

Sistema de servocontrol visual empleando redes neuronales y filtros en el dominio de CIELAB

Visual servo-control system using neural networks and filters based on CIELAB

Germán Buitrago-Salazar, Olga Ramos-Sandoval

Universidad Militar Nueva Granada, Facultad de Ingeniería en Mecatrónica, Bogotá, Colombia

Correo electrónico: gedabusa@gmail.com

Recibido: 15 de febrero de 2015

Aceptado: 28 de abril de 2015

Resumen

En este trabajo se presentan los resultados de un sistema servocontrol visual de un brazo robótico de seis grados de libertad. Para esto, se utiliza una red neuronal de tipo *feed forward*, entrenada por *back propagation*, para determinar la distancia entre el brazo robótico y un objeto de referencia, que permite ubicarlo en un espacio de trabajo. Las entradas de la red corresponden a la información obtenida de las imágenes capturadas por el *Kinect*, utilizando un filtro que discrimina la posición de los elementos, en el espacio de color CIELAB (*Commission Internationale de l'Eclairage L*a*b components*). El resultado de esta investigación

demonstró que la distancia estimada por la red tiene un margen de error menor, que el algoritmo propuesto en otros trabajos. Igualmente, se probó que el sistema de procesamiento de imágenes es más robusto a ruidos digitales, en comparación con los sistemas que utilizan filtros en el dominio RGB (*Red-Green-Blue*).

Palabras claves: sistema de servocontrol visual, CIELAB, redes neuronales, filtrado de imágenes.

Abstract

In this paper the results of visual servo-control system for a robotic arm with six degrees of freedom are presented. For this purpose, a feed forward neural network, which was trained by back propagation, is used to determine the distance between the robot arm and a reference object and sitting the robot in the workspace. The inputs of neural network correspond to the information obtained from the images captured by the *Kinect*, using a filter that discriminates the position of the elements in the CIELAB (*Commission Internationale de l'Eclairage L*a*b components*) color

space. The result of this research showed that the estimated distance with the network has an errorless than the algorithm proposed in other works. Similarly, it was proved that the image processing system is more robust to digital noise, compared to systems using filters in RGB (*Red-Green-Blue*).

Key words: visual servo-control system, CIELAB, neural networks, image filtering

Introducción

El sistema de servocontrol visual es un esquema de control donde utilizan cámaras para realimentar la ubicación de los elementos, mientras que son gobernados los movimientos articulares de un brazo robótico. Los sistemas típicos usan una o dos cámaras fijas sobre el manipulador o colocadas en el espacio de trabajo [1]. En la primera configuración, los dispositivos son situados sobre el efector final del manipulador con el objetivo de observar la posición de un objeto de referencia o guía. Esta configuración también es llamado *Eye-in-hand*. En la segunda, las cámaras son colocadas dentro del entorno y relacionan la postura del dispositivo robótico con un elemento de referencia [2]. En trabajos realizados, son diseñados este tipo de sistemas de control con modelos híbridos basados en la matriz Jacobiana de la cámara, modelos de Lyapunov y lógica difusa con métodos clásicos de control [3-4].

Para realizar estos trabajos, se han usado sistemas de visión de máquina para tomar las imágenes y extraer la información necesaria. Un caso particular de esto, ha sido la estimación de posición de elementos, para lo cual, se han propuesto soluciones como el uso de marcadores luminosos. Estos marcadores son detectados por un algoritmo de estimación basado en filtros no lineales, teniendo en cuenta la distorsión presente en la imagen [5]. Igualmente, desde una perspectiva geométrica, se ha empleado modelos epipolares que extraen la posición de los objetos utilizando dos imágenes de la misma escena desde diferentes puntos de vista [6]. Otras propuestas consideran una función para minimizar el error presente por la distorsión de la imagen, mientras que de forma iterativa es calculada la posición óptima del objeto [7] y la aplicación de un modelo parabólico para calcular la postura de la cámara a partir de una imagen de referencia [8].

De igual forma, los sistemas de visión de máquina han sido empleados en el área industrial, en conjunto con los sistemas de servocontrol, para el reconocimiento de patrones y la supervisión de procesos. Diferentes desarrollos han usado la visión de máquina en procesos de inspección y control de calidad con imágenes tomadas en el área de producción [9]. De la misma manera, se han usado para el posicionamiento X-Y de plataformas industriales [10], la planeación de trayectorias de robots [11] y el movimiento de manipuladores

[12]. A pesar de esto, los sistemas de visión involucran dispositivos y plataformas de alto costo, y que en ciertos casos, no proporcionan la velocidad de procesamiento adecuada para desempeñar tareas en tiempo real.

De acuerdo a esto, la aplicación de nuevos sensores para la adquisición de imágenes como el *Kinect*, permite un reconocimiento más rápido del espacio de trabajo, con costos de operación menores comparados con los sistemas de visión estereoscópica comunes y con un nivel alto en la fiabilidad y velocidad de procesamiento de la información[13]. Para localizar la posición de los puntos reflejados sobre los píxeles de la imagen, en [14] se presenta un algoritmo que utiliza las imágenes RGB-D (*Red-Blue-Greeny Depth*), que son las representaciones de color y profundidad de un fotograma tomado [15], con un tiempo de procesamiento de inferior a . Este lapso depende de la cantidad de información y la aplicación que se realizó con el sensor.

