Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

A Method for Collision-free Guidance of Mobile Robots in Unknown Dynamic Environments with Moving Obstacles

Yang Zhang
School of Electrical Engineering and Telecommunications
The University of New South Wales
Sydney, Australia
z5134422@ad.unsw.edu.au

***Abstract*— In this paper, a collision-free navigation algorithm for mobile robots is proposed. The algorithm is applied to dynamic environments. Shapes of moving obstacles may be non-convex, speed and angular velocity are variable. The algorithm is a sliding mode control law, which switches between two modes, namely, target approaching mode and obstacle avoidance mode. Moreover, a rigorous analysis of the proposed algorithm is provided and the performance is demonstrated with simulation results. It is a really practical algorithm as there is not too many assumptions for the obstacles compared with others.**

***Index Terms*— navigation algorithm; obstacle avoidance; dynamic environments; non-convex constraints; speed and angular velocity.**

I. Introduction

Planning path for mobile robots to avoid obstacles effectively is always an important problem in the robots controlling field. A lot of researches have been carried out in order to solve the problem, e.g. [1], [2]. Although these approaches are all giving solutions to the problem of finding trajectory for the robots, they are quite different from the assumptions they gave, to the model they used to describe the system, to the sensors they chose. Thus, there are many classifications according to different standards. Specifically, navigation approaches can be divided into two groups, which are global path planning and local reactive obstacle avoidance, respectively. Global path planning is based on the known environment which serves as *a priori* information to build a complete map so that the best route to the target can be found [3]. The other one is using sensors to get part information of an unknown environment first, and then determine the trajectory of the robot in a cluttered environment [4]. Alternatively, the two approaches can be combined together, such as [5]. For the status of obstacles, some approaches like the dynamic windows approach [6], the curvature velocity method [7] and the lane curvature method [8] are usually applied to static obstacles. It is more difficult to solve the problem of avoiding dynamic obstacles. Although there are a number of strategies have been proposed, like velocity obstacles [9][10], collision cone approach [11], and inevitable collision states [12][13]. They are not efficient for environment with rapid change.

In order to solve these problems, a biologically inspired reactive algorithm for dynamic environments with moving obstacles was proposed in 2013 [14]. It is an approach based on Equiangular Navigation Guidance (ENG) laws [15][16], where they described the mobile robot with non-holonomic model and gave a strict constraint to the movement of obstacles. The proposed algorithm can successfully guide the robot to the target point with a number of assumptions. Although the algorithm is realizable in some situations, it is not always possible in practice. It is because that on one hand the shapes of obstacles are limited too strictly and on other hand obstacles can only move along a straight line with a constant speed.

Therefore, we propose a navigation approach in this paper, which is based on the biologically inspired algorithm [14]. It is a speed and angular velocity based on-line reactive approach. The method can be applied to dynamic environments with moving obstacles. Without loss of generality, obstacles shapes are restricted with non-convex constraints, speed and angular velocity are changeable within a certain range. Our navigation approach is a sliding mode control law, which switches between two modes, i.e. target approaching mode and obstacle avoidance mode [17][18], respectively. Compared with the previous works, the constraints of the obstacles are easier to be satisfied. Furthermore, our proposed algorithm is computationally efficient, unlike those high-level decision-making algorithm, such as [19]. Hence, the significance for practice of our proposed method is obvious.

The remaining parts of the paper are organized as follows. Section II presents problem statements including system description and some assumptions. Section III introduces our navigation algorithm. Computer simulations and a comparison with the biologically inspired algorithm proposed by Andrey et al. are given in Section IV. Finally, section V presents brief conclusions.

