The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

Trajectory Control of Wheeled Mobile Robots Based on Virtual Manipulators

Kimitoshi Yamazaki and Masayuki Inaba

Abstract— **This paper describes a novel method of path following for wheeled mobile robots. A number of virtual manipulators mounted on the robot are used to make the robot poses which satisfy both of path following and collision avoidance. This method can calculate robot poses by using a single jacobi matrix despite of coping with several manipulators. We also present about the method of motion stabilization by using a singularity-avoiding criteria. Effectiveness of the proposed method is proven by means of simulation in both cases of omnidirectional wheelbase and 2 wheeled mobile platform.**

I. Introduction

In real environments as jumbled rooms or public crowded space, mobile robots should not collide with obstacles to arrive at its target position. In order to cope with such issues, many planning methods have been proposed and they often provided feasible solutions. In the case of simple environment, A* algorithm is one of basic ways to search path from start to goal. In dynamic environment, the algorithm can be extended as D*[18] and D*-Lite[11]. More complex case, randomized exploring method have often been utilized. For instance, there are RRT[12] and its extension[2][6]. Probabilistic roadmap(PRM)[8] is also general approach[13].

Some methods mentioned above provide a feasible path even in dynamic or crowded environment from the viewpoint of global path planning. On the other hand, one of the issues is to ensure motion smoothness because these methods are often based on grid space or randomly scattered nodes. Especially, it is difficult for two wheeled mobile platform to follow a planned path if its motion constraint does not considered. In such case, local motion modifier can generate a path which satisfies both path following and collision avoidance.

Reactive planning methods have potential for adjusting the issue. Typical past researches about such planning is Dynamic Window Approach [9] and Potential Field[7][9][10]. These approaches permit the robot to avoid dynamic or unforeseen obstacles adaptively. Our method based on virtual manipulators is classified as reactive planning. The manipulators enable the robot to change its direction of movement within the scope of maintaining of given path. That is, if the robot finds some static or dynamic obstacles locally, it will plan the avoiding path on the spot by imposing reactive force from the obstacles.

Because whole states of the manipulators can be represented by a single jacobi matrix, our method only needs an iterative calculation for the renewal of the robot pose. One of the advantages of our approach is that the shape of the mobile platform is explicitly considered in the pose calculation, unlike traditional potential based approach.

This paper is organized as follows: Section II addresses issues and an approach. Section III and IV introduces our control method by using virtual manipulators. Section V indicates an algorithm to satisfy both path following and collision avoidance. Section VI shows some simulation results to show the effectiveness of our method. Section VII presents our conclusion.

II. Approach

Mobile manipulators which mount mechanical arms on a mobile platform have been utilized as research platform. One of the research field is to detect mobile platform positions in which highly manipulability can be ensured [14][17]. In other researches, path following methods have also been proposed [5]. Bayle et al.[1] extended a definition of manipulability to the case of a nonholonomic mobile platform. They formulated a jacobi matrix for a mobile manipulator, and achieved path following based on a trajectory given to an end-effector.

In our method, several manipulators virtually mounted on the robot are used to follow a given path. One manipulator has the role of leading along given path. We call this manipulator "Leader" in the rest of this paper. On the other hand, other manipulators have the role of collision avoidance. We call these manipulators "Assistants". Because whole of motion parameters derived from manipulators and mobile platform is defined by a single jacobi matrix, the pose of the robot can be calculated by a simple equation.

III. Trajectory control by using virtual manipulators

We cope with planar mobile robots moving over flat floor in the rest.

A. Jacobi matrix in the case of omnidirectional platform

When a manipulator which has n joints is assumed, its state vector is written as $\mathbf{q} = (q_1, q_2, ...q_n)$. The relation between the $\mathbf{q}$ and the pose of end-effector $\mathbf{x}_h = (x_h, y_h, \theta_h)$ can be represented by using following jacobi matrix:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \frac{\partial f_{xh}(\mathbf{q})}{\partial q_1} & \frac{\partial f_{xh}(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial f_{xh}(\mathbf{q})}{\partial q_n} \\ \frac{\partial f_{yh}(\mathbf{q})}{\partial q_1} & \frac{\partial f_{yh}(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial f_{yh}(\mathbf{q})}{\partial q_n} \\ \frac{\partial f_{\theta_h}(\mathbf{q})}{\partial q_1} & \frac{\partial f_{\theta_h}(\mathbf{q})}{\partial q_2} & \dots & \frac{\partial f_{\theta_h}(\mathbf{q})}{\partial q_n} \end{bmatrix}, \quad (1)$$

