

APLICACIÓN DE UN ROBOT COLABORATIVO DE DOS BRAZOS PARA RESOLVER EL CUBO DE RUBIK

Garazi Fernández, Asier Herrán, Aitziber Mancisidor, César Pérez, Itziar Cabanes

gfernandez051@ikasle.ehu.eus, aherran014@ikasle.ehu.eus, aitziber.mancisidor@ehu.eus, itziar.cabanes@ehu.eus

Departamento de Ingeniería de Sistemas y Automática
Escuela de Ingeniería de Bilbao, UPV/EHU

Resumen

Con objeto de entretener y acercar al usuario al mundo de la robótica, este trabajo presenta una célula robotizada para la resolución de cubos de Rubik. El sistema está formado por el robot colaborativo YuMi de dos brazos y una cámara de visión artificial. Gracias a esta cámara, el sistema es capaz de detectar la posición y la configuración del cubo de Rubik, además de identificar si se trata de un cubo de 2x2 o de 3x3. Una vez detectado el cubo, el robot utiliza los dos brazos para resolver el cubo con el mínimo número de giros y movimientos posibles. Adicionalmente, con el fin de aumentar la comunicación entre el sistema y el usuario, se le ha añadido una pantalla táctil y un módulo de sonido.

Palabras clave: Robótica colaborativa, visión artificial, Matlab, RobotStudio, cubo de Rubik.

1 INTRODUCCIÓN

En las últimas décadas los robots han sido ampliamente utilizados en diferentes ámbitos de la industria, como es en aplicaciones de soldadura, ensamblaje o empaquetamiento. Estos, resultan muy útiles para operaciones repetitivas y producción en grandes lotes. Sin embargo, por motivos de seguridad, no es posible la directa interacción entre el operario y los robots. Este es el motivo por el cual en la cuarta revolución industrial se desarrollan los robots colaborativos o cobots [12].

Los robots colaborativos traen integrados sensores de fuerza, presión o tacto que posibilitan la colaboración entre los robots y sus usuarios, eliminando las barreras físicas que los han separado durante muchos años [6]. Estos sensores permiten medir y controlar la fuerza y la velocidad, lo que garantiza que no superen los umbrales definidos en caso de producirse un contacto, sea intencionado o por accidente, manteniendo en todo momento la seguridad del usuario y de los componentes de su entorno.

Además, los cobots permiten reducir de forma significativa los problemas de espacio en su instalación (por la eliminación de barreras), mejorar la facilidad y rapidez en la programación y agilizar la adaptación ante cambios en la producción [2].

Uno de los sectores que más se beneficia de estos robots es el de la industria. A día de hoy los cobots son ampliamente utilizados en las líneas de producción, realizando diversas tareas desde ensamblado hasta clasificación de objetos, pasando por la fabricación [1, 11], entre otras.

No obstante, uno de los grandes beneficios de la robótica colaborativa ha sido la posibilidad de utilizar los robots fuera de la industria. Hoy en día se pueden encontrar robots colaborativos en el sector sanitario [4], en terapias de rehabilitación [13] o en el ámbito de la enseñanza [10, 7].

Otra de las grandes aplicaciones de estos robots es el entretenimiento. En el mercado existen una gran cantidad de robots destinados al entretenimiento. Entre ellos, los más populares son las mascotas robóticas las cuales poseen un gran rango de funcionalidades; desde las más simples que únicamente se mueven por la casa, hasta las más complejas, dotadas de inteligencia artificial y equipadas con cámaras que les permiten reconocer comandos de voz, esquivar obstáculos y hasta ayudar en las tareas del hogar [5].

En este trabajo, se presenta una aplicación de un robot colaborativo de dos brazos capaz de resolver los cubos de Rubik de 2x2 y 3x3. Con ello se pretende lograr una aplicación llamativa que genere interés, entretenga al usuario y atraiga al usuario al mundo de la robótica.

El resto del artículo se estructura de la siguiente manera. En la sección 2 se realiza una descripción del diseño de la aplicación. En la sección 3 se detalla la programación e implementación de la misma, seguida de la sección 4 donde se resumen los resultados. Finalmente, en la sección 5, se resumen las conclusiones y trabajos futuros.

