Practical Obstacle Avoidance Using Potential Field for a Nonholonmic Mobile Robot with Rectangular Body

Hiroaki Seki, Satoshi Shibayama, Yoshitsugu Kamiya, and Masatoshi Hikizu
Department of Mechanical Systems Engineering, Kanazawa University
Kakuma-machi, Kanazawa 920-1192, Japan
hseki@t.kanazawa-u.ac.jp

Abstract

A real-time obstacle avoidace algorithm based on sensor data is required for mobile robots. When a mobile robot is nonholonomic and it has polygonal body, the method using configuration space is usually applied. But, it is complex and it needs much computing power. On the other hand, artificial potential field is often used for real-time obstacle avoidance, however, most of them consider a mobile robot as an ominidirectional movable point.

Therefore, we propose a new method of practical obstacle avoidance for a mobile robot with rectangular body. Action points of repulsive forces from obstacles are located in both the front and rear of robot's body. Their forces are generated according to the distances between obstacles and robot's body. Rear forces are transferred to the front by inverting and the resultant force moves the robot. This method is very simple and effective.

1. Introduction

Obstacle avoidance is an important function for mobile robots. Let us to discuss about the obstacle avoidance for a nonholonomic mobile robot like an autonomous wheelchair (Figure 1). It has two independently driven wheels and a body with a certain shape. If a mobile robot can be treated as an omunidirectional movable point, numerous methods have been proposed and applied for it. Collision free path can be easily found by artificial potential field [2, 9], graph theory [12], sensor based method and so on. The problem for a nonholonomic robot with two independently driven wheels can come down to that for an omnidirectional point by approximating robot's shape to a circle with the center at the midpoint of two wheels. As shown in Figure 2, obstacles should be expanded by the radius of the robot's circle and the robot should be contracted to a point. However, it isn't reasonable to regard the rectangular body like a wheelchair as a circle and its circle sometimes can't pass through the narrow place where the original body can do.

In case of an omnidirectional (holonomic) mobile robot, "configuration space" can be used for its path planning when the robot's shape is considered explicitly [11]. This problem is named "piano movers' problem" [10]. A set of position and orientation where a robot body doesn't collide with obstacles is represented by three dimensional configuration space. A path of robot's position and orientation should be searched in this space by probabilistic roadmap method [1] for example. There are some studies considering both shape of robot's body and nonholo-

Figure 1. Autonomous wheelchair moving through a narrow space.

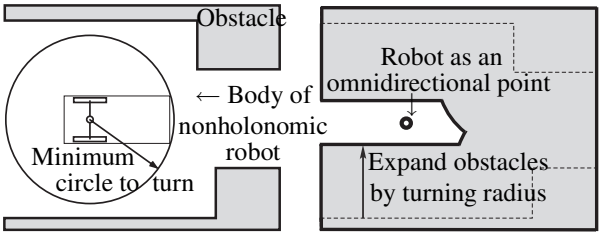

Figure 2. Approximation of robot's shape by a circle for path planning.

1-4244-1506-3/08/$25.00 ©2008 IEEE

nomic motion [3, 7, 8]. It is very difficult problem to search a path in the configuration space under the motion constraint. Laumond [5] solved this by modifying the collision free path obtained without motion constraint so as to satisfy motion constraint. Latombe [4] proposed that the configuration space is divided into cells, the cells where a nonholonomic robot can move by simple motion such as turnig, going straight, pulling over are connected by graph, and a path is searched in the graph. Anyway, these methods are too complicated for real-time obstacle avoidance using real sensor information although these ensure the solution of collision free path. Specially, calculation of configuration space needs much computing power.

Therefore, we propose a practical method of local obstacle avoidance for a nonholonomic mobile robot with rectangular body. Simple potential field using local sensor information of surrounding obstacles is applied.

2. Problem Statement

The obstacle avoidance problem to be solved is stated as follows.

(1) We consider a nonholonomic mobile robot with two independently driven wheels as shown in Figure 3 It moves in a planar environment. The configuration of a robot is defined by $\boldsymbol{R} = (X, Y, \Theta)^T$ in the base coordinates, where (X, Y) is the position of the midpoint of two wheels' axis and Θ is its orientation. The discrete kinematic model of this robot is written as

$$\begin{cases} X_n = X_{n-1} + v\Delta t \cos(\Theta_{n-1} + \dfrac{\omega\Delta t}{2}) \\ Y_n = Y_{n-1} + v\Delta t \sin(\Theta_{n-1} + \dfrac{\omega\Delta t}{2}) \\ \Theta_n = \Theta_{n-1} + \omega\Delta t \end{cases} \quad (1)$$

where Δt is the sampling time for control and $(v, \omega)^T$ are translational and rotational velocity. Suffix $n-1, n$ denote positions before and after the sampling time.

