Dynamic Obstacle Avoidance for 4 Wheeled Omni Wheelchair using a New Type of Mecanum Wheel

Dynamic Obstacle Avoidance for Omni Wheelchair

Carlos Erlan Olival Lima
Department of Mechanical Engineering
Toyohashi University of Technology
Toyohashi, Aichi, Japan

Shigenori Sano
Department of Mechanical Engineering
Toyohashi University of Technology
Toyohashi, Aichi, Japan

Abstract:- The main goal of this project is the construction of a new type of electric wheelchair using omnidirectional wheels. We implemented some algorithms for dynamic obstacle avoidance based on reactive algorithms of the Bug family. Time and distance from one point to another are evaluated autonomously, and we compare the results of a wheelchair using Mecanum wheels and another one using a conventional wheel. From this simulation, it is possible to verify how effective it is the wheelchair proposed in this paper.

Keywords:- *Obstacle Avoidance, Wheelchair, Mecanum Wheel, Autonomous, Robot.*

I. INTRODUCTION

Robots can have holonomic or non-holonomic constraints and there are many applications in both cases. There are many examples of holonomic constraints that we can mention, such as a manipulator constrained through the contact with the environment. Another example is multiple manipulators limited through a standard payload. When it comes to non-holonomic constraints, we have mobile robot wheels, because of the robots that use wheels are considered non-holonomic. When we are developing the navigation system for autonomous robots, there are many variables that make the movement much more complicated. Among those variables we can cite: robot joint limits, self-collision, environment collision avoidance constraints, steering angle constraints in mobile robots, etc [1].

The objective of this project is the construction of an electric wheelchair using a new type of omnidirectional wheel. The wheelchair is equipped with various sensors and actuators and distance sensors and bump sensor are used to avoid collisions and keep the user safe. These sensors and actuators are controlled autonomously by a computer program, in the same way as happens in autonomous vehicles [2].

The electric wheelchair that we are developing can be considered a type of smart wheelchair. Currently, this type is becoming a substitute for conventional wheelchairs as an assistive device [3]. There are many kinds of research about the development of smart wheelchairs, and one of the main topics is the enhancement of the maneuverability. Many researchers are focused on studies related to the use of the wheeled omnidirectional running mechanism. There are many applications of Mecanum wheels, with its use in military, storage and transportation, social services, and other fields [4].

Comparing conventional vehicles and omnidirectional vehicles, the last one possess multiple advantages in terms of their mobility in narrow spaces and crowded environments. We can relate many benefits to this type of robot, but one of the most important ones is the ability to perform specific tasks in environments full of obstacles. Also, it can avoid quickly static obstacles or dynamic obstacles in narrow spaces [5].

Omnidirectional vehicles have 3 degrees of freedom (DOF) on the ground. The 3 degrees of freedom are longitudinal motion, lateral motion and center-point steering motion. Besides, any composite motion of the three degrees of freedom mentioned can be executed, so Omnidirectional vehicles are suitable for highly maneuverable, narrow, or accurate positioning occasions [5].

Fig 1:- DOF's in Mecanum Wheel, by Rotacaster [6]

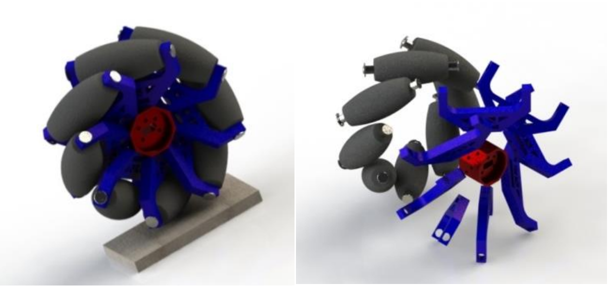
This project has as reference the research developed by Yanco [7]. This research has its focus on navigation in small environments, assisted driving, and human-AI interface. In this paper, the main focus is indoor navigation, more precisely inside an apartment containing multiple obstacles in the dynamic and static state. Besides, it also has its focus on obstacle avoidance. Objectively this means the wheelchair will collect data about the environment if it is available. However, it also can work without this information [2].

