American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Analysis, Design, and Implementation of an Omnidirectional Mobile Robot Platform

Omar Yaseen Ismael[a]*, John Hedley[b]

[a]*Rasty Distric, 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, an omnidirectional mobile robot is designed, analysed, evaluated and developed. A suitable design, of the robot is offered considering a number of practical factors. The methodology and operation of its omnidirectional locomotion is analysed by deducing a range of applicable equations for the robot kinematics, and trajectory plotting. The velocity PI algorithm is used to control the robot wheels' speeds. Several tests were applied to the robot platform such as repeatability and manoeuvrability tests. Reasonable results were shown after applying the analyses via computer programming. The designed omnidirectional robot platform is able to move instantaneously in any direction at any angle, without or with changing the orientation of the robot.

Keywords: Omnidirectional; Kinematics; Trajectory Plotting; Velocity PI Algorithm; Manoeuvrability.

1. Introduction

In recent years, using mobile robots in technical, industrial, and educational applications has been increased. The freely movement of an autonomous mobile robot within the work space from one location to another becomes necessary [1]. The mobile robot wheel design is a very important factor that gives the mobile robot the ability of moving freely in tight areas as well as avoiding obstacles. This design specifies the degree of mobility and manoeuvrability of the mobile robot. The omnidirectional term means that the capability of moving instantaneously from any configuration in any direction [2].

---

* Corresponding author.

Omnidirectional (holonomic) platforms compared with the conventional (non-holonomic) vehicles, have many advantages such as moving sideways and follow complex trajectories [3,4]. This kind of mobile robots is desirable for applications in narrow environments and has wide application prospect. There are two kinds for platforms with omnidirectional wheels: special wheels' kind and conventional wheels' kind. Special wheels have a passive moving direction and an active tracking direction and they are mostly studied for the omnidirectional platforms. Previous researchers have developed and proposed omnidirectional robot platforms using a wide variety wheels' kinds including Mecanum, roller,

and spherical wheels. The omnidirectional mobile robots show their positive performance in competitions such as the RoboCup and Search and Rescue international competition [5]. A positive trajectory planning and trajectory following that is necessary in this type of competitions, can be achieved by using omnidirectional platforms.

The aim of this study is to analysis, design, and implementation of an omnidirectional robot platform as the base for development of mobile robot projects such as RoboCup Competitions or Search and Rescue. 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. The robot performance can be tested by applying different algorithms. The study objectives can be summarised as:

- Mechanical design and manufacturing of the omnidirectional robot platform.
- Design and select electronics circuits that driving the robot, collecting the sensors (i.e. encoder) data and process them, and performing a communication with a laptop to plot.
- Proper selection of robot materials to achieve the following Criteria: speed, manoeuvrability, 1/cost, and aesthetics.
- Control the robot motion due to the kinematics analysis and the path planning.
- Test the robot performance (i.e. Speed, acceleration, and repeatability).

2. Design and Specifications

By taking into account all the limitations involved in the recent developments, the design of 'omnidirectional robot' has to consider a range of factors to eliminate most (or all) of the drawbacks and limitations. For that, a detailed evaluation is required to select the optimal design of three Omni-wheels mobile robot.

2.1. Robot platforms design

The design is based on three omnidirectional wheels with 120 degrees between each other. Several criteria need to be considered in the design of the robot platforms:

- Robot main platform dimensions.
- Mounting of other parts (e.g. Spacers, motors, electronics circuits, battery sensors and voltage regulator).

- Response (minimize mass and inertia), and stability.
- Positive look and minimizing the cost.

The robot designed by using Autodesk Inventor software and the robot is based on three platforms. The 3d models for the platforms were created as shown in figure 1. The thickness of the upper and mid platforms is 3mm, but the lower platform has a thickness of 5mm because it is needed to handle a higher loading. The robot platform has made from Acrylic sheet.

