

Control Servovisual de un Kuka youBot[®] para la manipulación y traslado de objetos[★]

F. Mirelez-Delgado^{*,**} A. Morales-Díaz^{*} R. Ríos-Cabrera^{*}
H. Pérez-Villeda^{*}

** Centro de Investigación y de Estudios Avanzados del Instituto
Politécnico Nacional Unidad Saltillo
Ramos Arizpe, Coah. 25900, México
Teléfono: (844) 4389600 Ext. 8500*

*** flabio.mirelez@cinvestav.edu.mx*

Resumen: Este trabajo presenta la implementación de un Control Servovisual Basado en Imagen en un robot manipulador móvil omnidireccional Kuka youBot. El sistema de visión está compuesto por un sensor RGB-D Asus Xtion Pro[®]. La ley de control implementada tiene la estructura de un PD clásico para la plataforma móvil. El manipulador móvil se desplaza a puntos 3D conocidos mediante el cálculo de cinemática inversa. En este artículo se demuestra la efectividad del algoritmo en la localización del objeto de interés así como en la manipulación del mismo para llevarlo de su lugar original a otro espacio deseado.

Keywords: Control servovisual, robot manipulador, robot móvil omnidireccional.

1. INTRODUCCIÓN

La información visual es una herramienta ampliamente utilizada en la actualidad en el reconocimiento de entornos desconocidos. El control basado en visión para sistemas robóticos ha despertado mucho interés en estos últimos años debido a la reducción de costos de las cámaras de visión y en general de todo el sistema de procesamiento asociado, ver Hutchinson et al. (1996).

El control visual ha madurado rápidamente y ha sido aplicado en el control de robots manipuladores ya que se basa en la percepción visual del robot y la ubicación de una pieza de interés, ver Corke (1996). En concreto, el control visual implica el uso de una o más cámaras y un sistema de visión por computadora para controlar la posición del efector final del robot según lo requiera la tarea a efectuarse con la pieza de trabajo.

Una clasificación de las estructuras de los sistemas con realimentación visual se introdujo en Weiss et al. (1987) y en Hutchinson et al. (1996). La manipulación de objetos con sistemas robóticos es una tarea que también ha recibido mucha atención por parte de la comunidad científica internacional, uno de los principales retos es tener conocimiento del objeto a manipular, por ejemplo, su localización en el mundo, su pose, puntos de agarre o incluso si el material es deformable o rígido. Uno de los primeros trabajos que se enfocaron en este tema es Murray et al. (1994) donde se establece una introducción a la manipulación de objetos con sistemas robóticos. Una investigación más extensa acerca del control servovisual para manipulación se presenta en Kragic and Christensen (2002) y en Corke (1996).

En este trabajo se establece un algoritmo de visión por computadora en un robot manipulador móvil omnidireccional, para la identificación de objetos de interés mediante segmentación por color. También se implementa una ley de control que permite al robot aproximarse a los objetos deseados para que el brazo tome los objetos y los traslade a un lugar deseado dentro del espacio de trabajo.

El artículo está estructurado de la siguiente manera; en la sección 2 se establece un panorama del trabajo relacionado con el Kuka youBot. La sección 3 muestra el modelo cinemático del Kuka youBot, para la plataforma omnidireccional el modelo se describe en 3.1, mientras que la subsección 3.2 se enfoca en el modelo cinemático para el brazo manipulador. El procesamiento de las imágenes obtenidas por el sensor RGB-D se explica en la sección 4. La sección 5 describe la ley de control usada este artículo. Los experimentos y los resultados obtenidos se muestran en la sección 6 y por último en la sección 7 se plasman las conclusiones y el trabajo a futuro.

2. TRABAJO RELACIONADO

El Kuka youBot es un robot manipulador móvil omnidireccional desarrollado especialmente para fines académicos y de investigación, como se menciona en Bischoff et al. (2011). Una de las principales ventajas de esta plataforma es que cuenta con computadora abordo que usa el sistema operativo Ubuntu, cuenta con una API (*Application Programming Interface*) desarrollada en C++, además se puede usar con ROS (*Robot Operating System*) brindando así una mayor flexibilidad para desarrollar aplicaciones con este robot.

