A Sensor Based Navigation Algorithm of a Mobile Robot with Moving Obstacles in Its Workspace Assuring Convergence Property

Tomita, Masaaki
Kyushu Polytechnic College

Yamamoto, Motoji
Department of Mechanical Engineering, Faculty of Engineering, Kyushu University : Professor

https://hdl.handle.net/2324/14897

出版情報：九州大学工学紀要. 69 (2), pp.25-35, 2009-06-22. Faculty of Engineering, Kyushu University
バージョン：
権利関係：

Memoirs of the Faculty of Engineering, Kyushu University, Vol.69, No.2, June 2009

A Sensor Based Navigation Algorithm of a Mobile Robot with Moving Obstacles in Its Workspace Assuring Convergence Property

by

Masaaki TOMITA* and Motoji YAMAMOTO**

(Received February 9, 2009)

Abstract

A sensor based navigation algorithm for a mobile robot which assures a convergence property is proposed when the robot is navigated from a given start position to a goal position while avoiding moving and fixed obstacles. This algorithm guides the robot toward the goal by using its sensor information in an unknown environment. Conventional sensor based navigation algorithms such as Bug algorithm and Tangent Bug algorithm do not assure the convergence property to the goal position, and they may fail because of dead-locks in the presence of moving obstacles. In the real world, a robot should move around in a workspace with both fixed and moving obstacles. Typical examples of moving obstacles include human beings in real world workspace. We propose a new algorithm to guide a mobile robot toward the goal position. The algorithm can be applied for such workspace. The basic concept of this algorithm is based on detecting a loop in the robot's path that is characteristic for the avoidance of moving obstacle. Simulation examples of the sensor based navigation in the presence of moving and fixed obstacles are shown to demonstrate effectiveness of the proposed algorithm.

Keywords: Moving obstacle, Sensor based navigation, Mobile robot, Loop of path, Convergence property, Simulation

1. Introduction

This paper proposes a sensor based navigation algorithm that guides a mobile robot toward a goal while avoiding unknown moving obstacles. When applying an intelligent mobile robot to execute a given task in a real world, especially in an unknown or partially known workspace environment, the robot must have some sensors to detect obstacles and it should move toward a given goal position while avoiding the unknown obstacles using its sensor information. A motion algorithm for the problem is called "sensor based navigation" which is very basic and important

---

* Kyushu Polytechnic College
** Professor, Department of Mechanical Engineering

function for intelligent mobile robots. In this study, we propose a new sensor based navigation algorithm for moving obstacles in an unknown environment. The unknown environment is defined as one in which geometrical information such as positions and shapes of the obstacles are not known for the mobile robot in advance. This is typical in the real world, because the positions of many objects, including human beings, always change and it is difficult to accurately model exactly for all fixed and moving obstacles in advance.

The navigation algorithm has two types: the sensor based navigation algorithm and the model based navigation algorithm. In the sensor based navigation algorithm, the sensor information of a mobile robot is used to guide the robot toward a goal while avoiding obstacles. In the model based navigation algorithms,1) 2) 3) apriori map information of the environment and the robot's current information are used to guide the robot. A typical resultant path by the model based navigation algorithm is effective and it is easy to assure the convergence property to a goal position for the algorithm, because the algorithm is based on known map information of the workspace. However it is difficult to apply the algorithm to the real dynamic world where the map should be changed dynamically. For this reason, the model based navigation algorithms are not practical. On the other hand, the sensor based navigation algorithms are practical, because the robot obtains local information of the unknown environment using its sensors and the information can be used for the navigation. Some sensor based navigation algorithms have been proposed such as "Dynamic Window Approach"4), and "Vector Field Histgram"5), however such algorithms only guarantee a convergence property for fixed convex obstacles in the workspace.

Lumelsky has proposed sensor based navigation algorithms such as Bug1, Bug26) and Tangent Bug7) that guide a robot toward a goal using the information of the robot's current position and the goal position. He has shown a convergence property to the goal position using his algorithms, even if the obstacle information is not known at all in advance. The algorithm works for any kind of fixed unknown obstacles including very complex multiple concave obstacles. However the convergence is guaranteed only for fixed obstacles. In the real world, there are many moving obstacles such as humans. A mobile robot using this type of the conventional sensor based algorithm may fail to reach the goal because of looping in its path due to the presence of unknown moving obstacles; this is called as a dead-lock problem of the sensor based navigation. Therefore, we require a new navigation algorithm which is applicable to mobile robots in the real world including unknown moving obstacles in addition to unknown fixed obstacles.

