

MANIPULACIÓN DE OBJETOS DIRIGIDA A LA ASISTENCIA DE PERSONAS CON MOVILIDAD REDUCIDA

Francisco J. Naranjo-Campos¹, Ainhoa de Matías-Martínez¹, Juan G Victores¹,
Nicolás Álvarez², Almudena Alcaide², Carlos Balaguer¹

¹RoboticsLab, Departamento de Ingeniería de Sistemas y Automática,
Universidad Carlos III de Madrid

²Departamento de Tecnología Accesible e I+D, Dirección de Accesibilidad Universal e Innovación,
Inserta Innovación, Fundación Once

{frnaranj, amatias}@pa.uc3m.es, jcgvicto@ing.uc3m.es,
{nalvarez, aalcaide}@fundaciononce.es, balaguer@ing.uc3m.es

Resumen

La robótica continúa en avance y se integra cada vez más en diversos ámbitos, entre los cuales se encuentra la robótica asistencial. En este artículo, la robótica asistencial se emplea en ayudar a las personas con movilidad reducida, un colectivo en aumento. El objetivo es incrementar la independencia de dicho colectivo y mejorar la calidad de la asistencia en hogares y centros especializados. En este artículo se ha implementado con éxito la tarea de manipulación para la asistencia usando el robot TIAGo, que consiste en determinar la posición de unos contenedores con marcadores ArUco por medio de visión por computador para luego cogerlos. Para ello, se han diseñado los contenedores y se ha desarrollado el software de visión y manipulación necesario. Finalmente, se han realizado y evaluado pruebas en las que se ha variado las posiciones del robot y del contenedor, midiendo el tiempo de ejecución y el desplazamiento del robot, con resultados satisfactorios.

Palabras clave: Robot asistencial, movilidad reducida, ArUco, TIAGo, manipulación

1. INTRODUCCIÓN

La robótica está evolucionando de tal manera que va integrándose poco a poco en diversos espacios de la sociedad. Este artículo tratará el ámbito de los robots asistenciales, que proporcionan servicios y apoyo a públicos muy diferentes, centrándose en la asistencia de personas con movilidad reducida.

Actualmente, garantizar una vida independiente a las personas con movilidad reducida es una cuestión social relevante. Por definición, las personas con movilidad reducida son aquellas que tienen limitadas sus capacidades motrices, ya sea de forma temporal o permanente [1]. Este colectivo está formado por quienes tienen algún tipo de lesión o afección y por las personas de avanzada edad con enfermedades crónicas asociadas al envejecimiento.

Además, en las próximas dos décadas se estima que la población envejecida se triplicará, lo que conllevará un aumento considerable de este colectivo de personas [2].

De esta manera, se pretende impulsar el uso de robots asistenciales para llevar a cabo tareas de asistencia a personas en situaciones de dependencia, las cuales han sido tradicionalmente desarrolladas en los hogares por las familias [8], pudiendo así liberarles o facilitarles estas tareas o, en el caso de centros especializados, poder dar una mejor asistencia a un mayor número de personas.

El fin del proyecto con el robot TIAGo es la asistencia a personas con movilidad reducida, el cual manipula y transporta distintos elementos previamente definidos, navegando por el entorno desde un punto inicial hasta un punto final donde se encontraría el usuario. En concreto, el presente artículo se centra en la manipulación de los objetos con el fin de detectarlos y cogerlos.

Este documento comienza con una introducción del marco en el que se desarrolla la parte del proyecto presentada en el artículo, describiendo al robot empleado TIAGo y los recursos de software empleados. Seguidamente, se explican los módulos desarrollados en el proyecto, el control de cada módulo y cómo se desarrolla la rutina de la manipulación. Por último, se exponen los resultados de las pruebas realizadas y las conclusiones al respecto.

2. DESCRIPCIÓN DE LOS RECURSOS EMPLEADOS

Con el objetivo de proporcionar asistencia a personas con movilidad reducida, se han implementado funcionalidades en el robot TIAGo [7], mostrado en la Figura 1, para transportar ciertos objetos que se han diseñado para el proyecto.

En concreto se trata de unos contenedores o cajas los cuales tienen un asa adecuada a la pinza de TIAGo que permite un agarre óptimo. Además, para la parte de visión el asa consta de un identificador que será captado



Figura 1: Robot móvil manipulador TIAGo.

por las cámaras de TIAGo y por el cual se le indica específicamente qué contenedor debe manipular.

