

Virtual Obstacle Concept for Local-minimum-recovery in Potential-field Based Navigation

Liu Chengqing*, Marcelo H Ang Jr*, Hariharan Krishnan*, Lim Ser Yong**

*Department of Mechanical and Production Engineering
National University of Singapore
mpeangh@nus.edu.sg

** Gintic Institute of Manufacturing Technology
71 Nanyang Drive
Singapore 638075

Abstract

We present a navigation algorithm, which integrates virtual obstacle concept with a potential-field-based method to maneuver cylindrical mobile robots in unknown or unstructured environments. This study focuses on the real-time feature of the navigation algorithm for fast moving mobile robots. We mainly consider the potential-field method in conjunction with virtual obstacle concept as the basis of our navigation algorithm. Simulation and experiments of our algorithm shows good performance and ability to overcome the local minimum problem associated with potential field methods.

1 Introduction

Autonomous navigation of mobile robots is widely recognized as a fundamental research issue and is of interest to many researchers. Among the two main navigation approaches, reactive control and planning control, planning control is the more widely used [3,10]. In reactive control, the robot moves around while avoiding collision using its perceptual system to gather information about its environment. In planning control, the robot motion is generated at the planning level and the robot executes the planned movement. In general, the robot executes this movement with no perceptual feedback from the environment. Important path planning results have been derived mathematically, yet few have been fully implemented for real-time navigation.

The potential-field method [6,9] is one of the most popular approaches used to navigate mobile robots within environments containing obstacles. Potential field methods treat the robot, represented as a point in configuration space, as a particle under the influence of an artificial potential field whose local variations are expected to represent the "structure" of the free space. The potential function is typically defined over free space as the sum of an attraction potential pulling the robot toward the goal configuration, and a repulsive potential pushing the robot away from the obstacles. Motion planning is performed in an iterative fashion. At each iteration, the artificial force induced by the potential function at the current configuration is regarded as the most promising direction of motion, and path generation proceeds along this direction by some increment.

Potential field was originally developed as an online collision avoidance approach, applicable when the robot does not have a-priori model of the obstacle, but senses them during motion execution. Emphasis was put on real-time efficiency, rather than guaranteeing the attainment of the goal. In particular, since an on-line potential field method essentially acts as a fast descent optimization procedure, it may get stuck at a local minimum of the potential function other than the goal configuration [1,8,10].

A variety of potential functions have been proposed in the literature. The most interesting of them are aimed at either one of these two goals: (1) improving the "local dynamic behaviour" of the robot along the generated paths [2,3,5]; (2) reducing the number of local minimum and/or the size of their attractive wells [4,7,11]. Here we review the interesting concept of "Rotational-Potential" [2,5]. The motivation is to ameliorate the behaviour and the movement of the robot by a dynamic adaptation of the parameters of avoidance.

In the classical method, the repulsive force acts only inside a zone defined by the clearing distance. In practice, the choice of this distance is delicate. One commonly used expression for repulsive force is:

$$F_o(X) = \begin{cases} K_o \left(\frac{1}{d} - \frac{1}{d_o} \right) \frac{1}{d^2} \frac{\partial d}{\partial X} & \text{if } d < d_o \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In the equation (1), d is the distance between the robot and obstacle, d_o is the clearing distance, the term $\frac{\partial d}{\partial X}$ represents the unity vector of the partial derivative vector. A small value of d_o induces a vigorous variation of the repulsive force and imposes constraints on the speed of the robot. A large value of d_o may over constrain the movement in places where the robot may pass.

In order to determine this distance d_o more easily, the following idea was proposed. At first fix a minimal clearing distance with a repulsive potential function which insures non-collision in the region defined by it. Then define another potential for a distance greater than the minimum distance that tries to guide the robot and correct its movement if necessary [2,5]. The essence of this method is that the robot is repelled by an obstacle only if it is close to the obstacle and its velocity points toward the obstacle. If the robot moves parallel to the obstacle, it does not have to be repelled by it.