In this paper, we propose a new navigation algorithm to avoid moving and fixed obstacles without a dead-lock. The algorithm distinguishes moving obstacles from fixed ones by detecting a crossing point of the mobile robot's path that is not observed for the case of fixed obstacles. By detecting the crossing point, the proposed algorithm leads a robot toward the goal without a dead-lock problem. We first discuss the drawbacks of the conventional algorithm such as Tangent Bug algorithm and other Bug type algorithm in dealing with moving obstacles, and then extend the algorithms to the application for moving obstacles in order to assure the convergence property. Some simulation examples to show the convergence of a robot for a goal position in the presence of moving obstacles in its workspace are shown.

Section two explains the dead-lock problem of the conventional algorithms for moving obstacles. Section three discusses how to detect the dead-lock situation peculiar to moving obstacles, and in section four, we present a new sensor based navigation algorithm for the dead-lock problem. Finally, effectiveness of the algorithm is demonstrated by some simulation results.

2. Dead-lock problem by moving obstacles for the conventional algorithm

We discuss a drawback of the conventional sensor based navigation algorithms based on the information of robot's position in dealing with moving obstacles. An important problem of the conventional sensor based algorithms such as Tangent Bug (T-Bug) algorithm is that they may fail due to dead-locks where they cannot guide the robot toward a goal in the presence of moving obstacles. Other conventional sensor based algorithms have the same problem of dead-locks, because they all assume that the obstacles in the workspace are fixed. To explain this problem, we first describe the behavior of a mobile robot by using T-Bug algorithm.

T-Bug algorithm consists of two basic behaviors. One is to move toward a goal where the goal position is known for the robot, and the other is to follow the boundary of an obstacle using sensor information. The dotted line path is for the former behavior and the solid line path is for the latter behavior in **Fig. 1**. The figure shows two cases of the path by T-bug algorithm. The left one is for a fixed convex obstacle and the right one is for a fixed concave obstacle. When the robot detects an obstacle within its sensor range, the robot selects the direction along the tangent lines to the obstacle, which minimizes the distance to the goal, by estimating a tangent point (a hit point) on the obstacle. Then the robot smoothly approaches the obstacle, changing the direction with the tangent line at each step. After approaching the obstacle, the robot follows the boundary of the obstacle. The robot moves to goal direction leaving from the boundary of the obstacle if the following distance condition and geometrical condition are both satisfied;

[Distance condition] $d(\boldsymbol{x},\boldsymbol{G}) < d(\boldsymbol{H},\boldsymbol{G})$ (1)
[Geometrical condition] no obstacle for goal direction within sensor range at $\boldsymbol{x}$

where $d(\boldsymbol{P},\boldsymbol{Q})$ is the distance from the point $\boldsymbol{P}$ to $\boldsymbol{Q}$ in the workspace, $\boldsymbol{x}$ is the current position of the mobile robot, $\boldsymbol{G}$ is the goal position and $\boldsymbol{H}$ is the last hit point in the robot's path. The last hit point $\boldsymbol{H}$ corresponds to the point A in the left path of **Fig.1**. The point $\boldsymbol{H}$ of the right path is a little bit complicated. The hit point $\boldsymbol{H}$ corresponds to the point C in the path from C to D, and corresponds to the point E in the path from E to F in the case of the right path in **Fig.1**. Note that the distance d in (1) is calculated by the current robot's position and the past path data. The current robot's position is measured by a dead reckoning technique or a global position sensing system, and the past path data is stored at every step of the robot's position. The convergence to the goal position is guaranteed by the distance condition (1) even if an unknown fixed concave obstacle. To guarantee the convergence property for a fixed obstacle, the distance condition should be satisfied.

However, conventional sensor based navigation algorithms including T-Bug algorithm do not always work for a moving obstacle. The path in **Fig. 2** is the success case for a moving obstacle. Where the number in the circle means the order of time for the mobile robot and the number in the box means the order of time for the moving obstacle including the case in **Fig.3**. When the robot finds the free workspace for the goal direction during the boundary following, the robot moves away from the boundary at the point L in **Fig. 2** by the distance condition (1) and the geometrical condition. The path in **Fig.3** is the failure case for the convergence to the goal position which means a dead-lock. In this case, the robot is taken to an opposite direction to the goal by following the boundary of the moving obstacle. During the following, the distance condition (1) is never satisfied where the last hit point is H in **Fig.3**, thus the robot cannot moves away from the moving obstacle. In another type of conventional sensor based navigation algorithms that make use of the rotation angle of the robot (which is called "the phase condition") to guide it toward the goal, the dead-lock problem for moving obstacles occurs in the phase condition as well as in the distance condition, because such algorithms assume only fixed obstacles.