(2) The shape of a robot is (or can be approximated by) a rectangle. Let the vertices of the robot's shape be $\boldsymbol{r_i}(i = 1, 2, ..., n_r)$ in the robot coordinates.

(3) A laser range sensor is mounted on the robot to detect obstacles. It has a circular detection area. Obstacles are scanned by this sensor every a certain angle. Let the detected points on the outline of obstacles be $\boldsymbol{p_j}(j = 1, 2, ..., n_p)$ in the robot coordinates. These points are called "obstacle points".

(4) Global path planning is given. After the goal position of a robot $\boldsymbol{R_G} = (X_G, Y_G, \Theta_G)^T$ is given relatively near the start position, a local path to avoid obstacles is found. We explain the case that the start position is behind the goal position and a robot go forward to the goal. When a robot go backward to the goal, the front and back of the robot should be swapped.

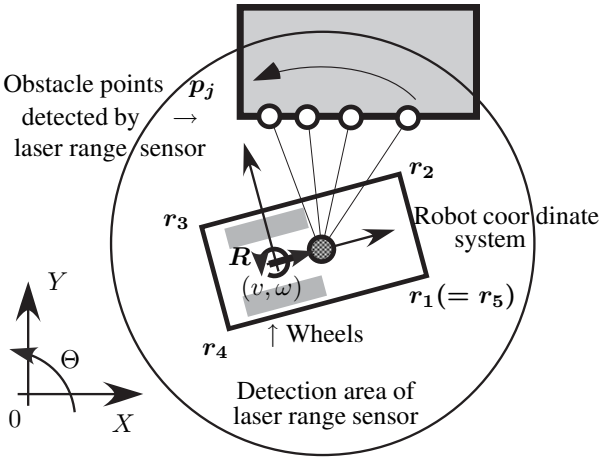

Figure 3. Model of nonholonimic robot with a laser range sensor.

3. Algorithm for Local Obstacle Avoidance

3.1. Outline

A method of local obstacle avoidance for a mobile robot with two driven wheels and rectangular body is proposed. This outline is shown in Figure 4. Basically, simple potential field is applied. Both an attractive force form the goal and repulsive forces from obstacles act on the robot and the resultant force moves the robot. Main differences between the general method using potential field and our proposed method are following two points.

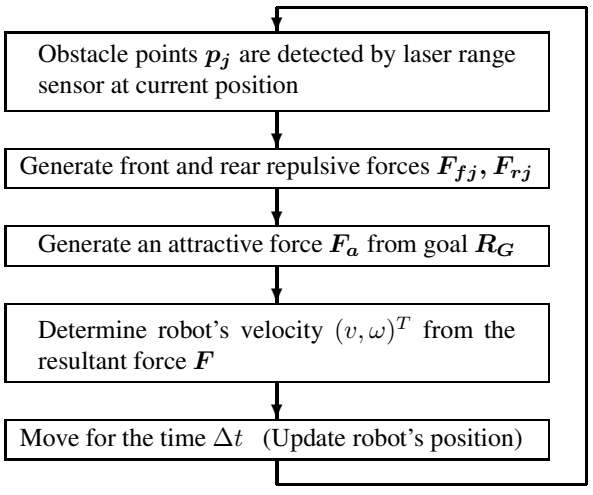

Figure 4. Flowchart of proposed algorithm for obstacle avoidance.

- In order to consider the motion constraint that a robot can't move just beside, two points of action where the attractive and repulsive forces act are placed on the front and rear body of a robot. Their forces at two points are treated as they work on a "lever" of which the flucrum is the midpoint of two wheels.
- In order to consider the shape of robot's body, repulsive forces form obstacles are determined by the distances between obstacle points and the outline of robot's body.

This idea can simply introduce the consideration about the motion constraint and the robot's shape into the potential field method. Proposed method needs almost same computing power as general potential field method because their calculations have little difference. Since the data of a laser range sensor (obstacle points) can be used directly, this method is suitable for real-time obstacle avoidance. However, this also has a disadvantage of the local minima problem.

