
LOCALIZACION Y POSICIONAMIENTO DE UN ROBOT CARTESIANO EN EL CENTROIDE DE UN OBJETO

Alumno: CHAPARRO REYES LUIS EDGAR, UPIITA-IPN, chaparroreyes@gmail.com

Abstract

En el presente trabajo, muestro cómo se puede localizar el centroide de un objeto mediante visión artificial utilizando la cámara de un celular, conectando la cámara a MATLAB por conexión wifi y colocándola en un robot cartesiano. El centroide del objeto se localiza con respecto al centro de un snapshot en x-píxeles e y-píxeles y haciendo una conversión de píxeles a milímetros se crean las entradas de un controlador Fuzzy Logic Sugeno para el control del robot cartesiano. El uso de este algoritmo lógico permite el control de 2 motores a pasos, que mueven al robot cartesiano tanto en su eje "x" como en su eje "y" para localizarse en el centroide del objeto. Una vez terminado el control lógico para rectificar que el robot cartesiano se encuentre en el centroide del objeto se toma una última imagen donde el objeto se debe encontrar justo en el centro del snapshot.

I. Introducción

Los robots cartesianos, abordan un gran abanico de tareas de automatización tales como la clasificación, la manipulación, el embalaje, el estampado, el montaje de piezas, entre muchas otras tareas (fig. 1 y fig. 2). La importancia de conocer el centroide de un objeto y posicionar el robot cartesiano en esta posición es, la certeza con la que se llevara a cabo la tarea a realizar con el objeto. Todas estas tareas son realizadas por tiempos, por accionamientos de sensores, por secuencias y otras por visión artificial siendo esta ultima la más atractiva ya que aparte de obtener las localizaciones de los objetos con lo que se trabajan, se pueden hacer procesos más complejos simultáneamente con un control de calidad reduciendo procesos y aumentando la velocidad de realizaciones de tareas.

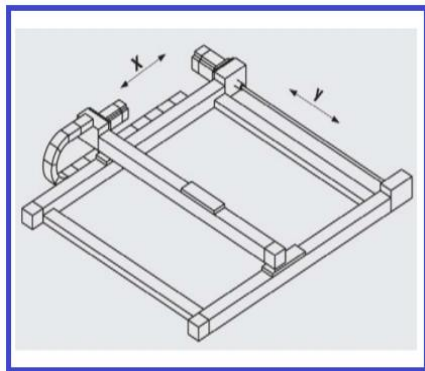


Figura 1. Ejemplo de un Robot Cartesiano



Figura 2. Ejemplos de Tareas de un Robot Cartesiano

Existen muchas formas de hacer un robot cartesiano ya que su principal característica es contar con estructuras articuladas que cuentan con una serie de elementos o eslabones que facilitan sus movimientos, y usando bandas dentadas o tornillos sin fin se logran movimientos lineales al girar los ejes sobre los que están montados los eslabones, con estas estructuras se logra translaciones verticales, horizontales y transversales.

En el estudio de robots cartesianos encontramos que la mayoría de estos robots son controlados por motores a pasos ya que estos ofrecen una alta precisión para estos tipos de mecanismos, su principal cualidad es poderlos mover desde un paso hasta una secuencia interminable de pasos dependiendo de la cantidad de pulsos que se les aplique a través de un microcontrolador, dándonos como resultados movimientos angulares que van desde 1.8° hasta los grados que uno desee mover, ejemplo de un motor a pasos vea figura 3.

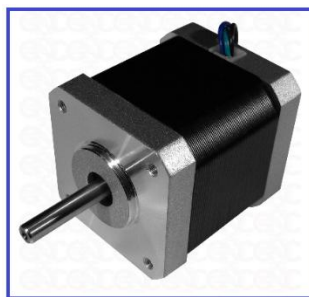


Figura 3. Ejemplo de un Motor a Pasos

II. Desarrollo

MECANISMO

Se construye el robot cartesiano sobre una estructura de metal para la base y rieles de acero de 5/16 de pulgada para los carros transversal y longitudinal. Al ser más firme la base reduce considerablemente las vibraciones mecánicas debido a los

movimientos de los ejes. Para el avance se utilizan tornillos sin fin de cuerda ACME de 5/16" de 4 hilos con un paso de 8mm por vuelta soportados por chumaceras de 5/16". En el tornillo sin fin se colocó un copleé 5-8 para tener una transmisión de movimiento del motor al tornillo. El motor utilizado fue un Motor a Pasos Nema 17 de 200 pasos a 2.4 A bipolar. Con estas conexiones en el robot cartesiano se obtuvo una alta precisión milimétrica de avance ya que 200 pasos es una vuelta del Motor a Pasos y una vuelta en el tornillo sin fin equivale a trasladarse 8mm lineales sobre el eje, por lo que cada paso tiene un avance de **0.04mm**, siendo esta la precisión del robot cartesiano construido. La figura 4 muestra una vista isométrica del robot cartesiano diseñado en SolidWorks.



