

Aplicación de la lógica difusa en la visión artificial bidimensional del robot humanoide de carrera

Application of a fuzzy controller in the bidimensional artificial vision of racing humanoid robot

Aplicação da lógica fuzzy na visão bidimensional do corrida de robô humanoide

Luis E. Beltrán Casana¹ y Luis A. Vargas Díaz²

Resumen

El fin de la investigación fue desarrollar un controlador difuso en la visión artificial bidimensional del robot humanoide de carrera para que mejore su locomoción en el seguimiento de trayectorias con respecto a un controlador clásico. Se diseñó y construyó el robot bípedo de carrera Mecatrón, de ocho grados de libertad rotatorios; su estructura fue de aluminio, con ocho servomotores insertados en los ejes articulados, de los cuales seis controlan la locomoción bípeda y dos la dirección. El robot presenta una cámara web para dotarle de un sistema de visión artificial bidimensional programado en MatLab; a partir de los datos obtenidos se aplica un controlador difuso también en MatLab, usando el toolbox de Fuzzy Logic. Los resultados evidencian que el robot bípedo Mecatrón usando el controlador difuso en la visión artificial recorrió la trayectoria recta en 72 segundos y usando el controlador clásico lo recorre en 91 segundos. Además el robot se desvió una distancia máxima de 4.5 cm respecto a la línea en 75 segundos, mientras que implementado con un controlador clásico se desvió 12 cm en 83 segundos.

Palabras clave: Lógica difusa, robótica, visión artificial.

Abstract

The purpose of the research was to develop a diffuse driver in two-dimensional vision of robot humanoid race that improves its locomotion in the monitoring of trajectories with respect to a classic controller. This was designed and built a biped robot race Mecatron, eight rotational degrees of freedom; its structure was aluminium with eight actuators inserted into articulated shafts, of which six control biped locomotion and two change direction. The robot presents a web camera to provide a two-dimensional artificial vision system programmed in MatLab; from the obtained data applies a diffuse controller in MatLab, using the Fuzzy Logic toolbox. The results demonstrate the biped robot Mecatron using the diffuse controller in machine vision toured the path straight in 72 seconds and using the classic controller runs it in 91 seconds. In addition the robot drifted to a maximum distance of 4.5 cm connection line in 75 seconds, while the same robot implemented with a classic driver deviated 12 cm in 83 seconds.

Keywords: Robotics, fuzzy logic, artificial vision.

Resumo

O objetivo da pesquisa foi desenvolver um driver difuso na visão bidimensional da raça de robôs humanóides que melhora sua locomoção no acompanhamento das trajetórias em relação a um controle clássico. Isto foi projetado e construído de uma corrida de robô bípede Mecatrón, oito graus de liberdade rotacionais; sua estrutura era de alumínio com oito atuadores inserido em eixos articulados, dos quais seis controle dois e locomoção bípede mudam de direção. O robô apresenta uma câmera da web para fornecer um sistema de visão artificial bidimensional programado em MatLab; a partir dos dados obtidos, aplica-se um controlador difuso no MatLab, usando o toolbox de lógica Fuzzy. Os resultados demonstram que o robô bípede Mecatron usando o controlador difuso na visão de máquina percorreu o caminho direto em 72 segundos e usando o controle clássico executou em 91 segundos. Além do robô degradado a uma distância máxima da linha de conexão de 4,5 cm em 75 segundos, enquanto o robô implementado com um driver clássico desviou 12 cm em 83 segundos.

Palavras-chave: Robótica, lógica fuzzy, visão artificial.

¹Universidad Particular del Norte, Facultad de Ingeniería, lebcoly@gmail.com

²Universidad Nacional de Trujillo, Facultad de Ingeniería, Escuela de Ingeniería Mecatrónica

Recibido, 2 de diciembre de 2015

Aceptado, 5 de enero de 2016

Introducción

En los últimos años, el uso de diversos métodos de control para implementar en sistemas mecatrónicos académicos, de servicios o industriales, ha aumentado considerablemente. En la Universidad de Nacional de Trujillo, en la Escuela de Ingeniería Mecatrónica se ha constituido el equipo de robótica Robotronics que diseñan robots para participar en competencias de robótica a nivel regional y nacional, pero no ha construido un robot humanoide dotado de visión artificial.