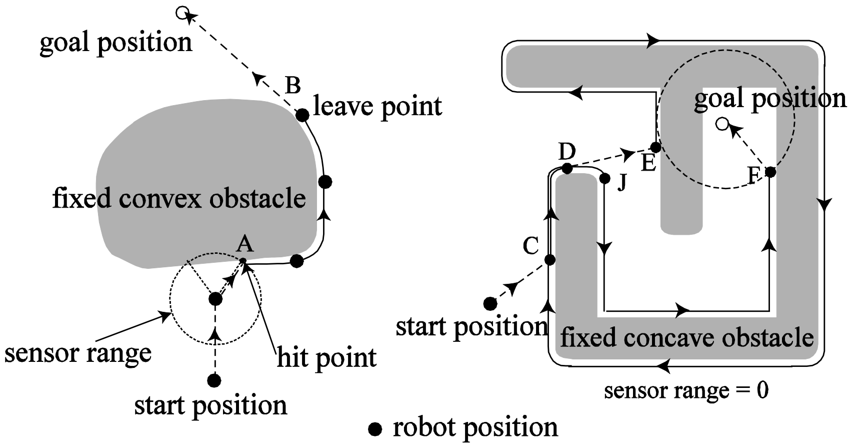
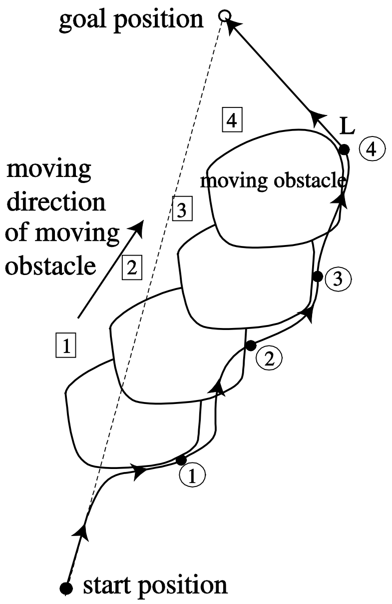


Fig. 1 The behavior of a mobile robot by T-Bug algorithm for fixed convex and concave obstacles.



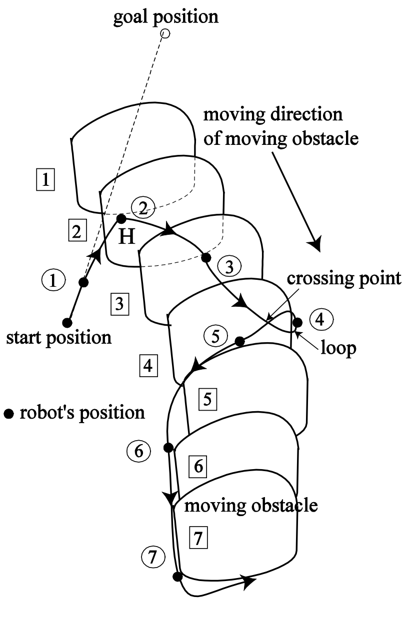


Fig. 2 Success case by T-Bug algorithm. **Fig. 3** Failure case by T-Bug algorithm.

3. Detecting the dead-lock situation for moving obstacles

We propose a new sensor based navigation algorithm for the problem of a dead-lock by a moving obstacle. The basic idea is to detect a loop of the path which is peculiar to the case of moving obstacle avoidance. The loop and the crossing point can be observed in **Fig.3**.

3.1 The problem statement of a sensor based navigation for unknown fixed and moving obstacles

This section describes the problem of the sensor based navigation including assumptions for the mobile robot and the workspace.

[Problem statement]

Design a motion algorithm of a mobile robot which navigates the robot to a given goal position in a workspace using the robot's sensor information.

[Assumption on a workspace]

(1) The workspace of a robot is two dimensional. In the workspace, there are finite fixed and moving obstacles. Their size and the number are limited. The shape of a fixed obstacle is arbitrarily concave or convex.
(2) The moving obstacle is arbitrary convex considering human beings are roughly approximated with a convex moving obstacle. The moving obstacle moves in an arbitrary direction. The velocity of the moving obstacle is less than that of the robot, which prevents the robot from being chased all the time by the moving obstacle leading to an obvious dead-lock.
(3) There is one goal position and the point is not occupied by the moving obstacle.

[Assumptions on a robot]

(4) The robot measures the current position by a dead reckoning technique or a global position sensing system.
(5) The robot has a local sensor which detects an obstacle's boundary within its sensor range. This is realistic consideration for a typical sonar sensor or a laser range finder.
(6) The robot knows the goal position in advance, but it does not know any obstacle's information in advance. Using the position information of the robot, the distance from the current robot's position to the goal position can be calculated.
(7) The robot is approximated by a circle with a radius *r*. By enlarging the obstacles with the radius of the robot artificially, the robot is expressed by a point.

