# Design and Implementation of Novel Motion Planning for a WMR with the Presence of Obstacles

Essa Alghannam, Lu Hong, Yang Zhen and Oula Esmail
School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan, China
Email: essaalghannam@whut.edu.cn, landzh@whut.edu.cn, 781289620@qq.com, oulaesmail@yahoo.com

***Abstract*—Making the robot move on its own gives extra challenges such as the robot's ability to localize autonomously, generate a map of its environment and avoid the obstacles. This paper presents new methods to solve problems caused by the navigation of a self-directed Wheeled Mobile Robot (WMR) in unknown environments which may contain unknown-positions & dimensions regular rectangle static obstacles. This work has been implemented in several stages. Firstly, image processing which is developed in MATLAB in order to detect the obstacles edges and divide the environment into various spaces. Secondly, motion planning based on "Grid Method" in order to find all the possible paths and choose the best one. Thirdly, Chained Form Algorithm with Cosine Switch Control is used to steer 3-WMR from the start-point to the goal. The methods, which are developed in this work to come up with the free collision motion planning for wheeled robot, are completely innovative and unique. As well these methods are able to solve the problem of local minima. The new methods provide the set of conditions under which a motion planning based on the proposed solutions in this paper would bring advantages.**

***Index Terms*—obstacle detection, obstacle avoidance, collision-free motion planning of WMRs**

## I. Introduction

Making the robot move on its own gives extra challenges such as the robot's ability to localize autonomously, generate a map of its environment and avoid the obstacles. This paper focuses on the problems caused by the navigation of a self-directed Wheeled Mobile Robot (WMR) in unknown environments which may contain static regular rectangle obstacles of unknown-positions and dimensions. Recently study the WMRs has received more attention by researchers due to its importance in both industry and academia, especially the robots that serve in the hospitals or the robots that work in dangerous environments where human can't access easily.

Chained non-holonomic systems, potential field method, fuzzy logic technique and Neuro-fuzzy technique are used recently in the field of autonomous WMR navigation. Murray, M. and Sastry, S. introduced two common non-holonomic systems jumping robot and 4-WMR [1]. Chained control system algorithm with sinusoidal inputs based on Lie Algebra has been implemented for steering these systems with non-holonomic constraints between arbitrary configurations. They introduced chained form of non-holonomic control system with two inputs also and show by sinusoidal inputs how to steer a car and a car with a trailer attached between arbitrary states [2]. Bushnell, L., Tilbury, D. and Sastry, S. discussed the sinusoidal input control algorithm and its application in three input chained form non-holonomic systems [3]. For a fire truck system, Tilbury, D. proposed a multi-rate controls algorithm [4]. This fire truck system can be converted into three input chain systems. Sheng, L. studied the motion planning of non-holonomic systems based on bang-bang control for n-dimensional chained form systems with two inputs [5]. Liu, Y., Li, L. and Tan, Y. proposed a cosine switch control method for non-holonomic chained form system [6]. Deng, M. presented parking control scheme of a 2-WMR using artificial potential Lyapunov function [7]. Urakubo, T. proposed a feedback controller that makes a two WMR converge to a desired point among obstacles such as a flat wall and a pillar [8].

Ge, S. and Cui, Y. proposed a new potential field method for motion planning of mobile robots in a dynamic environment where the target and the obstacles are moving [9]. Rusu, C. and Birou, I. developed a fuzzy obstacle avoidance system for an autonomous mobile robot using IR detection sensors [10]. Yuki, O. addressed two issues in robotic application: an issue concerned with the verification of how well the existing heuristic methods compensate for uncertainty caused by sensing the unstructured environment, and an issue focusing on the design and implementation of a control system that is easily expandable and portable to another robotic platform aiming to future research and application [11]. Wei, L., Chenyu, M. and Wahl, F. proposed Neuro-Fuzzy system architecture to control of a mobile robot in unknown environments [12]. A neural network is used to understand environments. Its inputs are a heading angle between the robot and a specified target, and range information acquired by an array of ultrasonic sensors. The output from the neural network is a trained reference motion direction for robot navigation. Rusu, P. discusses

Manuscript received February 11, 2018; revised April 1, 2018.


doi: 10.18178/joace.6.1.31-39

a Neuro-fuzzy controller for sensor-based mobile robot navigation in indoor environments [13]. The control system consists of a hierarchy of robot behaviors.

Therefore, the present work is concerned with providing methods to obtain the collision-free motion planning of a non-holonomic system such as 3-WMR moving in unknown environments which may contain static regular rectangle obstacles (walls or barriers) between the home and goal points. Compared with other works, this work is able to solve the problem of local minima and that would bring advantages.

The principle of the Grid Method is: firstly, find interim goals by dividing the environment into small rectangle areas. Secondly, detect the available rectangle areas in order to convert the global path between the start point and the goal point into a group of partial-paths (local paths). Finally, find the local paths by using “the cosine switch control”. The overall path which consists of a group of sequential local paths of a series of sequential available areas which are linked between the start point and goal point is called the Global Path. This paper also proposes an image processing method to detect and enlarge the static regular rectangle obstacles of the environments.