The focus of this work is the analysis of bug algorithms to the planning of trajectories in a four-wheeled Omni wheelchair. The time and distance the robots takes to move from one point to another are evaluated autonomously while the robot avoid the obstacles in the path. The simulations are done to define the effectiveness and efficiency of the algorithms. The analysis has been done through tests in a simulated environment developed by the software V-REP, which seeks to emulate the real behavior of robots in various conditions [8].

II. PROPOSED DESIGN FOR MECANUM WHEELS

In 1973, the Swedish engineer Bengt Ilon invented the first design of the Mecanum wheel. The design of it consists of a set of k congruent rolls placed symmetrically around the wheel body. The face of each roll is part of a surface of revolution R whose axis b is skew to the wheel axis a [9].

Many similar designs were proposed, and generally, the differences among these designs are the type of materials used, the way how the rollers are attached to the hub, the number of rollers, the relative angle of the rollers concerning the wheel, etc [4].



(a) (b)

Fig 2:- (a) Proposed wheel (b) Exploded view

In this project, the design developed for the Mecanum Wheel is composed of a central part to fix the other components (which we call hub) and eight forks with its respective roller as it can be seen in figure 2a and figure 2b. The diameter and width of the Mecanum were designed to be 229.22 millimeters and 74.5 millimeters. Wheelchairs have maximum dimensions defined by law. Therefore the characteristics of the Mecanum Wheel proposed were chosen based on the features of the wheelchair. Thus, the maximum speed of the wheelchair is 5 Km/h, and the weight capacity, a person of 80 kilograms. The material used for the hub is AISI 1020, for forks is aluminum, and for the rollers is nylon. The angle between each fork and the central axis of the wheel is 45°, and they are connected to the hub by two bolts for easy installation and replacement in case of damage or malfunction.

Moreover, we have eight rollers connected to the forks through the pins at each fork. That allows the rollers rolling freely by the action of frictional forces generated upon contact with the ground. In figure 3 is shown the design of the rollers with its dimensions.

III. NAVIGATION ALGORITHM

We chose the software V-REP to perform the simulations because it has an abundance of different programming techniques for its controllers. Besides, it allows to create scenarios similar to situations we can find in real life, and it reduces the difficulties for the users once it has a multitude of functions that eases the programming task.

V-REP can perform a multitude of different simulations, including simulation of complex assembly chains in a factory, robot controller, path planning, algorithm optimization, sensors test, and several other types of simulations.

During the simulations, dynamic obstacle avoidance is performed using Bug Algorithms. Bug algorithms are navigation methods for mobile robot using local planning. Three assumptions about the robot can be inferred from Bug algorithms. First, the robot is considered a point in space. Second, the robot has the perfect location capability, and third, the robot has accurate sensors. These assumptions are unrealistic in a real environment. However, Bug algorithms can be used as the first step towards a solution to the navigation problem, and they are considered the simplest algorithms to be used to solve the navigation problem [8]. The most relevant and mentioned in path planning are Bug1, Bug2, and DistBug algorithms. In this paper, a modified version of DistBug algorithms is applied for the navigation of the wheelchair. Below, it is given some advantages and disadvantages of each one of those algorithms.

A. Bug1 Algorithm

- Very simple;
- Memory used is minimum;
- It does not undergo local minimum;

In this algorithm, first, the robot moves along a line until it hits the obstacle. After that, it begins to follow the obstacle edge, searching the target in this position in order to obtain the minimum distance. Then, at the same time the robot calculates the distance from the current position to the destination. Lastly, the robot stores the point that has the minimum distance [10].

B. Bug2 Algorithm

This procedure follows the following steps:

- It can run at any point on a continuous path;
- The robot follows the path until it is interrupted by an obstacle;
- Follow the edge and recalculate the path as far as the new position develop into the original position on path;
- Moving accordingly the previous route generated;

C. DistBug Algorithm

- It is the improved version, based on distance;
- It defines the shortest distance to bring better navigation and then, the robot can get its objective in less time;
- When the robot finds an obstacle during its movement, it initiates to follow that obstacle and at the same time calculates the distance from the current position to the place it wants to go, generating a new path.