Such potential functions are interesting, but the problem of local minimum still exists. The most crucial drawback of this and other local obstacle avoidance methods has been that the mobile robot may become trapped in a local minimum. The study of local-minimum-free navigation function is still beyond the reach of real implementation. Since the potential-field is widely used in the robotics community, it is important to find an adequate and efficient solution for the local minimum problem of fast-moving robots while retaining the advantages of the potential-field method.

The former algorithms often fails in two cases "concave obstacle" or "concave-shaped obstacle." (Although the obstacles are convex, several convex-obstacles may together form a concave-shaped obstacle.) From our empirical experimentation, it was observed that concave obstacles usually result in local minimum. We present here a navigation algorithm that allows a fast-moving autonomous mobile robot to maneuver successfully in unknown or unstructured environments. The algorithm integrates our novel "virtual obstacle" concept with the potential-field method to navigate mobile robots. With this method, no assumption is required about the form of the obstacles (e.g., obstacles need not be convex).

Using this method the mobile robot will recover from local minimum in "real time". As the robot is guided towards a target based on the potential-field method, it maneuvers to avoid obstacles in its immediate path. When the robot approaches or encounters concave or concave-shaped obstacle where a local minimum is most likely to occur, virtual obstacle method is immediately invoked to compute intermediate via points to be used as temporary path targets. The robot continues on its journey by heading towards the closest intermediate path target under potential-field control. During the journey, a map is almost concurrently constructed based on local ultrasonic signal feedback.

2 Virtual Obstacle Concept

In this section, we describe in detail our virtual obstacle concept. At first, the mobile robot uses the detected distance readings to the obstacles based on ultrasonic sensors to construct boundary lines of the real obstacles. Then the following steps are performed.

Step1: Select the line segment which is nearest to the mobile robot. We label this line as L1.

The distance of a point to a line segment is defined as shown in Fig.1.

If the point is somewhere along the "middle" of the line segment (i.e. both two angles α_1 and α_2 are between 0° and 90°), then its distance to the line segment is the perpendicular distance to the line segment. If this point is further to the left end of the line segment (i.e. only α_1 greater than 90°) or further to the right end of the line segment (i.e. only α_2 greater than 90°), the distance of

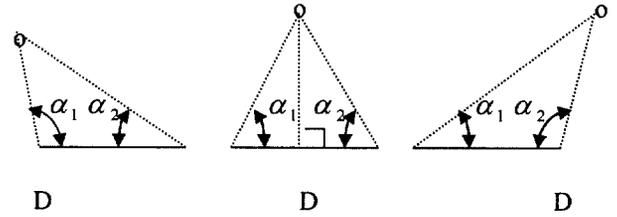


Fig 1: The distance between a point and a line.

the point to line segment is defined as the length of the lines which connect this point to the left end or right end respectively of the line segment. In Fig.1, OD is the distance between this point to the line segment. Using this distance definition, the nearest line segment to the mobile robot is determined.

Step2: Select another line segment which is nearest to L1 and at the same time on the other side of the mobile robot. Label this line segment as L2.

The distance between L1 and another line segment is the smaller distance between L1's ends (two points) and the line segment. The distance is used to select the line segment L2 nearest to L1. Two line segments being at different sides of the mobile robot can be determined by the following method.

Connect the ends of the line segments to the mobile robot. Define the angles between the ends of L1 to the robot as β_1 and β_2 . Define the angles of L2's two ends to the mobile robot as β_3 and β_4 as shown in Fig. 2. These angles are transformed to be in the domain ($0^\circ \sim 360^\circ$). Then define:

$$\begin{aligned} \beta_{\min} &= \min(\beta_1, \beta_2) \\ \beta_{\max} &= \max(\beta_1, \beta_2) \end{aligned} \quad (2)$$

