Simposio CEA de Robótica, Bioingeniería, Visión Artificial y Automática Marina 2025

RBVM
Simposio CEA de Robótica, Bioingeniería, Visión por Computador y Automática Marina

Desarrollo de un sistema para manejo del robot ScorBot-ER 4U con MATLAB a través de una interfaz visual y órdenes de voz

Fernández-Marín, A., García-Mañas, F., Rodríguez, F.*, Becerra-Terón, A.

Universidad de Almería, Departamento de Informática, CIESOL, ceiA3, 04120 Almería, España.

Resumen

Este trabajo presenta el desarrollo de un sistema para el manejo de un brazo robótico ScorBot-ER 4U mediante una interfaz en MATLAB que permite el reconocimiento de órdenes de voz emitidas por un operario, de tal forma que el robot efectúa una serie de movimientos en función de las órdenes recibidas. La comunicación entre el controlador del robot y MATLAB se realiza mediante el ScorBot Toolbox, que contiene un conjunto de comandos con los que se pueden definir movimientos específicos para las articulaciones del robot. El reconocimiento de voz se realiza con el Audio Toolbox de MATLAB, que permite procesar las señales de audio provenientes de un micrófono y hacer uso del comando `speech2text` para transcribir a texto las órdenes recibidas. En concreto, se utiliza la API (*Application Programming Interface*) de Google Cloud Speech-to-Text como servicio de transcripción de voz en la nube. Los resultados obtenidos muestran que las órdenes se identifican sin errores en un 94 % de los ensayos realizados, junto con una velocidad de transcripción media de 4,5 segundos.

Palabras clave: Brazo robótico, Robótica de manipulación, Interfaz de usuario, Reconocimiento de voz, Comunicación

Development of a system to manage the ScorBot-ER 4U robot with MATLAB through a visual interface and voice commands

Abstract

This work presents the development of a system for the management of a ScorBot-ER 4U robotic arm by means of a MATLAB interface that allows the recognition of voice commands issued by an operator, so that the robot performs a series of movements according to the commands received. Communication between the robot controller and MATLAB is performed by means of the ScorBot Toolbox, which contains a set of code commands to define specific movements for the robot joints. Speech recognition is performed with the Audio Toolbox of MATLAB, which allows processing the audio signals coming from a microphone and making use of the command `speech2text` to transcribe the received voice commands to text. Specifically, the Google Cloud Speech-to-Text API (Application Programming Interface) is used as a speech transcription service in the cloud. The results obtained show that voice commands are identified without errors in 94 % of the trials performed, along with an average transcription speed of 4.5 seconds.

Keywords: Robotic arm, Robotic manipulators, User interface, Speech recognition, Communication

1. Introducción

La robótica ha experimentado un crecimiento significativo en las últimas décadas, transformando la manera en la que se llevan a cabo procesos automatizados en distintos ámbitos, desde la industria hasta la investigación. En este contexto se pueden destacar los robots manipuladores, que ofrecen una amplia gama de aplicaciones gracias a su flexibilidad, precisión y capacidad de realizar tareas complejas en entornos controlados. Sin embargo, la interacción con estos robots y su manejo eficiente siguen siendo desafíos importantes, especialmente en términos de accesibilidad y facilidad de uso para operadores no especializados (Urrea and Kern, 2025).

En los últimos años, los avances en el reconocimiento de voz han permitido el desarrollo de sistemas más eficientes y accesibles para la interacción con máquinas. Las tecnologías de procesamiento de lenguaje natural y los algoritmos de aprendizaje automático han permitido una mejora significativa en la precisión y versatilidad de los sistemas de reconocimiento de voz, permitiendo su aplicación en una amplia gama de dispo-

*Autor para correspondencia: frrodrig@ual.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

sitivos y entornos. En el ámbito de la robótica, el control por órdenes de voz ofrece una interfaz natural, que reduce la dependencia de dispositivos físicos y mejora la interacción en tiempo real entre un usuario y un sistema robótico (Rohith and Asha, 2024).

