

**POSITION BASED VISUAL SERVOING OF A ROBOTIC ARM RV2AJ
IMPLEMENTED IN MATLAB AND JAVA****CONTROL VISUAL BASADO EN POSICIÓN DE UN BRAZO ROBÓTICO
RV2AJ IMPLEMENTADO EN MATLAB Y JAVA****MSc. José Gabriel Hoyos Gutiérrez *, MSc. Jaiber Evelio Cardona*,
Ing. Luis Miguel Capacho*, PhD. Flavio Prieto**, PhD. Marco Pérez Cisneros********Universidad del Quindío**

Carrera 15, Calle 12 Norte, Armenia, Quindío, Colombia. Tel.: (576) 746 0100.

E-mail: josegabrielh@uniquindio.edu.co, jaibercardona@uniquindio.edu.co

****Universidad Nacional de Colombia, Sede Bogotá, Colombia.**

E-mail: faprieto@unal.edu.co

**** Universidad de Guadalajara, CUCEI**

Guadalajara, México

E-mail: marco.perez@cucei.udg.mx

Abstract: Visual servoing is a control method that involves artificial vision and robotics; it allows placing the end effector near to an object by just looking it with one or more cameras. In this work a position based visual servoing of a RV2AJ robotic arm is presented, implementation and results are included.

Keywords: Visual servoing, stereo Vision, Robotic arm control.

Resumen: El control visual es un método de control que involucra visión artificial y robótica, el permite que un robot posicione su efector final cerca a un objeto con solo captarlo con una o más cámaras, en este trabajo se presenta la implementación y resultados reales de un sistema de control visual de un brazo robótico RV2AJ usando visión estéreo, con las cámaras mirando el robot.

Palabras clave: Control visual, visión estéreo, Control de Manipulador robótico.

1. INTRODUCCIÓN

El control visual (*Visual Servoing* o *VS*) es una área que permite que un sistema de visión integrado a un robot ubique su efector final cerca a un objeto con solo captarlo con una o más cámaras, incluso haga seguimiento si el objeto se encuentra en movimiento. A partir de visión monocular o estéreo se controlan los movimientos de un robot para alcanzar y/o seguir un objetivo distante de él, pero que está dentro del campo visual. Generalmente la literatura los clasifica en: basado en posición (PBVS), basado en imagen (IBVS) e híbridos, también por la ubicación de la cámara en

el robot: cámara en brazo "*eye in hand*" o cámara mirando el brazo "*eye to hand*", además si es monocular, estéreo o mezcla de ambos (Hutchison 2006; Pérez 2004). Cuando se trabaja directamente con puntos en el espacio (visión estéreo) o estrategias geométricas (visión monocular) pero siempre llegando a una estimación del espacio en tres dimensiones se le llama control visual basado en posición. Si en vez de tratar de encontrar o calcular posiciones en el espacio, se trabaja directamente con las imágenes, o sea se busca que ciertos píxeles deseados (puntos o formas) correspondan a ciertos píxeles captados por la cámara, se llama control visual basado en imagen.

Por otro lado para que el sistema de visión trabaje correctamente, este debe ser calibrado, este proceso establece la relación entre coordenadas tridimensionales de los objetos en el entorno, con sus correspondientes proyecciones bidimensionales en la imagen.

Aplicaciones del control visual se han realizado en procesos industriales como soldadura (Xu, 2005), toma y cuadre de piezas, micro-ensamblado, cirugía robótica (Morales, 2007), seguimiento de vehículos, juguetes como el Aibo, y en interacción humano robot.

La visión estereo es una serie de técnicas matemáticas que permiten la capacidad a un sistema de cómputo de modelar un espacio tridimensional a partir de dos o más imágenes tomadas desde 2 puntos diferentes, el proceso comprende tareas como calibración, extracción de características, apareamiento, reconstrucción (Cyganek, 2009; Pomarez, 2002). Se presenta un sistema de control visual basado en posición usando visión estereo, las cámaras mirando el robot RV2AJ como en la fig. 1 y configuración ECL (*End effector Close Loop*). Un aporte presentado es la implementación usando la diversidad de librerías existentes en Matlab (Bouget, 2000; Corke, 1996) y la flexibilidad de java para manejo de los periféricos y que puede ser llevado a otras plataformas como Linux, incluso en tiempo real (Java RTS- Linux).