Figura 4. Diseño de Robot Cartesiano en SolidWorks

Sistema de Vision Artificial

Para implementar el sistema de visión artificial se usó una cámara web mediante el teléfono celular, utilizando una aplicación Android. La conexión es mediante Wifi y el sistema sirve para detectar distintos objetos o figuras geométricas y localizar el centroide de cada una de ellas, este centroide está dado en pixeles en "x" e "y" con respecto al centro de la imagen, al obtener las coordenadas se hace una conversión de pixeles a milímetros para la siguiente etapa. El sistema difuso tiene por objetivo convertir estas coordenadas milimétricas en pasos para lograr mover los motores, ya sea en x o en y.

El sistema de reconocimiento de figuras cuenta con un método de binarización de imágenes en tiempo real, que pintará de blanco las figuras detectadas y el fondo estará en color negro, de las distintas figuras pintadas en blanco (binarias) se obtendrá su borde para hacer más fácil su reconocimiento y por ultimo con una función en Matlab llamada *regionprops* se obtiene el centroide de la figura, en la imagen 5 y 6 se muestra la imagen tomada por la cámara mediante Matlab y la imagen procesada para encontrar el centroide del objeto y sus coordenadas respectivamente, para la conversión de pixeles a milímetros se acotaron diversas variable, es decir que la toma de las imágenes siempre fueron a 65 cm del piso además que la resolución de la cámara fue constante (720 x 720), por lo que al tomar una imagen con la cámara con

las condiciones mencionadas a una figura rectangular en una posición paralela a los ejes "x" "y" de la imagen, se midió cuanto tenían sus lados en milímetros y cuantos pixeles ocupaba en la imagen tomada, teniendo un conversión de 1mm=1.192 pixeles.

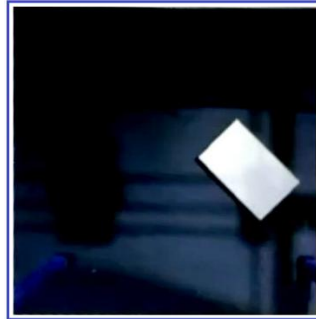


Figura 5. Imagen tomada desde la cámara montada en el Robot Cartesiano

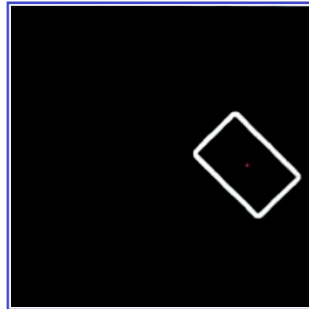


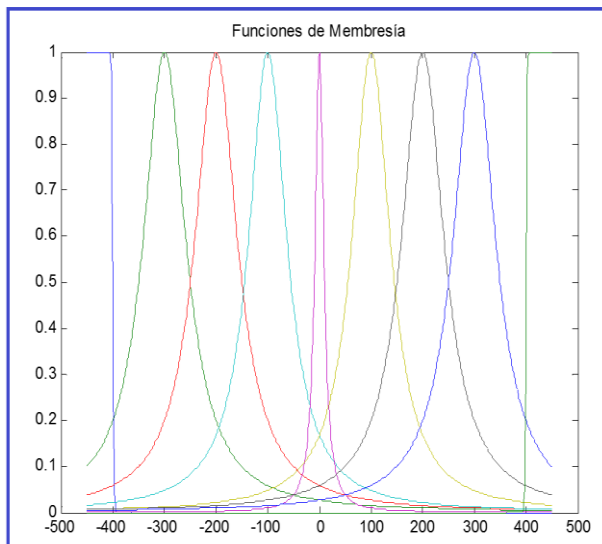
Figura 6. Imagen procesada encontrando el centroide del objeto

Sistema Neurodifuso

La defusificación es un proceso matemático usado para convertir un conjunto difuso en un número real. El sistema de inferencia difusa obtiene una conclusión a partir de la información de la entrada, pero es en términos difusos. Para implementar el Sistema Difuso es necesario saber las entradas del sistema y las salidas a las que queremos llegar, para seleccionar que sistema fusificador se utilizara y para nuestro control de motores a pasos necesitamos salidas certeras por lo que el mejor Sistema Fuzzy Logic para nuestro control es un Takagi-Sugeno (SISO).

