

# The Application of Image Moments and Gaussian Blur for Line Detection in Differential-Drive Mobile Robots

Mohammad Shamoushaki  
Babol Noshirvani University of Technology

Mohammad Hasan Ghasemi  
Associate Professor of Mechanical Engineering  
Babol Noshirvani University of Technology

## ABSTRACT

Over the last several years, a multitude of research studies have been carried out in the field of artificial intelligence to set up autonomous systems. Various research papers have used machine learning methods to enhance the efficiency of robots. Utilizing the information captured by the camera or evaluating the data acquired by diverse sensors in systems will also optimize their performance. The upcoming research investigates the programming of a differential-drive mobile robot (DDMR) capable of following environmental features in a laboratory. The robot processes RGB images using image moments and a Gaussian filter. Furthermore, the robot was simulated in ROS to verify the effectiveness of the image processing method. Obstacle avoidance was performed using ultrasonic sensors and the Bubble Rebound method. Furthermore, image moments assist in detecting the center points of guidelines and subsequently enable maintaining or changing lanes as required. A model has been trained to detect environmental signals using the Haar Cascade Classifier. The primary purpose of this paper is to study the application of one of the fundamental image features, called image moments, in lane detection, along with masking and feature-based image processing. Additionally, the effect of different sizes of kernel matrix in the Gaussian filter for noise reduction has been compared. The robot was eventually tested in a laboratory environment to validate the method and scenario.

## Keywords

Differential-Drive Mobile Robot, The Bubble Rebound Algorithm, Lane Detection, Haar Cascade Classifier, Image Moments, Gaussian Blur

## 1. INTRODUCTION

With the advancement of technology in artificial intelligence, various industries, such as food, pharmaceutical, agriculture, transportation, manufacturing, and packaging, have developed significantly. Also, in recent years, the demand for industrial automation to increase efficiency and reduce the wastage of resources has been felt more than ever. Therefore, significant research has been conducted on controlling autonomous systems using artificial intelligence to contribute to various industries' development. In the meantime, the development of industrial robots has gained significant importance. Among those types, the differential-drive mobile robots are the most prevalent. In this section, a brief introduction to some of the interdisciplinary studies of robots and the definition of visual servoing will be discussed.

### 1.1 Interdisciplinary Studies of Robots

The significance of developing autonomous robots has grown with the progress made in artificial intelligence and electronic devices. Autonomous robots have to deal with various tasks, including obstacle avoidance, navigation, perception, path planning, and decision making, which will be reviewed

respectively. This needs interdisciplinary studies on which scientists from all majors have conducted research. Among the issues related to autonomous robots' algorithms is how to avoid obstacles. In 2015, Bhavesh addressed an issue with the automated navigation of mobile robots, contrasting the various approaches to this challenge and outlining the benefits and drawbacks of each [1]. The study covered everything from the most straightforward "bug" algorithms to the most complicated "optical flow" method. Robot navigation is one of the most critical technologies for robotics and the immediate solution to the issue of intelligent robots. In a study, Guo and Sun [2] analyzed the problem of visual navigation for mobile robots and proposed a monocular image-based vision navigation algorithm for robots. Using the heuristic best search algorithm, they optimized and illustrated the robot's navigation path. In 2018, Manivannan and Ramakanth [3] developed control algorithms to detect and steer an intelligent road vehicle in a designated lane using a single camera and image processing techniques. The acquired frames were processed in MATLAB using image acquisition methods. Then, the line was identified through image processing, and the proper movement angle was calculated, helping the robot remain on the designated path. The obtained experimental results demonstrated that the algorithm worked with a maximum error of 2 cm under specific test conditions. In 2021, ABID et al. [4] came up with a way to control the position without sensors by using a single camera and QR codes. It was suggested that a Probabilistic Hough Transform be used to find lanes and QR codes to stay in lanes. They tried this method out on a car named EMO. The suggested strategy outperformed previous approaches in experiments. Regarding the perception of autonomous robots, lane departure warning systems have lately been suggested for development. The most critical aspect of lane tracking, lane detection, and lane departure warning for any system is whether it can detect lanes accurately and rapidly [5]. Besides, accuracy for specific situations and a wide range of scenarios, time efficiency, detection, and visible line tracking were defined as continuing research topics. Currently, machine vision and image processing approaches for recognizing marked lines may be split into two categories: standard image processing methods and semantic segmentation (including deep learning). The former consists primarily of feature-based and model-based stages, including straight lines, curves, and other parametric patterns, which can be categorized as either similar or discontinuous. Semantic segmentation incorporates machine learning, neural networks, and deep learning techniques [6].

### 1.2 Application of Vision in Path Tracking

Mobile robots must be able to locate themselves within their environment. Robot localization techniques consider a wide range of perceptual models. In recent years, position sensors such as Global Positioning Systems (GPS) and Inertial Measurement Units (IMU) have been widely used to estimate the movement of mobile robots. Optical wheel encoders are

position sensors widely used due to their low cost and simplicity. Besides, the use of computer vision to determine the robot's position has been investigated for several decades. Although the majority of research focuses on other sensors such as laser range finders and SONAR, vision remains an attractive choice of sensor due to the low cost and high information content of cameras. One of the methods that is still implemented on robots as a vision technique is called Visual Servoing.

