PATH FOLLOWING CONTROL FOR A MOBILE ROBOT PUSHING A BALL

Xiang Li, Andreas Zell

Wilhelm-Schickard-Institute, Department of Computer Architecture,
University of Tübingen, Sand 1, 72076 Tübingen, Germany
{lix, zell}@informatik.uni-tuebingen.de

Abstract: This paper focuses on the control problem of a mobile robot pushing a ball. In order to drive a rolling ball along a given path, the robot should provide the ball with appropriate force by consecutive pushing operations. A new control algorithm combining a linear feedback control with a normal proportional feedback control is proposed, which steers a reference point to follow the given path and the ball to move around this point simultaneously. In the end, the simulation and real-world experiments address the performance and robustness of this control algorithm.

Keywords: Robot control, follow-up control, linear output feedback, proportional controllers, feedback control methods.

1. INTRODUCTION

Moving an object from one position to another is one of the basic tasks of a robot. Normally equipped with some flexible manipulators, the robot could grip the object, pick it up and place it at the ideal position. But when the object is too large, too heavy or complex to be gripped, as one kind of nonprehensile manipulation, pushing could play a great roll in these complex tasks. Here we studied this pushing task, that a mobile robot pushes a rolling ball and the ball can follow a given path without sliding away from the robot.

Mason (1986) first presented research results on pushing a solid object by a manipulator. He analyzed the mechanics of quasi-static pushing operations, and gave algorithms to determine the rotation direction (clockwise, counter clockwise) of a pushed object, when the pressure distribution is unknown. Peshkin, *et al*. (1988a, b) extended this pushing seminal by attempting to solve the motion of the pushed object completely. With these research results, the pushing operation has been utilized in many applications.

Akella and Mason (1992) described a planner which is guaranteed to construct a sequence of pushing actions to move any polygonal object from any initial configuration to any final configuration. Lynch (1996) studied the problem of transferring a part from one state to another using nonprehensile manipulation, which includes the quasistatic nonprehensile manipulation and the dynamic nonprehensile manipulation. Agarwal, *et al*. (1997) considered the path-planning problem for a robot pushing a unit disk with point contact in an obstacleless environment. Besides these open-loop designs of pushing operation series, many researchers have studied the feedback control of the object pushed by robots. Takagi and Okawa (1991) took the rule-based control scheme to control a mobile robot pushing a box, based on their analysis of the moving equations of the robot and box. Okawa and Yokoyama (1992) resolved the same problem with the goal seeking strategy for robot's motor control. Lynch, *et al*. (1992) developed a control system to translate and orient objects using tactile feedback with the derived motion equations of the pushed objects.

In this paper we also pursue the motion control problem of the pushed object. We control a mobile robot to push a rolling ball, such that the ball follows a given path with high velocity and little derivation. Meanwhile, the ball should not slide away from the robot. Actually the push process consists of several consecutive impacts. Between any two adjacent impacts, only the friction between the ball and the support plane acts on the ball, which means the ball is not controllable for the robot and possibility moves away from the given path. To resolve this problem, we induce a reference point as the path following control object. Under the feedback control of robot, the reference point follows the given path, and simultaneously the ball moves around this point.

2. PROBLEM FORMULATION

The mobile robot used for this task is an omnidirectional robot. An omnidirectional robot has three degrees of freedom on the plane, two orthogonal translations and one rotation. This characteristics allows to control the translation and rotation of the omnidirectional robot independently and simultaneously. Fig. 1 shows the kinematics diagram of the omnidirectional robot.