IRT Research Initiative, The University of Tokyo, Engineering Building No. 2, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan {yamazaki, inaba}@jsk.t.u-tokyo.ac.jp

where we set $x_h = f_{xh}(\mathbf{q})$, $y_h = f_{yh}(\mathbf{q})$, $\theta_h = f_{\theta_h}(\mathbf{q})$.

Now we assume an omnidirectional mobile platform which has 3-DOF motion elements $\mathbf{x}_b = (x_b, y_b, \theta_b)$. If a manipulator is mounted on the mobile platform, a jacobi matrix can be written as follows:

$$\mathbf{J}_{omni} = \begin{bmatrix} 1 & 0 & b_1 & \\ 0 & 1 & b_2 & \mathbf{J}_m \\ 0 & 0 & 1 & \end{bmatrix}, \quad (2)$$

where $\mathbf{J}_m$ indicates a jacobi matrix with related to the manipulator. Variables b_1 and b_2 indicates the first derivation of root position of the manipulator.

B. Jacobi matrix in the case of 2 wheeled mobile platform

The jacobi matrix which includes nonholonomic constraint of the 2 wheeled platform can be calculated by multiplying the $\mathbf{J}_{omni}$ and following matrix $\mathbf{S}(\theta_b)$[1]:

$$\mathbf{S}(\theta_b) = \begin{bmatrix} \cos(\theta_b) & 0 & 0 & \dots \\ \sin(\theta_b) & 0 & 0 & \dots \\ 0 & 1 & 0 & \dots \\ 0 & 0 & \mathbf{I}_n & \\ \vdots & \vdots & & \end{bmatrix}, \quad (3)$$

where $\mathbf{I}$ indicates unit matrix. n means number of joints of a manipulator. New jacobi matrix $\mathbf{J}_{pws}$ is calculated by the multiplication of $\mathbf{J}_{omni}\mathbf{S}(\theta_b)$.

By using these matrices, the velocity v and the angular velocity ω of the mobile platform can be calculated from $\dot{x_b} = (\dot{x_b}, \dot{y_b}, \dot{\theta_b})$. As a result, the calculation of the robot motion which adjusts to the input of the end-effector $(\dot{x}_h, \dot{y}_h, \dot{\theta}_h)$ is summarized as following equation:

$$\begin{bmatrix} v \\ \omega \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \\ \dot{\theta}_n \end{bmatrix} = \mathbf{J}^{+}_{pws} \begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{\theta}_h \end{bmatrix} \quad (4)$$

where $\mathbf{J}^{+}_{pws}$ indicates the pseudo inverse matrix of $\mathbf{J}_{pws}$. All of joint angles of the manipulator and the pose of the mobile platform can be calculated by using above equation with considering nonholonomic constraint.

Fig.1 shows an example of path following simulation. Sine curve was given to the end-effector of manipulator, and 2 wheeled mobile platform was assumed. The manipulator had 2 rotational joints. Green points indicate the input curve, and blue rectangles show pose streams of the mobile platform (Red points represent the center position of it). The platform was moved with shifting forward and back motion properly under the condition of nonholonomic constraint.

C. Basic idea of path following by using virtual manipulators

We assume a planar mobile manipulator mounted a single arm. Moreover, several optional manipulators are also permitted to be settled on the mobile platform. Now we call the default manipulator “Leader manipulator”, and call other manipulators “Assistant manipulators”.