Además de sus aplicaciones industriales, el manejo de robots mediante órdenes de voz y una interfaz visual tiene un gran potencial en el ámbito educativo. Los robots educativos se utilizan ampliamente para enseñar conceptos fundamentales de robótica, programación y control en entornos académicos. La integración de tecnologías avanzadas como el reconocimiento de voz en estos entornos puede hacer que la enseñanza sea más atractiva para los estudiantes, facilitando la comprensión de conceptos técnicos de una manera aplicada e interactiva. Al trabajar con sistemas que combinan una interfaz visual y órdenes de voz, los estudiantes pueden desarrollar habilidades en programación y control en tiempo real, lo que les ofrece un acercamiento a la tendencias actuales en robótica.

En este trabajo se ha empleado el robot educativo ScorBot-ER 4U (Intelitek, 2025) para implementar un sistema de manejo que permita efectuar movimientos según órdenes de voz proporcionadas por un usuario. El ScorBot-ER 4U fue elegido por su demostrada versatilidad para proyectos educativos y de investigación, entre los que se pueden destacar el seguimiento de trayectorias con redes neuronales (Kumar and Chand, 2015), la teleoperación y control remoto del robot (Franklin Salazar et al., 2019; Salazar et al., 2020), el desarrollo de sistemas de realimentación de fuerza en su pinza (Chand et al., 2016), el uso de visión artificial para reconocimiento de objetos (Bernal and Aguilar, 2012), e incluso el diseño optimizado de un ScorBot-ER 4U para reducir la masa e inercia y mejorar la eficiencia energética del brazo robótico (Lakshmi Srinivas et al., 2023).

La principal contribución de este trabajo es el desarrollo de una interfaz visual en MATLAB combinada con una herramienta de reconocimiento de voz con el objetivo de dotar al ScorBot-ER 4U con la capacidad de seguir órdenes indicadas por la voz de un operario, para que el robot efectúe tareas o movimientos concretos.

2. Materiales y métodos

2.1. Robot ScorBot-ER 4U

El robot ScorBot-ER 4U (Figura 1) es un brazo robótico diseñado por Intelitek, Inc. (EE.UU.) para uso educacional. Se trata de un manipulador de cinco grados de libertad (GDL), que cuenta con seis servomotores de corriente continua (con sus respectivos codificadores de posición) y con transmisiones mecánicas mediante correas y engranajes. Las articulaciones son de tipo rotacional con un GDL, mientas que la muñeca que posee dos GDL (giro y elevación) además de contar con una pinza que puede abrirse o cerrarse. En la Tabla 1 se muestra un resumen del rango de movimiento del robot por cada eje motorizado. Tiene un peso de 10,8 kg, su velocidad máxima de trayectoria es de 700 mm/s y su radio máximo de operación es de 610 mm.

El ScorBot-ER 4U cuenta con una unidad controladora (con un microcontrolador NEC V853 RISC 32-bit) con la que se comunica mediante un cable de 62 pines tipo-D. Este controlador puede conectarse a un ordenador mediante cable USB. Además, dispone de ocho entradas y salidas digitales, cuatro entradas analógicas y dos salidas analógicas. A la unidad controladora también se le puede conectar una botonera de enseñanza, para manejar el robot de forma manual.

Figura 1: Robot ScorBot-ER 4U (Intelitek, Inc.).

Tabla 1: Características de los ejes del ScorBot-ER 4U (Intelitek, Inc.).

Eje	Rango (°)	Velocidad efectiva (°/s)
1. Torsión base	310	20
2. Rotación hombro	158	26,3
3. Rotación codo	260	26,3
4. Elevación muñeca	260	83
5. Giro muñeca	±570	106

En cuanto al software para programación y operación del robot, Intelitek proporciona el programa ScorBase para su ejecución en el sistema operativo Windows (Microsoft, EE.UU.). ScorBase forma parte de Robocell®, un paquete de software gráfico interactivo de Intelitek que permite la simulación del robot y de dispositivos auxiliares que formen parte de su celda de trabajo.

2.2. ScorBot Toolbox para MATLAB

ScorBot Toolbox (Wick et al., 2011) es una herramienta que permite conectar el robot ScorBot-ER 4U con MATLAB (The MathWorks, EE.UU.), ofreciendo al usuario expandir las posibilidades de programación y manejo del robot más allá de las ofrecidas por el entorno Robocell® y ScorBase (Kutzer et al., 2023). En ese sentido, ScorBot Toolbox permite a los usuarios controlar el robot y obtener información de sus sensores. Además, incluye una documentación de ayuda extensa y de herramientas de visualización para facilitar su uso. Entre sus características principales se encuentran, por ejemplo, el control del robot mediante comandos de puntos de referencia en el espacio de las articulaciones, y la creación de representaciones tridimensionales simuladas del robot. La instalación de la herramienta es sencilla y compatible con Windows.

