

Development of an Omnidirectional Mobile Robot Using Embedded Color Vision System for Ball Following

Omar Yaseen Ismael^{a*}, John Hedley^b

^a*Rasty District, Erbil 44001, Iraq*

^b*Newcastle University, Address, Newcastle Upon Tyne NE1 7RU, UK*

^a*Email: omar_mechatronics@yahoo.com*

^b*Email: john.hedley@ncl.ac.com*

Abstract

In this paper, three Omni-wheels mobile robot is developed by using a low cost embedded color vision system (CMUcam5 Pixy) for tracking and following a ball. The ball detection is done by distinguish its colour from the environment. A suitable design, of the robot is offered considering a number of practical factors. CMUcam5 Pixy gives the robot the ability to detect many of multi coloured objects at the same time. The robot motion is controlled due to the kinematics analysis and the path planning. The velocity PI algorithm is used to control the robot wheels' speeds. The robot sensors (i.e. CMUCAM 5 camera, and encoder) data are collected and processed then, transmitted to a laptop by the wireless communication. A Graphical user interface (GUI) is designed by using Matlab program in this project. This GUI is displaying the transmitted robot data such as the ball distance and the robot velocity in order to support studying the robot performance. The robot shows a positive performance after the demonstration. The designed robot can be used later for developing a self-located small sized RoboCup competition.

Keywords: Omni-wheels; CMUcam5 Pixy; velocity PI algorithm; GUI; Matlab; RoboCup.

1. Introduction

In many mobile robots' applications, the robot is required to have the ability to interact with the environment along with adapting itself to the external situation [1]. Predicting and estimating of the moving objects motion are the basis for the interaction tasks.

* Corresponding author.

For instance, it is very essential to detect and predict the ball's position when a robot plays the soccer [2]. One way of tracking the ball is by detecting its colour. This can be done by using a proper vision system. The flexibility of vision sensors makes it so beneficial. An image sensor can detect or sense practically anything by applying the right algorithm. Despite of that, there are two drawbacks with vision sensors. Firstly, a large amount of data (usually in Mb/s) is obtained outputted from the image sensors. Secondly, this data size needs a powerful processor. Even if the processor can carry on with this data, it would not be having the enough processing power for other tasks. CMUcam5 Pixy is a low cost, fast vision sensor. It can be connected directly to Arduino and other controllers. A quick teaching can be performed in this sensor to detect objects. CMUcam5Pixy addresses vision sensors problems by combining the image sensor with a powerful processor. After processing the images by CMUcam5 Pixy, the only useful information will be sent to the microcontroller. It does this very fast (frame rate 50 Hz). An interface, such as UART serial, SPI, I2C, digital out, or analog out can be used to transmit this information. So a plenty of processing power will be available in Arduino or other microcontroller when it connects to the CMUcam5 Pixy. It can detect hundreds of objects with multi colours at the same time [3]. Figure 1 is illustrated the CMUcam5 Pixy.



Figure 1: CMUcam5 Pixy.

The aim of the research is to developing of an omnidirectional mobile robot using low cost embedded color vision system for ball tracking so, this robot can be used later in developing a self-located small sized RoboCup competition. The factors to be taken into consideration, for the design and operation of the robot, are to be based on practical requirements and limitations as well as a comprehensive theoretical analysis.

2. Methodology

This section starts by developing theoretical analysis for the object and image distances then it shows the kinematics analysis model for the omnidirectional mobile robot, which is then followed by the robot Omni-wheels velocity control and the path planning. After that, the Ball following algorithm is presented in flow chart. Finally, the robot Graphical User interface (GUI) design is illustrated.