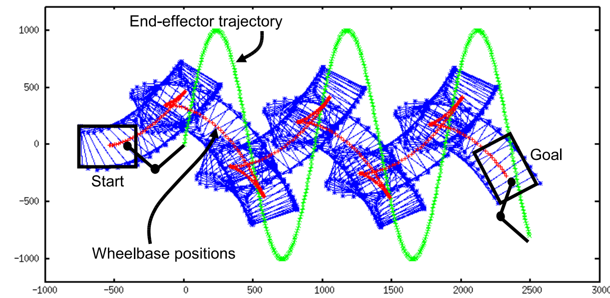

Fig. 1. Path following by using a single manipulator

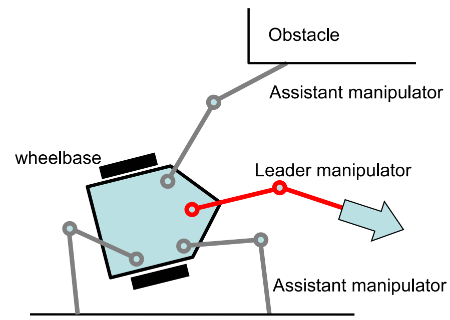

Fig. 2. Path following based on virtual manipulators

Fig.2 show the basic concept of our method. The leader manipulator has a role to keep up with a given path. On the other hand, assistant manipulators have a role to touch to arbitrary points onto environments. The aim of this approach is to generate the robot motion which satisfies not only following the leader manipulator but also generating the reactive force from assistant manipulators for collision avoidance.

In the rest of this paper, we utilize $\dot{\mathbf{x}}_\mathbf{h}$ which do not have 3 elements but only 2 elements $(\dot{x}_h, \dot{y}_h)$ as the input of an end-effector.

D. Jacobi matrix calculation considering multi-manipulators

A jacobi matrix regarding a mobile manipulator which mounts one manipulator can be divided to two matrices, that is, $\mathbf{J}_b$ and $\mathbf{J}_m$. $\mathbf{J}_b$ is calculated from the pose of the mobile platform, and $\mathbf{J}_m$ is calculated from the joints of the manipulator. When p number of manipulators are mounted on the mobile platform, and each manipulator has n joints, $(2p) \times (np+q)$ matrix is generated as follows;

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_b & \mathbf{J}_m & \mathbf{0} & \dots \\ \mathbf{J}_{b2} & \mathbf{0} & \mathbf{J}_{m2} & \dots \\ \dots & \dots & \dots & \dots \\ \mathbf{J}_{bp} & \mathbf{0} & \dots & \mathbf{J}_{mp} \end{bmatrix}, \quad (5)$$

where q becomes 3 in the case of omnidirectional platform or becomes 2 in the case of 2 wheeled mobile platform. By using this jacobi matrix, a velocity vector of the mobile platform and the manipulators can be calculated from the first derivation of their end-effector poses $\dot{\mathbf{x}} = (\dot{\mathbf{x}}_h, \dot{\mathbf{x}}_{h2}, \dot{\mathbf{x}}_{h3}, ..., \dot{\mathbf{x}}_{hp})$.

Basically, the $\dot{\mathbf{x}}_h$ which is the end-effector velocity of leader manipulator has a component to follow a given path, and other inputs $(\dot{\mathbf{x}}_{h2}, \dot{\mathbf{x}}_{h3}, ..., \dot{\mathbf{x}}_{hp})$ are set to $\mathbf{0}$ because assistant manipulators are touched to environments.

One of the characteristics of our approach is that the difference of robot's DOF between 2 wheeled mobile platform and omnidirectional platform affects only one column of the jacobi matrix. In case of approaches which use configuration space (for instance, dynamic window approach), such difference affects its searching time exponentially.

IV. Stable pose calculation

A. Issues on stable pose calculation

Our method uses a jacobi matrix which includes the information of multiple manipulators at once. Although it permits to achieve both path following and collision avoidance, there are some issues as follows:

1) Increase of singular configurations: Because several manipulators are coped with at once, whole pose of the platform becomes unstable when even one manipulator goes into singular configuration.

2) Occurrence of significant movement of mobile base when manipulators are added or deleted: The basic policy of collision avoidance is to make repulsive force by means of assistant manipulators when obstacles close to the platform. In the instant of addition or elimination of the manipulators, the size of jacobi matrix is changed. Significant pose changes may occur in some situation.

Our countermeasures are described in next subsection.