La cinemática directa del manipulador se expresa de manera clara en Dwiputra et al. (2014) donde se utiliza el software Modelica para la caracterización del brazo robótico. Por otra parte en Sharma et al. (2012) se es-

[★] El autor agradece a CONACyT por el apoyo número 395788 para beca de doctorado.

tablece una cinemática inversa unificada para el Kuka youBot, de tal manera que no es necesario tener al robot desacoplado en dos cuerpos (móvil y manipulador), además se resuelven de manera efectiva las posibles singularidades al considerar un modelo cinemático completo.

En Dogar et al. (2014) un grupo de Kuka youBots es usado para manipular piezas de muebles y realizar el ensamble de una mesa, en ese trabajo los robots son supervisados por un sistema de cámaras externo que permite conocer la ubicación de cada integrante del grupo en todo momento, adicionalmente algunos robots tienen cámara en mano por lo que el ambiente de trabajo está estrictamente controlado.

Finalmente en Keiser (2013) se desarrolla un control de par que se aplica a las articulaciones del manipulador del Kuka youBot para que éste pueda sujetar piezas que están en el piso. La identificación del objeto se hace a través de un sensor RGB-D y de la librería PCL (*Point Cloud Library*), el sensor está fijo en un pedestal en la parte posterior de la plataforma móvil, lo que permite al objeto estar en el rango del sensor de profundidad. Sin embargo en ese artículo el control del manipulador está en lazo abierto, por lo que el cálculo del movimiento para el brazo se hace fuera de línea y cuando se ejecuta no hay retroalimentación del sensor RGB-D.

3. MODELO CINEMÁTICO DEL KUKA YOUTBOT

3.1 Modelo cinemático de la plataforma omnidireccional

El robot manipulador móvil Kuka youBot cuenta con una plataforma omnidireccional de cuatro ruedas tipo Mecanum que permiten el libre desplazamiento del sistema en el plano Cartesiano, sin embargo es necesario considerar la disposición de los rodillos de este tipo de ruedas para obtener un modelo cinemático fidedigno. Las medidas de la plataforma móvil se muestran en la Figura 1 y están expresadas en milímetros.

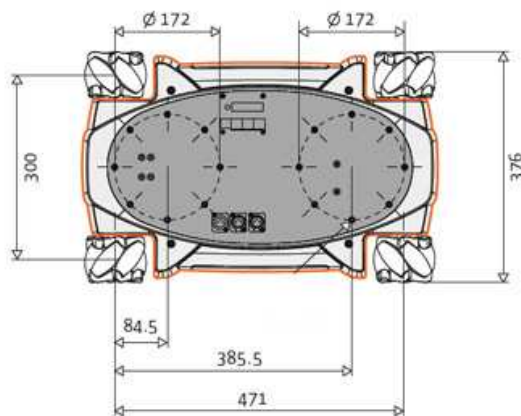


Fig. 1. Medidas de la plataforma móvil.

De acuerdo con Nagatani et al. (2000), la configuración cinemática de un robot móvil omnidireccional está dada por el conjunto de ecuaciones (1).

$$\begin{aligned}\dot{x} &= v_l = \frac{1}{4}(d_1 + d_2 + d_3 + d_4) \\ \dot{y} &= v_t = \frac{1}{4}(-d_1 + d_2 + d_3 - d_4) \tan(\alpha_b) \\ \dot{\theta}_b &= v_a = \omega = \frac{1}{4}(d_1 + d_2 + d_3 + d_4)\beta\end{aligned}\quad (1)$$

Donde $\dot{x}$ y $\dot{y}$ son la velocidad en los ejes X y Y , o bien las velocidades longitudinal y transversal (v_l y v_t), mientras que $\dot{\theta}_b$ es la velocidad angular (v_a) de la plataforma omnidireccional. Por otra parte d_i es la velocidad lineal de cada rueda de la plataforma móvil. Los parámetros α_b y β son calculados experimentalmente debido a que dependen del ángulo en que estén dispuestos los rodillos dentro de las ruedas Mecanum (ver Figura 2).