### 1.3 Visual Servoing

Visual Servoing, also known as Vision-Based Robot Control and abbreviated as VS, is a technique that controls the robot's movement using derived feedback from the vision sensor. Visual control techniques are typically categorized as one of the following: 1) Image-Based Visual Servoing (IBVS), 2) Position-Based Visual Servoing (PBVS), and 3) Hybrid approach. Weiss and Sanderson [7] were the first to suggest IBVS. The control law is based on the difference between the current and intended image features and does not include an estimate of the target's position. The coordinates of visual features, lines, or instances within an image can be considered features. In some cases, no information is provided about the coordinates of the case studies in terms of XYZ, where Visual Servoing plays a significant role in positioning using a camera. This method gives the robot a pair of eyes by which the perception of the environment would be possible, and the robot can be controlled using the coordinated attached to the camera. In 2013, Siradjuddin et al. used vision-based robot control (IBVS) techniques to develop a wheeled mobile robot. Using the definition of Instantaneous Center of Curvature (ICC), they tried to check the robot's movement in the controlled path to reach the target identified in the image [8]. Don Joven Agravante et al. used visual servoing in a quadratic optimization problem to handle visual constraints, such as field of view and occlusion avoidance [9]. Brent A. Griffin et al. proposed a method of identification for generic objects in a 3D environment using a mobile robot and an RGB camera with no need for camera calibration. They took advantage of visual servoing in the process of designing a control system [10]. Riccardo Parosi et al. have recently investigated a robust visual servoing approach using an impedance controller. They proved that this method works efficiently under some external disturbances or fast motions of the target object [11]. One of the IBVS challenges [12] can come up during significant rotational movements, also known as camera retreat [13].

### 1.4 Feature-Based Image Processing

LDW (Lane Departure Warning) and lane recognition and tracking technologies for self-driving cars are the essential parts of ADAS (Advanced Driver Assistance System) systems for which image features can be utilized. The difference in color between lines and their backgrounds is a key for drivers to identify where they are. HSV is often used to recognize colors because it breaks down information about hue, saturation, and brightness into three channels: H, S, and V. With certain limits in the H, S, and V channels, it is possible to find colors, but it is hard to use set numbers because lighting changes. In 2021, Zhou et al. used the adaptive HSV channels to account for different lighting situations. [14]. In 2020, Sun proposed using grayscale space to detect white road lines and HSV space to detect yellow road lines. Then, the team reconstructed the image using the Gaussian blur filter and the canny edge detection method to detect the lines. Then, the desired part of the image was cropped for a more detailed examination, and the final process was done on the Region of Interest, abbreviated as ROI [15]. Besides, Modified Canny Edge Detection and

Modified Hough Transform can also be helpful when lighting changes or when the robot works in poor environmental conditions like fog, haze, noise, and dust [16].

### 1.5 Learning-Based Vision

Recent research focused on the application of image-based deep learning to the monitoring of autonomous robots. Farkh et al. [17] came up with a smart way to control a service robot in 2021, using a pair of controllers simultaneously. A convolutional neural network (CNN) was used to track the target and predict its path. Besides, a proportional, integrator, and derivative (PID) controller was employed for speed control. They utilized a Raspberry PI to handle the robot processes based on the image processing. Most robots use acoustic sensors only for barriers that are on the paths. In 2021, Zhao et al. [18] performed a study on how to avoid cavities using an ultrasonic sensor. The study also included visual information as direction guidance signs. Furthermore, using an infrared sensor, the robot avoided falling into pits. This experimental study showed an innovative way for an autonomous mobile robot with multiple sensors to avoid obstacles in an area with multiple paths. Sign detection was performed successfully using a cascade classifier.

### 1.6 The Proposed Method in This Paper

In the upcoming research, the trajectory tracking of a differential-drive mobile robot will be studied using a single camera and path signs. Lane centroid can be found using Image Moments, Python, and the OpenCV library. The defined path for the robot consists of two discrete lanes. Consequently, the robot can change lanes using the Bubble Rebound Technique when facing an obstacle to avoid collision. Additionally, a stop sign was placed at the end of the path, where the robot had to stop moving. Sign detection was also performed using the Haar Cascade Classifier technique. The primary purpose of this study is to study the application of image moments in lane detection, along with masking and feature-based image processing. Additionally, the effect of different sizes of kernel matrix in the Gaussian filter for noise reduction has been compared. In order to validate the proposed code and algorithm, the robot was simulated in ROS, and the same scenario as the laboratory test was designed and implemented in the Gazebo. In section II, the kinematics of the differential-drive robots will be reviewed using the instantaneous center of curve method.

## 2. DDMR KINEMATIC MODEL

Kinematic modeling's main goal for DDMR is to show how the robot's speed depends on the speed of the drive wheels and the robot's shape. In this section, the equations related to the robot's movement in a curve and around a point called the instantaneous center of the curve (Figure 1) will be expressed.

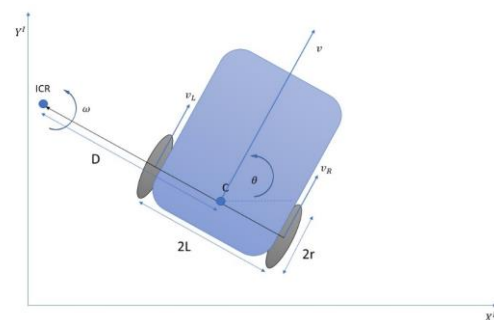


Figure 1, the robot coordinates

Having defined the robot parameters, the linear and angular velocity of each wheel is derived as follows [19-20]:

$$v_R = \omega(D + L), v_L = \omega(D - L) \quad (3)$$

Therefore, the linear speed of DDMR in the robot framework is the average linear velocity of two wheels.