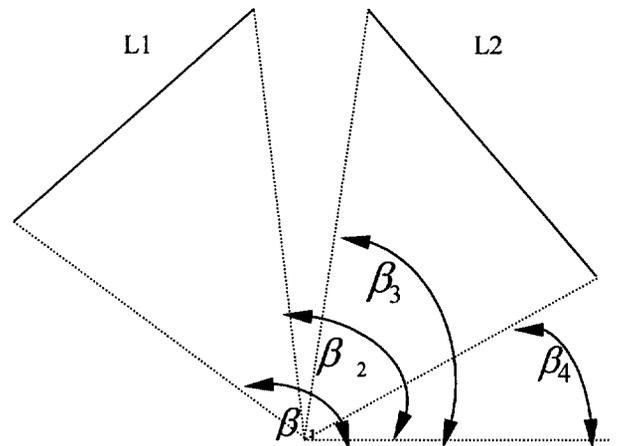


Fig 2: The angles between the lines and the robot

In the following two cases, the line segments L1 and L2 are on 2 different sides of the mobile robot.

Case 1: ($\beta_{\max} - \beta_{\min} < 180^\circ$ AND ($\beta_3 \geq \beta_{\max}$ OR $\beta_3 \leq \beta_{\min}$) AND ($\beta_4 \geq \beta_{\max}$ OR $\beta_4 \leq \beta_{\min}$))

Case2: $(\beta_{\max} - \beta_{\min}) > 180^\circ$ AND
 $(\beta_{\min} \leq \beta_3, \beta_4 \leq \beta_{\max})$

Step 3: Determine whether the robot is inside a concave obstacle or not.

Step 3.1: Connect the ends of the two line segments L1, L2 thereby creating four line segments V1, V2, V3, V4 (dashed lines shown in Fig. 3). Then select the shortest line VL1 as virtual line segment, assume the two ends of VL1 is point A and point B, then connect the remaining two ends (point C and point D), we refer to this line segment as the virtual line segment VL2.

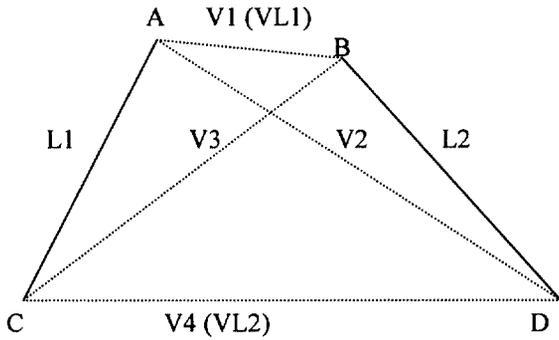


Fig 3: The method to determine virtual line

Step 3.2: As shown in Fig. 4, connect the ends of the real line segments (L1 and L2) and virtual line segments (VL1 and VL2) to the mobile robot. Four angles are created, namely $\gamma_1, \gamma_2, \gamma_3$ and γ_4 . The angle of the robot and the two ends of L1 is defined as γ_1 , the angle of the robot and the two ends of L2 is defined as γ_2 , the angle of the robot and the two ends of VL1 is defined as γ_3 , the angle of the robot and the two ends of VL2 is defined as γ_4 .

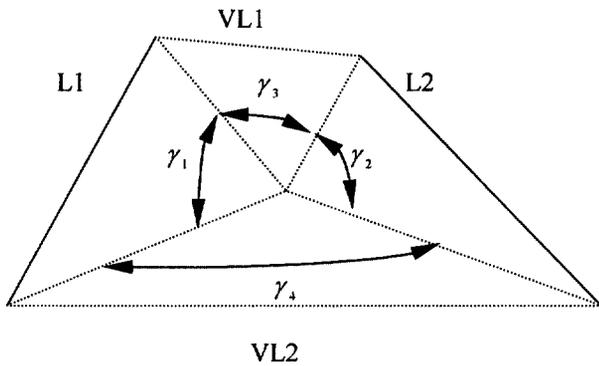


Fig 4: The angles between the robot and the real line segments and virtual line segments

If the sum of these four angles is 360° and the length of VL1 is less than 1.2 times the diameter of the mobile robot then the mobile robot is inside a concave-shaped obstacle.

