International Journal of Instrumentation Science and Engineering.
ISSN 2278-5825 Volume 14, Number 1 (2024) pp.1-8

http://www.ripublication.com

Coordinate Following Robot using the ArUco Marker

Pratyush Pallav

Department of Mechanical Engineering
Birla Institute of Technology,
Mesra, Ranchi, Jharkhand, India.

Abstract

The coordinate-following robot has derived its name from the input process, which uses the coordinate as the comparison factor for its closed-loop control system. An ArUco marker (unique identity given to the robot) has specific properties and usability, making it a primary camera influence. Several characteristics of the robot's positioning relative to the end state can be precisely defined using the ArUco marker. The interface is programmed to get information from the mouse about the desired final stage of the robot. A virtual enables the user to choose the desired halt point for the robot. The robot with an ArUco marker on its top tells the user its current centre of mass's coordinates. Thus, the robot minimizes the difference between its current and desired coordinates. This technology can be widely used in minor unmanned surgeries and space explorations to develop a Morse code for a robot, which the robot and the developer only understand. It also has a wide application with drones and military robots.

Keywords Arduino, ArUco marker, Coordinate Following, Computer Vision, Serial Communication**.**

INTRODUCTION

A coordinate-following robot is a rudimentary robot that takes input from the user with a mouse click. It is an interactive and precise way to manoeuvre the robot. The optimized PID (Proportional-Integral-Derivative) [1]-[4] control is the most common way to have precise movements. Coordinate-following robots have a wide application in the ariel and wheeled robotics. The same navigation can be performed using different modules and sensors like GPS (Global Positioning System), edge detection, proximity sensors, and Lidar mapping. However, these methods are more complex and less human-interactive, which can cause multiple disabilities for the new technology user.

ArUco markers [5]-[6] can seamlessly integrate with Python through dedicated libraries and APIs, facilitating efficient collaboration and interaction. ArUco markers contain unique properties, making them an essential tool to communicate information to the robot through OpenCV (Open Computer Vision Library). GPS modules have a more extensive range but are occasionally inefficient in detecting the exact halt point. Edge detection and contouring the environment have been an orthodox but efficient method to locate the halt point. However, there have been instances where some external changes were required to generate the specific stimuli in the robot. ArUco markers are easy to handle and don't require any external power, sensors, or electronics to produce accurate results.

The ArUco markers are easily recognizable by using OpenCV. A camera at the top of the experimental field detects the ArUco marker. It compares the ArUco ID with the desired ID number, and on getting the positive feedback, it calculates its centre point and returns its coordinates. Then, the user subsequently provides the halting coordinate by clicking in the generated OpenCV frame. Therefore, the system has two parameters: the start and end coordinate. A virtual line connecting these two points appears on the screen. This line is the robot's estimated path. Two PIDs work to align the robot to a certain angle, and another PID traverses the robot in a straight line. Thus, it first computes the required rotation angle independently, and continuous closed-loop feedback, which diminishes the error, runs simultaneously.

In the previous method, the GPS module [7], integrated with Arduino and a motor drive control device, was commonly used. It has the advantage of a broader range and is sometimes easier to automate. However, the current industrial and academic applications need precision in minute details. Signal interference and environmental factors often create disturbances in the communication of the GPS module. The ArUco marker doesn't consume any power, which makes it an even better choice for regular usage. Different ArUco markers can be programmed and coded for a specific task. For example, an ArUco with ID - 63 can be programmed to actuate the robotic arm at specific angles when encountered by the camera.

Similarly, different orientations can be programmed using different ArUco IDs. Thus, ArUco stimulation using IDs can be a better option than natural language processing, where minimal human interaction exists. The whole frame can be scaled up or down according to the camera's vertical position from the ground, which helps determine the robot's distance, velocity, acceleration, jerk, and other kinematic properties. An example of an ArUco Marker with ArUco ID – 0 is illustrated in Fig 1.