$$v = \frac{v_R + v_L}{2} = r \frac{(\omega_R + \omega_L)}{2} \quad (4)$$

And using relations (5) and (6), the angular velocity of DDMR is equal to

$$\omega = \frac{v_R + v_L}{2D} \quad (5)$$

Consequently, the velocity  $v$  and the angular velocity  $\omega$  are given as input, and the angular velocity of the left and right wheels ( $\phi_R$  and  $\phi_L$ ) are required as control variables. Using the previous relationships, the following equations can be derived:

$$D = L \left( \frac{v_R + v_L}{v_R - v_L} \right) \quad (6)$$

Therefore, the final equation calculates the angular velocity of the wheels.

$$\omega_R = \frac{v + \omega L}{r}, \omega_L = \frac{v - \omega L}{r} \quad (7)$$

### 3. PID POSITION CONTROLLER

The goal of position control is to compute the robot's linear and angular velocity in order to maintain the intended location. Position controls determine the robot's linear and angular velocity utilizing the error between robot states and the desired trajectory.

PID controllers are undoubtedly the most common industrial controllers. Even complicated industrial control systems may have a control network, with a PID control module serving as the primary control component [21]. This simple controller is still implemented in a wide variety of applications. Thai et al. have implemented a trajectory tracking control for differential drive mobile robots by a variable parameter PID controller [22]. In this paper, a PID position controller is proposed to control the motion of the robot using vision systems. The PID controller is used to control the angular velocity of the wheels using the features of the images received from an RGB camera.

The output of a PID controller is calculated using three coefficients of proportional (P), integral (I), and derivatives (D), as shown in Eq. 11 [23].

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (8)$$

Before the response signal, the error should be defined and calculated. Considering image-based visual servoing (IBVS), the center of the picture ( $pixel_{cp}$ ) has been defined in the image processing algorithm, and the center of the lane is defined by its pixel number ( $pixel_{cl}$ ) through the main image processing algorithm. Therefore, these pixel numbers are compared with a threshold as follows:

$$\begin{cases} pixel_{cp} - pixel_{cl} < -5 & \text{turn to left} \\ pixel_{cp} - pixel_{cl} > 5 & \text{turn to right} \\ -5 < pixel_{cp} - pixel_{cl} < 5 & \text{keep straight} \end{cases} \quad (9)$$

Therefore, the error can be defined as:

$$e = pixel_{cp} - pixel_{cl} \quad (10)$$

For instance, the pixel error is illustrated in Figure 2.a, where the robot is changing lanes. The error between the  $pixel_{cp}$  and  $pixel_{cl}$  is calculated simultaneously, which can be monitored through the Linux terminal, as depicted in Figure 2.b. The output returns the value of -104, showing that the robot has to turn left, which is evident in Figure 2.a, where the robot is approaching the obstacle.

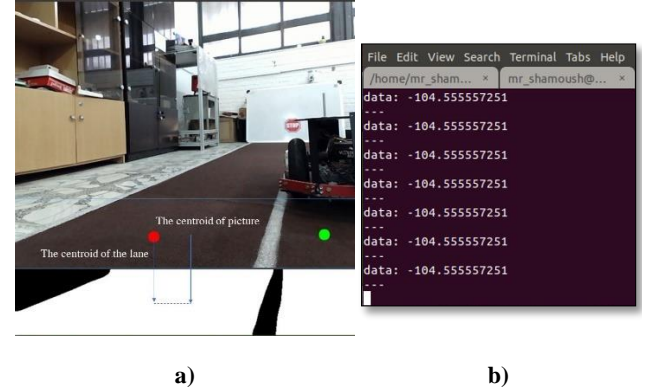


Figure 2, the pixel error after comparison a) schematic b) data number in ROS

Additionally, a PID controller is implemented for the angular velocity of the wheels, where the desired RPM is considered the input signal to the proposed system. The error ( $e$ ) is calculated from the following equation:

$$e = RPM_d - RPM \quad (11)$$

Where  $e$ ,  $RPM_d$ , and  $RPM$  are the error, the desired RPM calculated by the PID controller at the moment, and the current RPM measured by the rotary encoders. This controller is directly implemented using the Arduino Mega2560.

### 4. Haar Cascade Classification

Haar Cascade Classifier is an object recognition program among machine learning algorithms that identifies objects in images and videos. This algorithm is a combination of classification and localization algorithms. Bhaidasna et al. published a paper discussing the pros and cons of object detection using machine learning techniques. Efficiency, flexibility in training, and high performance in simple object shapes have been mentioned as the advantages of the Haar Cascade Classifier method. On the other hand, it demands a large dataset for training while offering low precision in bounding boxes for complex shapes [24].

### 5. LANE FOLLOWING

Automated lane detection or lane change maneuvers help autonomous robots choose lanes and navigate a collision-free path to reach their destination. The process of identifying lanes is to look for specific color lines or features on a path.

#### 5.1 Convert the Received Image to Grayscale

The image received from RGB digital cameras has three channels, R, G, and B, each ranging from 0 to 255, and these numbers ultimately make up color images. Processing three-channel images requires a relatively large amount of processing time. Therefore, converting the initial image to a grayscale mode can be more efficient before starting the processing.

Therefore, white pixels can be easily distinguished from non-white pixels (Figure 3).



a) b)  
Figure 3, a) grayscale image, b) RGB image

## 5.2 Gaussian Blur

The camera images have many noises, making image processing difficult, as shown in Figure 4. So, it would be more efficient to reduce the initial image noise.