El 2010, el Instituto de Ingeniería Eléctrica & Electrónica (IEEE) introduce una categoría en la Competición Latino Americano de Robótica (LARC) denominada RoboCup Humanoid, el cual consta de robots autónomos que juegan al fútbol. En dicho evento se pudo observar las considerables mejoras tecnológicas asociadas a la robótica debido a que ahora existen procesadores de menor tamaño pero más eficientes, sensores con mayor exactitud y precisión, dispositivos modernos a precios más accesibles. El 2012, la IEEE da a conocer la apertura de una nueva categoría en el LARC, llamada: Humanoid Robot Racing (HRR) o robots humanoides de carrera, donde un robot de tamaño menor a 1.20 m y mayor a 0.60 m, recorre una línea negra recta de 2cm de ancho y 400 cm de largo dentro de una superficie plana y blanca, el cual según las bases, el robot que llegue a la meta en el menor tiempo posible es el ganador (IEEE-RAS, 2013).

Para lograr que el robot siga la línea negra necesita de un sistema de visión artificial bidimensional integrada en el robot humanoide, el cual a través de los datos obtenidos por procesamiento de imagen, el robot humanoide sigue el trayecto de la línea negra con una locomoción bípeda en el menor tiempo posible (Cazorla, 2010; Fernández, 2009; Sobrado, 2003; Torres, 2009).

Brecht y Wyns (2010) emplean la lógica difusa a la visión artificial estereoscópica para evitar obstáculos cuando el brazo robótico industrial, FANUC Arc Mate, se mueva desde su punto efector hasta su punto de destino. En el estudio los autores dan importancia a la seguridad industrial y tratan de solucionar los peligros que existen cuando los brazos robot dejan caer partes pesadas o ejecutan un movimiento que pueden herir al humano en caso este se sitúe en su trayectoria. Para el diseño y construcción de este sistema de control de seguridad activo para un brazo robot FANUC utilizaron la técnica de visión artificial estereoscópica el cual consiste en detectar obstáculos en el espacio, al posicionar dos cámaras en diferentes lugares y obtener dos imágenes del mismo entorno en distintos ángulos.

Casanova (2015) propone el desarrollo de un controlador difuso para estabilizar el funcionamiento de un quadrotor en el seguimiento de trayectorias tanto en simulación como en la plataforma real hallando que el sistema difuso es más estable y se acerca más a la trayectoria deseada.

En ese sentido, el objetivo de la investigación fue desarrollar un controlador difuso en la visión artificial bidimensional del robot humanoide de carrera para que mejore su locomoción bípeda en el seguimiento de trayectorias con respecto a un controlador clásico.

Material y métodos

En la investigación se diseñó, programó y creó el robot humanoide de carrera Mecatrón (figura 1). Mecatrón presenta ocho servomotores, seis controlados por el programa Arduino Nano y dos por el programa Matlab. Mecatrón recorrió dos trayectorias: una recta (pista 1) y una curva (pista 2). Las pistas 1 (figura 2) y 2 fueron de 3m de longitud y 2 cm de ancho. Se realizaron tres desplazamientos en cada pista.

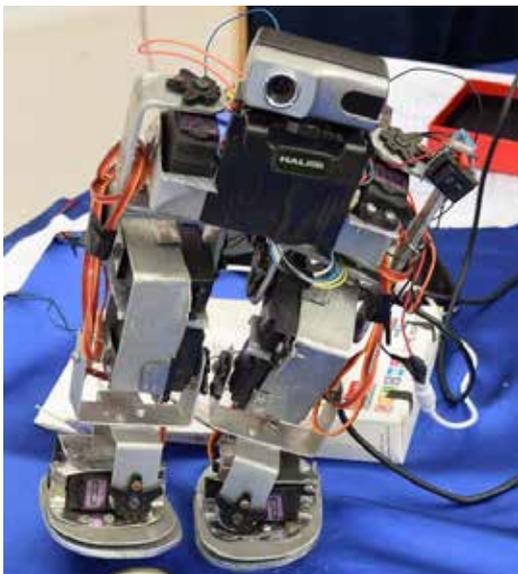


Figura 1. Vista frontal del robot Mecatrón con la cámara incorporada.



