[5] A. Rauf and J. Ryu, "Fully autonomous calibration of parallel manipulators by imposing position constraint," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, 2001, pp. 2389–2394.
[6] H. Zhuang and L. Liu, "Self-calibration of a class of parallel manipulators," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, pp. 994–999.
[7] H. Zhuang, "Self-calibration of parallel mechanisms with a case study on Stewart platforms," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 387–397, Jun. 1997.
[8] C. W. Wampler, J. M. Hollerbach, and T. Arai, "An implicit loop method for kinematic calibration and its application to closed-chain mechanisms," *IEEE Trans. Robot. Autom.*, vol. 11, no. 5, pp. 710–724, Oct. 1995.
[9] Y. J. Chiu and M. H. Perng, "Self-calibration of a general hexapod manipulator with enhanced precision in 5-DOF motions," *Mech. Mach. Theory*, vol. 39, no. 1, pp. 1–23, 2004.
[10] H. Zhuang, J. Yan, and O. Masory, "Calibration of Stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals," *J. Robot. Syst.*, vol. 15, no. 7, pp. 395–405, 1998.
[11] A. Nahvi, J. M. Hollerbach, and V. Hayward, "Calibration of parallel robot using multiple kinematic closed loops," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, pp. 407–412.
[12] P. Maurine and E. Dombre, "A calibration procedure for the parallel robot Delta 4," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1996, pp. 975–980.
[13] H. Ota, T. Shibukawa, T. Tooyama, and M. Uchiyama, "Forward kinematic calibration method for parallel mechanism using pose data measured by a double ball bar system," in *Proc. Year 2000 Parallel Kinematic Mach. Int. Conf.*, 2000, pp. 57–62.
[14] Y. Takeda, G. Shen, and H. Funabashi, "A DBB-based kinematic calibration method for in-parallel actuated mechanisms using a Fourier series," Proc. Des. Eng. Tech. Conf./Comput. Inf. Eng. Conf. DETC2002/MECH-34345, 2002, pp. 1–10.
[15] S. Besnard and W. Khalil, "Calibration of parallel robots using two inclinometers," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1999, pp. 1758–1763.
[16] J. I. Jeong, D. Kang, Y. M. Cho, and J. Kim, "Kinematic calibration for redundantly actuated parallel mechanisms," *J. Mech. Des.*, vol. 126, pp. 307–318, 2004.
[17] S. Besnard and W. Khalil, "Identifiable parameters for parallel robots kinematic calibration," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seoul, Korea, 2001, pp. 2859–2866.
[18] A. Rauf, S. G. Kim, and J. Ryu, "Complete parameter identification of parallel manipulators with partial pose information using a new measurement device," *Robotica*, vol. 22, pp. 689–695, 2004.
[19] ——, "A new measurement device for kinematic calibration of parallel manipulators," Proc. Des. Eng. Tech. Conf./Comput. Inf. Eng. Conf. DETC2004-57375, 2004, pp. 1–8.
[20] J. P. Merlet, *Parallel Robots*. Amsterdam, The Netherlands: Kluwer, 2000.
[21] I. Fassi and G. Legnani, "Automatic identification of a minimum, complete and parametrically continuous model for the geometrical calibration of parallel robots," in *Proc. Workshop Fundamental Issues, Future Res. Directions Parallel Mech. Manip.*, Quebec, QC, Canada, 2002, pp. 204–213.
[22] A. Rauf, "Kinematic calibration of parallel manipulators with partial pose information," Ph.D. dissertation, Dept. Mechatron., Gwangju Inst. Sci. Technol., , Gwangju, Korea, Jun. 2005.
[23] dSPACE [Online]. Available: http://www.dspace.de/ww/en/pub/home.htm
[24] Jewell Instruments [Online]. Available: http://www.jewellinstruments.com
[25] BEI Technologies, Inc. [Online]. Available: http://www.bei-tech.com/
[26] Schaevitz Sensors [Online]. Available: http://www.msiusa.com/schaevitz/
[27] ControlDesk [Online]. Available: http://www.dspace.de/ww/en/pub/products/sw.htm
[28] C. J. Stone, *A Course in Statistics and Probability*. Pacific Grove, CA: Duxbury Press, 1996.

Modified Newton's Method Applied to Potential Field-Based Navigation for Mobile Robots

Jing Ren, Kenneth A. McIsaac, and Rajni V. Patel

***Abstract*—This paper investigates the inherent oscillation problem of potential field methods (PFMs) in the presence of obstacles and in narrow passages. These problems can cause slow progress and system instability in implementation. To overcome these two problems, in this paper, we propose a modification of Newton's method. The use of the modified Newton's method, which applies anywhere C_2 continuous navigation functions are defined, greatly improves system performance when compared to the standard gradient descent approach. To the best of our knowledge, ours is the first systematic approach to the oscillation problems in PFMs. We have validated this technique by comparing its performance with the gradient descent method in obstacle-avoidance tasks with different potential models and parameter changes.**

***Index Terms*—Mobile robot navigation, optimization, potential field methods (PFMs).**

I. Introduction

The potential field method (PFM) is widely used in mobile robot navigation for its simplicity and elegance [1]–[6]. However, this approach has some significant inherent limitations. One severe limitation is the problem of oscillation in the presence of obstacles and in narrow passages, which has been identified in [7] and [8]. In [8], Koren and Borenstein showed that PFM-based methods may result in oscillations when the discrete system resulting from sampling of sensory inputs becomes unstable, which happens quite often when a robot moves in a long narrow passage or near obstacles. From our observations and analysis, we have found that even when the discrete system is stable, oscillations can still occur because the gradient descent method, which is traditionally used in potential field navigation, generates an oscillatory trajectory.