IV. SIMULATION ENVIRONMENT

A. Dynamical Systems

In this paper, it is compared the time necessary for two robots to achieve a goal in the shortest time. One of the robots, it is represented by a wheelchair that is using Mecanum wheels, and the other robot is represented by a wheelchair that is using conventional wheels. In other words, it is compared to a holonomic system with a non-holonomic system.

Vehicles using a non-holonomic system have the capacity of being applied in numerous projects, but the path planning for them is complicated because they are subject to rolling constraints that limit the possible directions of motion: it is impossible for them to move laterally directly, then they must make a lot of different movements until it reaches certain points on space. Therefore, such vehicles have difficulties in moving in narrow spaces, once complicated maneuvers must be required [11].

On the contrary, vehicles using a holonomic system, especially Mecanum wheels, they have the advantage of efficiently perform specific tasks in congested environments foreseen with static obstacles, dynamic obstacles, or narrow areas. That is why Mecanum wheels are adequate to use in wheelchairs, once the users are frequently dealing with small spaces.

B. State-Space

In this section, it is shown some standard equations that specify the vehicle configuration, velocities and accelerations [11]. In state-space X, it is found the following equation:

$$\dot{x} = f(x) \quad (1)$$

If it is more important to generate feasible paths rather than optimal trajectories, the state-space frequently will be configuration space [11]; in this case, the dynamical quantities can be ignored. Otherwise, it will usually contain velocity as well as configuration dimensions. Equation (1) shows the variable f(x), which specifies a vector field on state space.

In the case of the wheelchair, we assume the system has some controls u (steering, accelerator, etc.) and this modifies the dynamics accordingly (2).

$$\dot{x} = f(x, u) \quad (2)$$

Each setting of the controls gives a different vector field or dynamical law f (-, u). It is possible to drive the system to different regions of the space by manipulating u [11]. A feedback control law u=u (t, x) tells the vehicle how to respond at each time and at each point of state space. Such laws generate an implicitly closed-loop dynamics given by (3).

$$\dot{x} = f(t, x) \quad (3)$$

Equation (3) is chosen to achieve some given task such as path following or parking. Sometimes, in order to make the system linear in u it is attributed the following parameters:

$$\dot{x} = f_o(x) + \sum_i f_i(x)\, u^i \quad (4)$$

In this case, all the systems can be considered as linear, and then the drift-free f_o is equivalent to zero in this case. In this situation, the possible directions of motion at a point are linear combinations of the control fields, so the state velocity is implicitly constrained to lie in Span $\{f_i\}$ [12].

Even if the system is not linear, once we alternate between several different control vectors, it is possible to apply any convex linear combination of them in a practical way, even though there is still a kind of residual linearity in u. In fact, it turns out that many aspects of non-linear control are well captured by the linear theory [13].

C. Obstacle Avoidance

Outdoor or indoor environments, wheelchairs often have to deal with obstacles around it. Therefore it is important to develop a dynamic obstacle avoidance system. That is a complex problem, once the obstacles in the environment have to be recognized, and we need to calculate the distance until object. Moreover, the program has to identify whether the wheelchair is on a collision course. If the distance from obstacles is too far, that complicates obstacle avoidance [2].

Even if in this paper, we are keeping the focus in indoor environments, it is set with several obstacles; therefore, obstacle avoidance is an essential issue for this project.

The complexity of the obstacle avoidance algorithm depends mostly on the application. For a very simple wheelchair, it could be enough just to stop the movement once an obstacle is detected. More sophisticated algorithms define the next action based on the position and size of an object, then a path can be generated around the obstacle [14]. The set-back is that long computation times are necessary. Moreover, in this scenario, the implementation is often complicated. In some implementations, it is even required to stop the wheelchair, then the new route is calculate, and the wheelchair can continue its movement.

How it was mentioned before, in this paper, it is used the DistBug algorithm to avoid collisions with the obstacles around the robot. An initial position is set, and then all the

ISSN No:-2456-2165

possible paths are generated between the initial position and the target. In case the robot finds an obstacle, it avoids the obstacle following the borders of it, and once the robot overcomes the obstacle, it generates a new path towards the target. This method lends itself very well to an effective way of avoiding obstacles.