Figura 2. Mecatrón en desplazamiento sobre la trayectoria recta (pista 1).

La pista 2 presentó una curvatura a mitad de camino para que el robot se desvíe y así, usando el controlador difuso, corrija su recorrido. Inicialmente se programó a Mecatrón con lógica clásica la locomoción bípeda y visión artificial (Cazorla, 2010; Sobrado, 2003), haciendo que recorra tres veces la pista 2. En cada recorrido se calculó la distancia máxima de desvío del robot respecto a la curva gracias a los datos que obtuvo de la visión artificial. El tiempo en que corrigió su recorrido y el tiempo total de partida y llegada se obtuvo debido a las grabaciones que se realizaron durante el desarrollo del robot en la trayectoria.

Luego en la misma pista, con las mismas condiciones externas e internas de luminosidad, rugosidad, peso y carga de la batería, a Mecatrón se le programó con lógica difusa a fin que la visión artificial corrija la dirección de la locomoción bípeda (Guerrero y col., 2011). Mecatrón recorrió también tres veces la trayectoria.

Resultados y discusión

El peso total de la estructura robótica incluida el Arduino Nano, la Cámara Web y la batería, es de 950 gramos y tiene una altura de 24 cm. Cada servomotor tiene un consumo de corriente promedio de 320 mili amperios (mA) pero se calculó que los motores del pie generan picos de corrientes de hasta 1100 mA cuando soporta el peso

total del robot en un pie debido a su caminata estática, por lo que se necesitó 1 banco de batería de 5 v y 4 Amperios*Hora.

La cámara web fue útil para obtener los datos de la trayectoria deseada. La cámara fue capaz de calibrar la iluminación de la pista y enfocarla automáticamente, por lo que al procesar la imagen en el script de MatLab se obtenían los datos precisos y exactos de la línea respecto al robot.

El Arduino Nano controló con éxito los seis servomotores permitiendo que la estructura robótica tenga una locomoción bípeda. Los seis servomotores se controlaron independientemente a través de una interfaz programada en MatLab (figura 3). Durante la caminata se midió la longitud de avance de cada paso que es de aproximadamente 7.48 cm entre pie y pie; el tiempo que emplea en cada zancada es de 1.37 segundos.

