**Advances in Applied Sciences**
2023; 8(4): 122-130
http://www.sciencepublishinggroup.com/j/aas
doi: 10.11648/j.aas.20230804.11
ISSN: 2575-2065 (Print); ISSN: 2575-1514 (Online)

# Design of a Multi-Directional Autonomous Mobile Robot with Robotic Arm for Object Retrieval in Manufacturing Systems

**Tri Dung Nguyen, Trung Hao Tran, Duy Phuong Dinh**

Industrial Maintenance Training Center, Ho Chi Minh City University of Technology (VNU-HCM), Ho Chi Minh City, Vietnam

## Email address:
tridung.nguyen92@gmail.com (Tri Dung Nguyen)

## To cite this article:
Tri Dung Nguyen, Trung Hao Tran, Duy Phuong Dinh. Design of a Multi-Directional Autonomous Mobile Robot with Robotic Arm for Object Retrieval in Manufacturing Systems. *Advances in Applied Sciences*. Vol. 8, No. 4, 2023, pp. 122-130.
doi: 10.11648/j.aas.20230804.11

**Received**: September 26, 2023; **Accepted**: October 13, 2023; **Published**: October 28, 2023

**Abstract:** Transportation in warehouses and production workshops is a matter of urgency today. Most warehouses arrange routes for circulation along the shelves, transportation vehicles will move on this road to perform the task of exporting or importing goods. Routes will be arranged to move in one direction because vehicles do not have enough space to turn around in cramped warehouses. This causes many difficulties in planning the trajectory for transportation vehicles, especially self-propelled vehicles. In our study, we present the design and development of a self-propelled robot capable of omnidirectional movement and equipped with a 3-DoFs serial manipulator for object manipulation. The robot employs an image processing algorithm for object classification based on colors recognition. It utilizes four stepper motors for Mecanum wheels motion and three stepper motors to control the robotic arm. An Arduino Mega board enables precise control of these motors. The mechanical model is designed using Solidworks, with the robotic arm 3D printed and the mobile robot's structure laser-cut from Mika sheets. The final prototype is assembled by integrating these components. The image processing algorithm enhances object classification, accurately detecting objects based on color. Additionally, a line tracking algorithm aids in efficient path navigation. The robot autonomously moves to designated locations, picks up objects with its robotic arm, and transports them as needed. With omnidirectional mobility, robotic arm manipulation, and intelligent image processing, this robot proves valuable for various applications.

**Keywords:** Mobile Robot, Serial Robotic Arm, Mecanum Wheel, Omni-Directional, Object Retrieval

## 1. Introduction

Intelligent manufacturing and logistics have driven significant advancements in mobile robot technology in recent years. The demand for autonomous mobile robots in smart factories has grown substantially, necessitating robots that are flexible, adaptable, and safe to navigate complex work environments [1-5].

To address the requirements of such environments, the choice of wheels for mobile robots is of utmost importance. Omnidirectional wheels, equipped with rollers on their edges, solve the limitations of traditional wheels [6]. These rollers enable movement in one direction while allowing free sliding in the perpendicular direction. By using three or more of these wheels, robots gain full control over their movement, enabling free motion and rotation on a plane without the need for turning the wheels. This design simplifies the robot's overall structure and enhances maneuverability.

Mecanum wheels, a type of omnidirectional wheel, further enhance mobility by incorporating small rollers inclined at a 45-degree angle to the main axis of rotation. This unique design empowers the robot to make complex movements in various directions within confined spaces. Utilizing Mecanum wheels in forklifts, for example, significantly improves operational efficiency, thereby enhancing productivity for businesses. A robot moving in a plane will have a maximum of 3 degrees of freedom (DoFs). So that, the AGV must have at least three wheels [10-16]. In the field of industrial, four Mecanum wheels AGV is widely used. There are many kinds of 4-wheeled Mecanum configuration

for a mobile platform. However, in order to implement an omnidirectional AGV, it is necessary to select a properly configuration. To ensure the omni-directional movement ability, the axes of bottom roller of any three wheels have to intersect at two points [9].

Combining mecanum-wheeled omnidirectional robots with 3-degree-of-freedoms (DoFs) robotic arms addresses the mobility drawbacks of traditional robotic arms. Integrating the movement of omnidirectional wheels with the robot arm provides a solution to this limitation. However, achieving high accuracy in such integration can pose challenges [7].

Overall, the combined use of mecanum wheels and robotic arms represents a promising advancement in mobile robot technology, offering increased flexibility and efficiency for various industrial applications. Yet, careful consideration must be given to maintaining precision and accuracy during such integration.

# 2. Methodology

## *2.1. Mechanical Design*

The mechanical model is designed on SolidWorks environment. The model consists of 2 different mechanical parts that work together to ensure quick and accurate moving and picking. Parts used in the picture include a mobile robot and a robot arm.

The robot arm consists of three main clusters: the transmission assembly of joint 1, the transmission assembly of joints 2 and 3, and the lifting assembly.

Let's focus on the first drive assembly, as depicted in Figure 1. This assembly includes the motor (1) securely attached to the positioning surface (2) through a bolt joint. The motor shaft is connected to the driving pulley (5) via a belt, enabling the transmission of motion to the passive pulley (6), which in turn rotates the robot body (7). To ensure smooth rotation and reduce axial load, a ball bearing (4) is incorporated. Additionally, ball bearings (3) and (8) are used to fix the mechanical details in place.