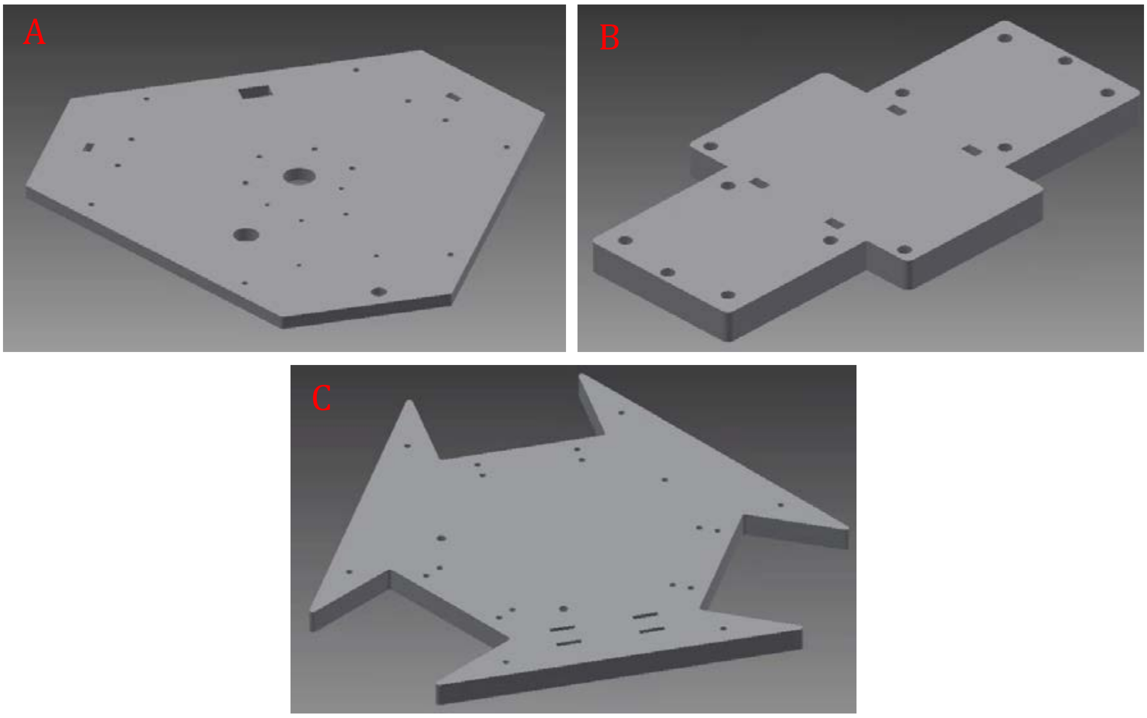

Figure 1: Robot platform (A) Upper Robot platform (B) Mid Robot platform (C) Base Robot platform.

2.2 Robot Controller selection

It has been decided to select the Arduino microcontroller to be the controller for the omnidirectional mobile for the following reasons [6]:

- Inexpensive.
- Clear, simple programming environment.
- Extensible and open source software and hardware.
- Cross – platform.

Arduino is based on a microcontroller board that has a physical computing open source platform. It can take the inputs from a variety of sensors and controls many types of actuators. Arduino can work independently (stand-alone) to control the projects or to communicate with software in the computer. In the project the Arduino is doing the following:

- Perform PI- speed algorithm to control the three motors speed, therefore, acquires the required omnidirectional wheels speed.
- Apply the robot kinematics analysis to get the omnidirectional movement.
- Collect the data from the Hall Effect sensor then apply the suitable algorithms to make use of this data.
- Perform path planning algorithm for the robot.
- Transmit the data from the robot to the Laptop in order to study it.

The final assembled model, including all the mechanical and electrical components is illustrated in figure 2.

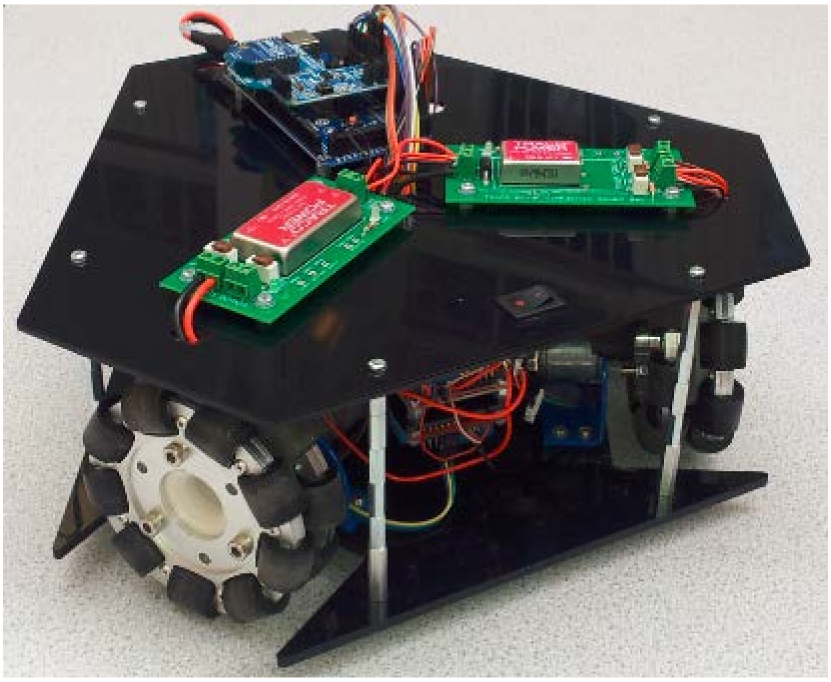

Figure 2: Final assembled model with all electrical and mechanical component.