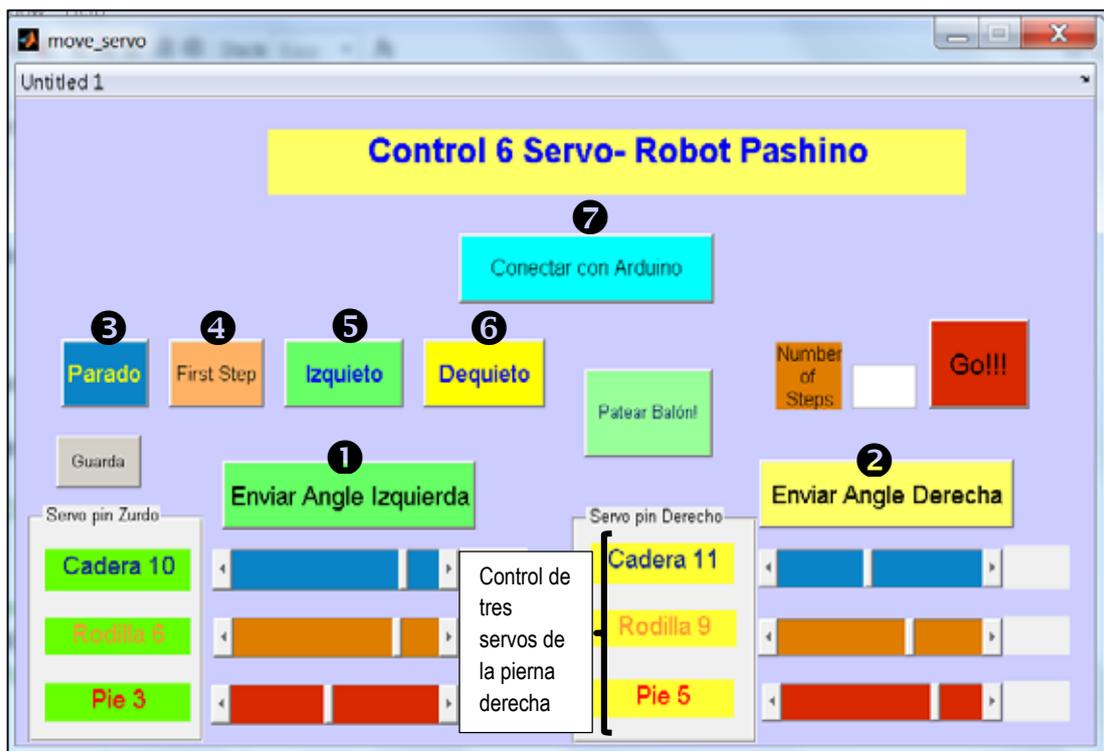


Figura 3. Interfaz programada en el Guide de MatLab por los autores

En el software de MatLab se programó un script de visión artificial bidimensional para obtener la distancia instantánea de la trayectoria respecto al robot bípedo y con ello se envíen al controlador difuso y al controlador clásico. Se implementó un controlador difuso en la visión artificial del robot bípedo para controlar los motores de la cadera. Tiene como datos de entrada la distancia del centro de la cámara respecto a la línea el cual consta de 5 conjuntos de membresía: muy izquierda, poco izquierda, centro, poco derecha y muy derecha. Como datos de salida los ángulos de los dos servomotores de la cadera que van desde 10 grados a -10 grados.

A partir de los datos previos obtenidos de la visión artificial bidimensional, se implementará un controlador difuso de tipo Mamdani en el software MatLab mediante

el comando *fuzzy* el cual contiene una entrada (error de posición) y una salida (Ángulo de los motores de la cadera). Luego de asignar los valores de los rangos de las entradas y salidas se procede a definir las reglas del controlador para crear la base de conocimiento. Para este sistema se propusieron 7 reglas del tipo IF, THEN. Estas reglas se diseñaron de acuerdo a las observaciones del autor. El gráfico del controlador difuso se presenta en la figura 4.

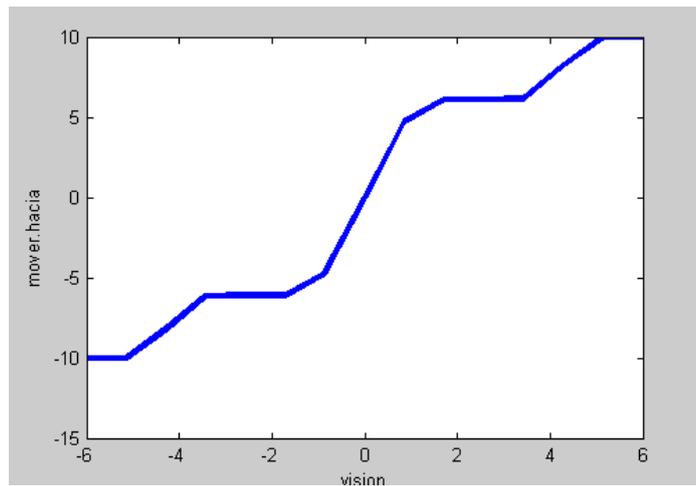


Figura 4. Trayectoria lineal obtenido del controlador difuso: error vs ángulo

Así mismo, se implementó un controlador clásico en la Visión Artificial bidimensional del robot bípedo para controlar los motores de la cadera. Tiene como datos de entrada la posición del robot respecto a la línea de la trayectoria: izquierda, derecha o adelante; y como datos de salida los ángulos de giro de 10 grados y -10 grados, de los servomotores de la cadera.