*This work was supported by the Australian Research Council. Also, this work received funding from the Australian Government, via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

II. Problem Statement

In this section, we consider a planar vehicle or wheeled mobile robot modeled as a unicycle. For the controlled object, the control inputs are angular velocity and speed, both limited by given constants. The kinematics of the considered vehicle is described as follows:

$$\dot{x}(t) = V_R(t)\cos\theta, x(0) = x_0, \tag{1}$$

$$\dot{y}(t) = V_R(t)\sin\theta, y(0) = y_0, \tag{2}$$

$$\dot{\theta}(t) = u_R(t)\cos\theta, \theta(0) = \theta_0, \tag{3}$$

where

$$V_R(t) \in [0, V_{max}], u_R(t) \in [-U_{max}, U_{max}]. \tag{4}$$

Here, $[x(t), y(t)]$ is the vector of the vehicle's Cartesian coordinates and $\theta(t)$ gives its orientation. $V_R(t)$ and $u_R(t)$ are the speed and angular velocity, respectively. The angle $\theta(t)$ is measured in the counter-clockwise direction. The maximum speed V_{max} and the maximum angular velocity U_{max} are given.

Now, describe the vector of the robot's coordinates with

$$c_R(t) := [x(t), y(t)]. \tag{5}$$

In order to introduce our proposed navigation approach, we set a scene example with essential elements. We definite a initial point ***I*** and a target point ***T***, which are always stationary. There are also several random moving obstacles $\Diamond_1(t), \Diamond_2(t), ..., \Diamond_k(t)$ in the plane. Each obstacle is closed bounded planar set with non-convex constraints. An illustration is given in Fig. 1(a). All points on the boundaries of obstacles should be seen by the robot, or it will not be considered, e.g. Fig. 1(b). Narrow U-trap situation is not considered because the width of the narrow space may be not enough for the robot to move and the robot would be trapped, e.g. Fig. 1(c). Moreover, collisions between obstacles are not considered.

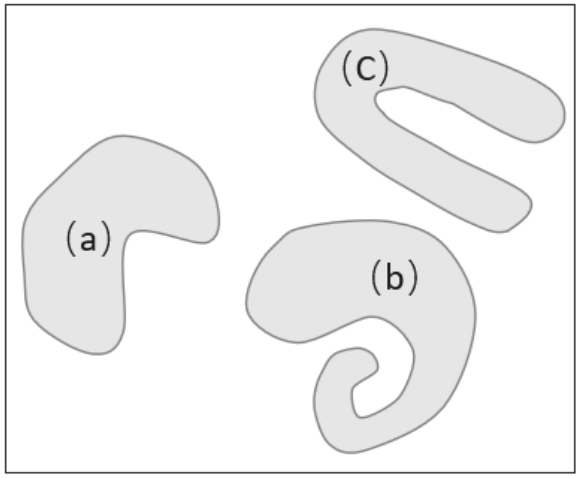


Fig. 1. Illustration of obstacles.

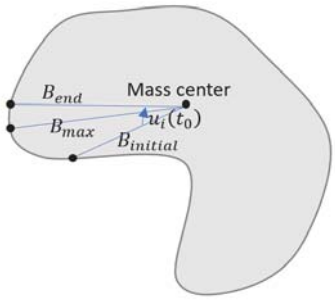


Fig. 2. The illustration of the constraints of motion of the obstacles to avoid collision.

The objective of our algorithm is to drive the vehicle to the target point ***T*** while avoiding collisions with all moving obstacles in the plane. The current distance between the vehicle and the border of the obstacle $\Diamond_i$ defined as

$$d_i(t) := \min_{r(t)\in\Diamond_i(t)} \|r(t) - c_R(t-1)\|. \tag{6}$$

Here, $\|\cdot\|$ denotes the standard Euclidean vector norm. We suppose that the point $r_{min}(t) = [x_{min}(t), y_{min}(t)]$ makes the minimum value achieved.

We define a minimum distance between the robot current position and the nearest obstacle $\Diamond_i$ as

$$d_{\min}(t) := \min_{r(t)\in\Diamond_i(t)} \|r(t) - c_R(t-1)\|. \tag{7}$$

The obstacle $\Diamond_i$ is moving with the velocity $v_i(t)$ and revolving around the mass center with the angular velocity $u_i(t)$. For obstacle $\Diamond_i$, we suppose that the shape and the mass center of each obstacle are known, the motion of the obstacles should be restricted rigorously, as $v_i(t)$ and $u_i(t)$ are contributing to the value of $d_{min}(t)$.

