Robotica (2008) volume 26, pp. 117–127. © 2007 Cambridge University Press
doi:10.1017/S0263574707003694 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, UOIT, Oshawa, ON, Canada 1LH 7K4.*
‡*Department of Electrical and Computer Engineering, University of Western Ontario, London, ON, Canada N6G 1H1*

(Received in Final Form: June 19, 2007. First published online: August 13, 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 multi-robot 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 the robot's movement, when compared to the standard gradient approaches.

1. INTRODUCTION

The potential field method is widely used in mobile robot navigation for its simplicity and elegance.[4,5,7] In the potential field method, 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.*[6] identified the problem of oscillations in potential fields that can result when the discrete system becomes unstable, 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,[11] we investigated the sources of oscillations and showed that even when the discrete system is stable, oscillations can result from the characteristics of the gradient descent method, and in fact roughly correspond to "oscillatory" behaviour of the gradient direction. It is well known in optimization theory[1] that the gradient method exhibits oscillation and slow convergence in the neighbourhood of the optimal point. This phenomenon results from the fact that the objective function, which is often a complex nonlinear function, at least twice differentiable, can be reasonably well approximated by a quadratic function in the neighbourhood of an optimum. The gradient descent method applied to such a function in the neighbourhood of an optimum generates descent directions at successive iterations that are almost orthogonal, resulting in a "zig-zag", or oscillatory behaviour of the optimization. The nature of the oscillations depends on the characteristics of the objective function—a function that has small curvature in one or more degrees of freedom will tend to exhibit greater oscillations and slow convergence to an optimum. Such a situation can arise in potential field navigation when the potential field is severely deformed due to a complex configuration of targets and obstacles, especially when the robots enter narrow passages. In such cases, the use of the standard gradient descent method will result in poor performance.

In our prior work, we proposed a modification of Newton's method to overcome the oscillation problem for omnidirectional robots in planar and static environments.[11] The modified Newton's method (MNM) is based on approximating the navigation function by a quadratic function around a point and contains both first and second derivative information. Therefore, the approximation of the function in the neighbourhood of an optimum is much better than for the gradient method, thereby reducing or even eliminating oscillations, achieving faster task completion and allowing a bigger step size (faster convergence).

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"[3,2] refers in general to non-integrable 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 non-slip 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 non-slip constraint is the Hilare mobile robot created at Laboratorie d'Automatique et d'Analyse des Systemes (LAAS), located in Toulouse, France.[16] 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

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

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

cannot be used with a wheeled robot because the gradient descent direction, in general, will not be parallel to the robot's heading direction and will, therefore, be forbidden by the constraint. Tanner and Kyriakopoulos[13] 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 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 unmodelled obstacles. As was the case for the wheeled robot subject to the non-slip constraint, the presence of an obstacle that is not modelled in the potential field will forbid the mobile robot to choose the gradient descent direction. Esposito and Kumar[2] 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 MNM for moving obstacles.

This paper is organized as follows. In Section 2, we breifly review the definition of MNM-based control laws for potential fields. In Section 3, we review Tanner's approach for wheeled robots and present an extension of MNM for nonholonomic robots. In Section 4, we review Esposito's approach for unmodelled obstacles and show how we can extend this approach using the MNM. We also investigate the performance of MNM with changes in different parameters. In Section 5, 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 non-slip constraint and the presence of unmodelled obstacles. In Section 6, we compare the performance of MNM and the gradient method in a search-and-rescue task. Finally, in Section 7, we conclude the paper and suggest some future research directions.

2. REVIEW OF THE MODIFIED NEWTON'S METHOD

A thorough introduction to the use of modified Newton's method (MNM) in potential fields is beyond the scope of this work; however, we include an overview of the technique for completeness. Interested readers are referred to our prior work.[11]

The traditional approach in potential field navigation is to define a scalar valued *navigation function* $V(q)$, with q being the configuration of the robot. Motion control of the robot is achieved by chosing the negative gradient direction as the direction for robot motion to yield:

$$\dot{q} = -\alpha \frac{\partial V(q)}{\partial q}. \tag{1}$$

Provided $V(q)$ is properly defined as a Lyapunov function without local minima in the neigbourhood, it is straightforward to show that this is a stable control that will drive the robot to a desired endpoint. However, as we discussed in the introduction, the solution tends towards oscillation and "zig-zag" trajectories in cluttered environments and narrow passages.

In our work, we replace the standard gradient descent controller with a new control law based on Newton's method, as

$$\dot{q} = -B(q) \frac{\partial V(q)}{\partial q}. \tag{2}$$

In Eq. (2), the matrix $B(q)$ is based on MNM. As in optimization theory,[1] $B(q)$ is a positive-definite matrix and is defined as $B(q) = (\varepsilon I + H)^{-1}$, for $\varepsilon = \varepsilon(q) \geq 0$ and H the Hessian matrix of $V(q)$. For low-dimensional problems, such as those considered in this paper, the computational cost incurred by the requirement to find H^{-1} (which often deters the use of Newton's method in optimization) is not significant.

3. APPLYING THE MODIFIED NEWTON'S METHOD TO ROBOTS WITH NON-SLIP 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. For convenience, we will also use a configuration vector $z = (q, \theta)$. 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.*[13,15,14] We will develop a corresponding MNM control law and compare the performance of the two control laws. Dipolar functions behave like the reciprocal of conventional navigation functions. They are designed to go to infinity at the equilibrium point and decay to zero at the obstacles. Stable controls are defined so that, for an inverse navigation function $V(z)$, $\dot{V}(z)$ is positive definite (i.e. control proceeds along gradient *ascent* rather than descent directions).