Then, our proposed method is explained in detail in the following sections. The generation of forces and the determination of robot's velocity are treated on the robot coordinate system.

3.2. Generation of attractive and repulsive forces

Two action points of forces are placed at the front end and the rear end of a robot's body as shown in Figure 5. Let the front end be $\boldsymbol{r_f} = (x_f, 0)^T$, and the rear end be $\boldsymbol{r_r} = (-x_r, 0)^T$ in the robot coordinate system. These points should not always be placed at the ends of a robot, however, acting forces at the ends makes robot's motion stable. When a obstacle point $\boldsymbol{p_j} = (p_{jx}, p_{jy})^T$ is detected in front of the line of two wheels' axes (y axis), a repulsive force $\boldsymbol{F_{fj}}$ is generated at the front point of action. When a obstacle point is behind this line, a repulsive force $\boldsymbol{F_{rj}}$ is generated at the rear point of action. The magnitudes of their forces are changed in inverse proportion to the squares of the distances between obstacle points and a robot's body. Then, their forces are given by

$$\boldsymbol{F_{fj}} = \frac{K}{|\boldsymbol{q_{fj}} - \boldsymbol{p_j}|^2} \frac{\boldsymbol{r_f} - \boldsymbol{p_j}}{|\boldsymbol{r_f} - \boldsymbol{p_j}|}, \quad \text{if } p_{jx} > 0 \quad (2)$$

$$\boldsymbol{F_{rj}} = \frac{K}{|\boldsymbol{q_{rj}} - \boldsymbol{p_j}|^2} \frac{\boldsymbol{r_r} - \boldsymbol{p_j}}{|\boldsymbol{r_r} - \boldsymbol{p_j}|}, \quad \text{if } p_{jx} < 0 \quad (3)$$

where $\boldsymbol{q_{fj}}$, $\boldsymbol{q_{rj}}$ are the intersections of the robot's body and the segments between obstacle points and the action points $\boldsymbol{r_f}$, $\boldsymbol{r_r}$ respectively. K is the coefficient of repulsive force.

Next, an attractive force $\boldsymbol{F_a}(|\boldsymbol{F_a}| = 1)$ pulls the front action point $\boldsymbol{r_f}$ of the mobile robot toward the goal position as shown in Figure 6. This attractive force is the tangential vector at the front action point $\boldsymbol{r_f}$ to the circle which comes in contact with the goal orientation of the front action point $\boldsymbol{r'_f}$. Without any obstacles, the robot moves on this circle and arrives at the goal position $\boldsymbol{R_G} = (X_G, Y_G, \Theta_G)^T$. The attractive force $\boldsymbol{F_a} =$

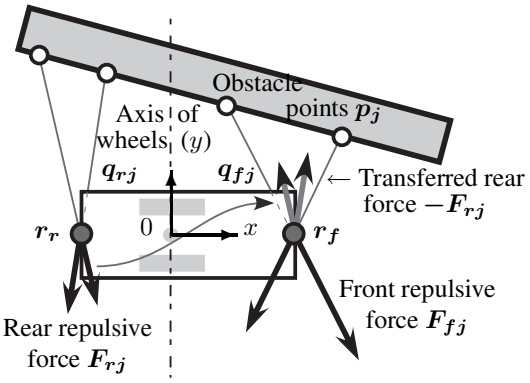

Figure 5. Generation of repulsive forces from obstacle points.

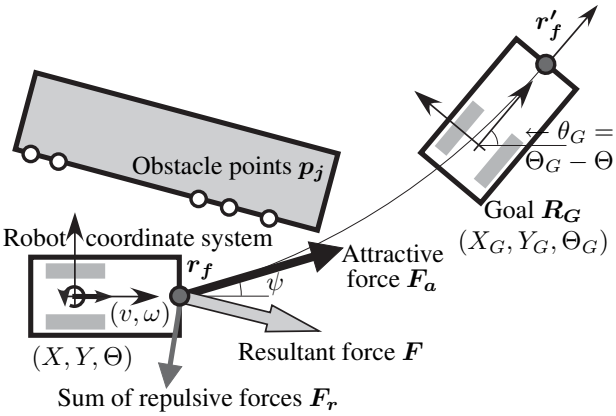

Figure 6. Generation of attractive force and determination of velocity for avoidance.