D. Localization

An apartment was developed by V-REP, and then several obstacles representing the furniture of this apartment were added to the environment scene as can be observed in Figure 1.

Fig 3:- Simulation Environment by V-REP

A path is generated from an initial point to an end point representing the target. The path is generated in a way it avoids the obstacles in the environment. In V-REP, a path created allows complex movement definitions in space (succession of freely combinable translations, rotations and/or pauses), and are used for controlling the robot along a predefined trajectory [15].

In this environment, a set of ultrasound sensors are used to avoid collisions. In case the wheelchair has no prior knowledge of the elements on space, this problem is similar to the Simultaneous Localization and Mapping (SLAM) [16] problem.

When it is performing SLAM to solve the robot localization, we assume there is already a predefined map of the environment in the setup. SLAM are performed in natural landmarks that are easily matched in subsequent observations of the environment. However, this is not relevant in the simulation applied in this paper, since no map of the environment must be generated [2].

Landmarks are localized in the environment, and they used to create the path. Those landmarks are calculated by the calculation modules offered by the software V-REP. The calculation module used is the Path/motion planning module. This module handles holonomic path planning tasks and non-holonomic path planning tasks (for car-like vehicles) via an approach derived from the Rapidly-exploring Random Tree (RRT) algorithm [17]. The position of each landmark is saved in the environment, and then, we calculate their position relative to the wheelchair. The wheelchair movement is then calculated based on its relative position to the landmarks, this way we make the wheelchair follow a path. When at the new position, landmarks are detected, and they match with the simulated location. The steps are shown in Figure 4.

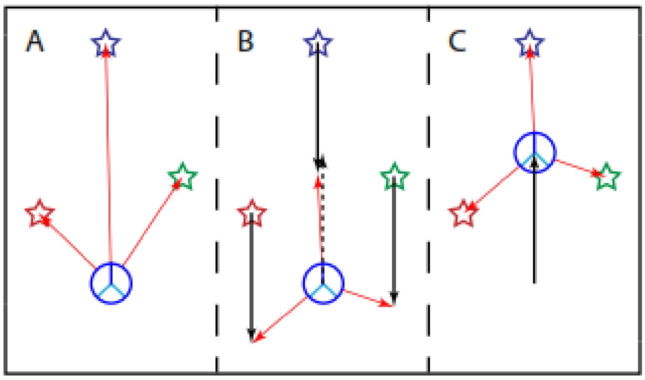


Fig 4:- Localization using landmarks. A: It is defined the position of landmarks in the environment. B: It is calculated the position relative to the wheelchair. C: Found landmarks are checked at a new position accordingly simulated locations of landmarks [2].

In order to detect the obstacles, this wheelchair uses an arrangement with seven ultrasound sensors. It is used a sensor of 40 kHz, +/- 30 degrees given the speed of sound in air at ambient temperature, which returns the distance to the nearest obstacle in its field of view [18]. It is used a combination of 7 ultrasound sensors to identify any obstacle been positioned in front of the wheelchair or positioned at right or left. This configuration is shown in Figure 5.

The range of the five sensors distributed around the center point of the wheelchair is 40 centimeters while the range of the sensors placed laterally at right and left is 20 centimeters. This range was set after several simulations taken with different values for the range, and in the end, this was the best configuration for this arrangement.