We define the distance between the mass center of the obstacle and the initial point on the obstacle's border as $B_{initial}$, and the distance between the mass center of the obstacle and the end point on the obstacle's border with the angular velocity $u_i(t)$ as B_{end}. The maximum value of distance between the initial point and the end point is B_{max}. We define $\Delta distance = B_{max} - B_{initial}$, see Fig. 2.

The constraints of velocities v_i and angular velocity u_i are:

$$0 < \|v_i(t)\| < \|v_R(t)\| < V_{max}, \tag{8}$$

$$-U_{max} < u_i(t) < U_{max}, \tag{9}$$

$$\Delta distance < \|v_R(t)\| - \|v_i(t)\|. \tag{10}$$

For some $V_R > 0$ and all $\Diamond_i$, t. the collision avoidance may be impossible if (6)-(8) does not hold.

III. Navigation Algorithm

In this section, we give a detailed description of our proposed collision-free navigation algorithm. We suppose that there are some moving obstacles on the plane, the robot starting moving from a initial point ***I*** and aiming to get to a target point ***T***. Fig. 3 is a block diagram which shows how the algorithm works.

There are three steps for the robot to get the coordinate of every single second when the robot is avoiding an obstacle, see the gray dotted box in Fig. 3. Then we will describe these three steps separately.

A. Decide the direction.

We define a line $L_T(t_0)$ and a line $L_M(t_0)$, which are the segment between the robot and the target ***T*** and segment between the robot and the mass center of obstacle $\Diamond_i$, respectively. Then, measure the angle of line $L_M(t_0)$ from line $L_T(t_0)$, which is denoted as $\alpha(t_0)$.

$$\alpha(t_0) \in (-\frac{\pi}{2}, \frac{\pi}{2}). \tag{11}$$

Moreover, we assume that the counter-clockwise direction is the positive direction. If $\alpha(t_0) \in \left[0, \frac{\pi}{2}\right)$, the robot will avoid the obstacle in the positive direction, denoted as '+', e.g. Fig. 4(a). If $\alpha(t_0) \in \left[-\frac{\pi}{2}, 0\right)$, the robot will avoid the obstacle in the negative direction, denoted as '-', e.g. Fig. 4(b).

B. Get the nearest point on the boundary of the obstacle.

The point on the boundary of the obstacle $\Diamond_i$ is defined as $r_{min}(t_0) = [x_{min}(t_0), y_{min}(t_0)]$, which satisfies:

$$\|r_{min}(t_0) - c_R(t_0 - 1)\| = d_{min}(t_0).$$

We modify the $r_{min}(t_0)$ to $r^*_{min}(t_0)$. Let α_0 be a given angle. The two intersections with the obstacle $\Diamond_i$ of two rays from the robot are $r^+_{min}(t_0)$ and $r^-_{min}(t_0)$, α^+_0 and α^-_0 are the positive intersection angle and negative intersection angle,

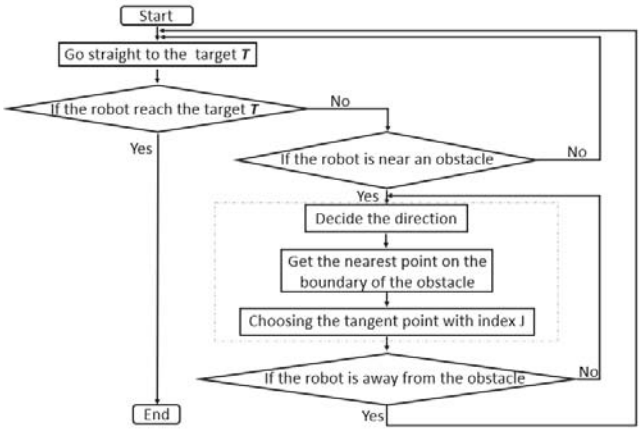


Fig. 3. Block diagram of the algorithm.

