión de órdenes verbales en ambientes ruidosos. Tras haber tenido unos resultados exitosos, cabe en la posibilidad mejorar el sistema a través de ajustar los algoritmos de reconocimiento y planificación de rutas. En líneas generales, este proyecto muestra las posibilidades de los robots controlados mediante comandos de voz y gestos en hogares, con beneficios para así poder ayudar en casa y mejorar la calidad de vida.

Bibliografía/Referencias

Kulyukin, V., Gharpure, C., Nicholson, J., y Osborne, G. (2006). Robot-assisted way finding for the visually impaired in structured indoor environments. Autonomous Robots, 21(1), p. 29-41.

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

INEGI Censo población 2020. (2020). Población. Discapacidad. Cuentame.inegi.org.mx. Recuperado en 15 de julio de 2024, de: http://www.cuentame.inegi.org.mx/poblacion/discapacidad.aspx?tema=P.

Villa-Angulo, R. y Hidalgo-Silva, H. (2005). A wearable neural interface for real time translation of Spanish deaf sign language to voice and writing. Journal of Applied Research and Technology, 3(3), pp. 169-186.

Nickel, K. y Stiefelhagen, R. (2007). Visual recognition of pointing gestures for human-robot interaction. Image and Vision Computing, 25(12), p. 1875-1884.

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

Trujillo-Romero, F., y García Bautista, G. (2021). Reconocimiento de palabras de la Lengua de Señas Mexicana utilizando información RGB-D. ReCIBE, Revista electrónica De Computación, Informática, Biomédica Y Electrónica, 10(2), C2–23. https://doi.org/10.32870/recibe.v10i2.209.

Posada-Gomez, R., Sanchez-Medel, L. H., Hernandez, G. A., Martinez-Sibaja, A., Aguilar-Laserre, A. y Leija-Salas, L. (2007). A Hands Gesture System of Control for An Intelligent Wheelchair. En: 4th International Conference on Electrical and Electronics Engineering, pp. 68-71.

Trigo, T. R. y Pellegrino, S. R. M. (2010). An analysis of features for hand-gesture classification. En: Proceedings of International Conference on Systems, Signals and Image Processing, pp. 412-415.

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

Deshmukh, A. M. (2020). Comparison of Hidden Markov Model and Recurrent Neural Network in Automatic Speech Recognition. European Journal of Engineering and Technology Research, 5(8), 958–965. https://doi.org/10.24018/ejeng.2020.5.8.2077.

Khanna, S. (2021). Identifying Privacy Vulnerabilities in Key Stages of Computer Vision, Natural Language Processing, and Voice Processing Systems. International Journal of Business Intelligence and Big Data Analytics, 4(1), 1–11. Retrieved from https://research.tensorgate.org/index.php/IJBIBDA/article/view/66.

Deuerlein, C., Langer, M., Seßner, J., Heß, P., & Franke, J. (2021). Human-robot-interaction using cloud-based speech recognition systems. Procedia CIRP, 97, 130–135. https://doi.org/10.1016/j.procir.2020.05.214.

Zemke, D. M. V., Tang, J., Raab, C., & Kim, J. (2020). How To Build a Better Robot . . . for Quick-Service Restaurants. Journal of Hospitality & Tourism Research, 44(8), 1235-1269. https://doi.org/10.1177/1096348020946383.

Bakouri, M., Alsehaimi, M., Ismail, H. F., Alshareef, K., Ganoun, A., Alqahtani, A., & Alharbi, Y. (2022). Steering a Robotic Wheelchair Based on Voice Recognition System Using Convolutional Neural Networks. Electronics, 11(1), 168. https://doi.org/10.3390/electronics11010168.

Calvo, M.T. (2004). Diccionario Español - Lengua de Señas Mexicana (DIELSEME): estudio introductorio. Dirección de Educación Especial: México.

Serafín de Fleischmann, M., González Pérez, R. (2011). Manos con voz, Diccionario de Lenguaje de Señas Mexicana. Primera edición, Libre Acceso, A.C., ISBN 978-607-9134-0I-3

Código estándar Mex-Esp. (2021). ISO 639 — Language codes. ISO. Retrieved 14 Mayo 2021, from https://www.iso.org/iso-639-language-codes.html.

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F.,Chang, C., Yong, M., Lee, J., Chang, W., Hua, W., Georg, M. & Grundmann, M. (2019). MediaPipe: A Framework for Building Perception Pipelines. arXiv preprint https://doi.org/10.48550/arXiv.1906.08172

MediaPipe. (2020). MediaPipe Hands. https://google.github.io/mediapipe/solutions/hands.

Rohmer, Eric, Surya P. N. Singh, and Marc Freese. (2013). V-REP: A Versatile and Scalable Robot Simulation Framework. Pp. 1321–26 in 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. Tokyo: IEEE.

Jiménez, Andrés C., John P. Anzola, Vicente García-Díaz, Rubén González Crespo, and Liping Zhao. (2020). PyDSLRep: A Domain-Specific Language for Robotic Simulation in V-Rep. PLOS ONE 15(7):e0235271. doi: 10.1371/journal.pone.0235271.

Camargo, Caio, José Gonçalves, Miguel Á. Conde, Francisco J. Rodríguez-Sedano, Paulo Costa, and Francisco J. García-Peñalvo. (2021). Systematic Literature Review of Realistic Simulators Applied in Educational Robotics Context. Sensors 21(12):4031. doi: 10.3390/s21124031.