3. Methodology

This section starts by developing a kinematics analysis model for the omnidirectional mobile robot, which is then followed by the robot trajectory plotting. After that, the Omni-wheel speed control methodology (Discrete PI Algorithm) is described in details. Finally, the robot performance measurement methodology is defined.

3.1 Kinematics Analysis

To have a precise control over the omnidirectional robot movement in the environment, developing a kinematics model for the robot is needed. Figure 3 illustrates the kinematics diagram of the robot. Before presenting the equations, it is required to define terminologies such as the local frame and the global frame. The local frame [x_l, y_l] is the frame attached to the robot body. The centre of gravity of the robot is coincided with the centre of this local frame. The three Omni-wheels are placed relative to the local frame at an angle α_i (i = 1, 2, 3). If count degrees in the clockwise direction as positive and the local axis x_L is taken as starting point then, $\alpha_1 = 0°$, $\alpha_2 =$

120° and $\alpha_3 = 240°$.

The global frame $[x, y]$ denotes the environment of the robot. The robot's orientation and location in this global frame can be denoted as (x, y, θ). The global velocity of the robot can be represented as $(\dot{x}, \dot{y}, \dot{\theta})$.

In addition, figure 3 shows that the elements link the robot with the environment are the wheels (wheel hubs), thus it will be the starting point to do the kinematics analysis.

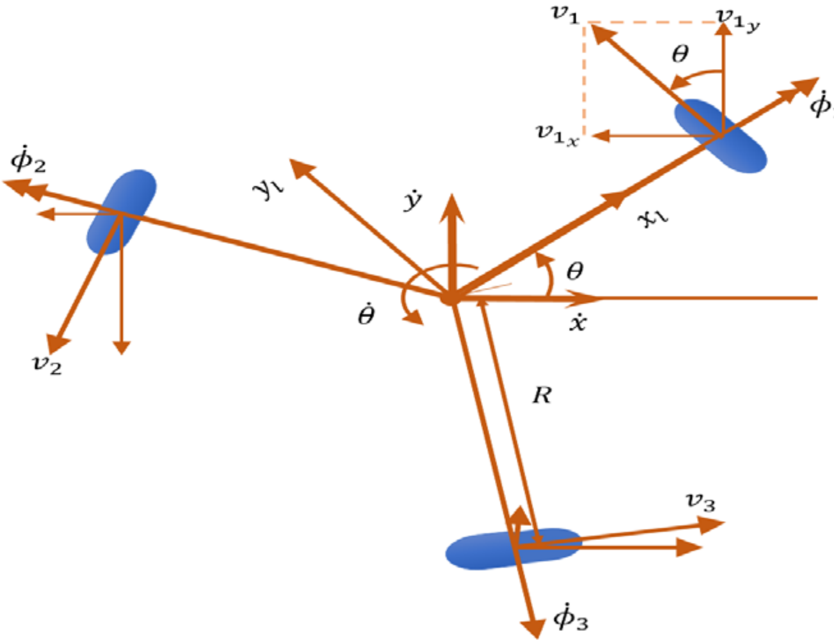

Figure 3: Kinematic diagram of the robot.

The angular velocity of the wheels ($\dot{\phi}_i$) can be defined from the following equation [7]:

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

For easier implementation in programming applications, equation (1) can be expanded into three separate equations:

$$\begin{aligned} \dot{\phi}_1 &= \left(-\sin(\theta)\cos(\theta)\,\dot{x}_L + cos^2\,(\theta)\dot{y}_L + R\dot{\theta}\right)/r \\ \dot{\phi}_2 &= \left(-\sin(\theta+\alpha_2\,)\cos(\theta)\,\dot{x}_L + cos\,(\theta+\alpha_2)\cos(\theta)\,\dot{y}_L + R\dot{\theta}\right)/r \\ \dot{\phi}_3 &= \left(-\sin(\theta+\alpha_3\,)\cos(\theta)\,\dot{x}_L + cos\,(\theta+\alpha_3)\cos(\theta)\,\dot{y}_L + R\dot{\theta}\right)/r \end{aligned} \qquad (2)$$