In the gradient descent approach, the control law for a robot navigating in a potential field uses the negative gradient direction to determine a vector that points toward the target. Whenever we consider a discrete system and the potential contour is not perfectly circular, solutions tend to exhibit oscillation, especially in proximity to obstacles or in narrow passages. To overcome oscillations, in practice, this phenomenon requires that controller sampling rates must be small or robot speeds must be slow. When a robot is in a complex environment with multiple obstacles and narrow passages, the use of the gradient method can result in an extremely long task completion time because many "unnecessary" control maneuvers are generated. A robot required to complete a navigation task in a limited time can be considered to have failed in its task if excessive oscillations cause it to exceed this limit. An additional problem in practice is that, even when the continuous model of a navigation system can be guaranteed to be stable, oscillatory controls generated by the gradient descent method will often lead

Manuscript received June 10, 2005; revised November 3, 2005. This paper was recommended for publication by Associate Editor D. Fox and Editor F. Park upon evaluation of the reviewers' comments. This work was supported by the National Science and Engineering Research Council (NSERC) of Canada under Grants RGPIN-249883-02 and RGPIN-1345. This paper was presented in part at the Conference on Decision and Control, Maui, HI, 2003.

The authors are with the Department of Electrical and Computer Engineering, University of Western Ontario, London, ON N6G 1H1, Canada (e-mail: kmcisaac@engga.uwo.ca).

Digital Object Identifier 10.1109/TRO.2006.870668

to instabilities in a discrete time implementation, unless the sampling rate is increased to match the oscillations in the trajectory.

This phenomenon can be explained intuitively from the optimization point of view. It is well known in optimization theory [9] that the gradient method exhibits oscillation and slow convergence in the neighborhood of an 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 in the neighborhood of an optimum. Similarly, in potential field-based navigation, whenever the potential field is severely deformed due to a complex configuration of targets and obstacles and exhibits a strongly noncircular shape, the navigation function becomes a complex nonlinear function.

One solution that is used in optimization theory to overcome the problem of oscillations is to introduce higher order information into the optimization problem. In this paper, motivated by the techniques of optimization theory, we propose the use of the modified Newton's method (MNM), which contains both first and second derivative information, for mobile robot navigation. The MNM can greatly alleviate oscillations by approximating the navigation function with a quadratic form. This method retains the merits of the gradient-based approach while at the same time eliminating oscillation, achieving faster task completion, allowing higher robot speeds, and greatly reducing failure rate. We believe, to the best of our knowledge, that this study represents the first time that ideas from optimization theory have been introduced to reduce oscillations in potential field-based motion planning.

It is worth mentioning that the use of the MNM requires very modest computation effort. In optimization, a high computational cost is considered to be the main drawback of using the MNM. However, this problem does not exist for the mobile robotics problem considered here, which is a low-dimensional problem.

This paper is organized as follows. In Section II, we recall the typical gradient method and illustrate its inherent oscillation phenomenon. In Section III, we present MNM, and then, in Section IV, we show how it can be applied to the problem of mobile robot navigation in an artificial potential field to overcome the oscillation problem and achieve faster movement. In Section V, we perform a thorough performance comparison for the gradient method and MNM for various scenarios and different potential models. In Section VI, we conclude the paper and suggest some future research directions.

II. PFM and the Inherent Oscillation Phenomenon

For simplicity, in this section, we only discuss mobile robots operating in two dimensions, but the results can be applied to mobile robots operating in three dimensions as well. We use q to denote the configuration of each omnidirectional robot: $q = (x, y)$, which is the position of the robot's center. Typically, control laws for PFMs generate motion controls along gradient descent directions [10] as

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

where $V(q)$ is a potential field navigation function.

To facilitate the discussion, in this paper, we adopt the notion of "step size" from optimization theory [9] and define it as the product of the speed and the sampling interval. From optimization theory, the gradient descent method is inherently associated with the oscillation problem due to its orthogonal search directions and the use of optimal step size in long, narrow-contour valleys. However, because we do not perform a one-dimensional (1-D) search for the optimal step size in PFM applications, oscillations in PFMs come not only from the stable oscillatory gradient trajectory, which roughly corresponds to the definition of oscillation in optimization theory, but also from unstable motions due to instability of the discrete system resulting from sampling of sensory inputs [8]. Although oscillations in PFMs is a blend of the two types, both types of oscillations result from the use of the gradient descent method. In the rest of this paper, we will not distinguish between these two types of oscillations.

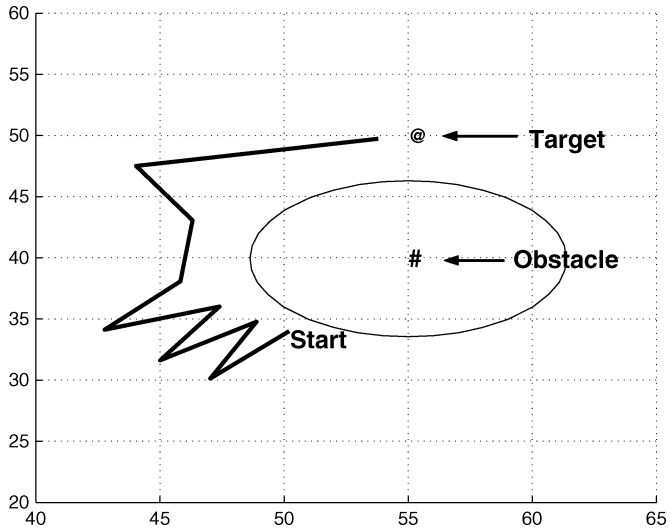

Fig. 1. Illustration of the oscillation phenomenon. In this figure and elsewhere in this paper, we use @ to represent the target and # to represent the obstacle.

In Fig. 1, we illustrate the oscillation problem in PFMs with a simple goal reaching application. The robot's trajectory exhibits strong oscillation when it is close to the obstacle.