Figure 4, the effect of using the raw image for image processing without filtering

Singhal et al. [25] quantitatively analyzed bilateral and Gaussian blur filters on noisy images. According to the analysis, bilateral filters are recognized for their edge preservation properties; however, the above-mentioned studies show that this filter performs less efficiently than the Gaussian blur filter as the noise intensity increases. Desai et al. [26] investigated and compared different filtering algorithms. It has been recommended that noise be minimized while maintaining picture peaks and edges using a median or bilateral filter. The Gaussian filter is adequate as it takes less computational cost if just the image's peaks need to be kept without affecting the edge. Notably, the previously mentioned paper suggested that the optimal kernel size is 7 or less. However, in this paper, the higher kernel sizes have also been analyzed in terms of noise reduction, leading to better image processing in this case where the lighting changes.

## 5.3 Masking

Masking is used in image processing for the ROI. First, the desired image is divided into different parts. This aims to remove unimportant parts of the images, leading to less processing time [27]. In the next step, a Bitwise Operation is used for masking, where the desired line will remain white, and the rest of the image components will turn black (Figure 5).

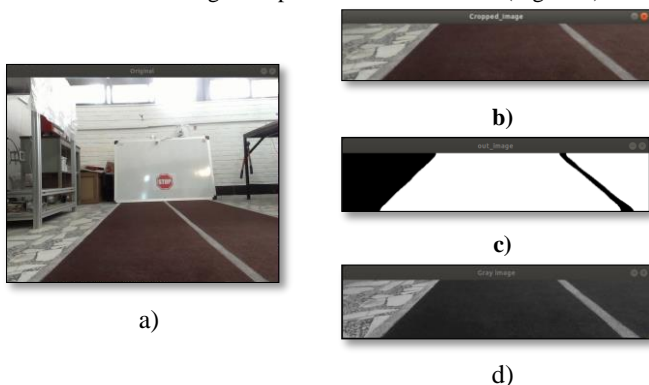


Figure 5, a) raw image b) ROI c) grayscale image d) masked image

## 5.4 Image Moment

Having the line defined by image processing, the center of the line must be determined. Among all image properties, there is a fundamental feature called image moment, which assists in calculating the center of a specific area in an image. An image moment is a particular weighted average of image pixel intensities or a function of such moments that are often used to describe or interpret an image during image processing, computer vision, and related disciplines [28-33]. Image moments may reveal simple picture properties such as area (or total intensity) or center. In a binary image, the image's zeroth order moment, which conveys the number of non-zero pixels, indicates the image's area. Mathematically, image moments ( $M_{ij}$ ) of order ( $i, j$ ) for a black and white image with pixel intensity  $I(x, y)$  are calculated as follows [32]:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x, y) \quad (12)$$

The centroid is simply the position of the arithmetic mean of all points, using the following equation shown in Figure 6.

$$\{\bar{x}, \bar{y}\} = \left\{ \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right\} \quad (13)$$

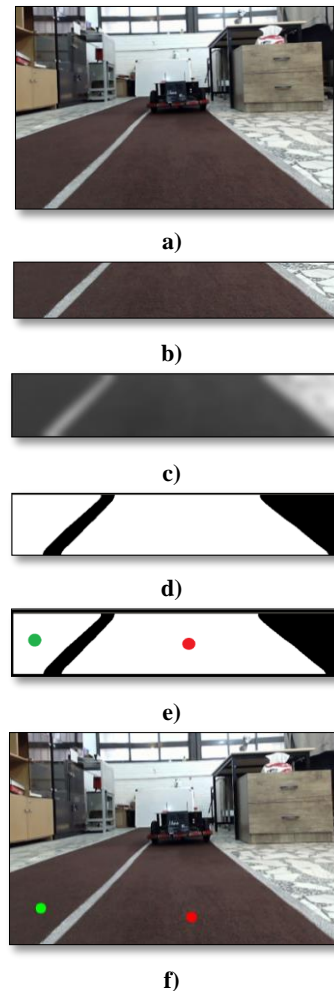


Figure 6, lane detection sequence: a) raw image, b) cropped image to obtain ROI, c) Gaussian filtering, d) Binary image, e) defining centroids using image moments, f) combining all the parts into one image.

## 6. OBSTACLE AVOIDANCE

Mobile robots are becoming more common in automated industrial places. Collision-free path planning is required in any of these situations where many mobile robots conduct duties simultaneously. Obstacle avoidance methods are divided into general categories: classical or traditional and heuristic. The Bubble Rebound method is one of the famous and classic obstacle avoidance methods. This method defines a bubble around the robot containing maximum free space in any direction. The shape and size of the bubble are a function of the robot's geometry model and sensor information (Figure 7).



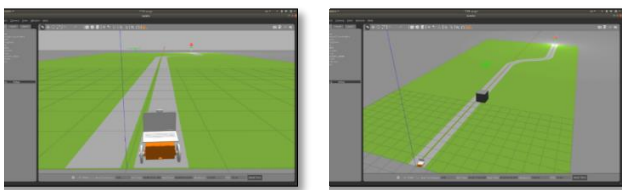
Figure 7, the Bubble Band method

Susnea et al. [34] proposed a basic algorithm for obstacle avoidance using a bubble around the robot. However, the bubble size and shape are dynamically adjusted and will depend on the robot's kinematics. The advantages of the proposed algorithm can be mentioned as better performance in narrow paths, noise compensation using a set of sensors, and avoiding not only static obstacles but also some dynamic obstacles. Additionally, this method requires microcontrollers with low cost and low power consumption. However, some disadvantages and weaknesses, like the possibility of failure, not being smooth, and losing the desired path, have to be considered. It would be beneficial to note that the range of the desired bubble is determined by the ultrasonic sensors around the robot, which in this research is considered to be 1.5 meters when moving along a straight line. This obstacle avoidance method has been analyzed, mentioning the pros and cons of each method. Regarding the investigation by Bhavesh et al. [1], it can be concluded that this method is more efficient in terms of computational cost, compatibility with different sensors, and passing narrow corridors.