3.2 Robot trajectory plotting

Trajectory is the path that a robot follows through space as a function of time. It helps to study the robot kinematics model behaviour and path planning. To develop a trajectory plotting model for the omnidirectional mobile robot, first it is needed to derive the forward kinematics for the robot. Then the robot position equations can be calculated. The forward kinematics model can be developed from equations (3, 4 and 5) as follow [8,9]:

$$\dot{x}_L = r \times \left(\frac{\dot{\phi}_3}{\sqrt{3}} - \frac{\dot{\phi}_2}{\sqrt{3}}\right) \tag{3}$$

$$\dot{y}_L = r \times \left(-\frac{\dot{\phi}_2}{3} + \frac{2 \times \dot{\phi}_1}{3} - \frac{\dot{\phi}_3}{3}\right) \tag{4}$$

$$\dot{\theta} = \frac{r}{3 \times l} \times \left(-\dot{\phi}_1 - \dot{\phi}_2 - \dot{\phi}_3\right) \tag{5}$$

The robot trajectory is drawn in global frame. Then to convert from local frame to global frame as shown in equation (6 and 7)

$$\dot{x} = \dot{x}_L \cos(\theta) - \dot{y}_L \sin(\theta) \tag{6}$$

$$\dot{y} = \dot{x}_L \sin(\theta) + \dot{y}_L \cos(\theta) \tag{7}$$

Finally, the trajectory can be plotted from the following equations:

$$x = \dot{x} \times t \tag{8}$$

$$y = \dot{y} \times t \tag{9}$$

$$\theta = \dot{\theta} \times t \tag{10}$$

Where: x, y are the robot offsets from the global coordinates reference point (along the x-axis and y-axis respectively) and θ is the robot orientation with respect to the global coordinates.

3.3 Discrete PI Algorithm

The omnidirectional robot is controlled by a digital system. The digital system operates in the discrete domain while the analogue electronics operate in the continuous domain. This indicates that any implementation of a control law should be made in discrete form. In a discrete system the integral control action is represented by difference equation. Figure 4 shows sampling of an analogue signal to a discrete signal.

Where k in figure 4 represents the sample number, and T is the sample period. e(k) is the error at the k_{th} sample period. The velocity PI algorithm can be written as [10,11]:

$$u_k = u_{k-1} + K_p[e_k - e_{k-1}] + K_i\, e_k \tag{11}$$

In more compact form:

$$\Delta\, u(k) = u_{k-1} - u_k = K_p[e_k - e_{k-1}] + K_i\, e_k \tag{12}$$

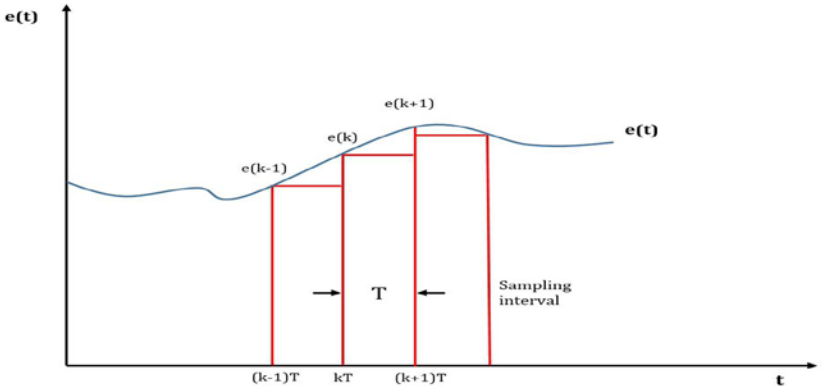

Figure 4: Sampling of an analogue signal to a discrete signal.

3.4 Robot Performance Measurement

The robot performance can be tested by applying different algorithms, using various sensors and motion capacities or by defining benchmarks such RoboCup Competitions, or search and Rescue. A robot must have sensing, decisional and actuation capabilities to be autonomous. Metrics and indicators are needed to be defined to achieve a useful evaluation. The following Attributes can characterize the task execution [12]:

- Energy consumption.
- Time limit.
- Robot reactivity.
- Reactivity is the time required by the system to respond to events. It is divided into two parts: time to processing the event and time for event detection.
- Robot robustness.
- Robustness is the robot’s capability to adapt to the change in the environment. The robot capacity to act robustly with respect to environment variations defines its autonomy.
- The environment as such.
- The environment is described by a complexity value and by a size. The complexity influences robot robustness and reactivity. The Environment size influences task duration and energy consumption.
- The quantity of information that robot has about the environment.
- The robot performance (such as velocity, time or power consumption) can be linked to environment information and parameters.
- Robot Repeatability.
- Robot kinematic model evaluation.

4. Calibration and Test

Adjustments and tests are required to acquire results that are more applicable practically with factors that may have been discounted (or were difficult to measure) theoretically.

4.1 Tuning the velocity PI algorithm parameters

To get the desired motor velocity response, the velocity PI algorithm gains (K_p and K_i) must be tuned. The robot is connected to a laptop through a USB cable in order to instantaneously monitoring and storing the robot wheels' velocity. Then the collected data are plotted in Matlap software to study the velocity response performance. A manual tuning is used to adjust the PI gains (K_p and K_i). The tuning procedure is run for both load and no-load conditions (see figure 5).

