Robotica (2008) volume 26, pp. 285–294. © 2007 Cambridge University Press
doi:10.1017/S026357470700389X Printed in the United Kingdom

Modified Newton's method applied to potential field based navigation for nonholonomic robots in dynamic environments

Jing Ren†, Kenneth A. McIsaac*‡ and Rajni V. Patel‡

†*Faculty of Engineering and Applied Science, University of Ontario Institute of Technology, Oshawa, Ontario, Canada 1 LH 7K4*
‡*Department of Electrical and Computer Engineering, University of western Ontario, London, Ontario, Canada N 6 G 1 H 1*

(Received in Final Form: August 15, 2007. First published online: November 7, 2007)

SUMMARY
This paper is to investigate inherent oscillations problems of Potential Field Methods (PFMs) for nonholonomic robots in dynamic environments. In prior work, we proposed a modification of Newton's method to eliminate oscillations for omnidirectional robots in static environment. In this paper, we develop control laws for nonholonomic robots in dynamic environment using modifications of Newton's method. We have validated this technique in a multirobot search-and-forage task. We found that the use of the modifications of Newton's method, which applies anywhere C_2 continuous navigation functions are defined, can greatly reduce oscillations and speed up robot's movement, when compared to the standard gradient approaches.

KEYWORDS: Potential field methods; nonholonomic constraints; mobile robotics.

1. Introduction

The potential field method is widely used in mobile robot navigation for its simplicity and elegance.[1–3] In the potential field method (PFM), a scalar field is constructed of artificial "hills" and "valleys" that represent attractive and repulsive locations in the workspace. Control laws are then usually generated along gradient directions of this scalar field. However, this approach has some significant inherent limitations.

Koren *et al*.[4] identified the problem of oscillations that can occur in potential fields that can result in oscillations (unstable motion) when the discrete system becomes unstable. As a result, solutions tend to exhibit oscillation, especially in proximity to obstacles or in narrow passages. To achieve stability, in practice, this phenomenon means that either step sizes (controller sampling rates) must be small or robot speeds must be slow. Also, gradient descent methods tend to complete their tasks slowly, as many "unnecessary" controls are generated. In our prior work,[5] we further investigated the sources of oscillations and showed that even when the discrete system is stable, oscillations can result from the oscillatory trajectory generated by the gradient descent method, and in fact roughly correspond to "oscillations" defined in the optimization theory. It is well known in optimization theory[6] that the gradient method exhibits oscillation and slow convergence in the neighborhood of the optimal point. This phenomenon results from the fact that the objective function is often a complex nonlinear function and the gradient method, which contains only linear information, cannot achieve a good approximation to the objective function. Similarly, in potential field based navigation, whenever the potential field is severely deformed and takes on a strongly noncircular shape due to a complex configuration of targets and obstacles, especially when robots enter narrow passages, the navigation function in that neighborhood becomes a complex nonlinear function. The use of standard gradient method will result in poor performance and oscillating steps.

In our prior work, we proposed a modification of Newton's method to overcome this oscillation problems for omnidirectional robots in planar and static environments.[5] Modified Newton's method (MNM), which contains both first and second derivative information, can greatly alleviate oscillation by approximating the navigation function as a quadratic form. This method retains all the merits of the potential-field approach, while at the same time eliminating oscillating, achieving faster task completion and allowing a bigger step size (faster speed).

In this paper, we extend the MNM and investigate its use in two situations that can complicate the use of potential field methods: robots that move subject to the non-slip constraint and dynamic environments that include moving obstacles.

The word "nonholonomic"[7,8] refers in general to nonintegrable constraints on robot motion that constrain the velocity of the system, but not its position. The most typical nonholonomic constraint encountered in mobile robotics is the nonslip constraint that forbids wheeled systems from making translations "sideways" (in directions perpendicular to the robot's heading). One common example of a wheeled mobile robot subject to the nonslip constraint is the Hilare mobile robot created at Laboratorie d'Automatique et d'Analyse des Systemes (LAAS), located in Toulouse, France.[9] This robot has two parallel wheels which can be controlled independently. By commanding the same velocity to both wheels, the robot moves in a straight line. By commanding velocities with the same magnitude but opposite directions, the robot pivots about its axis. Strictly speaking, the gradient descent approach for potential fields cannot be used with a wheeled robot, because the gradient descent direction in general, will not be parallel to the

* Corresponding author. E-mail: kmcisaac@engga.uwo.ca

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press

robot's heading direction, and will therefore be forbidden by the constraint. Tanner and Kyriakopoulos[10] introduced the dipolar inverse navigation function to allow the use of gradient descent methods for wheeled mobile robots. In this paper, we develop a corresponding control law using MNM and show that the MNM can reduce oscillations when compared to Tanner's approach.

The second problem we consider is the problem of dynamic environments that include moving and unmodeled obstacles. As was the case for the wheeled robot subject to the non-slip constraint, the presence of an obstacle that is not modeled in the potential field will forbid the mobile robot to choose the gradient descent direction. Esposito and Kumar[11] proposed a method of "feasible directions" based on a solution to an optimization problem to select a motion direction for the robot that would follow the gradient direction as closely as possible while avoiding the moving obstacles. In this paper, we adapt Esposito's technique to illustrate the use of the MNM for moving obstacles.

This paper is organized as follows: In Section 2, we review Tanner and Kyriakopoulos[10] approach for wheeled robots and present an extension of the MNM for nonholonomic robots. In Section 3, we review Esposito and Kumar[11] approach for unmodelled obstacles and show how we extend this approach using the MNM. We also investigate the performance of MNM with changes in different parameters. Then, in Section 4, we present a new technique, based on geometric transformations in the plane, in which we use the MNM to determine control laws for mobile robots that must contend simultaneously with the nonslip constraint and the presence of unmodeled obstacles. In Section 5, we compare the performance of MNM method and the gradient method in a search-and-rescue task. Finally, in Section 6, we conclude the paper and suggest some future research directions.

2. Applying the Modified Newton's Method to Robots with Nonslip Constraints

For the purposes of development in this paper, we represent the configuration of planar mobile robots with the position vector $q = (x, y)$ of the center of the robot and the orientation angle θ. We also associate a direction vector d with the robot's orientation. Since the Hilare system is kinematically constrained, it can be fully controlled by specifying the angular and linear velocities of the robot, so we can treat it as a kinematic control problem.