## II. Obstacles Detection and Enlarging Methods

Special kinds of unstructured environments with static obstacles were studied and analyzed in this paper. Fig. 1 describes the specification of the environment and obstacles, whereas the shape of both the environment and the obstacles is a rectangle, the borders of the obstacles are parallel or perpendicular to the borders of the rectangle environment “regular” and the obstacles have unknown-position and unknown-dimensions. An off-board sensor camera is installed on the top of the environment to be used in the image processing.

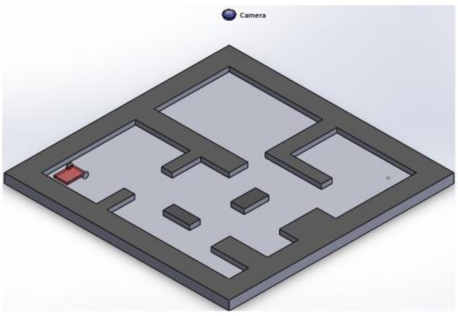


Figure 1. 3D-model of an environment with unknown regular-rectangle static obstacles.

The main goal of the proposed image processing method is to detect the borders of the studied obstacles. So the difference in color between the background and the obstacles must be clear enough to ensure the image processing success.

### A. Obstacles Detection Method

The goal of obstacles detection is to install a grid of horizontal and vertical pink lines which contact with the real borders of the obstacles; these lines will divide the original image of the real environment into available and forbidden spaces. By using the absolute value of the first order partial derivative of the original image function *f(x,y)* with respect to *x, y* respectively, the horizontal and vertical edges will be detected as shown in Fig. 2.


Figure 2. Edge detection image.

The gradient of an image is a change in color. An edge in an image occurs when the gradient is greatest. The Sobel operators can be used based on this fact to find the edges in an image. They calculate the approximate image gradient of each pixel by convolving the image with a pair of 3×3 filters. These filters estimate the gradients in the X and Y directions and the magnitude of the gradient is simply the sum of these two gradients.

| | | |
|---|---|---|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

x filter

| | | |
|---|---|---|
| +1 | +2 | +1 |
| 0 | 0 | 0 |
| -1 | -2 | -1 |

y filter

Figure 3. Sobel operators.

The resulting image may contain some noise due to the illumination. To reduce it, average filter, then thresholding process are applied to that image. The mean value m which is used for the thresholding process of the image is calculated by the image histogram analysis. Finally, colors are reversed.

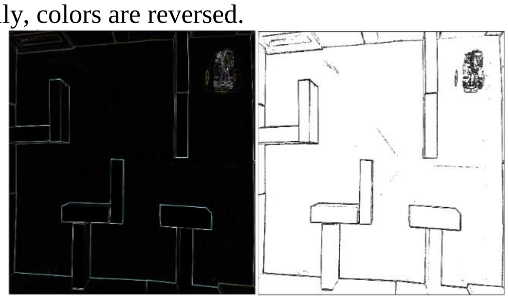

Figure 4. Enhancement in edge detection image.

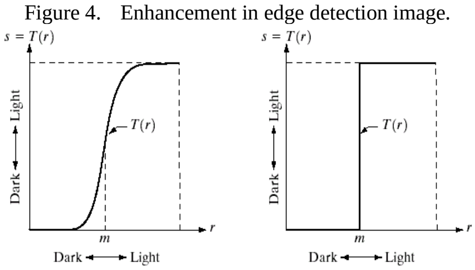


Figure 5. Thresholding process.

By reading those pixels approximating to the white color in the original image and considering the real borders of the environment in the original image, the obstacles are placed in the resulting image with a black color.

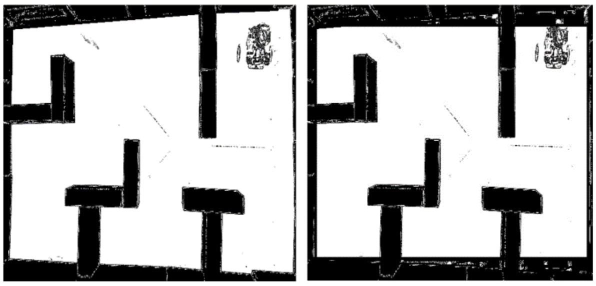

Figure 6. Obstacles placement and borders considering.

### *B. Obstacles Enlarging Method*

After detecting the real borders of the obstacles and environment and by considering the radius of the robot, it is wanted to enlarge the obstacles and environment borders from all the sides by the radius of the robot in order to convert in advance the path planning of a WMR into a path planning of a point robot. To accomplish this task, we proposed a special method based on dividing each obstacle into vertical and horizontal segments (each segment is found by counting pixels of each row and column of the image matrix from the left and top borders until the first change in pixel value and finish by the second change). If one row or column has more than one segment, these segments will be found by several pixel value changes and defined in pairs. To each segment found, a radius of the robot must be added to each side as shown in Fig. 7. The new borders of the obstacle and environment after the enlargement must be detected by using the obstacles detection method again.

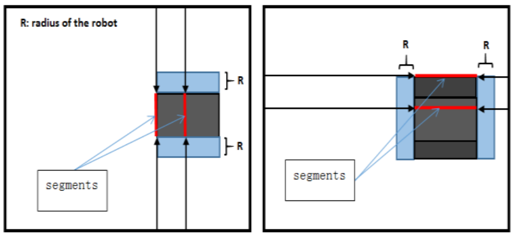


Figure 7. Segment method.

