

Motion Planning for a Mobile Manipulator with Imprecise Locomotion

Dong Hun Shin[†], Bradley S. Hamner[‡], Sanjiv Singh[‡], and Myung Hwangbo[‡]

[†] Dept. of Mechanical Eng.
University of Seoul
shin@uos.ac.kr

[‡] The Robotics Institute
Carnegie Mellon University
{bhamner, myung}@andrew.cmu.edu,
ssingh@ri.cmu.edu

ABSTRACT

This paper presents a motion planning method for mobile manipulators for which the base locomotion is less precise than the manipulator control. In such a case, it is advisable to move the base to discrete poses from which the manipulator can be deployed to cover a prescribed trajectory. The proposed method finds base poses that not only cover the trajectory but also meet constraints on a measure of manipulability. We propose a variant of the conventional manipulability measure that is suited to the trajectory control of the end effector of the mobile manipulator along an arbitrary curve in three space. Results with implementation on a mobile manipulator are discussed.

I. INTRODUCTION

Mobile robots have long been used in applications such as material transport and exploration where the task of the robot is stated as getting from one point to another. Other common applications involve moving a mobile robot to completely cover an area such as necessary for treating a surface. In this case, the base or a polygon simply related to the base must efficiently cover a specified area. Recent applications involve tasks that are performed by a manipulator mounted on a mobile robot. This is an efficient method of extending the reach of manipulators which are otherwise restricted to fixed locations. Manufacturing of large structures is an example of such an application. Instead of creating large and stiff workcells to perform tasks such as welding, we envision that such tasks can be performed by mobile manipulators.

The main issue in using a mobile manipulator has typically been coordinating locomotion and manipulation. This is particularly because the solution for the control parameters is typically redundant because a mobile manipulator has many more degrees of control than are necessary for a unique solution. Several researchers have addressed this problem. Carriker [1] formulated the coordination of manipulation and locomotion as a nonlinear optimization problem to solve its redundancy, where the cost function was only the Euclidean distance between base positions in application. Similar ways, but with different cost functions

or optimization methods, have been studied in [2][3][4][5]. Especially, in [6] Yamamoto controls the mobile platform so that the manipulator is always positioned at a preferred configuration, which maximizes its manipulability measure [7]. In [8] locomotion is planned from the desired manipulator end effector references by optimizing the LQ of dynamic manipulability [9].

Compensation of the dynamic interaction of manipulator and mobile platform has been studied in general in [10] and in specific areas, such as tip-over stability [11], vehicle suspension [12], and dynamics modeling [13]. Cooperating control of multiple mobile manipulators has been studied in [14][15][16], which have been derived from the force control methodologies.

We propose a motion planning method for mobile manipulators for which the base locomotion is less precise than the manipulator control. This is in fact common because the control of mechanisms with non-holonomic constraints is typically much harder than the control of serial mechanisms like manipulator arms. In such a case, it is advisable to move the base to discrete poses from which the manipulator can be deployed to cover a prescribed trajectory. The proposed method finds base poses that not only cover the trajectory but also meet constraints on a measure of manipulability. We propose a variant of the conventional manipulability measure that is suited to the trajectory control of the end effector of the mobile manipulator along an arbitrary curve in three space. Results with implementation on a mobile manipulator are discussed along with simulation results from an arbitrary path.

II. PATH DIVISION AND CONSTRAINTS

Here we describe a method that takes a curve specified in Cartesian space and find a sequence of poses for the base of the mobile manipulator from which the curve can be fully covered.

A. Proposed method

A robot is considered to consist of a mobile platform and an attached manipulator. The mobile platform has 3-DOF on the planar surface, while the manipulator has n-

DOF with serial links. The task of interest is described by the trajectory of the manipulator's end effector, and is defined by a sequence of desired position-and-orientation vectors of the form $\mathbf{r}(i) = [\mathbf{p}, \mathbf{o}] = [p_x, p_y, p_z, o_x, o_y, o_z]^T$.

Recall our paradigm: the locomotion system moves, achieves a specified pose, and stops. The manipulator moves its end effector to track a part of the path and the mobile platform repositions itself for the manipulator to track the next part of the path.