Las entradas del control son las distancias en "x" e "y" en milímetros de nuestro sistema anterior y como el área de trabajo del robot cartesiano es de 400mm por 400 mm este será el universo de nuestras entradas y para describir las posibles entradas difusas se proponen funciones de membresía de campana, vea figura 7. Para llevar a cabo las reglas del control se deben formar funciones que se aproximen a las salidas, estas están descritas por los intervalos de inicio y fin de las funciones de membresía de las entradas, estas reglas del control se muestran en la figura 8, una vez teniendo nuestras entradas y las reglas de fusificación el programa nos arroja la curva o función de control para que el sistema sepa cómo reaccionar (dar una salida) ante una entrada,

las salidas del control son en número de pasos que tendrá que mandar el microcontrolador a los motores a pasos para que el robot cartesiano avance hasta el centroide del objeto, la curva de control obtenida se muestra en la fig. 9.



Reglas Difusas

Si ENMG, entonces $P = f_1(e) = 25e$.

Si ENG, entonces $P = f_2(e) = 25e$.

Si ENM, entonces $P = f_3(e) = 25e$.

Si ENP, entonces $P = f_4(e) = 25e$.

Si EC, entonces $P = f_5(e) = 25e$.

Si EPP, entonces $P = f_6(e) = 25e$.

Si EPM, entonces $P = f_7(e) = 25e$.

Si EPG, entonces $P = f_8(e) = 25e$.

Si EPMG, entonces $P = f_9(e) = 25e$.

Figura 7. Entradas difusas en milímetros del centroide del objeto difusas para el control de robot cartesiano

Figura 8. Reglas

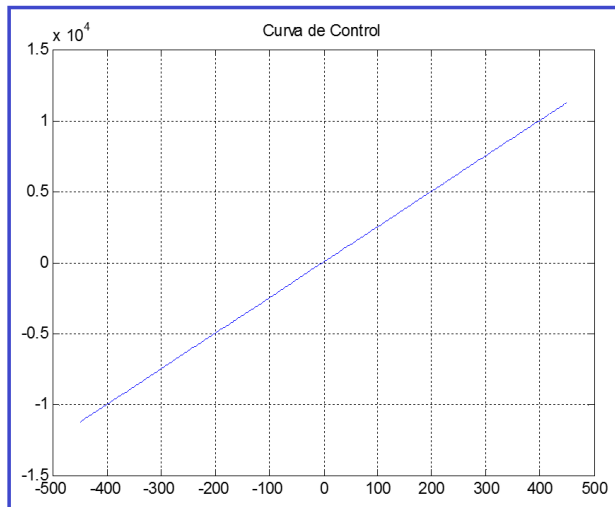


Figura 9. Curva de control para los Motores a Pasos

Conexión Serial de Matlab a Arduino

Es necesario un microcontrolador que dé señales PWM a los motores a pasos, para este proyecto se utilizó un Arduino Uno que es compatible con Matlab ya que existen librerías para su conexión serial, facilitando nuestro diseño de programa ya que todo el algoritmo se hace en Matlab y solo se envían las señales PWM, Enable y de Dirección al driver A4988 que se encarga de darle la potencia a los motores a pasos. Estos drivers son los Pololu A4988 que son controladores específicos para los motores a pasos bipolares sus principales características son limitar la corriente que consume el motor, protege al microcontrolador aislándolo de la parte de potencia que necesitan los motores para funcionar y logra controlar al motor hasta en 1/16 de paso, en la figura 10 se muestra un Pololu A4988 y su conexión básica para el control de motores a pasos.

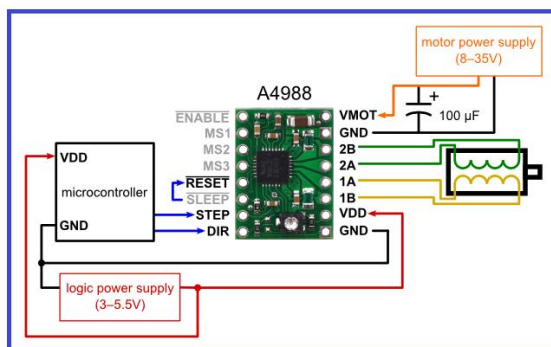


Figura 10. Driver Pololu A4988 para control de Motor a pasos y su conexión básica

El programa en Matlab para el control de los motores cuenta con la conexión al Arduino Uno y dependiendo de las respuestas del control difuso en pasos tenemos 4 posibilidades de movimiento según las direcciones de estos pasos en base al signo que contengan, si es positivo es un movimiento a la derecha y si es negativo es un movimiento a la izquierda. Los posibles movimientos son los dos a la derecha, los dos a la izquierda, uno a la derecha y el otro a la izquierda y al contrario de este último logrando movernos por toda la superficie de trabajo del robot cartesiano.