2.1. Object and Image distances

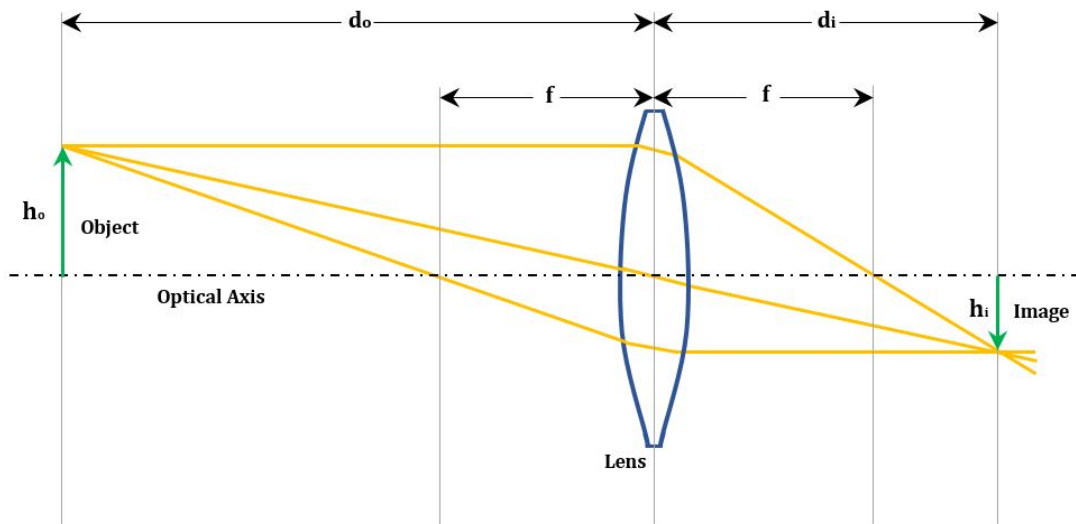


Figure 2: Lens analysis that shows the object and image relationship

Light can pass through vacuum or clear materials, but through a vacuum its speed will be faster. The direction of a light ray changes when it strikes the interface surface between two media at a non-orthogonal angle. This bending of ray of light is named as refraction. A lens is an optical device (a piece of glass) which refracts and transmits light, diverging or converging the beam. A section of a differently-sized spherical surface on one side and a section of spherical surface on the other are shaping the lens. These sections cause the light to spread out either after light beams converge to an area or immediately from the glass. Can be defined from the different sizes of the spheres. CMUcam5 Pixy uses M12 lens type. Figure 2 illustrates lens analysis [4]:

Where:

d_o : is the distance from the centre of the lens to the object (measured along the axis).

d_i : is the distance from the centre of the lens to the image (measured along the axis).

f : is the focal length of the lens.

h_o : is the object height.

h_i : is the image height.

The quantitative relationship between the image distance (d_i), the object distance (d_o), and the focal length (f) can be expressed by the lens equation. The equation is stated as:

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i} \tag{1}$$

The magnification equation relates the ratio of the image height and object height to ratio of image distance and object distance. The magnification equation is stated as:

$$M = \frac{h_i}{h_o} = - \frac{d_i}{d_o} \tag{2}$$

Equation (1) and equation (2) can be combined to give information about the object distance if the image distance, object height, and focal length are known.

Equation (2) can be rewritten as:

$$d_i = \frac{h_i \times d_o}{h_o} \tag{3}$$

Sub equation (3) into equation (1):

$$\frac{1}{f} = \frac{1}{d_o} + \frac{h_o}{h_i \times d_o}$$

Rearrange this to give the object distance as:

$$d_o = (f \times h_o) \cdot \frac{1}{d_i} + f \tag{4}$$

2.2. Robot Kinematics

The designed robot has three Omni-wheels. The angular velocity of the wheels ($\dot{\phi}_i$) can be defined from the following equation [5,6]:

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\theta) & \cos(\theta) & R \\ -\sin(\theta + \alpha_2) & \cos(\theta + \alpha_2) & R \\ -\sin(\theta + \alpha_3) & \cos(\theta + \alpha_3) & R \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & 0 \\ 0 & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_L \\ \dot{y}_L \\ \dot{\theta} \end{bmatrix} \tag{5}$$

Where: $\alpha_1 = 0^\circ$, $\alpha_2 = 120^\circ$ and $\alpha_3 = 240^\circ$. (\dot{x} , \dot{y} , $\dot{\theta}$) are the global velocity of the robot. θ is the robot orientation with respect to the global coordinates. R is the distance (along a radial path) from the wheels to the centre of gravity of the robot. The value of R is 0.13 m. r is an omnidirectional wheel radius: 0.051 m.