3.2 A method to know a moving obstacle

As observed in **Fig.1** (right), the robot has to move away from the boundary of fixed obstacles only for the case that the distance condition is satisfied while the robot follows the boundary, if the goal direction is not blocked by an obstacle. If the robot moves away from the boundary without satisfying the distance condition, it may reach a dead-lock state. For example, if the robot leaves at the point J from the boundary in the case of **Fig.1** (right), it produces a loop which means a dead-lock.

On the other hand, the robot should moves away from the boundary for the case of moving obstacle if the goal direction is not blocked by an obstacle, as observed in **Fig.3**. However, the robot cannot distinguish a fixed obstacle from moving one easily using its sensor information. When it follows the boundary of the moving obstacle, it makes a loop of the path by tracing around the moving obstacle as shown in **Fig.3**. This type of loop is never observed in the case of fixed obstacles. The loop or crossing point is generated by rotating for the surface of the moving obstacle with 360 (deg). The crossing never appears for fixed obstacles. However, the crossing point may not appear even if the rotation around a moving convex obstacle because of singularity of topology, which is still an open problem for the exact proof of the convergence property of this algorithm. Using the fact, when the loop of the path is detected, the robot judges that the obstacle is moving and the robot moves away from the boundary if the goal direction is open. Thus the problem is how to detect the loop or the crossing point of the robot's path. We next discuss a method to detect the loop of the robot's path which is peculiar to a moving obstacle.

In our navigation algorithm, the robot continues to store its position data at each motion step- *i*.

The set of the position data is called the path of the robot. At first, a vector product of a two dimensional vector is defined by using two vectors $\boldsymbol{A}=(A_x, A_y)$, $\boldsymbol{B}=(B_x, B_y)$

$$\boldsymbol{A}\times\boldsymbol{B}\equiv A_x B_y - A_y B_x \tag{2}$$

We here write the current robot's position vector as $\boldsymbol{P}_n$, and the position vector at one step before as $\boldsymbol{P}_{n\text{-}1}$. The past robot's point on the path is described with $\boldsymbol{P}_i$ at step-i. Then the condition that the vector $\boldsymbol{P}_n - \boldsymbol{P}_{n-1}$ lies between the vectors $\boldsymbol{P}_{i-1} - \boldsymbol{P}_{n-1}$ and $\boldsymbol{P}_i - \boldsymbol{P}_{n-1}$ is

$$((\boldsymbol{P}_n - \boldsymbol{P}_{n-1})\times(\boldsymbol{P}_{i-1} - \boldsymbol{P}_{n-1}))\cdot((\boldsymbol{P}_n - \boldsymbol{P}_{n-1})\times(\boldsymbol{P}_i - \boldsymbol{P}_{n-1})) < 0 \tag{3}$$

as shown in the case (A) of **Fig.4**. Similarly, the condition that the vector $\boldsymbol{P}_i - \boldsymbol{P}_{i-1}$ lies between the vectors $\boldsymbol{P}_{n-1} - \boldsymbol{P}_{i-1}$ and $\boldsymbol{P}_n - \boldsymbol{P}_{i-1}$ is

$$((\boldsymbol{P}_i - \boldsymbol{P}_{i-1})\times(\boldsymbol{P}_{n-1} - \boldsymbol{P}_{i-1}))\cdot((\boldsymbol{P}_i - \boldsymbol{P}_{i-1})\times(\boldsymbol{P}_n - \boldsymbol{P}_{i-1})) < 0 \tag{4}$$

as shown in the case (B) of **Fig.4**. The crossing condition of the two vectors $\boldsymbol{P}_i - \boldsymbol{P}_{i-1}$ (past path) and $\boldsymbol{P}_n - \boldsymbol{P}_{n-1}$ (current velocity of robot) is written by that the condition (3) and (4) are both satisfied as shown in the case (C) in **Fig.4**. By checking the condition (3), (4) from step i=1 to i=n-2 at all the motion steps, the robot detects the loop which means a moving obstacle.

