

DESARROLLO DE UN ROBOT MÓVIL E IMPLEMENTACIÓN DE SISTEMA DE VISIÓN ARTIFICIAL CON LIBRERÍAS ARUCO PARA NAVEGACIÓN AUTÓNOMA

*DEVELOPMENT OF A MOBILE ROBOT AND IMPLEMENTATION OF
AN ARTIFICIAL VISION SYSTEM USING ARUCO LIBRARIES FOR
AUTONOMOUS NAVIGATION*

Raúl Aramis Villagómez Ortiz

Tecnológico Nacional de México en Celaya
a22_ki4@hotmail.com

Jeovany Velasco Avella

Tecnológico Nacional de México en Celaya
jeovanyelite@gmail.com

Arturo López Gallardo

Tecnológico Nacional de México en Celaya
arturolg94@hotmail.com

José Guadalupe Zavala Villalpando

Tecnológico Nacional de México en Celaya
jg.zavala@itcelaya.edu.mx

Karla Anhel Camarillo Gómez

Tecnológico Nacional de México en Celaya
karla.camarillo@itcelaya.edu.mx

Resumen

Este artículo presenta la integración de un sistema de visión, comunicación y navegación enfocado a un robot móvil, donde el objetivo final del sistema de visión es proporcionar datos de posición y orientación respecto de un marcador ArUco. Luego, a través de su sistema de navegación, el robot generará una ruta óptima de navegación mandando órdenes a un protocolo de comunicación serial que las ejecuta por medio de un control de posición.

Palabra(s) Clave: Comunicación, Robótica Móvil, Navegación, Visión.

Abstract

This paper presents the proposal of a vision, communication and navigation system focused in a mobile robot, where the final purpose of the vision system is to provide position and orientation data respect an ArUco marker. Then through its system navigation the robot will generate an optimal way to navigate by generating orders to a serial communication protocol that executes the orders with a position control.

Keywords: *Communication, Mobile robot, Navigation, Robot, Vision.*

1. Introducción

Un robot móvil es una máquina automática capaz de moverse en un entorno. Son un punto importante de la investigación moderna debido a la planificación de la trayectoria y el control de las velocidades y posiciones del robot (Baltes, 2009). En los últimos años el desarrollo de sistemas inteligentes ha requerido la implementación de sistemas cuyas funciones puedan ser similares a los sentidos humanos, cómo la visión artificial o algoritmos cercanos al pensamiento humano o instinto animal.

El propósito principal de este robot es la integración de un sistema de visión, navegación, comunicación y de control para realizar una navegación autónoma en un entorno controlado:

- Para la implementación del sistema de visión se requieren las librerías ArUco que ayudan a estimar la posición del robot en un mapa mediante la detección de diferentes marcadores posicionados estratégicamente en un entorno.
- El sistema de navegación integra un algoritmo A* que ayuda al robot con las decisiones para tomar una trayectoria optima entre múltiples posibilidades basándose en variables como la distancia y dificultad del camino.
- Este algoritmo manda datos mediante un protocolo de comunicación serial al control de posición, con el cual se ejecutarán las ordenes de movimientos.

2. Métodos

Descripción del entorno

El sistema necesita un entorno controlado para una navegación autónoma efectiva en un mapa previamente programado. Para esto fue necesario definir el área donde se limitó la navegación del robot. Se seleccionó el laboratorio de mecatrónica del Tecnológico Nacional de México en Celaya limitando el mapa a la ubicación del área de robótica como se percibe en la figura 1.



Figura 1 Representación visual del entorno.

El laboratorio tiene las medidas de 21.2 x 5.8 m, de esta forma se pudo representar en una matriz de 51 x 28 nodos donde cada representa un espacio de 20 x 20 cm. Para discretizar este mapa se consideró que cada nodo de la matriz tenía que representar un costo valorado según la dificultad del camino. La dificultad del camino variaba según la cercanía a los objetos u obstáculos que estuvieran en el mapa. Se estableció la posibilidad de poder modificar la matriz para expandir el tamaño total del entorno o para introducir un nuevo objeto.

El valor de los nodos va de 0 a 9 donde; 0 es el valor más barato y 9 el más caro. Para los lugares inaccesibles y obstáculos se consideró el valor 9 mientras que el 0 representaba un camino óptimo para navegación libre. Los números del 1 al 8 variaban dependiendo de las distancias del camino libre a un obstáculo de tal forma que pudiera considerar el tratar de no navegar cerca de estaciones de trabajo por seguridad (figura 2).