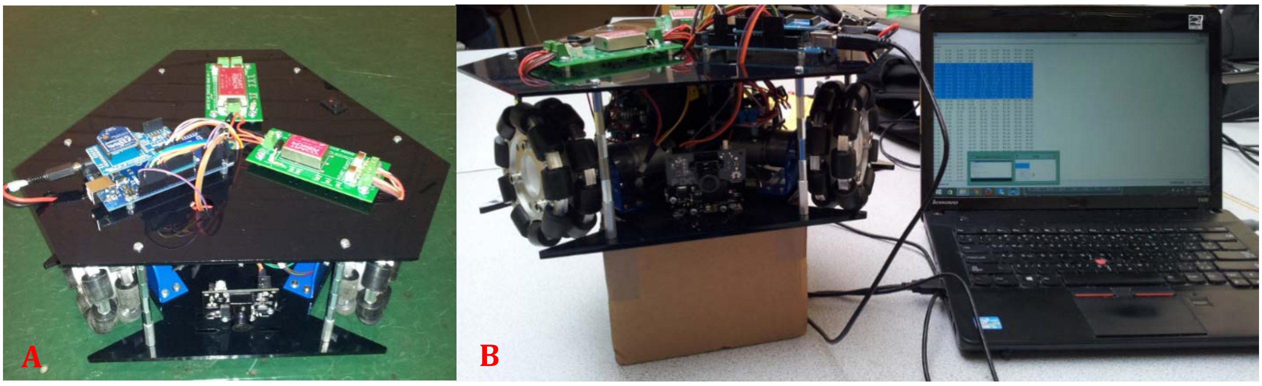

Figure 5: Tuning the velocity PI algorithm parameters (A) Load (B) No load.

A sample of three reading of 'tuning for performance' results at wheel velocity = 15 rad/sec are shown in table 1.

Table 1: Results of 'tuning for performance' experiment for velocity control.

Performance indicator	Kp=0.1 Ki=0.1	Kp=0.1 Ki=0.3	Kp=0.1 Ki=0.5	Loading condition
Settling time (sec)	7.5	6	8	No load
	8	4	3.5	Load
Overshoot	No overshoot	9%	10%	No load
	No overshoot	3%	7%	Load
Rise time (sec)	4	0.5	1	No load
	1.5	1	0.6	Load

The best system response is when $K_p = 0.1$ and $K_i = 0.3$, because of that, it has been decided to choose this gains value in the project.

From tuning the velocity PI algorithm parameters, it has been concluded the following:

- Both tuning parameters (K_p and K_i) can interact with each other and their effect must be balanced.
- The oscillatory or rolling behaviour of the motor response, tends to be increased by the integral term.
- PI control gives a slow but more accurate response than P control.
- There is usually an Overshoot in velocity control.

4.2 Theoretical Evaluation for the robot Kinematics

The equations derived for 'Inverse Kinematics' (in section 4.1) is required to be verified prior to applying and testing it physically in order to prevent any unpredicted result. In order to verify the Inverse Kinematic Analysis, an arbitrary robot local velocity and direction are selected as following: The local velocity of the robot $\dot{x}_L$= 0.5 m/sec, $\dot{y}_L = 0$ m/sec, and $\dot{\theta}$ = 0 rad/sec and choose θ= 0°.

$$\dot{\phi}_1 = \left(-\sin(\theta)\cos(\theta)\,\dot{x}_L + cos^2\,(\theta)\dot{y}_L + R\dot{\theta}\right)/r$$

$$= \frac{(-\sin(0)\cos(0) \times 0.5 + cos^2\,(0) \times 0 + 0.13 \times 0)}{0.051} = \ 0\,\frac{rad}{sec}$$

$$\dot{\phi}_2 = \ \left(-\sin(\theta + \alpha_2\,)\cos(\theta)\,\dot{x}_L + cos\,(\theta + \alpha_2)\cos(\theta)\,\dot{y}_L + R\dot{\theta}\right)/r$$

$$= \frac{(-\sin(0 + 120\,)\cos(0) \times 0.5 + \cos(0 + 120)\cos(0) \times 0 + 0.13 \times 0)}{0.051} = \ -8.49\,\frac{rad}{sec}$$

$$\dot{\phi}_3 = \ \left(-\sin(\theta + \alpha_3\,)\cos(\theta)\,\dot{x}_L + cos\,(\theta + \alpha_3)\cos(\theta)\,\dot{y}_L + R\dot{\theta}\right)/r$$

$$= \frac{(-\sin(0 + 240\,)\cos(0) \times 0.5 + \cos(0 + 240)\cos(0) \times 0 + 0.13 \times 0)}{0.051} = 8.49\,\frac{rad}{sec}$$

