Obstacle Avoidance of Manipulators with Rate Constraints

Toshiharu Sugie*, Yutaka Kito* and Kenji Fujimoto*

*Department of Systems Science, Graduate School of Informatics
Kyoto University, Uji, Kyoto 611-0011, Japan
sugie@i.kyoto-u.ac.jp, fujimoto@i.kyoto-u.ac.jp

Abstract

The paper presents a new control method which achieves autonomous obstacle avoidance for manipulators with rate constraints. More precisely, in order to achieve the autonomous obstacle avoidance, we exploit the freedom of the coordinate transformation for exact linearization of nonlinear systems. At the same time, we cope with the rate constraints by adopting the state-dependent time scale transformation. Furthermore, we apply this method to an actual 2-link robot manipulator, and evaluate its effectiveness by experiment, which is the most important part of the paper.

1 Introduction

Concerning to the motion planning or master-slave control of robot manipulators, if there exist obstacles in the work space, the manipulators has to avoid the obstacles while they are doing their jobs. In such a case, it is desirable for the manipulators to avoid the obstacles autonomously, because it would reduce both the risk of collision and the burden of planning or teaching.

Lozano-Peres [9] proposed a method to generate the trajectory which avoids the obstacle automatically, and it has been followed by many other works (e.g., [10, 6]). However, unfortunately these are off-line methods. Khatib [7] proposed an on-line obstacle avoidance method using some potential functions, and there are many papers which discussed the real-time obstacle avoidance problem based on the artificial potentials (e.g., [8, 1]). On the other hand, another important factor to consider in motion planning is the rate constraints, because manipulators may become unstable or cannot trace the planned trajectory accurately when they move so fast. Hollerbach [4] tried to cope with this problem by using constant time-scaling. Tanaka and co-workers [14] considered the convergence time in motion planning, where they utilize the idea of state-dependent time scaling which is originally proposed by Sampei and Furuta [11], and developed for the controller design with rate constraints [12, 13]. However, there are few which address both the on-line property and rate constraints in obstacle avoidance so far.

Recently, from the viewpoint of nonlinear control theory, Fujimoto and co-workers pointed out that there exists extra freedom in coordinate transformation for exact linearization [3], and showed that it is possible to utilize the extra freedom for the real-time obstacle avoidance of manipulators [2]. Though it is a quite different approach from the above robotics literature, its effectiveness has been evaluated by experiment. They, however, do not consider the rate constraint of robot manipulators. Therefore, it seems to be interesting to develop the method so as to cope with the rate constraints.

The purpose of this paper is to show how to achieve autonomous obstacle avoidance of manipulators with rate constraints based on Fujimoto's method. More precisely, we achieve the autonomous obstacle avoidance by exploiting the freedom of coordinate transformation in exact linearization, and cope with the rate constraints simultaneously by adopting the state-dependent time scale transformation. Furthermore, we apply this method to an actual 2-link robot manipulator, and evaluate its effectiveness by experiment.

2 Obstacle avoidance via coordinate transformation

2.1 Exact linearization

We consider a rigid link manipulator described by the following dynamic equation

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{k}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^m$ denotes the joint variable, $\boldsymbol{\tau} \in \mathbb{R}^m$ is the input torque or force, $\boldsymbol{k}$ denotes the nonlinear term such as centrifugal forces, and $\boldsymbol{M}$ is a positive definite inertia matrix.

(A) Conventional linearization

For this system, we usually use the linearizing feedback:

$$\boldsymbol{\tau} = \boldsymbol{k}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{M}(\boldsymbol{\theta})\boldsymbol{u}_\theta \tag{2}$$

where $\boldsymbol{u}_\theta$ is a newly defined input. Then, we obtain a linearized system

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{u}_\theta. \tag{3}$$

(B) Extra freedom in linearization

Now, it is known that we have extra freedom in linearization. Namely, all pairs of linearizing coordinate transformations and feedbacks for the system (1) are given by the following pairs with eq.(2) (see e.g. [3]).

$$\boldsymbol{x} = \boldsymbol{\psi}(\boldsymbol{\theta}) \tag{4}$$

$$\boldsymbol{u}_\theta = [\boldsymbol{J}(\boldsymbol{\theta})]^{-1}(\boldsymbol{u}_x - \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})) \tag{5}$$

where $\boldsymbol{\psi} : \boldsymbol{\theta} \mapsto \boldsymbol{x}$ is any diffeomorphism, $\boldsymbol{J} := \partial\boldsymbol{\psi}/\partial\boldsymbol{\theta}$ is the Jacobian matrix of $\boldsymbol{\psi}$, and $\boldsymbol{h}$ is defined by

$$\boldsymbol{h} := (\, \dot{\boldsymbol{\theta}}^{\mathrm{T}} \boldsymbol{H}_1 \dot{\boldsymbol{\theta}}, \cdots, \, \dot{\boldsymbol{\theta}}^{\mathrm{T}} \boldsymbol{H}_m \dot{\boldsymbol{\theta}} \,)^{\mathrm{T}} \tag{6}$$

where $\boldsymbol{H}_i := \partial/\partial\boldsymbol{\theta}(\partial\psi_i/\partial\boldsymbol{\theta})^{\mathrm{T}}$ is a Hessian matrix of ψ_i. In this case, we have

$$\ddot{\boldsymbol{x}} = \boldsymbol{u}_x \tag{7}$$

Then we can design a control system which autonomously avoid the obstacles by using the extra freedom $\boldsymbol{\psi}$. We only consider the case $m = 2$ in the following for simplicity, but it can be essentially generalized to the case $m > 2$.