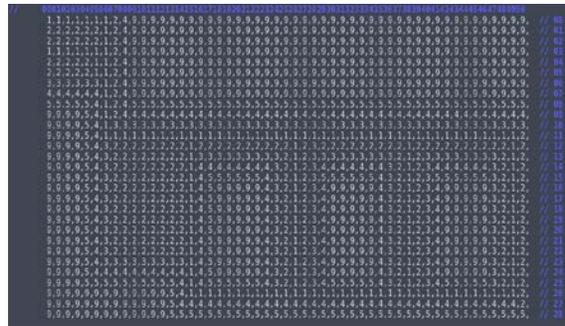


Figura 2 Discretización del entorno.

Para poder mejorar la experiencia al usar el programa se creó una interfaz gráfica que representará los datos de la matriz (figura 3).

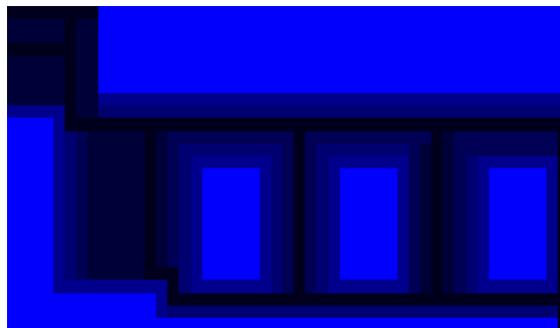


Figura 3 Interfaz gráfica del laboratorio.

Diseño de la estructura

El diseño de la estructura se realizó usando el software Solid Works 2016 donde el prototipo fue considerado en madera.

Para definir las medidas del robot fue necesario considerar los espacios más estrechos por los que podía navegar el robot. Los espacios más estrechos tenían una longitud de 80 cm tomando esto como una variable para considerar las medidas se decidió hacer una estructura de 40 cm de ancho, 50 cm largo y 55 cm de alto. La distancia del suelo a la cámara tenía que ser considerada en el programa de visión artificial por lo cual se decidió que la altura debía ser de 55 cm para que la cámara detectara el centro de los marcadores. También se consideró el diseño de un compartimiento de Laptop separado de los componentes electrónicos para protegerlos del sobrecalentamiento (figura 4).



Figura 4 Estructura del robot.

Descripción de software

Toda la programación del sistema se basa en Fedora, que es un sistema operativo de distribución de Linux. El lenguaje principal es C++ y la herramienta de compilación para este proyecto es la Terminal principal de Fedora.

Para la programación del AVR (AtMega328P), el software AtMel Studio es necesario porque proporciona todas las bibliotecas de C++ y puede programar toda la familia de microcontroladores AtMega.

El sistema de visión artificial se basa en las bibliotecas de OpenCv donde ArUco se puede encontrar incluido en la versión 3.0.

Hardware

El diseño del robot se basa en los requerimientos de la aplicación, el cual tiene las siguientes características:

- Unidad de control: procesa las variables del sistema y las señales, así como el algoritmo de control, consiste en un microcontrolador ATmega328P de 8 bits, a 500 MHz (figura 4).

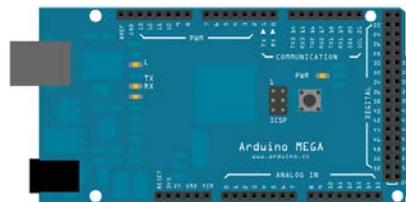


Figura 4 Microcontrolador Atmega328p en Arduino MEGA.

- Batería: la etapa de potencia que suministra energía al sistema de locomoción consiste en una batería Lipo de 7.4V con 3300 mAh para una autonomía estimada de 62 minutos (figura 5).



Figura 5 Batería lipo 7.4 V 3300 mAh.

- Elevador de voltaje: 4 módulos xl6009 para suministrar el voltaje controlado de operación, el cual puede suministrar un rango de voltaje de salida que varía de 2.7 V a 16 V (figura 6).



Figura 6. Modulo XL6009.

- Puente H: 4 módulos L298N, cuentan con protección incorporada contra voltaje inverso, bajo voltaje, sobre corriente y sobrecalentamiento, este controlador es una gran solución para alimentar a los pequeños, motores de baja tensión (figura 7).