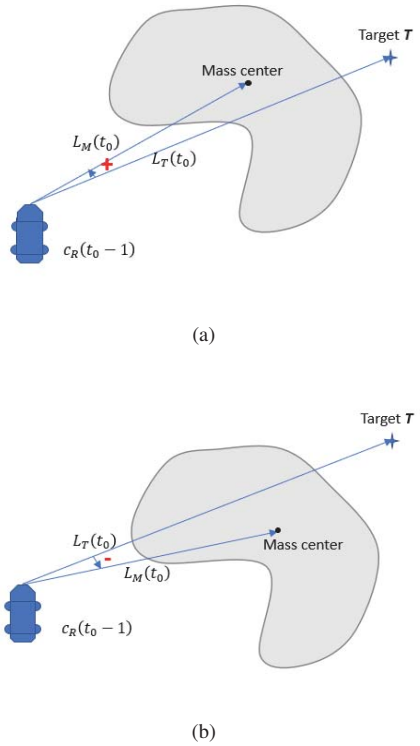


Fig. 4. The position relationship between $L_M(t_0)$ and $L_T(t_0)$.

respectively, e.g. Fig. 5. With the direction decided in the last step, we can get $r^*_{min}(t_0)$ equals to $r^+_{min}(t_0)$ or $r^-_{min}(t_0)$, given as $r^*_{min}(t_0) = [x^*_{min}(t_0), y^*_{min}(t_0)]$. In order to follow the boundaries of obstacles precisely, α_0 should be a small one. It is a constant value which depends on the shape of the obstacles.

C. Choosing the tangent point with index.

We define a forecast circle O_{t_0},

$$(x - x^*_{min}(t_0))^2 + (y - y^*_{min}(t_0))^2 = r_i(t_0)^2. \tag{12}$$

The center of O_{t_0} is $r^*_{min}(t_0)$ and the radius $r_i(t_0)$ is $\|v_i(t_0)\|$. There are two tangent lines for the forecast circle O_{t_0} from $c_R(t_0 - 1)$, the angles $\alpha^1_i(t_0)$ and $\alpha^2_i(t_0)$ are angles of the tangent lines measured from horizontal line in counter-clockwise direction, e.g. Fig. 6. The two intersections are called $p_1(t_0)$ and $p_2(t_0)$, respectively. Here, we define γ,

$$\gamma = \sin^{-1} \frac{\|v_i(t_0)\|}{d_{min}(t_0)}. \tag{13}$$

Then, we assume a pair of forecast vectors $l^1_i(t_0)$ and $l^2_i(t_0)$, given by

$$l^1_i(t_0) := \frac{\|v_i(t_0)\|}{\tan\gamma} \left[\cos\left(\alpha^1_i(t_0)\right), \sin\left(\alpha^1_i(t_0)\right)\right], \tag{14}$$

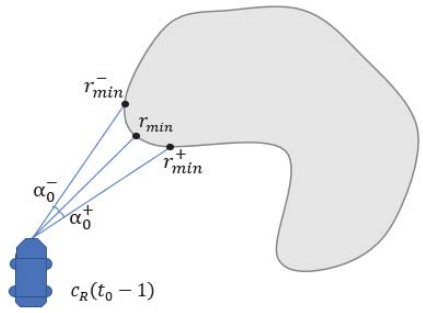


Fig. 5. The illustration of intersection angles.