Before introducing the modification to the MNM, we review the development from.[14] The navigation function is defined as:

$$V(z) = \frac{\|x\| \beta^{\frac{1}{k}}}{\|z\|^2} \tag{3}$$

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

Next, a scaled version of the gradient called $f = \{f_x, f_y, f_\theta\}$ is defined, the elements of which are:

$$f_r = \|z\|^4 \frac{\partial V}{\partial r}, \qquad r \in x, y, \theta \tag{4}$$

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

and the robot's dynamics are given by:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta\,. \\ \dot{\theta} = \omega \end{cases} \tag{5}$$

The control law generates the linear velocity as:

$$v = -k_v sgn(f_x\cos\theta + f_y\sin\theta)\left(f_x^2 + f_y^2 + f_\theta^2\right) \tag{6}$$

and the angular velocity as:

$$\omega = \begin{cases} k_\theta(\theta_d - \theta), & w(z) \geq 0 \\ v(f_x\cos\theta + f_y\sin\theta)f_\theta^{-1}, & w(z) < 0 \end{cases} \tag{7}$$

where k_θ and k_v are constants, and:

$$w(z) = v(f_x\cos\theta + f_y\sin\theta) + k_\theta f_\theta(\theta_d - \theta) \tag{8}$$

and the desired orientation θ_d is calculated by:

$$\theta_d = \arctan2(-\mathrm{sgn}(x)f_y, -\mathrm{sgn}(x)f_x) \tag{9}$$

that represents the desired direction of gradient ascent. The function $\mathrm{sgn}(x)$ is the sign function:

$$sgn(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \tag{10}$$

The effect of this controller is to drive the robot either forward or backwards, depending on which of these two directions has a positive projection on the gradient direction. At the same time, it generates changes in orientation to align the robot's orientation with the gradient direction.

This control law can be proven to be asymptotically stable (interested readers should refer to the original work for a full treatment[14]). The switching function for ω guarantees stability by assuring $\dot{V}(z) \geq 0$. The non-existence of an invariant set (and hence, convergence) is guaranteed by the fact that v does not vanish except at the origin, and because the definition of ω forbids the robot to settle on a trajectory that is orthogonal to the gradient.

3.1. Improving the performance with modified Newton's methods

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

$$d_n = B(f_x, f_y)^T = (d_{nx}, d_{ny})^T \tag{11}$$

where, in this case, B is defined as $B = (\epsilon I - H)^{-1}$, with $\epsilon \geq 0$ again chosen to ensure B is positive definite. This definition for B differs slightly from the standard technique introduced in Section 2 because the dipolar navigation function corresponds to a maximization, rather than a minimization problem. The rest of the control law is the same as in Section 2. In this case, the desired orientation θ_d in Eq. (9) is calculated based on the modified Newton direction d_n instead of the negative gradient direction defined by $(f_x, f_y)^T$

$$\theta_d = \arctan2(-sgn(x)d_{ny}, -sgn(x)d_{nx}). \tag{12}$$

The proof of convergence from ref. [14] applies to the MNM version of the control law with only one minor modification. The non-existence of an invariant set in this case is guaranteed by the fact that the Newton direction cannot be orthogonal to the gradient direction since B is positive definite by design.

In Fig. 1, we show the performance of the 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 of "step size" from optimization theory[1] 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 "@" symbol and the obstacle is represented by "#" symbol. This figure illustrates that

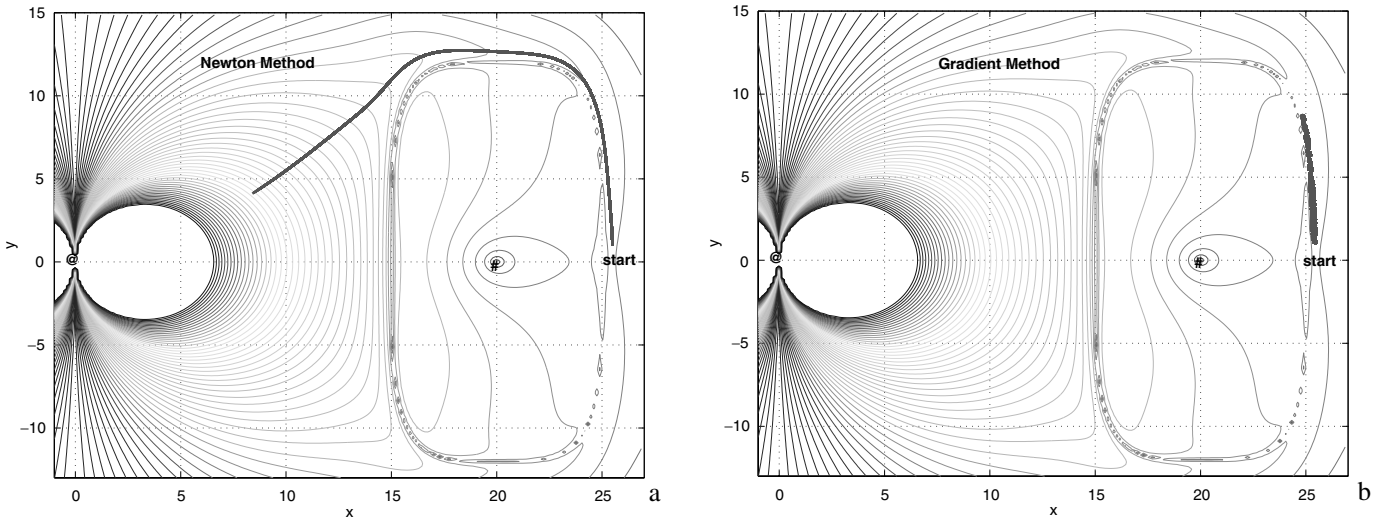


Fig. 1. Performance in a typical obstacle avoidance task. (a) MNM. (b) Gradient method.

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

Table I. Parameters used in a typical obstacle avoidance task.

k_v	k_θ	startX	startY	startθ
1	26	26	0.6	90^o

Table III. Parameters.