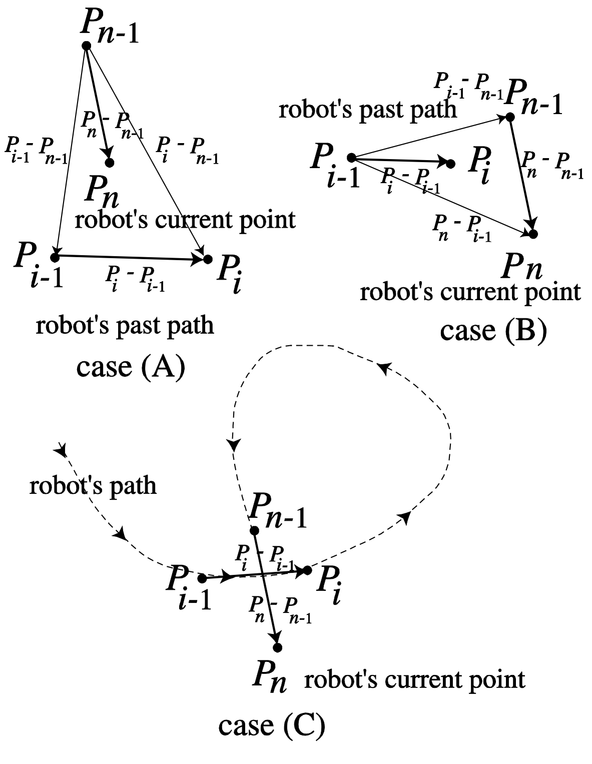


Fig. 4 Crossing of two vectors.

4. A sensor based navigation algorithm for moving and fixed obstacles

We propose a new navigation algorithm to assure the convergence of the robot toward the goal in the presence of moving and fixed obstacles. The algorithm is described by;

(At every motion step, the current position of the robot is stored. See the flow in **Fig.5**.)

(i) When the robot detects an obstacle within its sensor range, it approaches the obstacle and begins to follow the obstacle's boundary. The robot stores its position as the last hit point. Go to (iii).

(ii) When the robot does not detect an obstacle for the goal direction, it moves straight to the goal direction with one step. If the point is the goal position, then stop the navigation, otherwise go to (i).

(iii) The robot continues to follow the current obstacle with one step. If the point is satisfied with the distance condition (1) and the goal direction is free from an obstacle, then go to (i). If the point is satisfied with the distance condition (1), but the goal direction is not open by the obstacle, then go to (iii). If the point is not satisfied with the distance condition (1), go to (iv).

(iv) If the current point is satisfied with the crossing condition ((3) and (4)), go to (v). Otherwise go to (iii).

(v) If the direction for the goal is not open by an obstacle, the robot continues to follow the current obstacle with one step. When the direction for the goal is open, go to (i).

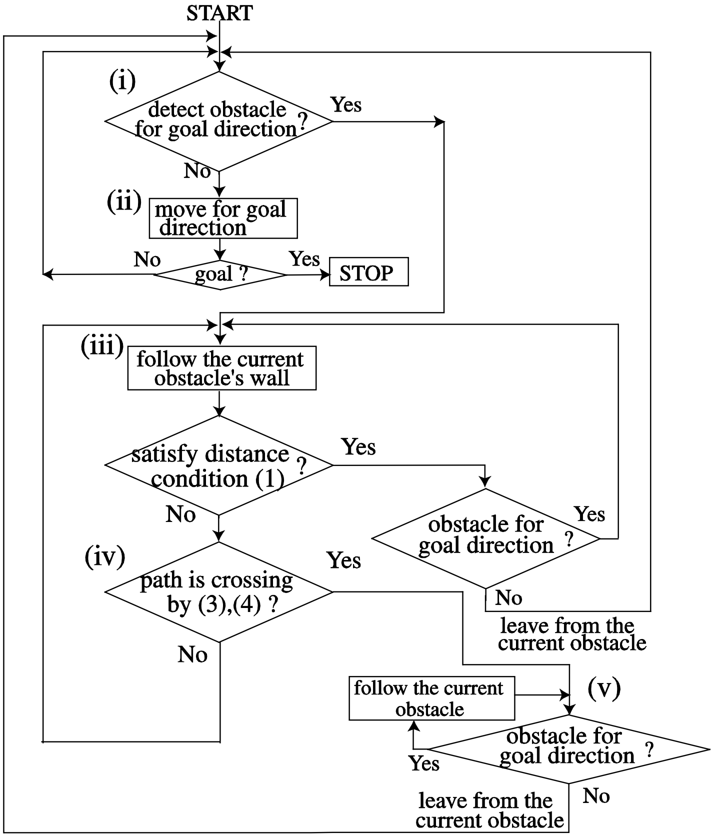


Fig. 5 The flow chart of the proposed algorithm.

5. Simulations

Some simulation examples of the sensor based navigation for fixed and moving obstacles are shown. In this simulations, the size of the workspace is 600cm×600cm. The moving obstacle is a quadrangular polygon. An example using the conventional navigation algorithm (T-bug algorithm) is shown in **Fig. 6**. The time order is from (a) to (f). The robot takes to the opposite direction to the goal by the moving obstacle and did not find the point satisfying the distance condition (1). The robot fails to reach the goal position.