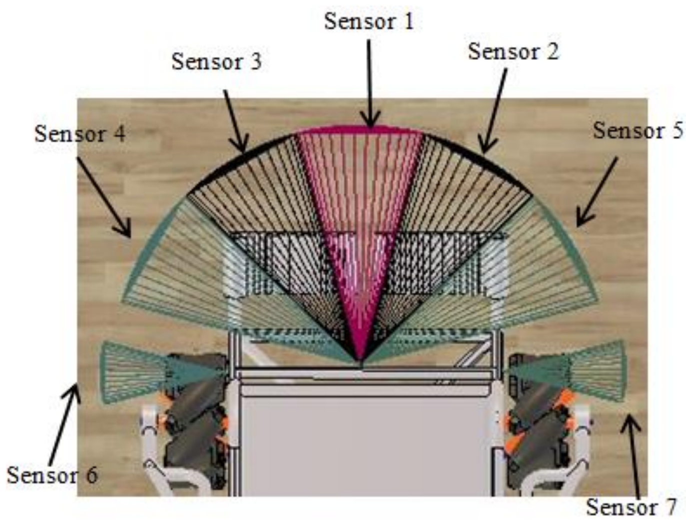


Fig 5:- Set of Ultrasound Sensors

ISSN No:-2456-2165

V. DESIGN OF THE EXPERIMENT

The experiment consists of a set of different simulations for a wheelchair performed by the software V-REP. In the simulations, the wheelchair has to go from an initial point to an objective avoiding various obstacles. These simulations were run on environments simulating a simple apartment as it was shown in Figure 3.

A start position is defined as the wheelchair, and six different locations are defined as the final position of the wheelchair. Two different types of wheelchair are used in those simulations. One of them uses the Mecanum Wheel proposed in this project and presented in Figure 2, and the other type of wheelchair uses a conventional wheel. The objectives of this simulation are comparing a holonomic system with a non-holonomic system and determine how efficient it is the Mecanum Wheel proposed in this project.

In table 1 it is shown the initial position of the wheelchair and the position of the goal in each one of the simulations, as well as the distance described by the wheelchair from the initial point to the objective.

The simulation with each type of wheelchair is executed six times, each time keeping the position of the wheelchair and changing the location of the target.

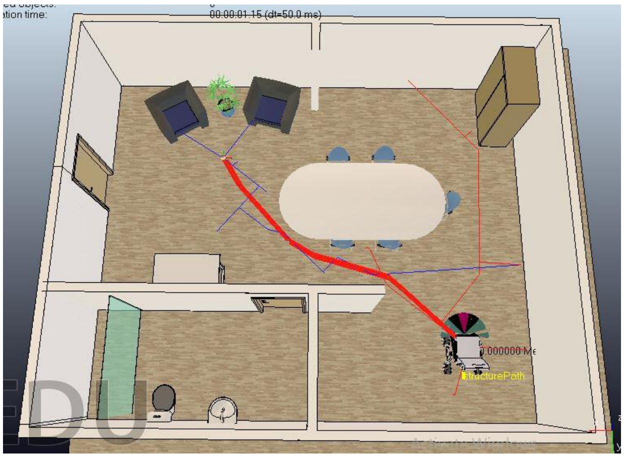
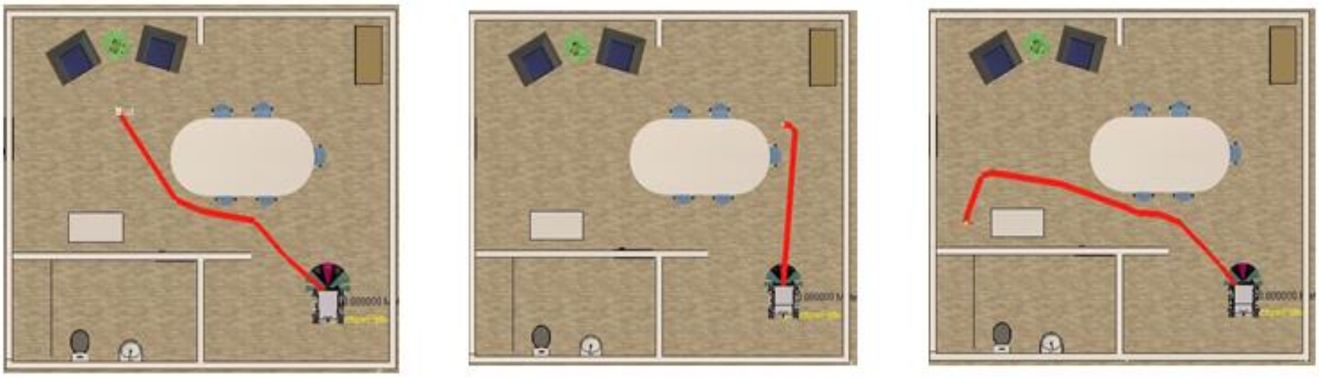