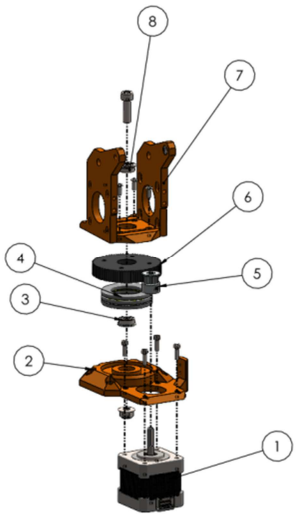



***Figure 1.*** *Actuator assembly of 1st joint.*

The 2nd and 3rd joint transmission assemblies consist of motors (2) and (3), securely fastened to the locating surface (1) using bolted joints. Each motor shaft is connected to an active pulley (4) and (5) respectively, through belts, allowing the transmission of motion. This motion is then transferred to the passive pulleys (6) and (7), which in turn drive the driving rods (8) and (9). These driving rods are responsible for generating the necessary rotation angles for joint 2 and joint 3, as illustrated in Figure 2.

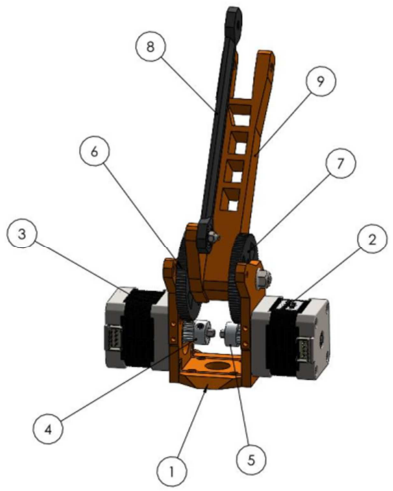



***Figure 2.*** *Actuator assembly of 2nd and 3rd joints.*

The lifting assembly shown in Figure 3 consists of a vacuum cup (1), which is securely attached to the locating surface (2) using a nut (3) and fixed to the connection face (4) through a bolt joint. This cluster serves the purpose of lifting objects when the vacuum pump is activated, ensuring that the vacuum suction cup holds onto the object.

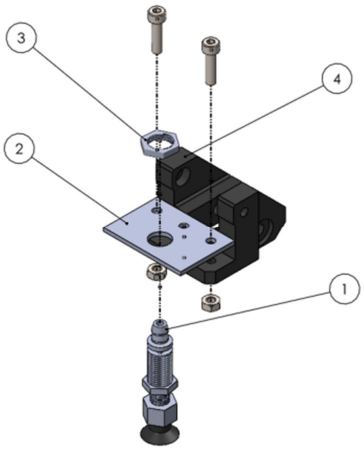



***Figure 3.*** *The lifting assembly.*

The mobile robot consists of two main assemblies: the robot body assembly and the motor assembly. The robot body is made up of mika plates linked together by bolts and copper pillars. This assembly includes stepper motor mount (1), side panel (2), robot arm (3), electrical control box 1 (4), electrical control box 2 (5), front and rear radiator fans (6),

battery holder (7) and 3 cell 18650 battery base (8) are shown as Figure 4.

The motor assembly responsible for the mobile robot's movement consists of a stepper motor (1) and a stepper motor mount (2). The motor shaft is connected to the Mecanum wheel (6) through a shaft coupling (4) secured in place by a bolt (3) and a flat head screw (7) shown in Figure 5.

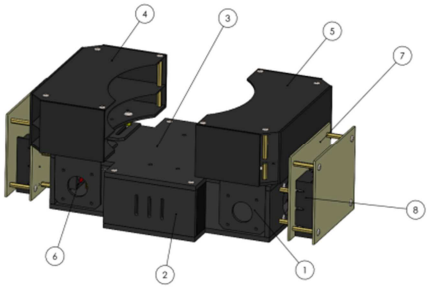


***Figure 4.*** *Robot body assembly.*

In this project we use two Arduino Mega boards to control the robot. One board is dedicated to controlling the robot arm, while the other is used for the mobile robot. Both boards are responsible for sending pulse signals to the motor driver, thus allowing precise control of the motor and generating rotation. In addition, we also use a Raspberry Pi embedded computer combined with a Logitech HD C270 camera to perform image processing. Details of the connections are shown in Figure 6.

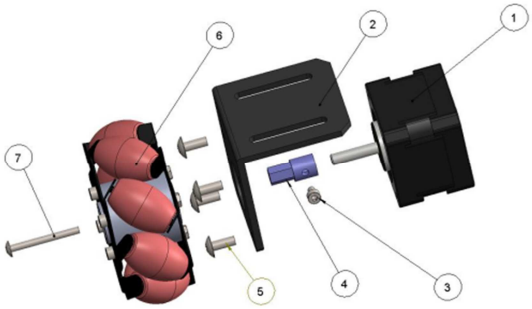


***Figure 5.*** *Engine assembly.*

### *2.2. Electrical Module Connection*

#### *2.2.1. The Arduino Mega*

*Arduino Mega Board* is a robust microcontroller board that utilizes the ATmega2560 chip. With its 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports), and a 16 MHz crystal oscillator, it provides extensive capabilities for driving various electrical components. In the robotic arm, the Arduino Mega board is responsible for controlling the stepper motors, which enable the movement of the joints. It also reads signals from the limit switches and controls the air pump motor for lifting objects. For the mobile robot, the Arduino Mega board reads signals from the TCRT5000 sensor module and then outputs signals to control the stepper motors, allowing the wheels to move. The Arduino board's significant advantage lies in its compatibility with the Arduino Library. These libraries are open source and readily available for seamless integration into the Arduino environment. With the help of these libraries, we can easily control and manage other devices and components, making the development process more straightforward and enhancing the robot's overall functionality.