Figure 7 shows the case of applying to the proposed algorithm for the same moving obstacle with the case in **Fig. 6**. In this example, the robot detects a crossing point in (e) of **Fig.7**, and it begins to move away from the obstacle toward the goal direction. By detecting the crossing point of the robot's path, the robot escapes from a dead-lock situation peculiar to the moving obstacle.

By the example in **Fig. 8,** we can confirm that the proposed algorithm is effective for the case that the workspace has both moving and fixed obstacles. This algorithm also works for a complicated unknown fixed obstacle as in **Fig.8**.

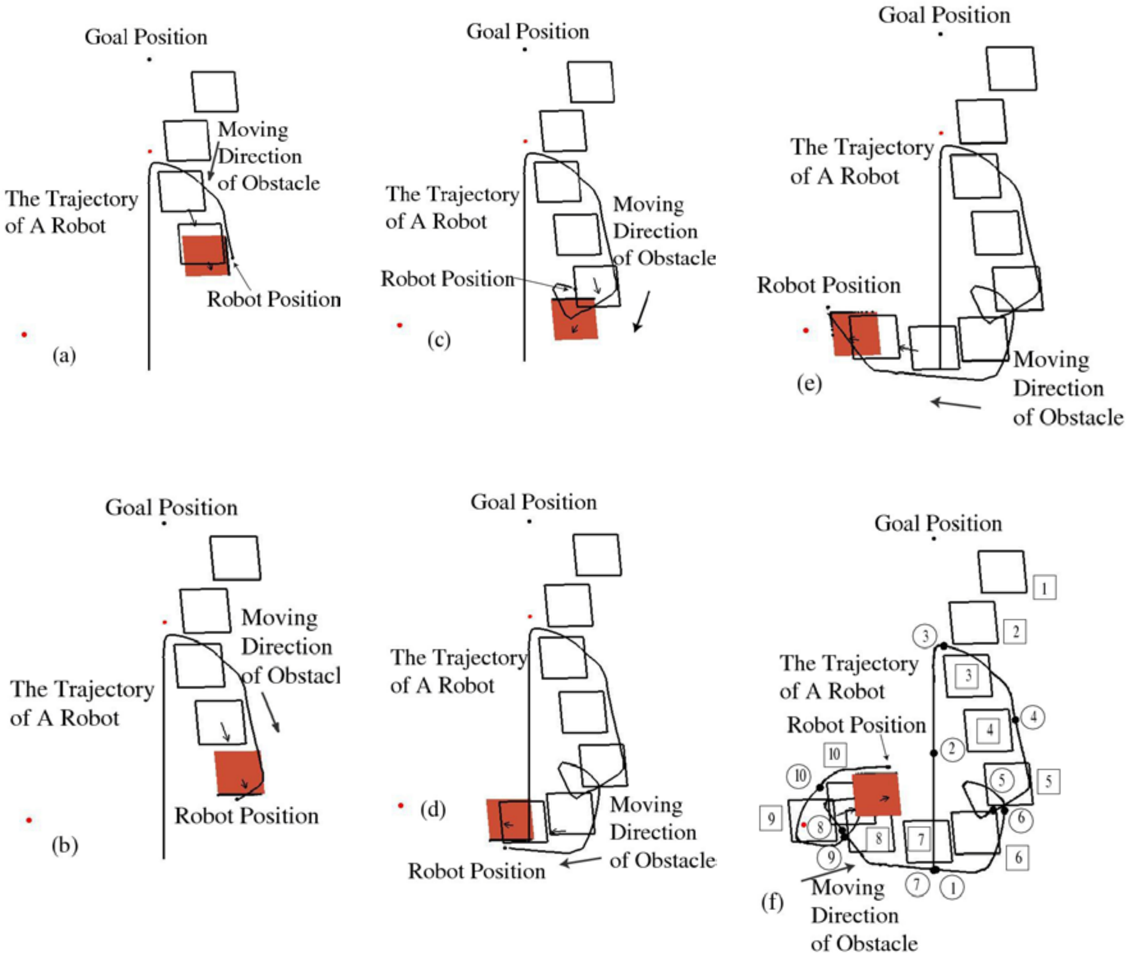


Fig. 6 Failure example for a moving obstacle by T-Bug algorithm.