In this section, we consider the dipolar inverse navigation function proposed by Tanner *et al.*[10,12,13] We will develop a corresponding MNM control law and compare the performance of the two control laws. Dipolar functions behave like reciprocal of the conventional function, it tends to infinity at the goal and is zero at obstacles, accordingly it uses gradient direction instead of negative gradient direction for motion generation. In this paper, in order to be consistent with other part of the paper, we choose to use negative gradient direction and changed formula and derivations accordingly. Note that all changes made are changes in notation and do not affect the results.

In ref. [12], the navigation function is defined as,

$$V(z) = \frac{\|x\|\beta^{\frac{1}{k}}}{\|z\|^2} \tag{1}$$

where $z = (x, y, \theta)$ and $\|z\|^2 = x^2 + y^2 + \lambda\theta^2$, $\beta(z)$ is the product of distance from the robot to obstacles and λ and k are positive constants.

For this development, we define the negative gradient $d_{ng} = -\frac{\partial V}{\partial z} \triangleq [d_{ngx}, d_{ngy}, d_{ng\theta}]^T$, $d_\theta \triangleq [\cos\theta, \sin\theta]^T$, $d_{ngxy} \triangleq [d_{ngx}, d_{ngy}]^T$ and

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{2}$$

The control law generates the linear velocity as

$$v = -k_v\, sgn\left(d_\theta^T d_{ngxy}\right)\left(d_{ngx}^2 + d_{ngy}^2 + d_{ng\theta}^2\right)\|z\|^4 \tag{3}$$

and the angular velocity as

$$\omega = \begin{cases} k_\theta(\theta_d - \theta) & \text{if } \varpi(z) \geq 0 \\ -v\left(d_\theta^T d_{ngxy}\right)d_{ng\theta}^{-1} & \text{if } \varpi(z) < 0 \end{cases} \tag{4}$$

where k_θ and k_v are constants and

$$\varpi(z) = -v d_\theta^T d_{ngxy} - k_\theta d_{ng\theta}(\theta_d - \theta) \tag{5}$$

and the desired orientation is calculated by

$$\theta_d = arctan\, 2(sgn\,(x)d_{ngy}, sgn(x)d_{ngx}) \tag{6}$$

while the sign function is defined as

$$sgn\,(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ -1, & \text{if } x < 0 \end{cases} \tag{7}$$

In plain language, this approach drives the robot either forward or backwards, depending on which of these two directions has a positive projection on the gradient direction, then generates a change in orientation to align the robot's orientation with the gradient direction.

2.1. Improving the performance with MNMs

In this section, we revise the above control law using MNM, the modified Newton direction is defined as,

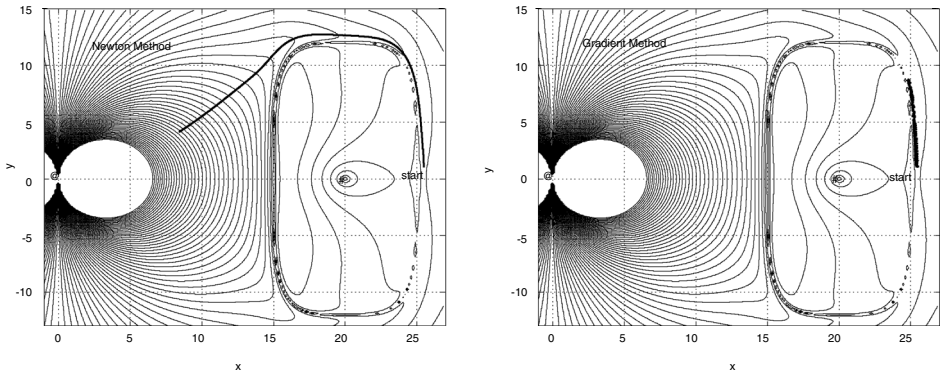
$$d_n = B d_{ngxy} = (d_{nx}, d_{ny}) \tag{8}$$

where $B = (\varepsilon I - H)^{-1}, \varepsilon \geq 0$.[6] Note that this dipolar inverse navigation function corresponds to a maximization problem, so we use negative Hessian matrix $(-H)$ of the navigation function $V(z)$. Other parts of derivation are the same as in the previous section. In this case, the desired orientation is calculated based on modified Newton direction d_n instead of negative gradient direction d_{ng}

$$\theta_d = arctan\, 2(sgn\,(x)d_{ny}, sgn\,(x)d_{nx}) \tag{9}$$

Theorem 1: Consider a system of the form (2) and assume the existence of a dipolar inverse Lyapunov function (DILF)-generted potential field. Then the control law

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press



(a) (b)

Fig. 1. (a) MNM and (b) gradient method.

Table I. Parameters.

k_v	k_θ	start x	start y	start θ
1	26	26	0.6	90°

Table II. Parameters.

k_v	k_θ	start x	start y	start θ
1	26	26	0.01	90°

(Eqs. (3)–(5),(9)) guarantees obstacle avoidance and global asymptotic convergence for the system.

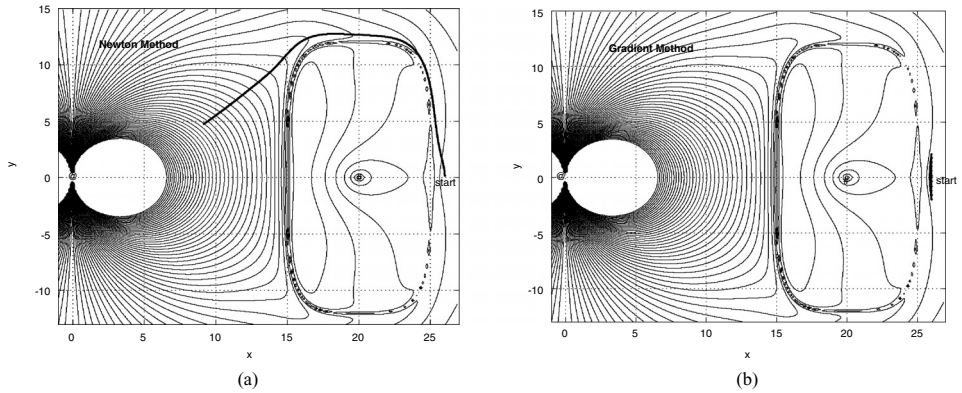
Proof: See the Appendix.

In Fig. 1, we show the performance of gradient method and Newton's method in a typical obstacle avoidance task. To facilitate the discussion, here and elsewhere in this paper, we adopt the notion "stepsize" from optimization theory[6] and define it as the product of the speed and the sampling interval. The solid dark line represents the trajectory of the robot, the target is represented by the symbol @ and the obstacle is represented by the symbol #. This figure illustrates that the MNM can achieve a much faster progress in presence of obstacles. MNM almost reached the goal after 150 steps, while the gradient method hardly made any progress. The parameters we used for this comparison is listed in Table I.