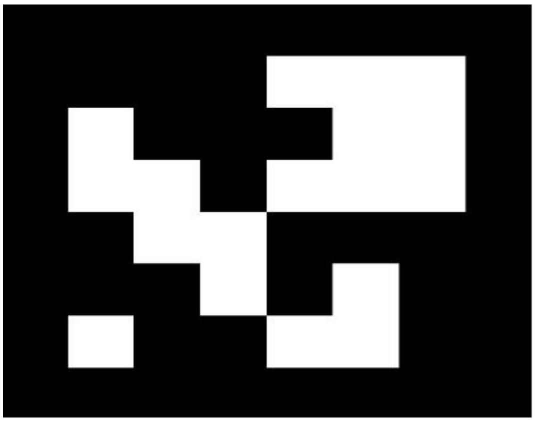

Figure 1. A 6×6 ArUco Marker with ID - 0

The robot's control system is PID-tuned and follows the feedback system, making it a closed-loop [8] control system. The working principle of the robot is illustrated using the following flowchart in Fig 2.

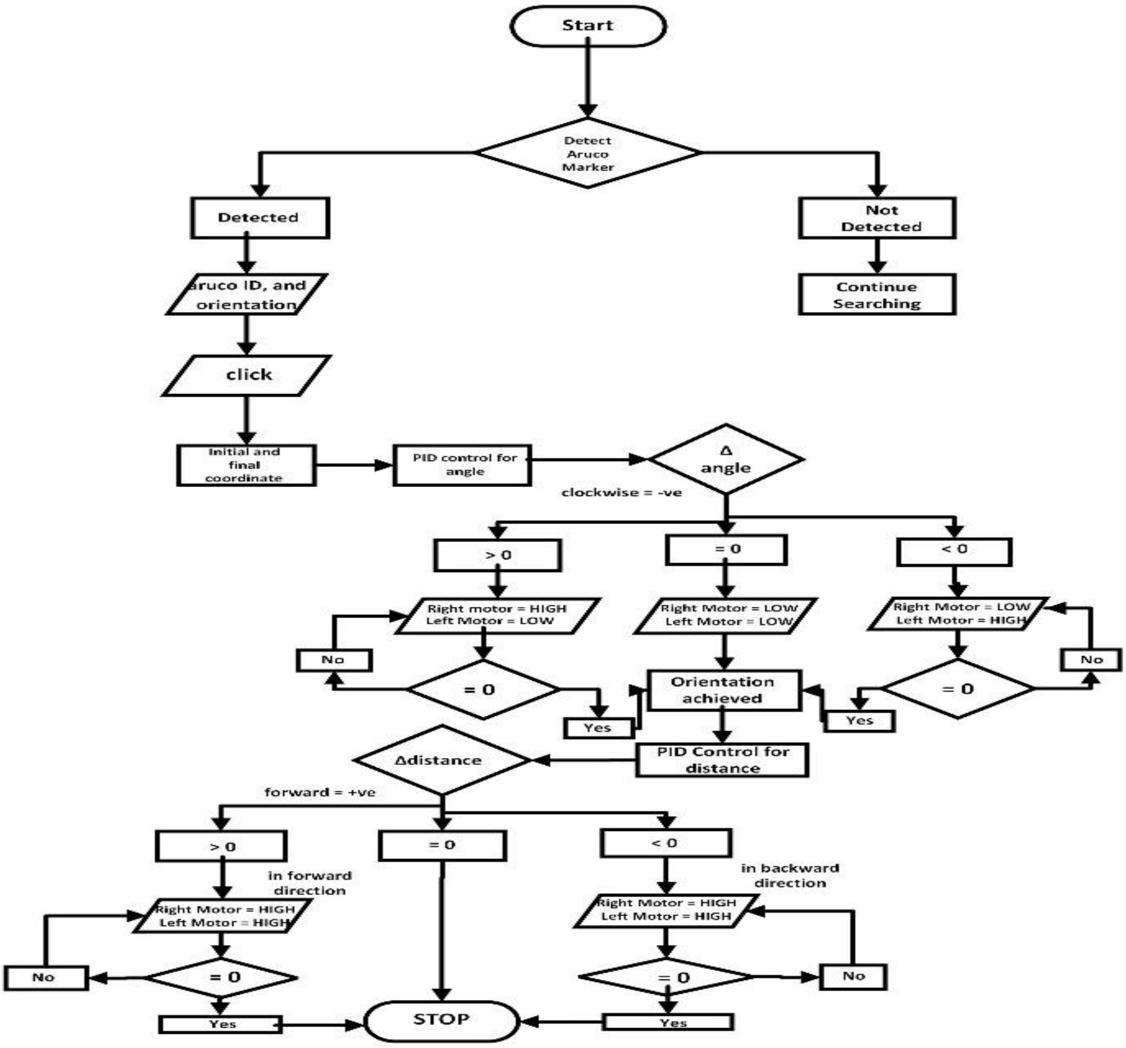

Figure 2. The flowchart represents the control of the coordinate following robot.

ARUCO DETECTION AND FEATURE EXTRACTION

ArUco markers are non-identical, two-dimensional patterns with unique and exclusive properties embedded in them. There are different specifications of markers like that if a square matrix, *i.e.*, 4×4, 5×5, 6×6 and 7×7 is some of the readily available options of an ArUco marker. Therefore, there are an equal number of rows and columns in an ArUco marker. However, an actual ArUco marker consists of n+2 (where n is the number of rows or columns). Fig 1 illustrates a 6×6 ArUco marker, but