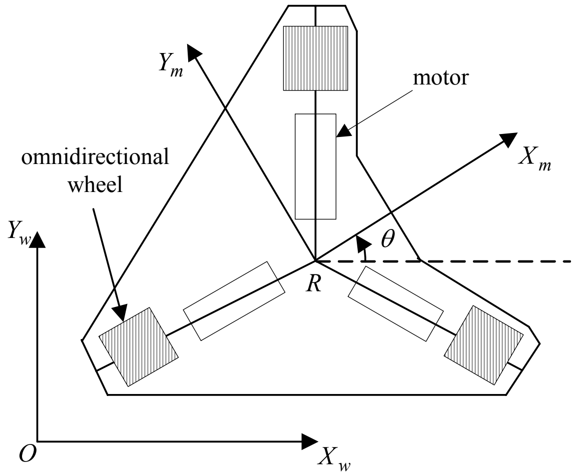

Fig. 1. Kinematics diagram of omnidirectional robot

where $[X_w, Y_w]$ is the world frame. $[X_m, Y_m]$ is the robot frame, which locates at the robot's mass centre. θ is the rotational angle between the world frame and the robot frame, and it denotes the robot heading direction. In the robot frame, the kinematics of the omnidirectional robot is given by the following equations

$$\dot{x}_r^m = v\cos\theta \tag{1}$$

$$\dot{y}_r^m = v\sin\theta \tag{2}$$

$$\dot{\theta} = \omega \tag{3}$$

where $\dot{x}_r^m, \dot{y}_r^m$ are the robot translation velocities with respect to the robot frame. v and ω are respectively the robot's translation velocity and rotation velocity. Transforming the coordinate from the robot frame to the world frame, the robot's absolute velocities are given by

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_r^m \\ \dot{y}_r^m \\ \omega \end{bmatrix} \tag{4}$$

where $\dot{x}_r, \dot{y}_r$ are the robot's translation velocities with respect to the world frame.

As the pushed object, the rolling ball moves like the unicycle mobile robot (de Wit *et al*. 1996). The kinematics of the ball can be described by the following equations

$$\dot{\theta}_b = \omega_b \tag{5}$$

$$\dot{x}_b = v_b\cos\theta_b \tag{6}$$

$$\dot{y}_b = v_b\sin\theta_b \tag{7}$$

where $\dot{x}_b, \dot{y}_b$ are the ball's absolute translation velocities. $\theta_b, \dot{\theta}_b$ are the ball's translation direction and rotation velocity, respectively. Equations (5)-(7) denote that two variables v_b and ω_b are enough to control the ball's movement completely.

2.1 Path Following

The path following problem is illustrated in Fig. 2. *P* denotes the given path. Point *B* is the centre of the ball, and point *Q* is the orthogonal project of *B* on the path *P*. x_t and x_n are the tangent and normal unit vectors at *Q*, respectively. l is the signed distance between the ball's centre and the path *P*. s is the signed distance along the path *P* between any point on path *P* to point *Q*. θ_r is the angle between axis x_t and axis X_w. θ_b denotes the ball's moving direction. The orientation error is defined as $\tilde{\theta}_b = \theta_r - \theta_b$.

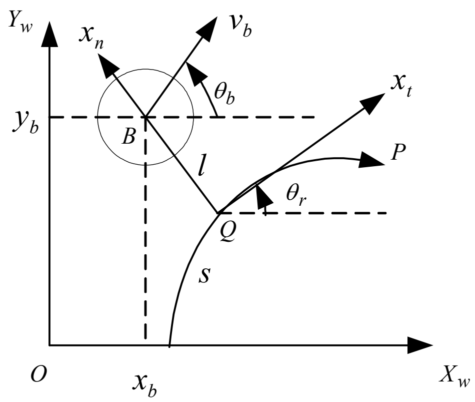

Fig. 2. Illustration of path following problem

With this given path, the path following problem consists of finding feedback control values of v_b and ω_b such that the deviation distance l and orientation error $\hat{\theta}_b$ tend to zero.

Using the previous definitions, the path following problem can be parameterized as

$$\dot{s} = v_b\cos\tilde{\theta}_b \frac{1}{1-cl} \tag{8}$$

$$\dot{l} = v_b\sin\tilde{\theta}_b \tag{9}$$

$$\dot{\tilde{\theta}}_b = \omega_b - v_b\cos\tilde{\theta}_b \frac{c}{1-cl} \tag{10}$$

where c is the path curvature at point *Q*.