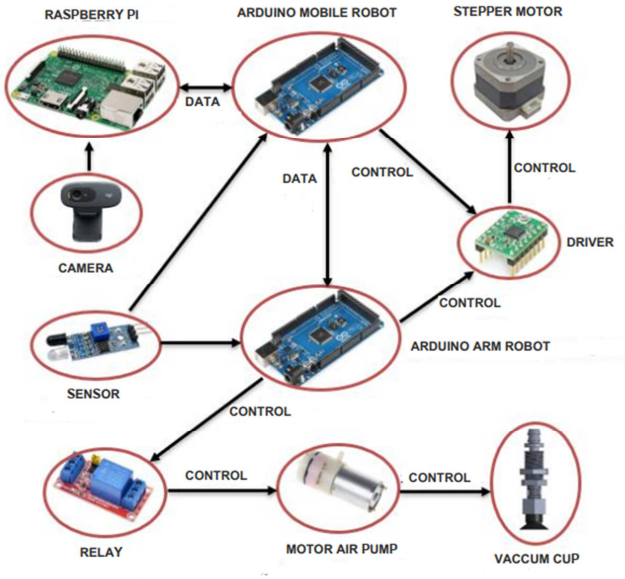



***Figure 6.*** *Diagram of the connection principle between devices.*

#### *2.2.2. The Raspberry Pi*

*Raspberry Pi* is a compact and energy-efficient computer circuit board equipped with a CPU, GPU, USB port, and GPIO pins. It runs on the Linux operating system and provides the capability to program embedded systems, perform image processing, and execute tasks similar to a regular computer.

In this project, the Raspberry Pi is combined with the Logitech HD C270 Webcam to create a signal processing system. The camera captures colour images and sends them to the Raspberry Pi for processing. The primary objective is to identify the blue object and determine its centre of gravity. Once this information is obtained, the coordinates of the object's centre of gravity are sent to the microcontroller responsible for controlling the robot.

#### *2.2.3. The A4988 Stepper Motor Driver Circuit*

This is an extremely compact stepper motor driver, supporting many working modes, adjusting the output current for the motor, and automatically disconnecting the power when it overheats. The A4988 driver circuit supports many operating modes of bipolar stepper motors such as Full, 1/2, 1/4, 1/8, and 1/16.

#### *2.2.4. Limit Switches*

These are used to determine the home position of the robot. When the robot arm is started, the joints are controlled to rotate till reaching the limit switches and halt.

### *2.2.5. Module Relay 5V*

They are used to control other devices with the large current and voltage. The output voltage from Arduino is only 5V. So, to control other device with the higher voltage (12V or 24V), the module relay is used. The module relay receives the 5V signal from the microcontroller to turn on an photocoupler inside this. The photocoupler will conduct current through the coil of the relay.

### *2.2.6. Arduino CNC Shield V3*

An expansion shield for Arduino Uno and Arduino Mega 2560, allowing one to control laser engraving machines, CNC milling machines, or mini 3D printers. The shield allows control of up to 4 stepper motors via the A4988 or DRV8825 driver (with jumpers to control the stepper motor in full step, half step, 1/4, 1/8, or 1/16 mode). In addition, it is also possible to add travel switches for the X, Y, Z, and E axes (only for 3D printers) or control the CNC engraving head, laser engraving head, and cooling fan.

### *2.2.7. The TCRT5000 Sensor Module*

This sensor module is designed with 5 TCRT5000 infrared sensors lined up to detect the contrast of different colours such as black and white, commonly used for making line detection robots (track the line according to the colour line).

### *2.2.8. The RF Transceiver Circuit NRF24L01*

A communication module that is used for applications of remote data transmission via RF waves between central processing circuits such as Microcontrollers, Arduino or Raspberry Pi, etc. In this project, because Raspberry Pi is placed in a fixed position and Arduino is placed on a robot, it requires wireless data transmission so we decided to use the NRF24L01 circuit.

### *2.2.9. The Mini Vacuum System*

They are used to create a vacuum environment with extremely low pressure inside the vacuum cup to help hold objects, the compact design suitable for small projects.

### *2.3. Kinetics of the Robot Arm*

This study presents a novel approach, combining a 3-DoFs serial robotic arm with a mobile robot equipped with omnidirectional wheels, resulting in the development of an AGV (Automated Guided Vehicle) with exceptional flexibility. We conducted a comprehensive analysis of the kinematics of the robot arm. We will apply the standard D-H parametric method to establish the coordinate system for each joint. The obtained D-H parameters are summarized in Table 1.

***Table 1.*** *DH parameters.*

| Joint no. i | Link length $a_i$ | Link Twist $\alpha$ | Link offset $d_i$ | Joint angle $\theta_i$ |
|---|---|---|---|---|
| 1 | 0 | 90 | $L_1$ | $\theta_1$ |
| 2 | $L_2$ | 0 | 0 | $\theta_2$ |
| 3 | $L_3$ | 0 | 0 | $\theta_3$ |

Based on the DH parameters the forward kinematics can be calculated through equation (1):

$$ {}^{0}_{3}T = {}^{0}_{1}T \times {}^{1}_{2}T \times {}^{2}_{3}T \tag{1} $$

where ${}^{i}_{j}T$ denotes the homogenous transformation from frame i to frame j and

$$ {}^{i}_{i+1}T = \begin{bmatrix} c\,\theta_i & -s\,\theta_i\,c\alpha_i & s\,\theta_i\,s\,\alpha_i & a_i c\,\theta_i \\ s\,\theta_i & c\theta_i\,c\,\alpha_i & -c\,\theta_i\,s\,\alpha_i & a_i s\,\theta_i \\ 0 & s\,\alpha_i & c\,\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