Then the problem is to determine the set of ordered 3-DOF poses, $\mathbf{x}_b = [x, y, \phi]$, of the mobile platform in the form

$$\mathcal{S} = \{\mathbf{x}_{b1}, \mathbf{x}_{b2}, \mathbf{x}_{b3}, \dots\}, \quad (1)$$

and to divide the task paths into segments corresponding to the poses, where each segment is covered by the limited capability of the manipulator from the pose of the mobile platform. However, since there can be multiple sets of solutions to divide a path, we propose a method to obtain a unique solution by optimizing a performance function based on the kinematic constraints below.

The first constraint to be considered for the path division is reachability. That is, whether the inverse kinematics solution of the manipulator is guaranteed to exist for the entire path segment from the corresponding pose of the mobile platform.

We consider manipulability as a second constraint. The manipulability measure conventionally used is given by

$$w = \sqrt{\det[\mathbf{J}(\theta) \mathbf{J}^T(\theta)]} \quad (2)$$

where $\mathbf{J}(\theta)$ is the manipulator's Jacobian matrix and $\theta \in \mathcal{R}^n$ is the joint vector of the n-DOF manipulator [7]. This formula reduces to $w = |\det \mathbf{J}|$ for non-redundant manipulators. This is a measure of the manipulating ability of the end effector in all directions from the current position. That is, the maximum value of the measure will be at a point in which the end effector can move freely in all directions, and it goes to zero if the end effector can not move in any one direction. Thus, manipulability is often depicted with an ellipsoid [4], as shown in Fig. 1. If a point on the task path is reachable, manipulability will be used to check how good the pose of the mobile platform is for allowing the manipulator to work on the path point. If the manipulability measure were quite small, then the pose would be considered bad, due to the manipulability properties mentioned above.

However, a low manipulability score does not necessarily mean that the path is not trackable, or that it requires infinitely fast joint motion to track the path. For example, the manipulability measure of the two-joint link manipulator shown in Fig. 1 is

$$w = |\det \mathbf{J}| = a_1 a_2 |\sin \theta_2| \quad (3)$$

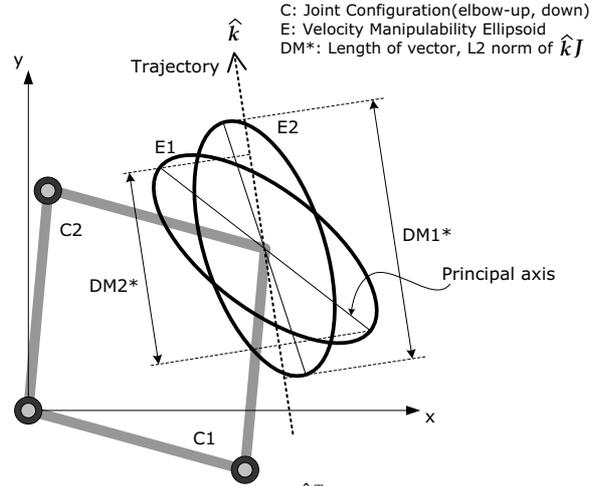


Fig. 1. Directional Manipulability($\hat{\mathbf{k}}^T \mathbf{J}$) on the velocity manipulability ellipse for a 2DOF manipulator

If the first and the second link form a straight line, then the end effector cannot move in an outward direction, and thus the manipulability measure is zero. However, if the path to be tracked is perpendicular to that direction, then even the smallest joint motions are sufficient for tracking it.

Thus, we notice that a good measure for path tracking should consider not only the manipulator's pose but also the path itself. We propose a new manipulability measure based on this idea. Suppose that $\hat{\mathbf{k}} = [\dot{p}/|\dot{p}|, \dot{o}/|\dot{o}|]$ is a directional unit vector of the derivatives of position and orientation along the trajectory of the end effector. The speed along the path is

$$\hat{\mathbf{k}} \cdot \dot{\mathbf{x}} = \hat{\mathbf{k}} \cdot \mathbf{J} \dot{\theta} = \hat{\mathbf{k}}^T \mathbf{J} \dot{\theta} \quad (4)$$

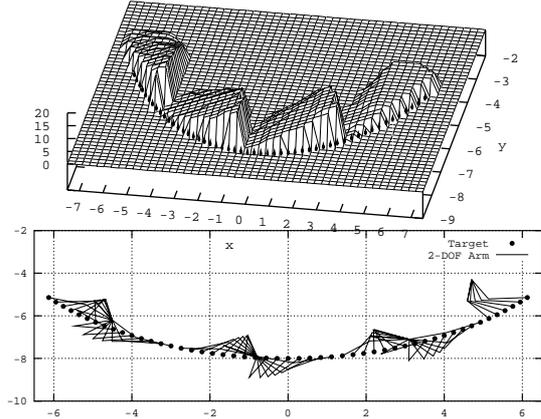
Then a path in which all the components of $\hat{\mathbf{k}}^T \mathbf{J}$ are not too small, but one is large enough, will result in a high degree of manipulability.