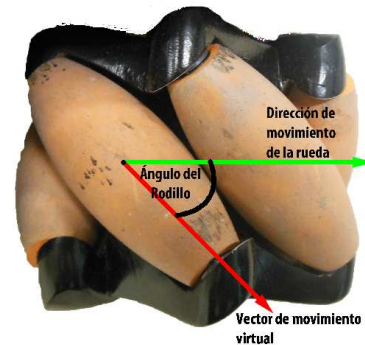


Fig. 2. Descripción de la rueda tipo Mecanum.

El movimiento del robot móvil en el plano dependerá entonces de la combinación de las velocidades lineales de cada rueda de la plataforma, particularmente los movimientos de traslación transversal y rotación se ilustran en la Figura 3 donde la plataforma difuminada indica hacia donde se moverá el robot móvil según la combinación de velocidades lineales de las cuatro ruedas.

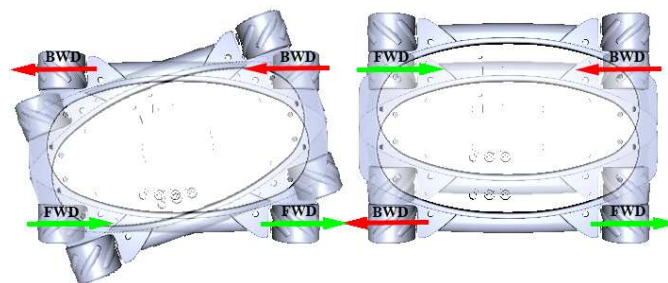


Fig. 3. Movimiento de la plataforma móvil.

3.2 Modelo cinemático del manipulador de 5 grados de libertad

El brazo manipulador del Kuka youBot es un robot de 5 grados de libertad (todos rotacionales), las medidas de cada eslabón así como sus límites articulares se pueden observar en la Figura 4, el efector final es una pinza de dos dedos que le permite sujetar y manipular objetos pequeños.

El modelo cinemático del robot manipulador se obtiene mediante el algoritmo de Denavit-Hartenberg, ver Lipkin

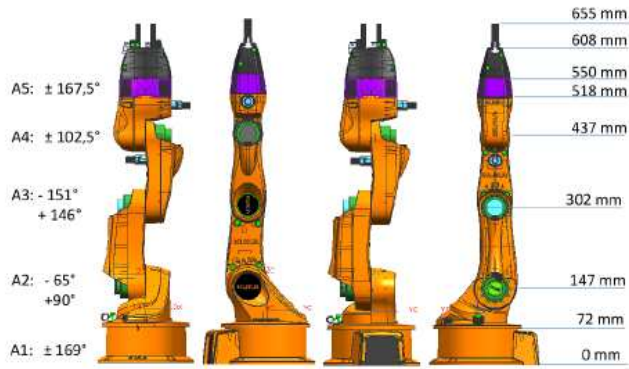


Fig. 4. Descripción del manipulador del Kuka youBot.

(2005). Los marcos de referencia para cada articulación se pueden apreciar en la Figura 5 y los parámetros de Denavit-Hartenberg obtenidos para el manipulador se muestran en la Tabla 1, mientras que en la Tabla 2 se muestran las distancias y ángulos entre los marcos de referencia para el brazo manipulador.

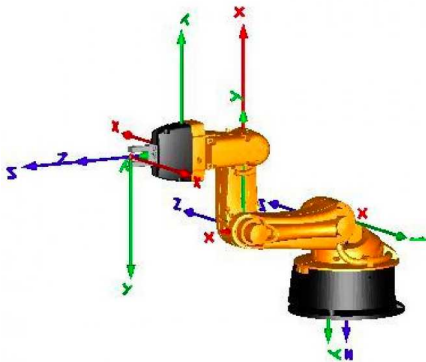


Fig. 5. Posicionamiento de los marcos de referencia para cada articulación.

Tabla 1. Parámetros de Denavit-Hartenberg para el manipulador del youBot.

Eslabón	θ	d	a	α
1	q_1	0.147	0.0330	$\frac{\pi}{2}$
2	q_2	0	0.1550	0
3	q_3	0	0.1350	0
4	$q_4 + \frac{\pi}{2}$	0	0	$\frac{\pi}{2}$
5	q_5	0.2175	0	0

Tabla 2. Cadena cinemática para el manipulador del youBot.