2.2 Ball Holding

Holding ball means that the robot should guarantee that its front consecutively contacts the ball during the whole pushing process, even if the given path has a strong turning angle. As shown in fig. 1, our mobile robot has a slightly convex shaped front, which provides for some lateral stability for the pushing operation, i.e. a contact that occurs on the left side of the pusher should result in a restoring impulse that moves the object to the right side. This structure helps the robot not only to hold the ball with some appropriate force, but also to provide a small centripetal force for the ball following the given path.

3. PUSHING CONTROL

Fig.3 shows the ideal situation in the pushing process. When the robot pushes the ball, the ball follows the given path with the required velocity, and is always located at the front of the robot. We divide our task into two parts: control the reference point E following the given path; control the ball around the reference point *E* during the whole pushing process.

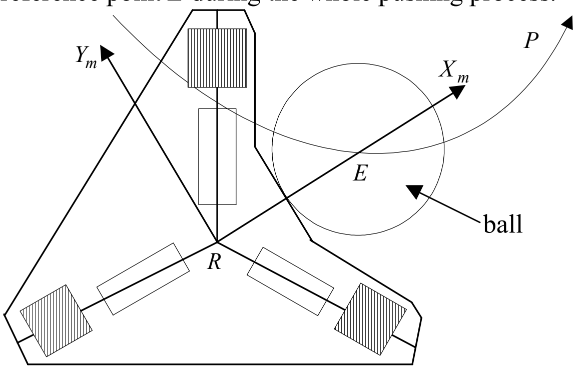

Fig. 3. Ideal positions of robot, ball and given path

3.1 Linear Feedback Control

Given the ideal path *P* and the ball's required translation velocity v_b, the path following problem could be resolved by only finding a feedback control law ω_b such that the deviation distance and orientation error tend to zero.

By inducing a control variable u

$$u = \omega_b - v_b \cos\tilde{\theta}_b \frac{c}{1-cl} \tag{11}$$

equation (10) becomes

$$\dot{\tilde{\theta}}_b = u \tag{12}$$

In the neighbourhood of the origin $(l = 0, \tilde{\theta}_b = 0)$, the linearization of (9) gives

$$\dot{l} = v_b \tilde{\theta}_b \tag{13}$$

This system is controllable and stabilizable when using a linear state feedback controller of the form

$$u = -k_1 v_b l - k_2 |v_b| \tilde{\theta}_b \tag{14}$$

where $k_1 > 0,\ k_2 > 0$, as discussed in (de Wit *et al.* 1996). For a constant v_b, this controller reverts to a classical linear time-invariant state feedback controller. Then the closed-loop equation for the output l is

$$\ddot{l} + k_2 |v_b| \dot{l} + k_1 v_b^2 l = 0 \tag{15}$$

This is a typical second-order system, whose characteristics is directly determined by the natural frequency and damping coefficient. After we select a big enough natural frequency to balance the steady error and response velocity, and the optimal damping coefficient $\sqrt{2}/2$, we obtain the controller parameters k_1, k_2 by

$$k_2 \cdot |v_b| = a^2 \tag{16}$$

$$k_1 \cdot v_b^2 = 2\xi a \tag{17}$$

where a is the natural frequency and ξ is the damping coefficient.

3.2 Robot Motion Control

According to the kinematics model (4) of the omnidirectional robot, three variables $(\dot{x}_r^m, \dot{y}_r^m, \omega)$ can be used to control the robot movement completely. The reference point E is the ideal position of the ball's centre, which is the point $(d, 0)$ in the robot frame and can be considered as a fix point of the robot itself. d is the distance between the centre of robot and the point *E*. The absolute positions and velocities of point *E* are given by

$$x_e = x_r + d\cos\theta \tag{18}$$

$$y_e = y_r + d\sin\theta \tag{19}$$

$$\dot{x}_e = \dot{x}_r - d\omega\sin\theta \tag{20}$$

$$\dot{y}_e = \dot{y}_r + d\omega\cos\theta \tag{21}$$

where x_e, y_e and $\dot{x}_e$, $\dot{y}_e$ are the positions and the translation velocities of point *E* with respect to the world frame. For the point *E* following the given path, we need to provide the point *E* with the rotation velocity ω_b calculated from the linear feedback control algorithm and the given required translation velocity v_b by controlling the robot. Considering the characteristics of our omnidirectional robot, which can translate and rotate simultaneously, control the robot to rotate around the point *E* besides its translation movement, as given by the following equations