Step 4: If the mobile robot is inside the concave obstacle, then the mobile robot must navigate out of the obstacle first. The mobile robot will go to free point K

defined according to following method as shown in Fig. 5.

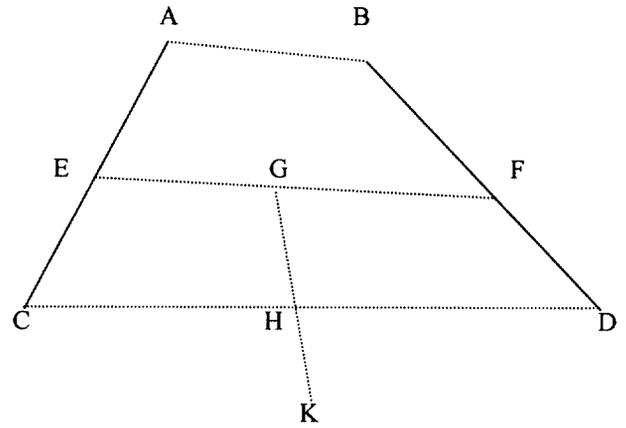


Fig 5: The method to decide free point K

Select the mid point (point E) of line segment AC, select the midpoint of line segment BD (point F), select the mid point of line segment EF as point G, select midpoint of line segment CD as point H. We then connect point G to H and then to K and let the length of HK be greater than 2 times the diameter of the mobile robot. The robot will go to point K first to avoid a local minimum. During this time, the mobile robot will forget its ultimate goal, i.e., the ultimate goal doesn't exert an attractive force to the mobile robot when it is moving to K.

Step 5: Once the robot reaches point K, the ultimate goal resumes its attractive force to the mobile robot. But the method to calculate the distance between the mobile robot and the obstacle will be changed.

First, we explain how to calculate the distance between the mobile robot and the virtual line segments. For example, if CD is a virtual line segment as shown in Fig. 6, RH is the direction of one ultrasonic sensor of the mobile robot, it is easy to calculate the intersection point, H. (The angle of the ultrasonic sensor is known). Another important condition is that this intersection point must be between the two ends of virtual line segment CD. The method is as follows, connect the ends of the virtual line segment CD to the mobile robot's position. Define the angle of line segment RD's direction as λ_1 , the angle of line segment RC's direction as λ_2 , the angle of this sensor's direction as λ_3 . All these three angles are transformed to $(0^\circ \sim 360^\circ)$. Then define:

$$\lambda_{\min} = \min(\lambda_1, \lambda_2)$$

$$\lambda_{\max} = \max(\lambda_1, \lambda_2) \quad (3)$$

In the following two cases, the intersection point (H) is between the two ends of virtual line segment CD.

$$\text{Case 1: } \lambda_{\max} - \lambda_{\min} < 180^\circ \text{ AND } (\lambda_{\min} < \lambda_3 < \lambda_{\max})$$

$$\text{Case 2: } \lambda_{\max} - \lambda_{\min} > 180^\circ \text{ AND } (\lambda_3 < \lambda_{\min} \text{ OR } \lambda_3 > \lambda_{\max})$$

Then the length of RH is defined as the distance between the mobile robot and the virtual line segment in this ultrasonic sensor's direction.