La Figura 2 muestra la ejecución del ScorBot Toolbox desde MATLAB. A continuación, se incluye un ejemplo de los comandos que se pueden emplear desde MATLAB para que el robot efectúe movimientos concretos (Wick et al., 2011):

```
ScorInit; %Inicia comunicación MATLAB con robot.
ScorHome; %Articulaciones en posición de inicio.
ScorSetSpeed(80); %Velocidad movimiento 80 %.
ScorDeltaCartMove([0 0 -10 0 0]); %Mueve 10 cm abajo.
ScorCartMove([40 0 30 -90 0]); %Mueve 40 cm adelante.
ScorSetGripper(-1); %Abre pinza.
ScorSetGripper(0); %Cierra pinza.
```

donde el comando `ScorCartMove ([X Y Z P R])` se utiliza para un movimiento en línea recta con coordenadas absolutas (desde la referencia en la base del robot), mientras que `ScorDeltaCartMove ([X Y Z P R])` se utiliza para un movimiento en línea recta con coordenadas relativas o incrementales, siendo `X Y Z` las coordenadas cartesianas del desplazamiento y `P R` las posiciones de elevación y giro de la muñeca, respectivamente.

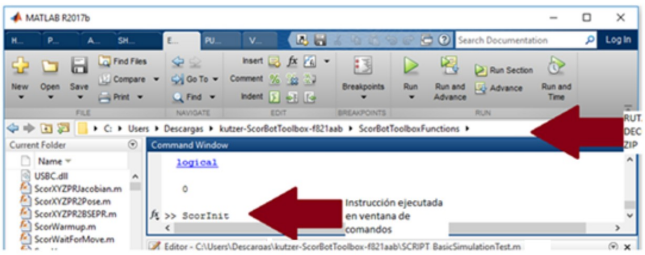

Figura 2: Ejecución del ScorBot Toolbox desde MATLAB.

2.3. *Herramientas para reconocimiento de voz*

Para el reconocimiento de voz, en este trabajo se empleó el Audio Toolbox de MATLAB que permite, por un lado, grabar, procesar y manipular las señales de audio provenientes de un micrófono y, por otro lado, hacer uso del comando `speech2text` para transcribir a texto las órdenes recibidas por micrófono. En concreto, el comando `speech2text` se utilizó junto con la API (*Application Programming Interface*) de Google Cloud Speech-to-Text (Google, 2025), como servicio de transcripción de voz en la nube. Esta API es conocida por su alta precisión y su capacidad para reconocer el habla a partir de diversas fuentes de audio, incluidas aquellas en entornos ruidosos.

Para integrar la API en MATLAB haciendo uso del comando `speech2text` se necesita una conexión a Internet y una suscripción activa al servicio de voz a texto de Google, que fue elegido para este trabajo por su velocidad de respuesta en la transcripción tras compararlo con otras APIs comerciales como las de IBM, Microsoft y Amazon. En el código MATLAB se tiene que indicar la API que se va a utilizar, incluyendo también el idioma con el que se va a trabajar para aclarar que las órdenes que se reciben por el micrófono y que se recibirán en el servicio de transcripción sean del idioma del usuario que interactúa con el robot. A continuación, se muestra un fragmento de código MATLAB para tal fin:

```
speechObject=speechClient('Google','languageCode','es-ES');
speechObject.Options;
tableOut=speech2text(speechObject,y,fs)
tableOut=table2array(tableOut);
command=tableOut(1,1);
execute_voice_command(command,hObject,eventdata,handles);
```

Cada vez que se recibe un texto tras la transcripción de audio, en MATLAB se realiza la identificación de ese texto para evaluar de qué orden se trata, de tal forma que se le pueda indicar al robot el movimiento que debe realizar (véase la Figura 3). Por tanto, previamente, en MATLAB se almacenan como variables de cadena de caracteres el conjunto de órdenes admitidas por el robot, por ejemplo: 'abrir pinza','cerrar pinza', 'posición uno', etc. Cuando se recibe una transcripción de audio a texto, ésta se compara con las órdenes almacenadas y se identifica con cuál de ellas coincide. Posteriormente, con el uso del ScorBot Toolbox, se envía la orden de movimiento concreta al robot y el usuario puede verificar que ha obedecido a su orden.