$$\dot{x}_m^r = v_b \tag{22}$$

$$\dot{y}_m^r = -d \cdot \omega_b \tag{23}$$

$$\dot{\theta} = \omega_b \tag{24}$$

where v_b is the ball's required translation velocity. Substituting these equations into equations (4) and (20)-(21), we obtain the absolute velocities of point *E*

$$\dot{x}_e = v_b \cos\theta \tag{25}$$

$$\dot{y}_e = v_b \sin\theta \tag{26}$$

Equations (25) and (26) denote that *E* has the ideal velocity v_b. The absolute moving direction θ_e of point *E* is given by

$$\tan\theta_e = \frac{\dot{y}_e}{\dot{x}_e} = \tan\theta \tag{27}$$

Apparently point *E* moves along the robot's heading direction. So the control actions (12) – (24) can also provide the point *E* with rotation velocity ω_b.

Unfortunately this moving direction θ_e may result in the robot losing contact of the ball when it turns, which is highly non-desirable.

Until now, only two variables ω_b and v_b are used to control the robot, which means there is one degree of freedom left, which could be used to control the robot not losing the ball during the path following process. We add another variable $\Delta\omega$ into the robot control actions

$$\dot{x}_m^r = \sqrt{v_b^2 - d^2 \cdot \Delta\omega^2} \tag{28}$$

$$\dot{y}_m^r = -d \cdot (\omega_r + \Delta\omega) \tag{29}$$

$$\dot{\theta} = \omega_r \tag{30}$$

Similarly we calculate the absolute translation velocity of point E by substituting (28) – (30) into (4) and (20)-(21)

$$\dot{x}_e = \dot{x}_m^r \cdot \cos\theta_e + d \cdot \Delta\omega \cdot \sin\theta_e \tag{31}$$

$$\dot{y}_e = \dot{x}_m^r \cdot \sin\theta_e - d \cdot \Delta\omega \cdot \cos\theta_e \tag{32}$$

$$v_e = \sqrt{\dot{x}_e^2 + \dot{y}_e^2} = v_b \tag{33}$$

The translation velocity v_e is equal to the ideal one v_b, and v_e is not relative to the control value $\Delta\omega$ and ω_r. The moving direction and rotation velocity of point E can be obtained from the following equations

$$\tan(\theta_e - \theta) = \frac{-d \cdot \Delta\omega}{\dot{x}_m^r} \tag{34}$$

$$\dot{\theta}_e - \dot{\theta} = -\frac{d \cdot \dot{x}_m^r \cdot \Delta\dot{\omega}}{v_b^2} - \frac{d^3 \cdot \Delta\dot{\omega} \cdot \Delta\ddot{\omega}}{v_b^2 \cdot \dot{x}_m^r} \tag{35}$$

In order to satisfy the requirement of the rotation velocity of E, the rotation velocity of robot ω_r should be

$$\omega_r = \omega_b + \frac{d \cdot \dot{x}_m^r \cdot \Delta\dot{\omega}}{v_b^2} + \frac{d^3 \cdot \Delta\dot{\omega} \cdot \Delta\ddot{\omega}}{v_b^2 \cdot \dot{x}_m^r} \tag{36}$$

From the equations (34) and (35), We can see that the third control variable $\Delta\omega$ determines the angle deviation between θ_e and θ and the corresponding rotation velocities. We define a feedback controller for the control value $\Delta\omega$ as follows

$$\Delta\omega = k_3 \cdot \dot{v}_b^2 \cdot c + k_4 \cdot y_b^m \tag{37}$$

where k_3 and k_4 are scale parameters, c is the path curvature, y_b^m is the ball's relative y-position in the robot frame. The first part of equation (37) describes the relationship between the variable $\Delta\omega$ and the centripetal acceleration, which is not only proportional to the centripetal acceleration in magnitude, but also has the same sign of the curvature. The second part is a proportional controller with the ball's relative y-position as the input. The goal of this simple linear controller is to stabilize the ball's relative y-position to zero. This variable $\Delta\omega$ produces an angle derivation between the absolute moving direction of point E and the heading direction of the robot. This angle derivation and the robot's convex front together make the robot capable of holding the rolling ball.