2.3. Omni- wheels velocity control

In velocity control, the control action is calculated based on the difference between the reference velocities

$(\dot{\phi}_{ref,1}, \dot{\phi}_{ref,2}, \dot{\phi}_{ref,3})$ and the actual angular velocities $(\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3)$. This can be expressed in a mathematical sense as [7,8]:

$$e_i = \dot{\phi}_{ref,i} - \dot{\phi}_i \tag{6}$$

Here e_i is the tracking error [rad/sec] and the subscript i ($= 1, 2, 3$) indicates which wheel is considered.

Figure 3 shows the block diagram of the closed loop motor speed control system that is used in this project to control the Omni-wheels velocity. To reach the required velocity, the delivered power to the motor is controlled by the closed loop controller. Adjusting the supplied electrical power to the motor is usually done by Pulse Width Modulation (PWM) technique.

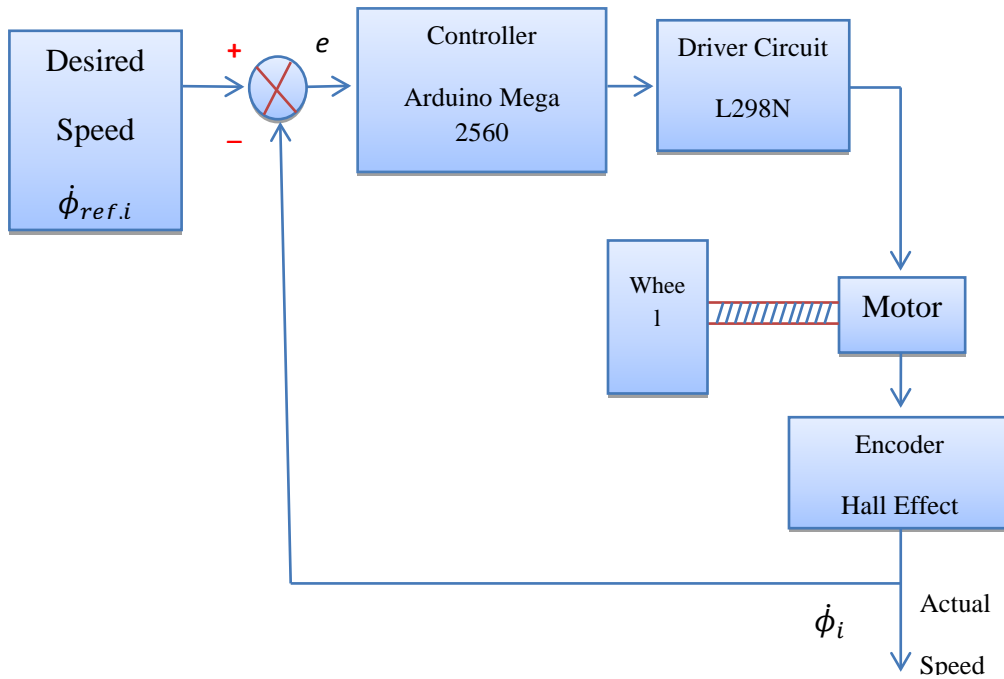


Figure 3: Block diagram of the closed loop motor speed control system

In Figure 3, the actual motor speed is measured by the encoder (Hall Effect sensor) then it entered to the Arduino board to be compared with desired motor speed. A suitable controlling Algorithm (in this project Proportional- Integral (PI) speed algorithm) is written inside the Arduino microcontroller. The Arduino controls the driver circuit (L298N H-bridge board) to give the motor the required power level.

2.4. Robot path planning for following a ball

Path planning is an important primitive for the robot that lets the robot find the optimal or shortest path between the robot and the ball. Optimal paths could be paths that minimize the braking amount, the amount of turning, or

whatever following the ball application requires. The algorithm that is use to perform the path planning is photography illustrated as shown in figure 4.