we can observe that there are 8 columns and 8 rows, respectively, making it an 8×8 matrix. However, it is a mere property of an ArUco marker that the first and last columns and rows of the ArUco marker are always filled with black. Thus, it is essential to note that the first and last columns and rows act as a boundary to the ArUco marker.

Some helpful information is embedded in the ArUco marker, which can be used and manipulated using mathematical tools to make it advantageous in our application. The primary advantage of using an ArUco marker is its easy handling and predefined modules in the OpenCV module. The unique identity of a robot or a set of robots helps maintain the individuality of different robots in complex situations. Fig 3 demonstrates the different parameters detected by the system using an ArUco marker.

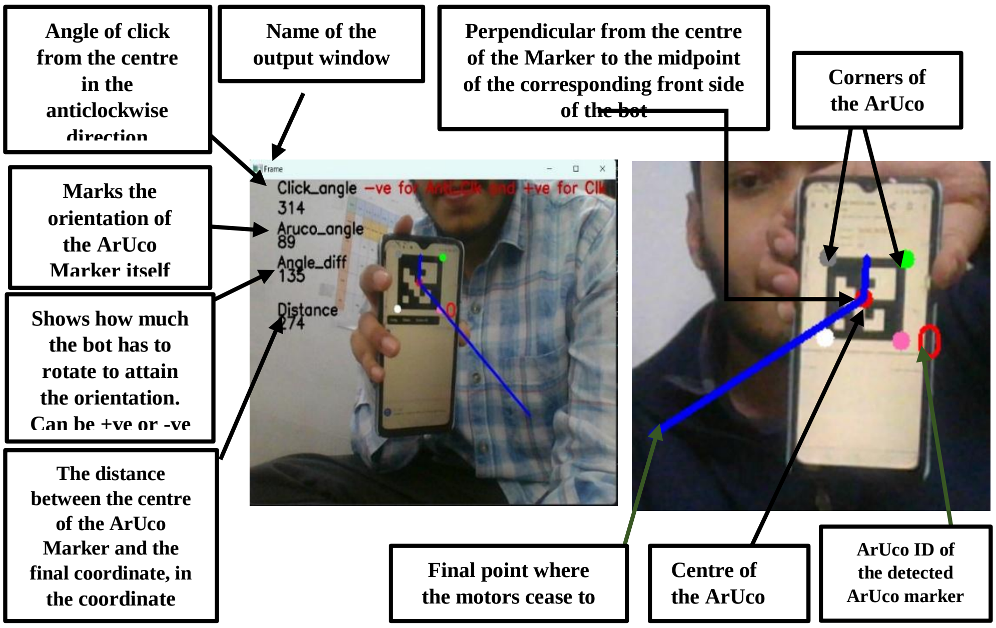

Figure 3. Output frame of the ArUco detection and feature extraction.