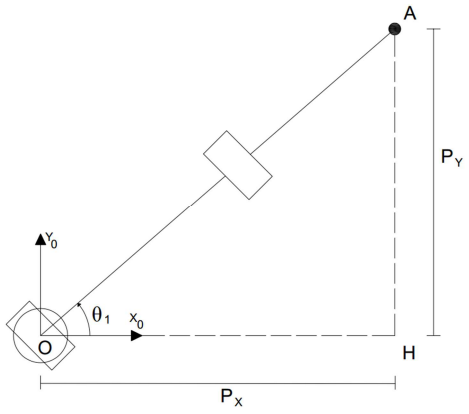
where $c = cos, s = sin$ and $i = 1,2,3$.

By substituting these parameters listed in Table 1 to (2), we can obtain the transformation matrix as shown in (3):

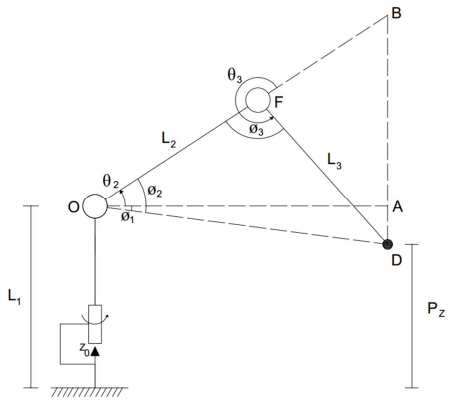
$$ {}^{0}_{3}T = {}^{0}_{1}T \times {}^{1}_{2}T \times {}^{2}_{3}T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & P_x \\ r_{21} & r_{22} & r_{23} & P_y \\ r_{31} & r_{32} & r_{33} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

In matrix (3), the first three columns represent the orientation of the end-effector and the last column shows the position of the end-effector. Therefore, the position of the end-effector denoted $P\ (P_x, P_y, P_z)$ is shown below:

$$ \begin{aligned} P_x &= c(\theta_1)(L_3 c(\theta_2 + \theta_3) + L_2\, c(\theta_2)) \\ P_y &= s(\theta_1)(L_3 c(\theta_2 + \theta_3) + L_2 c(\theta_2) \\ P_z &= L_1 + L_3 s(\theta_2 + \theta_3) + L_2 s(\theta_2)) \end{aligned} \tag{4} $$




(a) Top view




(b) Front view

***Figure 7.*** *Projections of 3-DoF robot arms.*

After analyzing the forward kinematics of the robot arm, next comes the analysis of its reverse kinematics. There are many methods to solve inverse kinematics problems such as the algebraic method, geometric method, Pieper's method, etc. In this paper, the problem of inverse kinematics is presented by geometric methods.

Figure 7 below shows the projection of the robot arm in practice. From these figures will calculate the joint angles $\theta_1$, $\theta_2$, $\theta_3$.

As shown in Figure 7, we convention $r_1$ = OA, $r_2$ = OD, and $r_3$ = OB for easy calculation.

With the positive direction of $\theta_1$, $\theta_2$, $\theta_3$ as shown in Figure 7 and the possible case of the robot arm in reality, we have the conditions to eliminate the solution:

$$\begin{cases} \theta_2 > 0 \\ \theta_3 > 180° \end{cases} \quad (5)$$

According to Figure 7(a) we can calculate $\theta_1$ by Eq. (6) and Eq. (7).

$$r_1 = \sqrt{(P_X)^2 + (P_Y)^2} \quad (6)$$

$$\sin\theta_1 = \frac{P_Y}{r_1} \quad (7)$$

According to Figure 7(b) we can calculate $\emptyset_1$ and $\emptyset_2$ by Eq. (8) from there as the basis for calculating $\theta_2$.

$$\begin{cases} r_2 = \sqrt{(L_1 - P_Z)^2 + r_1^2} \\ \cos\emptyset_2 = \frac{L_3^2 - L_2^2 - r_2^2}{-2L_2 r_2} \\ \sin\emptyset_1 = \frac{L_1 - P_Z}{r_2} \end{cases} \quad (8)$$

Use equation (5) to choose a solution for $\emptyset_1$ and $\emptyset_2$. From there, we can calculate $\theta_2$ by Eq. (9).

$$\theta_2 = \emptyset_2 - \emptyset_1 \quad (9)$$

Continue to calculate $\emptyset_3$ using equation (10) and use equation (5) to choose the solution. Finally, we can calculate $\theta_3$ using formula (11).

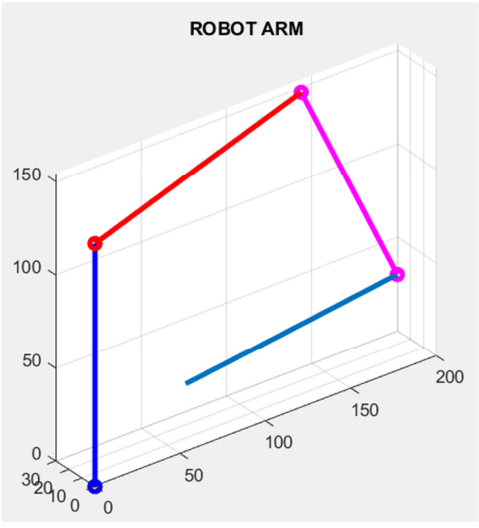



***Figure 8.*** *Simulate the trajectory of the robot arm.*

$$cos(\emptyset_3) = \frac{r_2^2 - L_2^2 - L_3^2}{-2L_2L_3} \quad (10)$$

$$\theta_3 = 180° + \emptyset_3 \quad (11)$$

To test the accuracy of the inverse kinematics equations, we use MATLAB software to simulate the motion trajectory of the robot arm. Let the robot arm move with constant altitude from position A (6, 0.9, 30) to position B (200, 30, 30) with a division of 0.01, then it will create a trajectory as a straight line, as shown in Figure 8.