Figura 3: Interpretación de órdenes de voz.

3. Resultados y discusión

3.1. *Diseño de interfaz visual para manejo del robot*

Para la interacción del usuario con el robot, se implementó una interfaz gráfica utilizando GUIDE (*Graphical User Interface Development Environment*) de MATLAB. Como se aprecia en la Figura 4, la interfaz cuenta con una pantalla de control de movimientos de cada eje del robot, permitiendo a un usuario mover manualmente al robot tanto en la realidad como en simulación. También contiene dos indicadores bajo los botones de «Simulación» y «Real» que cambian de color según el modo de actuación con el robot. Hay que destacar que la interfaz contiene el código basado en el ScorBot Toolbox, tal y como se ha descrito en la sección 2.2, lo que permite "traducir" la interacción del usuario con los botones de la interfaz a un movimiento efectuado por el robot.

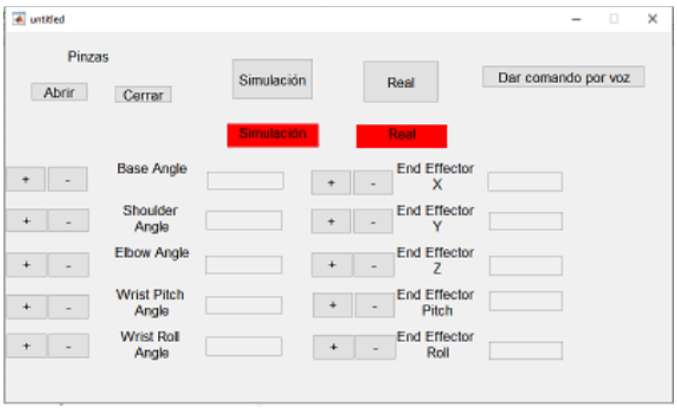

Figura 4: Interfaz para manejo del robot. Pantalla de control manual.

En la Figura 5 se muestra el modo simulación, que cuenta con una pantalla adicional en la que se representa el robot para visualizar la correspondencia de los movimientos introducidos por

el usuario con el movimiento realizado por el robot. El usuario debe introducir en los campos en blanco los valores numéricos para posiciones de cada eje o para movimientos lineales y con los botones + y − puede aumentar o disminuir dichos valores en incrementos de 0,1 mm o 0,1 rad, según corresponda.

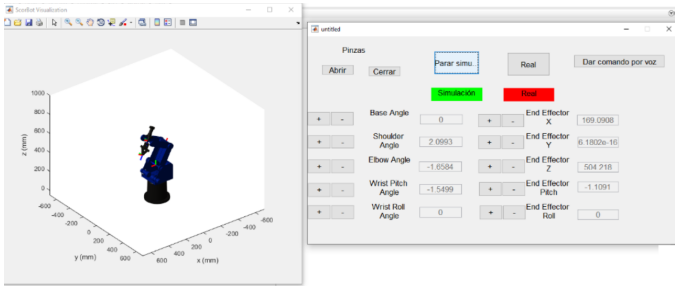

Figura 5: Interfaz para manejo del robot. Pantalla de simulación.

En cuanto a los colores de los indicadores de «Simulación» y «Real», su significado es el siguiente:

- Rojo: no iniciado. En simulación, significa que ninguna simulación se está llevando a cabo. En el caso real, significa que no se ha realizado la conexión con el robot.
- Verde: en ambos casos, indica que el robot se encuentra listo y a la espera de una orden (manual o por voz) para realizar alguna tarea.
- Amarillo: en ambos casos, indica que el robot se encuentra realizando alguna tarea. Permite al usuario saber si el robot se encuentra en movimiento o no. El robot no ejecutará ninguna otra tarea hasta que no termine la tarea que se encuentre en ejecución.

3.2. *Ejemplo de movimiento del robot con comandos del ScorBot Toolbox*

Para comprobar el funcionamiento del robot con el uso del Scorbot Toolbox, se efectuaron, mediante comandos en MATLAB, todos los movimientos necesarios con el robot para operaciones de *pick and place*, consistentes en coger y colocar un conjunto de piezas en distintas posiciones para completar una estructura como la que se muestra en la Figura 6.