Note that $\hat{\mathbf{k}}^T \mathbf{J}$ can also be used for measuring the induced torque of the manipulator. Let $\mathbf{f} \in \mathcal{R}^m$ denote the force (and the moment) applied to an object by the end effector, and let $\boldsymbol{\tau} \in \mathcal{R}^n$ denote the necessary joint driving force (and torque). Then, we have

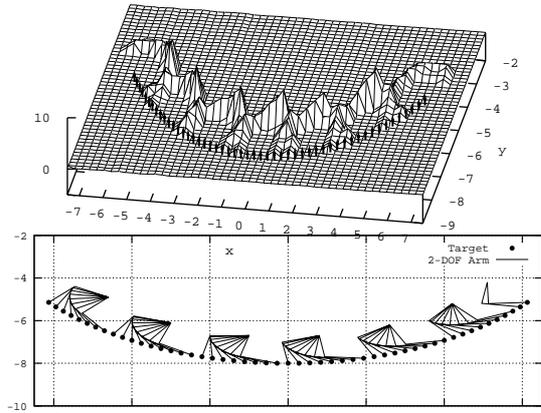
$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{f} \quad (5)$$

Since $\hat{\mathbf{k}}$ is the directional vector of the motion along the path, then the applied force or the friction force to the end effector is typically in the direction of $\hat{\mathbf{k}}$. Thus, the required joint torque (and force) is usually proportional to $\mathbf{J}^T \hat{\mathbf{k}}$ and will be very large if any component of $\mathbf{J}^T \hat{\mathbf{k}}$ is very large. We can also use $\hat{\mathbf{k}}^T \mathbf{J}$ to check if too much motion is required at one joint (all elements of $\hat{\mathbf{k}}^T \mathbf{J}$ are too small) or if too much torque is required at a joint (an element of $\hat{\mathbf{k}}^T \mathbf{J}$ is too large).

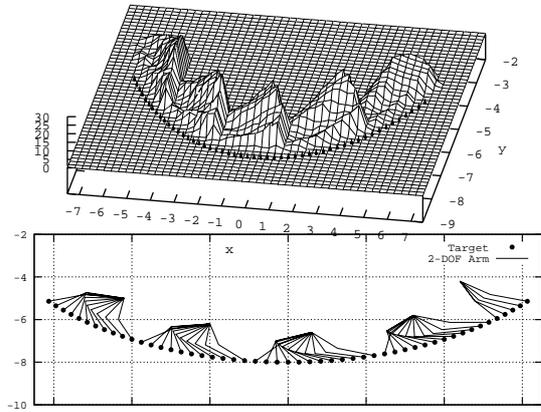
Hence we want to ensure that the maximum absolute value of the components of $\hat{\mathbf{k}}^T \mathbf{J}$ is neither too small nor



(a) Reachability score only



(b) Conventional manipulability: $\sqrt{\det[\mathbf{J}\mathbf{J}^T]} > 0.90$



(c) Directional manipulability: $\sqrt{\hat{\mathbf{k}}^T \mathbf{J}\mathbf{J}^T \hat{\mathbf{k}}}$, $0.3 < |j_1| < 0.7$, $0.5 < |j_2| < 1.5$

Fig. 2. On top, a distribution of scores where each grid cell represents a mobile base position. On bottom, the resulting path division including base positions with the arm configuration at each target point.