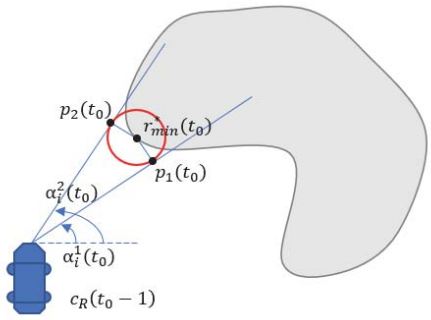


Fig. 6. The coordinate of the robot when time t_0 equals to $p_J(t_0)$.

$$l_i^2(t_0) := \frac{\|v_i(t_0)\|}{\tan\gamma}\left[\cos\left(\alpha_i^2(t_0)\right), \sin\left(\alpha_i^2(t_0)\right)\right]. \tag{15}$$

For moving obstacle $\Diamond_i$, $\beta_i^1(t_0)$ is the angle between $l_i^1(t_0)$ and $v_R(t_0-1)$ measured from $v_R(t_0-1)$ and $\beta_i^2(t_0)$ is the angle between $l_i^2(t_0)$ and $v_R(t_0-1)$ measured from $v_R(t_0-1)$.

We denote an index $J(t_0)$ by comparing the absolute value of $\beta_i^1(t_0)$ and $\beta_i^2(t_0)$, such that $J(t_0)=1$ when $\left|l_i^1(t_0)\right| \leq \left|l_i^2(t_0)\right|$, $J(t_0)=2$ when $\left|l_i^1(t_0)\right| > \left|l_i^2(t_0)\right|$.

Now, we introduce the following function:

$$f\left(l_J(t_0), v_R(t_0-1)\right)) = \begin{cases} 0 & \beta_J(t_0)=0 \\ 1 & 0<\beta_J(t_0)\leq\pi \\ -1 & -\pi<\beta_J(t_0)<0 \end{cases} \tag{16}$$

The navigation law is proposed as:

$$\begin{aligned} u_R(t) &= -U_{max} f\left(l_J(t), v_R(t-1)\right); \\ V_R(t) &= \|l_J(t)\|. \end{aligned} \tag{17}$$

Obviously, $d_{min}^2(t) = \|v_i(t)\|^2 + |l_J(t)\|^2$. As $d_{min}(t) < V_{max}, V_R(t) = \|l_J(t)\| < V_{max}$, satisfy constraints (2). Specifically, ***M1*** is target approaching mode. The robot goes straight to the target ***T*** with maximum speed:

$$u_R(t) = 0; V_R(t) = V_{max}. \tag{18}$$

M2 is obstacle avoidance mode. The motion of robot follows law (15). The robot starts to move with ***M1*** and switches to ***M2*** when the distance between the robot and the nearest obstacle reduces to d_{min}. When the robot is away from the obstacle and orientates towards the target, it will switch to ***M1***. Switching between these two modes until the robot reach the target ***T***. Furthermore, the robot is away from the obstacle means that the distance between the robot and the obstacle is more than d_{min}, line $L_T(t)$ does not across the obstacle.

So far, we have introduced our collision-free navigation algorithm for mobile robots which can be applied to dynamic environment with a number of moving obstacles.

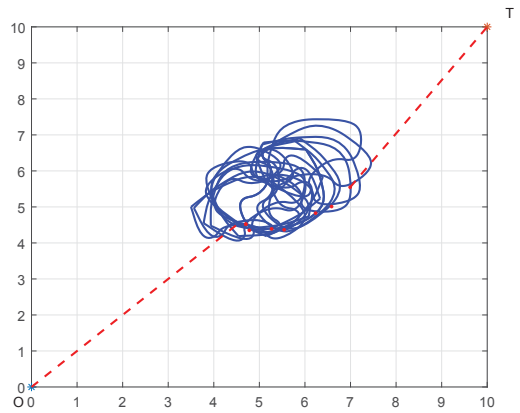
IV. Computer Simulation

In this section, we present computer simulation results of a wheeled mobile robot navigating in dynamic environments with moving obstacles. The simulations are performed with Matlab. Our aim is to show the simulation results of our proposed algorithm and compare it with others.