III. Modified Newton's Method

To overcome the problem of oscillation, we use a modification of Newton's method to adjust the gradient. The operation is motivated by finding a suitable direction for the quadratic approximation to the navigation function rather than using a linear approximation as in the gradient descent method. The dynamics of each robot based on the MNM is given by the control law

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

In (2), the matrix $B(q)$ is based on the MNM. As in optimization theory [9], $B(q)$ is a positive-definite matrix and is defined as $B(q) = (\varepsilon I + H)^{-1}$, where $\varepsilon = \varepsilon(q) \geq 0$ and H is the Hessian matrix of $V(q)$. It is the presence of the inverse of $\varepsilon I + H$ that makes the MNM unattractive in many optimization problems. Since the matrix inverse scales with the cube of the dimensionality of the problem, the computational cost for inverting this matrix at every step can become prohibitive. Fortunately, in the low-order problem we address in this paper, which is that of a single mobile robot navigating in the plane, this inverse is available in closed form and requires only very modest additional computation. There is also a straightforward extension to a single free-flying robot navigating in three dimensions that does not involve a large computational cost. For very high dimensional problems, such as robot manipulators, computational cost may make the MNM unsuitable for real-time motion planning. Even in such high-dimensional problems, which are beyond the scope of this paper, there are other techniques from optimization theory (such as the conjugate gradient approach) that might be used to eliminate oscillations. This is an opportunity for future work.

In mobile robot navigation, it is more practical to adopt a constant speed (step size) rather than optimal step size in order to let robots

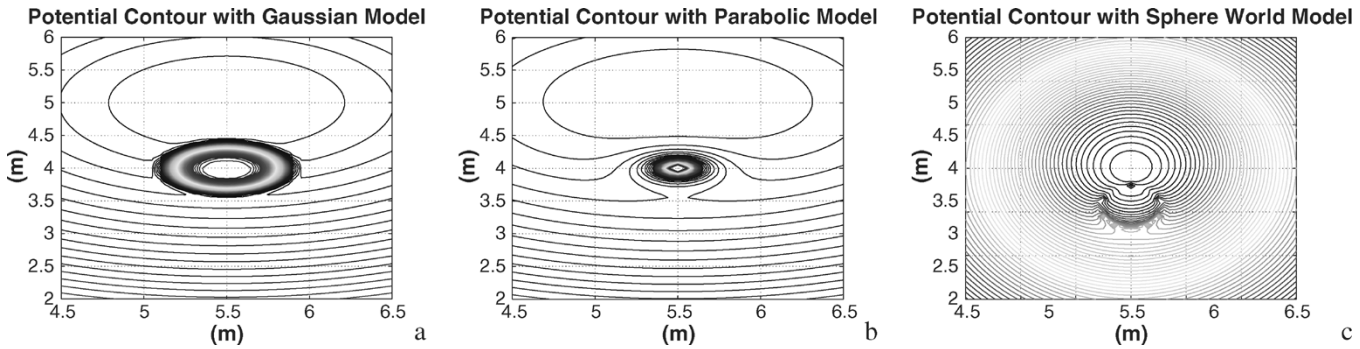

Fig. 2. Contours of three potential models: generalized Gaussian potential function, parabolic potential function, and generalized sphere world models.

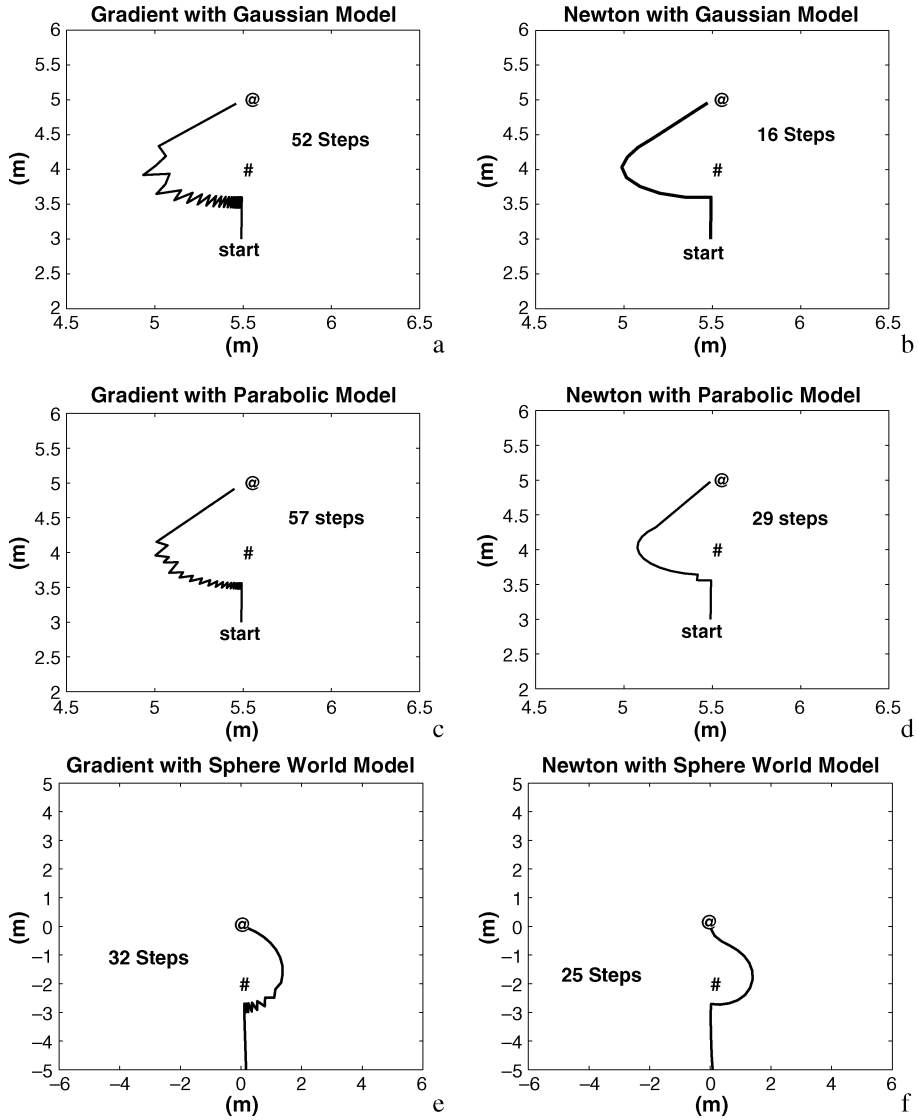

Fig. 3. Performance comparisons of the gradient descent and the MNM using three potential models: generalized Gaussian potential function, parabolic potential function, and generalized sphere world models.

move smoothly. The control law along gradient descent directions with a constant speed is defined as

$$\dot{q} = \alpha d_{n_g} \tag{3}$$

where α is a constant speed parameter and $d_{n_g} = -(\partial V(q)/\partial q)/(|\partial V(q)/\partial q|)$ is the unit negative gradient.