Luego, al robot Mecatrón se le implementó un controlador difuso en su visión artificial, realizándose las pruebas en dos tipos de trayectorias: pista 1 y pista 2. Debido al buen diseño y construcción del robot, y a la programación correcta del Arduino Nano para mover los servomotores, Mecatrón tuvo una desviación máxima de 1.2 cm respecto de su recorrido. Para comparar este resultado se tomó como referencia el video que muestra el uso del controlador clásico en un robot bípedo, el robot PUCP, el cual recorrió la línea mencionada en 91 segundos aproximadamente.

Según Casanova (2015) y Parra (2008) al comparar el desempeño del ArDrone 2.0 en el seguimiento de trayectorias utilizando el controlador difuso y un controlador proporcional, halló que el sistema difuso es más estable y se acerca más a la trayectoria deseada (figura 5). Este dato le proporciona al presente trabajo el soporte lógico adecuado.

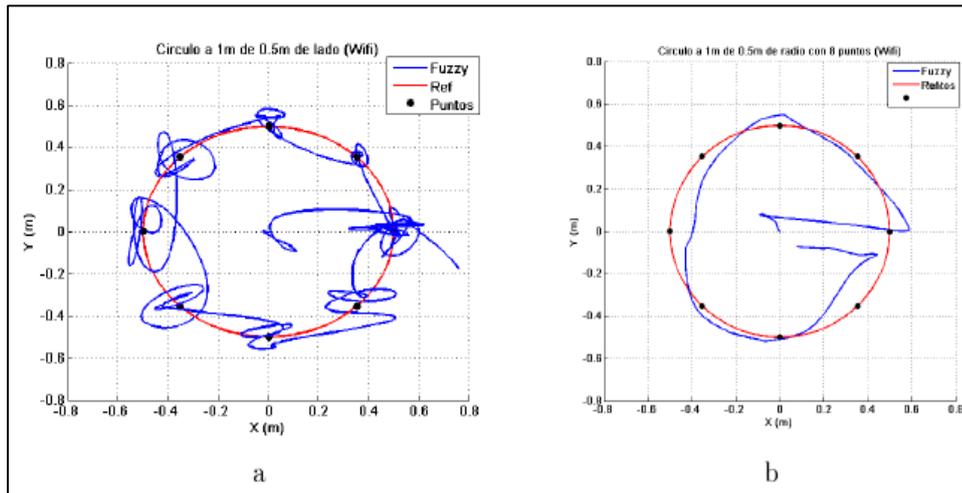


Figura 5. Comparación del seguimiento de un Círculo de 8pts:a)Control proporcional; b) Control difuso

En la tabla 1 se observa que al aplicar un controlador difuso en el robot Mecatrón, el tiempo record de partida y llegada es de 35 segundos siendo menor que el tiempo empleado por el otro robot implementado con el control clásico.

Tabla 1. Comparativo de los tiempos totales en recorrer la Pista 1

	Mecatrón	Pucp
Intento	Tiempo total de partida y llegada de la competencia (segundos)	
1	72	91
2	75	91
3	76	91

Fuente: Elaboración propia

La segunda prueba se realizó en la pista 2 el cual contiene una curva en la parte central de su recorrido, pero con la misma longitud que la anterior. En primer lugar se le programó con lógica clásica la visión artificial del robot Mecatrón, haciendo que recorra tres veces la pista. En cada recorrido se calculó la distancia máxima de desvío del robot respecto a la curva, el tiempo en que corrige su recorrido y el tiempo total de partida y llegada.

En la tabla 2 se observan el desarrollo de la locomoción bípeda usando un controlador clásico para el seguimiento de trayectorias. El primer dato es la distancia de desvío máxima en cada recorrido hasta que llegue a la meta. Luego el tiempo de corrección significa cuanto demora el robot en volver a establecerse en la línea negra del recorrido en la parte central de la cámara web, es decir alinear el robot a la trayectoria. Finalmente el tiempo total de partida y llegada que demora el robot Mecatrón para recorrer toda la trayectoria.