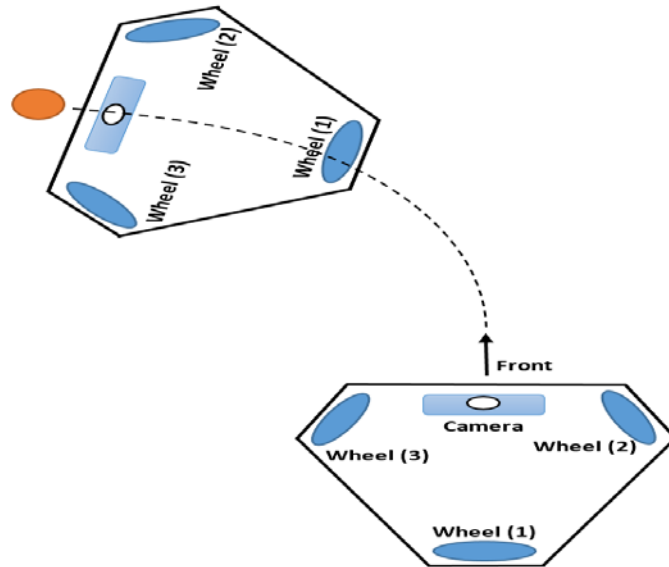


Figure 4: Robot path planning

As seen from figure 4, the robot path planning will be a combination of translation and rotation movement to generate a curve motion. At the end of the motion the robot will be facing the ball.

2.5. Ball following algorithm

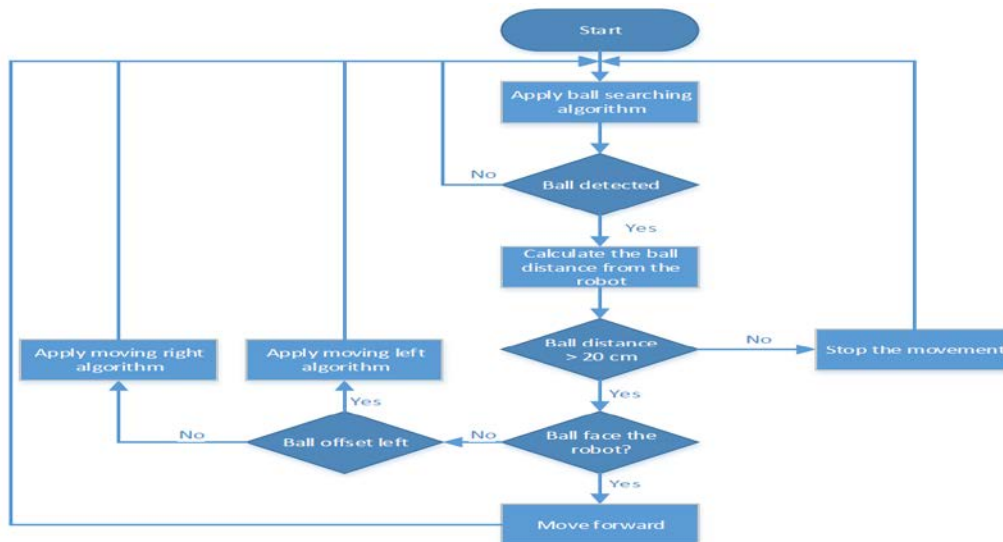


Figure 5: Flow chart for ball following algorithm

The fundamental structure of the programming including the operations and decisions to be followed by the robot autonomously when detect and follow the ball is illustrated as an algorithmic flowchart in figure 5.