2. CONTROL VISUAL BASADO EN POSICIÓN

El objetivo del control visual basado en posición (Wilson, 1996; Hager, 1995) es minimizar el error entre la ubicación del efector final del robot y la ubicación del objeto. En la Figura 1, se puede observar como las cámaras inicialmente ven la posición de la bola p y de la garra o efector final g , la idea del control visual es lograr que las articulaciones del robot se muevan de tal manera que el efector final se ubique en la posición de la bola. La matriz jacobiana J relaciona cambios en las velocidades de las articulaciones a cambios en la velocidad del efector final en el espacio:

$$\dot{r} = J(q)\dot{q} \quad (1)$$

Donde q es el vector de ángulos de las articulaciones, y r es un vector en el espacio cartesiano o incluso un vector que incluye además rotaciones de los ejes (x,y,z) , una ecuación de control (Hashimoto, 2002):

$$\dot{q} = IJ(q)^+(g - p) \quad (2)$$

El valor de λ es una ganancia proporcional que debe ser sintonizada para minimizar el tiempo de convergencia, $J(q)^+$ es la pseudo-inversa de la matriz jacobiana del robot y es una función dependiente de q . Como la ecuación de control (2) da la derivada de q , está se integra para obtener q y llevarla al controlador del robot, el cual tiene otro lazo interno de control (controlador dinámico).

En la figura 2 se muestra un esquema de control implementado con la ecuación 2.

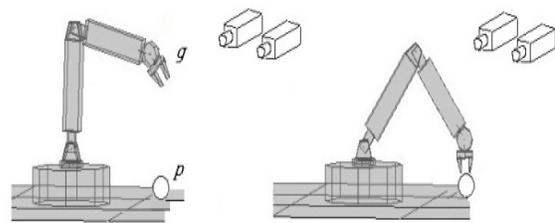
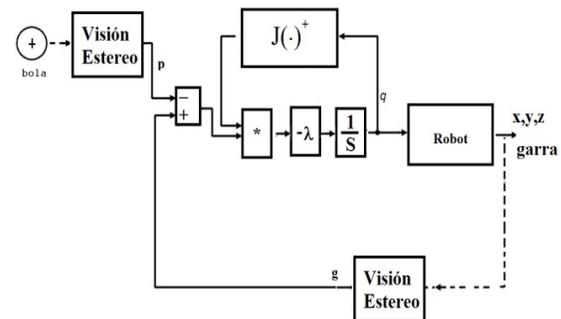


Fig. 1. Sistema de robot y cámaras.
Izquierda: Ubicación inicial del efector final.
Derecha: Ubicación final del efector.



3. MODELO DEL ROBOT Y SIMULACION

A partir de las características del robot Mitsubishi RV2AJ de 5 grados de libertad (DOF) Figura 3, se diseñó un robot simulado en Matlab y la librería de robótica de Corke (Corke, 1996) a esté lo llamaremos bloque cinemática del robot, en la fig. 3 se muestran también los parámetros Denavit Hartenberg (DH) respectivos para las 3 primeras articulaciones que es la cantidad de articulaciones requeridas para control de un punto 3D en el espacio. (Hutchison, 2006; Pérez, 2004).

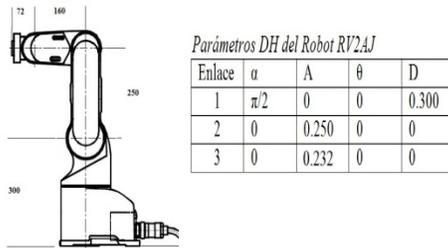


Fig. 3. Dimensiones robot RV2AJ y parámetros DH.

En las Figuras 4a y 4b se muestran los resultados de la simulación del control cartesiano, sin la parte de visión (basada en un ejemplo de la librería de Corke).

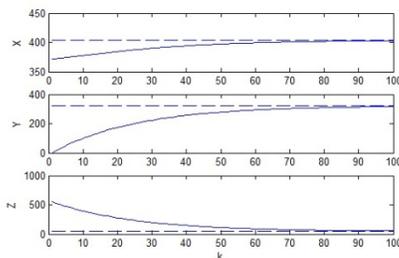


Fig. 4a. Variación del punto x,y,z del robot simulado.

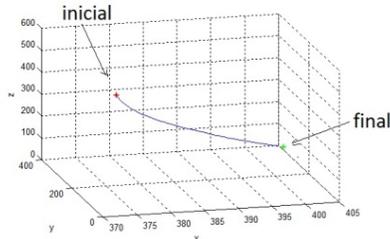


Fig. 4b. Trayectoria en el espacio 3D del robot simulado.