Como recursos se ha empleado el *framework* ROS (*Robot Operating System*) para el desarrollo de software de robot junto con el lenguaje de programación C++. Además, gracias a la herramienta de visualización RVIZ y el simulador Gazebo, es posible tener una recreación en simulación del robot TIAGo. Dicha simulación permite hacer pruebas iniciales con el fin de ser capaces de solucionar posibles problemas con antelación a la ejecución en el robot real.

2.1. TIAGo

El robot TIAGo [7] fue desarrollado por la compañía española PAL Robotics, que lanzó el prototipo de TIAGo en el año 2015 [6]. Este es un robot manipulador móvil destinado a la investigación que combina percepción, manipulación, navegación y capacidades de reconocimiento del habla para la interacción con humanos. Existen diversos modelos de TIAGo con variaciones en el tipo de movimiento, el efector final, la interfaz y el número de brazos, entre otros. Estas características permiten que el robot se adecúe tanto a entornos de investigación como a entornos socio-sanitarios en los que se puedan realizar tareas de asistencia a los usuarios.

El modelo de TIAGo empleado posee una base móvil que le proporciona movimientos diferenciales y un único brazo, con cuatro grados de libertad, que incorpora una pinza como efector final con tres grados de libertad. Cuenta con una cámara RGBD en la cabeza y adicionalmente tiene una cámara RGB endoscópica instalada sobre la pinza. TIAGo dispone de un torso regulable con el que consigue una altura de entre 110 y 145 cm, lo cual permite ampliar el campo de visión y el rango de manipulación.

2.2. CÓDIGOS ArUco

Para la identificación de los diferentes contenedores a manipular, cada uno posee un identificador propio haciendo uso de los códigos ArUco (*Augmented Reality University of Cordoba*), que fueron desarrollados en el año 2014 [5]. Inicialmente se propuso emplear códigos QR, sin embargo, estos conllevan un mayor tiempo de cómputo y se necesita una mayor resolución para su identificación.

Por esta razón, finalmente se han empleado los códigos ArUco, que consisten en un amplio recuadro negro que

contiene una matriz binaria que determina un identificador numérico codificado, cuyo tamaño determina el número de bits a codificar [4]. La codificación se realiza de tal manera que pueden distinguirse las esquinas del código, lo que permite determinar su orientación. Los códigos ArUco empleados en el proyecto se corresponden con los identificadores 15, 25 y 35, que representarían tamaños diferentes de los contenedores.

2.3. ARQUITECTURA SOFTWARE

La implementación software se ha realizado dentro del ecosistema de ROS en el lenguaje C++, y se organiza según su aspecto funcional siguiendo la estructura que se observa en la Figura 2, que consiste en tres nodos:

- Navegación. Nodo con un servicio para comandar órdenes de movimiento a los controladores de la base.
- Visión. Nodo para recibir las imágenes de las cámaras, analizarlas y publicar la información sobre los marcadores ArUco detectados.
- Manipulación. Nodo que coordina el uso de manejadores de control del brazo, la pinza y la cabeza, junto con el servicio de navegación y la información de visión para realizar la tarea implementada de buscar un contenedor y manipularlo.

3. IMPLEMENTACIÓN

La tarea de manipulación implementada consiste en detectar el contenedor con el marcador ArUco y cogerlo. Esta rutina sigue el diagrama de flujo que se muestra en la Figura 3, en la que se observa que consiste en cinco procesos. Primeramente, se asegura que el brazo se encuentra en la posición de reposo y a continuación, se busca el contenedor con la cámara RGBD de TIAGo. Luego, en caso de ser necesario, se ajusta la distancia al contenedor con la base móvil, y seguidamente se centra la pinza con el ArUco, haciendo uso de la cámara de la pinza. De este modo, se ajusta la posición de la pinza con el contenedor y finalmente se coge. Los procesos de búsqueda de ArUco y de corrección de la posición de la base pueden realizarse hasta un máximo de tres repeticiones, para evitar un bucle infinito.

A continuación, se explica el desarrollo software compuesto por los elementos mostrados en la Figura 2. Estos son el nodo de visión y los manejadores de control de navegación y manipulación. Por último, se explica en qué consiste cada uno de los procesos de la rutina de manipulación implementada.