2.6. Robot Graphical User interface (GUI) design

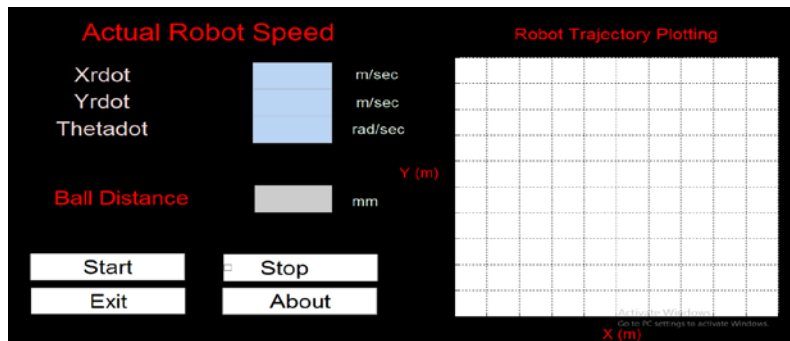


Figure 6: Robot GUI

In this project, a GUI has been designed by using Matlab program [9]. This GUI is displaying some of robot data in order to support studying the robot performance. The designed GUI is shown in figure 6.

By pressing the ‘start’ button that is shown in figure 6, the program starts acquiring the data from the robot and then displaying the robot speed, ball distance, and the robot trajectory. The ‘stop’ button is used to halt the program execution and the ‘exit’ button to quit from the GUI window.

3. Results and Discussion

This section discusses all the calibrations, theoretical and practical tests, and evaluations, performed on the omnidirectional robot, along with the practical and theoretical results obtained after applying the analyses and methodology that were studied and deduced in section 2.

3.1. Adjusting the CMUcam5 Pixy Parameters

A robustness teaching for the The CMUcam5 Pixy Parameters are adjusted by using Pixymon software to detect the ball in the desired operating environment. The final selected parameters are shown in table 1.

Table 1: CMUcam5 Pixy final selected parameters

Parameter	value
Max blocks per frame	1
Max blocks per colour	1
Minimum saturation	10
Hue spread	1
Saturation spread	1
Data out port (I2C)	1
Brightness of camera	120
Min block area in pixels	15

The result of what the robot see in the working environment after adjusting the CMUcam5 Pixy Parameters is illustrated in figure 7.