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

the MNM can achieve much faster progress in presence of obstacles. MNM almost reached the goal after 150 steps, while the gradient method has hardly made any progress. The parameters that we used for this comparison are listed in Table I.

It is well known[7] 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 the 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 the gradient method deteriorates substantially. The parameters that we used for this comparison are listed in Table II.

Because it is more reasonable for mobile robots to have limits on linear velocity, we will use Tanner's method but put a upper limit to the velocity and step size in the following discussions. Note that the benefit of using MNM is independent of the choice of fixed or changing or saturated linear velocity.[11]

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) and (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 as shown in Fig. 3(c). We have magnified Figs. 3(c) and (d), and shown parts of the enlarged versions in Figs. 3(e) and (f). The parameters that we used for this comparison are listed in Table III.

Table II. Parameters to compare performance starting from a rough alignment.

k_v	k_θ	startX	startY	startθ
1	26	26	0.01	90^o

4. MODIFIED NEWTON'S METHOD FOR ROBOTS IN A DYNAMIC ENVIRONMENT

In this section, we first recall an optimization-based gradient method, which treats moving obstacles as dynamic constraints.[2] 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 modified Newton's method can also greatly improve the performance in dynamic environments.

4.1. Model of moving obstacles

In the approach defined in ref. [2], 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:

$$g_j(q) = |q - P_j|^2 - D_{\text{obs}}^2 \geq 0 \tag{13}$$

where P_j is the position of the moving obstacle j. These constraints will be used in our control law to generate feasible

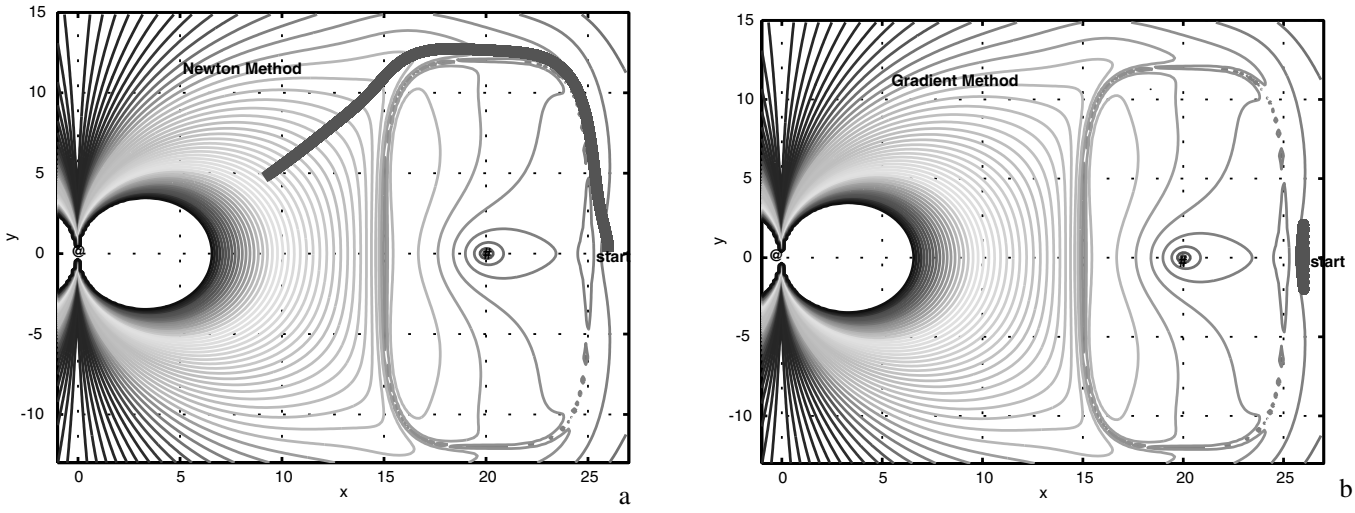


Fig. 2. Performance with a rough alignment starting point. (a) MNM. (b) Gradient method.