$(\cos\psi, \sin\psi)^T$ is given by

$$\psi = 2 \text{ atan2 } (y'_G, x'_G) - \theta_G \quad (4)$$

$$\theta_G = \Theta_G - \Theta \quad (5)$$

$$\begin{bmatrix} x'_G \\ y'_G \end{bmatrix} = R(-\Theta) \begin{bmatrix} X_G - X \\ Y_G - Y \end{bmatrix} + R(\theta_G)\boldsymbol{r_f} - \boldsymbol{r_f} \quad (6)$$

where $R(\theta)$ is a rotation matrix by angle θ.

3.3. Resultant force and determination of robot's velocity

A resultant force $\boldsymbol{F}$ is obtained from the attractive and repulsive forces $\boldsymbol{F_a}$, $\boldsymbol{F_{fj}}$, $\boldsymbol{F_{rj}}$. Since the action points of their forces are not same, we can't simply add their force vectors. After the repulsive forces at the rear action point $\boldsymbol{F_{rj}}$ are transferred to the front action point by inverting their vectors $-\boldsymbol{F_{rj}}$, all force vectors are added at the front action point, because the front action point should be moved in the opposite direction of the rear repulsive force in order to move the rear body of the robot

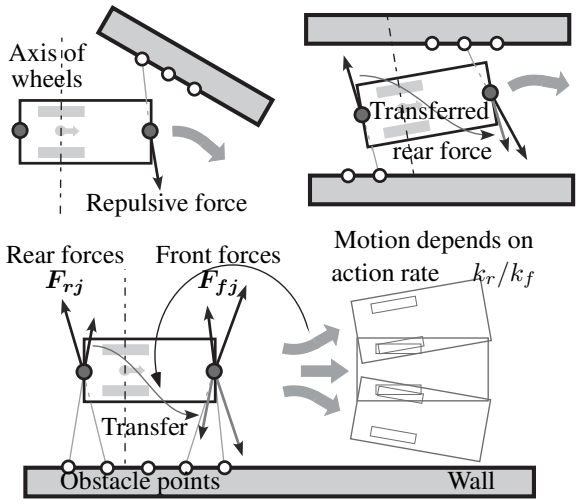

Figure 7. Effect of action rate between front and rear forces on robot's motion. (Wall following is sensitive and others are not.)

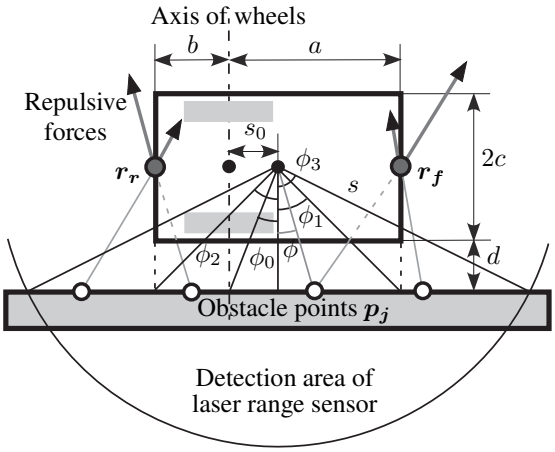

Figure 8. Geometry of robot's body and repulsive forces during wall following.

away from the rear obstacle point. That is, the front and rear action points have a relation like a "lever" of which the flucrum is the midpoint of two wheels. Then, the resultant force at the front action point $\boldsymbol{F}$ is defined by

$$\boldsymbol{F} = \boldsymbol{F_a} + k_f \sum_{p_{jx}>0} \boldsymbol{F_{fj}} - k_r \sum_{p_{jx}<0} \boldsymbol{F_{rj}} \tag{7}$$

$$k_f + k_r = 1 \tag{8}$$

where the coefficients k_f, k_r represent the action rate of the front and rear repulsive forces. The determination of these coefficients are mentioned later.

Finally, the resultant force $\boldsymbol{F}$ pulls the front action point to move the robot. In other words, the translational and rotational velocities of the robot $(v, \omega)^T$ are determined in order that the front action point $\boldsymbol{r_f} = (x_f, 0)^T$ moves in the direction of the resulatant force $\boldsymbol{F}/|\boldsymbol{F}| = (f_x, f_y)^T$.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = C \begin{bmatrix} f_x \\ \frac{f_y}{x_f} \end{bmatrix} \tag{9}$$

where C is the velocity coefficient. Since only the rate of translational and rotatinal velocities is obtained, suitable coefficient C should be given according to some limitations of velocity or acceleration. For example, when the maximum of the rotational velocity ω_{max} is specified, C becomes

$$C = \omega_{max} \frac{x_f}{f_y}, \ \text{if} \ |\omega| > \omega_{max} \tag{10}$$