Tabla 2. Valores calculados utilizando controlador clásico en la Pista 2

Control Clásico			
Número de intentos	Distancia de desvío (cm)	Tiempo de corrección (s)	Tiempo total partida-llegada (s)
1	9	11	83
2	10	12	84
3	12	15	87

Fuente: Elaboración propia

Finalmente en la misma pista, con las mismas condiciones, se le programó con lógica difusa a la visión artificial para que éste corrija la dirección de la locomoción bípeda (Del Brio y Sanz, 2002; Tibaduiza y col., 2011). El robot Mecatrón también recorrió tres veces la línea negra obteniendo en cada intento los valores de la distancia máxima de desvío del robot respecto a la curva, el tiempo en que corrige su recorrido y el tiempo total de partida y llegada.

En la tabla 4 se observa nuevamente tres valores los cuales determinaron su mejora en la locomoción bípeda utilizando un controlador difuso. El primer dato es la distancia de desvío máxima en cada recorrido hasta que llegue a la meta. El segundo dato es el tiempo de corrección que significa cuanto demora el robot en volver a establecer la línea negra del recorrido en la parte central de la cámara web, es decir alinear el robot a la trayectoria. Finalmente el tiempo total de partida y llegada que emplea el robot Mecatrón para recorrer toda la trayectoria.

Tabla 3. Valores calculados utilizando controlador difuso en la Pista 2.

Control Difuso			
Número de intentos	Distancia de desvío (cm)	Tiempo de corrección (s)	Tiempo total partida-llegada (s)
1	3	4.5	75
2	4	6.7	79
3	4.5	6.3	81

Fuente: Elaboración propia

Conclusiones

Se aplicó la lógica difusa en la visión artificial bidimensional del robot bípedo Mecatrón mejorando su locomoción en el seguimiento de trayectorias con respecto a un robot bípedo controlado con lógica clásica.

Se evidencia la mejora de la locomoción bípeda en el seguimiento de trayectorias en el robot con el sistema de control difuso a diferencia del robot con el sistema de control clásico.

El modelo propuesto de un robot bípedo de carrera con ocho grados de libertad (GDL) y simulado en el software SolidWorks, presenta una mayor eficiencia que con los diseños comunes.

La cámara web incorporada en el robot le permitió una adecuada formación de imágenes bidimensionales mediante el programa de visión artificial elaborado en MatLab.

El Arduino Nano seleccionado controló de manera efectiva los servomotores del robot bípedo.

Referencias bibliográficas

- Aybar, J., Tasayco, P. y Zevallos, R. (2013). *Control de Robot por Visión*. Universidad Peruana de Ciencias Aplicadas.
- Brecht, F. y Wyns, B. (2010). *Industrial robot manipulator guarding using artificial vision..* Alex Ude Edit.
- Casanova S., A. (2015). *Control difuso del quadrotor ArDrone 2.0 para el seguimiento autónomo de trayectorias*. Oaxaca- México. Universidad Tecnológica de la Mixteca.
- Cazorla, M. (2010). *Robótica y Visión Artificial*. Universidad de Alicante.
- Del Brio, M. y Sanz, A. (2002). *Redes neuronales y sistemas difusos* (2º ed.) RA-MA Editorial.
- Fernández S. (2009). *Locomoción bípeda del robot humanoide Nao*. Universidad Politécnica de Catalunya.
- Guerrero, J., Pajarez, G. y Guijarro, M. (2011). *Técnicas de procesamiento de imágenes estereoscópicas*. Universidad Complutense de Madrid.
- IEEE- RAS. (2013). *Rules of Humanoid Robot Racing.12nd IEEE Competição Brasileira de Robotica*.
- Parra, S. (2008). *Diseño e implementación de sistemas de control para robots bípedos*.
- Sobrado, E. (2003). *Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot*. Lima- Perú. PUCP.
- Tibaduiza, D., Amaya, I., Rodriguez, S., Mejía, N. (2011). Implementación de un control fuzzy para el control cinemático directo en un robot manipulador. *Revista chilena de ingeniería*, 19: 312- 322.
- Torres, J. (2009). *Diseño y construcción de un robot bípedo caminante*. Universidad Peruana de Ciencias Aplicadas.