Hence, these wheels' velocity values derived from Inverse Kinematics are to be verified by the Forward Kinematics equations (3), (4) and (5) as shown below:

$$\dot{x}_L = r \times \left(\frac{\dot{\phi}_3}{\sqrt{3}} - \frac{\dot{\phi}_2}{\sqrt{3}}\right) = 0.051 \times \left(\frac{8.49}{\sqrt{3}} - \frac{(-8.49)}{\sqrt{3}}\right) = 0.5\,\frac{m}{sec}$$

$$\dot{y}_L = r \times \left(-\frac{\dot{\phi}_2}{3} + \frac{2 \times \dot{\phi}_1}{3} - \frac{\dot{\phi}_3}{3}\right) = 0.051 \times \left(-\frac{(-8.49)}{3} + \frac{2\,\times\,0}{3} - \frac{8.49}{3}\right) = 0\,\frac{m}{sec}$$

$$\dot{\theta} = \frac{r}{3 \times l} \times \left(-\dot{\phi}_1 - \dot{\phi}_2\, - \dot{\phi}_3\right) = \frac{0.051}{3 \times 0.13} \times (-0 - (-8.49)\, - 8.49) = 0\,\frac{rad}{sec}$$

The results prove that the equations and methodology of Inverse Kinematics are accurate.

5. Practical Results

The practical results are shown and discussed in this section.

5.1 Robot Repeatability Test

Repeatability is an important measurement for the robot performance. In this test, the robot was programmed to perform a specific cycle routine. Then the ability of robot to return to the same mark point with only the smallest amount of variation has been considered by measuring the error distances between the reference position and the different attempts position. Figure 6 shows a triangle path that the robot was programmed to move in to perform the repeatability test.

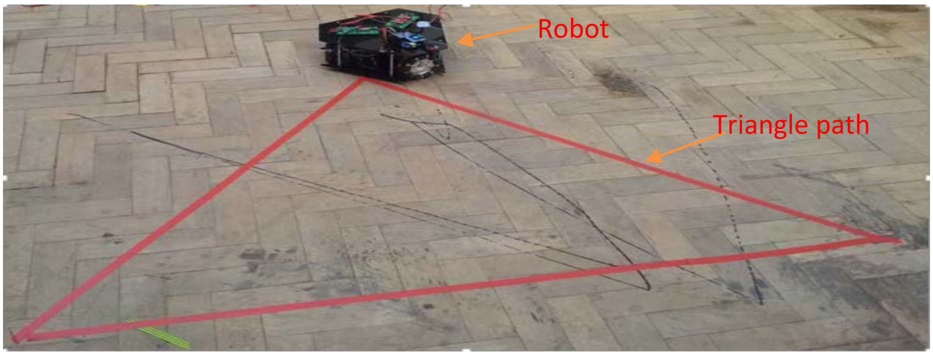

Figure 6: Robot repeatability test.

The test conditions include:

- The same procedure,
- The same observer,
- The same measuring instrument, used under the same conditions,
- The same location,
- Repetition over a short period of time.

The test cycle is:

Move with velocity =1 m/sec, and θ = 0.

Move with t velocity =1 m/sec, and θ = 120.

Move with velocity =1 m/sec, and θ = 240.

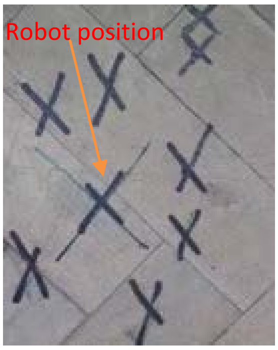

Figure 7: Errors in robot positions marked on the ground at the end of each cycle.

At the end of each cycle, the robot position was marked on the ground as shown in figure 7. Then the error distances were measured by using a measure tape.

In this test, it has been noticed that the robot almost successfully follows the triangle path. Table 2 shows the error at each cycle (attempt).

Table 2: Error in robot position at each cycle (attempt).

Cycle No	1	2	3	4	5	6	7	8	9	10
Error (cm)	3.75	3.5	3.25	4.5	4	3.75	3.9	3	3.25	4.25

Form table 2, it has been noticed that the errors in robot position is less than 5 cm for all attempts. These error values are within the acceptable limit for robot working in this environment. The error in robot position could be systematic or non-systematic, such as the Odometry error. Factors such as the error between the desired wheels velocity and the actual wheel's velocity, and contact surface (i.e. friction between the ground and the robot wheels, and surface conditions) appear to be the reason behind the robot position differences. If the repeatability test applied in better surface, the position error could be reduced. An improvement in robot system (such as using a different wheel design, changing the wheel's velocity control algorithm) could be have a positive effect on the measurements repeatability.