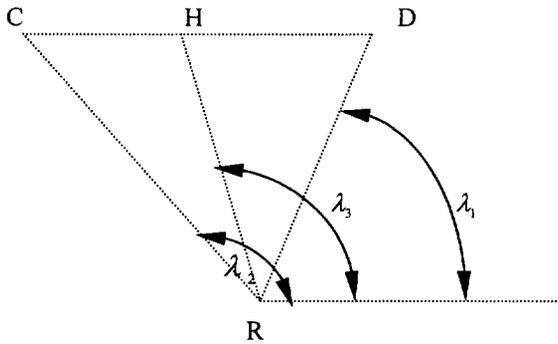


Fig 6: The distance between the robot and the virtual line segment in an ultrasonic sensor direction

For example as shown in Fig. 7, suppose one ultrasonic sensor of the mobile robot detects an obstacle's distance as the length of RE, at the same time, the mobile robot calculates the distance of the mobile robot to the intersection point H (point H must be between the two ends of virtual line segment CD) of the virtual obstacle CD in this ultrasonic sensor's direction. We compare the distances of RE and RH and select the smaller (RH) as the obstacle distance. Similarly, in another ultrasonic sensor's direction, we compare another sensor's detected distance RG and the distance (RM) between the mobile robot and intersection point M (point M must be between the two ends of virtual line CD) then select the smaller (RM) as the obstacle distance.

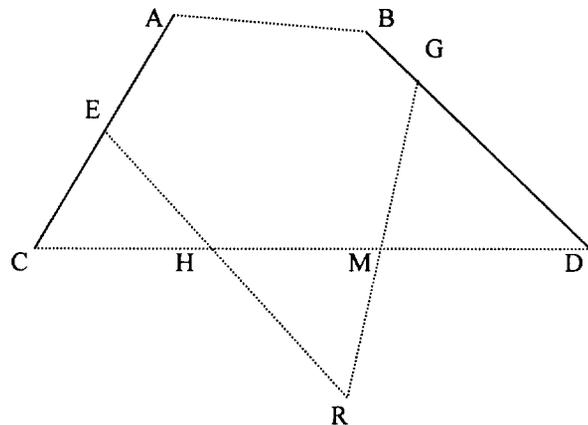


Fig 7: The method to decide calculating distance

3 Application

In this section, we apply the virtual obstacle concept to different goal-obstacle configurations. These configurations pose local minimum difficulties for standard potential field based methods.

Fig. 8 is the obstacle environment. The goal position of the mobile robot is (2200,2200) (the units are 0.1 inch). Fig. 9 and Fig. 10 are the paths of the robot using different algorithms. In Fig. 9, virtual obstacle concept is used. In Fig. 10, only standard potential field method is used. In Fig. 9, it can be seen that the robot doesn't get trapped in a local minimum. At first the mobile robot moves according to the potential field method. During the move, the mobile robot uses the least square method to form line segments representing obstacle boundaries based on readings from the ultrasonic sensors. When it encounters a concave shape obstacle, a virtual obstacle represented by a virtual line is created. Then it gets out of the concave shape obstacle. After that, in every ultrasonic sensor's direction, the mobile robot compares the real distance detected by the ultrasonic sensor and the distance to the virtual obstacle line segments in the direction of the ultrasonic sensor, then selects the smaller as the obstacle distance. It can be seen that the mobile robot successfully reaches the goal.

Using the same environment and the same goal position as in Fig. 9, only using the standard potential field method, the mobile robot gets trapped in a local minimum as shown in Fig. 10.

Fig. 11 illustrates a complicated environment where the mobile robot reaches the goal position using virtual obstacle algorithm.

4 Conclusion

Our virtual obstacle concept was developed to address the local minimum problem associated with potential field method. Through extensive testing and simulations, we found that the virtual obstacle concept is an effective method to deal with this problem.

Acknowledgements:

We are grateful for the stimulating discussions with Prof. Oussama Khatib, Stanford University. The support of Gintic Institute of Manufacturing Technology for this research is gratefully acknowledged.