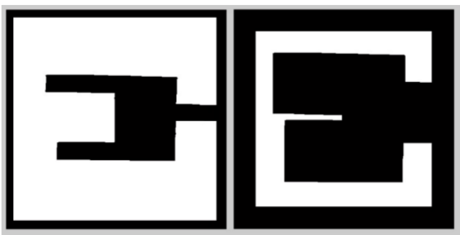

Figure 8. Obstacles enlarging.

## III. Motion Planning: The Grid Method

Grid method is developed in order to steer the WMR in unknown environments. The principle of the proposed method is to convert the global path between the start point and the goal point into a group of partial-paths (local paths) by finding interim goals. The main idea to achieve that is to divide the environment into small rectangle areas according to the dimensions and location of the obstacles, by this way we can convert the environment into a mini matrix. In the following sections, implementation steps using MATLAB show how this method works:

Based on the vision technology method introduced above, pink horizontal & vertical lines are established programmably crossing the new edges of the obstacles after enlarging process in order to divide the environment into a number of small rectangle-areas. There are basically two kinds of those rectangle areas as shown in Fig. 9: a- Forbidden rectangle areas: and they represent the unavailable areas of the obstacles' existence where it is forbidden for the robot to move. b- Available rectangle areas: and they stand for the possible areas where the robot can run freely without collision inside them.

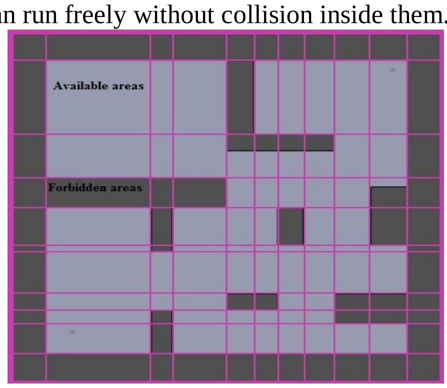


Figure 9. Divide the environment into various areas.

Then representing the environment as a mini matrix *mM* is wanted as shown in Fig. 10. The dimensions of *mM* are same as the number of the horizontal and vertical rectangle areas which appear in the grid. The elements of *mM* represent rectangle areas of the grid. The values of *mM* elements are given as following:

$$\begin{cases} 1 & \text{;for elements which represent the available areas.} \\ 2 & \text{;for elements which represent the start and goal areas.} \\ 0 & \text{;for elements which represent the unavailable areas.} \end{cases}$$

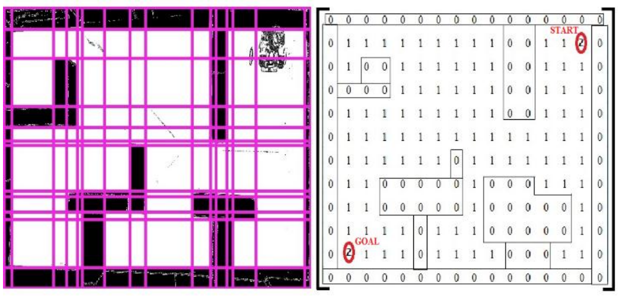


Figure 10. Mini matrix of an environment.

Modify the mini matrix: through the detection of the dead areas, the un-useful areas and the available areas, and then convert the un-useful areas into dead areas as shown in Fig. 11. Dead areas refer to the presence of the obstacles. The un-useful areas refer to the available areas where the robot will be trapped if it gets inside, whilst the available areas represent the possible spaces where robot can run without collision. This helps to solve the problem of local minima. Each of the grid rectangle areas are represented by an element in the mini-matrix and normally each element has 8 neighbours around it with the exception of peripheral elements. The principle to find the un-useful area depends on considering the zeros neighbours of each element in the mini matrix and element position also. For example, in Fig. 11 in the mini matrix the one in the orange circle is almost surrounded by zeros so it refers to un-useful area and in the new mini matrix it is converted into zero. Repeating the search process will detect more un-useful areas.

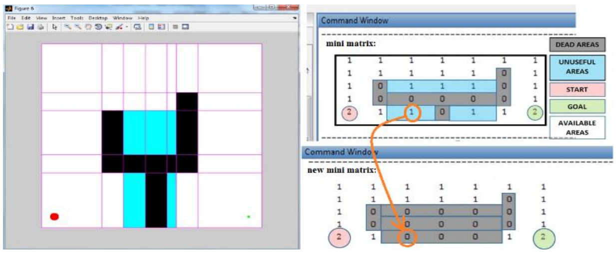

Figure 11. Convert un-useful areas into dead areas.

Generate the global paths: The global path is each of the possible paths which connect the area of the start point with the area of the goal point crossing the available areas. Generally, it consists of a set of local paths, where the local path is defined as the part of the global path that is located inside one available area. Here it must be noted that in the case in which the start and the goal points are in the same available area, there is only one optimal path (straight line) between them and it is by definition a local path. The process of generating the global paths consists of moving from the current position (the current rectangle-area) into the neighbouring available areas and repeating this until the target-area (which contains the goal point) is reached. As well as if the branching process leads to a trapped available rectangle area (i.e. the robot gets caught in a trap), in this case, the branching process from this area will be stopped and it is hopeless to reach to the target from this way. These areas must be ordered according to the distance to the goal point. The one with the shortest distance will be located on the left branch of the tree. The next selected area will also have neighbouring available areas which are to be ordered following the previously mentioned criteria. This process must be repeated until the goal is reached. Each set of the local paths connecting the original-area and the target-area represent a possible global path, but among them we have to choose the most suitable ones (the shortest). Fig. 12 shows the branching process, it starts from the original area 29th which has only two available rectangle areas 22nd and 30th (23rd is unavailable), and then repeats these criteria until reaching the target.