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

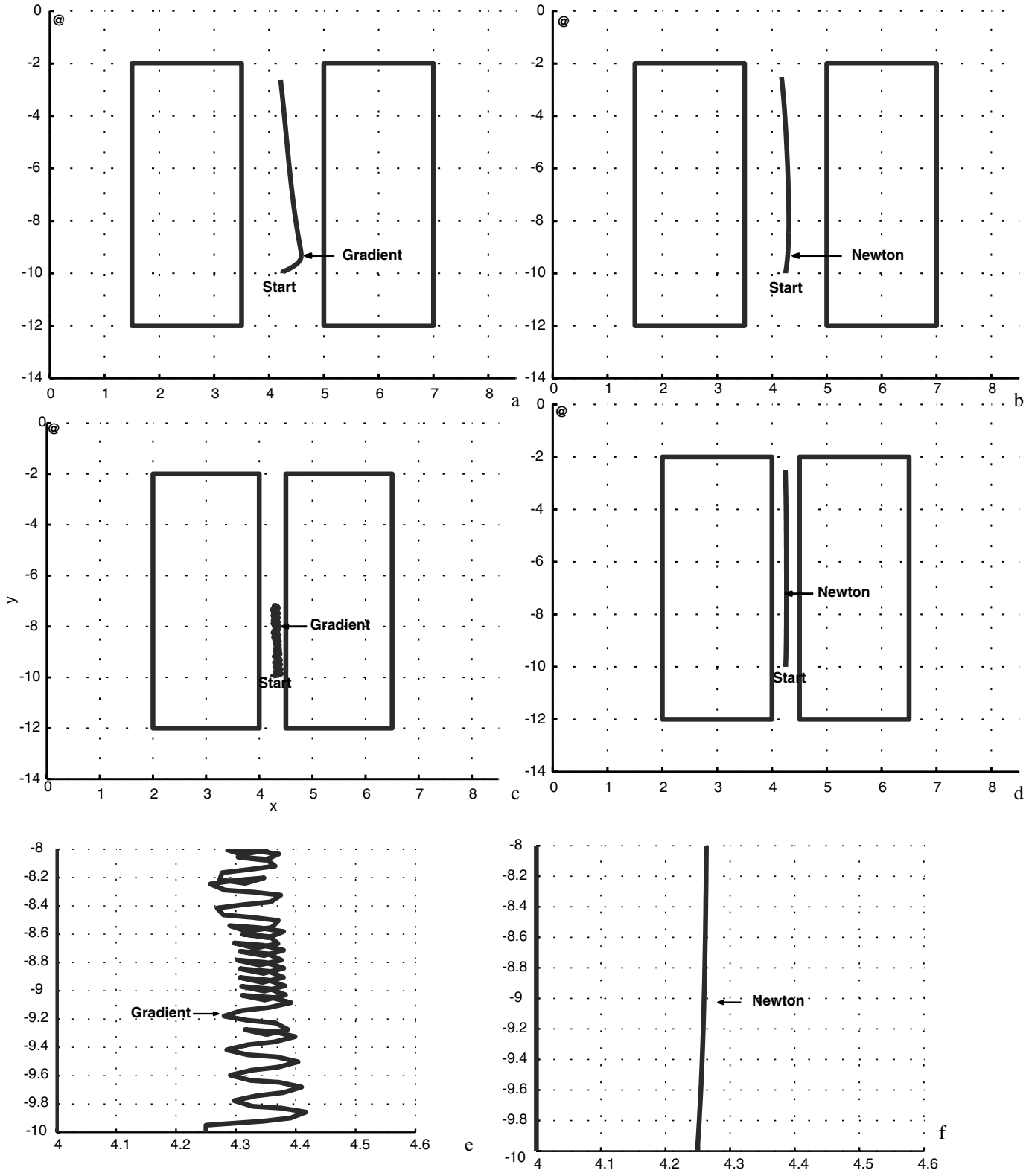


Fig. 3. (a) and (c) Performance of the gradient method in wide and narrow passages, respectively. (b) and (d) Corresponding trajectories for MNM in the same scenarios. (e) and (f) Parts of the enlarged version of (c) and (d). Totally 150 steps.

directions of motion, which move the robot away from the obstacle while still making progress towards the goal.

4.2. Definition of feasible motion direction

The basic idea of motion planning in ref. [2] is to choose a best direction from all feasible directions. In the context of moving obstacle avoidance, *feasible motion directions* are the directions that can stabilize the system and also move robots away from the activated constraints caused by moving obstacles.

We use $d_{ng} = -\partial V/\partial q$ to represent the negative gradient direction, and $d_{gj} = \partial g_j/\partial q$ to represent the gradient of the

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

activated constraint g_j. A feasible motion direction d_f is defined as the direction that satisfies the following conditions:

$$d_f^T d_{ng} > 0 \tag{14}$$

$$d_f^T d_{g_j} > 0, \qquad j = 1, 2, \ldots, m. \tag{15}$$

In Eq. (14), we require a control that can stabilize the system to have a positive projection on d_{ng}. In Eq. (15), we require a control that can move the robot away from the activated obstacle to have a positive projection on d_{g_j}.[2] Feasible directions must satisfy both conditions at once.

4.3. Optimization method using gradient method

If we denote the navigation function as $V(q)$, the control law $Z(q)$ defined by Esposito[2] is as follows:

$$Z(q) = -\left(1 - \sum_{i=1}^{n-1} \beta_i^2\right)^{\frac{1}{2}} \frac{\partial V(q)}{\partial q} + \sum_{i=1}^{n-1} \beta_i \left[\frac{\partial V(q)}{\partial q}\right]_i^{\perp} \tag{16}$$

subject to

$$\sum_{i=1}^{n-1} \beta_i^2 < 1 \tag{17}$$

where $\partial V(q)/\partial q$ is the gradient direction and $[\partial V(q)/\partial q]_i^{\perp}$ is a direction that is perpendicular to the gradient. In ref. [2], one solution of Eq. (16) can be obtained by solving the optimization problem defined as:

$$\min_{\beta_i \in (-1,1)} \frac{1}{2} \sum_{i=1}^{n-1} \beta_i^2 \tag{18}$$

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

4.4. Optimization method using modified Newton method

In this section, we propose a new optimization method based on the MNM. From Eq. (16), we solve the optimal solution of the following optimization problem for β:

$$\max_{\sum_{i=1}^{n-1} \beta_i^2 < 1} \left\{ -Z(q) B(q) \frac{\partial V(q)}{\partial q}^T \right\} \tag{19}$$

subject to

$$\frac{\partial g_j(q)}{\partial q} Z(q) > 0, \qquad j = 1, 2, \ldots, m \tag{20}$$

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 $(-B(q)\partial V/\partial q)$ 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:

$$\dot{q} = \alpha \frac{Z(q)}{\|Z(q)\|} \tag{21}$$

where α is the rate of the robot.

If both the gradient direction and Newton direction are in the feasible set, Eq. (18) will move the robot along the gradient direction and Eq. (21) will move the robot along the Newton direction. If neither the gradient nor Newton directions are in the feasible set, Eq. (18) will move in a direction that is in the feasible set and closest to the gradient direction. Equation (21) will move the robot in a direction that is in the feasible set and closest to the Newton direction.