2 DISEÑO DE LA APLICACIÓN

2.1 ESPECIFICACIÓN Y FUNCIONALIDAD

El cubo de Rubik es un mecanismo de ejes que permite que cada una de sus caras gire independientemente, de manera que los colores se mezclen. Para solucionar el rompecabezas, cada una de sus caras debe estar compuesta por un solo color. Completar dicho rompecabezas 3D requiere de una serie de habilidades tales como concentración, memoria, coordinación óculo-manual, matemáticas, buenas destrezas psicomotoras y agilidad mental, entre otras [3].

Con objeto de entretener y lograr la resolución del cubo de Rubik de forma automática, se plantea el diseño de una aplicación robótica, de manera que permita al usuario interactuar y retar al robot sin peligro alguno.

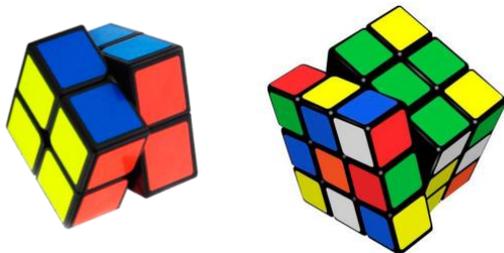


Figura 1: Cubos de Rubik de 2x2 y 3x3

Para ello, se selecciona el robot colaborativo de dos brazos Yumi, del fabricante ABB, con el que los usuarios puedan interactuar, de manera que logre atraer su atención. Una vez que el usuario deposite uno de los cubos de Rubik en cualquier punto arbitrario del escenario, el robot lo recogerá e identificará el tipo de cubo que es, considerando las opciones de 2x2 y de 3x3 (ver Figura 1). Una vez reconocido, realizará fotos de las 6 caras del cubo para conocer la configuración inicial, es decir saber con precisión la ubicación de las piezas de las diferentes caras que componen el cubo. Tras procesar dichas imágenes, el robot conocerá la posición y el color de cada una de las piezas móviles que componen el cubo y será necesario aplicar un algoritmo que calcule la mejor estrategia a seguir. Esto es, el mínimo número de giros y movimientos que se deberán realizar en el cubo para completar las 6 caras del mismo color respectivamente, y cómo debe realizarse (dirección y sentido). Finalmente, con toda la información adquirida, el robot mediante sus dos brazos ejecutará los movimientos calculados para resolver el cubo.

Adicionalmente, con objeto de incrementar la interacción y empatía del robot con el usuario, la aplicación contará con voz y con una pantalla táctil para interactuar de forma sencilla con él.

En la Figura 2 se muestran los componentes hardware para llevar a cabo esta aplicación: 1) el robot colaborativo Yumi de dos brazos, del fabricante ABB, que realizará los movimientos y giros deseados en el cubo como lo haría una persona; 2) una cámara de visión artificial, UI 5584LE-C-HQ de Infaimon, para identificar dónde deposita el usuario el cubo, de qué tipo de cubo se trata (2x2 ó 3x3) y también se encargará de realizar las fotos de las diferentes caras para saber la configuración inicial del cubo; 3) un PC el cual permite procesar las imágenes y desarrollar la algoritmia que minimice los giros para la resolución del cubo de Rubik mediante el entorno Matlab.

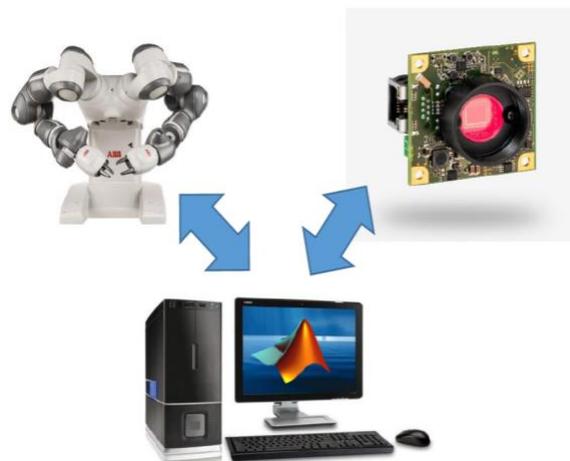


Figura 2: Relación de componentes

2.2 DISEÑO Y SELECCIÓN DE COMPONENTES

Para afrontar las tareas y funciones definidas en el sistema, es necesario realizar el diseño de varios componentes del escenario, los cuales se detallan en este apartado.