3.1. VISIÓN

Se ha creado un nodo de ROS que se mantiene en bucle captando información por ambas cámaras y tiene como objetivo guardar en un vector todos los ArUcos presentes en la imagen. Además, guarda las características de cada ArUco: su número identificador, la posición del centro en la imagen medida en píxeles y la de cada esquina. Las características obtenidas también se publican de modo visual marcándolas sobre los ArUcos en la imagen, como se muestra en la Figura 4.

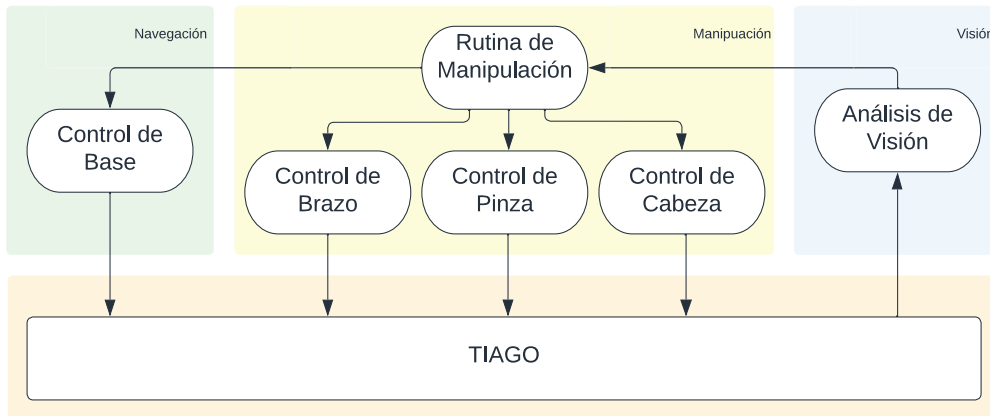


Figura 2: Estructura del software implementado.

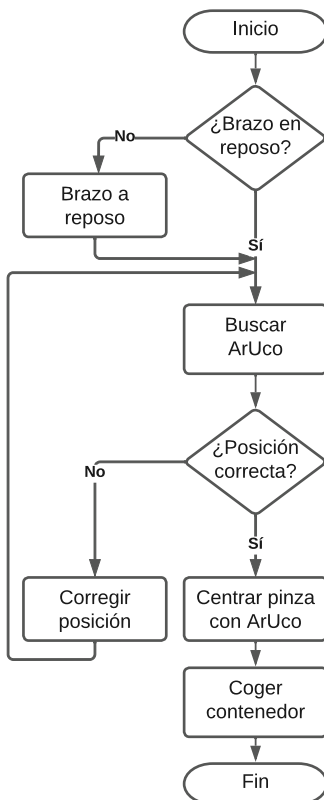


Figura 3: Diagrama de flujo de la tarea implementada de coger el contenedor.

Para emplear la detección de los ArUco en el proceso de manipulación, se ha creado un servicio de ROS en el que el usuario especifica el identificador deseado para obtener una respuesta con las características del ArUCO y su distancia 3D. El servicio buscará el identificador requerido en el vector y obtendrá sus características. Para obtener la distancia 3D, son necesarios los parámetros intrínsecos de la cámara RGBD de la cabeza, con los que se obtendrá la posición 3D del centroide del ArUco a la cabeza de TIAGO y se transforma para que tenga como

referencia la base del robot y poder comandar la pinza a un determinado punto.

Se ha realizado un estudio del rango de detección de ArUcos con la cámara de la cabeza. Con los marcadores empleados de 5cm, la distancia máxima a la que se detectan adecuadamente y de forma continua es a 100cm de distancia, con un rango de 105cm o aproximadamente 55°, mostrado en la Figura 5. Por otro lado, la distancia mínima a la que se pueden detectar es a 35cm y en un rango de 40cm.