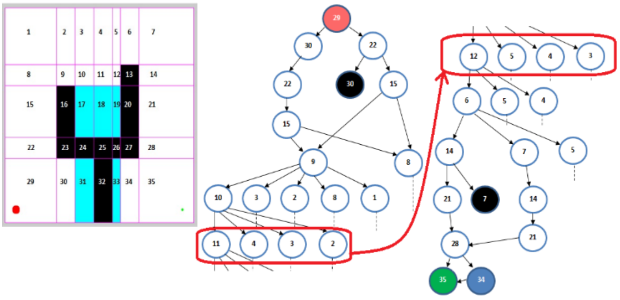

Figure 12. Branching tree diagram.

The method requires many times of paths generation. In addition, more obstacles will increase calculation time. So dynamic programming based on Dijkstra's algorithm using a priority queue is used to reduce the calculation time.

Entry-Point and Exit-Point: from the previous steps, each global path right now is consisting of a series of sequential available areas which link between the original-area and the target-area. In order to define the global path as a series of sequential points which link between the start-point and the goal-point, the local path inside each of those sequential available areas must be defined, and to achieve that the following rules were adopted to find the coordinates of two points (entry and exit points) for each of the available areas:

- The robot should enter the next available area from its entry-point coming from the previous available, area and should leave the current area from its exit-point passing to the next available area.
- The entry-point of the original-area is the start point. The exit-point of the target area is the goal point.
- The straight path between the entry-point and exit-point for one available area is a local path.
- The exit-point of each available area is the entry-point of the next available area, and finding its coordinates depends on the position of the next neighbouring available area in the global path. The shortest local path of one area to the horizontal or vertical neighbouring area is the shortest straight line from its entry-point to the border between the adjacent areas, and for the diagonal neighbouring areas, it is a straight line to the corner.
- Modify the global path: each global path right now is consisting of a series of sequential points which link between the start-point and the goal-point. In this step, the useless interim points of the series of sequential points composing the global path must be deleted. To make it clearer why would the robot have to go from the first point to the next one and then to the third one if it is able to go from the first to the third one directly without collisions.
- Choose the shortest possible global path by calculating the overall length of each set of the local paths which compose one global path.

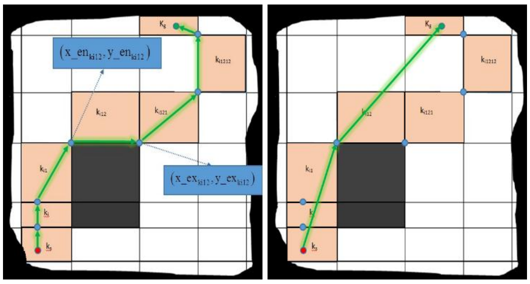

Figure 13. Entry & Exit-Points, modify the global path.

## IV. Cosine Switch Control of 3-WMR

Liu, Y., Li, L. and Tan, Y. proposed a Cosine Switch Control algorithm based on non-holonomic chained form system to steer a non-holonomic system in an empty environment (without obstacles). This method has proved that a 3-WMR system can be converted into 3-chained form system with two-inputs. The control inputs switch between two different modes to accomplish the cosine switch control. Cosine functions with unknown coefficients are taken as control inputs. In this work, this method is used to obtain the local path inside each of the available rectangle areas in order to steer the WMR from the entry-point to the exit-point of this available area.

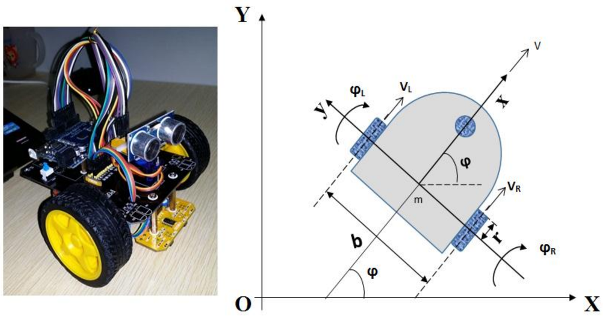


Figure 14. Wheeled mobile robot and its initial coordinate frame.

Fig. 14 shows the initial coordinate system for a differential-drive mobile robot, the initial coordinates of the robot are given by (*X*, *Y*), the angle that the velocity vector makes with the initial X-axis is given by $\varphi$. The robot position is given by the $(3\times 1)$ vector:

$$m = [X \quad Y \quad \varphi]' \tag{1}$$

The *n*-dimensional chained form system $z = [z_1\ z_2\ z_3 \dots z_n]$ with two inputs $[v_1\ v_2]$ can be described as follows:

$$\dot{z}_1 = v_1\ ,\ \dot{z}_2 = v_2\ ,\ \dot{z}_3 = z_2.v_1, \dot{z}_n = z_{n-1}.v_1 \tag{2}$$

3-WMR is 3-dimensional non-holonomic system (*n*=3). The non-holonomic constrains arise from constraining each wheel to roll without slipping during the movement. The kinematic model of 3-WMR, shown in Fig. 14, can be described as follows:

$$X^{\bullet} = \cos\varphi \cdot u_1\ ,\ Y^{\bullet} = \sin\varphi \cdot u_1\ ,\ \varphi^{\bullet} = u_2 \tag{3}$$

where $u_1$ refers to *V* the forward velocity of the robot centre which is perpendicular to the wheel axis, $u_2$ stands for the steering or angular velocity $\varphi^{\bullet}$. $[X^{\bullet}, Y^{\bullet}]$ are the forward velocity components.