Figura 7 Modulo puente H L298N.

- Sistema de visión: Logitech C921, esta cámara brinda una imagen de alta resolución de 1080p (figura 8).



Figura 8 Webcam 920c.

- Motor: 4 unidades de 107 RPM a 12 V y cada uno está equipado con un codificador de efecto Hall para la monitorización de posición (figura 9).



Figura 9 Motor DC con decodificador de efecto hall.

- Procesador: la CPU es una computadora portátil con un procesador de 4 núcleos con 2.4 GHz y 8 Gb de memoria RAM, seleccionado para el óptimo desempeño de los módulos del sistema (figura 10).



Figura 10 Cpu.

Sistema de visión

El sistema de visión hace uso de las librerías de open CV, los cuales proveen un diccionario completo de marcadores desarrollados por ArUco. El diccionario que se usa en este sistema proporciona 50 marcadores con diferentes códigos de identificación en cada uno (figura 11).

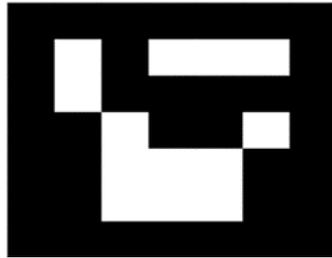


Figura 11 ArUco Marker.

Es necesario calibrar la cámara antes de la implementación de un sistema artificial para eliminar los factores que pueden distorsionar la percepción de la imagen. Para este propósito, ArUco tiene una herramienta llamada ChArUco que contiene las propiedades de un tablero de ajedrez tradicional y marcadores individuales, esta combinación es útil para la corrección de los parámetros intrínsecos de la cámara (figura 12).

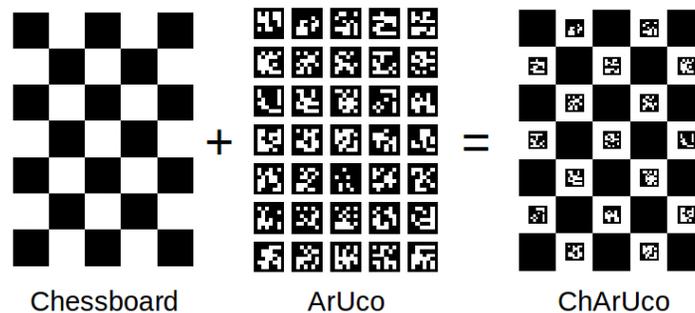


Figura 12 Herramienta de calibración ChArUco.

Para el uso de marcadores en la programación del sistema de visión artificial es necesario seleccionar el tamaño de cada marcador y la selección del ID en cada uno. Con estos datos, el marcador se coloca en un punto estratégico en el mapa para la estimación de la posición entre la cámara y el marcador (figura 13).

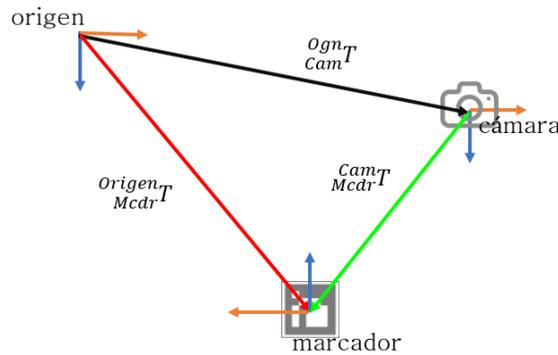


Figura 13 Estimación de pose.

El sistema de visión envía los datos necesarios de posición y orientación del robot al sistema de navegación.

Sistema de navegación

Para el sistema de navegación, se implementa un algoritmo de inteligencia artificial, para buscar y seleccionar trayectorias óptimas en el mapa. Para esta aplicación, un algoritmo A* funciona con un origen y un destino en el mapa.

El Algoritmo A* no calcula todas las combinaciones posibles para encontrar una solución al problema, porque usa una función heurística, que ayuda a encontrar una solución ahorrando tiempo de procesamiento.