It is well-known[3] that when the starting point, the obstacle and the target are perfectly aligned, it will create a saddle point on the side of the obstacle away from the target. Perfect alignment rarely occurs in practice, however, rough alignments happen frequently and are the main cause of oscillations. By comparing the performance of gradient method and MMN with a rough alignment starting point, we show in Fig. 2 that MNM is more robust to starting conditions and can maintain much better performance. In the case shown, the performance of MNM is not affected much by moving the starting point, however, the performance of gradient method deteriorates substantially. The parameters we used for this comparison is listed in Table II.

Because it is more reasonable for mobile robots to have limit on linear velocity, we will use Tanner's method but put a upper limit to the velocity and stepsize in the following discussions. Note that the benefit of using NMN is independent of the choice of fixed or changing or saturated linear velocity.[5]

In PFM applications, oscillations occur independent of obstacle shapes. The results derived in this paper can be applied to obstacles of arbitrary shapes. To illustrate this, in this section, we compare the two methods in a task of passing through a passage. In Fig. 3(a), (b), we set the width of the passage to be 1.5 m, we can see that both the gradient and MNM move fast and the trajectories are smooth. When we narrow the passage down to 0.5 m, MNM in Fig. 3(d) can still maintain a fast and smooth movement. The gradient method, however, exhibits a large amount of oscillation which is shown in Fig. 3(c). We have magnified Fig. 3(c), (d) and



(a) (b)

Fig. 2. (a) MNM and (b)gradient method.

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press

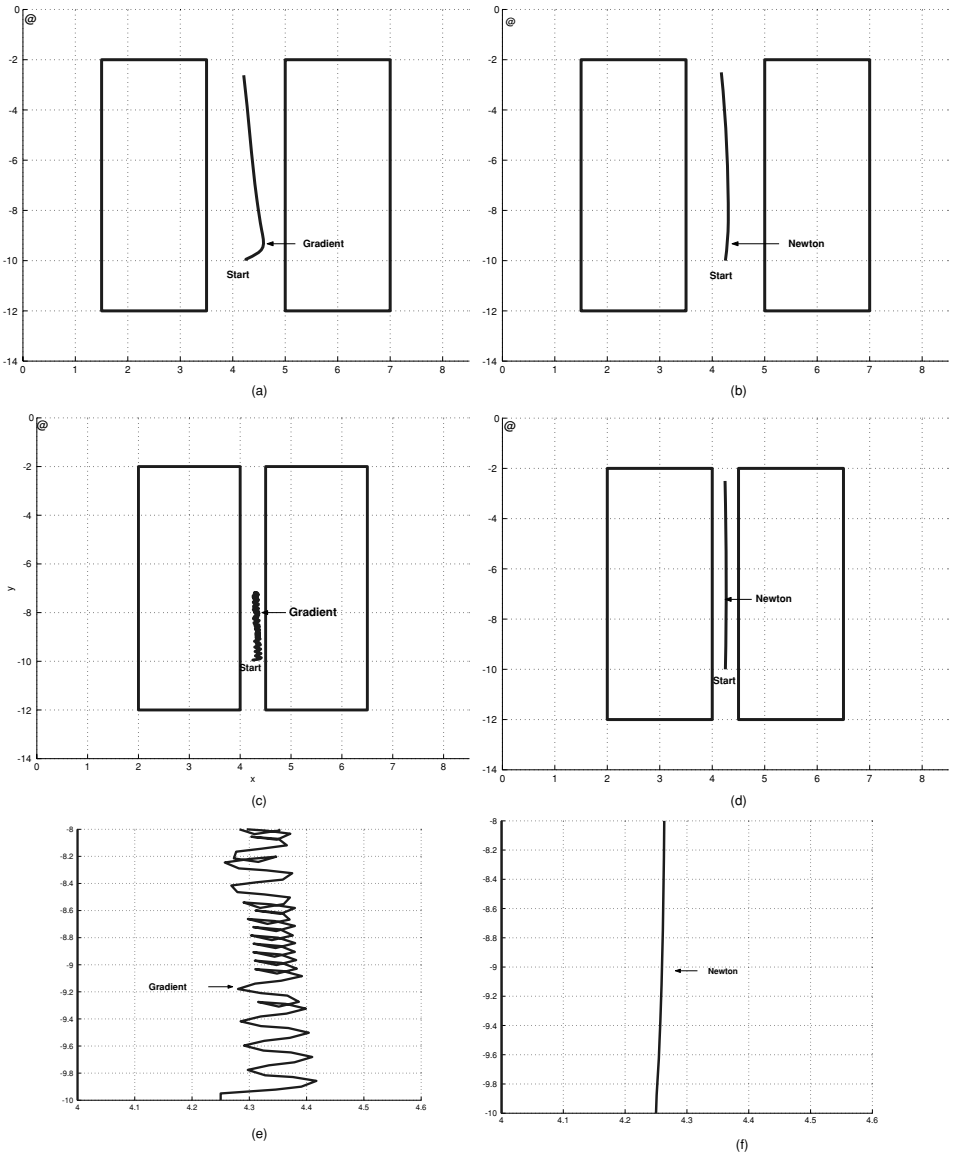


Fig. 3. (a) and (c): Performance of the gradient method in wide and narrow passages, respectively, (b) and (d): Corresponding trajectories for MNM in the same scenarios, and (e) and (f): Parts of the enlarged version of (c) and (d). Totally 150 steps.

Table III. Parameters.

k_v	k_θ	width	stepsize
1	26	1.5(wide)/0.5(narrow)	0.05

shown parts of the enlarged versions in Fig. 3(e), (f). The parameters we used for this comparison is listed in Table III.

3. MNM for Robots in a Dynamic Environment

In this section, we first recall an optimization-based gradient method which treats the moving obstacles as dynamic constraints.[11] Then, we show how to formulate our modified Newton's method into an optimization problem and solve the motion planning problem in dynamic environments. Once again, we show that the MNM can also greatly improve the performance in dynamic environments.

3.1. Model of moving obstacles

In the approach defined in ref. [11], the movements of the moving obstacles are assumed to be unpredictable. When moving obstacles are far away from robots, they are ignored by the planning algorithm. However, when the robot moves within a protective distance D_{obs} of a detected obstacle, we associate with the obstacle a dynamic constraint of the form:

$$g_j(q) = |q - P_j|^2 - D_{\text{obs}}^2 \geq 0. \tag{10}$$

where P_j is the position of the moving obstacle j. These constraints will be used in our control law to generate feasible directions of motion which move the robot away from the obstacle while still making progress towards the goal.