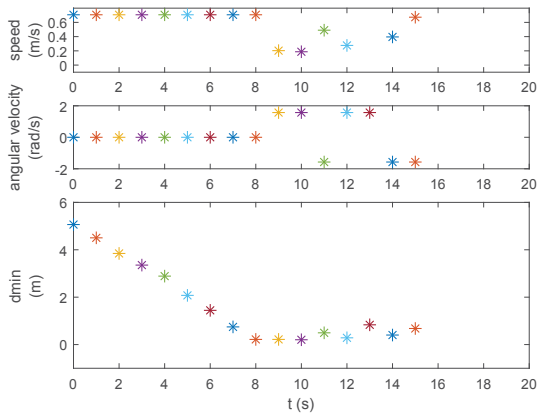
First, we examine our approach with the simplest case: the environment with only one moving obstacle. The result is shown in Fig. 7. Fig. 7(a) shows the path of the moving obstacle and the path of the robot from the initial point ***I*** to the target point ***T***. The speed and angular velocity of the robot and the minimum distance between the robot and the nearest obstacle are shown in Fig. 7(b). We can see that the robot is moving towards the target point ***T*** until it is near the moving obstacle, and the robot is close to the moving obstacle when it eludes the moving obstacle. It means that the robot could spend less time reaching the destination.

Fig. 8 shows a more challenging situation with six obstacles. As we set, three of the obstacles are in the robot's way, and others are not. The robot would only pay attention to those obstacles which are in its way and take action to the nearest one each time, although the robot can see all the moving obstacles appeared in its sight. We can confirm above information from the simulation results. Fig. 8(a) shows the paths of both the robot and the obstacles. Fig. 8(b) shows the speed of the robot $v_R(t)$, angular velocity of the robot $u_R(t)$ and the shortest distance between the robot and the nearest obstacle $d_{min}(t)$. Fig. 8(c) shows the speed and angular velocity of the obstacles, $v_i(t)$ and $u_i(t)$. Obviously, the robot can reach the target successfully without any collision with obstacles as implementing our proposed algorithm.

Moreover, we compare our proposed algorithm with the biologically inspired algorithm [14]. A simulation result of the biologically inspired algorithm is shown in Fig. 9. For implementing the biologically inspired algorithm, shapes of obstacles must be circle and obstacles can only move with a constant speed without changing directions. As mentioned, our proposed algorithm can be applied to more complex environments, for both shape and motion of obstacles. Here, we try a simple situation for comparison these two algorithms,



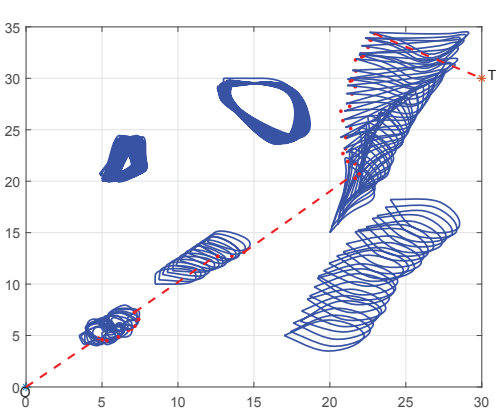
(a)



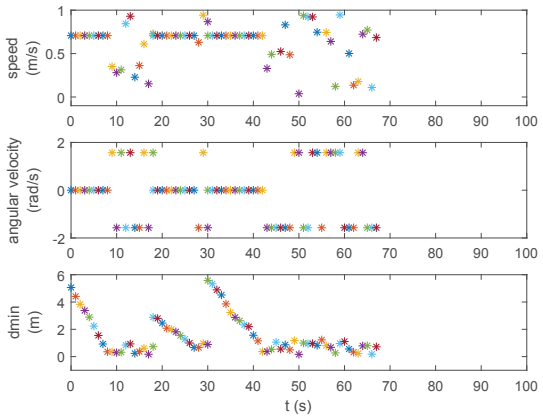
(b)

Fig. 7. Simulation results with one moving obstacle.