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 w.r.t. the robot. As we have seen in previous comparisons, the path generated by the modified Newton's

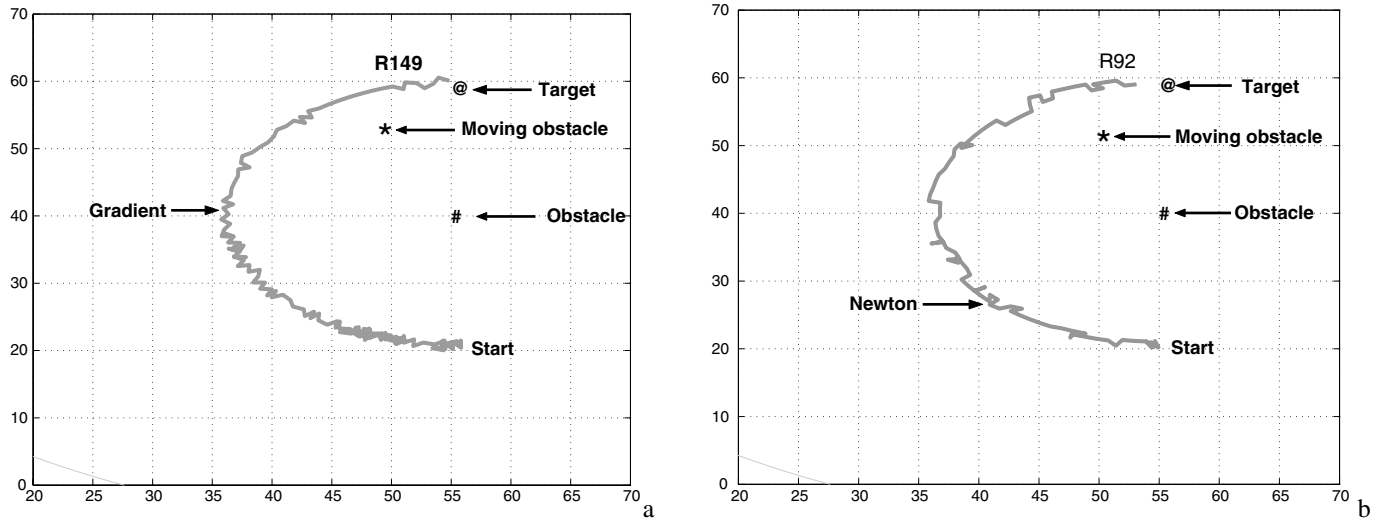


Fig. 4. Performance comparison with moving obstacles. R149 and R92 represent that the Robot moved 149 steps and 92 steps, respectively.

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

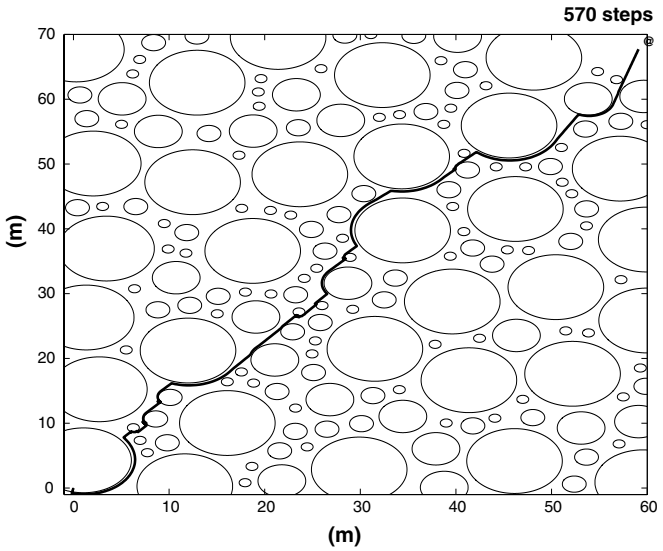


Fig. 5. A randomly generated setup.

method planner has reduced oscillation and the target is reached in fewer steps.

4.5. 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 × 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 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 the 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) to 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 and Table IV. We compare two algorithms with two different inter-obstacle spacings and two different step sizes. In these figures, the data 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.

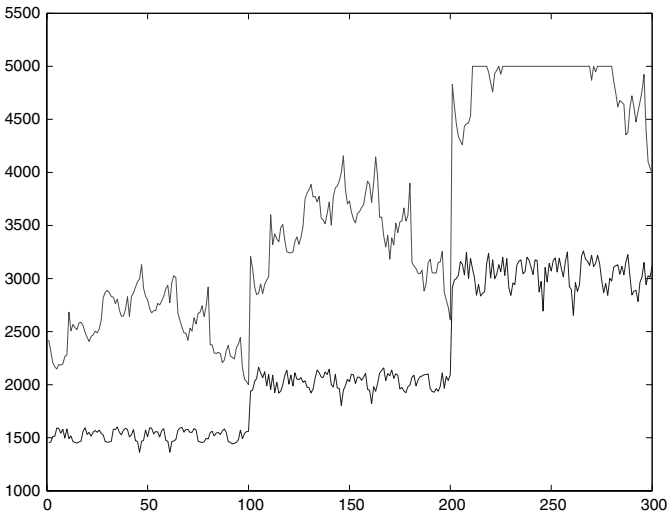


Fig. 6. Performance comparison with different number of moving obstacles. Raw data are presented, the first 100 runs are with one moving obstacle, the second 100 runs are with two moving obstacles, and the last 100 runs are with three moving obstacles.

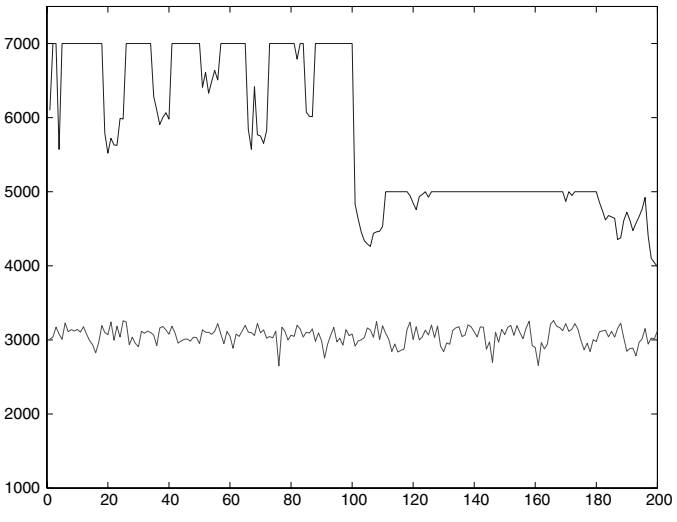


Fig. 7. Performance comparison with two different inter-robot spacings. Raw data are presented. The first 100 runs are spacing = 0.63, the second 100 runs are spacing= 0.7.

