

# Vehículo robot para gestión de almacenes

Paolo Bejarano y Álvaro Cárdenas  
{p.bejaranosamayani, al.cardenasq}@alum.up.edu.pe

Facultad de Ingeniería,  
Universidad del Pacífico, Lima 15072

---

## Resumen

En este paper presentamos un prototipo de vehículo robot capaz de llevar cajas a pedido del usuario y coordinar con otros robots para gestionar un almacén. El objetivo es maximizar el uso del espacio y reducir al mínimo la intervención del humano. El robot consiste en una chasis con 3 llantas, sensores ultrasonido y en la parte de arriba sostiene una caja donde pueden ponerse objetos. Puede conectarse por Bluetooth a una computadora y de esta manera recibir órdenes. La programación del robot es en Arduino y el programa de computadora fue hecho con Python 3.

Palabras clave: robots, almacenes, Arduino

## Abstract

In this paper we present a prototype of a robot vehicle capable of carrying boxes at the user's request and coordinating with other robots to manage a warehouse. The objective is to maximize the use of space and minimize human intervention. The robot consists of a chassis with 3 wheels, ultrasonic sensors and, in the upper part, it holds a box where objects can be placed. It is possible to connect via Bluetooth to a computer and thus the robot can receive orders. The programming of the robot is in Arduino and the computer program was made with Python 3.

Key words: robots, warehouse, Arduino

## 1. Introducción

Los robots pueden ser usados para automatizar muchos procesos. Uno de estos es la gestión de un almacén que forma parte de la logística de entrada y de salida de muchas empresas. La automatización permite ahorrar tiempo, ahorrar espacio y reducir la necesidad de trabajadores humanos.

El prototipo llamado BoxBot consiste en un robot-caja capaz de transportarse por sí solo a pedido de un humano y es capaz de comunicarse con otros robots para poder salir del almacén. La idea es poder ocupar todo el espacio disponible en un almacén con estos robots y estas cajas de forma que se maximiza el espacio.

Por ejemplo, se tiene un espacio de 3 metros por 3 metros y se quiere usar el mayor área posible. Una opción es poner todas las existencias en muebles y que estos estén juntos pero si se quiere sacar el mueble que se encuentra al fondo se deben sacar todos los muebles que obstruyen el paso. Lo que BoxBot propone es que todos los muebles sean robots y estén juntos, y cuando se requiera sacar algún mueble, estos robots pueden coordinar entre sí y evitar que el humano tenga que quitar los obstáculos.

## 2. Marco conceptual

### Cinemática para hacer avanzar y girar el robot

Los motores usados para el prototipo no puede ser programados tomando como input la distancia que quiere que se recorra o el ángulo en el que se desea girar, solo puede tomar como input el tiempo.

Entonces, si se quiere dar la instrucción de que avance hacia adelante una determinada distancia en metros es necesario expresar esta medida en unidades de tiempo. Para ello se puede calcular la velocidad del robot, usando la definición de velocidad  $v = \Delta x / \Delta t$  (Serway 2010) y tomando el tiempo con un cronómetro desde que el robot pasa por un punto  $x_1$  hasta un punto  $x_2$  (ver Figura 1). Para el prototipo, se usó una distancia de 1 metro y el tiempo fue de 2.5 s. Por lo tanto la velocidad es de 0.4 m/s y si se quiere introducir una distancia al robot en segundos, tendría que dividirse esa distancia entre 0.4 y poner esto en el código.

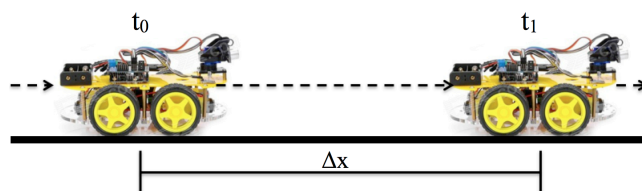


Figura 1

Para calcular el tiempo que debe introducirse al código del robot para que gire un ángulo determinado se procede de forma parecida. El objetivo ahora es calcular la velocidad angular asumiendo que es constante. Para ello se deja que el robot complete una vuelta entera ( $2\pi$  radianes) y se determina el tiempo que demora. Para este prototipo se determinó que demora 1.5 segundos en dar una vuelta. Por lo tanto la velocidad tangencial es de  $4\pi/3$  rad/s y si se quiere introducir un ángulo de giro al robot en segundos, tendría que dividirse este ángulo entre  $4\pi/3$  y poner esto en el código.

#### **Sonido y cinemática para el cálculo de distancia entre el robot y un obstáculo**

El sensor HC-SR04 es un sensor de distancia, este funciona mediante transductores de ultrasonido para detectar objetos. Su funcionamiento consiste en emitir un sonido ultrasónico por uno de sus transductores, luego esperar a que las ondas del sonido reboten con un objeto y el eco será captado por el segundo transductor. La distancia es proporcional al tiempo que demora en llegar el eco, es decir, mientras más demore en captar el sonido el segundo transductor, la distancia es mayor entre el sensor y el objeto.

La distancia se calcula a partir de la formula Distancia = Velocidad x Tiempo. Dentro del robot, este sensor cumplirá la función de identificar un espacio vacío para desplazarse a ese puesto. Se programaran a todos los robots para que a través de un algoritmo se escoja qué robot será el elegido para poder ocupar el espacio vacío y que de esta manera el que se quiera retirar acabe en el lugar deseado para retirar el contenido.

#### **Ondas**

La red del bluetooth transmite datos a través de ondas de radio de baja potencia. Opera en la frecuencia de radio de 2,4 a 2,48 GHz con amplio espectro y saltos de frecuencia. Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo. “Todas las ondas llevan energía” (Hewitt, 2010), en este caso, esta transmite información para crear una conexión y comunicación entre el robot y el dispositivo externo, sin la necesidad de la conexión física a través de cables.