A left mouse click on a desired coordinate is a stimulus to set the "final parameters" for the robot. A virtual line connecting the centre of the ArUco marker to the desired coordinate is established. This feature of the robot can also be labelled as the predefined or desired path of the robot. The robot first aligns with the path, using a PID control for the angular traversing. A perpendicular line that connects the centre of the ArUco marker to the midpoint of the corresponding front side of the ArUco marker compares the angular difference between the robot's orientation.

Further, when both lines coincide, they possess the equivalent equations of the straight line. This condition signals the initiation of the subsequent process by traversing the robot on the exact desired path or the virtual line joining the two vital points. A PID control is used to have precise traversing of the robot, with effective stoppage before the endpoint is achieved.

CONTROLS OF THE ROBOT

The robot uses serial communication between the Arduino and the system running the image processing on it. It can be replaced with other modules like HC-06 (Bluetooth module) or ESP 8266 (Node MCU). The motor performance is enhanced by using PID controls. The image processing is done using the OpenCV module in Python, and then the data is transferred to the serial monitor of the Arduino IDE. This communication uses an open-source library in Python – "PySerial". The mathematical background of the PID control is demonstrated by the following block diagram (Fig 4) [9].

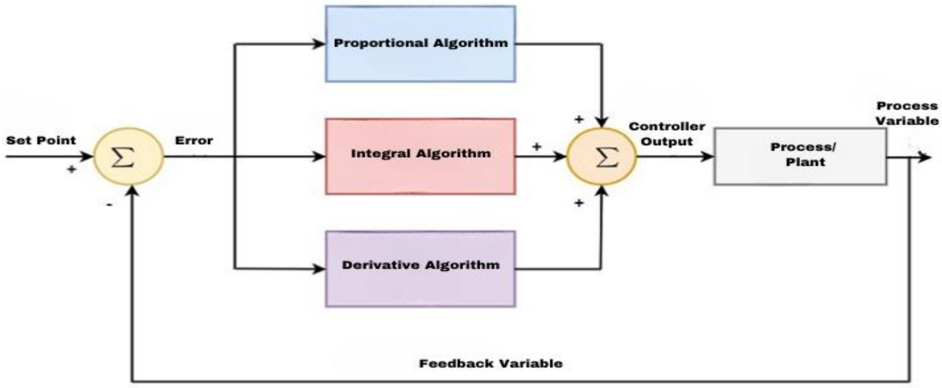

Figure 4. Block diagram of process control using PID. [9]

The control system's P, I, and D parameters were tuned using the classical method [9]. In this method, the deadtime to the time constant ratio is assumed to be between 0.1 and 1. It has also been observed that the approximate 1.5 gain margins are attained with Ziegler-Nicholas [10]-[11], and optimized tuning formulas for calculating several parameters like Integral Absolute Error (IAE), Integral Square Error (ISE) and Integral Time Absolute Error (ITAE) are used.

Pulse Width Modulation [12], with the help of an L298N motor driver, controls the rotational speed of the motors according to the track. The following table 1 demonstrates the general commands for a single motor using combinations of different IN (input) and EN (enable) pins.

Table 1.demonstrating the motor control using L298N.

ENA	INA	INB	Output
0	×	×	Motor is off
1	0	0	Stop/Break
1	0	1	Forward
1	1	0	Backwards
1	1	1	Stop/Break

RESULTS AND DISCUSSION

The coordinate-following robot is a fully tuned robot that traverses accurately. The robot uses extensive image processing to get the mathematical requirements in synchronization with the control system. The following flow chart (Fig 5) demonstrates the step-by-step working of the robot.