Fig 6:- Obstacle Avoidance (Simulation 1)

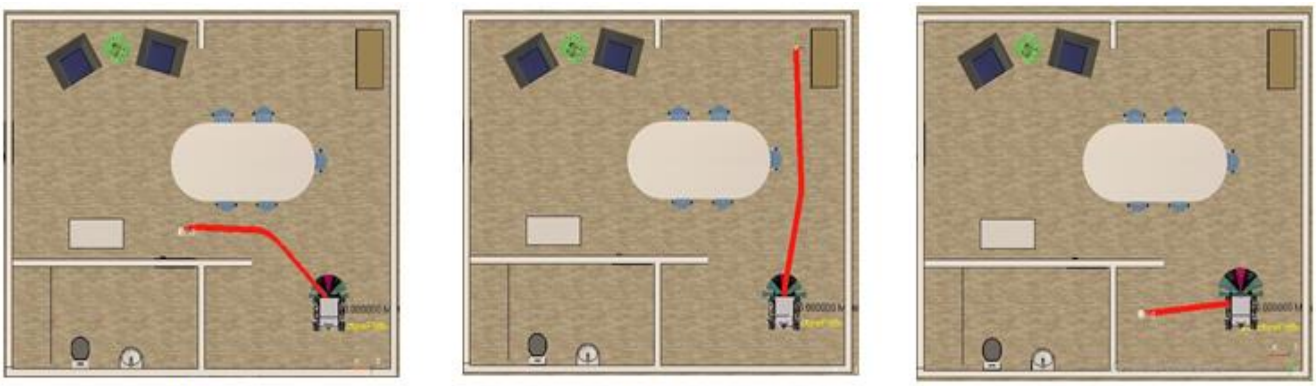
Figure 6 shows the possible path that can be followed by the wheelchair to go from the initial point to the objective. The wider line represents the shortest path that must be followed by the wheelchair.

In Figure 7, it is possible to verify the path described by the wheelchair related to each one of the positions in Table 1. The path that each type of wheelchair must follow initially is the same in each position; however, the time necessary to achieve the goal will be different as it is shown in Table 1.



(A) (B) (C)

Fig 7:- Comparison of the Paths Generated by the Wheelchair Related to table 1 (A) Position 1 (B) Position 2 (C) Position 3



(D) (E) (F)

Fig 8:- Comparison of the Paths Generated by the Wheelchair Related to table 1 (D) Position 4 (E) Position 5 (F) Position 6

	Position	Time (s)		Distance (m)	
Target	(x, y, z)	Omnidirectional	Conventional	Omnidirectional	Conventional
1	(2.36, -2.02, 0.57)	48.79	53.49	6.3024	6.3660
2	(-2.26, -1.75, 0.57)	10.15	19.74	3.7074	3.6244
3	(4.04, 0.38, 0.57)	46.14	60.49	7.6617	7.7810
4	(0.98, 0.35, 0.57)	30.99	32.69	3.9009	3.8253
5	(-2.54, -3.40, 0.57)	31.39	31.69	5.1823	5.1735
6	(-0.09, 2.00, 0.57)	8.35	18	2.1985	2.1985

Table 1:- Positions and Distance Metrics for a Four-Wheeled Omni Wheelchair

Figure 9 and Figure 10 show the results obtained in relation to the evaluation of the average speed of displacement (m/s). Figure 9 presents the results related to the wheelchair using a conventional wheel, while Figure 10 shows the results related to the wheelchair using omnidirectional wheels. It can be noted that the wheelchair is more effective when it is using omnidirectional wheels, once it gets to reach the target faster compared when it is using conventional wheels. This proves how effective it is the use of omnidirectional wheels when we need to deal with narrow spaces. This happens because, with omnidirectional wheels, the wheelchair can move in any direction quickly; this way, it can overcome obstacles faster and efficiently. On the contrary, when the wheelchair is using conventional wheels, it has some difficulties to overcome some obstacles because it needs to make many turns until it gets to overcome the obstacle. As the main example, we can observe the Figure 8 (F), where the target is located on the left side of the wheelchair. In this case, if the wheelchair is using omnidirectional wheels, it can move directly from the start point to the endpoint, but when the wheelchair is using conventional wheels, first it needs to turn left until it faces the direction of the target, then it can move along the path until it finds the target.