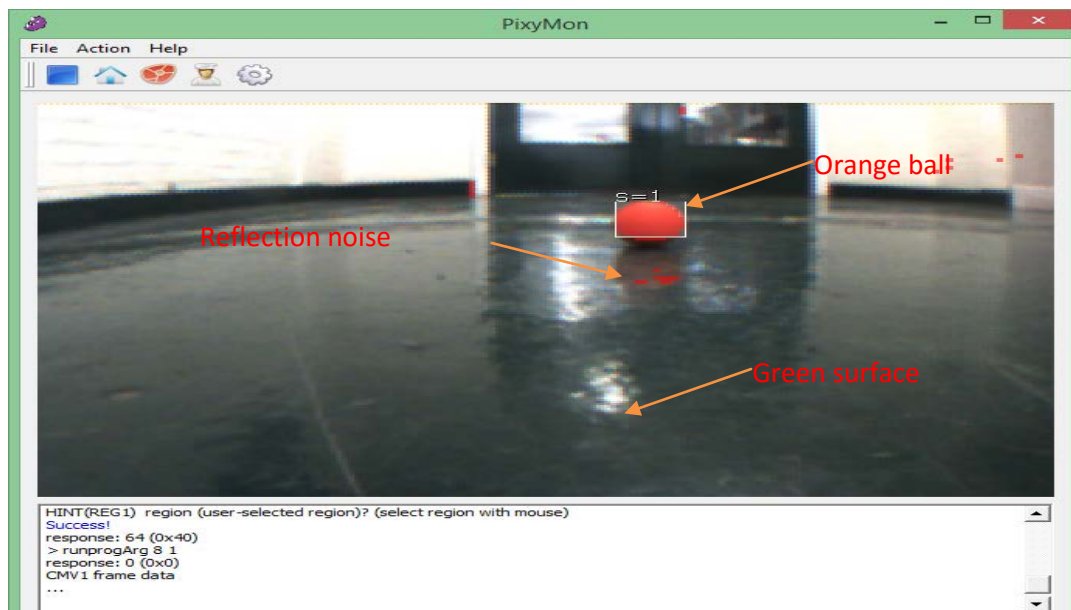


Figure 7: Robot camera vision

From figure 7 its concluded that to detect the ball is required to avoid any colour interference. Although the colour filtering algorithm within the CMUcam5 Pixy is positive in the sense that it can process an entire frame very fast, discrimination can sometimes be a problem. Generally, two types of errors may occur, the first one when CMUcam5 Pixy considers the ball matches a particular colour, but it is not the correct object but an object with a similar hue. The second error type is when CMUcam5 Pixy be unsuccessful to detect the ball that is required to detect with a particular colour. These errors can be eliminated by adjusting several parameters within the CMUcam5 Pixy such as the Brightness, the Min saturation and the Hue spread.

3.2. Ball distance calibration

CMUcam5 Pixy does not give a distance measurement between the image sensor and the detecting ball. This distance can be calculated with respect to other information (i.e. ball height from image sensor). As a result of that the distance between the spotting ball and the robot platform will be known. Figure 8 shows the distance calibration that has been done between the ball (in this case an golf size orange ball) and the vision sensor which placed on the robot base.

In figure 8, a measure tape is used to physically obtain the actual distance between the image sensor and the ball. In this test, it is required to always see the full object, also the ball is needed to face the image sensor lens (x=319 pixels, y=199 pixels) in order to get correct readings. The data from the vision sensor are collected by the Arduino Mega 2560 microcontroller board and then sent to the computer to be displayed. At the beginning, the orange ball is placed at zero distance from the robot body. Then it moved with 5cm steps while recording the

actual distance and the image sensor readings each time. It is noticed that after 180cm distance the image sensor readings are started to give the same value due to this ball size. From the CMUcam5 Pixy object height readings, the calibration line has been plotted as shown in figure 9.

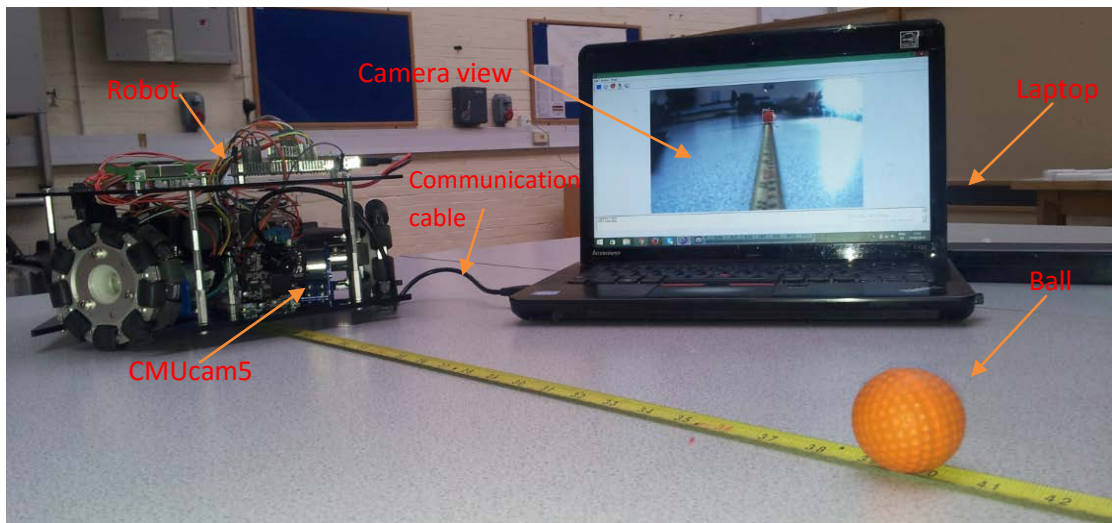


Figure 8: Distance calibration between the ball and the robot

From figure 9 the distance calibration equation is:

$$d_o = \frac{8700}{d_i} + 35$$

By making comparison with equation (4) in section (2):

$$f \times h_o = 8700$$

The ball actual size (h_o) = 41mm then the above equation can be modified to be for any ball as:

$$d_o = \frac{212.2 \times h_o}{h_i} + 35 \quad (5.1)$$

Note that for large ball size the measuring distance range will be larger.