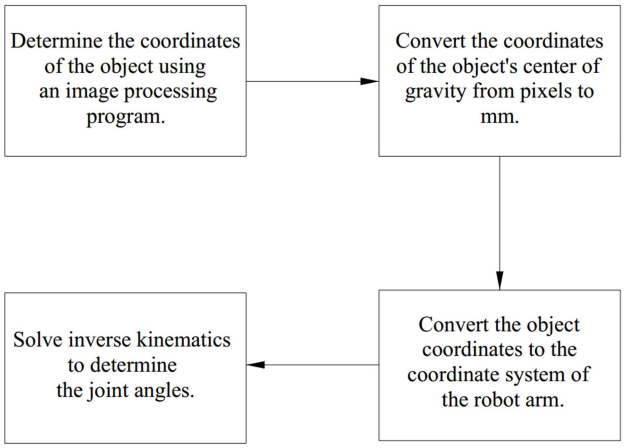



***Figure 9.*** *Steps to apply image processing to the robot arm.*

### *2.4. Image Processing Algorithms*

Applying image processing technology to determine coordinates and classifying objects, combined with the previously solved inverse kinematics problem, helps the robot arm automatically pick up objects. Figure 9 details the steps to apply image processing for the robot arm.

In this project, the computer vision system was built with a Raspberry Pi 3 embedded computer for image processing and a C270 HD camera with a resolution of 1280 × 720 pixels and a field of view of 60 degrees used to capture objects. The image processing program running on the Raspberry Pi 3 embedded computer written in Python and using the OpenCV library will resolve with two main objectives: determining the centre of gravity of the objects and classifying them based on their colours. In the scope of this topic, we are solely interested in identifying the blue objects. Hence, we will set the colour threshold specifically for detecting the blue colour. To improve processing speed, images taken from the camera are set at a resolution of 640 × 480 pixels. The original image is then converted from the RGB (Red, Green, Blue) colour space to the HSV (Hue, Saturation, Value) colour space. The HSV colour space separates colour information into three components, making it easier to manipulate specific aspects of an image, such as hue, saturation, and brightness. This approach allows for better control and understanding of the colours present in the image.

To classify objects based on colour, specific upper and lower colour threshold values are set in the HSV colour space. For the blue colour range, the lower bound parameters are (90, 60, 0), and the upper bound parameters are (121, 255,

255). These values determine the range of colours considered as "blue" in the image. Any pixel in the image with HSV values falling within this range will be identified as blue, allowing for object detection or segmentation based on colour.

Using the HSV colour space and defining appropriate colour thresholds is a common technique in computer vision and image processing to isolate and analyse specific colours in an image efficiently and accurately.

Following the steps outlined in Figure 9, once the object has been classified, and its centre of gravity determined, we proceed to convert the measurements from pixels to millimetres. Considering an actual resolution of 640×480 pixels, we obtain the corresponding frame size in millimetres and scale it accordingly. Next, we multiply the scaled frame size by the coordinates of the object's centre of gravity, which were obtained through the image processing program. This completes the unit conversion. However, it's important to note that the object's centre of gravity coordinates is relative to the frame's origin, so an additional step is required to convert them to the robot arm's origin. To achieve this, we transform the coordinates of the object's centre of gravity, making them relative to the robot arm's origin. Subsequently, we input these converted coordinates into the inverse kinematics equations, enabling us to determine the values of the robot arm's joint angles. From these joint angle values, we can then calculate the number of pulses needed for the stepper motor, allowing the robot to accurately reach the desired position.

### *2.5. Line Tracking Algorithms*

To design a line detection algorithm for an omnidirectional mobile robot, it is necessary to consider its kinematics. An omnidirectional mobile robot using 4 Mecanum wheels with a coordinate system mounted at the center of the robot is shown in Figure 10. Both the forward kinematics (FK) and inverse kinematics (IK) problems of the robot are considered.

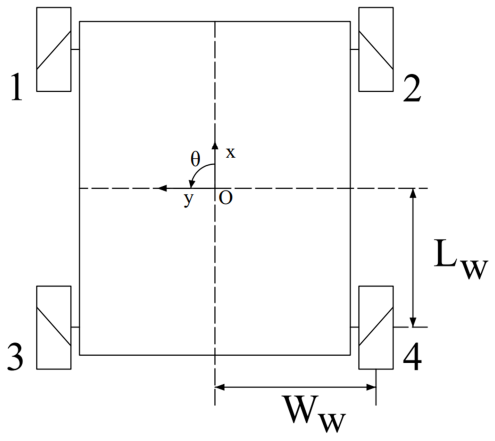



***Figure 10.*** *Mobile robot using Mecanum wheel.*

Forward kinematics focuses on determining the vehicle's motion using the angular velocities of the wheels, while inverse kinematics utilizes the vehicle's velocity to calculate the corresponding angular velocities of the wheels. Equations (12) and (13) represent the FK and IK, respectively, and further details can be found in [8].