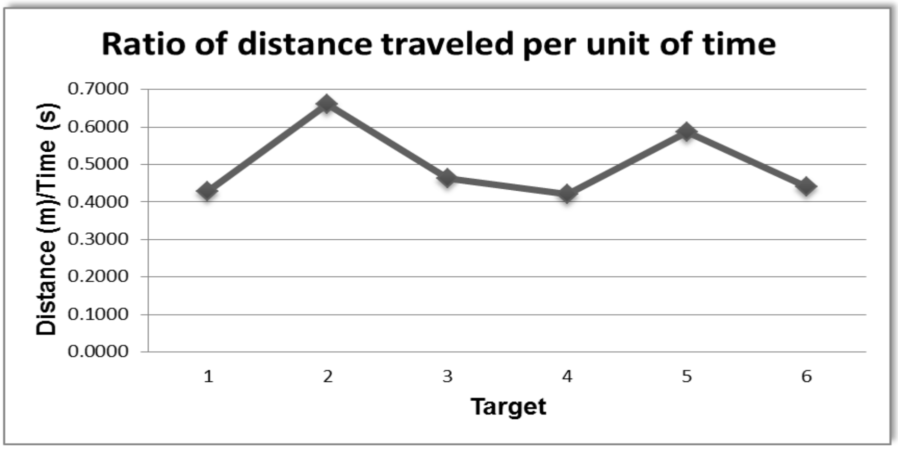


Fig 9:- Average Speed of the Wheelchair using Conventional Wheels

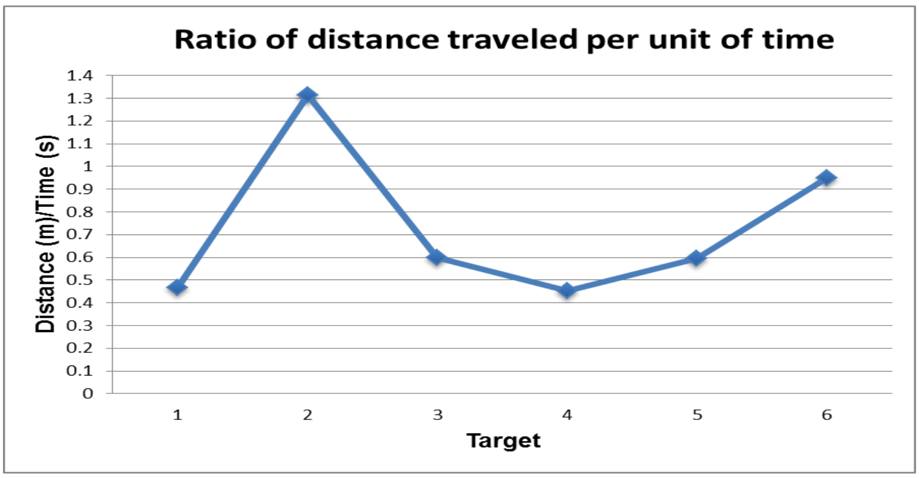


Fig 10:- Average Speed of the Wheelchair using Omnidirectional Wheels

VI. CONCLUSION

It has performed an evaluation on the simulator V-REP to test the efficiency of a new type of electric wheelchair using Mecanum wheels. To achieve this goal, a set of simulations is executed on an environment similar to a small apartment with some furniture representing the obstacles. The DustBug algorithm has been used in the simulations presented in this paper, because it offers a simple solution to go from an initial point to a target point. And also two metrics have been defined and presented along the paper. Those metrics are based on run-time and distance traveled.

The results are comparing how effective a wheelchair can avoid obstacles and reach a certain point on space when it is using Mecanum Wheel and when it is using conventional wheels. In those simulations, it is possible to verify that an Omni wheeled wheelchair is more useful to overcome obstacles in narrow spaces.

In the future, we are planning to implement the potential field method to make the wheelchair capable of finding the target even faster and also guarantee the obstacle avoidance even when a dead end is found.