too large. Thus, we define the performance function to divide the path as follows. Suppose $\hat{\mathbf{k}}^T \mathbf{J} = [j_1, j_2, j_3, \dots, j_n]$ and j_x is the maximum absolute value among $\hat{\mathbf{k}}^T \mathbf{J}$ components as ($j_x \geq |j_i|$ for $i = 1, 2, \dots, n$), and we want to keep $b_{\min} \leq j_x \leq b_{\max}$, where b_{\min} and b_{\max} are the lower and upper bounds, respectively, of our arbitrarily chosen desired range. Then we set the performance function as

$$f(\theta, \mathbf{x}, \mathbf{r}) \begin{cases} = \frac{|j_x - b_{\text{mean}}|}{b_{\text{max}} - b_{\text{min}}} \times \text{Reachability} & \text{if } j_x \in [b_{\min}, b_{\max}] \\ = 0, & \text{if } j_x \notin [b_{\min}, b_{\max}] \end{cases} \quad (6)$$

where

$$b_{\text{mean}} = (b_{\text{max}} + b_{\text{min}})/2,$$

$$\text{Reachability} \begin{cases} = 1, & \text{if } \mathbf{r} \text{ is reachable from } \mathbf{x} \\ = 0, & \text{if } \mathbf{r} \text{ is not reachable} \end{cases}$$

To make computation easier, we discretize the path. Thus, the problem is to find the locomotive poses and divide the path into segments by maximizing $\sum_{r(i)} f(\theta, \mathbf{x}, \mathbf{r})$ over the path \mathbf{r} .

Note that for all path points $\mathbf{r}(i)$, there exist joint angles θ and a locomotive pose \mathbf{x} that reaches $\mathbf{r}(i)$. Furthermore, there is a distinct maximum value, f_{max} that the performance function can reach. Then

$$\max_{r(i)} \sum f(\theta, \mathbf{x}, \mathbf{r}) = \sum_{r(i)} f_{\text{max}} \quad (7)$$

That is, the maximum value of $f(\theta, \mathbf{x}, \mathbf{r})$ applied to the entire path is the sum of f_{max} when applied to each point on the path. This results in infinitely many poses for the mobile platform. Thus, the mobile platform simply moves while the manipulator works.

However, our paradigm does not allow for such an arrangement. So we must derive a different solution, which will be sub-optimal, through a different method. Our method is to consider one mobile platform position at a time, and maximize $\sum_{r(i)} f(\theta, \mathbf{x}, \mathbf{r})$ over the path. Then, we select the pose of the mobile platform with the maximum value, remove the segment of the path that it covers, and repeat the process with the remaining path.

B. Example: Two revolute joint manipulator

In this section, we show the result of the path division by the proposed method and compare it with the other methods. Consider the two-link manipulator, shown in Fig. 1, on the mobile platform to track along a circular path. The arm segments are denoted a1 and a2, respectively.

Fig. 2-(a) shows the result when the path is divided using only with the reachability condition. Since we are using the sub-optimal strategy that maximizes the sum of the performance function over the path from one locomotive pose, then each locomotive position tries to

cover as much of the path as possible. This results in few positionings which are close to the path, in order to get the most of the manipulator's circular working range. However, in our real life case this results in poor control for the points that the manipulator must approach singularity in order to reach. So we add manipulability to the path division algorithm. Fig. 2-(b) shows the result when the path is divided with reachability and the conventional manipulability measure. Since the manipulator Jacobian matrix is given by

$$\mathbf{J} = \begin{bmatrix} -a_1 \sin \theta_1 - a_2 \sin \theta_{12} & -a_2 \sin \theta_{12} \\ a_1 \cos \theta_1 + a_2 \cos \theta_{12} & a_2 \cos \theta_{12} \end{bmatrix} \quad (8)$$

and the conventional manipulability measure is $w = |\det \mathbf{J}| = a_1 a_2 |\sin \theta_2|$, then the measure reaches its maximum at $\theta_2 = 90^\circ$. The manipulator is never allowed to approach singularity. This restricts the working range of the manipulator, and thus requires more locomotive positionings, which are farther away from the path. As we discussed, though, we can allow the manipulator to approach singularity, to a degree, in situations in which tracking the path requires moving away from the singularity. So we replace conventional manipulability with our new directional manipulability measure and run the algorithm again. The results are shown in Fig. 2-(c). The manipulator is allowed to extend outward, though not as far as in Fig. 2-(a), since the direction of the path brings it away from singularity. Since we use more of the manipulator's working range than in the conventional manipulability algorithm, we require fewer locomotive positionings, while still retaining better control than had we used reachability alone.

Fig. 3-(a) compares the conventional manipulability measure of each of the three criteria along the target path. As expected, the use of only the reachability condition produces configurations where the manipulability measure drops to very low levels, while using conventional manipulability shows the highest manipulability values. However, as shown in Fig. 3-(b), use of directional manipulability as the criteria keeps the important measure of being able to control the end effector in the direction of the path at high levels.