Con la función A* (ecuación 1), el sistema elige una forma basado la solución en el costo de las posibles trayectorias, la función heurística que elimina los nodos innecesarios en el mapa para las principales comparaciones.

$$f(n) = g(n) + h(n) \quad (1)$$

La función heurística (ecuación 2) puede determinar el comportamiento del sistema, para este propósito la selección de la función heurística del taxi puede ser óptima, las trayectorias funcionan para movimientos verticales y horizontales (figura 14).

$$h(n) = \text{abs}(\text{distance}.x) + \text{abs}(\text{distance}.y) \quad (2)$$

El algoritmo A* nos arroja una serie de puntos con los cuales se deberán realizar los desplazamientos requeridos para llegar al punto destino, este tipo de

desplazamiento no son aptos para la realización del recorrido para el robot ya que generaría movimientos cortos en repetidas ocasiones (figura 15).

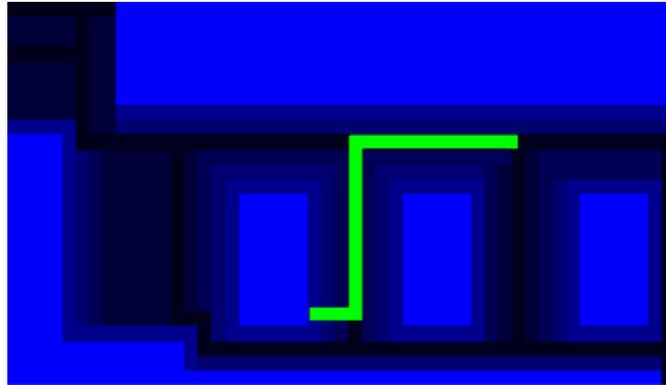


Figura 14 Trayectoria generada por el algoritmo.

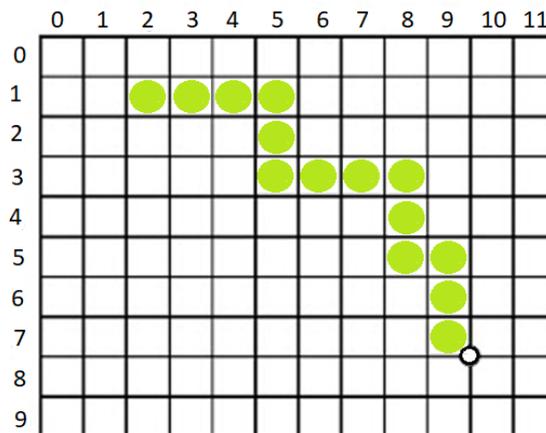


Figura 15 Salida de algoritmo A*.

Los datos de trayectoria son procesados por el algoritmo de reducción de nodos que toma el cambio en los puntos “x” y “y” para determinar la dirección y magnitud del desplazamiento para el próximo recorrido que realizara el robot. Este algoritmo se diseñó para disminuir la cantidad de movimientos comandados hacia el robot (figura 16).

Se implementó un algoritmo de definición de movimientos para que se pudieran enviar las instrucciones, la salida de este algoritmo es el tipo y magnitud del movimiento tomando en cuenta los giros necesarios para él robot (figura 17).

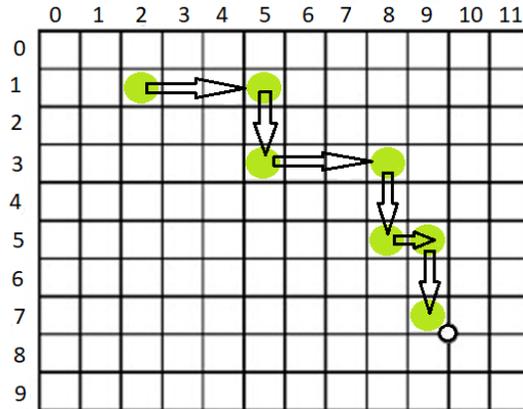


Figura 16 Algoritmo de reducción de nodos.

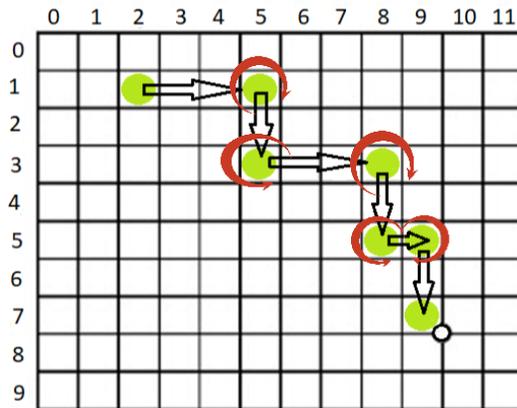


Figura 17 Algoritmo de definición de movimientos.