Para todos los casos mencionados siempre se tiene que habilitar el pin de Enable del driver con un 0 lógico cuando ya se vayan a mover los motores además de mandar PWM al pin de STEP del driver dependiendo de la dirección que este necesite si es a la derecha se necesita enviar un 1 lógico en DIR y un PWM del 90% en alto al pin STEP del Pololu y si es a la izquierda se necesita enviar un 0 lógico a DIR y un PWM del 10% en alto al pin STEP del Pololu. Por último para detener los motores es necesario invertir las direcciones, es decir cambiar de 0 a 1 lógico en DIR o al contrario si es el otro caso, en seguida se deshabilita el Enable del driver con un 1 lógico para dejar clavados los Motores a Pasos y no se puedan mover.

III. RESULTADOS

Una vez terminadas las etapas del proyecto se procede a ver el funcionamiento del robot cartesiano logrando su objetivo de localizar al objeto encontrando su centroide utilizando sus contornos con ayuda de visión artificial y posicionándolo mediante el control difuso, en el siguiente link muestro todo el proyecto funcionando.

<https://www.youtube.com/watch?v=nkA88eU8jf8>

Este sistema al tener dos entradas y dos salidas debería de implementarse un control difuso MIMO, pero al hacer la consideración de que la respuesta del motor del eje X depende única y exclusivamente de la entrada "X" (distancia en el eje X en milímetros), así como el análogo para el Motor del eje "Y", se hace el control difuso con dos sistemas SISO logrando controlar los dos motores como si fuese un sistema MIMO.

Al analizar la curva de control se ve una recta con la misma pendiente durante toda la gráfica. Se decidió implementarlo de esta manera porque la respuesta del sistema, es decir el número de pasos que deben de avanzar los motores para llegar a su objetivo

con respecto a la entrada es directamente proporcional dándonos valores certeros para el movimiento de los motores a pasos. Es importante mencionar que no importa si la distancia al objetivo es muy grande o muy pequeña, los motores siempre funcionan a la misma velocidad, el parámetro que se modifica es la duración del PWM para activarlos, que es el resultado de la defusificación del controlador.

IV. Conclusiones

El Robot Cartesiano responde de una forma muy precisa gracias al diseño del sistema, la estructura y los actuadores utilizados. Los motores a pasos se volvieron una parte fundamental para el sistema debido a la resolución de cada uno de sus pasos (0.04mm). Una de las características del sistema es la no variación de la velocidad de operación de los motores, como se ha explicado anteriormente el único parámetro que se modifica para activar los motores a pasos es la duración de los PWM, es decir los pasos multiplicados por los ciclos del PWM para realizar la activación de los motores a pasos para que se llegue a la distancia requerida.

La forma de construcción del controlador Takagi-Sugeno permitió que la respuesta fuera lineal, dependiendo de la entrada (como se pudo apreciar en la superficie de control). La respuesta es directamente proporcional a la entrada. El tiempo de respuesta es de 1.02 milisegundos por lo que el robot es bastante rápido, además basta con activar el sistema una sola vez para que se centre la cámara con el objetivo. Sin necesidad de realizar diferentes análisis para lograr esta acción.

V. REFERENCIAS

- [1] Acarnley, P. (2007). Stepping Motors. A guide to theory and practice. London, United Kingdom: The Instituto of Engineering and Technology.
- [2] Belmar, G. G. (Septiembre de 2012). Diseño de construcción de un móvil controlado con técnicas de lógica difusa . Obtenido de <http://www.sepi.esimez.ipn.mx/electronica/archivos/992.pdf>
- [3] Hughes, A. (2006). Electric Motors and Drives Fundamentals, Types and Applications. Great Britain: Elsevier Ltd. .
- [4] Jorge Rojas V., I. M. (02 de Diciembre de 2003). Diseño de un Sistema Robotico Cartesiano para Aplicaciones Industriales. Obtenido de <http://www.scielo.cl/pdf/rfacing/v11n2/ART03.pdf>
- [5] Parker, J. R. (2011). Algorithms for Image Procesing and Computer Vision. Wiley Publishing, Inc.
- [6] Pololu. (s.f.). Stepper Motor Drivers A4988. Obtenido de <https://www.pololu.com/product/1182>

- [7] Willy, G. H. (23 de Octubre de 2007). Diseño y Modelamiento de un Robot Cartesiano. Obtenido de Facultad de Ingeniería de Diseño y Automatización:
<http://repository.lasalle.edu.co/bitstream/handle/10185/16112/44022025.pdf;jsessionid=26D71DFA54D72314BDDE7815ED7440E7?sequence=2>
- [8] Ying, H. (2000). Fuzzy Control and Modeling, Analytical Foundations and Applications. New York: IEEE Press Series on Biological Engineering.