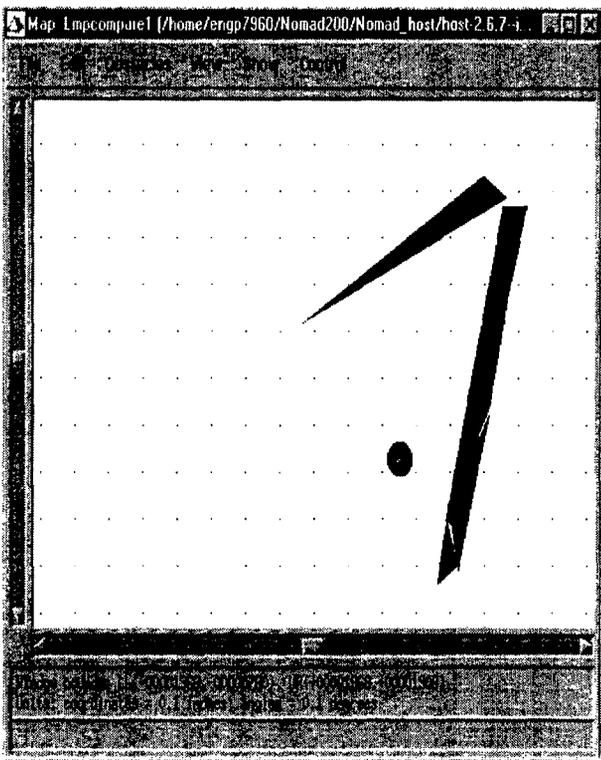


Fig 8: The mobile robot and the obstacle

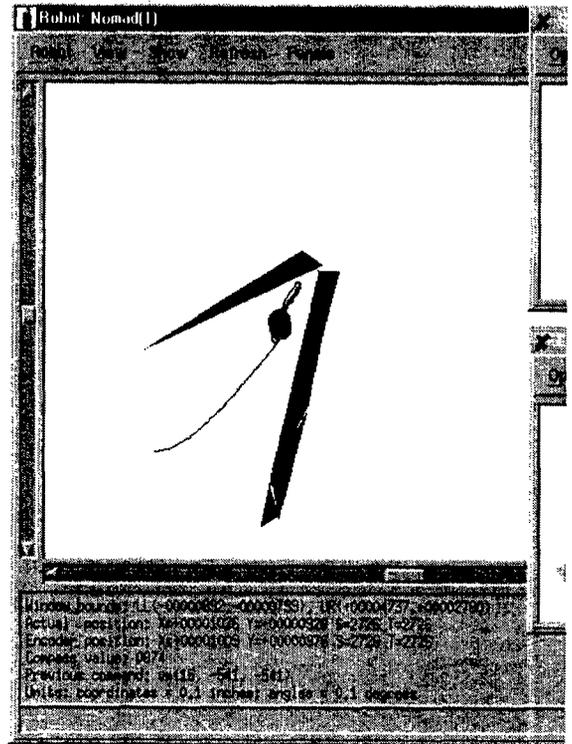


Fig10: The path generated by the robot using standard potential field method

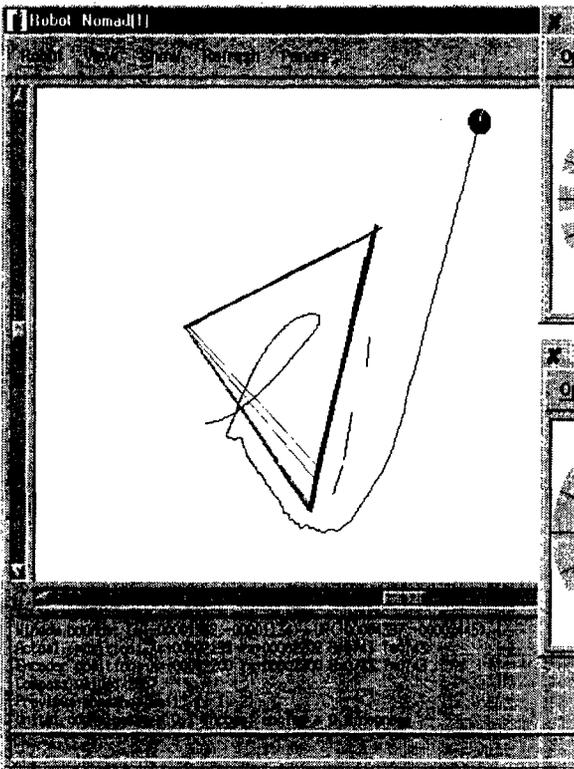


Fig9: The path generated by the robot using virtual obstacle concept and the environment detected by the sensor