B. Pose calculation procedure

Following equation is used to calculate the whole pose of the mobile robot mounted virtual manipulators.

$$\dot{\mathbf{q}} = \mathbf{J}_w^{\#}\dot{\mathbf{x}} + \lambda(\mathbf{I} - \mathbf{J}_w^{\#}\mathbf{J})\mathbf{y}, \tag{6}$$

where $\dot{\mathbf{q}}$ is the vector including the velocity and angular velocity both of a mobile platform and manipulator's joints. $\mathbf{J}^{\#}$ indicates SR-inverse[15] of original jacobi matrix $\mathbf{J}$. $\mathbf{J}_w^{\#}$ indicates the result of multiplication $\mathbf{J}^{\#}$ and weight matrix $\mathbf{W}$. $\mathbf{y}$ is a difference vector of joint angles of the manipulators. In the case of ith joints, the element is calculated as follows:

$$y_i = (\theta_i^{ref} - \theta_i), \tag{7}$$

this equation widely affects to $\dot{\mathbf{q}}$ when present joint angles of the manipulators get away from predefined angles.

Above equation gives the pose calculation to following effects:

- by using SR-inverse, it can prevent a whole body from significantly pose changing when some manipulators go into singular configuration,
- by using a weight matrix $\mathbf{W}$ which can adjust the amount of manipulator's reaction, it can prevent the pose of mobile platform from being moved excessively,
- by using null space to maintain the distance between the mobile platform and environments, it can prevent singular configuration of manipulators.

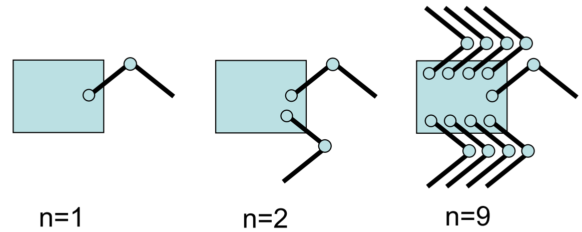

Fig. 3. Manipulator setting

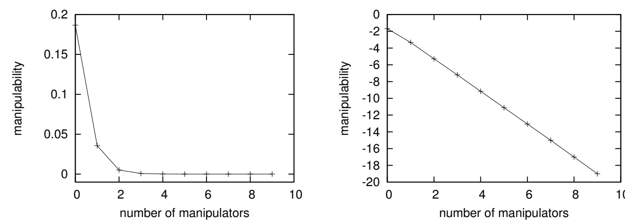

Fig. 4. Manipulability transitions with changing number of manipulators

C. Automatic threshold detection for SR-inverse

SR-inverse is calculated from an original jacobi matrix by using following equation:

$$\mathbf{J}^{\#} = \mathbf{J}^t(k\mathbf{I} + \mathbf{J}\mathbf{J}^t)^{-1}. \tag{8}$$

One of the way to calculate the k is

$$k = \begin{cases} (1 - \frac{w}{w0}) * (1 - \frac{w}{w0}) & w < w_0 \\ 0 & otherwise, \end{cases} \tag{9}$$

where $w = \sqrt{det(\mathbf{J}\mathbf{J}^t)}$ is the measure of manipulability[19] and w_0 is a positive constant. This equation indicates that common pseudo jacobi matrix will be used when the value of manipulability exceeds the predefined threshold.

However, this pose calculation becomes unstable under the condition that manipulators are often added or deleted. To proof it, a simulation was performed. A mobile platform was assumed and manipulators which have 2 rotational joints were mounted on the base as shown in Fig.3. Joint angles of each assistant manipulator are set to $(\theta_1, \theta_2) = (\pi/4, \pi/2)$ (in the case of manipulators settled on right side, joint angles are set to $(-\pi/4, -\pi/2)$). Left graph in Fig.4 shows the relationship between number of manipulators and its manipulability. The value falls steeply in this graph. This fact tells that tracking performance will be degraded when number of manipulators increases because k in eq.(8) always has a value other than zero.

We take an approach to determin the w_0 along number of manipulators. When manipulators which have n joints are added or deleted, the jacobi matrix shown at eq.(5) takes the form of singly-bordered band diagonal matrix which duplicates $\mathbf{J}_{mp}$ (each $\mathbf{J}_m$ consists of 2 rows and n columns) in its diagonal section. Right graph of Fig.4 shows the calculation result of the manipulability by using logarithm natural. The values decrease linearly. From this fact, the eigenvalue of $\mathbf{JJ}^T$ goes along the role as shown below:

$$\begin{aligned}\log_e w &= \log_e(\sigma_1\sigma_2...\sigma_{i-1}\sigma_i) \\ &\approx ai+b,\end{aligned} \tag{10}$$

where σ_i indicates i th singular value of the matrix. e is base of natural logarithm. $a(<0)$ and b are constants. From this equation, we can get

$$w \approx e^{ai+b} = e^{a(i-1)+b}e^a. \tag{11}$$

Because the value of e^a can be estimated from the right graph of Fig.4, it is only necessary to calculate w based on pth power of e^a.