3.2. Definition of feasible motion direction

The basic idea of motion planning in ref. [11] is to choose a best direction from all feasible directions. In the context of moving obstacles avoidance, feasible motion directions are the directions that can stabilize the system and also move

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press

robots away from the activated constraints caused by moving obstacles.

We use $d_{g_j} = \frac{\partial g_j}{\partial q}$ to represent the gradient of the activated constraint g_j. A feasible motion direction d_f is defined as the direction that satisfies the following conditions:

$$d_f^T d_{ng} > 0 \tag{11}$$

$$d_f^T d_{g_j} > 0, \quad j = 1, 2, \ldots m \tag{12}$$

In Eq. (11), we require any control which can stabilize the system to have a positive projection on d_{ng}. In Eq. (12) we require any control which can move robot away from the activated obstacle to have a positive projection on d_{g_j}.[11] Feasible directions must satisfy both conditions at once.

3.3. Optimization method using gradient method

If we denote navigation function as $V(q)$, the control law $Z(q)$ defined by Esposito[11] is as follows:

$$Z(q) = -\left(1 - \sum_{i=1}^{n-1} \beta_i^2\right)^{\frac{1}{2}} \frac{\partial V(q)}{\partial q} + \sum_{i=1}^{n-1} \beta_i \left[\frac{\partial V(q)}{\partial q}\right]_i^{\perp} \tag{13}$$

subject to

$$\sum_{i=1}^{n-1} \beta_i^2 < 1 \tag{14}$$

where $\frac{\partial V(q)}{\partial q}$ is the gradient direction and $\left[\frac{\partial V(q)}{\partial q}\right]_i^{\perp}$ is a direction that is perpendicular to the gradient. In ref. [11], One solution of Eq. (13) can be obtained by solving the optimization problem defined as follows:

$$\min_{\beta_i \in (-1,1)} \frac{1}{2} \sum_{i=1}^{n-1} \beta_i^2 \tag{15}$$

subject to $d_{g_j} Z(q) \geq 0$. The resulting control is the optimal $\beta = (\beta_1, \beta_2, \ldots, \beta_{n-1})$ such that optimized feasible direction is a direction which is in the feasible set and is closest to the gradient direction.

3.4. Optimization method using MNM

In this section, we propose a new optimization method based on the MNM. From Eq. (13), we solve the optimal solution of the following optimization problem for β, $\max_{\sum_{i=1}^{n-1} \beta_i^2 < 1} \{-Z(q)B(q)\frac{\partial V(q)}{\partial q}^T\}$ subject to $\frac{\partial g_j(q)}{\partial q} Z(q) > 0$, $j = 1, 2, \ldots, m$, where m is the number of active constraints for the robot.

In this way, we can find the optimal β_i so that the optimized feasible direction $Z(q)$ is the modified Newton's direction $(-B(q)\frac{\partial V}{\partial q})$ when the modified Newton's direction is a feasible direction and is closest to the modified Newton's direction otherwise.

We define the modified control law for the robots as:

$$\dot{q} = \alpha \frac{Z(q)}{\|Z(q)\|} \tag{16}$$

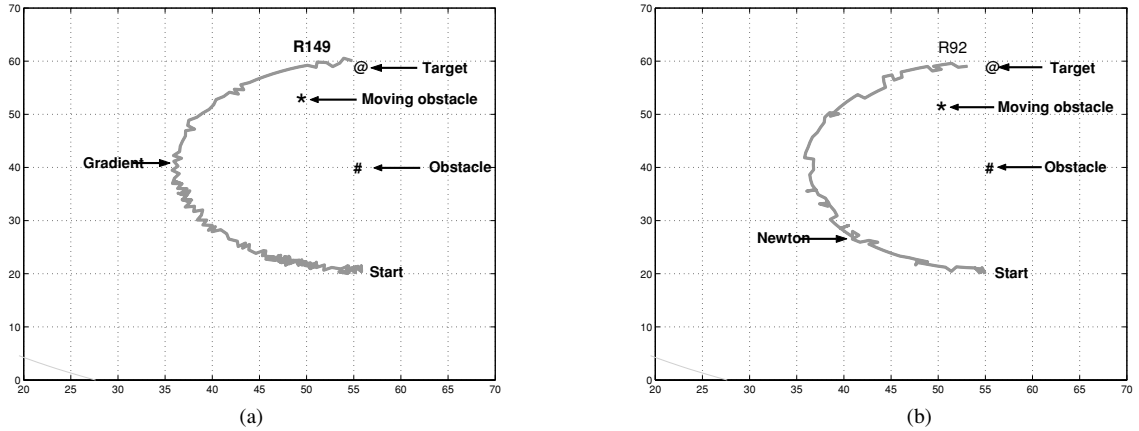
where α is the rate of the robot.

In plain language, if both the gradient direction and Newton direction are in the feasible set, Eq. (15) will move the robot along gradient direction and Eq. (16) will move the robot along Newton method. If neither gradient nor Newton directions are in the feasible set, Eq. (15) will move in a direction that is in the feasible set and closest to the gradient method. Eq. (16) will move the robot in a direction that is in the feasible set and closest to the Newton method.

In Fig. 4, we compare the two algorithms for moving obstacle avoidance in a simple scenario with one moving obstacle. The moving obstacle is activated at every step with random angle with respect to the robot. As we have seen in previous comparisons, the path generated by the modified Newton's method planner has reduced oscillation and the target is reached in fewer steps.

3.5. Performance comparison in passing a randomly generated obstacle area

In this section, we use randomly generated obstacles in an area to compare the performance of the two algorithms. The robot starts from (0, 0) and is required to reach the target, which is at (60, 69), by passing an obstacle-filled area which is 60 m by 70 m. The obstacle area is randomly packed with obstacles of different sizes and a minimum passage width between the obstacles is specified. In each



(a) (b)

Fig. 4. Performance comparison with moving obstacles. R149, R92 represent that robot moved 149 steps and 92 steps, respectively.

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press

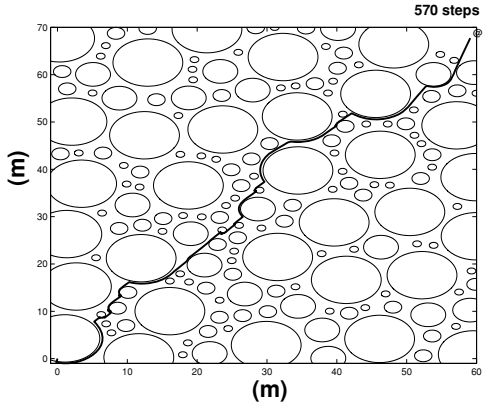


Fig. 5. A randomly generated setup.

run, more than 5000 obstacles of random locations are generated: they are added to the obstacle area if they satisfy the required minimum passage among obstacles and are discarded otherwise. In this way, our testbed (Fig. 5) represents a complex, unknown, random environment. We introduced moving obstacles into the playing field. It was assumed that the moving obstacle would move randomly, but be constrained to stay close (within the protective distance) of the robot at all times.