ACKNOWLEDGMENT

The authors acknowledge the scholarship received from Ministry of Education, Culture, Sports, Science, and Technology – Japan (MEXT) and the Toyohashi University of Technology for their support and encouragement in the development of this research.

REFERENCES

[1]. Divelbiss, Adam, et al. “Kinematic Path Planning for Robots with Holonomic and Nonholonomic Constraints.” SpringerLink, Springer, New York, NY, 1 Jan. 1998, link.springer.com/chapter/10.1007/978-1-4612-1710-7_5.

[2]. (PDF) Autonomous Wheelchair: Concept and Exploration. www.researchgate.net/publication/236882346_Autonomous_Wheelchair_Concept_and_Exploration..

[3]. Kundu, Ananda Sankar, et al. “Design and Performance Evaluation of 4 Wheeled Omni Wheelchair with Reduced Slip and Vibration.” Procedia Computer Science, vol. 105, 2017, pp. 289–295., doi:10.1016/j.procs.2017.01.224.

[4]. Ramirez-Serrano, A., et al. “Elliptical Double Mecanum Wheels for Autonomously Traversing Rough Terrains.” IFAC Proceedings Volumes, vol. 43, no. 16, 2010, pp. 7–12., doi:10.3182/20100906-3-it-2019.00004.

[5]. Florentina Adascalitei, Ioan Doroftei. "Practical Applications for Mobile Robots based on Mecanum Wheels – a Systematic Survey." Proceedings of International Conference On Innovations, Recent Trends And Challenges In Mechatronics, Mechanical Engineering And New High-Tech Products Development – MECAHITECH'11, vol. 3, year: 2011, link: https://www.academia.edu/1079217/Practical_Applications_for_Mobile_Robots_based_on_Mecanum_Wheels-a_Systematic_Survey

[6]. Triple Rotacaster commercial industrial omni wheel, Rotacaster Wheel Pty Ltd, 2011.

[7]. H. Yanco, “Integrating robotic research: a survey of robotic wheelchair development,” in AAAI Spring Symposium on Integrating Robotic Research. Citeseer, 1998, pp. 136–141.

[8]. Yadira Quiñonez, Fernando Barrera, Ian Bugueño, Juan Bekios-Calfa. “Simulation and path planning for quadcopter obstacle avoidance in indoor environments using the ROS framework,” J. Mejia et al. (eds.), Trends and Applications in Software Engineering, Advances in Intelligent Systems and Computing 688, Springer International Publishing AG 2018.

[9]. A. Gfrerer. "Geometry and kinematics of the Mecanum wheel." Computer Aided Geometric Design 25 (2008) 784-791.

[10]. Sezer, V., Gokasan, M.: A Novel Obstacle Avoidance Algorithm: "Follow the Gap Method." Robotics and Autonomous Systems vol. 60(9), pp. 1123-1134 (2012)

[11]. Motion Planning for Nonholonomic Vehicles: An Introduction. hal.inria.fr/inria-00548415/document.

[12]. Zexiang Li and J. F. Canny, editors. Nonholonomic Motion Planning. Kluwer Academic Publishers, 1993.

[13]. H. J. Sussmann. Lie brackets, real analyticity and geometric control. In R. W. Brockett, R. Millman, and H. Sussmann, editors, Differential Geometry and Geometrical Control Theory, volume 27 of Progress in Mathematics, pages 1–116, Boston, 1983. Birkhäuser.

[14]. J. Crowley, “World modeling and position estimation for a mobile robot using ultrasonic ranging,” in Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on. IEEE, 1989, pp. 674–680.

[15]. V-REP simulator: http://www.coppeliarobotics.com

[16]. M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[17]. J. J. Kuffner Jr., “RRT - Connect: an Efficient Approach to Single-Query Path Planning,” in *Proc of IEEE Int. Conf. of Robotics and Automation*, San Fransisco, USA, Apr. 2000.

[18]. G. Gibbs, H. Jia, I. Madani, “Obstacle Detection with Ultrasonic Sensors and Signal Analysis Metrics,” *Transportation Research Procedia*, Volume 28 (2017) pp. 173-182, ISSN 2352-1465.