4. IMPLEMENTACIÓN CON EL ROBOT REAL

Además del robot RV2AJ (figura 5), se utilizaron dos cámaras IP AXIS PTZ 214 (resolución de 704x480 píxeles) para la visión estéreo, se enumera a continuación un resumen del procedimiento.

4.1 Calibración de las cámaras:

Se realizaron varias calibraciones usando la librería Matlab desarrollada por Bouget (Bouget, 2000), con lo que se obtuvo los parámetros intrínsecos y extrínsecos, con estos y las parejas (x,y) de puntos de las dos cámaras se tiene una función que

reconstruye o estima el punto 3D en el espacio (bloque de visión estéreo). Luego, conociendo posiciones 3D vistas por las cámaras del efector final del robot se obtiene una función que permite transformar de coordenadas de las cámaras a coordenadas del robot (bloque *transf*).



Fig. 5. Brazo Robótico RV2AJ de prueba (Ejes coordenados de la base del robot)

4.2 Segmentación objetivo y efector final

Para la detección y cálculo de los centroides del efector final y el objetivo, se utilizó una bola anaranjada en la punta del efector y una tapa azul para el objetivo, en la Figura 6 se muestran las segmentaciones para la bola (b) y la tapa (c). Los centroides (x,y) se calcularon para las cuatro imágenes (2 bola, 2 tapa), por promedio de los puntos blancos tanto para x como para y , luego con estos se aplicó una triangulación para obtener los puntos en el espacio (usando el bloque visión estéreo).

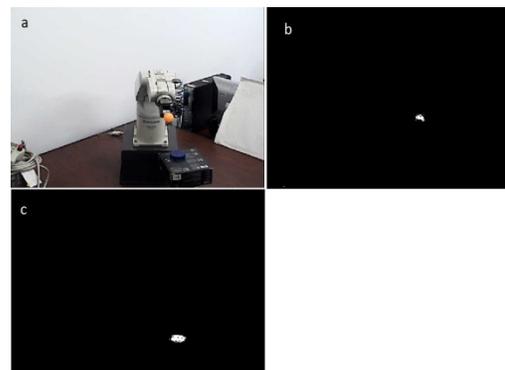


Fig. 6. (a) Imagen normal, (b) segmentación bola (efector final), (c) segmentación tapa azul (objetivo)

4.3 Programa en Java

Como ya se contaba con un driver para comunicación con el robot RV2AJ escrito en java, se decidió trabajar en este lenguaje. Se tomó el esquema de la Figura 7, a partir de él se generó una librería con las funciones de los bloques utilizando la herramienta *builder JA* de Matlab (convertor de

funciones de Matlab a librerías java “.jar”), se reemplazó el bloque de cinemática del robot simulado por los respectivos comandos del robot real, luego se escribió un programa en java, del que se muestra un Algoritmo en pseudocódigo (Fig. 8).

También fue necesario desplazar en el eje z el dato del objetivo para evitar que el efector final chocara con la tapa.

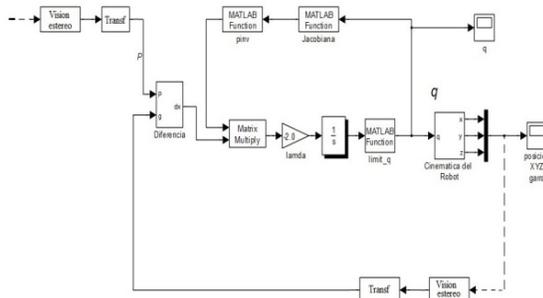


Fig. 7. Esquema de control visual de posición, la línea punteada indica la realimentación visual

En la Figura 7, el bloque *limit_q* es una función que limita los valores calculados de las articulaciones a los rangos del robot real, el bloque *Jacobiana* calcula la matriz jacobiana para un vector de articulaciones y *pinv* realiza la pseudo inversa de la jacobiana.

```

Principal
Inicialización de variables
Ciclo {
    Lectura de cámaras
    Segmentación imágenes garra y objetivo
    Calculo centroides ( Visión estéreo )
    Transformación coord. Robot
    Diferencia y producto matriz
    Jacobiana Inversa
    Calculo del controlador
    Envío de articulaciones al robot
}
Fin
  
```

Fig. 8. Algoritmo de la implementación del control visual.