	Marco anterior	Traslación [cm]			Rotación [grados]		
		x	y	z	x	y	z
Articulación 1	Base	2.4	0	11.5	180°	0°	0°
Articulación 2	Articulación 1	3.3	0	0	90°	0°	-90°
Articulación 3	Articulación 2	15.5	0	0	0°	0°	-90°
Articulación 4	Articulación 3	0	13.5	0	0°	0°	0°
Articulación 5	Articulación 4	0	11.36	0	-90°	0°	0°
Pinza	Articulación 5	0	0	5.716	90°	0°	180°

Con los parámetros de las Tablas 1 y 2 es posible obtener las matrices de transformación homogéneas de acuerdo a la ecuación (2).

$$A_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & -S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Donde los términos S y C denotan las funciones $\sin()$ y $\cos()$ respectivamente. La matriz de transformación total se obtiene de la multiplicación sucesiva de cada una de las matrices homogéneas.

$$T_n^0 = A_1^0 \dots A_n^{n-1} \quad (3)$$

4. PROCESAMIENTO DE LA IMAGEN

El sensor Asus Xtion Pro (ver Figura 6) permite obtener una imagen a color (RGB), una matriz (imagen) con valores de profundidad y a partir de éstas se obtiene una nube de puntos 3D. La ventaja de este tipo de sensores es que son de código abierto y se pueden programar usando las librerías OpenNI y OpenCV.



Fig. 6. Descripción del sensor Asus Xtion Pro.

El procedimiento para identificar un objeto de interés consiste en obtener el mapa de puntos 3D de la escena captada por el sensor RGB-D, esto con el objetivo de tomar dos puntos alineados en el eje Y de la imagen pero separados en el eje X , de tal manera que se obtenga un error de profundidad que permita al robot orientarse perpendicularmente a los dos puntos de interés y por ende al objeto de interés. Posteriormente se captura una imagen RGB con el sensor Asus en cada iteración del ciclo. La imagen a color es transformada al espacio de color HSV puesto que éste es más robusto a cambios de iluminación en el ambiente. Una vez en el espacio HSV se eliminan los píxeles que no son del color deseado. Los objetos deseados usados en este artículo son tres figuras geométricas que difieren en color (azul, rojo y verde), además contrastan con el ambiente (pared y repisa blancos) de tal manera que la segmentación por color es más sencilla de realizar.

Una vez que el color del objeto ha sido aislado de la imagen, ésta se binariza para poder obtener el contorno del objeto en cuestión. El contorno permite calcular el centroide del objeto así como el área en píxeles del mismo (ver Figura 7). La Figura 8 muestra el proceso para segmentar un objeto de interés a partir de una imagen RGB.

5. LEY DE CONTROL

El área del objeto y su centroide que se obtienen en el proceso de segmentación son utilizados para definir tres errores (en los ejes X, Y, Z de la imagen) que son la entrada al algoritmo de control del robot manipulador móvil.

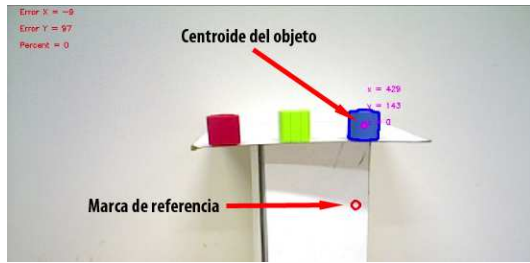


Fig. 7. Procesamiento de la información visual obtenida con el sensor RGB-D.

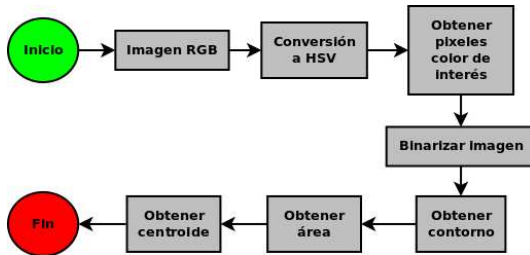


Fig. 8. Algoritmo de segmentación de objetos.