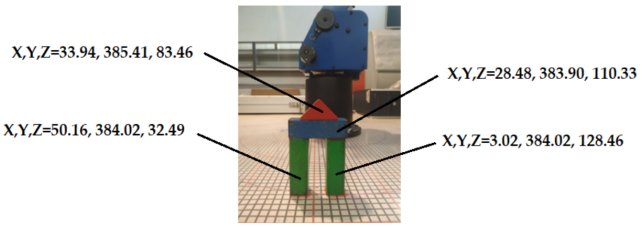

Figura 6: Estructura realizada por el robot mediante comandos del ScorBot Toolbox ejecutados desde MATLAB.

Es importante destacar que las posiciones de las piezas tanto en el estado inicial (distribuidas en el espacio de trabajo del robot) como en el estado final (estructura completa) fueron definidas en el código MATLAB mediante coordenadas absolutas, usando el comando `ScorSetXYZPR`. La Figura 7 muestra algunos de los movimientos realizados de forma satisfactoria con el robot tras haber recibido las órdenes desde MATLAB.

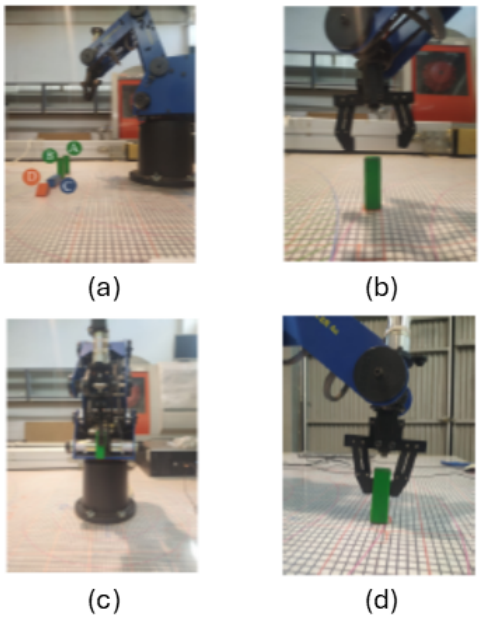

Figura 7: Movimientos realizados por el robot mediante comandos del ScorBot Toolbox ejecutados desde MATLAB.

3.3. *Movimiento del robot con reconocimiento de voz*

Para reproducir la estructura indicada en la sección 3.2 pero mediante reconocimiento de voz, como se observa en la Figura 4, en la parte superior derecha de la interfaz gráfica se encuentra el botón «Dar comando por voz». Cuando un usuario lo pulsa, se realiza una grabación de audio de cinco segundos y por pantalla se le advierte del inicio y final de ese tiempo, para que pueda emitir la orden de voz deseada. Una vez tomada la muestra de audio, se procede a su transcripción con los métodos comentados en la sección 2.3. Tanto el tiempo de grabación como las advertencias en pantalla se pueden modificar en el código embebido en la interfaz, tal y como se muestra a continuación:

```
recObj=audiorecorder(Fs,nBits,nChannels,ID);
disp('Comience a hablar');
recordblocking(recObj,5);
disp('Grabación finalizada');
y=getaudiodata(recObj);
filename='audio.wav';
```

Posteriormente, una vez realizada la transcripción del audio a texto y almacenado su resultado en la variable `voice_command`, se ejecuta el siguiente código que permite buscar la orden en un directorio de órdenes predefinidas para el robot (véase la sección 2.3):

```
list_files=dir('*.m');
for k1 = 1:length(list_files)
 if strcmp(voice_command,list_files(k1).name)==1
  run(list_files(k1).name(handles.sim_obj));
  break;
 end
end
```

En cuanto al código para que se ejecuten los movimientos del robot asociados a cada orden predefinida, en la Figura 8 se explica un fragmento del conjunto de funciones `if, elseif, else` que permite comprobar que la transcripción recibida de la API de Google coincide con la orden predefinida que se ha identificado y, por tanto, el robot puede proceder a realizar el movimiento correspondiente, empleando comandos del ScorBot Toolbox.