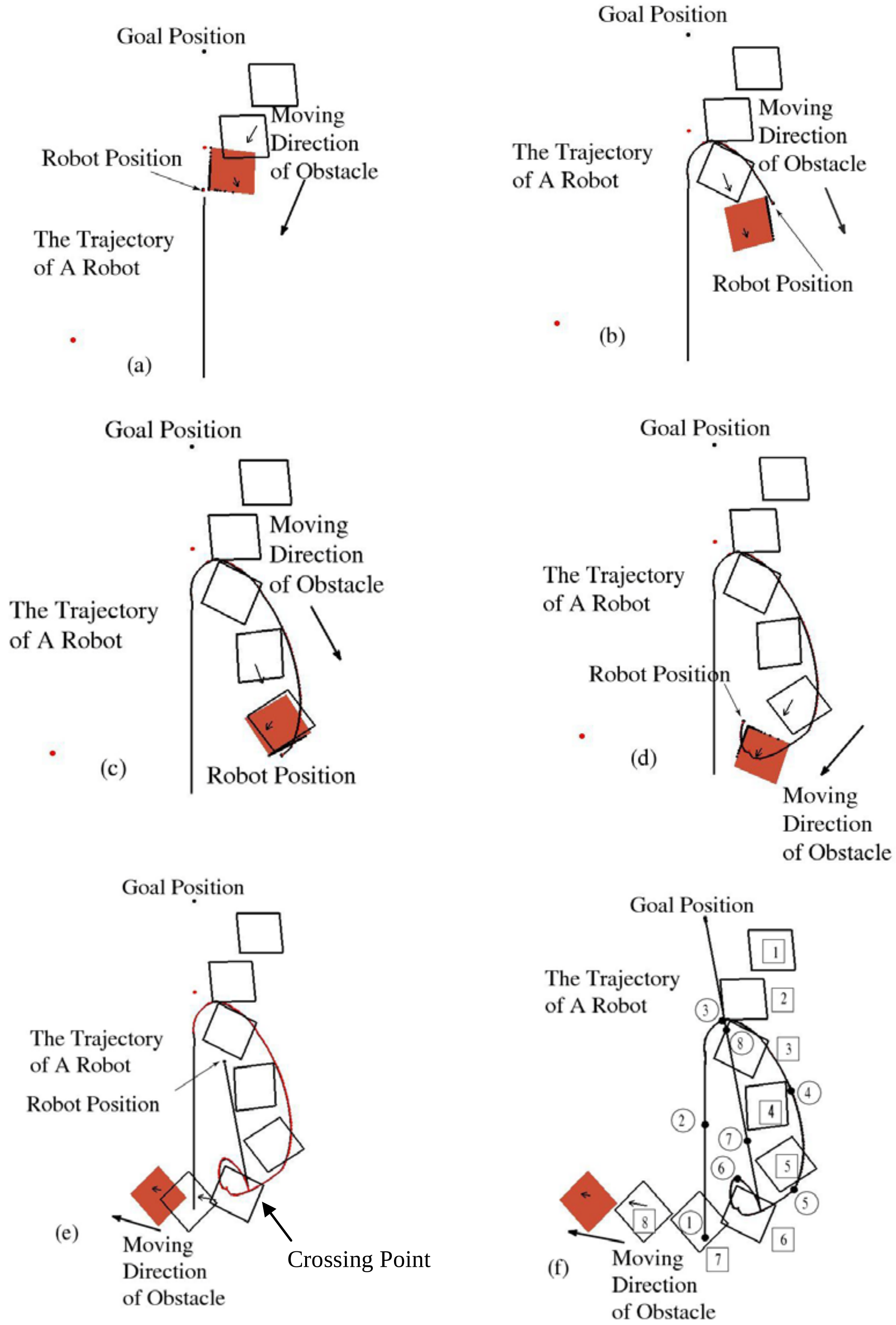


Fig. 7 Example of robot motion by proposed algorithm in the presence of moving obstacle.