We performed 100 runs each for one, two and three moving obstacles and presented the results in Figs. 6–8, we compare two algorithms with two different inter-obstacle spacings and two different stepsize. In these figures, data

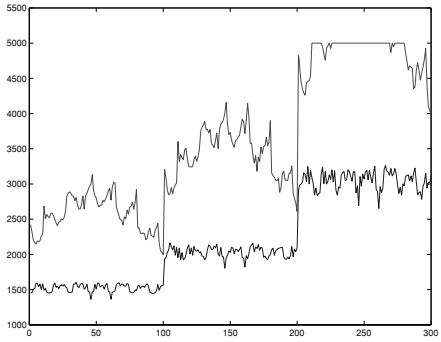


Fig. 6. Performance comparison with different number of moving obstacles. Raw data are presented, the first 100 runs are with one moving obstacle, the second 100 runs are with two moving obstacles, and the last 100 runs are with three moving obstacles.

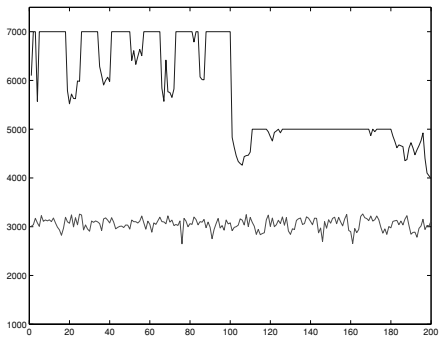


Fig. 7. Performance comparison with two different interrobot spacing. Raw data are presented. The first 100 runs are spacing = 0.63, the second 100 runs are spacing = 0.7.

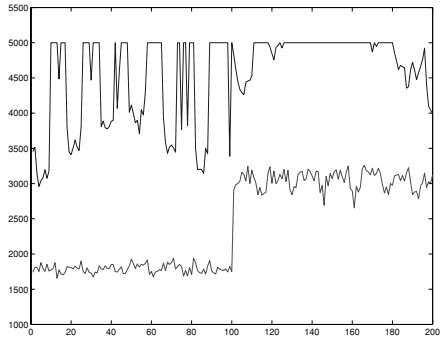


Fig. 8. Performance comparison with two different stepsize. Raw data are presented, the first 100 runs are stepsize = 0.2, the second 100 runs are stepsize = 0.1.

Table IV. Performance comparison for average number of steps.

Model		Gradient (steps)	Newton (steps)
Number of moving obstacles	1	2580	1542
	2	3428	2039
	3	4861	3063
Spacing (m)	0.63	6703	3080
	0.7	4861	3060
Stepsize (m)	0.2	4262	1806
	0.1	4847	3063

are sorted in ascending order of the task completion time for the gradient method (upper curve), with the corresponding task completion time for the MNM plotted at the same point on the x-axis (lower curve). We can see that the MNM-based controller significantly outperforms the gradient descent based controller for all scenarios.

We make the following general observations based on this data:

- MNM has a better performance in all cases. Progress is faster with less oscillation.
- As we saw in the case of the static environment, the advantages of the MNM become more obvious with faster speed (bigger stepsize) or narrower passages.

In summary, the use of MNM can improve the performance (Table IV) in almost all cases. We also notice that as the environment becomes more complex, the advantages of MNM become more pronounced.

4. MNM for Nonholonomic Robots in Dynamics Environment

In this section, we show that motion planning based on MNM can achieve better performance when applied to nonholonomic robots in dynamic environments. The methods presented in Section 3 can be applied to N-dimensional motion planning problems, however, it cannot be easily incorporated into nonholonomic robot motion planning. In this section, we adopt a geometric transformation method and construct the feasible direction set using the gradients of the navigation function and the activated constraints of moving obstacles, which is proposed in ref. [14] and presented in Appendix. Once the feasible direction set is obtained, the

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press

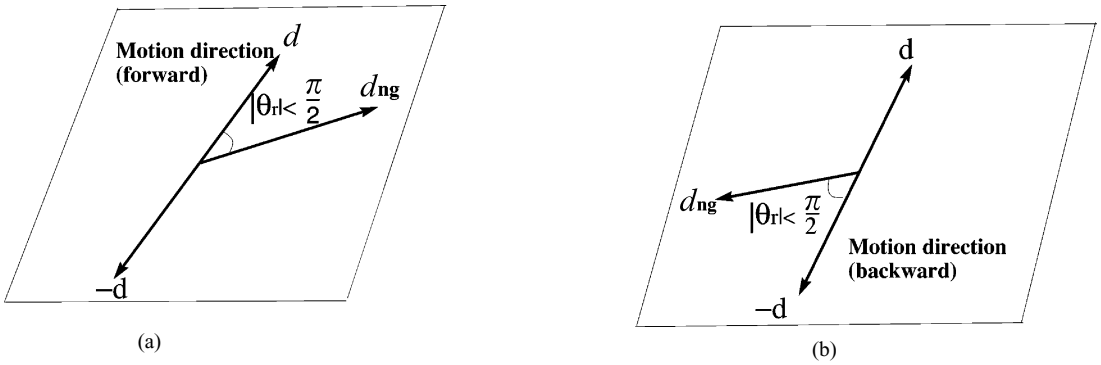


Fig. 9. (a) Configuration leading to forward motion and (b) Configuration leading to backward motion.

optimal motion direction can be found by projecting the gradient direction d_{ng} or MNM d_n to the closest feasible direction. In our technique, if the gradient direction or modified Newton direction is in the feasible space, we choose them as the optimal feasible direction, otherwise, the optimal feasible direction is defined as the projection of d_{ng} or d_n to the nearest edge of the feasible space with smallest rotation.

Once the optimal feasible direction has been chosen, we allow the wheeled robot to travel either forwards or backwards along its current heading based on whichever has a positive projection on the negative gradient direction (and therefore makes progress towards the goal). The heading is simultaneously updated through a linear control law to match the current heading to the optimal feasible direction. The optimal feasible direction d_{opt} can be expressed in a vector form $(d_{\text{opt}_x}, d_{\text{opt}_y})$ and the corresponding orientation angle θ_{opt} can be calculated as

$$\theta_{\text{opt}} = \arctan(d_{\text{opt}_x}, d_{\text{opt}_y}) \tag{17}$$

where $-\pi < \theta_{\text{opt}} \leq \pi$.