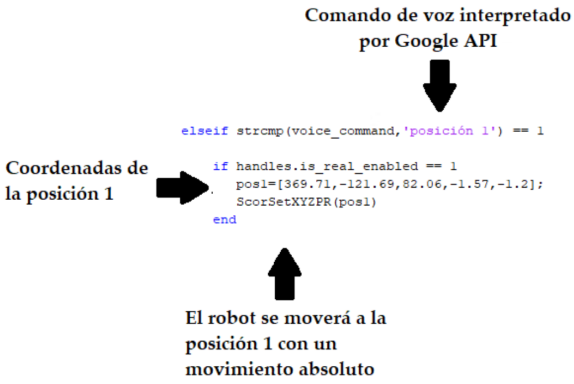

Figura 8: Explicación del código de interpretación de órdenes para efectuar movimientos.

El código se ha diseñado para que cada orden de voz resulte en la ejecución de un sólo movimiento del robot. En cualquier caso, el código implementado es fácilmente adaptable para programar que con una orden de voz se ejecuten varios movimientos del robot. Parte de los resultados de este trabajo en un ensayo real con el robot recibiendo órdenes de voz se pueden visualizar en el siguiente enlace: `https://youtu.be/ijs4hpmuOQw`, que fueron obtenidos en el marco del trabajo fin de grado desarrollado por el primer autor de este artículo.

En lo que respecta a la evaluación del rendimiento del sistema de reconocimiento de voz implementado, cabe destacar que el uso de la API de Google fue una decisión acertada ya que al transcribir muestras de audio a palabras se pudo comprobar que es capaz de hacer distinción de palabras con distinta acentuación y entonación, sin presentar problemas para los comandos de órdenes predefinidas, tales como 'Abrir/Cerrar pinza' o 'Posición 1' (hasta 18 órdenes fueron definidas). En ese sentido, la Figura 9 muestra el número de aciertos en ensayos para comprobar la correcta transcripción de órdenes de voz por la API de Google. Se puede observar que se consigue interpretar correctamente casi todas las órdenes salvo en el caso de 'Posición 10', que en algunas ocasiones se interpretaba como 'Posición Díaz'. Asimismo, se comprobó que la máxima distancia admisible entre el usuario que emite la orden y el micrófono (integrado en el ordenador en el que se ejecuta MATLAB) debe ser de hasta un metro. Si dicha distancia aumenta, se pierde calidad en el audio grabado y la transcripción de las órdenes no será correcta.

En cuanto a la evaluación de la velocidad de transcripción ofrecida por la API de Google y teniendo en cuenta que se requiere de una conexión a Internet, la Figura 10 muestra el tiempo transcurrido entre la emisión de la orden y la ejecución del movimiento por parte del robot. Como se puede observar, los tiempos de respuesta oscilan en torno a los 4,5 segundos, que en los ensayos reales se ha percibido como aceptable por el usuario, sin ser una demora excesiva para aplicaciones académicas.

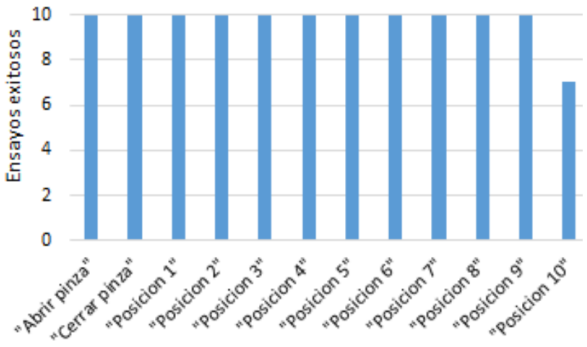

Figura 9: Aciertos de la API de Google para transcripción de órdenes por voz.

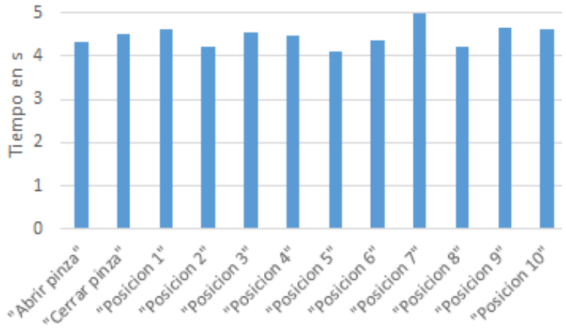

Figura 10: Tiempos de la API de Google para transcripción de órdenes por voz.