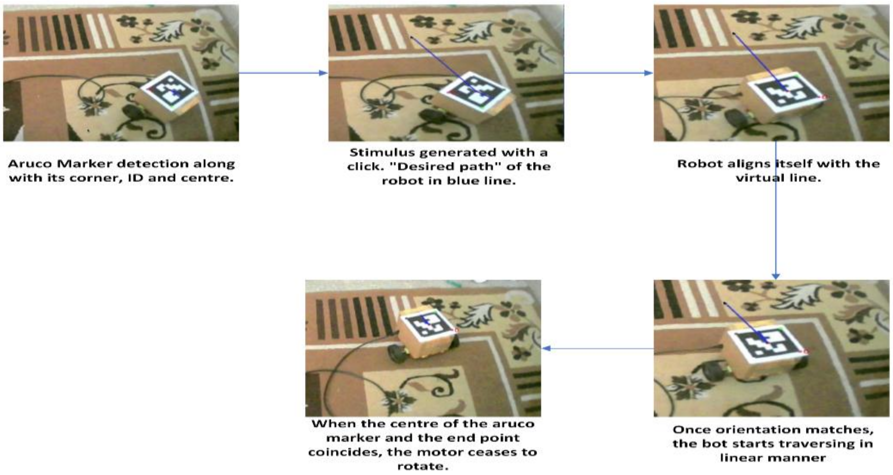

Figure 5. Working process of the coordinate following robot

Initially, the robot is placed randomly in the observable environment of the camera, with an ArUco marker placed on its top. The camera detects and extracts features like ID, corners and, most importantly, the centre from the ArUco marker. Two virtual lines joining the centre and midpoint of the front side of the ArUco marker, and the second line joining the centre and the final point of the path is created. The user's left mouse determines the final point or the final coordinate click. Further, the angular difference between these two lines is calculated automatically, and the motors correspond to it according to the nature of the value (negative or positive). Hence, when the two lines coincide, the angular orientation of the robot is confirmed to be perfect. After that, the robot starts traversing on the line until the centre of the ArUco marker coincides with the centre of the final point or the final coordinate, as shown in Fig 5. The acting parameters have also been demonstrated on screen to create a Graphic User Interface (GUI) for the whole system and the robot, as shown in Fig 3.

Different parameters, including the angles and distances, are determined using the feature extraction from the ArUco Marker. The results obtained are altogether physical, where the Coordinate Following robot traverses by checking the angle and the distance simultaneously. This feature of the Coordinate Following robot can list it as one of the most accurately traversing robots when its potential is enhanced with the PID control.

Some similar robots are doing the same task by extracting other crucial features of the camera frame. The basic structure and algorithm of such robots are almost congruent to one another. One such robot, which is similarly applied, is a line follower robot using OpenCV [13]. They are also equipped with similar sensors, microcontrollers, motor drivers and motors; only the stimulus to act differs in some kind. The Coordinate Following robot can be compared to a certain extent with the line follower, colour detection robot [14] or other OpenCV-infused robot, inhibiting some environmental stimulus through a visual sensor. However, the novelty of the robot is that it gets affected easily by the environment. The ArUco Markers are uncommon in the real world and can be individually programmed to the advantage of the user. For example, a line follower robot can easily be distracted by detecting a false line in the environment. A colour-detecting robot can easily be faulted by detecting a similar colour from some unwanted object or area.

Therefore, the individual ArUco Markers can be programmed with different values to the motor. For example, a Coordinate Following robot can be programmed to detect an ArUco Marker with a specific ID, indicating a steep turn ahead. It would cause the motors to stop rotating. Further, the Coordinate Following robot doesn't depend on the predefined path but makes a virtual path, indicating its path of traversing in advance. Therefore, it can be concluded that it has some notable advantages over other robots of similar domains.