3.3. Robot 'following the ball' Demonstration

Figure 10 demonstrates the robot 'following the ball' application. Figure 10A, shows the initial position for the robot and the ball. When the robot turns on, it starts to apply the ball following algorithm that discussed in section (2). While the robot is detecting the ball, the path planning algorithm is applied. Figures (10 B, C and D) show the robot movement until reaches to the ball. At the end of the motion the robot will be facing the ball. The robot performance in terms of following the desired curved motion was evaluated. In addition, the velocity, acceleration and deceleration of the robot were tuned to get the best performance in this task. At the end the

robot had a positive kinematics performance and the path that robot followed was very close to the desired one. A snapshot for the robot GUI is shown in figure 11.

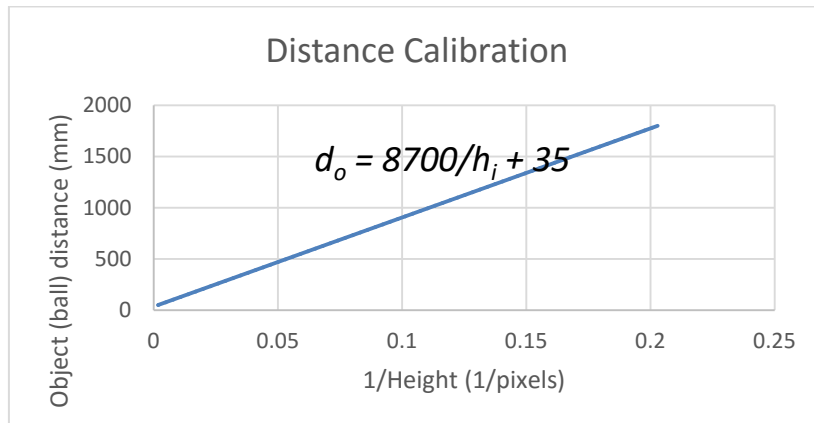


Figure 9: Object (ball) distance calibration curve.

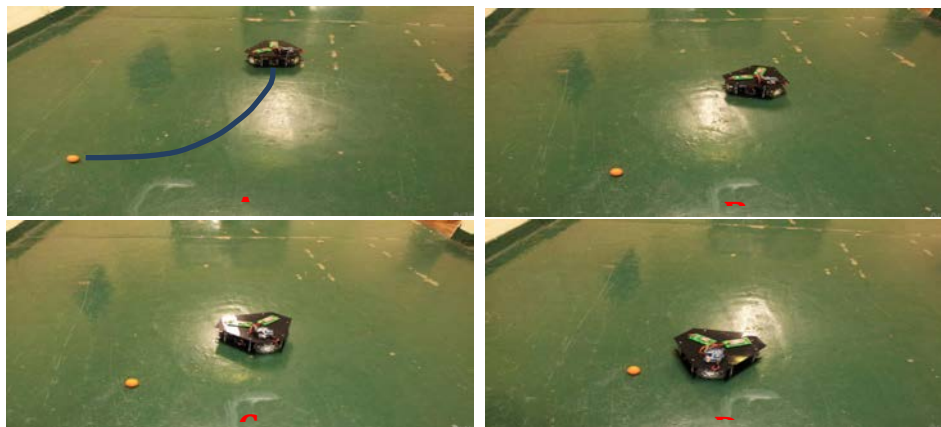


Figure 10: Demonstration of the robot 'following the ball' application.