4. Conclusiones y trabajos futuros

En este trabajo se ha implementado una interfaz que consigue que el robot ScorBot-ER 4U sea capaz de moverse en su espacio de trabajo siguiendo órdenes de voz dadas por un usuario. A la vista de los resultados obtenidos, se puede concluir que se ha conseguido combinar de forma satisfactoria un conjunto de herramientas en MATLAB para tal fin. En este sentido, además del ScorBot Toolbox para comunicación y manejo del robot ScorBot-ER 4U, el uso de `speech2text` junto con la API externa de Google ha permitido transcribir órdenes de voz provenientes de un usuario a comandos de movimiento con gran porcetanje de acierto y una velocidad aceptable, considerando que se requiere de una conexión a Internet para el servicio de transcripción.

En lo que respecta a trabajos futuros, la transcripción de las órdenes podría realizarse con herramientas de código abierto y que permitan un procesamiento en local, sin necesidad de enviar las grabaciones de voz a un servicio en la nube. Por otro lado, se puede incrementar la complejidad de las órdenes a cumplir por el robot, transcribiendo mensajes más extensos sin necesidad de haber definido órdenes predeterminadas. Por ejemplo, un usuario podría ordenar que el robot se mueva de forma lineal a una posición con unas coordenadas determinadas. Para ello, será necesario evaluar algunas de las técnicas de reconocimiento de voz que han surgido en los últimos años y que muestran un rendimiento prometedor.

Agradecimientos

Al grupo de investigación «Automática, Robótica y Mecatrónica» (TEP-197) de la Universidad de Almería.

Referencias

Bernal, A. C., Aguilar, G. M., 2012. Vision system via USB for object recognition and manipulation with Scorbot-ER 4U. International Journal of Computer Applications 56 (18), 10–15.

Chand, P., Lal, S., Kumar, R., 2016. Development of a force feedback system for a SCORBOT robotic arm gripper. ARPN Journal of Engineering and Applied Sciences 11 (23), 13766–13774.

Franklin Salazar, L., Buele, J., Velasteguí, H. J., Soria, A., Núñez, E. E. T., Benítez, C. S., Orejuela, G., 2019. Teleoperation and remote monitoring of a SCORBOT ER-4U robotic arm in an academic environment. International Journal of Innovative Technology and Exploring Engineering 8 (6), 40–5.

Google, 2025. Google Cloud Speech-to-Text. Accessed on April 11, 2025. URL: `https://cloud.google.com/speech-to-text`

Intelitek, 2025. ScorBot-ER 4u educational robot. Accessed on April 22, 2025. URL: `https://intelitek.com/scorbot-er-4u-educational-robot/`

Kumar, R. R., Chand, P., 2015. Inverse kinematics solution for trajectory tracking using artificial neural networks for SCORBOT ER-4u. In: 2015 6th International Conference on Automation, Robotics and Applications (ICARA). pp. 364–369.

Kutzer, M. D. M., Donnal, J. S., Wick, C. E., 2023. The ScorBot Toolbox for MATLAB: an open-source hardware interaction and simulation library for the Intelitek SCORBOT-ER 4u educational robot. In: 2023 ASEE Annual Conference & Exposition.

Lakshmi Srinivas, G., Jaya Aadityaa, G., Pratap Singh, S., Javed, A., 2023. Energy efficiency enhancement of SCORBOT ER-4U manipulator using topology optimization method. Mechanics Based Design of Structures and Machines 51 (8), 4635–4654.

Rohith, S., Asha, C., 2024. Systematic review of mobile robot control by using human-robot interaction. In: 2024 International Conference on Futuristic Technologies in Control Systems & Renewable Energy (ICFCR). pp. 1–6.

Salazar, F. W., Buele, J., Llugsa-Hinojosa, J., Domínguez, A. B., Pérez, G. E., Ibazeta, A. F., 2020. Sistema prototipo para la teleoperación de un robot ScorBot ER-4U en ambientes peligrosos. Revista Ibérica de Sistemas e Tecnologias de Informação (E29), 312–324.

Urrea, C., Kern, J., 2025. Recent advances and challenges in industrial robotics: a systematic review of technological trends and emerging applications. Processes 13 (3), 832.

Wick, C., Esposito, J., Knowles, K., 2011. The MATLAB toolbox for the Intelitek Scorbot (MTIS): an open source educational robotics development library. In: American Society of Engineering Education's Annual Conference. pp. 1189—1288.