We denote the angle between the current heading direction d and the optimal feasible direction as $\theta_r \in [-\pi, \pi]$ and define it as follows:

$$\theta_r = \begin{cases} \theta_{\text{opt}} - \theta & \text{if } \|\theta_{\text{opt}} - \theta\| \leq \pi \\ \theta_{\text{opt}} - \theta - 2\pi & \text{if } \theta_{\text{opt}} - \theta > \pi \\ \theta_{\text{opt}} - \theta + 2\pi & \text{if } \theta_{\text{opt}} - \theta < -\pi \end{cases} \tag{18}$$

If $\|\theta_r\| \leq \frac{\pi}{2}$, the robot's heading has a positive projection on the optimal feasible direction, that is $d_{\text{opt}}^T d > 0$. In this case the robot will move forward, and we define the actual direction $d_m = d$ (Fig. 9(a)).

If $\|\theta_r\| > \frac{\pi}{2}$, the direction opposite to the robot's heading $-d$ has a positive projection on negative gradient direction d_{opt}, that is $d_{\text{opt}}^T(-d) > 0$. In this case, the robot will move backward, and we define the actual moving direction $d_m = -d$ (Fig. 9(b)).

The control law of linear velocity for the robot is then given by:

$$\dot{q} = k\frac{d_m}{|d_m|} \tag{19}$$

where k is a constant used to determine the robot's speed.

The control law for angular velocity is a simple linear control law intended to turn the robot as fast as possible towards the desired heading. The desired angular velocity is defined by:

$$\dot{\theta} = k_\theta \theta_r \tag{20}$$

where k_θ is a constant.

In Fig. 10, we use a task of target reaching with obstacles to illustrate the performance of Newton method and Gradient method using the geometric transform. One moving obstacles is activated at each step for this scenario. For the same task, the Newton method reaches the goal in 113 control intervals contrast to gradient method with 243 steps. The parameters we used for this comparison is listed in Table V.

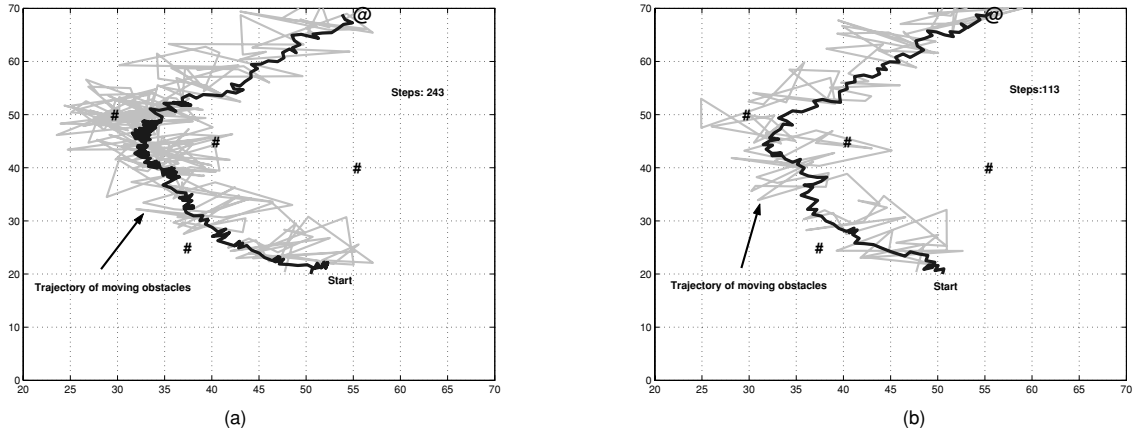


Fig. 10. Comparison of (a) feasible gradient method and (b) Newton method in dynamic environment.

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press

Table V. Parameters.

k	k_θ	Obstsize	θ
1	1	1	90°

Table VI. Parameters.

Obstsize	start θ	stepsize
0.5	90°	0.33

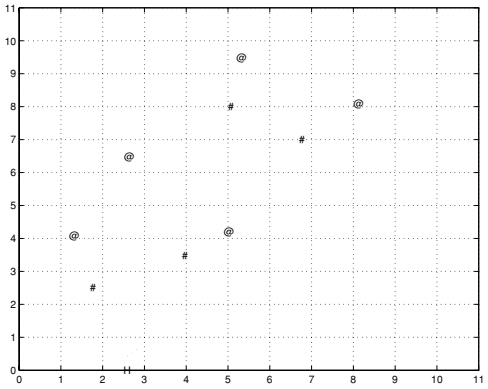


Fig. 11. Before the task starts.

5. Application Examples: Search-and-Forage

In this example, we consider the application of MNM in a search-and-forage task for a cooperating robot team.[15] In this task, a team of three robots is required to carry back the target objects (@) to the goal while avoiding collisions with each other and with the obstacles (#). Fig. 11 shows the workspace before the task start.

The potential field of the obstacles and target are generated by using generalized Gaussian function.[15] Notice that using the MNM, the amount of path oscillation in the presence of obstacles is greatly reduced, when compared to a standard gradient descent control. As shown in Fig. 12, the average steps is 150/task for the gradient method and is 115/task for MNM. The reason for this is the performance of the two methods are similar when is far from the obstacle. The advantage of MNM should be more evident when there are more obstacles in the fields. The parameters we used for this comparison is listed in Table VI.

6. Conclusions and Future Work

In this paper, we propose a general solution to the inherent oscillation problem for nonholonomic robot motion planning in dynamic environments using MNM. We studied the performance of the MNM in various scenarios. The results show that the proposed MNM can greatly improve system performance with very modest computation efforts.

The proposed method can either be used to directly generate robot's motion in some cases, or in other cases, it can be used to generate feasible path, the actually trajectory is then generated by exploiting other control method such as backstepping[7] which will control the robot to track the predefined path. In the future, we will investigate extensions of MNM to robot manipulators with high degrees of freedom.

Appendix

Proof of Theorem 1: The proof borrows from its counterpart in ref. 13. In order to prove this, we need to introduce Lemma 1.