The corresponding control law using MNM is defined as

$$\dot{q} = -\alpha \frac{B(q)\frac{\partial V}{\partial q}}{\left|B(q)\frac{\partial V}{\partial q}\right|}. \tag{4}$$

The major difference between our version of the MNM and the definition of the MNM in the literature of optimization is that the step size

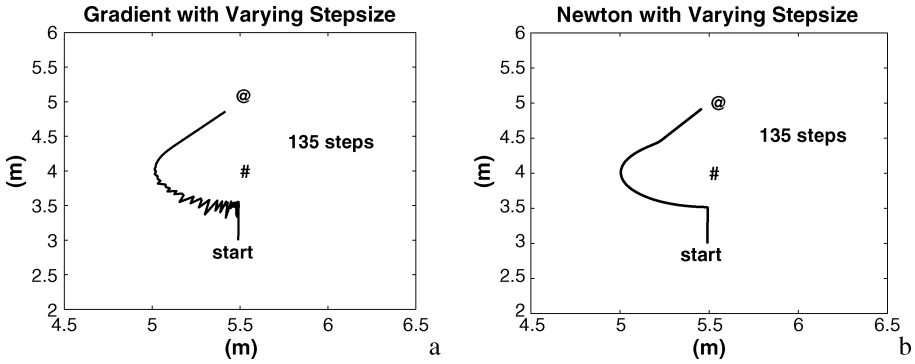

Fig. 4. Oscillation phenomenon with varying step sizes. (a) Gradient method. (b) MNM.

(speed) in the literature of optimization can be optimized, but, in our applications, it is restricted by the capability of the physical robots and is assumed to be constant. We show later that this restriction does not affect the utility of the technique.

The modified Newton's method has the following three advantages over the gradient descent method in PFM applications:

1) reduced task completion time;
2) smoother trajectories;
3) longer steps: unstable oscillations may occur when oscillation is present in the trajectory and the step size becomes too large. By comparing these two methods, we found that the MNM, because of its inherently smoother trajectory, allows a much bigger step size than the gradient descent method.

IV. Application of the MNM

A. Applying the MNM to Different Potential Models

In PFM applications, oscillations depend greatly on the shape of the potential contour, which is determined by the potential field model chosen to construct the navigation function $V(q)$. In this section, we compare the MNM and the gradient descent method in several potential models. We found that the MNM can greatly improve performance independent of the choice of potential models.

In this study, we limit our interest to obstacles of convex shape. PFM motion planners very commonly become "stuck" in concavities, because concavities create local minima in the potential field from which the motion planner (using gradient descent or the MNM) cannot escape. We consider the problem of local minima to be orthogonal to the problem of oscillations, and therefore we have not addressed it in this study. Instead, we focus on convex obstacles and do not attempt to solve the problem of local minima in the navigation function.

The navigation functions $V(q)$ for the different potential models (i.e., generalized Gaussian, parabolic model, and generalized sphere world) that we have studied in this paper can be found in [2], [5], and .

In Fig. 2(a)–(c), we show the contour plots of the typical target-reaching task in Fig. 1 using the generalized Gaussian potential function, the parabolic potential function, and the generalized sphere world model, respectively. In Fig. 3(a), (c), and (e), we show the trajectories generated by the gradient descent method, and in Fig. 3(b), (d), and (f), we show the corresponding trajectories using the MNM. From this typical example, we show that the presence of the obstacle creates a noncircular shape around the obstacle in the contour plot, and the MNM, compared with the gradient descent method, can better follow the contour and achieve much smoother and faster progress.

Due to space limitations, we have presented three potential models to illustrate the advantages of the MNM. However, the results are applicable to most of the existing potential models [10], [12], [13] in the literature. The gradient descent method's inability to find a good approximation to noncircular contours in the presence of interaction between attractive and repulsive navigation functions is common among potential models and becomes more pronounced as the number of obstacles and targets increases.

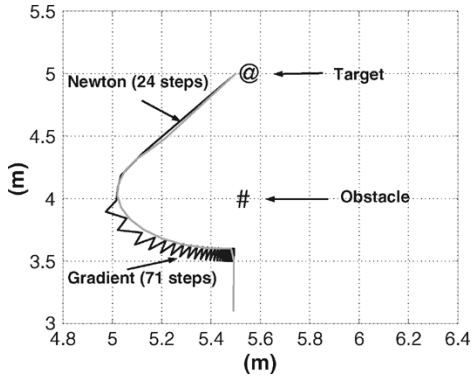

Fig. 5. MNM (which is represented by the dotted line) is faster and smoother than the gradient method (which is represented by the solid line).

TABLE I
Performance Comparisons in the Single-Obstacle Case

Model	Planner	start= 0.001	start = 0.01	start=0.1
Gauss.	Grad.	115	99	82
Gauss.	MNM	31	31	31
Para.	Grad.	89	73	57
Para.	MNM	29	29	29
S.W.	Grad.	28	25	22
S.W.	MNM	17	17	17

B. Applying the MNM With Varying or Fixed Speed

Although the MNM and the gradient descent method both require a 1-D optimization to determine the step size in optimization theory, we modify them to adopt a fixed speed (step size) in many applications in mobile robotics. One reason is that searching for 1-D optimization increases the computation time that is limited in real-time applications; another reason is that frequent changes to robot speed and/or sampling rate are not practical, due to constraints on the robot's dynamics. In this section, we show that the benefits of the MNM exist independently of the choice of fixed or changing step size.