### **3. Funciones del robot**

#### **3.1. Requisitos**

El principal requisito para poder usar el robot es una computadora con Python 3 instalado y conexión Bluetooth. Antes de usarlo, deben emparejarse ambos dispositivos. El robot se identifica con el nombre BoxBot. Adicionalmente, el almacén debe estar separado en 9 bloques cuadrados distribuidos en 3 filas y en 3 columnas, y se debe dejar al menos un bloque libre para que el robot o los robots puedan avanzar.

### 3.2. Modos de uso y funciones

#### Modo de uso de un solo robot en almacén

En caso solo se disponga de un robot en el almacén, el robot es capaz de analizar el entorno mediante los sensores ultrasónicos. De esta forma puede esquivar los obstáculos e informar la ubicación de los obstáculos al software para computadora. Cuando el usuario da la orden, el robot, según los obstáculos que encuentre, avanzará de bloque en bloque y usará la ruta más corta para salir del almacén.

#### Modo de uso con varios robots en almacén

Para que sea posible que el robot pueda salir del almacén debe coordinar con aquellos robots que obstaculicen el paso. Como se mencionó en los requisitos, se debe formar un almacén y distribuir las cajas como un puzzle deslizante. Debe haber un espacio libre para que los robots puedan avanzar y permitan salir al robot que ha pedido el usuario. Es posible que no todas las cajas sean robots, en este caso, el robot hace uso de los sensores ultrasonido para esquivar los obstáculos al igual que en el primer modo de uso.

Por ejemplo, se tienen 8 robots numerados del 1 al 8 y el usuario desea sacar el robot 1 pero todos los otros robots obstaculizan el paso. Entonces, los robots deben moverse siguiendo una serie de pasos que se describen en la Figura 1. El orden de ejecución en la figura es de izquierda a derecha y de arriba hacia abajo.

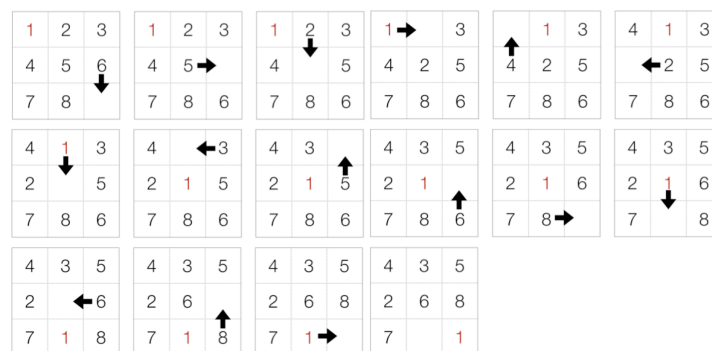


Figura 2

## 4. Anatomía del robot

### 4.1. Hardware

#### Tarjeta microcontroladora

Para este trabajo se ha escogido una tarjeta Uno de Arduino. El microcontrolador es la unidad central del robot. Todos los componentes están conectados a esta unidad.

#### Driver L298N

Este módulo permite controlar hasta dos motores de corriente continua. Es usado para girar el robot y avanzar hacia adelante.

#### Bluetooth HC-05

Este módulo permite controlar el robot por medio de una conexión inalámbrica a otro equipo con bluetooth.

#### Sensores ultrasonido HC-SR04

Este módulo envía un ultrasonido y lo recibe de vuelta. El prototipo tiene 3 sensores y de esta manera puede saber si hay algo adelante, a su izquierda o a su derecha.

## 4.2. Software

### Programación del robot

El robot está programado en C++ y está programado de tal forma que espera recibir órdenes de la computadora para ejecutar ciertos comandos. Así como recibe información de la computadora, también puede enviarla.

El código se encuentra en:

<https://github.com/BoxBot1/Prototipo/blob/master/BoxBot.ino>

### Aplicación para computadora

Está escrita en Python 3 y usa una interfaz gráfica. La aplicación de computadora es esencial para controlar el robot y consiste en una interfaz simple que permite pedir que el robot salga del almacén cuando el usuario lo requiera y también muestra que lugares del almacén están ocupados.

El código se encuentra en:

<https://github.com/BoxBot1/Prototipo/blob/master/BoxBot.py>

## 5. Conclusiones

En conclusión, la solución propuesta ayudaría a un manejo más eficiente de un almacén pues se maximiza el espacio disponible, se reduce al mínimo la intervención de un ser humano y permite ahorrar tiempo. Para programarlo fue necesaria la comprensión de los conceptos físicos del sonido y cinemática para calcular la distancia entre el robot y un obstáculo.

### 5.1. Escalabilidad

El robot puede levantar más peso a medida que su chasis es más firme y los motores tienen mayor fuerza. También se puede instalar un módulo WiFi al robot y de esta manera puede controlarse el robot de forma remota, es decir, sin necesidad de tener una computadora cerca al robot como sucede con el Bluetooth. El software para controlar el robot puede ser implementado en plataformas móviles como Android o iOS.

Para un almacén con una gran cantidad de robots sería necesario usar un algoritmo de búsqueda de rutas. Hart, Nilsson y Raphael (1968) propusieron un algoritmo llamado *A\* Search Algorithm* que podría permitir obtener las instrucciones para sacar al robot del almacén con el menor número de pasos.

## Referencias

- Serway, R. A., Vuille, C., & Faughn, J. S. (2010). *Fundamentos de física* (Octaván. ed.). México, D.F: Cengage Learning.
- Hewitt, P. G. (2007). *Física conceptual* (Décimán. ed.). México, D.F: Pearson Educación.
- Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. IEEE Transactions on Systems Science and Cybernetics SSC4.