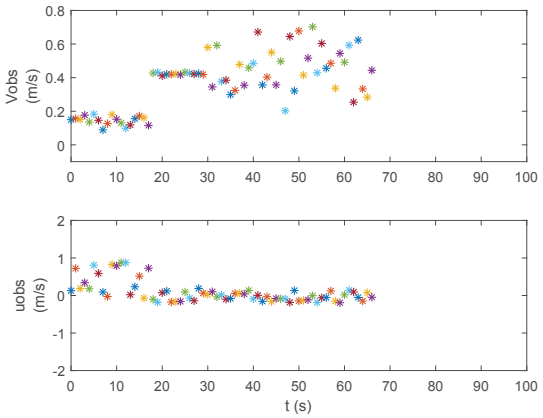
see Fig. 10. Fig. 10(a) and Fig. 10(b) show the paths of the robot implementing the biologically inspired algorithm and our proposed algorithm, respectively. It is cleat that if the robot moves to the target directly, the less time the robot spends on avoiding collisions with the obstacles, the shorter the path length is. Obviously, the robot performs better on following the boundaries of obstacles when implementing our proposed algorithm. Fig. 10(c) and Fig. 10(d) illustrate that clearly with showing the whole distance of paths implementing the two algorithms, the experimental results are 15.6793(m) and 14.9150(m), respectively. Obviously, the path implementing our proposed algorithm is shorter.



(a)



(b)



(c)

Fig. 8. Simulation results with six moving obstacles.

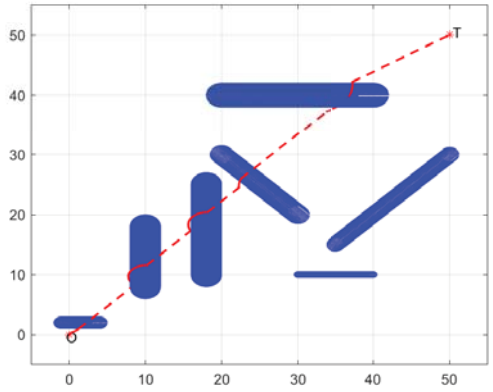


Fig. 9. Example of implementing the biologically inspired algorithm.

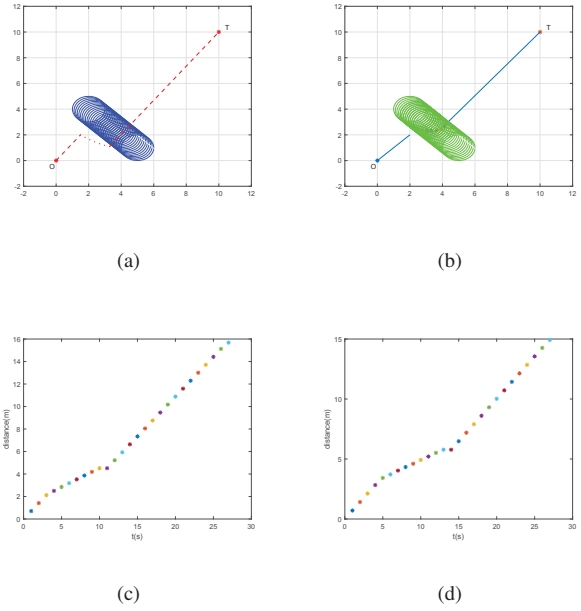


Fig. 10. Comparison of the two algorithms.

V. Conclusion

We have proposed a collision-free navigation approach for mobile robots. It can be applied to dynamic environments with moving obstacles. The shapes of moving obstacles are restricted to non-convex constraints, speed and angular velocity of them are changeable with some constraints. We also gave a detailed mathematical analysis of both the algorithm and the constraints. The performance is demonstrated by simulation results. Also, compared with the previous works, the path length for the robot implementing our proposed algorithm is shorter, the range of application is wider. Moreover, the computational efficiency is better than those algorithms which can achieve the same goals. Thus, it is really a practical method.