## 7. ROS

### 7.1 Gazebo

Gazebo can be used for the world simulation, illustrated in Figure 8.



a) b)

Figure 8, the simulation environment a) without obstacle  
b) with obstacle

### 7.2 ROS Core

The ROS core has the ability to communicate with numerous nodes at the same time in a simulation. Therefore, each node is responsible for a specific task. In this study, four different nodes were considered to perform the tasks simultaneously, as shown in Figure 9.

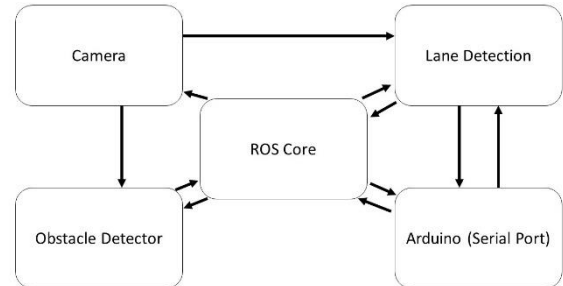


Figure 9, ROS core connections

### 7.3 Simulation of the Scenario in Gazebo

In the first scenario, the robot moves straight, following a lane where no obstacle is on the way (Figure 10).

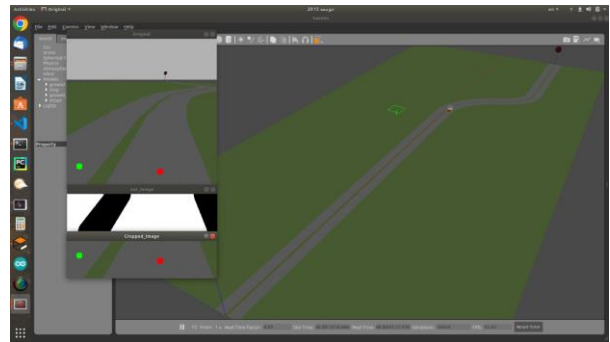


Figure 10, the simulation of the robot while moving on a curved path with no obstacles

Figure 11 illustrates the angular velocities of the wheels during the first scenario.

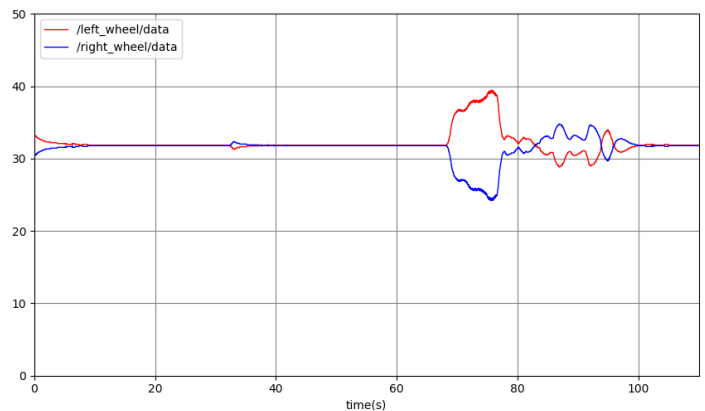
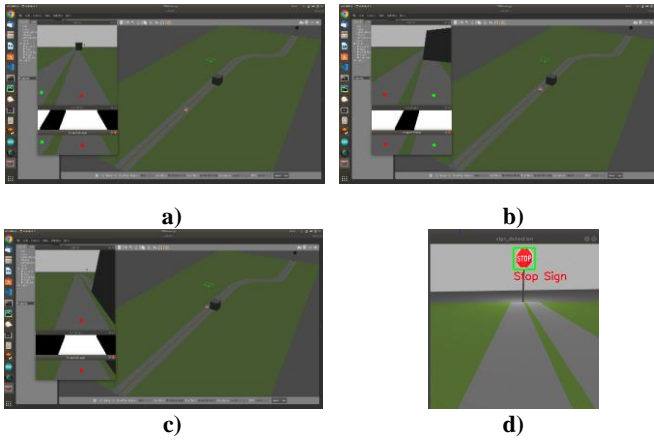


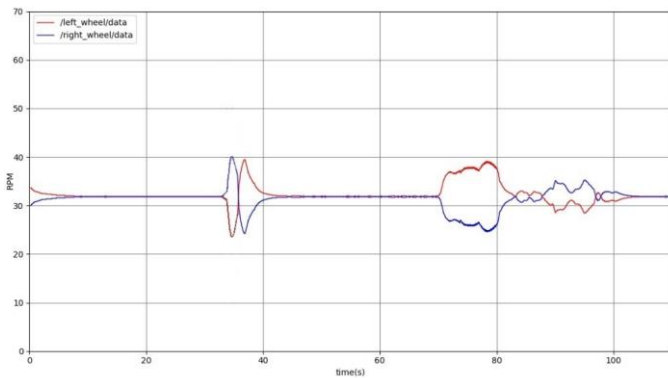
Figure 11, the angular velocity of the robot's wheels in the simulation

In the second scenario, an obstacle is placed on the path near the curve. Therefore, the robot performs the obstacle avoidance algorithm to avoid collision with the black box, as shown in Figure 12.



**Figure 12, the simulation of the robot while a) moving straight, b) changing the lane, c) passing the obstacle, d) facing a stop sign at the end of the path**

Considering all the maneuvers, including straight movement, obstacle avoidance, and turning to right and left, the angular velocities of the wheels are shown in Figure 13.