III. IMPLEMENTATION

A. Experimental Robot System

In our implementation, we consider the problem of mock-welding a piece of machinery with our mobile manipulator. In this case that piece of machinery is a four-ton boom provided by CATERPILLAR, INC. The mobile robot we use is BULLWINKLE, an ATRV-2 model from REAL WORLD INTERFACE, now a division of iROBOT [17]. It is a four-wheel drive robot with differential wheels. The manipulator attached to it, called WHIPLASH, was designed under a contract by NASA's JOHNSON SPACE

CENTER and built by METRICA, INC. It has five degrees of freedom, with no wrist yawing. Therefore, the end effector's orientation is determined by the angle of the shoulder joint.

An important aspect of our problem is the ability to determine the mobile robot's position very accurately and precisely. We use the CONSTELLATION 3DI system from ARCSECOND, INC. in order to do this [18]. The system uses four laser transmitters placed around the work area and a sensor to detect the signals. Each transmitter rotates a planar beam, which has a signature unique to the transmitter, at a fixed rate known to the sensor. Simply explained, by timing when the beam is detected, the sensor can calculate its angle to each transmitter. These angles are sent to software on a laptop computer, which triangulates the sensor's position. In order to triangulate, however, the software requires at least three angles, which means the sensor must have an unobstructed line of sight to at least three transmitters. When it has the necessary data, the software can accurately and precisely calculate the position of the sensor to within a millimeter.

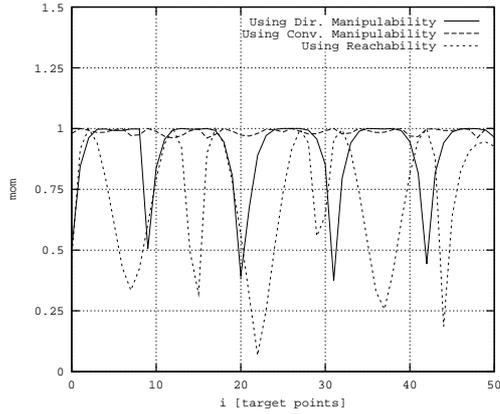
We arranged the ARCSECOND transmitters to form the corners of an eleven by four meter rectangle, with the boom placed along one of the long sides. This setup assured that a point on the side of the boom would have an unobstructed line of sight to each of the four transmitters.

We mounted two ARCSECOND sensors on top of the mobile base, one in front and the other in back. The robot's position is its center of rotation. We know where the receivers are on the robot, so by reading in their positions we may calculate the position of the center of rotation. We may also calculate the robot's heading using this setup. During execution, a program broadcasts the position and heading to the rest of our system, as described below.

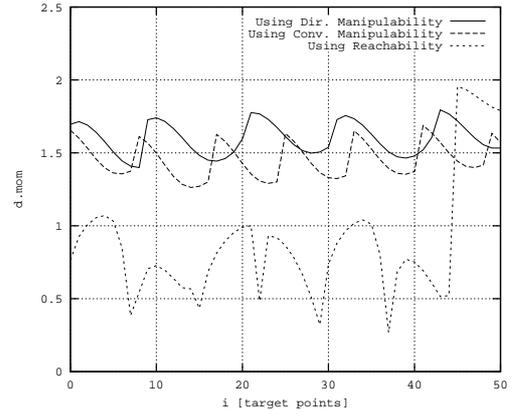
We arbitrarily constructed a zig-zag pattern on the side of the boom as the path to be tracked, as shown in Fig. 4. To derive the path, we used an ARCSECOND sensor to measure points in short intervals along the zig-zag. We fed these points to a program that fills in the path by interpolating a straight line between each successive pair of points. The complete path is then saved in a file as a series of positions in x-, y-, and z-coordinates.

Then we ran the path division program. This program inserts break points into the path's file to indicate where Whiplash should stop working and have BULLWINKLE reposition itself. The program also creates a file containing the calculated robot positions corresponding to each segment of the path.