For current robot design it can be said that the measurements in table 2 are repeatable because the robot position variations are smaller than the agreed limit.

5.2 Test the robot velocity and acceleration limitations

In this test, the maximum velocity and acceleration of the robot have been measured.

Both values are collected by using two methods, one of them by connecting the robot to the laptop and them read the values, the other method by using a measure tape and stopwatch as shown in figure 8.

Figure 8: using a measure tape and stopwatch to measure the robot linear and velocities.

The procedure is to:

- Drive two wheels with the rated voltage (+12 v) in order to move the robot forward with the maximum linear velocity and acceleration, then record the results by using the two method.
- Drive the three robot wheels with the rated voltage (+12 v) in order to rotate the robot with the maximum angular velocity and acceleration, then record the results by using the two method.

Both procedures are shown in table 3.

Table 3: velocity and acceleration limitations of the robot.

Property	First method	Second method
Maximum linear velocity	1.13 m/sec	1.1 m/sec
Maximum linear acceleration	0.869 m/sec^2	x
Maximum angular velocity	7.5 rad/sec	7.4 rad/sec
Maximum linear acceleration	6.63 rad/sec	x

5.3 Forward motion test

In this test, the robot was moving forward while maintaining wheels' velocities of $\dot{\phi}_1 = 0, \dot{\phi}_2 = 10\frac{rad}{sec}, \dot{\phi}_3 = -10\, rad/sec$. The results have been collected and then plotted by using Matlab software as shown in figure 9.

As seen from figure 9, there are small different between the velocity response of the two wheels. This small difference will cause the robot to drift from the desired straight path by a small distance. From the trajectory plot (see figure 9) it can see that at an X-axis distance of 2.5m, there is a drift of 0.15m towards the Y-direction. This amount of drift is with the acceptable limit. The robot shows a fast response and the overshoot in velocity was acceptable. The error in wheels' velocities could be comes from factors such as the slip in the robot wheels, the surface condition, and systematic error. In conclude it can say that the robot has a positive performance in this test.

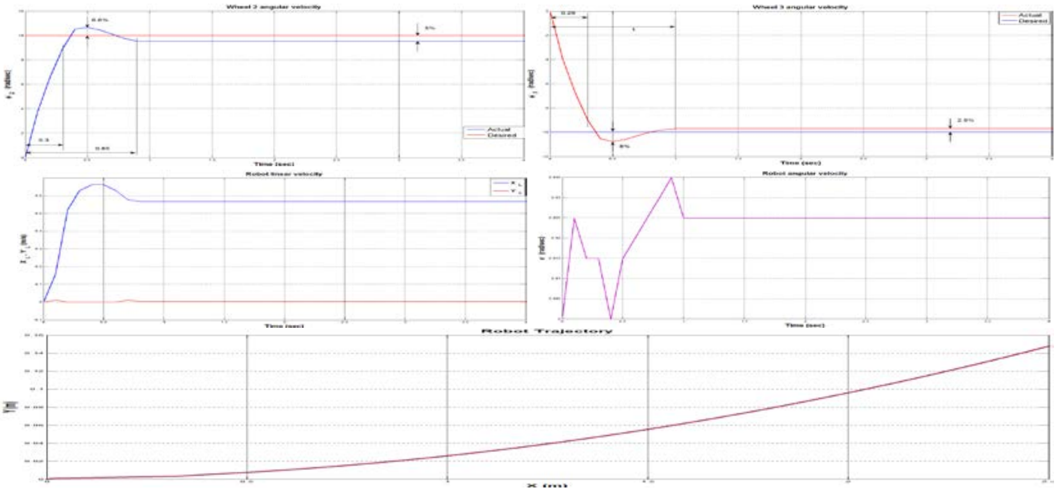

Figure 9: Robot data while moving forward with $\dot{\boldsymbol{\phi}}_1 = \mathbf{0}, (\dot{\boldsymbol{\phi}}_2) = \mathbf{10}\, \boldsymbol{rad/sec}, \dot{\boldsymbol{\phi}}_3 = -\mathbf{10}\, \boldsymbol{rad/sec}$.

5.4 Manoeuvrability test

This test was performed to prove that the robot is capable of doing an arbitrary motion in an arbitrary direction without changing the direction of wheels. An arbitrary starting angle was applied to the robot and the robot movement was studied. In order to verify that, a study for the robot movement with $\theta = 65^o$ and robot linear velocity = 1 m/sec will be mention in this dissertation. The robot has successfully moved from starting with 65^o, and it shows a positive manoeuvrability capability. To study the error in the robot orientation, the robot was moved for 4 m in the X-axis direction and then the distance and orientation drifts have been calculated. After moving 4 m in the X-direction, the robot actual trajectory has been plotted by using Matlab program form the collected robot velocities as shown in figure 10.