4.4 Discusión de los resultados

Una diferencia entre la respuesta simulada (fig. 4a y 4b) y la real (fig. 9a y 9b), es que ésta oscila sobre el punto objetivo (sobre todo en el eje x), debido a errores de calibración, por ejemplo se agregó un valor aleatorio en el rango de 0 a 5 mm a la posición que entrega el robot simulado, y se obtuvieron resultados similares al del robot real.

Algunos problemas encontrados fueron la oclusión del sistema garra y bola sobre el objetivo (tapa), cuando la bola se acerca a la tapa la segmentación de la tapa varía, la iluminación que influye sobre los parámetros encontrados de calibración, la lenta respuesta del robot debido a su puerto serie.

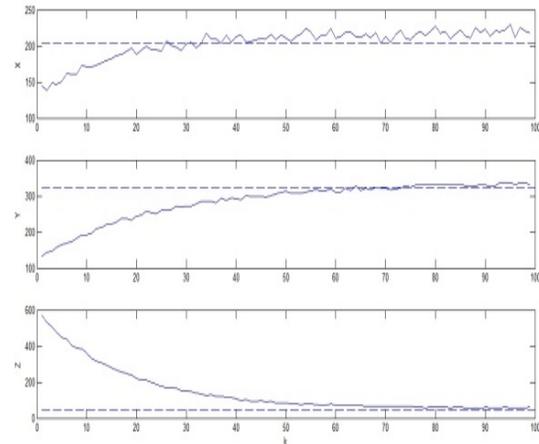


Fig. 9a. Línea punteada: valores x,y,z del objetivo. Línea continua: Variación de x,y,z visto por las cámaras del efector final (transformado a coord. del robot)

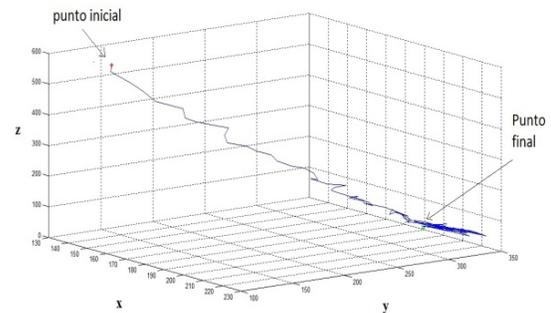


Fig. 9b. Trayectoria del efector final con el robot real.

5. CONCLUSIONES

Se presentó un sistema de control usando realimentación visual, usando dos cámaras y un brazo robótico, se contrastan los resultados simulados con los reales.

El paso entre la simulación y la implementación no fue crítico, gracias al conversor de funciones de Matlab a Java. Es posible usar el Jacobiano del robot simulado en la implementación real, ya que la realimentación elimina las posibles diferencias que se puedan presentar entre la cinemática simulada y la cinemática del robot real.

REFERENCIAS

- Bouget Jean Yves.(2000). Camera calibration toolbox for matlab. Página en internet.
- P.I. Corke. (1996). A robotics toolbox for matlab.IEEE Robotics and Automation Magazine, 3(1):24_32.
- Cyganek B. y Siebert J. P. (2009). An Introduction to 3D Computer Vision Techniques and Algorithms. Wiley and Sons.
- Koichi Hashimoto, Akio Namiki and Masatoshi Ishikawa. (2002). A visuo-motor control architecture for high speed visual servoing. In Workshop on Visual Servoing IROS 2002, pages 45_53, EPFL Switzerland.
- Hutchison S. Chaumette F. (2006).Visual servo control part i: Basic approaches. IEEE Robotics and Automation Magazine, pages 82_90.
- M. Pérez-Cisneros. (2004). Intelligent Model Structures in Visual Servoing. PhD thesis, University of Manchester.
- Pomares J. y Torres F. (2002) Robots y sistemas sensoriales. Editorial Pearson.
- R. Morales, N. García-Aracil, C. Perez, J.M. Sabater, J.M. Azorin, (2007). Sistema de Realimentación Visual de Fuerzas en entornos de Cirugía Asistida por Robots, XXVIII Jornadas de Automática, Huelva - España.
- W. Wilson, C. Hulls, and G. Bell, (1996). Relative end-effector control using cartesian position based visual servoing, *IEEE Trans. Robot. Automat.*, vol. 12, pp. 684–696.
- Xu De et al, (2005) .Hybrid visual servoing control for robotic arc welding based on structured high vision, *Acta Automatica*, V31.N4.
- Hager Gregory D., Wen-Chung Chang, and A.S. Morse (1995), Robot Hand-Eye Coordination Based on Stereo Vision, *IEEE Control Systems*.