Lemma 1:[16] Let M^1, M^2 be two open and connected subsets of R^n, such that $M^1 \bigcup M^2 = R^n \setminus \{0\}$. Let f^i: $M^i \longrightarrow R^n$, $i = 1, 2$ be two vector fields, and assume there exists a separating surface Γ with $0 \in \Gamma$ and $\Gamma \setminus \{0\} \subset M^1 \bigcap M^2$. Let C^1, C^2 be two connected subsets of $R^n \setminus \Gamma$ such that $C^i \subset M^i$, and assume that f^i on Γ is pointing toward the interior of C^i for $i = 1, 2$. Finally, assume that f^1, f^2 are asymptotically stable on M^1, M^2. Then, the vector field $f: R^n \longrightarrow R^n$ is defined as

$$f(x) = \begin{cases} f^1(x) & \text{if } x \in (\Gamma \setminus \{0\}) \bigcup C^1 \\ f^2(x) & \text{if } x \in C^2 \\ 0 & \text{if } x = 0 \end{cases} \tag{21}$$

is globally asymptotically stable.

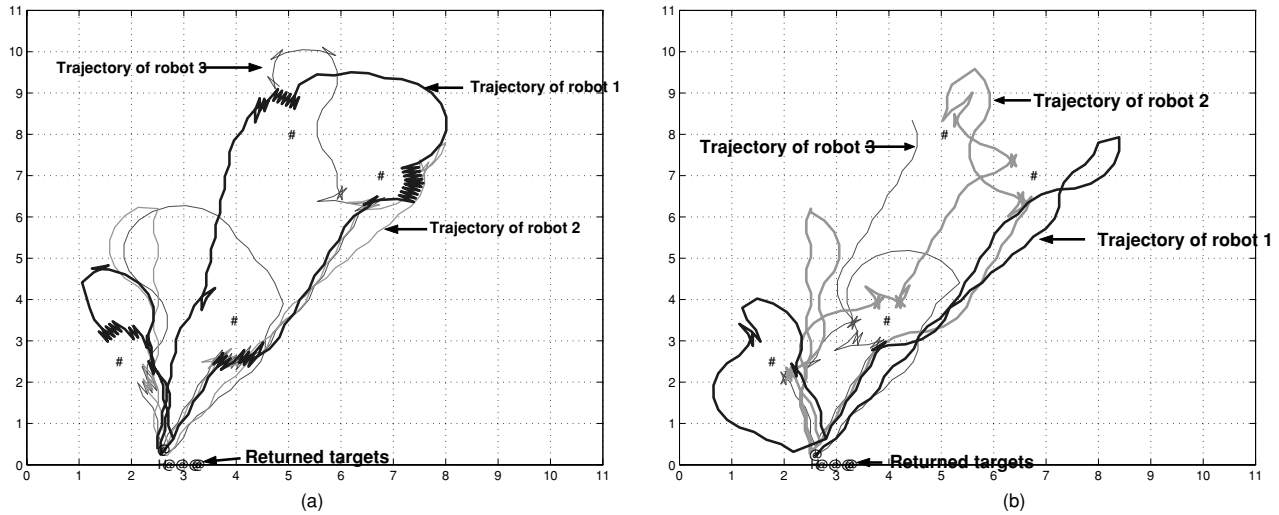


Fig. 12. Comparison of the search-and-forage task for (a) the gradient descent method and (b) the MNM.

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press

We define the inverse Lyapunov function as follows:

$$V(z) = \frac{\|x\|\beta^{1/k}}{\|z\|^2} \tag{22}$$

$$\|z\|^2 \triangleq x^2 + y^2 + \lambda\theta^2 \tag{23}$$

$$z \triangleq \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{24}$$

$$\begin{cases} V^1(z) = \frac{x\beta^{1/k}}{\|z\|^2} & x \geq 0 \\ V^2(z) = -\frac{x\beta^{1/k}}{\|z\|^2} & x < 0 \end{cases} \tag{25}$$

Consider the partition of the configuration space $Q = C^1 \bigoplus C^2 \bigoplus \Gamma$ induced by the navigation function $V(z) = V^j(z)$ for $z \in C^j$, $j = 1, 2$. Then the time derivative $\dot{V}^j(z)$ of $V(z)$ in C^j is

$$\dot{V}^j(z) = \frac{\partial V^j}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial V^j}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial V^j}{\partial \theta} \cdot \frac{d\theta}{dt} \tag{26}$$

$$= v\cos\theta \frac{\partial V^j}{\partial x} + v\sin\theta \frac{\partial V^j}{\partial y} + \omega \frac{\partial V^j}{\partial \theta} \tag{27}$$

$$= v[\cos\theta \sin\theta] \begin{bmatrix} \frac{\partial V^j}{\partial x} \\ \frac{\partial V^j}{\partial y} \end{bmatrix} + \omega \frac{\partial V^j}{\partial \theta} \tag{28}$$

$$= -v d_\theta^T d_{ngxy} - \omega d_{ng\theta} \tag{29}$$

$$\tag{30}$$

If $\varpi(z) \geq 0$, then $\omega = k_\theta(\theta_d - \theta)$ and

$$\dot{V}^j(z) = -\left[v d_\theta^T d_{ngxy} + k_\theta(\theta_d - \theta) d_{ng\theta}\right] \tag{31}$$

$$= \varpi(z) \geq 0 \tag{32}$$

If $\varpi(z) < 0$, then $\omega = -v\left[d_\theta^T d_{ngxy}\right] d_{ng\theta}^{-1}$ and

$$\dot{V}^j(z) = -v d_\theta^T d_{ngxy} + v d_\theta^T d_{ngxy} d_{ng\theta}^{-1} d_{ng\theta} \tag{33}$$

$$= 0 \tag{34}$$

Thus, $\dot{V}^j(z) \geq 0$. Now let $S \triangleq \{z \in C^j \bigcup \Gamma | \dot{V}^j(z) = 0\}$. If there exists an invariant set $\Omega \in S$, then in Ω, $v = \omega = 0$. From Eqs. (3) and (4) it follows that v vanishes only at the origin. By Lasalle's principle for $1/V^j(z)$ on $C^j \bigcup \Gamma$, asymptotic stability for the system is established on $M^j \triangleq C^j \bigcup \Gamma \setminus \{0\}$. By Lemma 1, the system is globally; asymptotically stable. QED.

Theorem 2: The feasible motion direction set can be constructed only using the gradients of the navigation function and the activated constraints of moving obstacles.