V. Application to navigation

A. An algorithm

In an environment having several obstacles, the velocity $(\dot{x}_h(t), \dot{y}_h(t))$ is input to the end-effector of a leader manipulator at time t. External sensors were assumed to be mounted on the robot, they enables the robot to measure the distance up to the obstacles. Under the above conditions, an algorithm for managing both path following and collision avoidance is proposed as follows:

1) only a leader manipulator initially exists($t = 0$),
2) get the velocity $(\dot{x}_h(t), \dot{y}_h(t))$ as input to the end-effector of the leader manipulator,
3) measure the shape of the environment by using external sensors, and change the configuration of assistant manipulators obeying the following conditions:
   - calculate the distance between the environment and roots of assistant manipulators. The roots on the robot are fixed in advance. If the distance between the environment and one of the roots becomes lower than a predefined threshold, add a new assistant manipulator from the root.
     Joint angles of the added manipulator are calculated to touch the environment where is the nearest spot to the root.
   - if the distance between the end-effector of a certain manipulator and the nearest environment become over the threshold, the manipiulator is deleted.
4) Calculate eq.(6) to renew the whole pose of the robot. If difference between reference position and current position of the end-effector of the leader manipulator becomes lower than a predefined threshold, return 2) with renewing time to $t+1$.

There are several ideas how to generate the input to the leader manipulator. For instance, global path planning is performed in advance, and the input can be calculated as velocity vector to follow the path. On the other hand, one of the methods in dynamic environment is that a path is decided by finding open space. Anyway, because the direction of the platform movement tends to be decided obeying its motion constraint, feasible motion can be generated on the spot.

VI. Verifications through simulation

A. Settings

A planar mobile robot which mounts several 2-DOF virtual manipulators was assumed. The shape of the robot was defined with 400*mm* long and 350*mm* wide rectangle. The link length of a leader manipulator was set 250*mm*. On the other hand, the link length of assistant manipulators were set 200*mm*. Up to 6 assistant manipulators were permitted to exist. The reason why the link length of the leader manipulator is longer than that of assistant manipulator is to keep the condition that the robot separates from the given path when avoiding obstacles.

The root positions of the manipulators were set as following coordinates. Leader manipulator: $(x, y) = (200, 0)$, assistant manipulator: $(150, 120)$, $(150, -120)$, $(0, 120)$, $(0, -120)$, $(-120, 120)$, $(-120, -120)$ in each. These coordinates indicated relative position from the center of the robot. The initial joint angles of the leader manipulator were set as $(\theta_1, \theta_2) = (\pi/4, -\pi/2)$, the robot tried to keep these angles by means of $\mathbf{y}$ at eq.(7).

B. Setting for feasible path following

We concluded how to adjust the variables as follows:

1) **The configuration of virtual manipulators:** In generally, if the length of a manipulator becomes shorter, the motion of the robot will be influenced from the manipulator. In the meantime, the influence of the leader manipulator becomes smaller with increasing assistant manipulators.

Some simulations with considering these things told us the following results: (i) the length of the manipulator should be set as 2 or 3 times longer than the distance between the center of the robot and the root of the manipulator, (ii) assistant manipulators should be settled near to the outer frame of the robot for effective collision avoidance, (iii) 4 to 8 assistant manipulators are preferable because of the trade-off between avoidance ability and calculation cost.

2) **The element of the matrix W:** weight coefficients should be changed whether a robot has motion constraints or not. Moreover, a leader manipulator should have a priority to pull the robot no matter what multiple assistant manipulators exist.