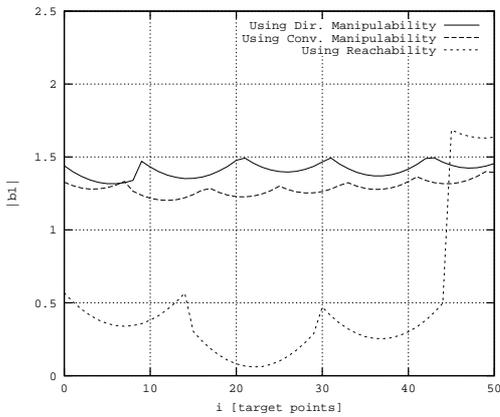
For robot control, we designed a system in which each aspect of the control was modular. These modules transmit commands and data via IPC, a multi-platform publish/subscribe protocol [19]. We split the system into three levels, with separate programs for low-level control (*move wheels at a certain velocity*) and high-level



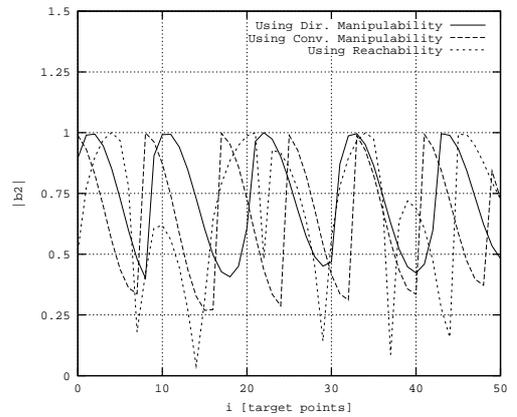
(a) Measure of manipulability along path



(b) Directional manipulability ($\sqrt{\hat{\mathbf{k}}^T \mathbf{J} \mathbf{J}^T \hat{\mathbf{k}}}$) along path



(c) Directional manipulability for joint 1 ($|j_1|$)



(d) Directional manipulability for joint 2 ($|j_2|$)

Fig. 3. Comparison of manipulability measures along the target path for three criteria, using the 2-DOF arm

control(*go forward a certain distance*) of the mobile robot, low-level control(*move joints to given angles*) and high-level control(*touch a given point*) of the manipulator, and finally a *driver* program which accepts high-level commands and issues the necessary low-level commands to execute them. This model is analogous to a cab driver system; the *master* gives the commands like *turn left here* and *go three blocks*, and the driver responds by pressing the accelerator and turning the steering wheel as required.

However, in our system we use two master programs, which need to share control and make sure only one has control at a time. We designed *control* messages to accomplish this. A master only has control when it has received a control message, and it has exclusive control until it publishes a control message for the other master to receive. For example, during execution, the mobile robot's master has control first. It reads in the first robot position, and sends the driver the appropriate move forward and turn commands. The driver steers BULLWINKLE to the

desired point using the position and heading data from the ARCSECOND system, and the mobile robot's master sends a control message to the manipulator's master. The manipulator's master then reads in the path segment and tells the driver to direct WHIPLASH to the given points, using BULLWINKLE's actual position rather than the prescribed position from the path division program. When the segment has been tracked, control is sent back to the mobile master, and the process repeats until the entire path has been tracked.

To measure accuracy, we marked the path off with tape that was two centimeters wide. The entire path was about three meters long, and the path division program using directional manipulability resulted in three locomotive positionings. We ran the system, and Whiplash was able to track the entire path on the tape (no more than 1 centimeter error), despite the impreciseness of BULLWINKLE's positionings.