The following equation describes the kinematic model of the robot, where the angular velocity of right wheel $\varphi_R^{\bullet}$ and left wheel $\varphi_L^{\bullet}$ are the two actual inputs of mobile robot:

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} r/2 & r/2 \\ r/b & -r/b \end{bmatrix} \cdot \begin{bmatrix} \varphi_R^{\bullet} \\ \varphi_L^{\bullet} \end{bmatrix} \tag{4}$$

where *b* is the distance between the two wheels and r is the radius of wheel. The steering velocity is caused by the differential of two wheels. The kinematic model of 3-WMR can be converted into a two-input chained form system with three states by coordinate conversion and input feedback conversion as:

$$\begin{cases} z_1 = X \\ z_2 = \tan\varphi \\ z_3 = Y \end{cases} \Rightarrow \begin{cases} v_1 = \cos\varphi \cdot u_1 \\ v_2 = \dfrac{1}{\cos^2\varphi} \cdot u_2 \end{cases} \tag{5}$$

When cosine switch control algorithm is applied to the 3-WMR, the forward velocity and steering velocity effect alternately. Therefore, motion of the car is divided into three steps: rotary motion, linear motion and rotary motion. This is due to the simply kinematic model and input conversion.

For *n*-dimensional chained form system, cosine switch control can steer it from a given initial configuration *z*(0) to a desired configuration *z(T)* through *2(n - 2) + 1* times of switch mostly. In our case *n=3* so the total time *T* is divided into *2(3 - 2) +1=3* intervals equally. The length of each time interval is:

$$\varepsilon = T / [2(\mathrm{n} - 2) + 1] \tag{6}$$

In odd time intervals, i.e., when $t \in [2i.\varepsilon, (2i+1).\varepsilon]$ , $(i = 0,1,2 \cdots n-2)$ ,the control inputs are represented by:

$$v_1 = 0\ ,\ v_2 = c_{2i+1}(1 - \cos wt) \tag{7}$$

where $c_{2i+1}$ are undetermined coefficients, w is the angular frequency and w $= 2\pi / \varepsilon$. Using inputs in Eq. (7) to steer system in Eq. (2), we can get the configuration of system at the end-point of odd time interval through integral operation.

$$\begin{aligned} z_1(t_{2i+1}) &= z_1(t_{2i}) \\ z_2(t_{2i+1}) &= c_{2i+1}.\varepsilon + z_2(t_{2i}) \\ z_3(t_{2i+1}) &= z_3(t_{2i}) \\ &\vdots \\ z_n(t_{2i+1}) &= z_n(t_{2i}) \end{aligned} \tag{8}$$

In even time intervals, i.e., when $t \in [(2j+1).\varepsilon, (2j+2).\varepsilon]$, $(j = 0,1,2 \cdots n-3)$, the control inputs are represented by:

$$v_1 = c_{2j+2}(1-\cos wt)\ ,\ v_2 = 0 \tag{9}$$

where $c_{2j+2}$ are undetermined coefficients. Using inputs in Eq. (9) to steer system in Eq. (2), we can get the configuration of system at the end-point of even time interval through integral operation.

$$\begin{aligned} z_1(t_{2j+2}) &= c_{2j+2}.\varepsilon + z_1(t_{2j+1}) \\ z_2(t_{2j+2}) &= z_2(t_{2j+1}) \\ z_3(t_{2j+2}) &= c_{2j+2}.z_2(t_{2j+1}).\varepsilon + z_3(t_{2j+1}) \\ &\vdots \\ z_n(t_{2j+2}) &= \sum_{k=1}^{n-2} \frac{(c_{2j+2}.\varepsilon)^k \cdot z_{n-k}(t_{2j+1})}{k!} + z_n(t_{2j+1}) \end{aligned} \tag{10}$$

The final configuration at $T$ can be calculated by iterative operation via Eq. (8) and Eq. (10):

$$\begin{aligned} z_1(T) &= \sum_{j=0}^{n-3} c_{2j+2} \cdot \varepsilon + z_1(0) \\ z_2(T) &= \sum_{i=0}^{n-2} c_{2i+1} \cdot \varepsilon + z_2(0) \\ z_3(T) &= \sum_{i=0}^{n-3} \left( \sum_{j=i}^{n-3} c_{2j+2} \cdot \varepsilon \right) \cdot c_{2i+1} \cdot \varepsilon + \sum_{j=0}^{n-3} c_{2j+2} \cdot \varepsilon \cdot z_2(0) + z_3(0) \\ &\vdots \\ z_n(T) &= \sum_{i=0}^{n-3} \frac{\left( \sum_{j=i}^{n-3} c_{2i+2} \cdot \varepsilon \right)^{n-2}}{(n-2)!} \cdot c_{2i+1} \cdot \varepsilon + \sum_{k=1}^{n-2} \frac{\left( \sum_{j=0}^{n-3} c_{2j+2} \cdot \varepsilon \right)^{k}}{k!} \cdot z_{n-k}(0) + z_n(0) \end{aligned} \tag{11}$$

Specify a set of coefficients $c_{2j+2}$, and they must be satisfied with Eq. (12):

$$\sum_{j=0}^{n-3} c_{2j+2} = \frac{z_1(T) - z_1(0)}{\varepsilon} \tag{12}$$

Then we can get the remaining coefficients $c_{2i+1}$ by substituting initial state $z(0)$, final state $z(T)$ and total time $T$ into Eq. (11).