4. EXPERIMENTS

We applied this control algorithm to both simulated and real-world environments. After we tested the control method in the simulator environment, the effectiveness of this control system was evaluated in the real-world environment.

4.1 Simulation Experiments

In our simulator, the movements of the robot and the ball have been calculated from their kinematics equations. The pushing process is considered as a consecutive high frequency and low magnitude compact process. We utilize the basic collision equations, which describe the collisions of two rigid bodies, and the Poisson's Hypothesis to calculate the motion of the two objects before and after the collision. This method was introduced by Yu Wang (1993). The difficulty in this collision simulation is the determination of the collision moment exactly. As a normal method, we detect whether the robot and the ball overlap by some degree due to the limited time, if they have, the simulation calculates back *n* time steps within the last cycle such as the overlap disappears. Since the simulation cycle is enough short, the accuracy of the collision detection is satisfiable. The parameters of our simulator such as the friction coefficient, restitution coefficient are adopted as the experience values. The masses and the moment of inertia of robot and ball are adopted as the real values.

In the simulation, the ball is tested with sinusoidal paths and constant required translation velocities. The simulation results are illustrated in fig. 4,5,6,7, where the normal position error denotes the ball's relative y-coordinate in the robot frame, and the velocity error presents the difference between the ball's real translation velocity and the required one. These results show us the following control method could fast and stable steer the ball converge to the given path, meanwhile the ball doesn't slide away from the robot, because the normal position error is always less than the maximum values ± 0.17 m.

4.2 Real-world Experiments

In this section we applied the following control strategy to the real omnidirectional robot in the real Robocup playing field. From the omnidirectional vision system of our robot, we can receive the real-time location information of our robot and the ball in real time. We also controlled the robot to push the ball following some sinusoidal paths with the constant desired translation velocities. Although the measurement noise and environment disturbance were induced, the experiment results, illustrated in fig. 8,9,10,11, demonstrate the control method has good performance and robustness. Because of the limitation of the experiment environment, the robot only moved about 5m along the x-direction in these experiments.

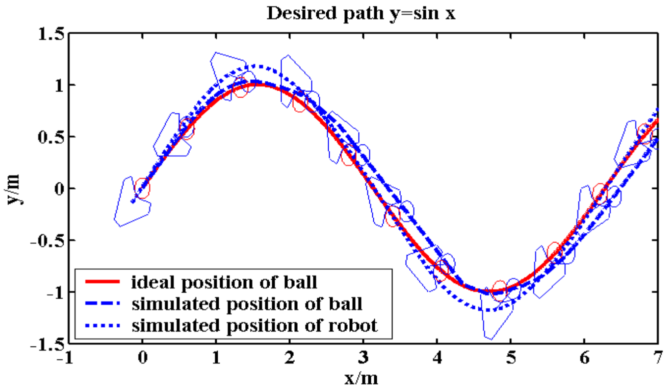

Fig. 4. Simulated following along the Path $y = \sin x$ with $v_b = 0.5$ m/s

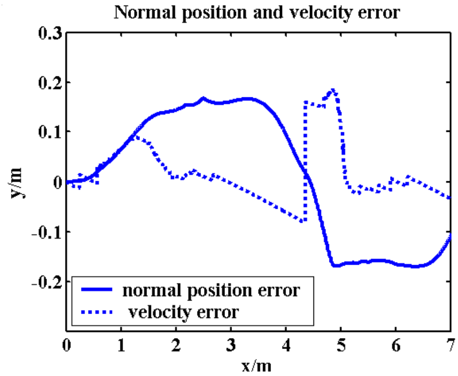

Fig. 5. Simulated following error along the Path $y = \sin x$ with $v_b = 0.5$ m/s