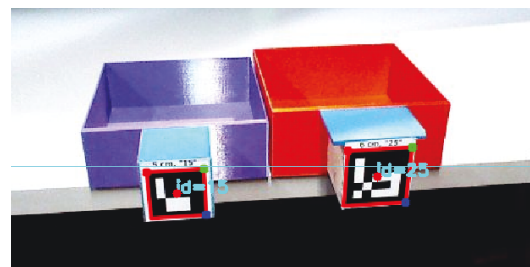


Figura 4: Detección marcadores ArUco

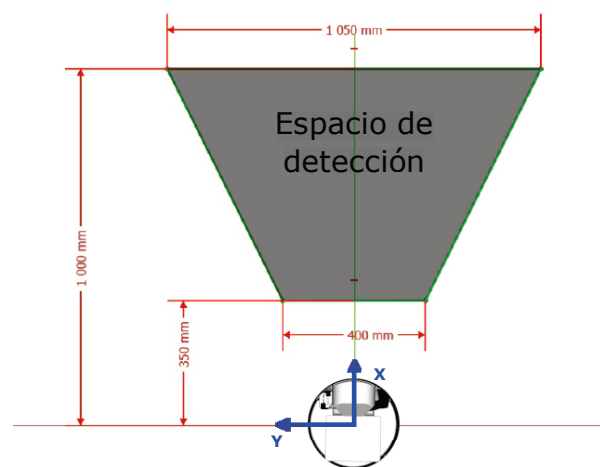


Figura 5: Rango de detección de ArUcos

3.2. CONTROL

El control consiste en comandar órdenes a determinados componentes para ejecutar un movimiento. Se ha implementado un control de la base móvil, de la cabeza, de la pinza y del brazo-torso.

3.2.1. Control de base móvil. El control de la base móvil se implementa en el nodo de navegación ofreciendo un servicio de ROS. Este, mediante un *topic* de ROS, proporciona a los controladores de la base móvil la velocidad lineal y angular objetivo que ha recibido como petición.

3.2.2. Control de la cabeza. Para el control de la cabeza se ha implementado una clase de C++ que permite crear un objeto con el que manejar las órdenes de movimiento de la cabeza, la cual consiste en un sistema de *pan&tilt*. El objeto permite comandar la cabeza a los ángulos de *pan* y de *tilt* deseados, o a orientar la dirección de la visión hacia un punto respecto al sistema de referencia de la cámara.

3.2.3. Control de la pinza. El control de la pinza se ha implementado de la misma forma que el de la cabeza, por medio de una clase que permite definir un objeto para su manejo. En este caso, a la pinza se le puede ordenar adoptar un espacio de apertura determinado, o realizar un agarre. Este último consiste en el cierre de la pinza hasta detectar resistencia, y es comandado por medio de un servicio de ROS ofrecido por el software nativo de TIAGo.

3.2.4. Control del grupo brazo-torso. De nuevo, para controlar este grupo se ha creado una clase para definir un objeto manejador. Esta clase además se sirve de la librería MoveIt [3] para la planificación de los movimientos del brazo. Esta integra distintos algoritmos de planificación, así como detección y prevención de colisiones a través de una interfaz consistente en un objeto manejador. El software nativo de TIAGo usa también esta librería y posee la configuración previa para su uso con el robot.

De esta forma, el objeto de control del grupo brazo-torso permite comandar cuatro tipos de movimientos:

- **Movimientos articulares:** se proporciona la pose del efector final objetivo y el planificador computa y comanda la trayectoria articular del brazo para alcanzarla.
- **Movimientos cartesianos:** se proporciona la pose del efector final objetivo y el planificador computa y comanda una trayectoria que siga una línea recta entre la pose inicial y objetivo del efector final.
- **Movimientos articulares según configuración:** se proporciona la configuración articular objetivo del grupo brazo-torso y el planificador computa y comanda la trayectoria para alcanzarla.
- **Movimiento a posición de reposo:** se comanda el brazo a una configuración articular fija de reposo, pasando por configuraciones intermedias que aseguran una trayectoria confiable sin auto-colisiones. Esta posición de reposo coloca al brazo recogido cerca del torso, de forma que no moleste en la navegación ni en el campo de visión de la cámara de la cabeza. Este movimiento es uno de los procesos que se observan en la Figura 3.

Como se ha mencionado, una de las ventajas de MoveIt es la posibilidad de evitar obstáculos. Para ello, con el objeto de control del grupo es posible añadir obstáculos virtuales a la escena de planificación de MoveIt, de forma que el algoritmo de planificación de trayectorias los tenga en consideración.

3.3. PROCESOS

La rutina de la tarea es implementada en el nodo de manipulación, el cual se encarga de coordinar los manejadores de control presentados previamente junto con la información de visión. De esta forma, este nodo implementa el diagrama de flujo presentado previamente en la Figura 3, compuesto por cuatro procesos principales: búsqueda del marcador, corrección la posición de la base, centrado del brazo y coger el contenedor.