El algoritmo de control propuesto para este trabajo es un Control Visual Basado en Imagen con ganancias Proporcional y Derivativa. Los errores de profundidad, en los ejes X y Y de la imagen y en el área se definen de acuerdo a la siguiente ecuación:

$$\begin{aligned} e_p^c &= p_2 - p_1 \\ e_x^c &= C_{x_d} - C_x \\ e_y^c &= C_{y_d} - C_y \\ e_z^c &= \text{área}_d - \text{área} \end{aligned} \quad (4)$$

Donde p_2 y p_1 son los valores de profundidad para dos puntos de interés en la escena al principio del experimento, una vez que el robot queda perpendicular a una superficie plana (pared) se considera que la diferencia de profundidad entre los dos puntos de referencia no aumenta. El punto (C_{x_d}, C_{y_d}) es la referencia en la imagen para indicar a donde se debe desplazar el centroide del objeto de interés (C_x, C_y) , de manera similar área_d es el área deseada que debe tener el objeto en relación con la imagen, este valor deseado está directamente relacionado con la proximidad del efector final al objeto.

La plataforma móvil no tiene restricciones de movimiento, de tal forma que el error en el eje X de la imagen se puede minimizar con movimientos transversales de la plataforma, mientras que el error de área se reduce con movimientos longitudinales. Por lo tanto la ley de control queda definida de la siguiente forma:

$$\begin{aligned} v_t &= K_p(e_x^c) + K_d(\dot{e}_x^c) \\ v_l &= K_p(e_z^c) + K_d(\dot{e}_z^c) \\ v_a &= K_p(e_p^c) + K_d(\dot{e}_p^c) \end{aligned} \quad (5)$$

Donde K_p y K_d son las ganancias proporcional y derivativa respectivamente. De manera general el proceso para lograr que el robot identifique un objeto de interés y que posteriormente lo manipule para llevarlo a otro lugar de interés se representa en la Figura 9.

La primera parte del proceso consiste en inicializar el robot y el sensor RGB-D, en la siguiente etapa se obtiene

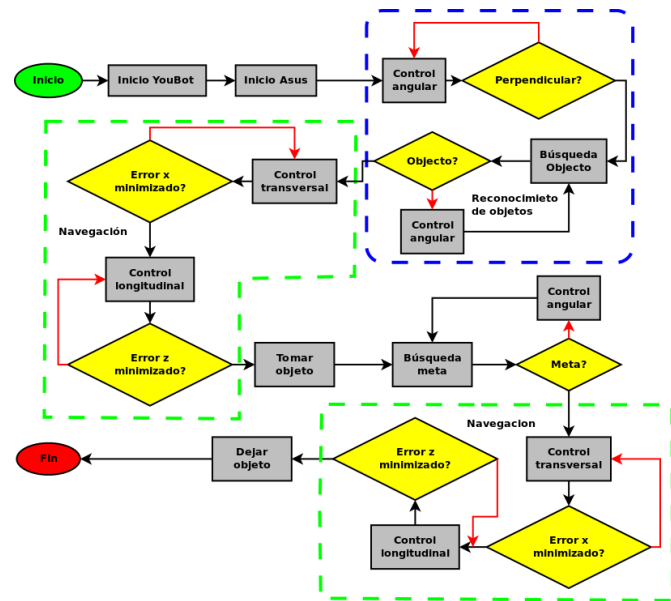


Fig. 9. Metodología a seguir para la identificación y manipulación de objetos.

la nube de puntos de la escena vista por el sensor para obtener un error definido por la diferencia de profundidad entre dos puntos del mapa 3D. El error de profundidad es minimizado por la ley control aplicada en la velocidad angular de la plataforma móvil, de tal manera que cuando el error es mínimo se considera que el robot está perpendicular a los dos puntos de interés.