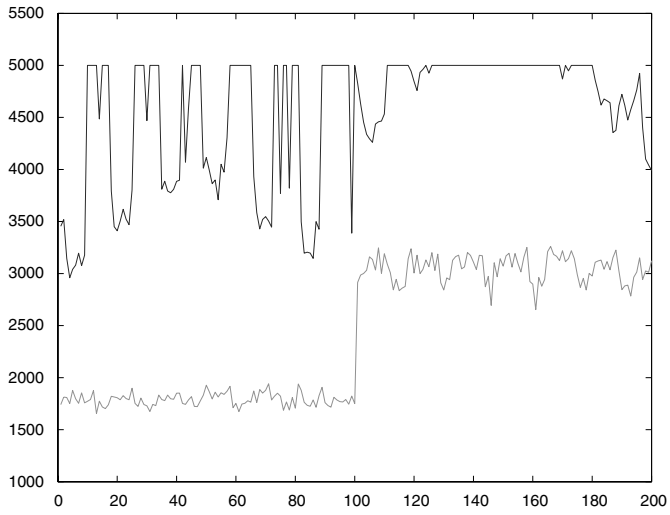


Fig. 8. Performance comparison with two different step size. Raw data are presented, the first 100 runs are step size = 0.2, the second 100 runs are step size = 0.1.

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

We make the following general observations based on this data:

- Modified Newton's method 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 step size) or narrower passages.

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

5. MODIFIED NEWTON'S METHOD FOR NONHOLONOMIC ROBOTS IN DYNAMIC ENVIRONMENTS

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 4 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. We first proposed this technique in prior work,[10] and we include it here in the Appendix for completeness. Once the feasible direction set is obtained, the optimal motion direction can be found by projecting the gradient direction d_{ng} or modified Newton's direction 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:

$$\theta_{\text{opt}} = \arctan(d_{\text{opt}_x}, d_{\text{opt}_y}) \tag{22}$$

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:

$$\theta_r = \begin{cases} \theta_{\text{opt}} - \theta & \text{if } \|\theta_{\text{opt}} - \theta\| \leq \pi \\ \theta_{\text{opt}} - \theta - 2\pi & \text{if } \theta_{\text{opt}} - \theta > \pi \\ \theta_{\text{opt}} - \theta + 2\pi & \text{if } \theta_{\text{opt}} - \theta < -\pi \end{cases}. \tag{23}$$

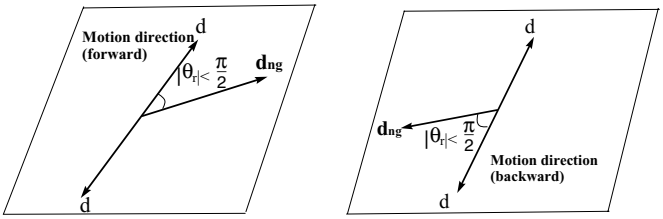


Fig. 9. (a) Configuration leading to forward motion. (b) Configuration leading to backward motion

Table IV. Performance comparison for average number of steps.

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

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

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

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

$$\dot{q} = k\frac{d_m}{|d_m|} \tag{24}$$

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:

$$\dot{\theta} = k_\theta \theta_r \tag{25}$$

where k_θ is a constant.

In Fig. 10, we consider a task of target reaching with obstacles to illustrate the performance of the MNM and gradient method using the geometric transform. One moving obstacle is activated at each step for this scenario. For the same task, Newton's method reaches the goal in 113 control intervals whereas the gradient method does so with 243 steps. The parameters that we used for this comparison are listed in Table V.

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

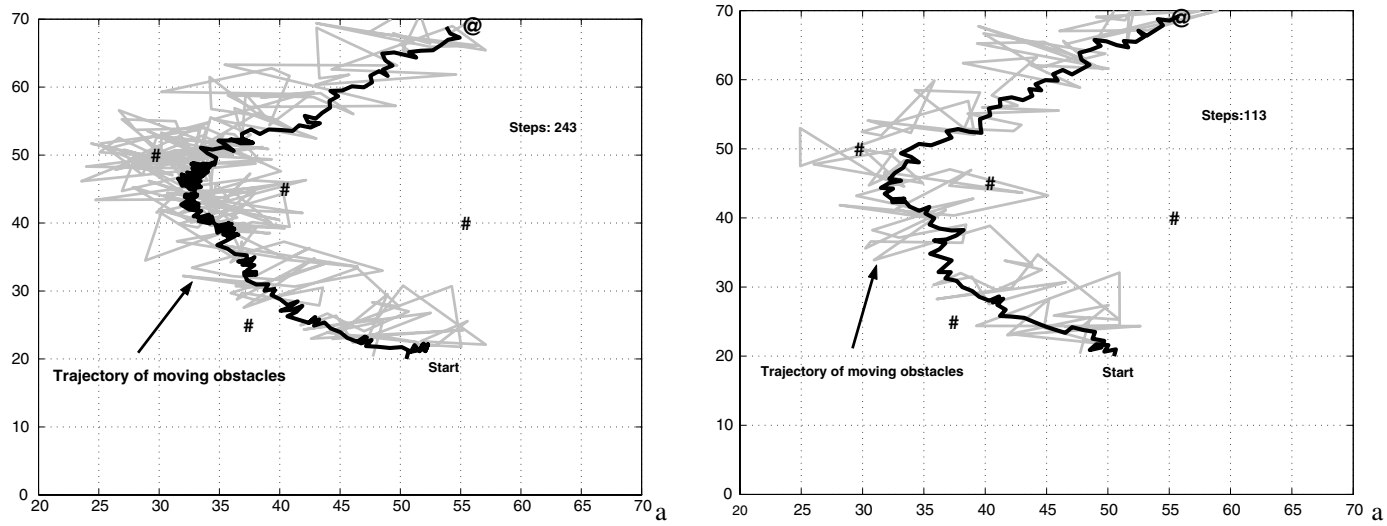


Fig. 10. Comparison of feasible direction (a) gradient method and (b) Newton method in dynamic environment.