3.4. Action rate of front and rear forces

How to determine the action rate of the front and rear repulsive forces k_f, k_r is discussed. When a mobile robot avoids a block of obstacle as shown in Figure 7, the force action rate doesn't affect robot's motion so much because repulsive forces mainly work at either action point. When a robot moves between walls on both sides, robot's motion isn't also sensitive to the action rate because repulsive forces at two points turn the robot in the same way. The case where the action rate affects robot's motion relatively is wall following. Repulsive forces are generated at two action points to move their points away from the wall and their direction to turn the robot is different because they are treated like a lever of which the flucrum is the midpoint of two wheels. During wall following, larger action rate of front forces k_f makes the robot turn away from the wall and larger action rate of rear forces k_r makes the robot turn close to the wall as shown in Figure 7. Therefore, the force action rate should be determined so as that the robot goes straight along a wall, i.e. the resultant force vector $\boldsymbol{F}$ should be palallel to the wall without considering the attractive force $\boldsymbol{F_a}$. Let the components of repulsive force vectors at the front and rear action points in the vertical direction to the wall be F_{fyj}, F_{ryj} respectively, this condition becomes

$$k_f \sum F_{fyj} - k_r \sum F_{ryj} = 0 \tag{11}$$

Then, the action rate

$$\frac{k_r}{k_f} = \frac{\sum F_{fyj}}{\sum F_{ryj}} \tag{12}$$

is obtained. This depends on the shape of a robot, the detection area of a laser range sensor and so on.

The action rate for a rectangular robot is concretely calculated as shown in Figure 8. Let the front length, rear length, width of a robot be a, b, $2c$, respectively. Let the distance between the wheels' axis and the laser range sensor be s_0 and the detection limit distance of the sensor be

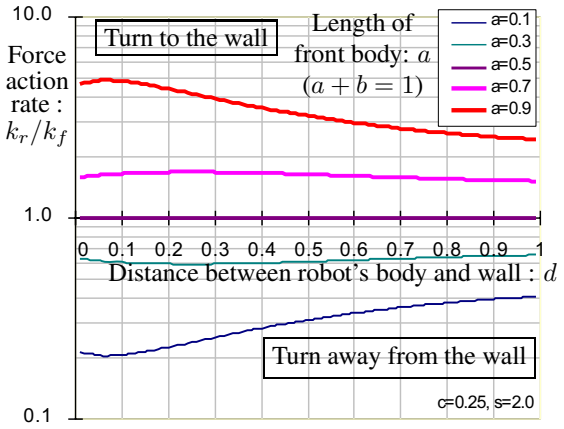

Figure 9. Force action rate k_r/k_f to go straight along a wall.

s. We can get the sum of components of repulsive force vectors in the vertical direction to the wall after repulsive forces, which are inversely proportional to the squares of the distances between the robot's body and the wall, are calculated. When the gap between a robot and a wall is d, they are given by

$$\sum F_{fyj} = \frac{KD^3}{d^2} I(a, -\phi_0, \phi_1) + KDI(a, \phi_1, \phi_3) \quad (13)$$

$$\sum F_{ryj} = \frac{KD^3}{d^2} I(-b, -\phi_2, -\phi_0) + KDI(-b, -\phi_3, -\phi_2) \quad (14)$$

$$I(\alpha, \phi_s, \phi_e) = \int_{\phi_s}^{\phi_e} \frac{d\phi}{((\alpha - s_0 - D\tan\phi)^2 + D^2)^{\frac{3}{2}}} \quad (15)$$

$$D = c + d \quad (16)$$

where ϕ is the angle from the sensor to an obstacle point on the wall, ϕ_0, ϕ_1, ϕ_2, ϕ_3 are the angles from the sensor to the intersections between the wall and the wheels' axis, the front line of the body, the rear line of the body, the circle of detection limit of the sensor. Finally, the action rate becomes

$$\frac{k_r}{k_f} = \frac{D^2 I(a, -\phi_0, \phi_1) + d^2 I(a, \phi_1, \phi_3)}{D^2 I(-b, -\phi_2, -\phi_0) + d^2 I(-b, -\phi_3, -\phi_2)} \quad (17)$$

This value is calculated by numerical integration.