El objetivo de este trabajo es presentar un algoritmo de inteligencia artificial que estime la distancia presente entre dos elementos en el espacio de trabajo, a partir de la ubicación de los objetos dentro de la imagen y el mapa de profundidad del dispositivo *Kinect*, aplicado a un sistema de servocontrol visual. Este proceso fue enfocado a la disminución de los tiempos de procesamiento de la información, gracias al trabajo en paralelo de la red, con dispositivos de bajo costo, mejorando la velocidad de los robots cuando realizan cierta actividad específica dentro de los procesos industriales.

Métodos

El desarrollo de este sistema de servocontrol visual se divide en tres etapas. En la primera se presenta el procesamiento de imágenes para extraer la información cromática y de profundidad del brazo robótico Mitsubishi de 6 grados de libertad y el objeto guía en el dominio de CIELAB, a partir de las imágenes tomadas del *Kinect*. En la segunda se explica el proceso y parámetros considerados para la creación, entrenamiento y funcionamiento de la red neuronal para la estimación de la posición de los objetos. Finalmente, en la tercera etapa se presenta el sistema de control de movimiento, el modelo articular del manipulador y el diseño del controlador PD (proporcional-derivativo).

Procesamiento de la imagen

El *Kinect* es un sensor 3D compuesto por una cámara RGB, para capturar las imágenes del entorno de trabajo, y una cámara infrarroja con su respectivo proyector, que mide la distancia existente entre el dispositivo y los diferentes elementos, como se observa en figura1. El *Kinect* está ubicado sobre la celda de trabajo del brazo robótico, orientada hacia la base del manipulador y configurada para que ambas imágenes estén a una resolución de 640x480. La imagen RGB-D presenta una desventaja en cuanto a su componente de color, puesto que la imagen en este espacio presenta ruido causado por la correlación cercana entre los niveles de iluminación e intensidad de los colores que se desea reconocer [16].

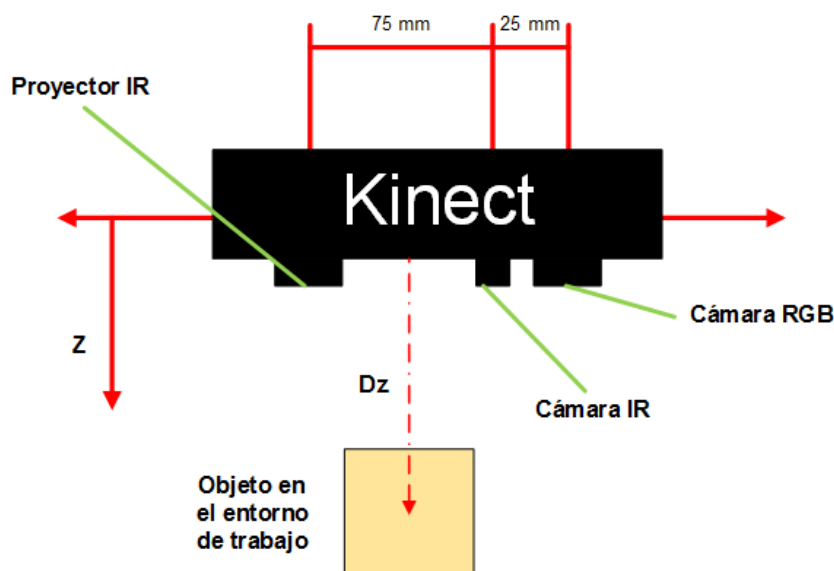


Fig. 1. Estructura del Kinect

De acuerdo a esto, para mejorar la calidad de la imagen, ésta es transformada a la representación tricromática XYZ y posteriormente a la representación CIELAB, para distinguir los colores en un espacio de color más uniforme. Para la primera se utiliza la ecuación 1, donde (i,j) es la posición de cada uno de los píxeles dentro de la imagen.

$$\begin{bmatrix} X_{(i,j)} \\ Y_{(i,j)} \\ Z_{(i,j)} \end{bmatrix} = \begin{bmatrix} 0,4303 & 0,3416 & 0,1784 \\ 0,2219 & 0,7068 & 0,0713 \\ 0,0202 & 0,1296 & 0,9393 \end{bmatrix} \begin{bmatrix} R_{(i,j)} \\ G_{(i,j)} \\ B_{(i,j)} \end{bmatrix} \quad (1)$$

La transformación directa entre la representación XYZ y CIELAB, donde los niveles de luminosidad son L y sus posiciones entre colores opuestos son a, b de acuerdo a [17], están dadas por:

$$L_{(i,j)} = 116f\left(\frac{Y_{(i,j)}}{Y_0}\right) - 16 \quad (2)$$

$$a_{(i,j)} = 500 \left[f\left(\frac{X_{(i,j)}}{X_0}\right) - f\left(\frac{Y_{(i,j)}}{Y_0}\right) \right] \quad (3)$$

$$b_{(i,j)} = 200 \left[f\left(\frac{Y_{(i,j)}}{Y_0}\right) - f\left(\frac{Z_{(i,j)}}{Z_0}\right) \right] \quad (4)$$

Donde,

$$f(t) = \begin{cases} \sqrt[3]{t} & t > 0,008856 \\ 7,787t + \frac{16}{116} & t \leq 0,008856 \end{cases} \quad (5)$$