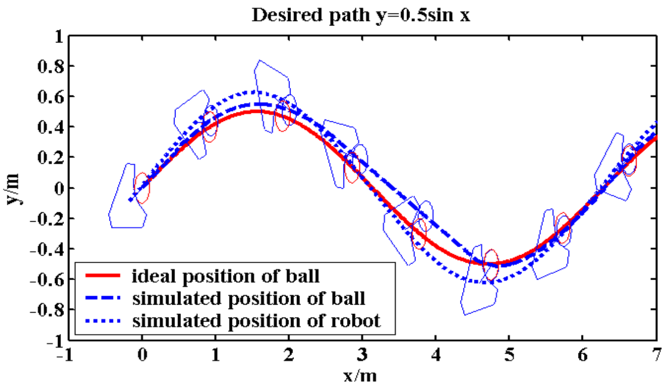

Fig. 6. Simulated following along the path $y = 0.5 \sin x$ with $v_b = 0.6$ m/s

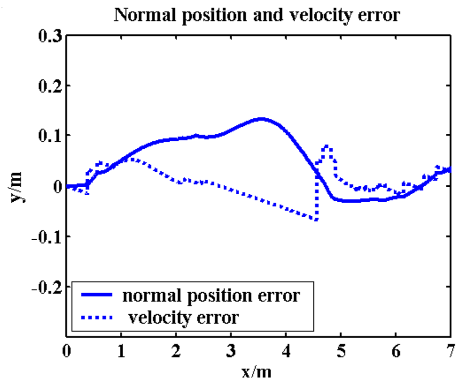

Fig. 7. Simulated following error along the path $y = 0.5 \sin x$ with $v_b = 0.6$ m/s

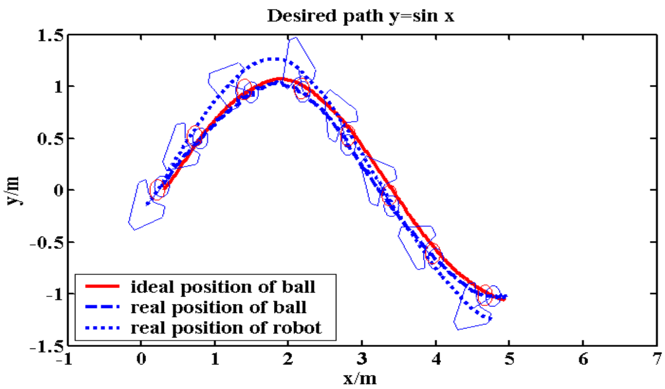

Fig. 8. Real-world following along the path $y = \sin x$ with $v_b = 0.5$ m/s

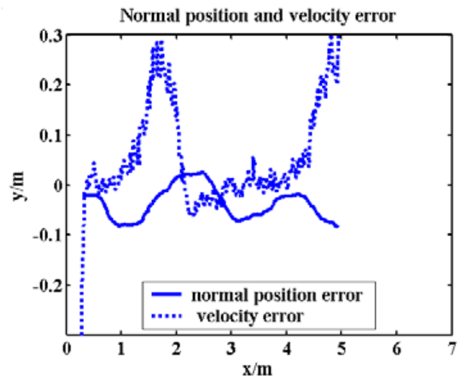

Fig. 9. Real-world following error along the path $y = \sin x$ with $v_b = 0.5$ m/s

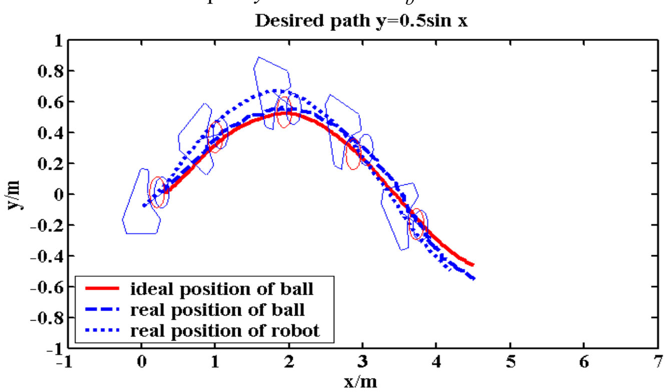

Fig. 10. Real-world following along the path $y = 0.5 \sin x$ with $v_b = 0.6$ m/s