En la siguiente parte del proceso se considera sólo la imagen de color para segmentar un objeto de interés (azul, rojo o verde), el proceso de segmentación se realiza en el espacio de color HSV para tener mayor robustez a los cambios de iluminación. Después de que el objeto es segmentado se obtienen tres características principales; el contorno, el centroide y el área en pixeles. Estas características se usan para obtener los errores que son minimizados por la ley de control. Cuando los errores han sido reducidos a valores muy cercanos a cero, el brazo manipulador se desplaza hacia el objeto de interés para tomarlo, el movimiento del manipulador se logra usando cinemática inversa, puesto que se conoce la distancia entre el objeto de interés y el punto Cartesiano del efector final cuando los errores son mínimos.

Una vez que el objeto ha sido sujetado por el robot manipulador se realiza un proceso de búsqueda del lugar donde el objeto debe ser liberado. Para localizar el lugar de interés se procesa la imagen en color para encontrar un código QR que es colocado en una pared, para ello se utiliza un algoritmo de identificación de códigos QR y el control de velocidad angular de la plataforma móvil. Cuando el lugar ha sido identificado se obtiene el centroide de la imagen QR y posteriormente se efectúa el control de velocidad angular para posicionar al robot de manera perpendicular al plano del código QR. El siguiente paso es minimizar los errores e_x^c y e_z^c a través de los controles en velocidad transversal y longitudinal.

Cuando el robot está en la posición deseada, es decir que los errores e_x^c y e_z^c han sido reducidos a valores muy cercanos a cero, el manipulador se desplaza a un punto

conocido usando cinemática inversa para finalmente dejar el objeto en el lugar deseado.

6. EXPERIMENTOS Y RESULTADOS

El escenario experimental se muestra en la Figura 10 donde tres objetos de diferente color son puestos en una mesa, el código QR está colocado en otra posición dentro del espacio de trabajo y es usado para identificar el lugar a donde el objeto deberá ser transportado. La pieza de trabajo a tomar en este experimento es el cilindro azul, para poder segmentarlo, el sensor RGB-D se coloca encima del quinto eslabón del brazo, por lo que el robot manipulador móvil tiene cámara “en mano”. El procesamiento de imágenes

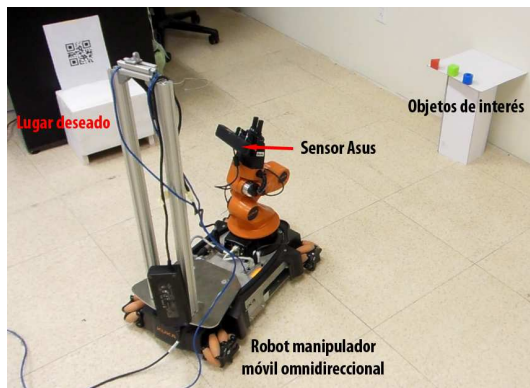


Fig. 10. Escenario experimental.

así como el control se implementaron directamente en el robot ya que cuenta con una computadora abordo con 32 GB de almacenamiento en estado sólido, 2 GB de memoria RAM y un procesador Intel Atom D510 Dual Core a 1.66 GHz. La computadora tiene instalado Ubuntu 12.04 LTS y la programación fue realizada en C++.

En la Figura 11 se muestra la trayectoria total que recorrió el robot. Los datos mostrados son proporcionados por la odometría del robot, en este caso las muestras de posición son sensadas antes de que el control comience a actuar.

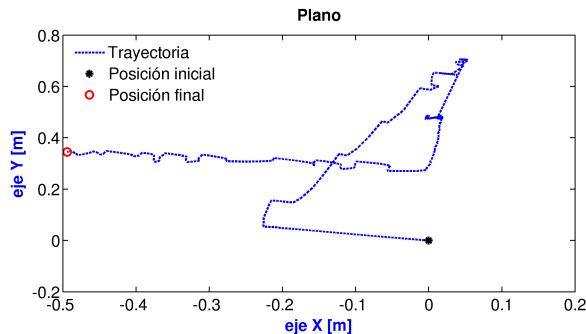


Fig. 11. Trayectoria recorrida por el Kuka youBot.

En la Figura 12 se muestra el error de profundidad que se considera para posicionar al robot de manera perpendicular a una superficie plana, en este caso una pared. Este control angular usando la diferencia de profundidad de dos puntos sólo se usa al principio del experimento.