Las constantes X_0, Y_0, Z_0 son dependientes de la muestra de blanco. Como se menciona en [18], para la CIE Illuminant D65 estos valores son 95,05, 100,00 y 108,90 respectivamente. Para la extracción de las regiones con la tonalidad de color de los marcadores, es aplicado un filtro que relaciona la distancia euclidiana de color de los pixeles $F_{(i,j)}$, respecto a una muestra de referencia $P_{ref}(L_{ref}, a_{ref}, b_{ref})$, aplicando la ecuación 6.

$$F_{(i,j)} = \sqrt{(L_{ref} - L_{(i,j)})^2 + (a_{ref} - a_{(i,j)})^2 + (b_{ref} - b_{(i,j)})^2} \quad (6)$$

Tomando una ventana w con tamaño $(2N+1) \times (2N+1)$ centrado en el pixel (i,j) , se relaciona los valores de la vecindad, promediando la norma de cada uno y expresando la salida del proceso $IF_{(i,j)}$ en la ecuación 7.

$$IF_{(i,j)} = \sum_{s=1}^{2N} \sum_{t=1}^{2N} F_{(i+s,j+t)} W_{(s,t)} \quad (7)$$

Luego, el filtro umbraliza, extrae las regiones en donde se ubica el marcador y ubica el valor del centroide de las mismas. Con la ubicación de los marcadores del robot y del objeto de referencia dentro de la imagen, se obtiene del mapa de profundidad de la *Kinect*, la distancia de la cámara a estos objetos, formando los vectores $q_r = [u_r, v_r, z_r]^T$ y $q_o = [u_o, v_o, z_o]^T$, los cuales representan la posición de los elementos y son las variables de entrada de la red neuronal.

Modelo matemático de la red neuronal

Una red neuronal está basada en un modelo bayesiano que emula el procesamiento de información del sistema nervioso y es empleada para la clasificación, optimización, predicción y reconocimiento de patrones [19]. Esta red se compone de neuronas artificiales, que a su vez son compuestas por una n cantidad de entradas conectadas a un mismo nodo. A cada entrada se le asocia un peso sináptico que corresponde al nivel de excitación o inhibición entre ellas. Adicional a esto, se le suma un bias, que umbraliza la salida de la neurona. Luego, una función de activación es aplicada por el valor devuelto por la sumatoria de las entradas con el fin de acotar la información de la neurona.

De acuerdo a [20], el modelo matemático lineal de la neurona se representa por la ecuación 8, donde W_i es el peso sináptico para la entrada t_i , $f(a)$ es la función de activación, m es la salida de la neurona y b es el bias.

$$m = f\left(b + \sum_{i=1}^n W_i t_i\right) \quad (8)$$

Como se evidencia en la figura 2, la red neuronal programada se compone de dos capas ocultas y una capa de salida. La primera capa oculta está formada por G_1 neuronas, conectadas a IR entradas, a través de la matriz con los pesos sinápticos de las neuronas W_1 . La matriz bias B_1 es de tamaño $G_1 \times 1$. La segunda capa

tiene G_2 neuronas y sus matrices con los pesos y los bias están representados por las letras W_2 y B_2 respectivamente. La función de activación de ambas capas es la tangente hiperbólica que limita las salidas entre $[-1; 1]$.

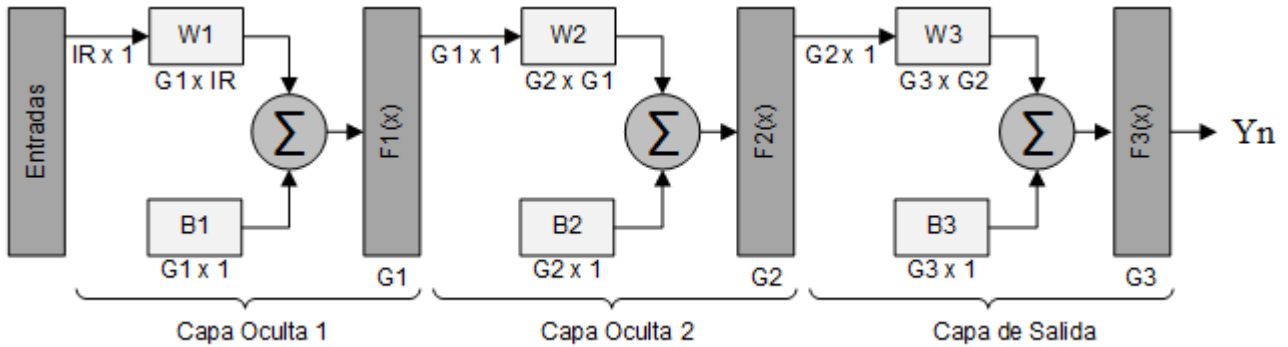


Fig. 2. Arquitectura de la Red Neuronal con dos capas ocultas

Por otro lado, la capa de salida está compuesta por 3 neuronas que representan cada una de las componentes vectoriales de la distancia estimada, por lo cual, las matrices con pesos W_3 y bias B_3 son de tamaño $3 \times G_2$ y 3×1 respectivamente. Asimismo, la función de activación es una función lineal. De la figura 2, la ecuación 8 y la anterior descripción, la salida de la red neuronal Y_n está dada por:

$$Y_n = f_3(f_2(f_1(IW_1 + B_1)W_2 + B_2)W_3 + B_3) \quad (9)$$

Para la etapa de aprendizaje de la red neuronal, se emplea el método de *backpropagation*. Este método actualiza los pesos y bias de cada una de las neuronas, minimizando el error entre la salida de la red t_m y los valores esperados o_m , cuando es colocado cierto valor en la capa de empezando en la capa de salida y terminando en la capa de entrada. De acuerdo al planteamiento de [21], la actualización de los pesos y bias están dadas por:

$$W_l[t+1] = W_l[t] - \alpha \delta^l \quad (10)$$

$$B_l[t+1] = B_l[t] - \alpha \delta^l \quad (11)$$

Donde α es la variación de aprendizaje en función de la iteración de entrenamiento de la red neuronal, l la posición de la capa dentro de la red neuronal, I^l es el vector con las entradas de las neuronas en la capa l y δ_l es el vector con los valores de sensibilidad de la capa l . Para la capa de salida, la sensibilidad en cada una de las neuronas está representada por la ecuación 12, mientras que para las capas ocultas, ese valor se refleja en la ecuación 13.

$$\delta^l[k] = o_m - t_m \quad (12)$$

$$\delta^l[k] = (1 - O^l[k]^2) \sum_{k=1}^n \delta^{l+1}[k] W_{l+1}[k] \quad (13)$$

Siendo n la cantidad de neuronas de la capa l y O^l la matriz con las salidas de las neuronas de la capa l .

Sistema de control del manipulador

Los movimientos del manipulador se gobiernan con un controlador PD que suaviza la velocidad de los motores en los cambios de postura del robot. Este controlador toma las señales de error por cada imagen sensada $e(f)$ entre la referencia del sistema y la distancia evaluada por la red neuronal, como se presenta en el esquema de la figura 3.

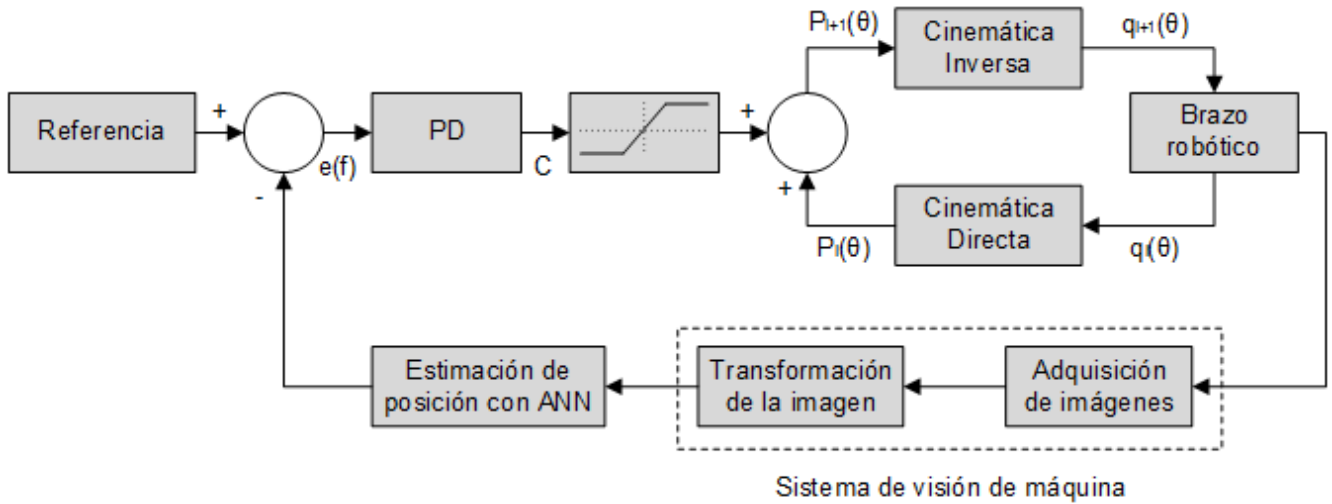


Fig.3. Arquitectura del sistema de control del Manipulador

El controlador proporcional k_p y derivativo k_d son matrices diagonales, cuyos componentes relacionan los ejes X, Y, Z del sistema de coordenadas. Estas matrices se referencian en las ecuaciones 14 y 15 respectivamente.

$$k_p = \begin{bmatrix} k_{p_x} & 0 & 0 \\ 0 & k_{p_y} & 0 \\ 0 & 0 & k_{p_z} \end{bmatrix} \quad (14)$$

$$k_d = \begin{bmatrix} k_{d_x} & 0 & 0 \\ 0 & k_{d_y} & 0 \\ 0 & 0 & k_{d_z} \end{bmatrix} \quad (15)$$

De acuerdo a esto, la ley de control para el sistema de servocontrol visual se presenta en la ecuación 16(16), donde $[C_x \ C_y \ C_z]^T$ es la señal de control.

$$\begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = k_p * e(f) + k_d * \frac{de(f)}{df} \quad (16)$$

La nueva pose que debe asumir el robot P_{i+1} se representa en la ecuación, donde P_i es la posición del robot recibida desde el controlador del dispositivo. Finalmente, para enviar la información al manipulador, es necesario aplicarlos parámetros de Denavit-Hartenberg[22], para calcular el modelo cinemático inverso del manipulador. Los parámetros están presentados en la tabla 1.

$$\begin{bmatrix} P_{i+1_x} \\ P_{i+1_y} \\ P_{i+1_z} \end{bmatrix} = \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} + \begin{bmatrix} P_{i_x} \\ P_{i_y} \\ P_{i_z} \end{bmatrix} \quad (17)$$