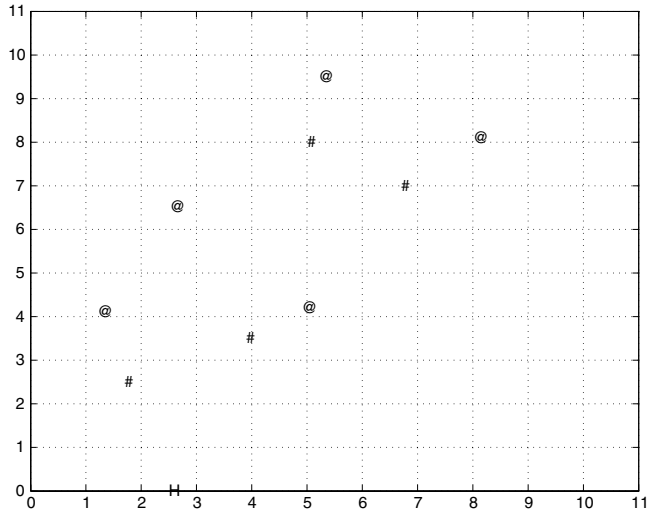


Fig. 11. Workspace Before the task starts.

Table V. Parameters.

k	k_θ	ObstSize	θ
1	1	1	90°

6. 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.[9] 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 (#). Figure 11 shows the workspace before the task starts.

The potential field of the obstacles and target are generated by using generalized Gaussian function [9] 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

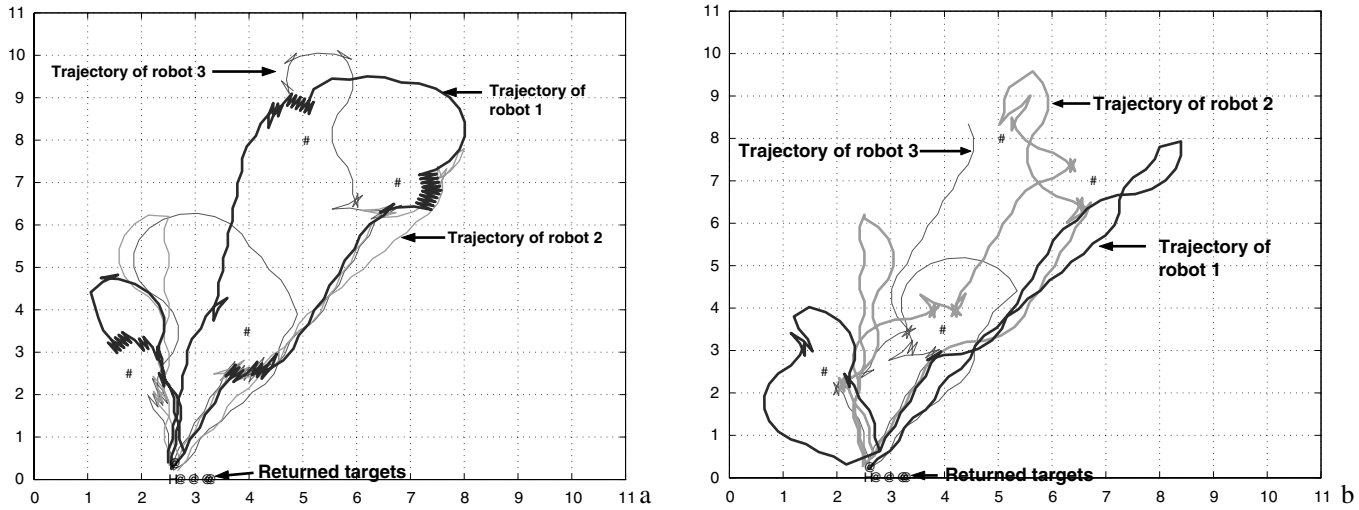


Fig. 12. Comparison of the search-and-forage task for (a) the gradient descent method and (b) the modified Newton's method.

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

Table VI. Parameters.

ObstSize	startθ	Step Size
0.5	90°	0.33

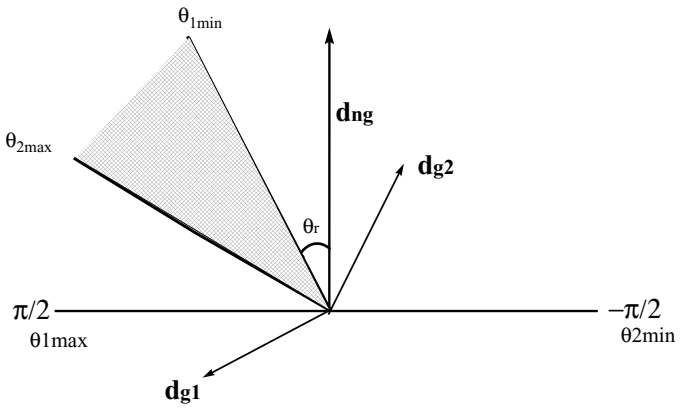


Fig. 13. Optimal feasible direction scheme. The filled area is the resulting feasible area.

steps 150/task for the gradient method and is 115/task for MNM. Although the MNM results are superior, the results for the two competing methods are close together in this case. The explanation is that the Newton direction and the gradient direction tend to be very similar when far away from obstacles. The benefit of the MNM is most noticeable in close proximity to obstacles and would be more pronounced in a field that were more cluttered. The parameters that we used for this comparison are listed are Table VI.