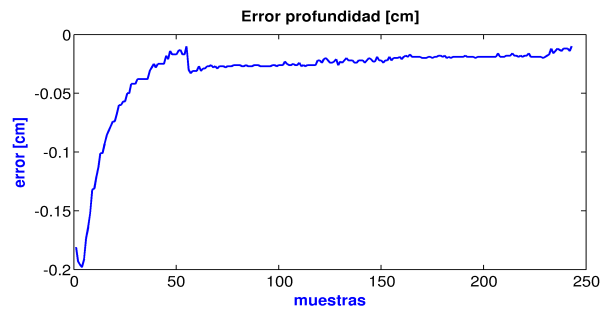


Fig. 12. Error de profundidad entre dos puntos en la nube de puntos.

En la Figura 13 se pueden apreciar los diferentes cambios de orientación que sufre el robot durante el experimento, éstos se efectúan con la finalidad de tener al robot orientado de manera perpendicular al objeto de interés para que cuando el robot se mueva longitudinalmente el error en el eje X de la imagen se afecte lo menos posible.

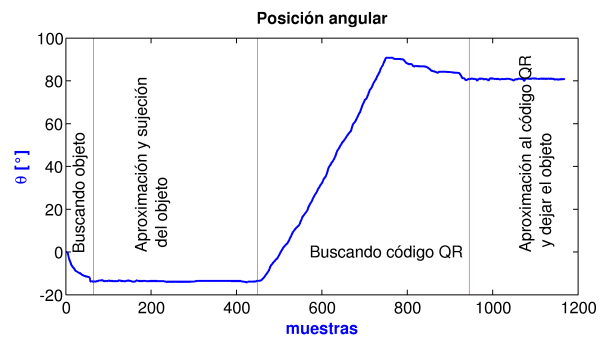


Fig. 13. Evolución de la orientación del robot.

El error en el eje X de la imagen se muestra en la Figura 14, en apariencia el error no disminuye, esto se debe a que la referencia en la imagen no está en el centro de ésta, sino que está desplazada a lo largo del eje X de la imagen; esto logra que la pinza y el objeto estén alineados. Por lo tanto conforme el robot se acerca a la pieza de trabajo, el centroide del objeto se aleja del punto de referencia en la imagen, sin embargo la ley de control se encarga de que el robot vuelva a alinearse con el objeto a pesar de los incrementos en el error e_x^c .

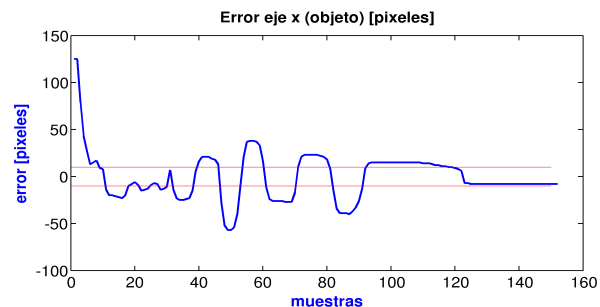


Fig. 14. Error en el eje X de la imagen.

Por último, la Figura 15 muestran la evolución del error de área del objeto en la imagen cuando el robot se

está acercando a éste y de manera similar la Figura 16 muestra como disminuye el error de área del código QR.

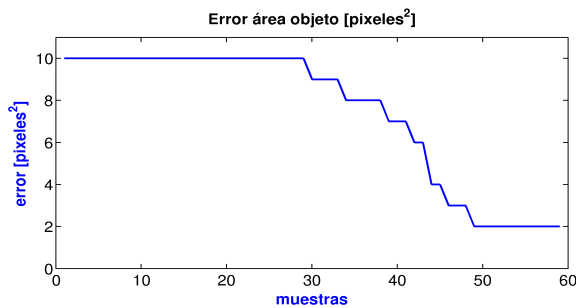


Fig. 15. Error de área del objeto segmentado.

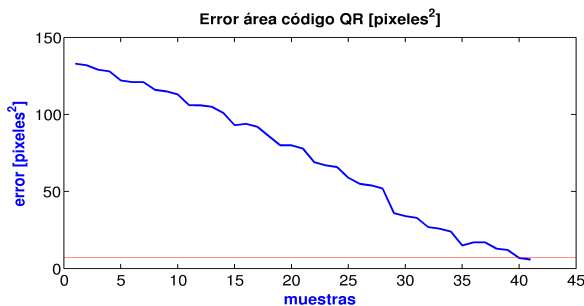


Fig. 16. Error de área del código QR.