2.2.1 Componentes auxiliares

Para que el robot pueda localizar de forma autónoma el cubo, cogerlo, resolverlo, y depositarlo una vez resuelto, hace falta una superficie plana donde hacerlo. Debido a que hay más elementos en el escenario que forman parte de otras células de montaje, se ha optado por diseñar una mesa mediante la herramienta AutoCAD que tenga la facilidad de integrarse cuando se requiera. En dicha mesa el usuario depositará el cubo en un punto arbitrario para que el robot lo recoja y comience con su función.

La ubicación de la cámara es muy importante para esta aplicación. Concretamente, para el sistema de visión (cámara, placa base, lente y un ventilador), se ha realizado el diseño de una carcasa que además de

protegerla ayude a ubicarla en la célula de trabajo, permitiendo la orientación de la cámara en los ángulos que se desee sin que entorpezca el correcto funcionamiento del robot. Esta carcasa consta de dos componentes, tapa y base, las cuales han sido diseñadas mediante AutoCAD. Adicionalmente, también se ha diseñado un soporte que sujete de forma conjunta la pantalla táctil al robot. En la Figura 3 se muestra dicho componente.

Finalmente, puesto que es necesario que el robot sea capaz de recoger, girar y manipular con destreza el cubo de Rubik, se han diseñado unos dedos con mayor capacidad de apertura y mejor adecuación que los que poseía el Yumi (ver Figura 3).

Tras realizar el diseño, se han generado todos los componentes mediante una impresora 3D (Artillery X1 Sidewinder).

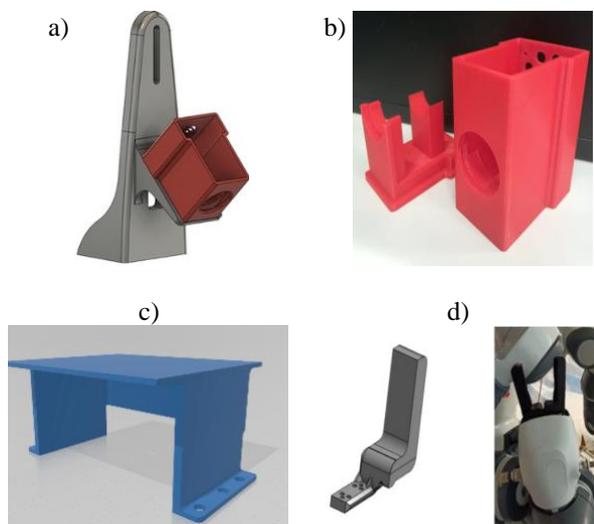


Figura 3: Componentes auxiliares diseñados:

a) soporte para la carcasa de la cámara y la pantalla táctil, b) carcasa donde se aloja la cámara, c) mesa donde se recoge y deposita el cubo, d) dedos de la pinza.

2.2.2 Componentes interacción

Debido a que uno de los objetivos principales es lograr la interacción usuario-robot, se ha instalado una pantalla táctil, TFT-LCD Monitor, mediante la cual se podrá interactuar con el robot, controlando tanto los movimientos del robot como el procesamiento de las imágenes.

Adicionalmente, para lograr una mayor relación con el robot se han integrado un módulo de sonido, el cual será utilizado por el Yumi para anunciar al usuario sus futuras acciones. El módulo de sonido consta de dos

altavoces, a los cuales se les envían los datos mediante Matlab.

En la Figura 4 se muestra el escenario completo, con todos los componentes detallados en esta sección.

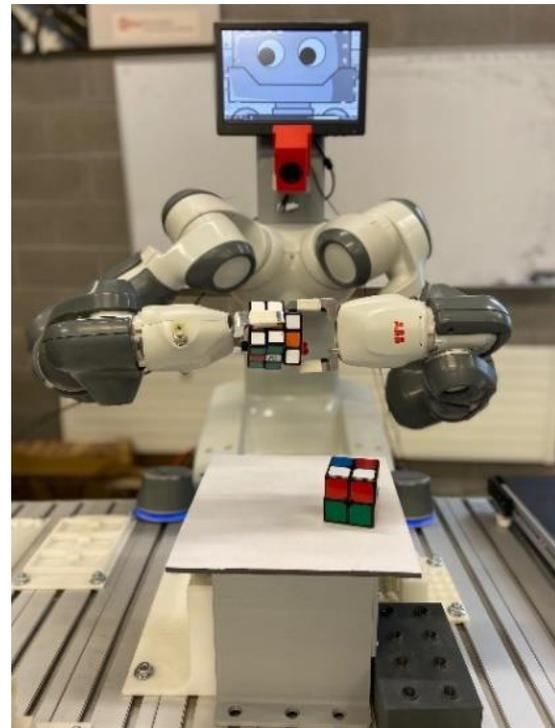


Figura 4. Escenario completo.