As seen in figure 10, when the robot moves distance of 4 m in the X-direction, it moves a distance of about 9 m in the Y-axis. By applying Pythagoras:

Robot moved distance = $\sqrt{(4)^2 + (9)^2} = 9.85\, m$

After moving 4 m in the X-axis direction, the robot has an angle with the X-axis = $\tan^{-1}\left(\frac{9}{4}\right) = 66°$

Then, there is an error in the in the angle by $66 - 65 = 1°$

This will cause a drift towards the Y-axis direction by $9 - 8.58 = 0.42\, m$

The drift in distance may come from several factors such as the slip in the robot wheels, the surface condition, and systematic error. In conclude the drift in distance will be acceptable for short distances, however when the robot moves for large distances the error will be bigger and the drift may cause a problem.

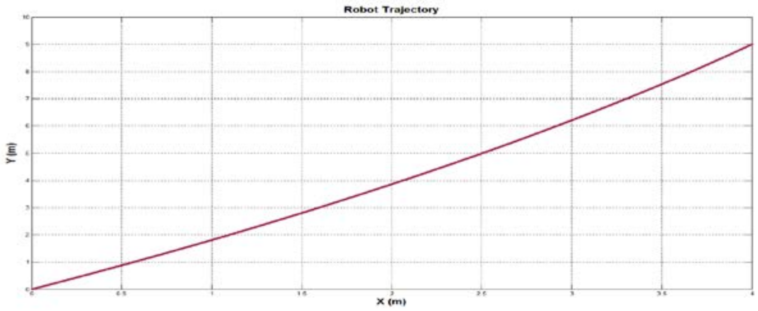

Figure 10: robot actual trajectory with $\boldsymbol{\theta} = \mathbf{65}°$ and robot linear velocity = 1 m/sec.

6. Conclusions

The aim of the research was to analysis, design, and implementation of an omnidirectional robot platform as the base for development of mobile robot projects such as RoboCup Competitions or search and Rescue. A detailed evaluation for the three Omni-wheels robot platform design has been discussed in terms of the mechanical and controller design requirements. Some of the mechanical parts were manufactured in the workshop, then all the mechanical and electrical components were assembled together in order to implement the robot. Inverse Kinematics analyses and trajectory plotting equations were presented, evaluated and then successfully applied on the robot. The velocity PI algorithm is used to control the robot wheels' speeds. Several tests were applied to the robot platform such as repeatability and manoeuvrability tests. In all tests, the data were collected by the laptop and has been plotted and evaluated. The robot showed the desired performance in these tests.

In conclusion, the primary aims and objectives of this dissertation has been successfully accomplished, since the robot design 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.

7. Future Work

- 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 object (i.e. ball) with a smoother path.
- By adding more sensors to the robot such as camera, SONAR, and compass, it would be possible for this robot to perform navigation applications.
- By Adding a kicking mechanism and with few modification, it would be possible for this robot to be tested in RoboCup competitions.
- Building a team of robots (with few modifications) and preform communication between them to

establish Search and Rescue application.

References

[1] T. P. do Nascimento, A. L. da Costa, and C. C. Paim, "AxeBot Robot the Mechanical Design for an Autonomous Omnidirectional Mobile Robot," in Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09., 2009, pp. 187-192.

[2] DOROFTEI, I., GROSU, V. & SPINU, V. 2007. Omnidirectional mobile robot–design and implementation, na.

[3] BORENSTEIN, J., FENG, L. & EVERETT, H. 1996. Navigating mobile robots: systems and techniques, AK Peters, Ltd.

[4] 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.

[5] 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.

[6] A. D. Team. (2010). Introduction about Arduino. Available: http://arduino.cc/en/Guide/Introduction

[7] T. A. Baede, "Motion control of an omnidirectional mobile robot," Traineeship report DCT 2006, 2006.

[8] J. Hedley, "Omnidirectional Robot Kinematics Modelling," Lecture note, 2014.

[9] M. T. Zaw, "Kinematic and dynamic analysis of mobile robot," 2004.

[10] R. Bicker, "Velocity PI Algorithm," Lecture note, 2014. 52

[11] J. Moreno–Valenzuela and V. Santibáñez, "Robust saturated PI joint velocity control for robot manipulators," Asian Journal of Control, vol. 15, pp. 64-79, 2013.

[12] D. Conrad and G. N. DeSouza, "Homography-based ground plane detection for mobile robot navigation using a modified EM algorithm," in Robotics and Automation (ICRA), 2010 IEEE International Conference on, 2010, pp. 910-915.