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \frac{R_w}{4}\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & -1 & 1\\ -\frac{1}{L_w+W_w} & \frac{1}{L_w+W_w} & -\frac{1}{L_w+W_w} & \frac{1}{L_w+W_w}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} \quad (12)$$

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \frac{1}{R_w}\begin{bmatrix}1 & 1 & -(L_w+W_w)\\ 1 & -1 & L_w+W_w\\ 1 & -1 & -(L_w+W_w)\\ 1 & -1 & L_w+W_w\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} \quad (13)$$

In formulas (12) and (13), $R_w$ is the radius of the wheel, and assume that $R_1 = R_2 = R_3 = R_4 = R_w$ for the calculation. Furthermore, $\omega_w = [\omega_1, \omega_2, \omega_3, \omega_4] \in R^4$ is the angular velocity of each wheel, respectively. $L_w$ is the distance between the wheel and the vehicle's center of gravity in length and $W_w$ is the distance between the wheel and the vehicle's center of gravity in width.

Velocity and angular velocity obtained from the above equations are provided in the vehicle's frame of reference. However, for the purpose of line detection in this study, it is necessary to establish a model that describes the vehicle's position and direction in the global reference system. To achieve this, we introduce the $V_G = [x_G y_G \theta_G]^T$ vector, which represents the vehicle's position and orientation in the global frame. By utilizing the rotation matrix, the forward kinematics in the global frame can be represented by equation (14):

$$V_G = J^+(\theta) \times v_w \quad (14)$$

where $\mathrm{v_w} = [\mathrm{v_{1w}}\ \mathrm{v_{2w}}\ \mathrm{v_{3w}}\ \mathrm{v_{4w}}]^{\mathrm{T}}$ is the respective velocity of each wheel; $V_G$ is the velocity vector in the global frame and the matrix $J^+(\theta)$ is represented as follows:

$$J^+(\theta) = \frac{1}{4}\begin{bmatrix}\sqrt{2}\,\mathrm{s}(\theta_1) & \sqrt{2}\,\mathrm{c}(\theta_1) & \sqrt{2}\,\mathrm{c}(\theta_1) & \sqrt{2}\,\mathrm{s}(\theta_1)\\ -\sqrt{2}\,\mathrm{c}(\theta_1) & \sqrt{2}\,\mathrm{s}(\theta_1) & \sqrt{2}\,\mathrm{s}(\theta_1) & -\sqrt{2}\,\mathrm{c}(\theta_1)\\ -\frac{1}{L_w+W_w} & \frac{1}{L_w+W_w} & -\frac{1}{L_w+W_w} & \frac{1}{L_w+W_w}\end{bmatrix} \quad (15)$$

where $c = cos, s = sin$

Matrix $J^+(\theta)$ is the inverse of the matrix $J(\theta)$:

$$J(\theta) = \begin{bmatrix}\sqrt{2}\,\mathrm{s}(\theta_1) & -\sqrt{2}\,\mathrm{c}(\theta_1) & -(L_w+W_w)\\ \sqrt{2}\,\mathrm{c}(\theta_1) & \sqrt{2}\,\mathrm{s}(\theta_1) & (L_w+W_w)\\ \sqrt{2}\,\mathrm{c}(\theta_1) & \sqrt{2}\,\mathrm{s}(\theta_1) & -(L_w+W_w)\\ \sqrt{2}\,\mathrm{s}(\theta_1) & -\sqrt{2}\,\mathrm{c}(\theta_1) & (L_w+W_w)\end{bmatrix} \quad (16)$$

where $\theta_1 = \theta + \frac{\pi}{4}$ and $\theta$ is the angle of deviation of the vehicle's center from the line that will be presented in the following section.

Finally, the inverse kinematics model of the mobile robot in the global framework is represented by:

$$v_w = J(\theta)V_G \quad (17)$$

The design of the line tracking algorithm plays a crucial role in enabling the new robot to follow the line and move

with stability. For this project, a single infrared sensor module TCRT5000 (with 5 channels) will be utilized. The sensor module will be positioned 25 mm from the center of the vehicle. The arrangement of the sensor is depicted in Figure 11.

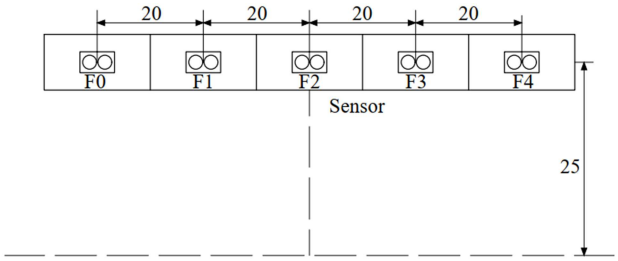



***Figure 11.*** *Position of sensor module TCRT5000 on mobile robot.*

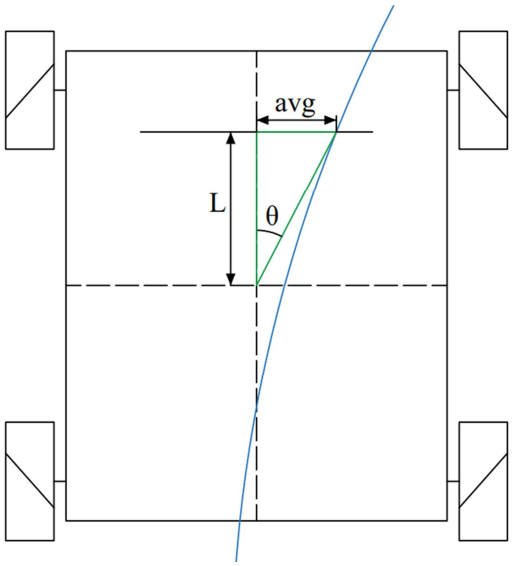



***Figure 12.*** *The angle of deviation of the center of the vehicle relative to the line.*

According to Figure 12 we can calculate angle $\theta$ as equation (16):