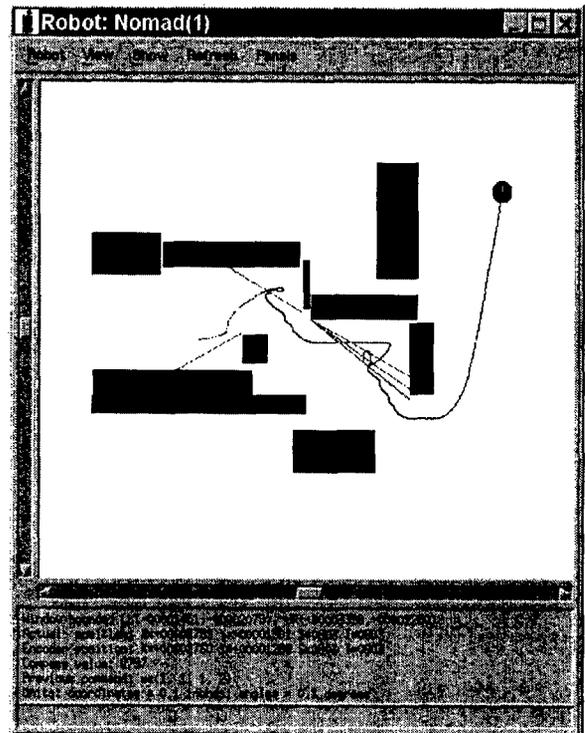


Fig11: The path generated by the robot using virtual obstacle concept in a complicated environment

References:

- [1] J. Borenstein, Y. Koren, "Real-time obstacle avoidance for fast mobile robots", IEEE Transactions on Systems, Man and Cybernetics, Vol. 19, No. 5, pp. 1179-1187, 1989.
- [2] H. Haddad, M. Khatib, S. Lacroix and R. Chatila, "Reactive navigation in outdoor environments using potential fields", Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Leuven, Belgium, pp. 1232-1237, 1998.
- [3] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots", Proceedings of the IEEE International Conference on Robotics & Automation, Scottsdale, pp. 2-7, May 1989.
- [4] K. Sato, "Deadlock-free motion planning using the Laplace potential field", Advanced Robotics, Vol. 7, No. 5, pp.449-461, 1993.
- [5] M. Khatib, "Sensor-based motion control for mobile robots", PhD thesis, LAAS-CNRS, Toulouse, France, December 1996.
- [6] O. Khatib., "Real-time obstacle avoidance for manipulators and mobile robots", Proceedings of the IEEE International Conference on Robotics & Automation, , pp. 500-505, 1985.
- [7] J.O. Kim, P. K.Khosla, "Real-time obstacle avoidance using harmonic potential functions", IEEE Transactions on Robotics & Automation, Vol. 8, No. 3, pp.338-349, 1992.
- [8] Y. Koren, J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation", Proceedings of the IEEE Conference on Robotics and Automation, Sacramento, California, April 7-12, pp.1394-1404, 1991.
- [9] B. H. Krogh, "A generalized potential field approach to obstacle avoidance control", Proceedings of the International Robotics Research Conference, pp.1150-1156, 1984.
- [10] J. C. Latombe, "Robot Motion Planning", Kluwer Academic Publishers, Boston/Dordrecht/London: 1991.
- [11] El. Rimon, D. E. Kodischek, "Exact robot navigation using artificial potential functions", IEEE Transactions on Robotics & Automation, Vol. 8, No. 5, pp.501-517, October 1992.