CONCLUSION

The Coordinate Following bot can be used to enhance the user experience with lesser environmental constraints. Moreover, has several applications in society, like aerial delivery, precise communication between robots by manipulating and programming different ArUco IDs, navigation without GPS module, automation, warehouse management and surveying over a large landmass are some of them. The houses of a locality can be given a unique ArUco ID for the precise delivery of goods and food in the area. The robots doing a repetitive task can be labelled with an ArUco Marker with a specific ArUco ID to initiate and do the task of detecting the required ArUco Marker. A factory can label different storage units with different ArUco IDs to complete a specific task. Further, it enhances the security of a system which currently seems an important factor of assessment due to certain unreliable threats induced in IoT.

ACKNOWLEDGEMENTS

Author sincerely acknowledges the facilities for experimental work provided in “Robolution Club” of the Institute and motivation received from Dr. Rakesh Kumar Sinha, Professor, Department of Bioengineering and Biotechnology at Birla Institute of Technology, Mesra, Ranchi (India) in preparation of this manuscript.

REFERENCES

[1] Li, Y., Ang, K.H. and Chong, G.C., 2006. PID control system analysis and design. *IEEE Control Systems Magazine*, *26*(1), pp.32-41.

[2] Johnson, M.A. and Moradi, M.H., 2005. *PID control* (pp. 47-107). London, UK: Springer-Verlag London Limited.

[3] Hägglund, T. and Åström, K.J., 1985. Method and an apparatus in tuning a PID-regulator. U.S. Patent No. 4,549,123. 22 Oct. 1985.

[4] Astrom, K.J., 1995. PID controllers: theory, design, and tuning. *The International Society of Measurement and Control, Research Triangle Park.*

[5] Kam, H.C., Yu, Y.K. and Wong, K.H., 2018, June. An improvement on aruco marker for pose tracking using kalman filter. In *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* Busan, Korea (South), (pp. 65-69). IEEE.

[6] Sani, M.F. and Karimian, G., 2017, November. Automatic navigation and landing of

an indoor AR. drone quadrotor using ArUco marker and inertial sensors. In *2017 international conference on computer and drone applications (IConDA)* Kuching, Malaysia, (pp. 102-107). IEEE.

[7] Panetta, M.P., Abbrescia, M., Avanzini, C., Ferroli, R.B., Baldini, L., Batignani, G., Battaglieri, M., Boi, S., Bossini, E., Carnesecchi, F. and Chiavassa, A., 2019. The new trigger/GPS module for the EEE project. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *936*, pp.376-377.

[8] Ploplys, N.J., Kawka, P.A. and Alleyne, A.G., 2004. Closed-loop control over wireless networks. *IEEE Control Systems Magazine*, *24*(3), pp.58-71.

[9] Borase, R.P., Maghade, D.K., Sondkar, S.Y. and Pawar, S.N., 2021. A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*, *9*, pp.818-827.

[10] Patel, V.V., 2020. Ziegler-Nichols Tuning Method: Understanding the PID Controller. *Resonance*, *25*(10), pp.1385-1397.

[11] Kumar, V. and Patra, A., 2016. Application of Ziegler-Nichols method for tuning of PID controller. *International Journal of Electrical and Electronics Engineering*, *8*(2), pp.559-570.

[12] Sun, J., 2012. Pulse-width modulation in dynamics and control of switched electronic systems. *Advanced perspectives for modeling, simulation and control of power converters*. London: Springer London, 2012. 25-61.

[13] Gomes, M.V., Bassora, L.A., Morandin, O. and Vivaldini, K.C.T., 2016, November. PID control applied on a line-follower AGV using a RGB camera. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 194-198). IEEE.

[14] Shukor, A.Z., Natrah, N.A., Tarmizi, A.I., Afiq, A.A., Jamaluddin, M.H., Abd Ghani, Z., Shah, H.N.M. and Ab Rashid, M.Z., 2018. Object tracking and following robot using color-based vision recognition for library environment. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *10*(2-7), pp.79-83.