We tried both case of omnidirectional platform and 2 wheeled mobile platform, the proper weights were as follows: 0.1 to a mobile platform, 1 to a leader manipulator and 0.6 to assistant manipulators in case of omnidirectional platform. On the other hand, 1 to a mobile platform, 0.6 to a leader manipulator and 0.4 to assistant manipulators in case of 2 wheeled mobile platform.

3) **Coefficient k of SR-inverse:** We already presented how to decide k in section IV.C. The w was set by using eq.(11) and $w_0 = 0.08$ in our simulation.

4) **Coefficient λ for considering redundant DOFs:** If λ in eq.(6) has big value, the effect of the optimization becomes quite large. In the viewpoint of path following, It is not good that the pose at time t is significantly changed comparing with that of time $t-1$. This value was experimentally decided as 0.01.

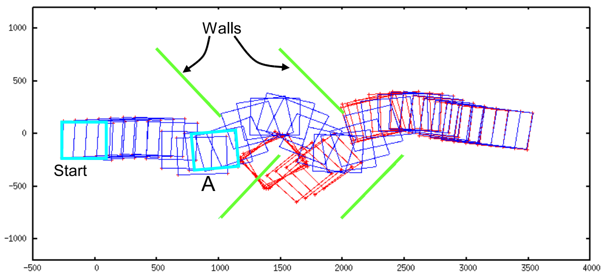

Fig. 5. Experimental results of obstacle avoidance in case of omnidirectional mobile platform

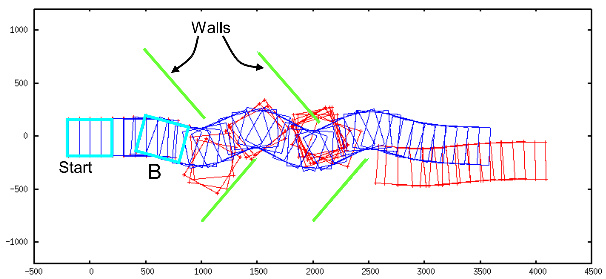

Fig. 6. Experimental results of obstacle avoidance in case of 2 wheeled robot

*5) **The distance threshold up to environment when assistant manipulators are added or deleted:*** This variable should be set for preventing large pose change just when assistant manipulators are added or deleted. Through simulation, it is suitable to set the distance under the role of 0.6 times of manipulator length, because such manipulator pose ensures high manipulability.

C. Navigation in narrow space

Simulations to advance in narrow passage were tried about omnidirectional and 2 wheeled robot. A leader manipulator was guided by straight line from $(x,y) = (200,0)$ to $(x,y) = (4000,0)$. In both case, because of virtual manipulators, the robot could avoid walls existing near its passage. As omnidirectional platform can move any direction, the motion can be planned with swinging its back shown as blue rectangles in figure.5. In the meantime, our method also enabled to archive the path following and collision avoidance in case of nonholonomic platform shown as blue rectangles in figure.6.

Red rectangles shown in figure.5 and figure.6 indicate the pose calculation results without the method described in IV.C. When number of manipulators increased, the pose calculation became unstable. As a result, planned robot poses occurred jumps at point A and B.

Fig.7 shows transition of number of manipulators in case of blue rectangles in figure.6, and Fig.8 shows several examples when assistant manipulators were utilized to avoid collisions. In this simulation, 350 times pose calculation while following start to goal took 0.23 seconds at 2.4 GHz CPU.

Fig.9 shows an example of path following by elongated robot. In this case, number of virtual manipulators were increased to 8 and their root positions were redefined, but other parameters were set as same as that of above simulations. The example shows that our method permits non-circular robots by changing the configuration of virtual manipulators. This is a functional dissimilarity comparing with traditional potential based approach.