It is difficult in simulations using optimal step sizes to ensure that the two different robots takes steps of equal sizes in each simulation. Instead, in the simulations of Fig. 4, we allow the step sizes to vary but force the total number of steps to be the same.

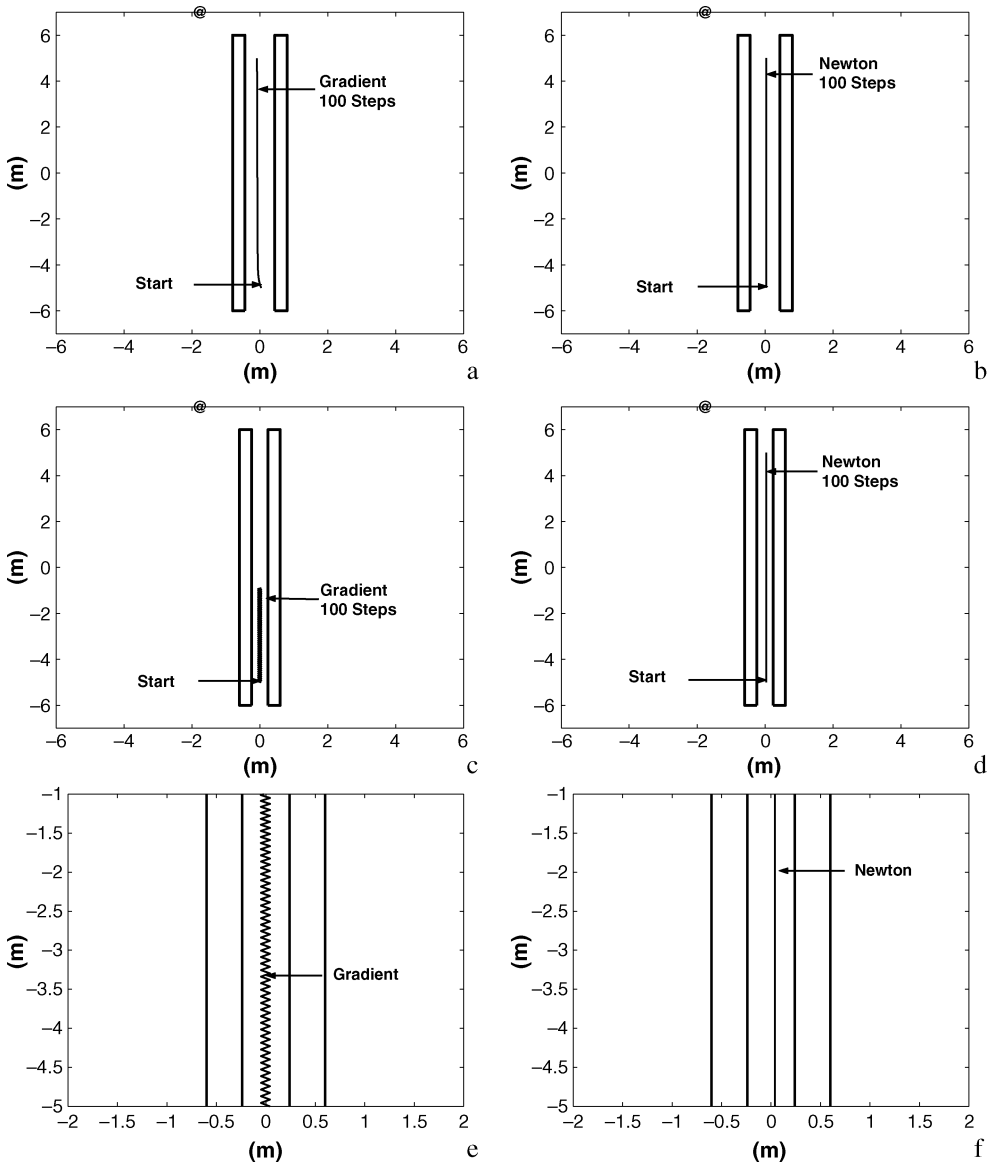

Fig. 6. (a), (c) Performance of the gradient method in wide (0.88-m) and narrow (0.48-m) passages, respectively. (b), (d) Corresponding trajectories for the MNM in the same scenarios. (e), (f) Parts of the enlarged version of (c) and (d).

We observe a large amount of oscillation in the trajectory generated by the gradient descent method, and the constraint of equal completion times means that the MNM controller is forced to adopt an extremely small step size. In practical terms, if we had allowed the MNM controller to move as quickly as the gradient descent controller, it would complete the task more rapidly.

V. Performance Comparison

An experiment on the 2.4-GHz Pentium IV machine shows that, for each step, the time to calculate the gradient direction is 1.56 μs and to calculate the MNM direction is 3.94 μs. For many vision-based robots, the maximum sampling rate is less than 10 Hz due to the time devoted to image processing. For these applications, the time to calculate the gradient is negligibly small compared to the time required for other computations in each step such as image processing. Since the speeds are the same for both methods and there is only a slight difference in the sampling rate, the unit "step size" (defined in Section II) is almost the same for the two methods. Therefore, the number of steps is a metric equal to the total completion time.

Consider a typical Hilare-type robot quipped with one camera, where the typical speed of the robot is 0.5 m/s. The sampling interval is set to be 200 ms, during which it needs to perform some image processing for navigation for each step. The step size is therefore ≈ 0.1 m. The performance of the gradient and the MNM in this case is illustrated in Fig. 5.

Based on this analysis, the performance of the two methods will be mainly evaluated on two criteria:

- how quickly (in steps) the robot can reach the target;
- the smoothness of the trajectory.