3.3.1. Búsqueda del marcador. Este proceso tiene como fin detectar la posición del marcador ArUco respecto de la base del robot. Sigue el siguiente orden:

- Paso 1. Se realiza un barrido vertical con la cabeza y la cámara RGBD, con el fin de detectar un marcador ArUco con el identificador deseado que se encuentre en frente en su campo de visión. Se detiene cuando es detectado en el nodo de visión y se obtiene su posición en la imagen.
- Paso 2. Se centra la mirada de la cabeza hacia el centro del ArUco y se vuelve a obtener su posición en la imagen, pasando al paso 4. En caso de no detectarse el marcador tras el centrado, se va al paso 3.
- Paso 3. Como no se ha detectado el ArUco al centrar la mirada, se mueve la cabeza en vertical 5° con el fin de variar la imagen actual y comprobar si en esta nueva situación sí se detecta el ArUco. Si es detectado, se obtiene su posición en la imagen y se pasa al paso 2. En caso contrario, se pasa al paso 1.
- Paso 4. Se solicita al nodo de visión la posición 3D del ArUco respecto a la base, terminando así el proceso con éxito. En caso de no ser posible, se vuelve al paso 3.

Cada paso puede ser repetido hasta un máximo de tres ocasiones, para evitar bucles infinitos de búsqueda de marcador.

3.3.2. Corrección de la posición de la base. Este proceso se sirve del servicio del nodo de navegación para colocar la base del robot a la distancia óptima del contenedor para ser cogido. Estas distancias óptimas son las que se presentan en la Tabla 1. El proceso comandará movimientos lineales hasta llegar a una distancia óptima. Estos movimientos son a un velocidad de 0.1m/s, para garantizar seguridad y precisión.

3.3.3. Centrar la pinza. Este proceso consiste en centrar la pinza con el centro del ArUco para así obtener una mejor aproximación a la posición real del contenedor.

Primeramente, el proceso comienza situando un obstáculo virtual que consiste en una pared cúbica debajo

Tabla 1. Distancia óptima en X desde la cuál coger el contenedor según la posición en Y de la misma, respecto al sistema de referencia de la base de TIAGo.

$y_{contenedor} \text{ (m)}$	$d_{\text{óptima}} \text{ (m)}$
$-0.2 < y < 0.2$	$0.85 < x < 0.9$
En otro caso	$0.8 < x < 0.85$

de la pose del ArUco detectado, para prevenir al brazo de colisiones con la mesa en la que se sitúa el contenedor.

Seguidamente, se comanda el efector final a situarse a 0.15m en el eje X de la posición del ArUco y se hace coincidir el centro de la imagen de la cámara de la pinza con el centro del ArUco. Para ello, se comandan al brazo movimientos cartesianos en los ejes X e Y proporcionales al error en píxeles entre los centros. Esta proporción es a razón de 0.32mm por píxel de error, dato que ha sido obtenido empíricamente para la situación de la cámara en dicha posición.

En caso de que en algún punto del proceso no se detecte el marcador ArUco, se inicia un movimiento en espiral del brazo con el fin de encontrarlo. Una vez detectado, se continua con el centrado proporcional.

De nuevo, los pasos de centrado y espiral tendrán un máximo de tres repeticiones para evitar bucles infinitos.

3.3.4. Coger el contenedor. Este sería el último proceso por realizar, el de coger el contenedor o *pick*. Se toma la pose de la pinza alineada con el ArUco y se le añade un determinado *offset* para obtener la posición de agarre del contenedor. Este *offset* sirve para compensar la pose de la pinza, de forma que lo que se veía centrado en la imagen de la cámara de la pinza se encuentre en el espacio de agarre de la pinza.

Seguidamente, se aproxima la pinza al asa del contenedor y se procede al agarre. Una vez agarrado, se eleva el brazo y se retira. Finalmente, se mueve la base móvil 10cm hacia atrás, quedando el robot retirado de la mesa.

4. RESULTADOS EXPERIMENTALES

Para este artículo se han realizado pruebas que han consistido en realizar la tarea de coger el contenedor variando la posición inicial de este en el eje Y y la posición de la base del robot con respecto al contenedor en el eje X, partiendo con el brazo del robot en una posición distinta a la de reposo. Este sistema de referencia se puede observar en la Figura 5. De esta forma se contemplan las 9 posibilidades que se extraen de la Tabla 1. En total se han realizado 45 repeticiones, lográndose con éxito en todas ellas coger el contenedor. Cabe destacar, que en estas pruebas solo se busca la manipulación del objeto, por lo que no han participado usuarios.