Figure 9 shows the relation between the distance from a wall d and the action rate k_r/k_f for a robot to go straight along the wall. In this calculation, it is assumed that the sensor is placed at the center of the robot ($s_0 = (a-b)/2$) and the length of the robot is normalized ($a + b = 1$). Some cases of the front and rear length of the robot with the width $2c = 0.5$ are shown in the graph. It can be seen that the action rate for the robot to go straight does not change so much according to the distance from the wall if the driven wheels are not close to either end of the body

Table 1. Standard parameters for simulation

Range of laser range sensor	$0 \sim 1$ [m]
Directional resolution of lager range sensor	1 [deg.]
Sampling time for control: Δt	0.1 [s]
Coefficient of repulsive force: K	0.004
Coefficient of velocity: C	0.2
Maximum angular velocity: ω_{max}	0.2 [rad/s]

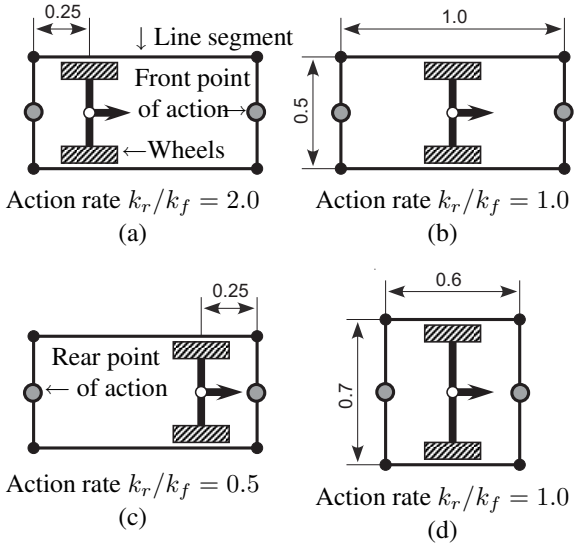

Figure 10. Shape of robots for simulation

($a = 0.3 \sim 0.7$). Even if the wheels are close to the end, there is no problem for obstacle avoidance because the action rate below these curves in the graph makes the robot turn away from the wall. When the front length a is short (Ex. $a = 0.1$), the minimum of the curve should be taken for the action rate. When a is long (Ex. $a = 0.9$), the value on the curve at a certain distance should be taken for the action rate because it makes the robot turn away from the wall if the robot goes inside its distance.

4. Simulation

Our proposed method of local obstacle avoidance has been tested. All simulation programs were written in C language on PC Linux system. Table 1 shows standard parameters for the simulation. We assumed the following situation. A laser range sensor is mounted on the center of the rectangular body of a robot. Since the scan resolution angle is 1 degree, the max. number of detected obstacle points is 360. An obstacle point is calculated as the nearest intersection of obstacles and a direction of a laser range sensor within its detection area. Scan time is short enough to be neglected as compared with robot's speed. 4 types of robot's bodies were prepared as shown in Figure 10. The action rate of the front and rear repulsive forces k_f, k_r was determined for each body by Equation (17).

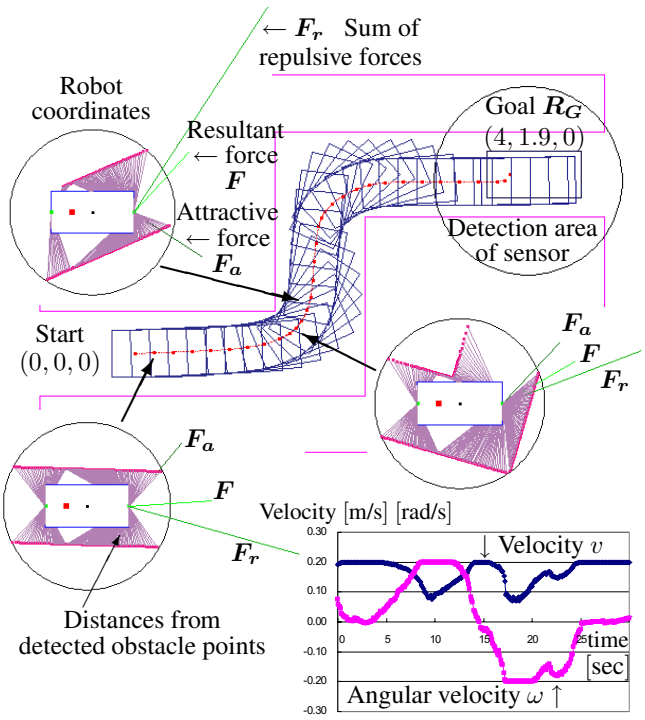

Figure 11. Simulation result of the robot with shape (a).