$$\begin{cases} avg = \frac{\sum(F(i)\times(i-2)\times 20)}{\sum F(i)} \\ \tan\theta = \frac{avg}{L} \end{cases} \quad (18)$$

where *avg* is the distance of the line centerline to the vehicle centerline at the sensor; *F(i)* is the digital value returned by sensor i; *i* is the serial number of each sensor in the module (values from 0 to 4).

When calculating the angle θ, what value will we put into equation (17) from which we can calculate the angular velocity of each wheel so that the robot can follow the line.

## 3. Experimental Results

The mechanical system and the electrical system are fully connected to create a mobile robot using omnidirectional wheels and the robot arm as shown in Figure 13.

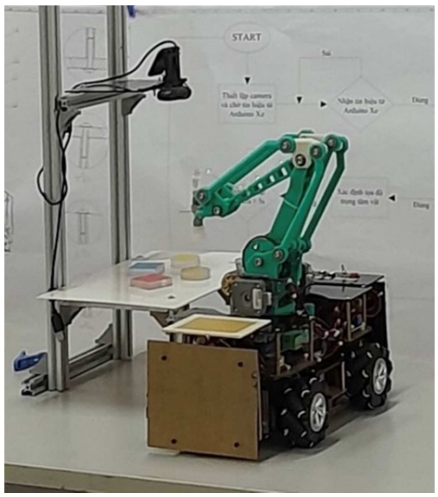



***Figure 13.*** *Experiment robot.*

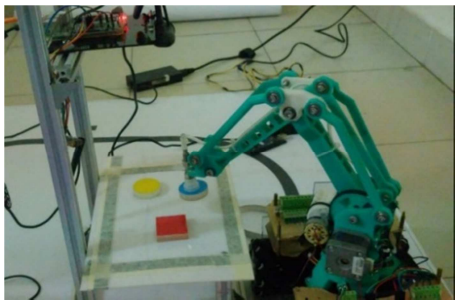
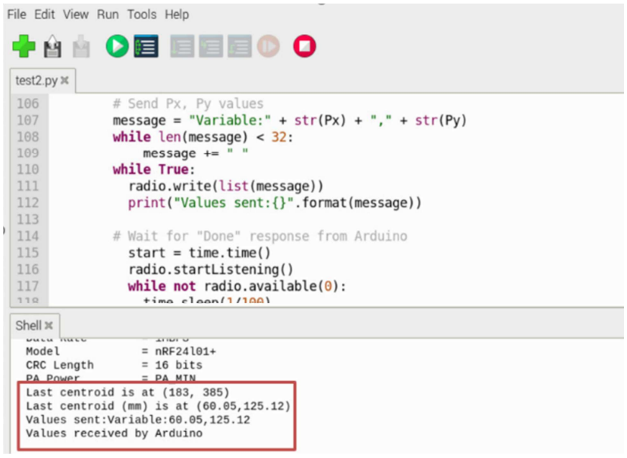
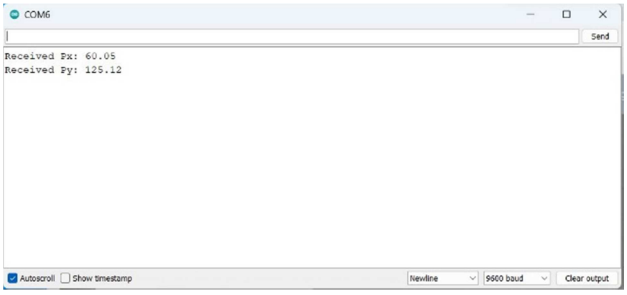


***Figure 14.*** *The robot is lifting an object.*



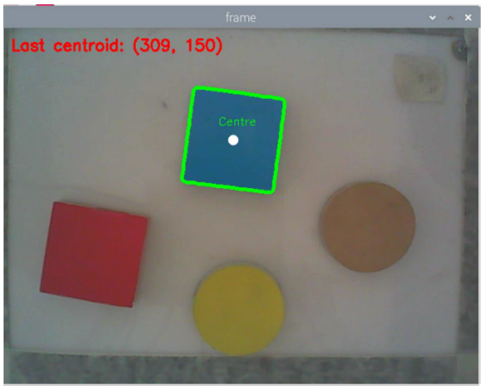

(a) The data is sent from the Raspberry Pi




(b) Data received on the Arduino Mega of the robotic arm.

***Figure 15.*** *Data transmission between Raspberry Pi and Arduino Mega.*

Upon powering on, the robot follows a predefined path to approach the targeted object and subsequently executes the object-picking operation, as depicted in Figure 15. The data transmission process between the Raspberry Pi embedded computer and the Arduino board controlling the robot arm is illustrated in Figure 14. After picking up the object, the robot will move to the drop position and proceed to drop the object, then return to the start position.




(a) The square blue object is identified


(b) The circled blue object is identified

***Figure 16.*** *Detect object with blue color.*

Figure 16 shows a graded blue object and its centre of gravity defined in pixels. Indeed, the application of suitable HSV colour thresholds for each specific colour greatly facilitates the straightforward classification of objects based on their colours. When this colour classification capability is combined with the solution to the inverse kinematics problem of the robot arm, it empowers the robot to perform automated tasks of picking up and dropping objects with precision and efficiency.