## V. Experimental Work: Simulation and Results

Experiments of different cases of environments have been performed to achieve the motion planning methods of the 3-WMR. The experiments showed that the implementation of motion planning based on the proposed image processing and grid methods works with the available robot hardware. So that it must ensure the safe navigation through the obstacles in the environments and perform the simulation displayed. Fig. 15 shows a colored snapshot of a real environment.

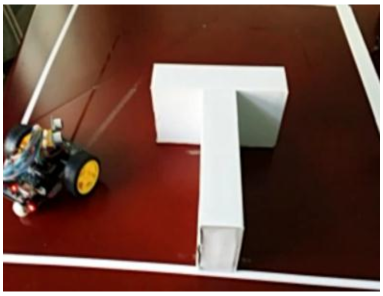

Figure 15. Colored snapshot of a real environment.

Applying the image processing method: using "first order derivative function", the real borders of the environment were detected. By performing the "segment method", the obstacles were enlarged from all the sides by a radius of the robot R=7.5 cm. As a result of applying "first order derivative function" again, the new borders of the obstacles were detected:

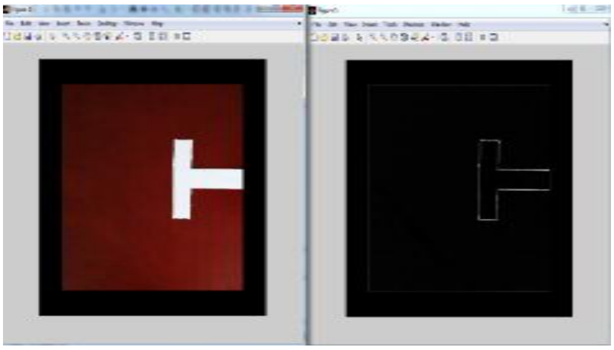

Figure 16. Detect the real borders.

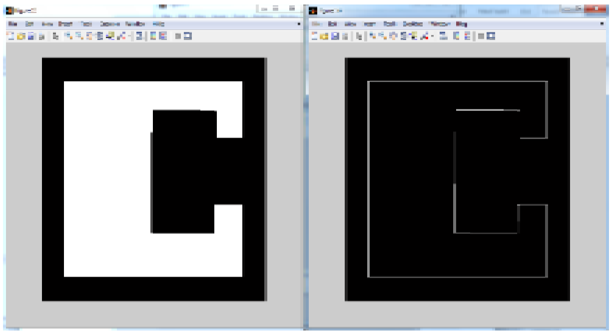

Figure 17. Obstacles enlarging and detect the new borders.

The grid lines were established and the summarized mini-matrix was defined as a matrix $[7 \times 5]$:

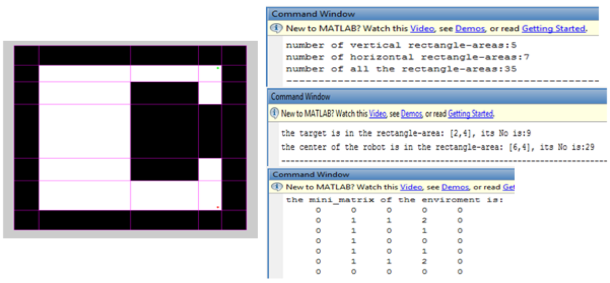


Figure 18. The mini-matrix.

In this step, the summarized mini-matrix was shortened: both of the 20th area and the 41st area are not a target area (they don't contain the goal point) as well as both of them seem as a trap (6 neighbors of each of them are zeros) so they are not useful, and it is wanted to convert them into dead areas.

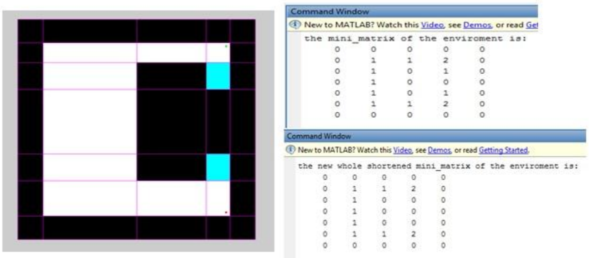

Figure 19. Modify the mini-matrix.

Generate the possible global paths and represent them by a tree: The process of generating the global paths consisted of moving from the original area "48th" into the neighboring available areas and repeating this until the target-area "13th" is reached.

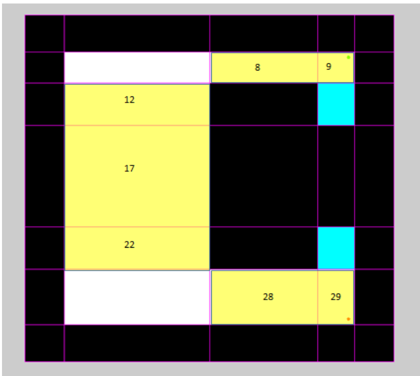


Figure 20. A possible global path.