Tabla 1. Parámetros de Denavit-Hartenberg para el brazo robótico de 6 grados de libertad RV-3SQB

Articulación	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	$-\pi/2$
2	$\theta_2 - \pi/2$	0	a_2	0
3	$\theta_3 - \pi/2$	0	a_3	$-\pi/2$
4	θ_4	d_4	0	$\pi/2$
5	θ_5	0	0	$-\pi/2$
6	θ_6	d_6	0	0

Resultados

Los sistemas de servocontrol visual usan como dispositivo de control, una cámara que recoge con imágenes, la posición de los elementos reflejada sobre su campo visual. Para discriminar la ubicación de los elementos dentro del entorno. Este trabajo presentó una red neuronal compuesta por dos capas ocultas y una de salida. La primera capa de la ANN tiene 70 neuronas conectadas a las entradas, mientras que la segunda se creó con 40 neuronas. Las capas conectadas en cascada, son entrenadas por *back propagation*, con $\alpha = 0,1$. Para entrenar la red neuronal fueron tomadas 9000 muestras, las cuales evitan que la red sea sobreentrenada y no funcione correctamente.

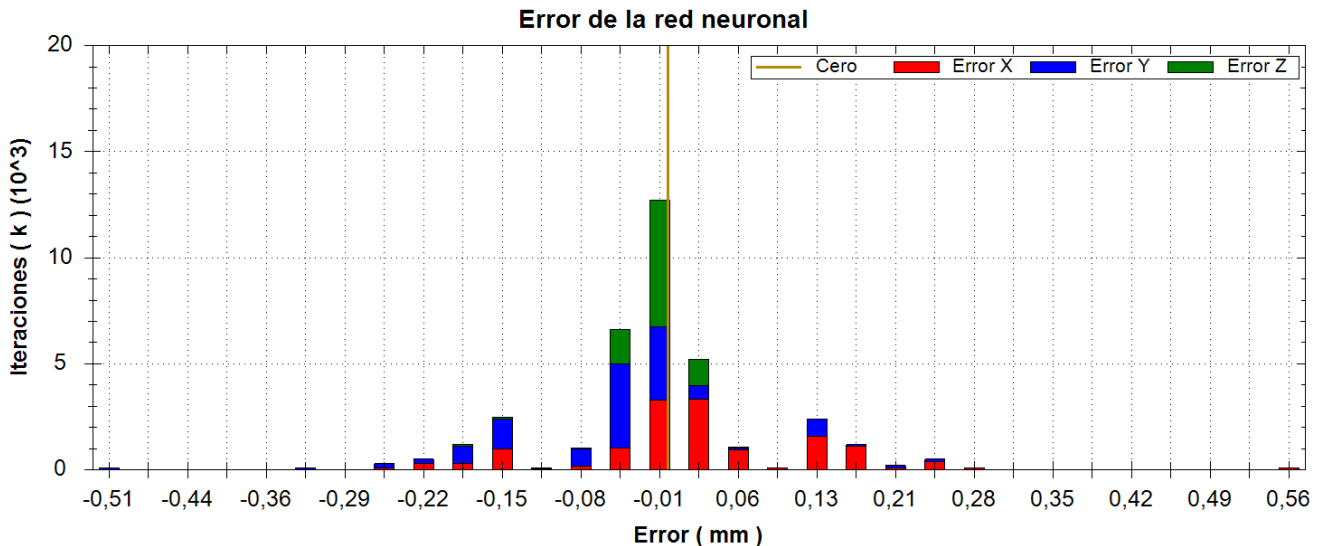


Fig. 4. Error final de la red neuronal después del entrenamiento por Backpropagation

En la figura 4, se presenta el error del entrenamiento de la red, que relaciona las salidas de ella respecto a los valores esperados. Es apreciable que la tendencia del error es cero y la desviación estándar para cada una de las componentes σ , es $[0,0357 \ 0,0299 \ 0,0204]^T$. Este resultado implica que la red se asemeja, en un alto índice de precisión, a la función de estimación de la distancia entre el robot manipulador y el objeto guía.

A continuación, se programa una interfaz de usuario en C#, que sincroniza el procesamiento de las imágenes, la simulación de la red neuronal entrenada, la comunicación con el manipulador y el algoritmo de control de movimiento del brazo robótico. Utilizando dos marcadores, uno de color verde para el manipulador y otro de color rojo para usarlo como objeto guía, la trayectoria del robot es trazada, manteniendo la distancia entre los dos elementos.

El programa realizado en C# puede ser descargado desde: <https://github.com/gedabusa/ServoRM>

En la figura 5, se ve el recorrido de ambos elementos dentro del espacio de trabajo. El desfase entre ambas trayectorias se debe a dos razones. La primera es los valores de referencias del sistema de control, debido a que el manipulador debía mantener una distancia de -130, 140 y 100 mm respecto a la posición del marcador visual. La segunda razón del desfase se debe a la posición de la cámara. Como la cámara fue ubicada dentro del espacio de trabajo, existían puntos ciegos donde el objeto guía quedaba oculto por el manipulador y viceversa. Para solucionar esto, se decidió que la distancia entre ambos marcadores no debía ser inferior a 100 mm en cada una de sus componentes coordenadas.