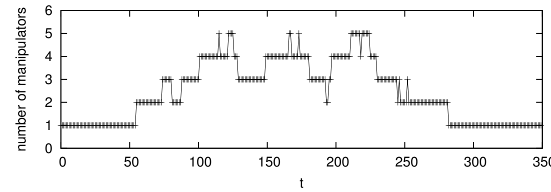

Fig. 7. Number of manipulators

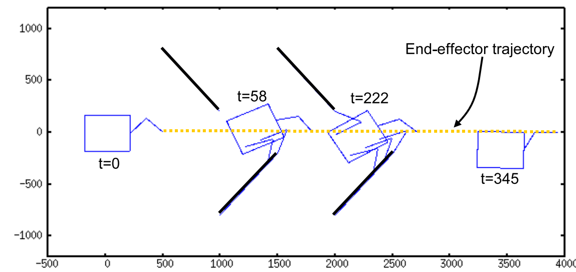

Fig. 8. Examples collision avoidance by using virtual manipulators

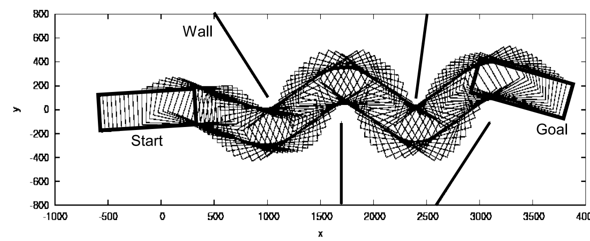

Fig. 9. A tracking result in the case of elongated shape

D. Navigation with combining global path planning

Blue rectangles in figure.10 and figure.11 show the planning results of an imitation environment. In this simulation, the purpose is to create a robot motion which can follow the end-effector path from $(x,y) = (0,0)$ to $(x,y) = (1600,-500)$. Laplacian potential method[4] was used to detect the end-effector path. In spite of the lack of path smoothing, the robot motion was planned with keeping its motion continuity and also achieving collision avoidance. In this case, our pose calculation also worked effectively. Without the method described in IV.C., planned robot poses occurred jumps at point C and D. In the meantime, Fig.12 and Fig.13 shows the comparison with or without assistant manipulators. Red rectangles show the planning results without asssitant manipiulators. These results shows that our method is useful to cope with path following and collision avoidance at once.

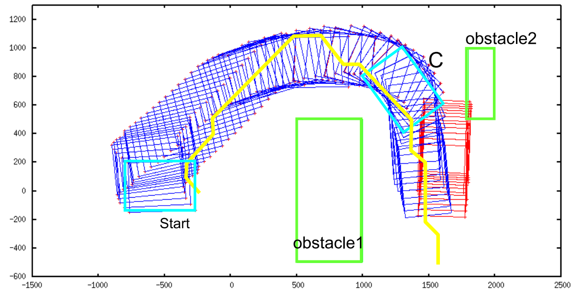

Fig. 10. An experimental result of passing narrow space in the case of omnidirectional robot

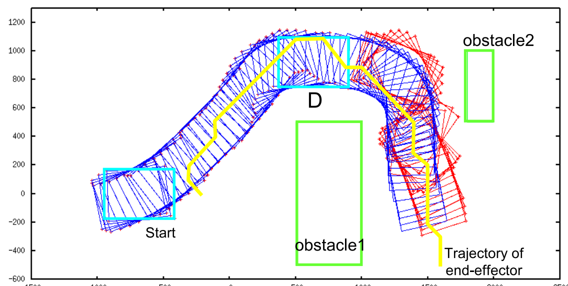

Fig. 11. An experimental result of passing narrow space in the case of 2 wheeled robot

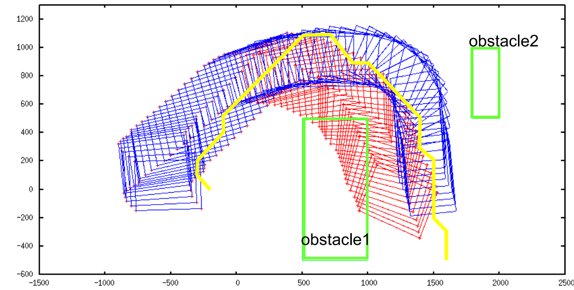

Fig. 12. A tracking result without assistant arms in the case of omnidirectional robot

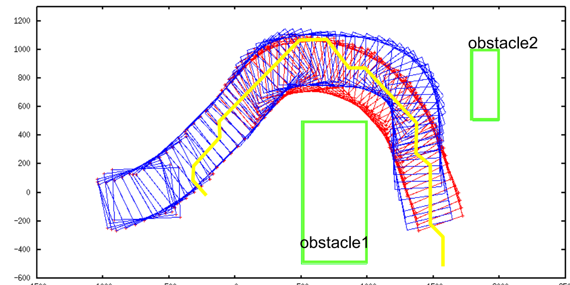

Fig. 13. A tracking result without assistant arms in the case of 2 wheeled robot

VII. Conclusion

We described a path following method for mobile robots. Multiple manipulators are virtually mounted on the robot, both path following and collision avoidance are achieved. We proposed the policy for stable pose calculation and proofed the effectiveness of our method by means of simulation.