Para poder realizar una comparación entre los 9 casos experimentados, en cada prueba se han medido la posición inicial en la que se detecta el ArUco en los ejes Y y X respecto de la base del robot, la distancia que recorre el robot para adecuarse a la distancia óptima y el tiempo de ejecución de la prueba.

En la Tabla 2 se muestra la media de cada medida para cada caso, donde x_o e y_o son la media de las posiciones iniciales del contenedor en metros, dx es el desplazamiento medio en metros que ha realizado el robot para coger

el contenedor y t_e es la media del tiempo de ejecución de la tarea en segundos. En la gráfica de la Figura 6 se presenta una visualización de las medidas, mostrando en el eje horizontal la posición inicial del ArUco en el eje Y respecto a la base del robot y en el eje vertical el tiempo de ejecución. Además, se diferencian entre los rombos azules en las pruebas donde el robot no se desplazó, los cuadrados naranjas en las pruebas en las que el robot se desplazó hacia atrás y los triángulos grises en las pruebas en las que se desplazó hacia adelante.

Tabla 2. Media de las medidas resultantes de las pruebas.

$y_o \text{ (m)}$	x_o	$dx \text{ (m)}$	$t_e \text{ (s)}$
-0,03	0,89	0,00	53,5
0,00	0,74	-0,09	66,9
-0,04	1,02	0,15	66,3
0,22	0,85	0,00	61,6
0,28	0,75	-0,05	67,0
0,24	0,97	0,15	72,0
-0,22	0,83	0,00	59,3
-0,23	0,76	-0,03	62,8
-0,25	0,92	0,10	76,6

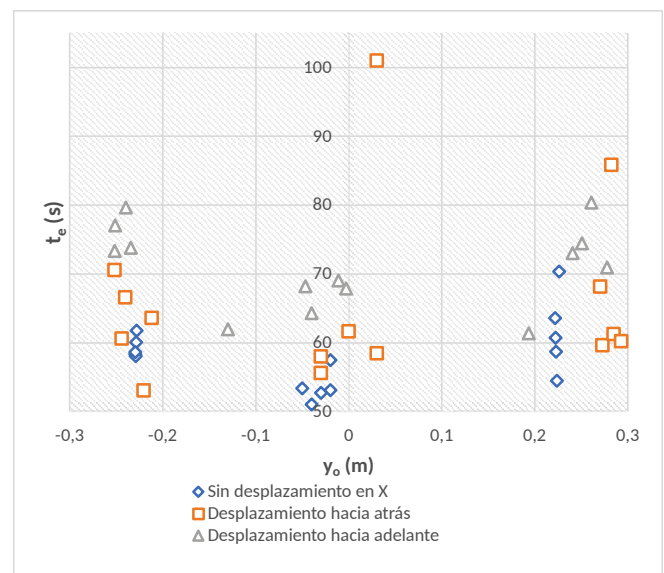


Figura 6: Gráfica de las medidas de las pruebas. El eje X es la posición inicial en Y en metros y el eje Y es el tiempo de ejecución en segundos. Los rombos azules son las pruebas iniciadas a distancia óptima, los cuadrados naranjas las pruebas iniciadas más cerca de la óptima y los triángulos grises las iniciadas más lejos.

Tanto de la Tabla 2 como de la gráfica de la Figura 6 se puede extraer que los tiempos de ejecución más bajos se logran cuando el robot no tiene que adecuar la posición de la base a la distancia óptima, incrementándose cuando sí tiene que desplazarse. Esto es debido a que en dichos casos es necesario realizar y repetir más procesos de la rutina (Figura 3). Además, por lo general y viendo el valor medio, cuando el desplazamiento es hacia adelante se produce el mayor incremento de tiempo. Esto se debe a que en dichas pruebas la distancia a recorrer es mayor que la distancia a recorrer cuando el robot debía retroceder, ya que en este segundo caso se tiene la mesa más cerca

y no es posible acercar más el robot al ArUco, para evitar colisiones del brazo cuando este va a la posición de reposo. Por último, se observa que sin desplazamiento el tiempo es mejor cuando el contenedor está centrada respecto a TIAGo.