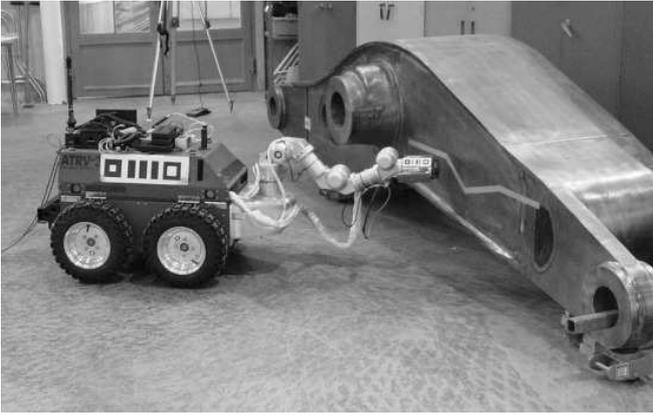


Fig. 4. Experimental robot system and the boom

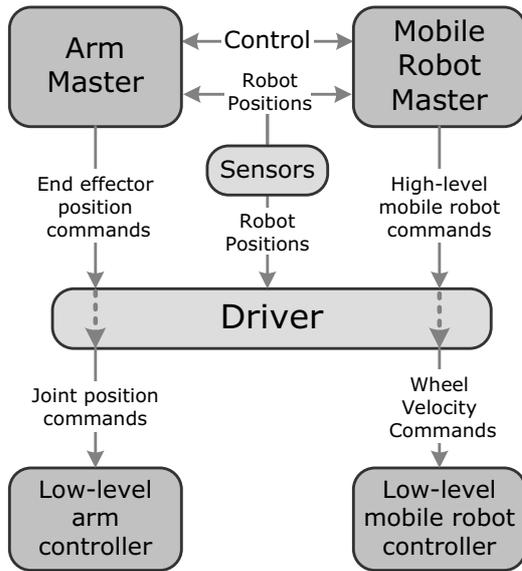


Fig. 5. Control modules of the experimental robot

B. Path division on the boom

Fig. 6 shows a sample path on the side of the boom. The three figures depict the results of the path division program, using three different constraints we mentioned, for the 5-DOF manipulator Whiplash. The curved line is the path itself, and the zig-zag lines represent the configuration of the manipulator touching the point from the mobile platform's position. The path shown in Fig. 6-(a) was divided using only the reachability condition. Notice that it requires only three mobile platform positionings, and each positioning covers a wide area. In reality we get poor manipulator control at the edges of these wide working ranges.

Fig. 6-(b) is the result of dividing the path using conventional manipulability. Notice the much narrower working ranges. While this does provide better control, it also

results in requiring more positionings to cover the entire path. Locomotion with the imprecise mobile platform can take time, so we would prefer to reduce the number of times we have to move. Fig. 6-(c) shows the divided path using directional manipulability. Here the working ranges are not so wide. Yet we have reduced the number of positionings to five, and have thus reached the kind of balance between control and minimizing of movement that we had hoped for.

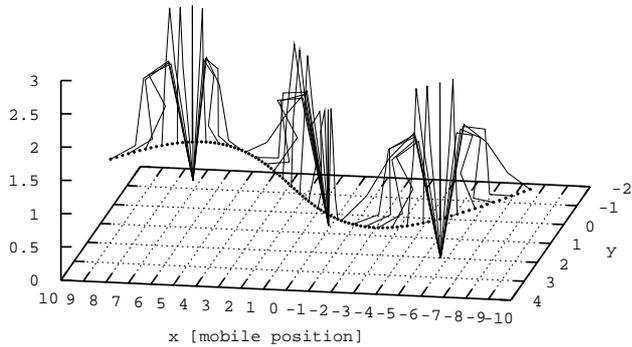
IV. CONCLUSION

In this work we proposed a motion planning method for mobile manipulators that have an imprecise mobile platform, when compared to their attached manipulators. It is based on the paradigm that the mobile platform moves, sets a pose and stops, at which time the manipulator moves its end effector to track a part of the path, followed by the mobile platform repositioning itself for the manipulator to track the next part of the path. This differs from previous research, which treats the locomotion and manipulation equally and actuates both of the systems simultaneously in order to utilize the full redundancy of the robot system. Thus, our method is to consider one mobile platform position at a time, and maximize the performance function over the path. Then we select the pose of the mobile platform with the maximum value, remove the segment of the path that it covers, and repeat the process with the remaining path.

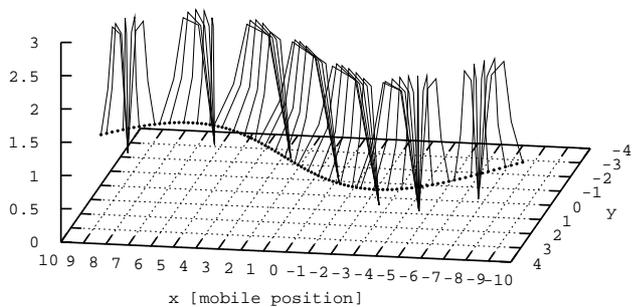
The performance function that we have used is determined by reachability and manipulability conditions. Especially, we proposed to use the newly presented *directional manipulability* condition, which not only depends on the joint angles of the manipulator, but also the path to be tracked, rather than the conventional manipulability measure. The effectiveness of the proposed method has been shown from the simple case of a two-link robot along a straight-line path and the real mock-welding case of the commercial robot system, consisting of BULLWINKLE and WHIPLASH, along the path on the excavator's boom.