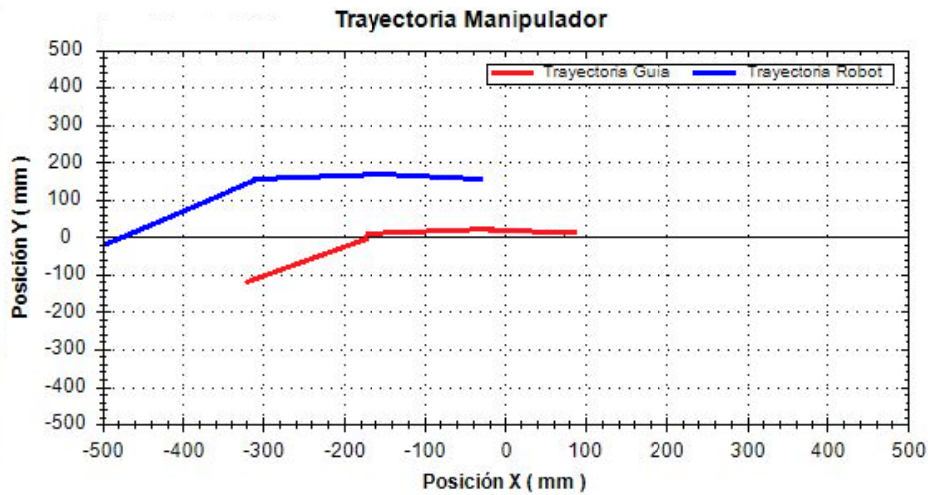


Fig. 5. Trayectorias del brazo robótico y del objeto de referencia

Finalmente, cuando el sistema de control se estabiliza en el punto final de la trayectoria, se forman fluctuaciones en la posición del manipulador, como se presenta en la figura 6. Estas oscilaciones son provocadas por los movimientos del robot, que son sensibles a cambios pequeños en su valor.

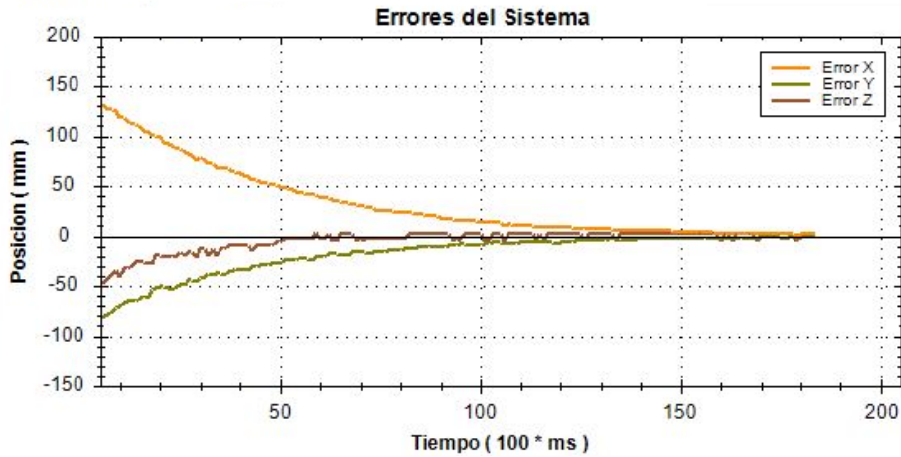


Fig. 6. Error del sistema de servocontrol cuando el manipulador se estabiliza en el punto final de la trayectoria

Discusión

Los resultados tomados de la trayectoria realizada con el sistema de servocontrol, fueron comparados con la posición obtenida en cada instante de tiempo, basados en la función de conversión de sistemas de coordenadas de [14], la cual se observa en la ecuación 18, y modificada para calcular la distancia entre dos elementos en la imagen, a partir de la posición del elemento y la distancia asociada a cada pixel de la imagen. Esta última está representada en la ecuación 19.

$$P(x, y, z) = \begin{cases} X = \frac{2 * \tan(28,5) * u_{(i,j)} * z_{(i,j)}}{640} \\ Y = \frac{2 * \tan(21,5) * v_{(i,j)} * z_{(i,j)}}{480} \\ Z = z_{(i,j)} \end{cases} \quad (18)$$

$$d(x, y, z) = \begin{cases} X_{or} = \frac{2 * \tan(28,5) * (u_r * z_r - u_o * z_o)}{640} \\ Y_{or} = \frac{2 * \tan(21,5) * (v_r * z_r - v_o * z_o)}{480} \\ Z_{or} = z_r - z_o \end{cases} \quad (19)$$

En la figura 7, se presentan los errores entre la trayectoria generada por la red neuronal y el algoritmo de conversión presentado en la ecuación 19, con la distancia real entre los dos elementos para una trayectoria descrita por el objeto guía. Es apreciable que el error entre la distancia real de los objetos y el algoritmo modificado de [14] es inferior a 0,6 mm, mientras que el error entre la distancia real y la red neuronal no supera a 0,5 mm.