Los valores de las ganancias usados en el experimento fueron $K_p = 1$ y $K_d = 0.1$, la programación del Kuka youBot en conjunto con el sensor RGB-D se desarrolló en C++ usando OpenCV y OpenNI.

7. CONCLUSIONES Y TRABAJO A FUTURO

En este artículo se demostró que un robot manipulador móvil omnidireccional dotado con un sistema de visión puede realizar de manera efectiva los procesos de identificación, sujeción y manipulación de objetos. La ley de control propuesta demostró minimizar los errores de interés de tal forma que el acercamiento al objeto así como la aproximación al lugar meta se realizaron de manera efectiva.

A pesar de trabajar en el espacio de color HSV que es más robusto a variaciones en la luminosidad del ambiente, se presentaron inconvenientes para obtener una buena segmentación, ya que dependiendo de la hora del día en la que se realizara el experimento, era necesario hacer pequeños cambios en el rango de valores para segmentar el color deseado.

Como trabajo a futuro se obtendrá un modelo cinemático total que permita diseñar un control general para el robot manipulador móvil así como una prueba de estabilidad que demuestre el buen comportamiento del control propuesto. En cuanto al procesamiento de la imagen se implementará un algoritmo de identificación de objetos basado en modelos 3D para evitar tener problemas con la iluminación. También se desarrollará un software que permita a un humano comunicarse con el robot de manera remota y así poder elegir el objeto de interés a partir

de una galería de imágenes. Por último el sensor RGB-D se colocará en un pedestal en la parte posterior de la plataforma móvil para explotar el mapa de puntos 3D.

REFERENCIAS

- Bischoff, R., Huggenberger, U., and Prassler, E. (2011). Kuka youbot - a mobile manipulator for research and education. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 1–4. doi: 10.1109/ICRA.2011.5980575.
- Corke, P. (1996). *Visual control of robots: high-performance visual servoing*. Robotics and mechatronics series. Research Studies Press.
- Dogar, M., Knepper, R.A., Spielberg, A., Choi, C., Christensen, H.I., and Rus, D. (2014). Towards coordinated precision assembly with robot teams. *Proceedings of the International Symposium of Experimental Robotics*.
- Dwiputra, R., Zakharov, A., Chakirov, R., and Prassler, E. (2014). Modelica model for the youbot manipulator. In *Proceedings of the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*, 1205 – 1212. doi:10.3384/ECP140961205.
- Hutchinson, S., Hager, G., and Corke, P. (1996). A tutorial on visual servo control. *Robotics and Automation, IEEE Transactions on*, 12(5), 651–670. doi: 10.1109/70.538972.
- Keiser, B. (2013). *Torque Control of a KUKA youBot Arm*. Master's thesis, Robotics and Perception Group, University of Zurich.
- Kragic, D. and Christensen, H.I. (2002). Survey on visual servoing for manipulation. Technical report, Computational vision and active perception laboratory.
- Lipkin, H. (2005). A note on denavit-hartenberg notation in robotics. In *29th Mechanisms and Robotics Conference*, volume 7, 921–926. doi:10.1115/DETC2005-85460.
- Murray, R.M., Sastry, S.S., and Zexiang, L. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition.
- Nagatani, K., Tachibana, S., Sofne, M., and Tanaka, Y. (2000). Improvement of odometry for omnidirectional vehicle using optical flow information. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 1, 468–473 vol.1. doi:10.1109/IROS.2000.894648.
- Sharma, S., Kraetzschmar, G.K., Scheurer, C., and Bischoff, R. (2012). Unified closed form inverse kinematics for the kuka youbot. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, 1–6.
- Weiss, L., Sanderson, A., and Neuman, C.P. (1987). Dynamic sensor-based control of robots with visual feedback. *Robotics and Automation, IEEE Journal of*, 3(5), 404–417. doi:10.1109/JRA.1987.1087115.