Future works, we will implement this method to real robots. Combining global path planning, we will try to move the robot in daily environment. In this paper, although we only showed planar robots which mount 2 joints manipulators, other types of robot should be covered.

Acknowledgement

This research is supported by Special Coordination Funds for Promoting Science and Technology, "IRT Foundation to Support Man and Aging Society".

References

[1] B.Bayle, J.Fourquet and M.Renaud: "Manipulability Ayalysis for Mobile Manipulation," Proc. of IEEE Int. Conf. on Robotics and Automation, pp.1251–1256, 2001.
[2] J. Bruce and M.Veloso: "Real-time randamazed pth planning for robot navigation," in Proc. of Int'l Conf. on Robotics and Autonomous Systems, 2002.
[3] W. Burgard D. Fox and S. Thrun: "Controlling synchro-drive robots with the dynamic window approach to collision avoidance," Proc. of the IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems, 1996.
[4] C. Connolly, J. Burns and R. Weiss: "Path Planning using Laplace's Equation," Proc. of Int. Conf. on Control, Automation, and Systems, pp.2102 – 2106, 1990.
[5] M. Egerstedt and X. Hu: "Coordinated Trajectory Following for Mobile Manipulation," Proc. of IEEE Int'l Conf. on Robotics and Automation, pp.3479–3484, 2000.
[6] D. Ferguson, N. Kalra and A. T. Stentz: "Replanning with RRTs," in Proc. IEEE Int'l Conf. on Robotics and Automation, 2006.
[7] H. Huang, R. Fajen, R. Fink and H. Warren: "Visual navigation and obstacle avoidance using a steering potential function," Robotics and Autonomous Systems 54, pp.288–299, 2006.
[8] L.E. Kavraki, P. Svestka, J. Latombe and M.H. Overmars: "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Trans. on Robotics and Automation, vol.12, no.4, pp.566580, 1996.
[9] O. Khatib: "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Intl. Journal of Robotics Research, Vol.5–1, pp.90–98, 1986.
[10] D.H. Kim and S. Shin: "Local path planning using a new artificial potential function composition and its analytical design guidelines," Advanced Robotics vol. 20, pp.115-135, 2006.
[11] S. Koenig and M. Likhachev: "D* Lite," in Proc. of the National Conf. on Artificial Intelligence, 2002.
[12] J. Kuffner and S. LaValle: "RRT-connect: An efficient approach to single-query path planning," in Proc. IEEE. Int'l Conf. on Robotics and Automation, vol.2, 2001.
[13] T. Li and Y.Shie: "An incremental learning approach to motion planning with roadmap management," Proc. IEEE Int'l Conf. on Robotics and automation, 2002.
[14] K. Nagatani, T. Hirayama, A. Gofuku, Y. Tanaka: "Motion Planning for Mobile Manipulator with Keeping Manipulability,", Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp.1663-1668, 2002.
[15] Y.Nakamura and H. Hanafusa: "Inverse kinematic solutions with singularity robustness for robot manipulator control," Trans. of the ASME. Journal of Dynamic Systems, Measurement and Control., Vol. 108, pp. 163–171, 1986.
[16] G. Oriolo and C. Mongillo: "Motion Planning for Mobile Manipulators along Given End-effector Paths," Proc. of IEEE Int'l. Conf. Robotics and Automation, pp.2166– 2172, 2005.
[17] D. Shin, B. Hamner, S Singh and M Hwangbo: "Motion Planning for a Mobile Manipulator with Imprecise Locomotion," Proc. of Int. Conf. on Control, Automation, and Systems, 2003.
[18] A. Stentz: "The focussed D* algorithm for real-time planning," Proc. of the Int'l Joint Conf. on Artificial Intelligence, pp. 1652–1659, 1995
[19] T. Yoshikawa "Analysis and Control of Robot Manipulators with Redundancy," Preprints of 1st Int. Symposium of Robotics Research, 1983.