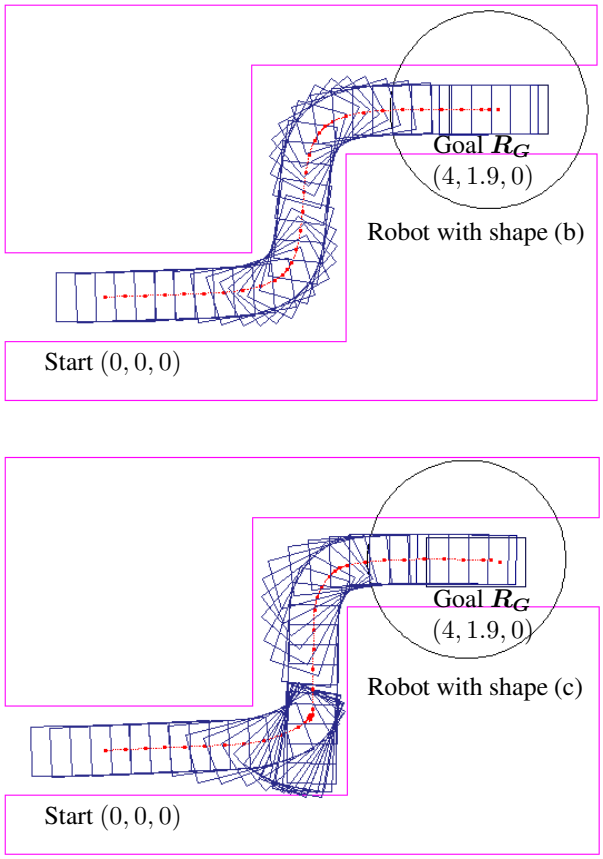

Figure 12. Simulation results of robots with various shape.

Figure 11 $\sim$ 14 are simulation results. Goal position R_G was given as shown in each figure. Figure 11 shows the generated path for the robot with shape (a) to pass through a narrow crank course. It can be seen that a smooth collision free path considering both rectangular body and motion constraint is generated by our proposed method. Obstacle points detected by the laser range sensor p_j, distances between the robot's body and them, sum of repulsive forces F_r, attractive forces F_a and resultant forces to avoid obstacles F are also shown in the robot coordinate system at some positions (See each circle in Figure 11). A collision free direction can be determined from the sensor information directly. Moreover, the translational and rotational velocities of the robot v, ω are plotted in the graph and we can see they changes smoothly.

Figure 12 shows the cases of other robots' bodies and Figure 13 shows the cases of other environments. It turns out that our proposed method is effective for various situations. Figure 14 shows the results for various coefficient of repulsive force $K = 0.0001 \sim 0.01$. Larger coefficent generates the path farther away from obstacles, however, it isn't too sensitive (See also Figure 11 of $K = 0.004$). When the coefficient is large, there seen some cases where the robot gets stuck at a local minimum. Many algorithms [6] to escape from the local minimum have been already proposed for general potential field method and some of them can be also applied to this case.

5. Conclusion

In this paper, a practical method of local obstacle avoidance for a nonholonomic mobile robot with rectangular body has been proposed. Simple potential field directly using local sensor data is applied. Repulsive forces according to distances between obstacles and robot's body

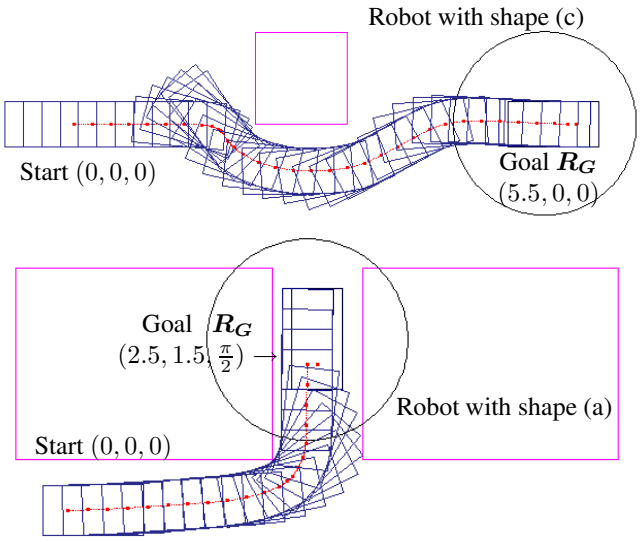

Figure 13. Simulation results for various environment.