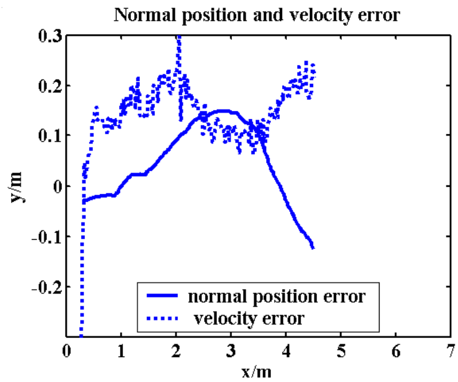

Fig. 11. Real-world following error along the path $y = 0.5 \sin x$ with $v_b = 0.6$ m/s

5. CONCLUSIONS AND FUTURE WORK

In this paper a new path following control method was presented. This approach resolves the control problem of the consecutive mobile robot pushing operation by inducing a reference point as the ball's ideal centre. This reference point is also regarded as a fixed point of the robot itself. The linear feedback control method is used to steer the reference point to follow the given path, and the ball is steered to move around the reference point by the proportional control method.

The simulation and real-world experiments utilized sinusoidal curves with different magnitudes as the ideal paths, and a constant required translation velocity of the ball. The results show that this path following control method can control the omnidirectional robot to apply appropriate pushing operations such that the ball follows the given path stably and robustly up to velocities of 0.6 m/s for a sinusoidal curve.

The desired ball's translation velocity is determined not only by the feedback error, but also by the curvature characteristics of the smooth desired path. A constant translation velocity cannot suit any desired paths. Future work includes steering the ball to follow the desired path with different velocities, especially with lower velocities at high curvature strongly bent points and higher velocity along straight segments.

6. ACKNOWLEDGMENT

The authors would like to thank Patrick Heinemann and Maosen Wang for their creative suggestions and precious help.

REFERENCES

De Wit, C.C., B. Siciliano. and G. Bstin. (1996). *Theory of Robot control*, Springer-Verlag, London.

Lynch, Kevin M. (1996). *Nonprehensile robotic manipulation: controllability and planning*. PhD thesis, Robotics Institute, Carnegie Mello University, Pittsburg, Pennsylvania,

Lynch, Kevin M., Hitoshi Maekawa, and Kazuo Tanie. (1992). *Manipulation and active sensing by pushing using tactile feedback.* In Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 416-421.

Mason, M.T.. (1986). *Mechenics and planning of manipulator pushing operations*. International Journal of Robotics Research, **5(3)**, 53-71, Fall.

Peshkin, Michael A.. and Arthur C. Sanderson. (1988a). T*he motion of a pushed, sliding workpiece.* IEEE Journal of Robotics and Automation, **4(6)**, 569-598.

Peshkin, Michael A. and Arthur C. Sanderson. (1988b) *Planning robotic manipulation strategies for workpieces that slide*. IEEE Journal of Robotics and Automation, **4(5)**, 524-531.

Agarwal, Pankaj K., Jean-Claude Latombe, Prabhakar Raghavan and Rajeev Motwani. (1997). *Nonholonomic path planning for pushing a disk among obstacles*. In proceedings of IEEE International Conference on Robotics and Automation.

Takagi, Seiji and Yoshikuni Okawa. (1991). *Rule-based control of a mobile robot for the push-a-box Operation.* In proceedings of IEEE/RSJ International Workshop on Intelligent Robots and System.

Akella, Srinivas and M.T. Mason. (1992). *Posing polygonal objects in the plane by pushing*. In IEEE International Conference on Robotics and Automation, pages 2255-2262.

Okawa, Yoshikuni and Ken Yokoyama. (1992). *Control of a mobile robot for the push-a-box Operation*. In proceedings of IEEE International Conference on Robotics and Automation.

Wang, Yu and M.T. Mason. (1993). *Two-Dimensional Rigid-Body Collisions With Friction*. Journal of Applied Mechanics **Vol.60**, June, pp. 566.