Proof: We use $dg_j = \frac{\partial g_j}{\partial q}$ to represent the gradient of the activated constraint g_j. We denote the angle corresponding to the gradient of constraint g_j to be θ_{v_j} and define it as follows:

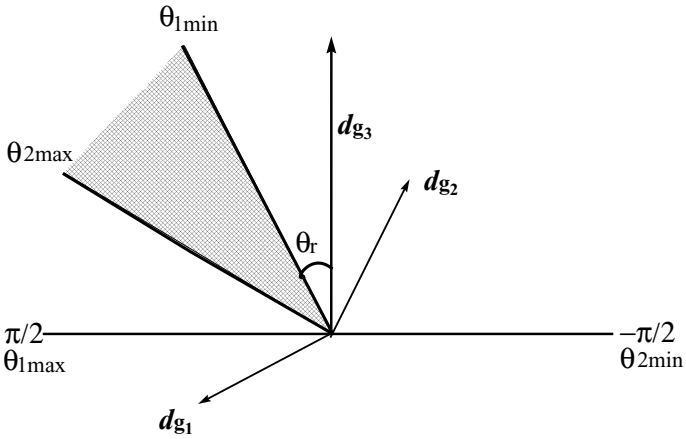


Fig. 13. Optimal feasible direction scheme. Filled area is the resulting feasible area.

$$\theta_{v_j} = \arctan\left(\frac{\partial g_j}{\partial y}, \frac{\partial g_j}{\partial x}\right) \tag{35}$$

where $-\pi < \theta_{v_j} \leq \pi$.

To better illustrate our method, we represent all directions in the form of rotation angles with respect to d_{ng}. Θ_j is defined as the angle between d_{ng} and dg_j. We define this angle to be positive if a counterclockwise rotation takes d_{ng} to dg_j and negative otherwise.

$$\Theta'_j = \theta_{vj} - \theta_{ng} \tag{36}$$

The resultant Θ'_j falls in the interval $[-2\pi, 2\pi]$ and we transform it to the interval $[-\pi, \pi]$ by the following formula:

$$\Theta_j = \begin{cases} \Theta'_j & \text{if } -\pi \leq \Theta'_j \leq \pi \\ \Theta'_j - 2\pi & \text{if } \Theta'_j > \pi \\ \Theta'_j + 2\pi & \text{if } \Theta'_j < -\pi \end{cases} \tag{37}$$

Next we construct a feasible direction set $[\Theta_{j\min}, \Theta_{j\max}]$ for each of the activated moving obstacles (Fig. 13).

$$\Theta_{j\min} = \begin{cases} \Theta_j - \pi/2 & \text{if } \Theta_j > 0 \\ -\pi/2 & \text{if } \Theta_j < 0 \end{cases} \tag{38}$$

$$\Theta_{j\max} = \begin{cases} \Theta_j + \pi/2 & \text{if } \Theta_j < 0 \\ \pi/2 & \text{if } \Theta_j > 0 \end{cases} \tag{39}$$

Note that the feasible direction set $[\Theta_{j\min}, \Theta_{j\max}]$ for each activated moving obstacle constraint is a subset of $[-\pi/2, \pi/2]$ because all feasible directions must lie in the same half-tangent plane as d_{ng}.

Next, we find the intersection $\theta_f = [\Theta_{min}, \Theta_{max}]$ of the feasible direction sets $[\Theta_{j\min}, \Theta_{j\max}]$ that works for all activated moving obstacles.

$$\Theta_{\min} = \max\{\Theta_{1\min}, \Theta_{2\min}, \ldots, \Theta_{m\min}\} \tag{40}$$

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press

$$\Theta_{\max} = \min\{\Theta_{1\max}, \Theta_{2\max}, \ldots, \Theta_{m\max}\} \quad (41)$$

By construction, we know that θ_f is also a subset of $[-\pi/2, \pi/2]$. QED.

Acknowledgment

This work was supported by the National Science and Engineering Research Council (NSERC) of Canada under grants RGPIN-249883-02, RGPIN-1345, and RGPIN 303802.

References

1. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(1), 90–98 (1986).
2. S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Robot. Automat.* **16**(5), 609–615 (2000).
3. J. C Latombe, *Robot Motion Planning* (Kluwer Academic Publishers, Boston, Massachusetts,1991).
4. Y. Koren, *et al*., "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," *IEEE International Conference on Robotics and Automation*, Sacramento (April, 1991) pp. 1398–1404.
5. J. Ren, K. A. McIsaac and R. V. Patel, "Modified Newton's method applied to potential field based navigation for omnidirectional mobile robots," *IEEE Trans. Robot.* **22**(2), 384–391 (2006).
6. M. Bazaraa, H. Sherali, and C. Shetty. *Nonlinear Programming Theory and Algorithms* (John Wiley and Sons, Inc., New York, 1995).
7. T. Fukao, H. Nakagawa and N Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Trans. Robot. Automat.* **16**(5), 609–615 (2000).
8. H. S. Shim and Y. G. Sung, "Stability and four-posture control for nonholonomic mobile robots," *IEEE Trans. Robot. Automat.* **20**(1), 148–154 (2004).
9. "Hilare 2 Robot Specifications," Available at: http://www.laas.fr/matthieu/robots/h2+.
10. H. G. Tanner and K. J. Kyriakopoulos, "Nonholonomic Motion Planning for Mobile Manipulators," *IEEE International. Conference on Robotics and Automation*, San Franscisco (April, 2000) pp. 1233–1238.
11. J. Esposito and V. Kumar, "A Method for Modifying Closed-Loop Motion Plans to Satisfy Unpredictable Dynamic Constraints at Run-Time," *IEEE International Conference on Robotics and Automation*, Washington, DC, (May, 2002), pp. 1691–1696.
12. H. G. Tanner, S. Loizon and K. J. Kyriakopoulos, "Nonholonomic Stabilization with Collision Avoidance for Mobile Robots," *IEEE International Conference on Intelligent Robots and Systems*, Hawaii (October/November, 2001) pp. 1220–1225.
13. H. G. Tanner, S. G. Loizou and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot. Automat.* **19**(1), 53–64 (2003).
14. J. Ren, K. A. McIsaac and R. V. Patel, "A Fast Algorithm for Moving Obstacle Avoidance for Mobile Robots," *Proceedings of the IEEE Conference on Control Applications*, Canada (August, 2005) pp. 209–214.
15. J. Ren and K. A. McIsaac, "A Hybrid-Systems Approach to Potential Field Navigation for a Multi-Robot Team," *IEEE International Conference on Robotics and Automation*, Taiwan (September, 2003) pp. 3875–3880.
16. G. A. Lafferriere and E. D. Sontag, "Remarks on Control Lyapunov Functions for Discontinuously Stabilizing Feedback," *IEEE International Conference Decision and Control*, San Antonio, TX (December, 1993) pp. 2398–2403.

https://doi.org/10.1017/S026357470700389X Published online by Cambridge University Press