3 PROGRAMACIÓN

Para llevar a cabo la resolución del cubo de Rubik hace falta realizar una serie de instrucciones y códigos. Estos deben permitir tanto, obtener las imágenes y procesarlas, extrayendo de ellas la información necesaria para trasladarle al robot los giros a realizar.

3.1 PROCESAMIENTO DE IMÁGENES

Al detectar un cubo de Rubik en la mesa, el sistema, mediante la cámara de visión artificial y el software Matlab, efectúa la primera fotografía la cual se utiliza para localizar la posición exacta del cubo.

En este primer procesamiento de la imagen, se aplica una máscara de color con objetivo de aislar los colores del cubo y seguido se binariza la imagen. Posteriormente utilizando funciones como *strel*, *cornermetric* e *imregionalmax* se logra filtrar el ruido y obtener el contorno de cada uno de los objetos, para después mediante el código creado con el comando *regionprops* conseguir las posiciones de los puntos (ver Figura 5).

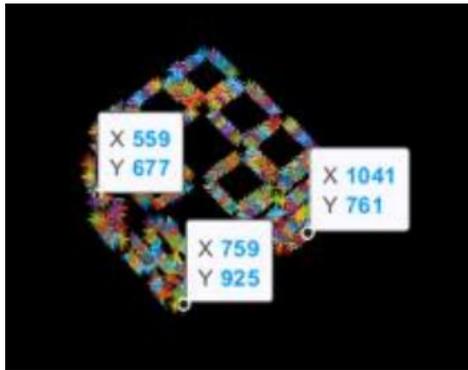


Figura 5: Procesamiento mediante *regionprops*

Por otro lado, una vez que el robot ha recogido el cubo, se realiza la toma de imágenes de las caras del mismo (ver Figura 6), siendo un total de 6 fotos, una por cada cara. Tras esta toma de imágenes del cubo, se realiza un procesamiento de los colores de cada cara que permita conocer la posición de cada una de las nueve piezas que las forman y, por ende, la forma en la que está mezclado el cubo. Para este procesamiento de color se ha utilizado la función *ColorThresholder*, que genera una máscara que muestra solo los colores seleccionados (ver Figura 6).

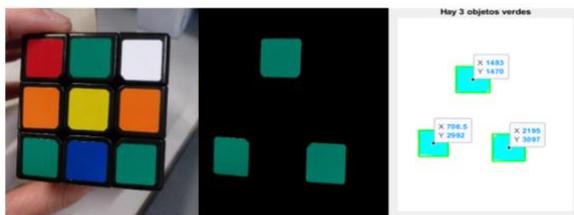


Figura 6: Procesamiento de imágenes

Una vez procesados todos los colores de la imagen, se genera la matriz *Cara* donde cada color se representa mediante un número siendo, Rojo = 1; Azul = 2, Naranja = 3; Verde = 4; Blanco = 5 y Amarillo = 6. Por tanto, la fotografía original de la cara se reduce a una matriz de dimensión 3x3. Por último, la información de cada *Cara* se traslada a un array multidimensional *R* que contiene toda la información del cubo mezclado. Dicho array está compuesto por seis matrices de 3x3, una por cara. Este array se enviará al algoritmo que se presenta en el siguiente apartado para calcular la resolución del cubo.

3.2 ALGORITMO RESOLUCIÓN CUBO DE RUBIK

Una vez identificadas las posiciones de los bloques de colores del cubo, es necesario desarrollar el algoritmo de resolución, con el que se obtienen los giros a realizar.

En este caso, se aborda la resolución del cubo mediante la estrategia de Joren Heit [9]. A partir de imágenes tomadas, se lograrán los movimientos para resolver el cubo mediante la función *Solve45*, asegurando una media de 31 movimientos para la resolución completa del cubo (siendo 45 su máximo). El algoritmo se basa en unas tablas de poda generadas automáticamente y formadas por más de un millón de entradas que le permiten elegir la mejor solución en función de la forma en la que esté mezclado el cubo.