In some scenarios, zigzagging can be so severe that it will require a large number of steps to reach the goal, so we define a notion of "failure" for the cases in which the robot cannot reach the goal within a maximum number of steps. This choice is motivated by practical applications, in which the inability to complete a task within a maximum time is considered to be a failure. In addition, allowing indefinite completion times would introduce a bias when we use the average time to compare the gradient method and the MNM. The maximum number of steps is chosen arbitrarily to be 3000 steps. The choice of the maximum

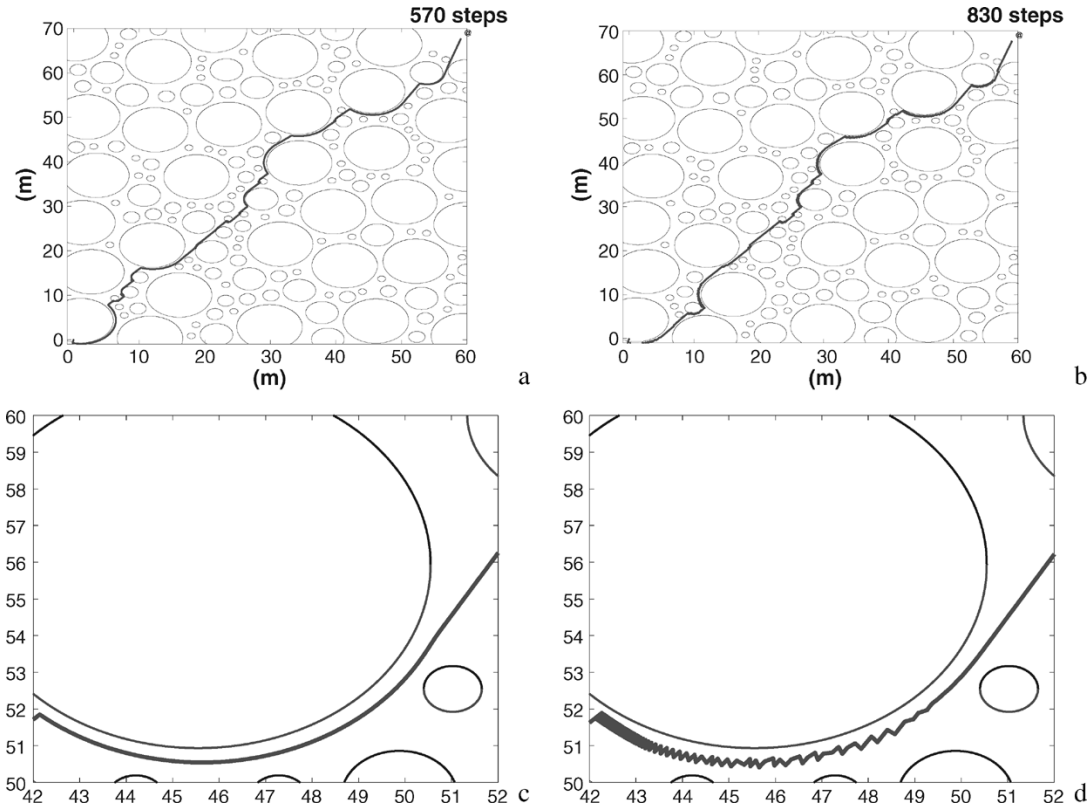

Fig. 7. Overall trajectories with random obstacles for (a) the MNM and (b) the gradient method. On a gross scale, trajectories for both methods tend to be similar. (c), (d) Close-ups of (a) and (b).

number of steps should not affect the conclusions we derived from the comparisons.

A. Single-Obstacle Case

In this section, we investigate the performance of the two algorithms in a one-obstacle scenario (as in Fig. 2) with different starting points. It is well known [10] that, when the starting point, the center of 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 a cause of oscillations.

We analyze the performance of both methods around the perfect alignment scenario and show that the MNM can maintain a much better performance compared to the gradient method as the starting point slides closer to the perfect alignment (Table I). All starting points are measured in the distance from the perfect alignment. The data presented are steps to reach the goal. In the table, "Gauss.," "Para," and "S.W." indicate the Gaussian, parabolic, and sphere world potential models, respectively; "Grad." indicates the gradient method planner and "MNM" indicates the MNM-based planner. All numbers are in steps, with each step requiring approximately 200 ms.

B. Passing Through a Passage

In PFM applications, oscillations occur independently of obstacle shapes. The results derived in this paper can be applied to convex obstacles of arbitrary shapes. To illustrate this, in this section, we compare the two methods in a task of passing through a passage (obstacles of sharp edges). The robot is moving at 0.1 m/s for all scenarios. In Fig. 6(a) and (b), we set the width of the passage to be 0.88 m, and we can see that both the gradient and the MNM move fast and the trajectories are smooth. When we narrow the passage down to 0.48 m, the MNM in Fig. 6(d) can still maintain a fast and smooth movement. The gradient method, however, exhibits a large amount of oscillation, which is shown in Fig. 6(c). We have magnified Fig. 6(c) and (d) and shown parts of the enlarged versions in Fig. 6(e) and (f).

C. Performance 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 $\times$ 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 represents a complex, unknown, random environment.

Fig. 7(a) and (b) shows trajectories of the MNM and the gradient method. We can see that both the gradient method and the MNM can reach the target through a complex obstacle area. However, the MNM reaches the target in fewer steps (570 versus 830). If we enlarge parts of Fig. 7(c) and (d), we can see the large amount of oscillation that is caused by the gradient descent method when obstacles are nearby.

We have run 1000 random trials each for three different fixed step sizes. The raw data are displayed in Fig. 8. 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). The figure illustrates both the much higher likelihood of failure with the gradient method and that the completion time with the MNM is lower in almost all cases.