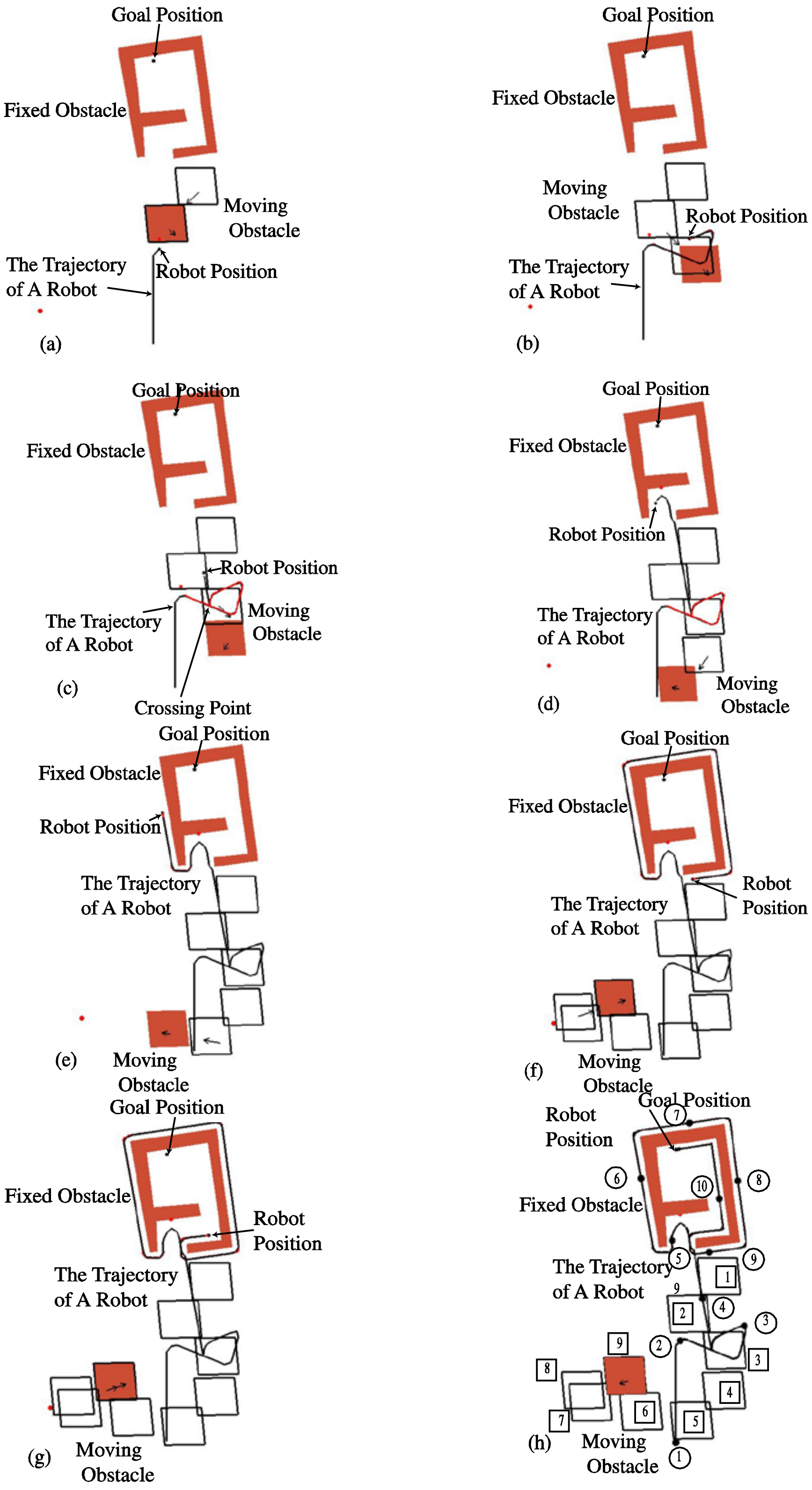


Fig. 8 Example of robot motion by proposed algorithm in the presence of both moving and fixed obstacle.

6. Conclusion

A dead-lock problem of the conventional sensor based navigation algorithms in the presence of a moving obstacle is discussed. When the robot is following an obstacle's boundary around the moving obstacle using the sensor based navigation algorithm, a loop is produced for the robot's path. By the loop condition (crossing condition of path), the robot moves away from the obstacle's boundary even if the distance condition is not satisfied in the proposed navigation algorithm. The algorithm assures the convergence to the goal position for a moving obstacle including fixed complicated obstacles. Numerical simulations show an effectiveness of the proposed navigation algorithm.

References

1)J. Ohta, et al.: Intelligent Robots-Motion Planning Problem,
Corona Publishing Co.Ltd, pp.34-67, 2001.[in Japanese]
2) H.Choset, et al.: Principles of Robot Motion,
The MIT Press, pp.77-187, 2005.
3) S. Ishikawa and S. Asaka,: A Method of Piloting an Autonomous Mobile Robot in Dynamically Chainging Environment Including Moving Obstacles,
Journal of Robotics Society of Japan, Vol.11, No.6, pp.856-867, 1993.
4) D. Fox, et al.: The Dynamic Window Approach to Collision Avoidance,
IEEE Robotics and Automation Magazine, Vol.4,No.1,pp.22-33, 1997.
5) I. Ulrich and J. Borenstein,: VFH:Local Obstacle Avoidance with Look-Ahead Verification,
Proc. of the Int. Conf. on Robotics and Automation, pp.2505-2511, 2000.
6) V. J. Lumelsky and A. A. Stepanov,: Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles for Arbitrary Shape,
Algorithmica, Vol.2,pp.403-430, 1987.
7) I. Kamon, et. al.,: A New Range-Sensor Based Globally Convergent Navigation Algorithm for Mobile Robot,
Proc. of the IEEE Int. Conf. on Robotics and Automation, pp.429-435, 1996.