La implementación y ejecución del algoritmo conlleva cuatro fases:

- 1) Orientación de los bordes.
- 2) Orientación de las esquinas y colocar los bordes de las caras derecha e izquierda.
- 3) Colocar el resto de bordes y esquinas.
- 4) Resolver el cubo.

Para cada una de dichas fases el algoritmo calcula mediante las tablas de poda el número mínimo de movimientos. Este número de movimientos mínimo se denomina *d*, y tras elegir un giro, se comprueba cómo varía. Si *d* disminuye, el movimiento es correcto, ya que se encuentra un movimiento más cerca de la siguiente fase. Cuando *d* sea igual a 0, el algoritmo pasa de fase. Este algoritmo devuelve un array de celdas en las que cada una hace referencia a un movimiento a realizar para resolver el cubo.

3.3 MOVIMIENTOS ROBOT

El robot posee dos brazos, que trabajarán de forma sincronizada a la hora de realizar movimientos. El brazo derecho se encarga de coger el cubo de la mesa y llevar el cubo hasta la posición de las 3 primeras fotos. El brazo izquierdo deberá coger el cubo en la posición de la tercera foto, y trasladarlo para realizar las 3 restantes.

Posteriormente, para que los brazos ejecuten los movimientos necesarios para la resolución del cubo, es necesario enviar al robot colaborativo YuMi los movimientos codificados obtenidos. Para ello hace falta realizar la comunicación entre Matlab y el robot de forma que permita el intercambio de datos por ambas partes. En este caso se ha optado por una comunicación mediante protocolo TCP/IP.

Una vez establecida la comunicación, se procede al envío de los datos al robot. En primer lugar, es necesario contar los movimientos a enviar y guardar la información en una variable. El siguiente paso es enviar el número de movimientos al robot de forma que sepa la cantidad de caracteres que tiene que recibir y procesar. En caso de ser menos de 26 movimientos se realiza un único envío, mientras que si son más, se realizarán dos.

4 DISCUSIÓN Y RESULTADOS

Una vez configurado el escenario con la mejor ubicación de los componentes, y realizado el código de programación en el entorno Matlab y RobotStudio, se valida la solución.

Dado que uno de los objetivos principales era retar al robot para lograr que la resolución del cubo sea lo más rápida posible, se han realizado varias pruebas con los cubos de 2x2 y 3x3. De esta manera se han obtenido diferentes resultados de tiempos y cantidad de movimientos que se necesitan para resolver los distintos cubos.

Finalmente, se han realizado las pruebas de resolución del cubo de 2x2 y 3x3, ejecutando un total de 10 pruebas con cada uno de ellos. En la tabla 1 se resume el número de movimientos realizados (Mov) para resolver cada una de las 10 configuraciones de partida con cada cubo y el tiempo que tarda el robot en conseguirlo (T).

Tabla 1: Ensayos con cubos de Rubik de 2x2 y 3x3

Cubo de Rubik 2x2										
Mov	10	12	13	15	11	19	14	12	16	9
T (min)	2,3	3,16	3,4	3,5	2,5	4,4	3,2	3,16	4,1	2
Cubo de Rubik 3x3										
Mov	29	34	35	30	28	33	29	27	32	34
T (min)	7	8,1	8,4	7,2	6,7	8	7	6,5	7,7	8,1

Se puede concluir que el tiempo medio que necesita el robot para resolver el cubo de 2x2 es de 3,17 minutos y la media del número de movimientos es de 13,1. En el caso del cubo 3x3, el tiempo medio es de 7,47 minutos y la media de movimientos es de 31,1.

Como era de esperar, el robot tarda mucho menos tiempo y necesita menos movimientos para resolver el cubo de 2x2, puesto que este posee un menor número de combinaciones posibles.

5 CONCLUSIONES

En este trabajo se ha diseñado y construido una célula robótica de entretenimiento que permite resolver cubos de Rubik de 2x2 y 3x3 de forma automática.

El sistema propuesto está compuesto por el robot colaborativo YuMi de dos brazos, una cámara de visión artificial, un ordenador de procesamiento con Matlab y elementos auxiliares como la mesa, el

soporte de la cámara o dedos de la pinza del robot impresos con una impresora 3D.