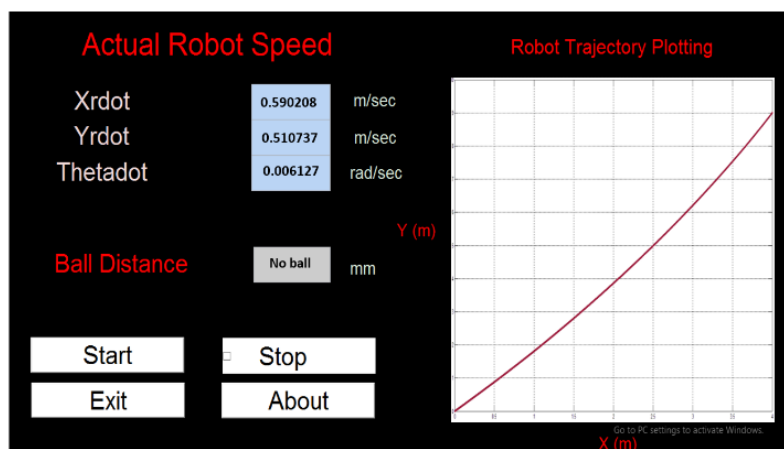


Figure 11: Snapshot for the robot GUI

4. Conclusions

The aim of the research was to developing of an omnidirectional mobile robot using low cost embedded color vision system (CMUcam5 Pixy) for ball tracking and following. A detailed evaluation of using the vision system with in the three wheels' omnidirectional robot has been discussed. The robot was having the ability to detect many of multi coloured objects at the same time by using the (CMUcam5 Pixy). Kinematics analyses and path planning were evaluated and then successfully applied on the robot. The robot wheels' velocity was closed looped control by using the velocity PI algorithm and showed a positive response while following the ball. The demonstration for the robot following the ball has been successfully done by using a golf size orange ball and a green field to be close to the RoboCup competition standards. The ball detection was done by distinguish its colour from the environment. The robot sensors (i.e. CMUCAM 5 camera, and encoder) data were collected and processed then, transmitted through wireless communication to the Laptop in order to graphically illustrated. This GUI was designed by using Matlab program and was continuously displaying the ball distance from the robot, the instantaneous velocity of the robot and the robot trajectory while following the ball. This has supported in studying the robot performance. In conclusion, the primary aims and objectives of this research has been successfully accomplished, since the robot ball tracking and following application was effectively developed and all the analyses, once applied on the actual prototype, showed expected results. However, in order to improve the efficiency of the existing design and due to the aspiration of the scope of this research, further adjustments may be performed in future.

5. Future Work

It would be possible for this robot to be tested in Small league RoboCup competitions after doing few modifications to the robot body such as adding a kicking mechanism and extra sensors. With a more sophisticated path planning algorithm and analyses, it would be possible for this robot to follow complex trajectories. This will support the robot to follow the ball with a smoother path. It is also possible to Build a team of robots and perform a communication between them.

References

- [1] X. Li and A. ZELL. H_{∞} "Filtering for a mobile robot tracking a free rolling ball," In: Robot Soccer World Cup, Springer Berlin Heidelberg, 2006, pp. 296-303.
- [2] E. Torres and A. Weitzenfeld, "RoboCup small-size league: Using neural networks to learn color segmentation during visual processing," in Robotic Symposium, 2008. LARS'08. IEEE Latin American, 2008, pp. 14-19.
- [3] Scott. (2014, 20 August). CMUcam5 Pixy. Available: <http://www.cmucam.org/>.
- [4] B. Chen, R. Kantowski, and X. Dai, "A Simple Gravitational Lens Model For Cosmic Voids," arXiv preprint arXiv:1310.7574, 2013.

- [5] T. A. Baede, "Motion control of an omnidirectional mobile robot," Traineeship report DCT 2006, 2006.
- [6] MA, S., REN, C. & YE, C. An omnidirectional mobile robot: Concept and analysis. *Robotics and Biomimetics (ROBIO)*, 2012 IEEE International Conference on, 2012. IEEE, 920-925.
- [7] K. J. Åström and B. Wittenmark, *Adaptive control*: Courier Dover Publications, 2013.
- [8] G. A. Hayter, M. B. Taub, D. M. Bernstein, and M. K. Sloan, "Closed Loop Control System with Safety Parameters and Methods," ed: Google Patents, 2013.
- [9] A. S. Andreatos and A. Zagorianos, "Matlab GUI application for teaching control systems, " In: *Proceedings of the 6th WSEAS International Conference on Engineering Education*, 2009, pp. 208-211.