**Figure 13, the angular velocity of the robot's wheels in the simulation**

## 8. ENVIRONMENTAL TESTS

### 8.1 Sign Detection

Previously, the classifier was evaluated to detect the signs in simulation. In this section, the classifier model will be evaluated in different laboratory test conditions. Figure 14 shows the correct detections in different camera modes, which shows the high accuracy of this trained model.



**Figure. 14, stop sign detection in environmental tests.**

### 8.2 The Kernel Matrix Effect in Filtering

Using the OpenCV library and cv2.GaussianBlur command in the Python code, the received image is reconstructed, and its noise is reduced. By increasing the size of the kernel matrix, the noise in the edges of the image decreases. Also, the transparency of the image is reduced, making image processing and color recognition more accurate. Table 1 shows the images received by the camera when changing the direction of movement. As seen in the following tables, the reconstructed

images with the Gaussian filter still have a little noise by increasing the size of the kernel matrix to 37. However, by adding the value to 41, all the noises vanished. Therefore, the image can be used to detect moments easily. Also, Tables 2 and 3 show the effect of increasing this parameter in two different conditions when changing lanes and passing the obstacle.

**Table 1. The effect of Kernel size on the edges of the lane**

Kernel size	Gaussian filter	Binary
17		
27		
31		
37		
41		

**Table 2. The effect of Kernel size on the midline**

Kernel size	Gaussian filter	Binary
17		
27		
31		
37		
41		

**Table 3. The effect of Kernel size when passing obstacle**

Kernel size	Gaussian filter	Binary
17		
27		
31		
37		
41		

Also, Table 3 shows the effect of the kernel matrix size parameter on removing image noises when crossing the obstacle. In this situation, by choosing the size of the kernel matrix up to 31, image noises are still visible. However, the noises disappear by increasing this value, and a more suitable image will be available.

### 8.3 Laboratory Test Preparation

Figure 15 illustrates the mobile robot that was used in this test. Also, the test environment is a straight path 6 meters long and 1.10 meters wide. This path has two separate lines with a width of 0.5 meters. An obstacle has been placed at a distance of about 3 meters from the starting point of the movement (Figure 16).

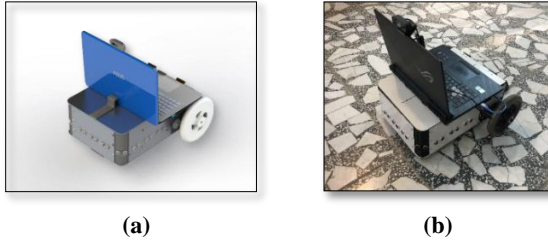


Figure 15, the robot's main body (a) in SOLIDWORKS (b) in the laboratory



Figure 16, the robot's path in the laboratory

### 8.4 The First Scenario

In this part of the scenario, the robot will move on a smooth line without obstacles until the stop sign is detected. The robot is supposed to stop at a distance of 50 cm from the stop sign. Figure 17 shows the image received by the robot at the starting point in a straight path.

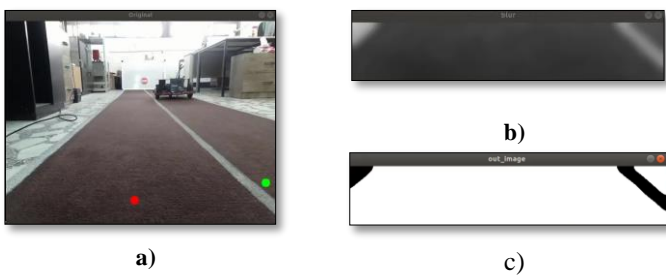
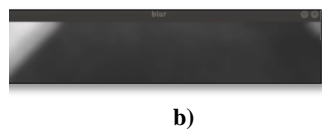


Figure 17, a) the captured image at the start point, b) Gaussian filtering, c) masked image

Figure 18 shows the robot while passing near the obstacle. The robot detects and follows the left lane. The right lane is not detected due to the obstacle. By the way, when no obstacle is detected by the front sensors, there would be no lane changing due to the defined algorithm. This means that the robot is robust to some noises that make the robot change and follow the wrong lane.



b)



Figure 18, a) the captured image when passing the obstacle, b) Gaussian filtering, c) masked image

At the end of the path, the robot detects the stop sign, which makes the robot fully stop as soon as the distance between the sign and the robot reaches 0.5 meters (Figure 19). Also, the angular velocities of the wheels are illustrated in Figure 20.