Adicionalmente, con la intención de atraer a los usuarios, se le ha añadido una pantalla táctil y un sistema de sonido que aumenta la interacción entre el robot y el usuario.

Después de realizar varios ensayos experimentales, se puede concluir que el sistema propuesto es capaz de resolver el cubo de Rubik de 2x2 en un tiempo medio de 3,17 y el de 3x3 en 7,47 minutos.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el proyecto GIU19/045 de la UPV/EHU.

English summary

APPLICATION OF A TWO-ARMED COLLABORATIVE ROBOT TO SOLVE RUBIK'S CUBE

Abstract

In order to entertain and get the robotics closer to the user, this work presents a robotic cell for the resolution of Rubik's cubes. The system consists of the YuMi two-armed collaborative robot and an artificial vision camera. Thanks to this camera, the system is able to detect the position and configuration of the Rubik's cube, in addition to identifying whether it is a 2x2 or 3x3 cube. Once the cube is detected, the robot uses its both arms to solve the cube with the minimum number of turns and movements possible. Additionally, in order to increase communication between the system and the user, a touch screen and a sound module have been added.

Keywords: Collaborative robotics, computer vision, Matlab, RobotStudio, Rubik's cube.

Referencias

- [1] Gómez, E., & Andrés, D. Diseño e implementación de una celda colaborativa robotizada mediante robots móviles y humanoides para clasificación de objetos. Recuperado el 22 de junio de 2021, de Edu.ec
- [2] G. Du, M. Chen, C. Liu, B. Zhang, and P. Zhang, "Online Robot Teaching With Natural Human-Robot Interaction", IEEE Transactions on Industrial

Electronics, vol. 65, no. 12, pp. 9571–9581, Dec. 2018.

[3] Julita, J. R., & Viejo, V. A. (2015). Cubo de Rubik. XVIII Concurso de Trabajos Estudiantiles (EST 2015) - JAIIO 44.

[4] J.F. Avila-Tomás, M.A. Mayer-Pujadas, & V.J. Quesada-Varela (2020). La inteligencia artificial y sus aplicaciones en medicina I: introducción antecedentes a la IA y robótica. *Atención Primaria*, 52(10), 778-784.

[5] López Rodríguez, R., & Ospina Saldaña, J. (2019). Robot domótico con control remoto y cámaras, para pruebas de Mayordomo y Jardinero: Rob-Erto. *Letras ConCiencia Tecnológica*, (15), 13-17. Recuperado de <https://revistas.itc.edu.co/index.php/letras/article/view/155>

[6] Moctezuma Gutiérrez, S. G., Cruz Pazarán, A., Galicia Mejía, R., & Oliva Moreno, L. N. (2018). Desarrollo de plataforma para implementación de robots colaborativos. *Visión electrónica*, 12(1), 22–31. <https://doi.org/10.14483/22484728.13308>

[7] Moreno, I., Muñoz, L., Serracín, J. R., Quintero, J., Patiño, K. P., & Quiel, J. (2012). La robótica educativa, una herramienta para la enseñanza-aprendizaje de las ciencias y las tecnologías. *Teoría de la Educación. Educación y Cultura en la Sociedad de la Información*, 13(2), 74-90.

[9] Rubik's cube simulator and solver - file exchange - MATLAB central. (2011, octubre 23). Recuperado el 23 de junio de 2021, de Mathworks.com website: <https://es.mathworks.com/matlabcentral/fileexchange/31672-rubik-s-cube-simulator-and-solver>

[10] Salamanca, M. L. P., Lombana, N. B., & Holguín, W. J. P. (2010). Uso de la robótica educativa como herramienta en los procesos de enseñanza. *Ingeniería Investigación y Desarrollo: I2+ D*, 10(1), 15-23.

[11] Salimbeni, S., & Mamani, D. (2020). Marco de referencia para la incorporación de Cobots en líneas de manufactura. *Podium*, 38(38), 159–180.

[12] Schwab, K. (2016). The fourth industrial revolution. Ed. Crown Business.

[13] Tucan, D. (2021). Ankle Rehabilitation of Stroke Survivors Using Kuka LBR Iiwa. In *New Trends in Medical and Service Robotics* (pp. 29–36). Springer International Publishing.



© 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).