References

[1] M. Hoy, A. S. Matveev and A. V. Savkin, "Algorithms for Collision-free Navigation of Mobile Robots in Complex Cluttered Environments: A survey", Robotica, vol. 33, pp. 463-497, 2015.

[2] N. Sariff and N. Buniyamin, "An Overview of Autonomous Mobile Robot Path Planning Algorithms", 4th Student Conf. on Research and Development, Selangor, Malaysia, 2006.

[3] Y. H. Liu and S. Arimoto, "Path Planning Using a Tangent Graph for Mobile Robots Among Polygonal and Curved Obstacles: Communication", the International Journal of Robotics Research, vol. 11, pp. 376-382, 1992.

[4] A. V. Savkin, A. S. Matveev, M. Hoy and C. Wang, "Safe Robot Navigation among Moving and Steady Obstacles", Elsevier, 2015.

[5] A. V. Savkin and M. Hoy, "Reactive and the shortest path navigation of a wheeled mobile robot in cluttered environments," Robotica, vol. 32, pp. 323-330, 2013.

[6] D. Fox, W. Burgard and S. Thrun, "The Dynamic Window Approach to Collision Avoidance", IEEE Robot. Autom. Mag., vol. 4, 23-33 (1997).

[7] R. Simmons, "The Curvature-Velocity Method for Local Obstacle Avoidance", the 1996 IEEE International Conf. on Robotics and Automation, Minneapolis, Minnesota, 1996.

[8] N. Y. Ko and R. G. Simmons, "The Lane-Velocity Method for Local Obstacle Avoidance", the 1998 IEEE/RSJ International Conf. on Intelligent Robots and Systems, Victoria, B.C, Canada, 1998.

[9] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles", International Journal of Robotics Research, vol. 17, no. 7, pp. 760-772, 1998.

[10] J. V. D. Berg, M. Lin and D. Manocha, "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation", 2008 IEEE International Conf. on Robotics and Automation, Pasadena, CS, USA, 2008.

[11] A. Chakravarthy and D. Ghose, "Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach", IEEE Trans. on Systems, Man and Cybernetics-Part A: Systems and Humans, vol. 28, no. 5, 1998.

[12] T. Fraichard and H. Asama, "Inevitable Collision States-a Step towards Sager Robots", Advanced Robotics, vol. 18, no. 10, pp. 1001-1024, 2004.

[13] A. Bautin, L. Martinez-Gomez and T. Fraichard, "Inevitable Collision States: A Probabilistic Perspective", 2010 IEEE International Conf. on Robotics and Automation, Anchorage, Alaska, USA, 2010.

[14] A. V. Savkin and C. Wang,"A simple biologically inspired algorithm for collision-free navigation of a unicycle-like robot in dynamic environments with moving obstacles", Robotica, vol. 31, pp. 993-1001, 2013.

[15] H. Teimoori and A. V. Savkin, "A Biologically Inspired Method for Robot Navigation in a Cluttered Environment", Robotica, vol. 28, pp. 637-648, 2010.

[16] H. Teimoori and A. V. Savkin, "Equiangular Navigation and Guidance of a Wheeled Mobile ", Robotics and Autonomous Systems, vol. 58, pp. 203-215, 2010.

[17] A. S. Matveev, C. Wang and A. V. Savkin, "Real-Time Navigation of Mobile Robots in Problems of Border Patrolling and Avoiding Collisions with Moving and Deforming Obstacles", Robotics and Autonomous Systems, vol. 60, pp. 769-788, 2012.

[18] A. V. Savkin and C. Wang, "Seeking a Path through the Crowd: Robot Navigation in Unknown Dynamic Environments with Moving Obstacles Based on an Integrated Environment Representation", Robotics and Autonomous Systems, vol. 62, pp. 1568-1580, 2014.

[19] Y. Zhu, T. Zhang, J. Song and X. Q. Li, "A New Hybrid Navigation Algorithm for Mobile Robots in Environments with Incomplete Knowledge", Knowledge-Based Systems, vol. 27, pp. 302-313, 2012.