Next, we performed 1000 random trials for each of the three different inter-obstacle passage sizes. The results are plotted in Fig. 9. In this case, the three sets of data correspond to an inter-obstacle spacing of 0.8, 0.7, and 0.55 m. Once again, we see that the MNM has lower

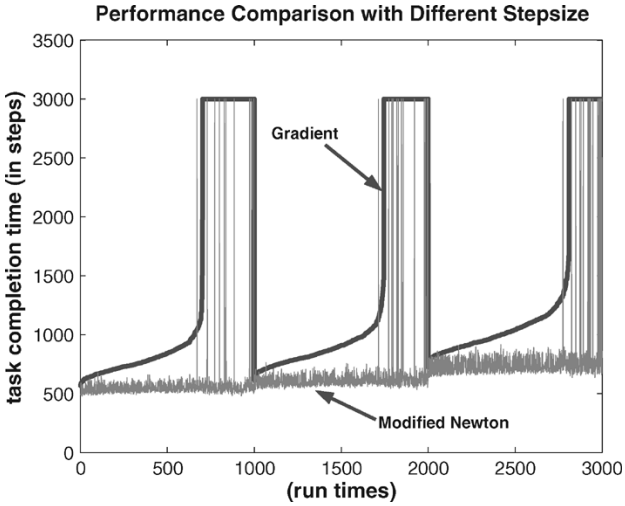

Fig. 8. Raw data to compare the performance for 1000 runs with three step sizes: 0.15, 0.18, and 0.2 m.

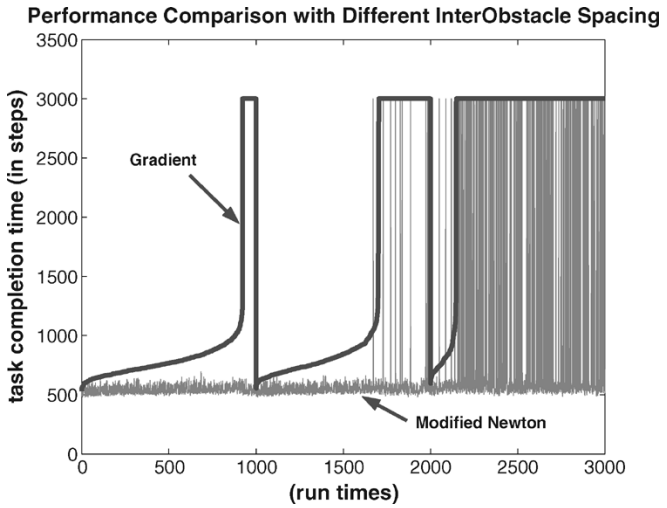

Fig. 9. Plot of the raw data to compare the performance for different choices of the minimum inter-obstacle spacing; the first 1000 runs are for spacing $= 0.8$ m, the second 1000 runs are for spacing $= 0.7$ m, and the last 1000 runs are for spacing $= 0.55$ m. The failure rate (requiring more than 3000 steps) is 853/1000 for the gradient method and 201/1000 for the MNM.

average completion times and a much lower failure rate than the gradient descent method does.

Finally, we performed 1000 random trials for each of the three different protective range of obstacles. The results are presented in Fig. 10. As in our previous two cases, the MNM controller performs better than the gradient descent method does (see Fig. 11).

In Table II, we show the average numbers of steps to completion for successful runs with the MNM and the gradient method for each scenario.

D. Observations From Simulations

We observe the following from our simulation results.

- When the passage width becomes narrower, the gradient method deteriorates dramatically and tends to fail in many cases. The MNM, however, can tolerate a much narrower passage with good performance.
- In general, increasing the size of the obstacles creates bigger oscillations for the gradient method. An intuitive explanation is that the obstacles of bigger size cause longer valleys in the contour, which the gradient method has difficulty in following.
- Comparing the two algorithms, we see a much higher failure rate with the gradient method in all cases.
- The 0.68% of cases in which the gradient method outperforms the MNM result from the fact that the PFM is inherently a local method and the randomness of test construction. In some pathological cases, a locally smart choice of direction around one obstacle can confront a robot with a large crowd of obstacles that would be avoided otherwise.

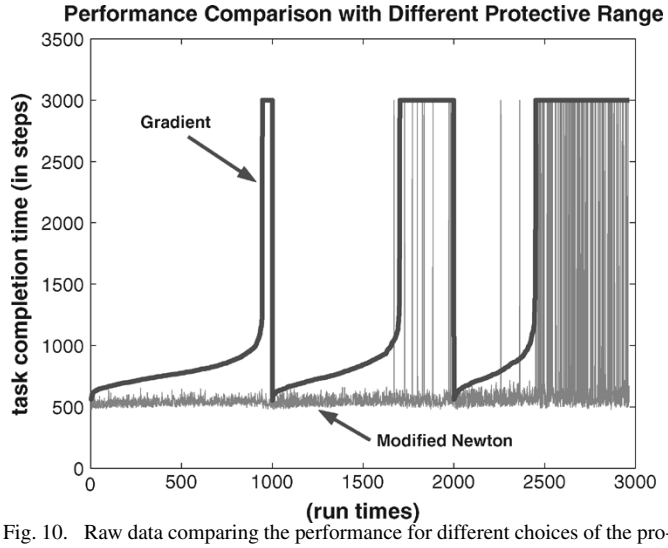

Fig. 10. Raw data comparing the performance for different choices of the protective range of obstacles. First 1000: variance $= 4$; second 1000: variance $= 5$; third 1000: variance $= 6$.

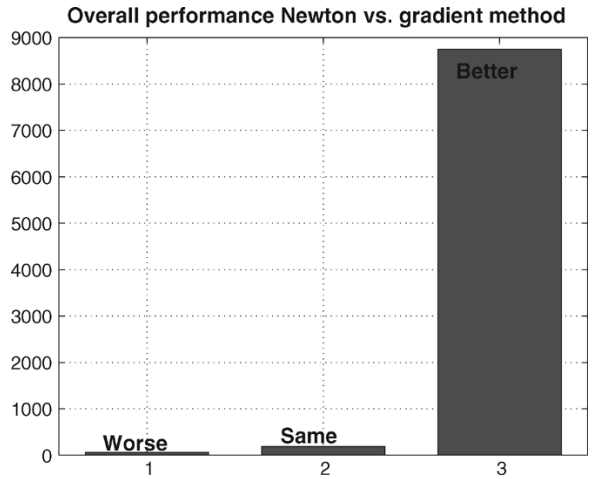

Fig. 11. Histogram showing relative performances of the MNM and gradient controllers. The MNM outperformed the gradient method in 97.21% of cases and performed equally well in 2.11% of cases.