Con la información del algoritmo de definición de movimientos los datos pueden ser enviados como ordenes por medio del protocolo de comunicación Oriana. El cual se representa en las tablas 1 y 2.

Tabla 1 Protocolo de comunicación Oriana.

Encabezado		Instrucción	Parámetro			
'C'	'M'	Instrucción	Caracter 3	Caracter 2	Caracter 1	Caracter 0

Tabla 2 Instrucciones.

Instrucción	Descripción
'0'	Retroceso
'1'	Avance
'2'	Giro negativo
'3'	Giro positivo

En el apartado de parámetros se incluye la magnitud que la instrucción que se requiere ejecutar. El protocolo de comunicación Oriana se utilizó para generar una comunicación entre esclavo (Microprocesador) y maestro (CPU), este protocolo tiene la cualidad de eliminar interferencias entre la comunicación.

Se utilizó el microcontrolador AtMega 328p para controlar los movimientos a realizar del robot. Este microprocesador recibe el mensaje por medio del protocolo de comunicaciones Oriana vía Bluetooth.

Control

Se implementó control PI de posición en AtMega 328p mediante el monitoreo de pulsos generados por un encoder para cerrar el ciclo de control, la resolución se calculó mediante el número de pulsos en el encoder magnético (ecuación 3).

$$Resolution = 11 * 90 = 990RPM \quad (3)$$

Para el ajuste del control PI, se puede utilizar el segundo método de Ziegler y Nichols (figura 18) probando la señal para una oscilación sostenida. Eso resulta en $Kp = 24.75$ y $Ki = 1$.

Las ganancias finales del control PI son $Kp = 22$ y $Ki = 3$.

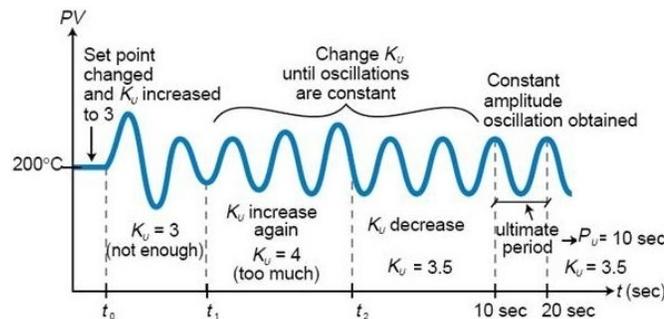


Figura 18 Segundo método de Ziegler y Nichols.

3. Resultados

El resultado de la unión de todos los módulos en un sistema le da al robot la capacidad de navegar por sí mismo, estimando su posición en el mapa y dándole la orden de ir a cualquier punto en el mapa por sí mismo (figura 19). Sin embargo,

el error en la posición final es mínimo y depende de factores externos como polvo o imperfecciones en el piso que pueden acumular errores en la navegación.



Figura 19 Prototipo final.

4. Discusión

La necesidad de un control de visión de ciclo cerrado podría eliminar el error de posición final en la navegación del robot al tomar una última estimación de la posición del robot en el mapa. El sistema funciona para un entorno controlado y el sistema es muy amigable con el usuario en el momento de las actualizaciones del sistema. Por lo tanto, este sistema puede implementarse mediante módulos en otros robots mediante el uso del módulo específico de programación.

5. Bibliografía y Referencias

- [1] Anderson, J., Baltes, J., & Tu, K. Y. (2009, March). Improving Robotics Competitions for Real-World Evaluation of AI. In AAAI Spring Symposium: Experimental Design for Real-World Systems (pp. 1-8).
- [2] Egly, U., Novak, G., & Weber, D. (2005). Decision making for MiroSOT soccer playing robots (pp. 69-72). na.
- [3] Atmel. ATmega328p (2016). Datasheet.
- [4] Santoyo-Mora, M., Cacique-Borrego, V. H., Camarillo-Gómez, K. A., & Ibarra-Zannatha, J. M. (2013, August). Programming the Goalkeeper of 20 DOF for FIRA Cup. In FIRA RoboWorld Congress (pp. 64-76). Springer, Berlin, Heidelberg.
- [5] Acerca de OpenCV, <http://opencv.org/about.html>.