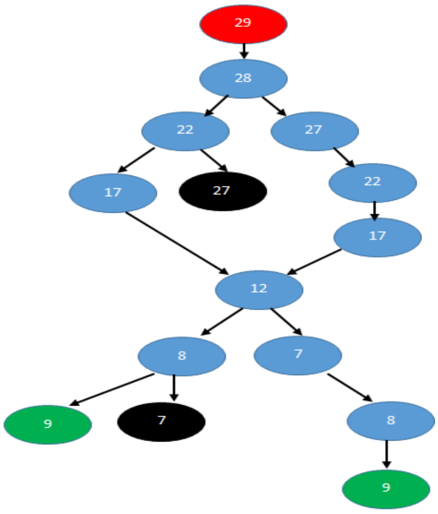


Figure 21. Branching tree of possible paths.

Generate the possible global paths as a series of sequential points which link between the start-point and the goal-point by finding the coordinates of two points (entry-point and exit-point) for each of the available areas.

Modify the result path by deleting the useless interim points of the series of sequential points composing the global path.

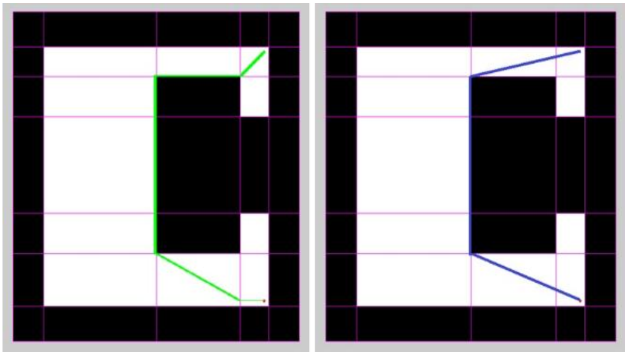

Figure 22. The best path.

Finally, the cosine switch control algorithm is applied to the 3-WMR 3-times to realize the motion planning. The 3-WMR should travel form a configuration to the next configuration and the time required to finish this task is T=9 seconds. So the total time required to finish the whole motion is 27 seconds. The simulation results and the real movements are shown in Fig. 23 and Fig. 24.

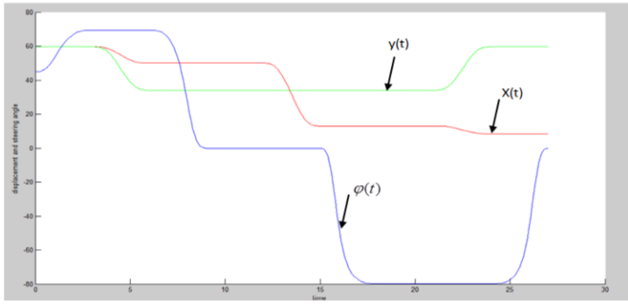


Figure 23. Simulation results: displacement and steering angle versus time.

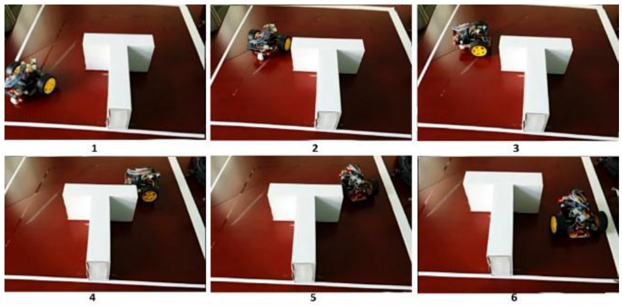


Figure 24. Real robot movements.

## VI. Conclusion

The current work deals with special kinds of rectangle environments which may contain regular-rectangle, unknown-position and unknown-dimensions static obstacles. It is worth pointing out that it could be developed to include more dynamic environments with different kinds of obstacles by modifying the generated path based on sensed information from the environment. As well the methods, which are developed in this work to realize the motion planning for wheeled robot, are innovative especially the grid method and the segment method More than that, these methods save time due to the truth that the possible paths are generated programmatically and then the robot must follow the generated path. As well by converting the environment into a mini matrix and modifying it, these methods are able to solve the problem of local minima “or the trap” easily. The practical experiences have shown that the methods work well: the vertical and horizontal edges of the obstacles are highlighted using first order derivative theory and enlarged by a radius of the robot from all sides using segment theory, and the robot has been able to move in this kind of environments and reach the goal without collision. And to reduce the calculation time of paths generation, it is worthwhile to use the dynamic programming based on Dijkstra's algorithm with a priority queue.

As for the current research, the main limitations are the studied kinds of obstacles. To solve these limitations, the following points are suggested for further developments of the methods which have been used:

- The proposed grid method can be developed by adding the fuzzy concepts to cover more complex environments which may contain static-obstacles in different forms and shapes (not only regular-rectangle). And the new method will be called fuzzy-grid method. The areas will be represented in the mini matrix with a fuzzy function (such as the ratio of the white pixels to the all pixels for each square area).
- Develop the fuzzy grid method into dynamic fuzzy grid method to cover the case of presence the moving obstacles in the environment. In this case, the fuzzy values of the mini matrix will not be constant, and they will change upon the time based on obstacles movements.
- Implementation of the obstacle detection in a three dimensions’ environment by adding another camera to solve the problem of the shadow in case of using high obstacles.