V. REFERENCES

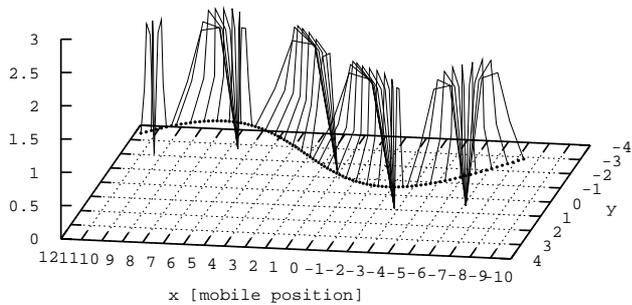
- [1] W. Carriker, P. Khosla, and B. Krogh, "Path planning for mobile manipulators for multiple task execution", *IEEE Trans. Robotics and Automation*, Vol 7, No. 3, pp.403-408, 1991.
- [2] H. Seraji, "An on-line approach to coordinated mobility and manipulation", *Proc. IEEE Int. Conf. On Robotics and Automation*, pp.28-35, 1993.
- [3] H. Seraji, "Configuration control of rover-mounted manipulators", *Proc. IEEE Int. Conf. On Robotics and Automation*, pp.2261-2266, 1995.



(a) Using reachability only



(b) Using conventional manipulability: $m \Rightarrow 6.5$



(c) Using directional manipulability

Fig. 6. Division of a path on the side of the boom with a 5-DOF manipulator like WHIPLASH and the resulting arm configurations

[4] F. Pin, K. Morgansen, F. Tulloch, C. Hacker, and K. Gower, "Motion planning for mobile manipulator with a non-holonomic constraints using the FSP", *J. of Robotic Systems*, Vol. 13, No. 11, pp723-736, 1996.

[5] M. Chen and A. Alzala, "A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration", *J. of Robotic Systems*, Vol. 14, No. 7, pp.529-544, 1997.

[6] Y. Yamamoto and X. Yun, "Coordinating locomotion and manipulation of a mobile manipulator", *IEEE Trans. Automatic Control*, Vol 39, No. 6, pp.1326-1332, 1994.

[7] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. of Robotics Res.* Vol. 4, No. 1, pp.3-9, 1985.

[8] W. Miksch, D. Schroeder, "Performance-functionall based controller design for a mobile manipulator", *Proc. IEEE Int. Conf. On Robotics and Automation*, pp227-232, 1992.

[9] T. Yoshikawa, "Dynamic manipulability of robot manipulators", *J. of Robotic Systems*, Vol. 2, No. 2, pp.113-124, 1985.

[10] Y. Yamamoto and X. Yun, "Effect of the dynamic interaction on coordinated control of mobile manipulators", *IEEE Trans. Robotics and Automation*, Vol 2, No. 5, pp.816-824, 1996.

[11] Q. Huang and S. Sugano, "Motion planning of stabilization and cooperation of a mobile manipulator", *Proc. IROS*, pp.568-575, 1996.

[12] N. Hootsmans and S. Dubowsky, "Large motion control of mobile manipulators including vehicle suspension characteristics", *Proc. IEEE Int. Conf. On Robotics and Automation*, pp.2336-2341, 1991.

[13] Y. Yamamoto and X. Yun, "A modular approach to dynamic modeling of a class of mobile manipulators", *Int. J. of robotics and Automation*, Vol 12, No. 2, pp.41-48, 1997.

[14] H. Osumi, M. Terasawa, and H. Jojiri, "Cooperative control of multiple mobile manipulators on uneven ground", *Proc. IEEE Int. Conf. On Robotics and Automation*, pp.3198-3203, 1998.

[15] T. Sugar and V. Kumar, "Decentralized control of cooperating mobile manipulators", *Proc. IEEE Int. Conf. On Robotics and Automation*, pp.2916-2921, 1998.

[16] O. Khatib, k. Yokoi, K. Chang, D. Ruspini, R. Homberg, and A. Casal, "Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation", *Proc. IROS*, pp.546-553, 1996.

[17] <http://www.irobot.com/rwi/p02.asp>

[18] <http://www.constellation3di.com/>

[19] <http://www-2.cs.cmu.edu/afs/cs/project/TCA/www/ipc/ipc.html>