TABLE II
PERFORMANCE COMPARISON FOR AVERAGE NUMBER OF STEPS

Model		Gradient (steps)	Newton (steps)
Stepsize (m)	0.15	1401	783
	0.18	1419	647
	0.2	1454	577
Spacing (m)	0.55	2680	1040
	0.7	1452	575
	0.8	917	548
Variance	4	910	541
	5	1453	576
	6	1968	886

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

VI. Conclusion

In this paper, we have proposed a general solution to the inherent oscillation problem in potential field-based navigation by proposing the use of the MNM. We have performed a thorough comparison between these two algorithms, and the results show that the proposed MNM can greatly improve the overall system performance for all scenarios considered.

A major difficulty in using the MNM in optimization theory is the computation time. However, in the problems considered in this paper, the use of the MNM only requires a very modest computational effort due to the low-dimensional nature of the mobile robot motion planning problem.

For high-dimensional problems, when the computational cost of the MNM becomes prohibitive for real-time use, other optimization techniques, such as the conjugate gradient method, may prove useful in eliminating oscillations. We also propose that the MNM may still prove useful in high-dimensional problems as an off-line path planner, for example, for a manipulator arm or a parallel robot. The actual trajectory could then be generated by exploiting other control methods to track the predefined path.

Although we consider kinematic models in this paper, MNM is also applicable to robots with dynamic models. In preliminary work with dynamic models, we have observed similar performance improvements.

References

[1] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1986, pp. 1419–1424.

[2] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Robot. Autom.*, vol. 16, no. 5, pp. 609–615, Oct. 2000.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[4] D. E. Koditschek, "Exact robot navigation by means of potential functions: Some topological consideration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1987, pp. 1–6.

[5] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. Robot. Autom.*, vol. 8, no. 5, pp. 501–518, Oct. 1992.

[6] R. Volpe and P. Khosla, "Manipulator control with superquadric artificial potential functions: theory and experiments," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, no. 6, pp. 1423–1436, Nov./Dec. 1990.

[7] B. H. Krogh, "A generalized potential field approach to obstacle avoidance," *Robot. Res.*, 1984, MS84-484/1-15.

[8] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1991, pp. 1398–1404.

[9] M. Bazaraa, H. Sherali, and C. Shetty, *Nonlinear Programming Theory and Algorithms*. New York: Wiley, 1995.

[10] J. C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer, 1991.

[11] J. Ren and K. A. McIsaac, "A hybrid-systems approach to potential field navigation for a multi-robot team," in *Proc. IEEE Int Conf. Robot. Autom.*, Sep. 2003, pp. 3875–3880.

[12] P. Khosla and R. Volpe, "Superquadric artificial potentials for obstacle avoidance and approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1988, pp. 1778–1784.

[13] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of cooperating mobile manipulators," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 53–64, Feb. 2003.

Stability Analysis of a Simple Walking Model Driven by an Oscillator With a Phase Reset Using Sensory Feedback

Shinya Aoi and Kazuo Tsuchiya

***Abstract*—This paper deals with the analytical examination of the dynamic properties of the walking motion of a biped robot based on a simple model. The robot is driven by rhythmic signals from an oscillator, which receives feedback signals from touch sensors at the tips of the legs. Instantly, the oscillator resets its phase and modifies the walking motion according to the feedback signals. Based on such a simple model, approximate periodic solutions are obtained, and the stability of the walking motion is analytically investigated by using a Poincaré map. The analytical results demonstrate that the modification of the step period and the walking motion due to the sensory feedback signals improves the stability of the walking motion.**

***Index Terms*—Central pattern generator (CPG), phase reset, Poincaré map, sensory feedback signal, stability analysis.**

I. Introduction

Rhythmic motions such as animal walking are achieved by interaction between the dynamics of a musculoskeletal system and the rhythmic signals from the central pattern generator (CPG) [6], [12]. The CPG comprises a set of neural oscillators present in the spinal cord, and spontaneously generates rhythmic signals even if it does not receive outer signals, such as a sensor signal. However, it is very sensitive to outer signals and modifies the rhythmic signals influenced by the outer signals, resulting in adaptive motions. CPG is widely modeled using nonlinear oscillators. Many studies have been carried out in order to elucidate the role for CPG in locomotion using quadruped robots [4], [16], biped robots [1], [10], [15], a simulated salamander [8], and human models [11], [13], [14], [17]. In particular, a recent approach that incorporates the resetting of the CPG's phase, depending on outer signals, has been proposed [1], [10], [15]–[17]. For example, in our previous work [1], [15], a locomotion control system for a biped robot was developed using nonlinear oscillators. The nominal trajectories of the joints are designed by maps from the phases of the oscillators. A controller composed of nonlinear oscillators receives feedback signals from touch sensors. Instantly, the phase of the oscillator is reset, and the nominal trajectories of the joints are modified according to the phase reset. Studies revealed that the phase reset achieves robust walking motions by numerical simulations and hardware experiments. However, they are not sufficient to clarify the adaptability mechanism. The purpose of this paper is to clarify

Manuscript received February 7, 2005; revised June 7, 2005. This paper was recommended for publication by Associate Editor Q. Huang and Editor H. Arai upon evaluation of the reviewers' comments. This work was supported in part by the Center of Excellence for Research and Education on Complex Functional Mechanical Systems (COE Program of the Ministry of Education, Culture, Sports, Science and Technology, Japan), and in part by a Grant-in-Aid for Scientific Research on Priority Areas "Emergence of Adaptive Motor Function through Interaction between Body, Brain, and Environment" from the Japanese Ministry of Education, Culture, Sports, Science and Technology. This paper was presented at the International Conference on Systems, Man, and Cybernetics, Hague, The Netherlands, October 2004.

The authors are with the Department of Aeronautics and Astronautics, Graduate School of Engineering, Kyoto University, Kyoto 606-8501, Japan (e-mail: shinya_aoi@kuaero.kyoto-u.ac.jp; tsuchiya@kuaero.kyoto-u.ac.jp).

Digital Object Identifier 10.1109/TRO.2006.870671