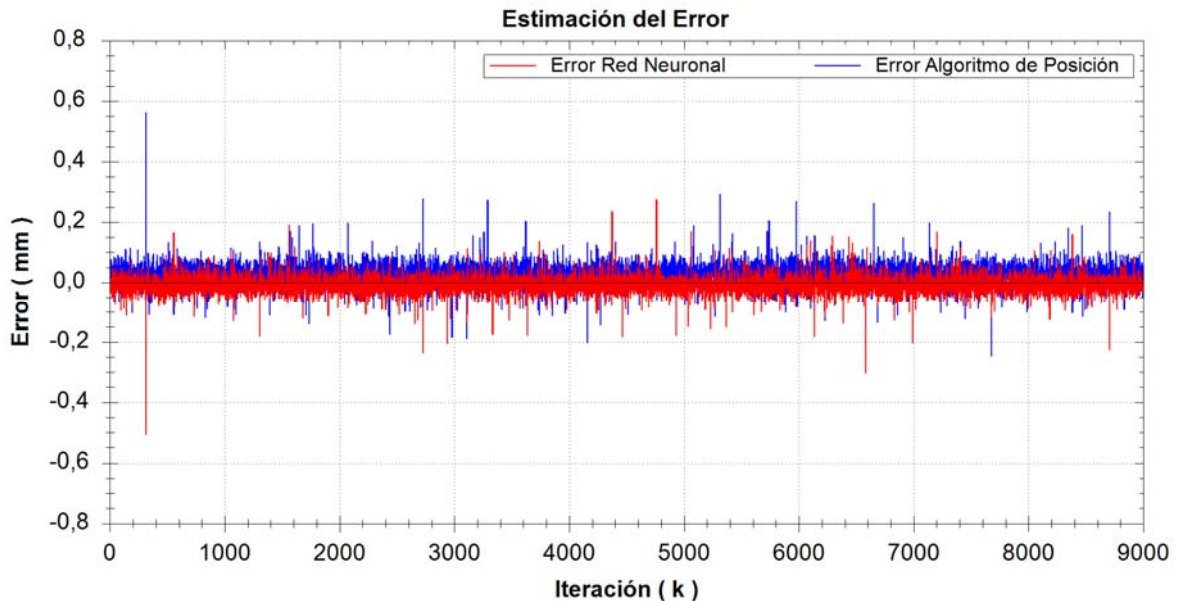


Fig. 7. Error presente en la salida de la red neuronal y el algoritmo matemático para una trayectoria helicoidal

En la tabla 2, se presentan los errores acumulativo absoluto y promedio para los dos algoritmos de estimación de posición, donde la trayectoria calculada por la red neuronal es más precisa que la evaluada por el algoritmo modificado. Este desempeño proporciona un método más veloz en los procesos industriales, donde a pesar de utilizar el mismo dispositivo para captar las imágenes del entorno, la velocidad del procesamiento de la red neuronal, permite que la imagen obtenida del *Kinect* sea procesada en un tiempo menor, antes de obtener el siguiente fotograma del dispositivo.

Tabla 2. Error entre los algoritmos de estimación de distancias y la distancia real

Error	Algoritmo modificado de [12]	Red Neuronal
Acumulado absoluto	315,198 mm	210,830 mm
Promedio Total	0,035 mm/iteración	0,023 mm/iteración

Conclusiones

La aplicación del Kinect en sistemas de servocontrol visual, mejora la velocidad de procesamiento, respecto a dispositivos de visión como las cámaras web y sistemas de visión convencionales, lo que con lleva a que el sistema controle, con retardos mínimos, la trayectoria descrita del robot. Adicional a esto, por ser un dispositivo de bajo costo, respecto a sistemas de visión estereoscópica y compatible con varias plataformas de programación, puede ser utilizado fácilmente para la detección de elementos en procesos industriales y sistemas de control de calidad.

Por otro lado, se comprobó que el filtro en el espacio de color de CIELAB respecto al filtro en RGB, disminuye el ruido presente en la imagen causado por la mezcla de la información de policromática de los componentes encontrados en el espacio de trabajo del robot en un proceso industrial, al igual que por los niveles de iluminación, intensidad y brillo reflejados dentro de la imagen, aunque aumenta el tiempo de procesamiento de la imagen.

Finalmente, la aplicación de redes neuronales para detectar la distancia entre objetos, presenta un error bajo respecto al algoritmo de evaluación producto de la investigación de [14]. Con una diferencia de error máxima de 0.1 mm entre ambas, la estimación de posición con ANN presenta una mayor precisión, con menor tiempo de procesamiento en la evaluación de la red.

Agradecimientos

Los autores agradecen a la Vicerrectoría de Investigaciones de la Universidad Militar Nueva Granada por el apoyo y la financiación del proyecto IMP-ING 1573 en el año 2014.