## 4. Discussion and Conclusion

The article presents the development and testing of a highly flexible mobile robot, which combines omnidirectional wheels and a 3-degree-of-freedom robotic arm to execute tasks in a production environment. The robot is capable of autonomously picking up and dropping objects using image processing techniques. The robot's design begins with software-based planning, followed by 3D printing for the robotic arm and laser cutting of plastic material for the mobile robot, ultimately leading to the assembly of a fully functional model.

To control the robot, two Arduino boards are employed: one for managing the robot arm and the other for controlling the mobile robot. Additionally, a Raspberry Pi embedded computer is utilized for image processing tasks. The NRF24L01 circuit facilitates wireless communication between the Arduino boards and the Raspberry Pi, enabling seamless and flexible movement of the robot.

The integration of the kinematics problem of the robot arm with the image processing algorithm enables the robot to effectively detect and pick up objects. Furthermore, a line detection algorithm is implemented using the infrared sensor module TCRT5000, which allows the robot to navigate its path.

Experimental results demonstrate the robot's ability to successfully detect and pick up blue objects, utilizing the thresholding method in the HSV color system. Overall, the developed robot showcases promising performance in real-world scenarios, showcasing its potential for various applications in automated tasks and production environments.

## References

[1] A. Odry, R. Fuller, I. J. Rudas, P. Odry, Kalman filter for mobile-robot attitude estimation: novel optimized and adaptive solutions, Mechanical Systems and Signal Processing, vol. 110, pp. 569–589, 2018.

[2] K. J. Kalinski, M. Mazur, Optimal control of 2-wheeled mobile robot at energy performance index, Mechanical Systems and Signal Processing, vol. 70, 2016, 373–386.

[3] S. Zhao, B. Huang, F. Liu, Localization of indoor mobile robot using minimum variance unbiased FIR filter, IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp 410–419, 2018.

[4] P. Kassaeiyan, B. Tarvirdizadeh, K. Alipour, Control of tractor-trailer wheeled robots considering self-collision effect and actuator saturation limitations, Mechanical Systems and Signal Processing, vol. 127, pp. 388–411, 2019.

[5] Y. Kantaros, M. M. Zavlanos, Distributed intermittent connectivity control of mobile robot networks, IEEE Transactions on Automation Science and Engineering, vol. 62, no. 7, pp 3109–3121, 2017.

[6] A. M. Univ, E. C. Marseille, ST. M. Rousset, Trajectory reconfiguration for time delay reduction in the case of unexpected obstacles: application to 4-mecanum wheeled mobile robots (4-MWMR) for industrial purposes, vol. 53, no. 2, pp 15653-15658, 2020.

[7] G. Ding, J. Bai, H. Lu, X. Qin, Development of a High Precision UWB/Vision-based AGV and Control System, 2020 the 5th International Conference on Control and Robotics Engineering, vol. 978, no. 1, pp 7281-6791, 2020.

[8] Z. Sun, H. Xie, J. Zheng, Z. Man, D. He, Path-following control of Mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode, Mechanical Systems and Signal Processing, vol. 147, pp. 107128, 2021.

[9] Y. Li, S. Dai, L. Zhao, X. Yan, and Y. Shi, Topological design methods for mecanum wheel, *Journal of Symmetry in Engineering Sciences II*, Vol. 11, No. 10, pp. 1-27, 2019.

[10] I. Doroftei, V. Spinu, and V. Grosu, "Omnidirectional mobile robot – design and implementation", *Journal of Bioinspiration and Robotics: Walking and Climbing Robots*, pp. 544, 2007.

[11] P. Alvito, C. Marques, P. Carriço, and J. Freire, A Robotic Platform for the Social Robot Project, In Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication (ROMAN 2014) Workshop on Interactive Robots for Aging and/or Impaired People, Edinburgh, UK, 25–29 August 2014.

[12] J. Qian, B. Zi, D. Wang, Y. Ma, and D. Zhang, the design and development of an omni-directional mobile robot oriented to an intelligent manufacturing system, *Journal of Sensors* 2017, pp. 17, 2073.

[13] F. Adăscăli¸tei and I. Doroftei, Practical applications for mobile robots based on mecanum wheels-a systematic survey, In Proceedings of the 3rd International Conference on Innovations, Recent Trends and Challenges in Mechatronics, Mechanical Engineering and New High-Tech Products Development (MECAHITECH'11), Bucharest, Romania, 22–23 September 2011; pp. 112–123.

[14] P. Hryniewicz, A. Gwiazda, W. Banas, A. S'ekala, and K. Foit, Modelling of a mecanum wheel taking into account the geometry of road rollers, *In Proceedings of the IOP Conference Series: Materials Science and Engineering*, Sibiu, Romania, 14–17 June 2017; p. 012060.

[15] C. He, D. Wu, K. Chen, F. Liu, and N. Fan, Analysis of the Mecanum wheel arrangement of an omnidirectional vehicle, *Proc. Inst. Mech. Eng. Part C J. Mech*. Eng. Sci. 2019, 233, 5329–5340.

[16] Y. N. Zhang, S. Wang, J. Zhang, and J. Song, Research on motion characteristic of omnidirectional device based on Mecanum wheel, In Proceedings of the 2011 International Conference on Electric Information and Control Engineering, Wuhan, China, 15–17 April 2011; pp. 6094–6097.