7. CONCLUSIONS AND FUTURE WORK

The trajectory generated by standard gradient-based motion planners can lead to severe oscillations in cases when the navigation function is severely deformed, as tends to happen in proximity to obstacles and in narrow passages. The MNM can generate a smoother path with fewer oscillations and zig-zagging, without losing the stability and convergence properties common to artificial potential-field-based methods.

In this paper, we propose extensions to the use of the MNM. First, we proposed an MNM-based motion planner for a class of wheeled robots subject to non-slip constraints based on the dipolar inverse navigation functions first proposed in.[14] By attempting to steer the robot to the Newton direction, the overall path followed by the robot can be smoothed even in the presence of the nonholonomic constraint.

Next, we proposed an optimization-based method for avoiding unmodelled, possibly moving, obstacles in a dynamic environment. The motion planner is based on prior work from the literature,[2] but once again the use of the MNM leads to smoother trajectories with fewer oscillations. Finally, we proposed a geometric method based on planar rotations to permit a wheeled robot to select the optimal trajectory for navigation in a potential field in the presence of unmodelled obstacles.

In general, the MNM can be used to directly generate robot's motion in some cases, generally for low-dimensional problems. In other cases, where the computational cost becomes burdensome, it could be used by an off-line motion planner to generate a feasible path for the robot. The actual trajectory would then generated by exploiting other control techniques such as backstepping[3] which will control the robot to track the predefined path. In future, we will investigate the extensions of MNM to robot manipulators with several degrees of freedom.

Acknowledgement

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. M. Bazaraa, H. Sherali and C. Shetty, *Nonlinear Programming Theory and Algorithms* (Wiley, New York, 1995).
2. J. Esposito and V. Kumar, "A Method for Modifying Closed-Loop Motion Plans to Satisfy Unpredictable Dynamic Constraints at Run-Time," *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC (2002) pp. 1691–1696.
3. T. Fukao, H. Nakagawa and N Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *IEEE Trans. Robot. Autom.* **16**(5), 609–615 (2000).
4. S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Robot. Autom.* **16**(5), 609–615 (2000).
5. O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.* **5**(1), 90–96, 1986.
6. Y. Koren *et al.* "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," *Proceeding of the IEEE International Conference on Robotics and Automation* (1991) pp. 1398–1404.
7. J. C Latombe, *Robot Motion Planning* (Kluwer, Boston, MA, 1991).
8. G. A. Lafferriere and E. D. Sontag "Remarks on Control Lyapunov Functions for Discontinuously Stabilizing Feedback," *Proceeding of the IEEE International Conference on Decision and Control*, San Antonio, TX (1993) pp. 2398–2403.
9. J. Ren and K. A. McIsaac, "A Hybrid-Systems Approach to Potential Field Navigation for a Multi-Robot Team," *Proceeding of the IEEE International Conference on Robotics and Automation*, Taiwan **3** (2003) pp. 3875–3880.
10. 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 (2005) pp. 209–214.
11. 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).
12. H. S. Shim and Y. G. Sung, "Stability and four-posture control for nonholonomic mobile robots," *IEEE Trans. Robot. Autom.* **20**(1), pp. 148–154 (2004).
13. H. G. Tanner and K. J. Kyriakopoulos, "Nonholonomic Motion Planning for Mobile Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation* (2000) pp. 1233–1238.
14. H. G. Tanner, S. Loizon and K. J. Kyriakopoulos, "Nonholonomic Stabilization with Collision Avoidance for Mobilr Robots," *Proceedings of the IEEE International*

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press

Conference on Intelligent Robots and Systems (2001) pp. 1220–1225.
15. H. G. Tanner, S. G. Loizou and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.* 19(1), 53–64 (2003).
16. Hilare 2 Robot Specifications. Available: http://www.laas.fr/ matthieu/robots/h2+

Appendix

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

Proof Refer to Fig. 13. We use $dg_j = \partial g_j / \partial q$ 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_{vj} and define it as:

$$\theta_{vj} = \arctan\left(\frac{\partial g_j}{\partial y}, \frac{\partial g_j}{\partial x}\right) \tag{26}$$

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

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

$$\Theta'_j = \theta_{vj} - \theta_{ng}. \tag{27}$$

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:

$$\Theta_j = \begin{cases} \Theta'_j & \text{if } -\pi \leq \Theta'_j \leq \pi \\ \Theta'_j - 2\pi, & \text{if } \Theta'_j > \pi \\ \Theta'_j + 2\pi, & \text{if } \Theta'_j < -\pi \end{cases}. \tag{28}$$

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

$$\Theta_{j\,\min} = \begin{cases} \Theta_j - \pi/2, & \text{if } \Theta_j > 0 \\ -\pi/2, & \text{if } \Theta_j < 0 \end{cases} \tag{29}$$

$$\Theta_{j\,\max} = \begin{cases} \Theta_j + \pi/2, & \text{if } \Theta_j < 0 \\ \pi/2, & \text{if } \Theta_j > 0 \end{cases}. \tag{30}$$

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.

$$\Theta_{\min} = \max\{\Theta_{1\,\min}, \Theta_{2\,\min}, \ldots, \Theta_{m\,\min}\} \tag{31}$$

$$\Theta_{\max} = \min\{\Theta_{1\,\max}, \Theta_{2\,\max}, \ldots, \Theta_{m\,\max}\}. \tag{32}$$

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

https://doi.org/10.1017/S0263574707003694 Published online by Cambridge University Press