## ACKNOWLEDGMENT

This work is supported by the National Key R&D Plan, key projects of green building and building industrialization (No.2017YFC0703903), Wuhan High-Tech Development Project Foundation (NO. 2017010201010138) and Innovation experimental program for national college students (NO. 20171049704011)

## REFERENCES

[1] M. Murray, and S. Sastry, “Nonholonomic motion planning: steering using sinusoids,” *IEEE Transactions on Automatic Control*, vol. 38, no. 5, pp. 700-716, 1993.

[2] M. Murray and S. Sastry, “Steering nonholonomic systems in chained form,” in *Proc. the 30th Conference on Decision and Control*, Brighton, England, December 1991, pp. 1121-1126.

[3] L. Bushnell, D. Tilbury, and S. Sastry, “Steering three-input chained form nonholonomic systems using sinusoids: the fire truck example,” in *Proc. the European Control Conference*, Groningen, The Netherlands, June 28-July 1, 1993.

[4] D. Tilbury and A. Chelouah, “Steering a three-input nonholonomic system using multi-rate controls,” in *Proc. the European Control Conference, Groningen*, The Netherlands, June 28-July 1, 1993.

[5] L. Sheng, M. Guoliang, and C. Qingwei, “Nonholonomic motion planning: Steering using bang-bang control,” in *Proc. the 2004 American Control Conference*, Boston, Massachusetts, June 30 - July 2, 2004, pp. 4653-4656.

[6] Y. Liu, L. Li, and Y. Tan, “Steering nonholonomic systems with cosine switch control,” in *Proc. the Eighth International Conference on Intelligent Systems and Knowledge Engineering*, Shenzhen, China, Nov. 2013 (ISKE 2013), pp. 167-176

[7] M. Deng, A. Inoue, and K. Sekiguchi, “Parking control of a two wheeled mobile robot,” in *Proc. the 2007 IEEE International Conference on Mechatronics and Automation*, 2007, pp. 539 – 544.

[8] T. Urakubo, K. Okuma, and Y. Tada, “Feedback control of a two wheeled mobile robot with obstacle avoidance using potential functions,” in *Proc. the IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2004)*, 2004, vol. 3, pp. 2428 – 2433.

[9] S. Ge and Y. Cui, “Dynamic motion planning for mobile robots using potential field method,” *Autonomous Robots*, vol. 13, no. 3, pp. 207-222, 2002.

[10] C. Rusu and I. Birou, “Obstacle avoidance fuzzy system for mobile robot with IR sensors,” in *Proc. 10th International Conference on Development and Application Systems*, Suceava, Romania, May 27-29, 2010, pp. 25-29.

[11] O. Yuki, “Corridor navigation of a mobile robot using a camera and Sensors—Multi-agent approach,” M.Sc. Thesis, University of Georgia.

[12] L. Wei, M. Chenyu, and F. Wahl, “Neuro-fuzzy system architecture for behavior-based control of a mobile robot in unknown environments,” *ELSEVIER Fuzzy Sets and Systems*, vol. 87, no. 2, pp. 133-140.

[13] P. Rusu, E. Petriu, and A. Cornell, “Behavior-based neuro-fuzzy controller for mobile robot navigation,” in *Proc. the 19th IEEE Instrumentation and Measurement Technology Conference, IMTC/2002*, vol. 2, pp. 1617 – 1622.

**Essa Alghannam** was born in December 1987 in Syria. He received his master’s degree in mechatronics engineering from Wuhan University of Technology, China in 2015. He received his bachelor degree in mechatronics engineering from Tishreen University, Syria, in 2010.

From 2010 to 2012 he has been a research associate teacher at the school of Mechatronics Engineering, Tishreen University, Latakia, Syria. He is currently a Ph.D. student at the School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan, China. His research interests include robotics, robot vision, and human interface.

**Lu Hong** was born in April 1965 in China. She received Ph.D. in mechanical engineering from Wuhan University of Technology, China, in 2000. She received M.S. in mechanical engineering from Huazhong University of Science and Technology, China, in 1990. She received B.S. in mechanical manufacturing and its automation from Huazhong University of Science and Technology, China, 1987. From 2004-present, she has been a professor and a PhD tutor at the School Of Mechanical And Electronic Engineering, Wuhan University of Technology, China. Her research interests include mechanical CAD / CAM, CNC basic theory and manufacturing. Prof. Lu is a member of the robot branch of the Chinese society of mechanical engineering (CMES), and universities of China, and the director of the Mechanical Engineering Society of Hubei province (HBMES).

**Yang Zhen**, as the corresponding author of this paper, was born in May 1997 in China. He is now studying for the bachelor degree in mechanical engineering in Wuhan University of Technology. He is the leader of the Innovation Experimental Program for National College Students “Automatic Detection and Sorting System for Roller End Surface Quality Based on Image Recognition".

**Oula Esmail** was born in April 1984 in Syria. She received her bachelor degree in mechatronics engineering from Tishreen University, Syria, in 2010. She is currently a master student at the School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan, China. Her research interests include robotics, robot vision, and human interface.