Referencias

1. Chaumette F y Hutchinson S. Visual Servo Control Part I: Basic Approaches. *IEEE Robotics & Automation Magazine*. 2006;13(4):82-90. ISSN 1070-9932. DOI 10.1109/MRA.2006.250573.
2. Hutchinson S, Hager G y Corke P. A Tutorial on Visual Servo Control. *IEEE Transactions on Robotics and Automation*. 1996;12(5):651-70. ISSN 1042-296X. DOI 10.1109/70.538972.
3. Marey M y Chaumette F. Analysis of Classical and New Visual Servoing Control Laws. In: *IEEE International Conference on Robotics and Automation*; CA, EEUU: IEEE; 2008. DOI 10.1109/ROBOT.2008.4543705. [Citado 25 de abril de 2015] Disponible en: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4543705>
4. Jie D, Yu H y Kaixiang P. Robot Visual Servo Control Based on Fuzzy Adaptive PID. In: *2012 International Conference on Systems and Informatics (ICSAI)*; Yantai, China: IEEE; 2012. DOI 10.1109/ICSAI.2012.6223282. [Citado 24 de abril de 2015] Disponible en: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6223282>
5. Riipkema E, Muthukrishnan K, Dulman S, et al. Pose Estimation with radio-controlled visual markers. In: *2010 IEEE 7th International Conference on Mobile Adhoc ans Sensor Systems (MASS)*; CA, EEUU: IEEE; 2010. DOI 10.1109/MASS.2010.5663786. [Citado 26 de abril de 2015] Disponible en: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5663786>
6. Mattar E. Robotics Arm Visual Servo: Estimation of Arm-Space Kinematics Relations with Epipolar Geometry. In: *Robotics Systems - Applications, Control and Programming Rijeka, Croatia: InTech*; 2012. DOI 10.5772/25605. [Citado 23 de abril de 2015] Disponible en: <http://www.intechopen.com/books/robotic-systems-applications-control-and-programming/robotics-arm-visual-servo-estimation-of-arm-space-kinematics-relations-with-epipolar-geometry>
7. Long L, Zong Quan D, Bing L, et al. Fast vision-based pose estimation iterative algorithm. *Optik - International for Light and Electron Optics*. 2013;124(12):1116-21. ISSN 0030-4026. DOI 10.1016/j.ijleo.2012.03.018.
8. Wei L, Wenxiao S, Yaowen L, et al. A novel method of camera pose estimation by parabolic motion. *Optik - International Journal for Light and Electron Optics*. 2013;124(24):6840-5. ISSN 0030-4026. DOI 10.1016/j.ijleo.2013.05.074.
9. Jaramillo A, Jiménez R y Ramos O. Inspección de calidad para un sistema de producción industrial basado en el procesamiento de imágenes. *Tecnura*. 2014;18(41):76-90. ISSN 0123-921X. DOI 10.14483/udistrital.jour.tecnura.2014.3.a06.
10. Huang S y Lin S. Application of visual servo-control X-Y table in color filter cross mark alignment. *Sensors and Actuators A: Physical*. 2009;152(1):53-62. ISSN 0924-4247. DOI 10.1016/j.sna.2009.03.014.
11. Kohrt C, Stamp R, Pipe AG, et al. An online robot trajectory planning and programming support system for industrial use. *Robotics and Computer-Integrated Manufacturing*. 2013;29(1):71-9. ISSN 0736-5845. DOI 10.1016/j.rcim.2012.07.010.
12. De la Fuente E y Trespaderne F. Detección Eficiente de Elipses en Imágenes. Aplicación al Posicionamiento 3D de un Robot Industrial. *Revista Iberoamericana de Automática e Informática Industrial*. 2012;9(4):419-28. ISSN 1697-7912. DOI 10.1016/j.riai.2012.09.005.
13. Smisek J, Jancosek M y Pajdla T. 3D with Kinect. In: *2011 IEEE International Conference on Computer Vision Workshops*; Barcelona, España: IEEE; 2011. DOI 10.1109/ICCVW.2011.6130380. [Citado 25 de abril de 2015] Disponible en: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6130380>
14. Ballester J y Pheatt C. Using the Xbox Kinect sensor for positional data acquisition. *American Journal of Physics*. 2013;81(1):71-7. ISSN 0002-9505. DOI 10.1119/1.4748853.
15. Hernández J, Quintanilla A, López J, et al. Detecting objects using color and depth segmentation with Kinect sensor. *Procedia Technology*. 2012;3:196-204. ISSN 2212-0173. DOI 10.1016/j.protcy.2012.03.021.
16. Cheng H, Jiang X, Sun Y, et al. Color image segmentation: advances and prospects. *Pattern Recognition*. 2011;34(12):2259-81. ISSN 0031-3203. DOI 10.1016/S0031-3203(00)00149-7.
17. Connolly C y Fliess T. A Study of Efficiency and Accuracy in the Transformation from RGB to CIELAB Color Space. *IEEE Transactions on Image Processing*. 1997;6(7):1046-8. ISSN 1057-7149. DOI 10.1109/83.597279.
18. Laparra V, Jiménez S, Camps Valls G, et al. Nonlinearities and Adaptation of Color Vision from Sequential Principal Curves Analysis. *Neural Computing*. 2009;24(10):2751-88. ISSN 0899-7667. DOI 10.1162/NECO_a_00342.
19. Khosravi A, Nahavandi S, Creighton D, et al. Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances. *IEEE Transactions on Neural Networks*. 2011;22(9):1341-56. ISSN 1045-9227. DOI 10.1109/TNN.2011.2162110.
20. Chojaczyk A, Texeira A, Neves L, et al. Review and application of Artificial Neural Networks models in reliability analysis of steel structures. *Structural Safety*. 2015;52:78-9. ISSN 0167-4730. DOI 10.1016/j.strusafe.2014.09.002.
21. Hasan M, Rahaman A, Talukder M, et al. Segmented Hidden NN - An Improved Structure of Feed Forward NN. In: *3rd International Conference on Informatics, Electronics & Vision*; Dhaka, Bangladesh: IEEE; 2014. DOI 10.1109/ICIEV.2014.6850688. [Citado 27 de abril de 2015] Disponible en: <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6850688>
22. Denavit J y Hertenberg R. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanism*. 1955;22:215-21. ISSN 0021-8936.