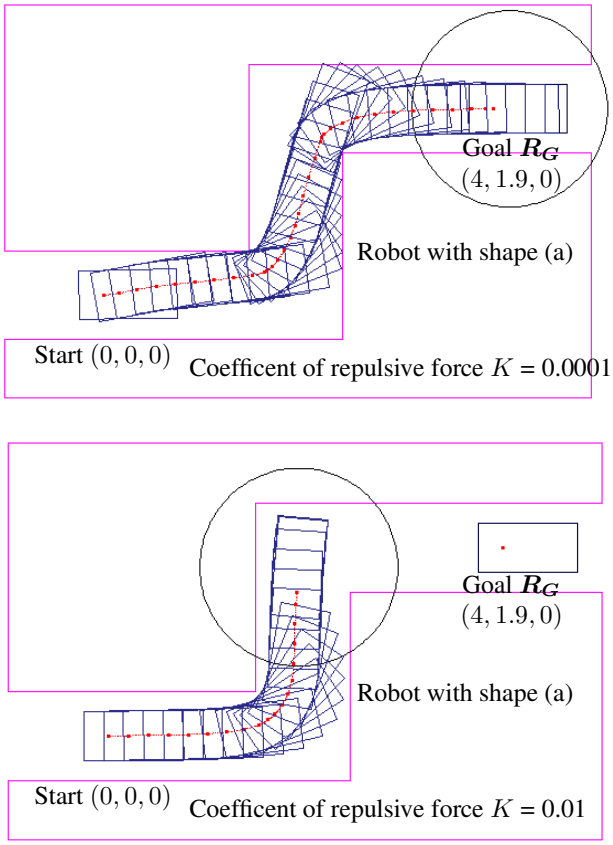

Figure 14. Simulation results by using various coefficent of repulsive force.

are generated at either front or rear point of action on the robot and their forces are treated like a lever. Both motion constraint and shape of a robot can be considered by this simple idea. Simulation results for various situations have proven effectiveness of our algorithm. Although this method has a disadvantage of local minima as well as general potential field method, it is intended for practical use because adequate path for local obstacle avoidance can be obtained with a little computing power. Furthermore, this algorithm may be applied to not only mobile robots with two independently driven wheels but also car-like robots. Implementation for autonomous wheelchairs and consideration about general shape of mobile robots are remained for our further works.

References

[1] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.

[2] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. of Robotics Research*, 5(1):90–98, 1986.

[3] K. Kondak and G. Hommel. Computation of time optimal movements for autonomous parking of non-holonoimic mobile platforms. *Proc. of 2001 IEEE Int. Conf. on Robotics and Automation*, pages 2698–2703, 2001.

[4] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[5] J. P. Laumond, P. E. Jacobs, M. Taix, and R. M. Murray. A motion planners for nonholonomic robots. *IEEE Trans. on Robots and Automation*, 10(5):577–592, 1994.

[6] C. Liu, H. Marcelo, K. Hariharan, and S. Lim. Virtual obstacle concept for local-minimum-recovery in potential-field based navigation. *Proc. of 2000 IEEE Int. Conf. on Robotics and Automation*, pages 983–988, 2000.

[7] J. Minguez, L. Montano, and J. Santos-Victor. Abstracting vehicle shape and kinematic constraints from obstacle avoidance methods. *Autonomous Robots*, 20:43–59, 2006.

[8] G. Ramirez and S. Zeghloul. Collision-free path planning for nonholonomic mobile robots using a new obstacle representation in the velocity space. *Robotica*, 19:543–555, 2001.

[9] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Trans. on Robotics and Automation*, 8(5):501–518, 1992.

[10] J. T. Schwartz and M. Sharir. On the piano movers' problem: I. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics*, 36:345–398, 1983.

[11] M. Strobel. Navigation in partially unknown, narrow, cluttered space. *Proc. of 1999 IEEE Int. Conf. on Robotics and Automation*, pages 28–34, 1999.

[12] I. Ulrich and J. Borenstein. Vfh*: Local obstacle avoidance with look-ahead verification. *Proc. of 2000 IEEE Int. Conf. on Robotics and Automation*, pages 2505–2511, 2000.