Figure 19, a) the captured image near the stop sign, b) the detected sign

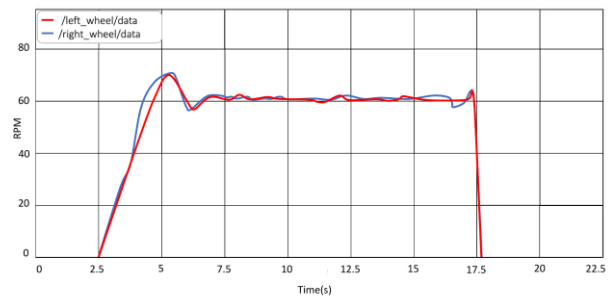


Figure 20, the angular velocity of the robot's wheels

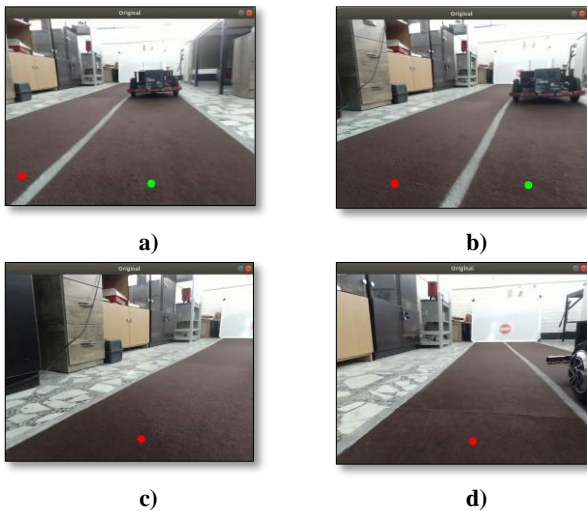
### 8.5 The Second Scenario

In this part of the scenario, the robot moves in a straight line at a constant speed and changes its path when an obstacle is detected. It stops moving at the end of the road when seeing the stop sign. In this scenario, the robot starts moving from the initial point shown in Figure 21.



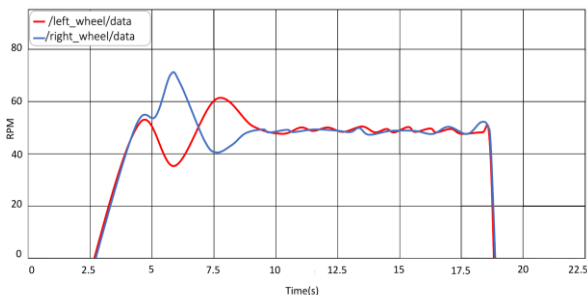
Figure 21, the captured image at the start point

Then, upon reaching a distance of 1.5 meters from the obstacle, the lane change command is sent to ROS, and the robot chooses a new lane to avoid collision with the obstacle (Figure 22).



**Figure 22, the captured image a) when obstacle detected, b) when changing lanes, c) when rotating in the new lane, d) when fully changed lane**

Eventually, the robot continues its path, like in the previous scenario, and stops completely by detecting the stop sign at a distance of 0.5 meters. The angular velocities of the wheels are illustrated in Figure 23.



**Figure 23, the angular velocity of the robot's wheels**

## 9. CONCLUSION AND ANALYSIS

In this study, the main objective of the robot is to navigate a two-lane path from the starting point to the endpoint while addressing potential obstacles and adhering to the stop sign. This study emphasizes the use and evaluation of image moments in the line detection process. Based on the laboratory test results, image moments demonstrate potential as a valuable tool for path planning and recognizing predefined shapes, aiding the robot's navigation toward its goal in indoor applications across various industries. This study demonstrated the feasibility of using image moments in vision-based robot control or visual servoing (VS) for indoor applications. However, conducting laboratory tests proved challenging without applying image filters. To address this, the impact of Gaussian blur was evaluated in various scenarios. The tests concluded that the Gaussian filter is effective in reducing noise during image processing, particularly in environments with varying light intensity. It has been observed that increasing the kernel matrix size beyond 7 enhances image processing, particularly around the edges of shapes. Although a kernel size of 7 or smaller is typically optimal, deliberately using a larger size has proven beneficial in dynamic lighting conditions,

significantly reducing noise. This improvement enhances the robot's perception by minimizing the impact of illumination changes that could otherwise introduce noise along the edges of the image. The tests revealed some challenges and errors in specific scenarios. Notably, minor fluctuations in the robot's movement were observed when following a straight line, which could be addressed using robust controllers. While the PID controller is energy-efficient for real-time applications, it may still introduce some errors.

For example, Figure 21 illustrates the angular velocity of the wheels in a straight-line scenario, where a slight fluctuation occurs as the wheels reach the set point of 60 RPM. Similarly, Figure 23 shows the angular velocity from the start to the end of the path, with a set point of 50 RPM. These fluctuations, coupled with image processing errors, become more pronounced when the ambient light intensity changes sharply. Considering the performance of the Gaussian blur filter when the kernel size increases, image processing errors may also cause problems when the ambient light intensity changes sharply. This might be reduced using an adaptive grayscale range. According to the observations and laboratory test results, different thresholds of RGB channels may need to compensate for different light conditions. To mitigate these issues, adopting an adaptive grayscale range is recommended. Adjusting RGB channel thresholds could compensate for varying light conditions. A light intensity sensor could also be utilized to dynamically adjust the pixel intensity threshold received by the camera. This approach would enhance the compatibility of the image processing system with ambient light variations, enabling broader applications of the robot across industries and environments. Implementing these adjustments could be a valuable focus for future research to advance this study.

## 10. REFERENCES

- [1] Bhavesh, V. A. (2015). Comparison of various obstacle avoidance algorithms. *Int. J. Eng. Res. Technol*, 4, 629-632.
- [2] Guo, L., & Sun, C. (2021, September). Research on Mobile Robot Vision Navigation Algorithm. In *Journal of Physics: Conference Series* (Vol. 2010, No. 1).
- [3] Manivannan, P. V., & Ramakanth, P. (2018). Vision based intelligent vehicle steering control using single camera for automated highway system. *Procedia computer science*, 133, 839-846.
- [4] Abid, S., Hayat, B., Shafique, S., Ali, Z., Ahmed, B., Riaz, F., ... & Kim, K. I. (2021). A Robust QR and Computer Vision-Based Sensorless Steering Angle Control, Localization, and Motion Planning of Self-Driving Vehicles. *IEEE Access*, 9, 151766-151774
- [5] Zhang, X., & Zhu, X. (2019). Autonomous path tracking control of intelligent electric vehicles based on lane detection and optimal preview method. *Expert Systems with Applications*, 121, 38-48.
- [6] Chen, W., Wang, W., Wang, K., Li, Z., Li, H., & Liu, S. (2020). Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation: A review. *Journal of traffic and transportation engineering (in English)*, 7(6), 748-774.
- [7] Sanderson, A. C., & Weiss, L. E. (1983). Adaptive visual servo control of robots. *Robot vision*, 107-116.
- [8] Siradjuddin, I., Siradjuddin, I. A., & Adhisuwignjo, S. (2015). An image based visual control law for a