5. CONCLUSIONES

En este artículo se ha implementado con éxito la tarea de manipulación para la asistencia usando el robot TIAGo. Esta tarea consiste en determinar la posición de unos contenedores con marcadores ArUco por medio de visión por computador para luego cogerlos.

Para ello, se han creado cajas de distinto tamaño con un marcador ArUco y se ha desarrollado software de visión para la identificación y localización de estos marcadores, que combinado con el uso de información de profundidad de la cámara RGBD, proporcionara la posición en el espacio de los ArUco. También, se ha desarrollado el software necesario para el manejo y control de los componentes de TIAGo (base, cabeza, brazo, torso y pinza). Finalmente, el uso conjunto de este software ha permitido implementar procesos y rutinas que lleven a la consecución de la tarea de asistencia.

Por otro lado, se han realizado con éxito numerosas pruebas de la tarea de coger el contenedor, variando la posición de este respecto al robot. Los resultados de las pruebas indican que el principal factor que aumenta el tiempo de ejecución es tener que desplazar la base para ajustar su posición respecto al contenedor, ya que implica realizar y repetir más procesos. Además, se extrae que situar el contenedor justo enfrente del robot reduce también los tiempos medios de ejecución.

Agradecimientos

Esta investigación ha sido financiada por la Asociación Inserta Innovación (parte del Grupo Social Once) a través de Prosvasi Ciencia y Tecnología Para la Inclusión, A.I.E y en el marco del proyecto ACCESSROBOTS; el proyecto de I+D+i, ROBOASSET, “Sistemas robóticos inteligentes de diagnóstico y rehabilitación de terapias de miembro superior”, PID2020-113508RB-I00, financiado/a por AEI/10.13039/501100011033; “RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub”, S2018/NMT-4331, financiado por “Programas de Actividades I+D en la Comunidad de Madrid”; y fondos estructurales de la EU.

English summary

MANIPULATION OF OBJECTS AIMED AT ASSISTING PEOPLE WITH REDUCED MOBILITY

Abstract

Robotics continues to advance and is increasingly being integrated into various fields, including assistive robotics. In this article, assistive robotics is used to help people with reduced mobility, a growing group. The aim is to increase the independence of this group and improve

the quality of care in homes and specialized centers. In this article, the handling task for assistance has been successfully implemented using the TIAGo robot, which consists of determining the position of containers with ArUco markers by means of computer vision and then picking them up. For this purpose, the containers have been designed and the necessary vision and manipulation software has been developed. Finally, tests have been performed and evaluated in which the positions of the robot and the container have been varied, measuring the execution time and the displacement of the robot, with satisfactory results.

Keywords: Assistive robot, reduced mobility, ArUco, TIAGo, manipulation

Referencias

- [1] F. G. Aznárez y G. L. Morante, *Situación y perspectivas del transporte público de las personas con movilidad reducida en España*. Madrid: Escuela Libre, 1997.
- [2] S. Carretero, “Mapping of Effective Technology-based Services for Independent Living for Older People at Home,” European Commission, Joint Research Center, Institute for Prospective Technological Studies, rep. de inv., 2015.
- [3] S. Chitta, “MoveIt!: An Introduction,” en *Robot Operating System (ROS): The Complete Reference (Volume 1)*, A. Koubaa, ed. Cham: Springer International Publishing, 2016, pp. 3-27. DOI: 10.1007/978-3-319-26054-9_1.
- [4] “Detection of ArUco Markers,” OpenCV. (s.f.), [En línea]. Disponible en: https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html (Acceso: 04-06-2022).
- [5] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas y M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” *Pattern Recognition*, jun. de 2014. DOI: 10.1016/J.PATCOG.2014.01.005.
- [6] “PAL.” (s.f.), [En línea]. Disponible en: <https://pal-robotics.com/es/> (Acceso: 16-06-2022).
- [7] “Tiago. El robot manipulador móvil que se adapta a tu investigación.” PAL. (s.f.), [En línea]. Disponible en: <https://pal-robotics.com/es/robots/tiago/> (Acceso: 01-06-2022).
- [8] M. P. Venturiello, “Los adultos con discapacidad motriz y sus familiares: la organización del hogar, los afectos y el trabajo,” *Revista Española de Discapacidad*, vol. 2, n.º 2, 103-120 2014. DOI: 10.5569/2340-5104.02.02.06.



Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento - No Comercial - Sin Obra Derivada**