- differential drive mobile robot. *International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS*, 15(6).
- [9] Agravante, D. J., Claudio, G., Spindler, F., & Chaumette, F. (2016). Visual servoing in an optimization framework for the whole-body control of humanoid robots. *IEEE Robotics and Automation Letters*, 2(2), 608-615.
- [10] Griffin, B., Florence, V., & Corso, J. (2020). Video object segmentation-based visual servo control and object depth estimation on a mobile robot. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1647-1657).
- [11] Parosi, R., Risiglione, M., Caldwell, D. G., Semini, C., & Barasuol, V. (2023, October). Kinematically-decoupled impedance control for fast object visual servoing and grasping on quadruped manipulators. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 1-8). IEEE.
- [12] Chaumette, F. (2007). Potential problems of stability and convergence in image-based and position-based visual servoing. In *The confluence of vision and control* (pp. 66-78). London: Springer London.
- [13] Corke, P. I., & Hutchinson, S. A. (2001). A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4), 507-515.
- [14] Zhou, Y., Wang, Z., & Wang, J. (2021). Real-time adaptive threshold adjustment for lane detection application under different lighting conditions using model-free control. *IFAC-PapersOnLine*, 54(20), 147-152.
- [15] Sun, Z. (2020, August). Vision based lane detection for self-driving car. In *2020 IEEE International conference on advances in electrical engineering and computer applications (AEECA)* (pp. 635-638). IEEE.
- [16] Kaur, G., & Kumar, D. (2015). Lane detection techniques: A *International Journal of Computer Applications*, 112(10).
- [17] Farkh, R., Quasim, M. T., Al Jaloud, K., Alhuwaimel, S., & Siddiqui, S. T. (2021). Computer vision-control-based cnn-pid for mobile robot. *CMC-Computers Materials & Continua*, 68(1), 1065-1079.
- [18] Zhao, J., Fang, J., Wang, S., Wang, K., Liu, C., & Han, T. (2021). Obstacle avoidance of multi-sensor intelligent robot based on road sign detection. *Sensors*, 21(20), 6777.
- [19] Auzan, M., Hujja, R.M., Fuadin, M. R., & Lelono, D. (2021). Path Tracking and Position Control of Non-holonomic Differential Drive Wheeled Mobile Robot. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 7(3), 368-379.
- [20] Shijin, C. S., & Udayakumar, K. (2017, March). Speed control of wheeled mobile robots using PID with dynamic and kinematic modelling. In *2017 international conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)* (pp. 1-7). IEEE.
- [21] Crowe, J., Chen, G. R., Ferdous, R., Greenwood, D. R., Grimbale, M. J., Huang, H. P., ... & Zhang, Y. (2005). *PID control: new identification and design methods*. Springer-Verlag London Limited.
- [22] Thai, N. H., Ly, T. T. K., Thien, H., & Dzung, L. Q. (2022). Trajectory tracking control for differential-drive mobile robot by a variable parameter PID controller. *International Journal of Mechanical Engineering and Robotics Research*, 11(8), 614-621.
- [23] Borase, R. P., Maghade, D. K., Sondkar, S. Y., & Pawar, S. N. (2021). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*, 9, 818-827.
- [24] Bhaidasna, M. H., & Bhaidasna, M. Z. (2023). Object Detection Using Machine Learning: A Comprehensive.
- [25] Singhal, P., Verma, A., & Garg, A. (2017, Jan.). A study in finding effectiveness of Gaussian blur filter over bilateral filter in natural scenes for graph based image segmentation. In *2017 4th international conference on advanced computing and communication systems (ICACCS)*. IEEE.
- [26] Bhisham, K. (2020). Image Filtering-Techniques Algorithm and Applications. *GIS science journal*, 7(22).
- [27] Yan, X., & Li, Y. (2017, October). A method of lane edge detection based on Canny algorithm. In *2017 Chinese Automation Congress (CAC)* (pp. 2120-2124). IEEE.
- [28] M.K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. Inform. Theory*, vol. 8, pp. 179–187, 1962.
- [29] R.Mukundan and K.R.Ramakrishnan, *Moment Functions in Image Analysis: Theory and Application*, Singapore: World Scientific, 1998.
- [30] R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," in *Proc. Computer Vision, Graphics, Image Processing Conf.*, vol. 54, Sept. 1992, pp. 438–460
- [31] Chaumette, F. (2004). Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4), 713-723.
- [32] Senthilnathan, A., Acar, P., & De Graef, M. (2021). Markov random field based microstructure reconstruction using the principal image moments. *Materials Characterization*, 178.
- [33] Mebarki, R., Krupa, A., & Chaumette, F. Image moments-based ultrasound visual servoing. *2008 IEEE International Conference on Robotics and Automation* (pp. 113-119).
- [34] Susnea, I., Filipescu, A., Vasiliu, G., Coman, G., & Radaschin, A. The bubble rebound obstacle avoidance algorithm for mobile robots. In *IEEE ICCA 2010*.