2.2 Design of coordinate transformation

In this section, we describe how to design a control system for autonomous obstacle avoidance. We will construct such a system by choosing the coordinate transformation $\boldsymbol{\psi}$ in an appropriate way. The objectives of this design are:

(a) The manipulator tracks the reference trajectory precisely when it is far away from the obstacles.

(b) The manipulator autonomously avoids the collision with the obstacles when the reference trajectory approaches to the obstacles.

For the objective (a), we choose $\boldsymbol{\psi}$ as identity when the manipulator is away from the obstacles. When $\boldsymbol{\psi}$ is identity, the manipulator simply tracks the reference trajectory.

For the objective (b), we choose $\boldsymbol{\psi}$ in such a way that it maps a connected obstacle domain into a single point. Then it looks like as if the obstacle consists of the single point from the manipulator in the new coordinate, so the manipulator achieves autonomous obstacle avoidance whenever the reference trajectory dose not go through the single point. Actually, the mapping $\boldsymbol{\psi}$ plays a role to deform the reference trajectory in the neighborhood of the obstacle region.

The obstacle domain can be shown in the joint space. Let the center position of the obstacle be $\boldsymbol{\theta}_0 = (\theta_{01}, \theta_{02})^T$ in the joint space, and the manipulator position be $\boldsymbol{\theta} = (\theta_1, \theta_2)^T$, and define the distance r and the phase φ of $\boldsymbol{\theta}$ from $\boldsymbol{\theta}_0$ by

$$\varphi = \mathrm{atan2}(\theta_2 - \theta_{02}, \theta_1 - \theta_{01}) \tag{8}$$

$$r = \sqrt{(\theta_1 - \theta_{01})^2 + (\theta_2 - \theta_{02})^2} \tag{9}$$

as depicted in the Figure 1, where the domain S_1 represent the obstacle, and S_2 corresponds to the domain in which we deform the coordinate for the above purpose (b). For notational convenience, we define the mapping $\boldsymbol{\psi}_0$ by

$$\begin{bmatrix} \varphi \\ r \end{bmatrix} = \boldsymbol{\psi}_0 \left(\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \right) \tag{10}$$

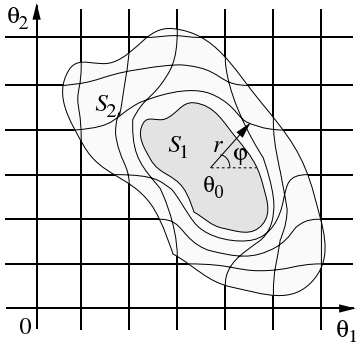

Figure 1: New coordinate

In order to obtain a $\boldsymbol{\psi}$ satisfying the two objectives, we proceed the following steps:

Step1. Define the boundary ∂S_1 of the obstacle region S_1 as:

$$\partial S_1 = \left\{ \mu(\varphi) \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} + \boldsymbol{\theta}_0 \,\middle|\, 0 \le \varphi < 2\pi \right\} \tag{11}$$

where $\mu > 0$ is a smooth periodic function with period 2π.

Step2. Define the boundary ∂S_2 of the deforming domain S_2 which contains S_1 as:

$$\partial S_2 = \left\{ k(\varphi)\mu(\varphi) \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix} + \boldsymbol{\theta}_0 \,\middle|\, 0 \le \varphi < 2\pi \right\} \tag{12}$$

where $k > 1$ is also a smooth periodic function with period 2π.

Step3. Define the function ν that is monotonously increasing with respect to r such that

$$\begin{cases} \nu(r,\varphi) = 0 & (\, r = \mu(\varphi) \,) \\ 0 < \nu(r,\varphi) < r & (\, \mu(\varphi) < r < k(\varphi)\mu(\varphi) \,) \\ \nu(r,\varphi) = r & (\, r \ge k(\varphi)\mu(\varphi) \,) \end{cases} \tag{13}$$

Then we introduce the coordinate transformation $\boldsymbol{\psi}$ which maps the joint angle $\boldsymbol{\theta}$ to the modified polar coordinate $\boldsymbol{x} = (\varphi, \nu)^T$ as follows.

$$\begin{bmatrix} \varphi \\ \nu(\varphi, r) \end{bmatrix} = \boldsymbol{\psi} \left(\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \right) \tag{14}$$

At this stage, we can adopt any stabilizing linear controller K as

$$\boldsymbol{u}_x = K(\boldsymbol{x}_r - \boldsymbol{x}) \tag{15}$$

where $\boldsymbol{x}_r$ denotes the reference trajectory in the (φ, r) coordinate. Then we obtain a control system which autonomously avoid the obstacle.

However, since the above procedure dose not take the joint velocity into account, it may yields very high speed movement of the manipulator, and as a result the system may go unstable. In order to avoid this situation, we adopt the state-dependent time scaling which is proposed by Sampei [11] in the next section.

3 Rate compression via time scale transformation

In this section, we introduce two types of time scale transformation in the previous control system to cope with the rate constraints of manipulator joints.

3.1 Time scale transformation in feedforward part

Suppose that the reference trajectory $\boldsymbol{x}_r(t)$ in the (φ, r) coordinate is given, and that we want to keep the angular velocity of each joint within a specified upper bound $\bar{V}$. To this end, we adopt a new time scale τ which is defined by

$$\frac{dt}{d\tau} = s(\boldsymbol{x}_r) > 0 \tag{16}$$

where s is a positive function to be designed. Then we transform the time-scale of the reference, that is, we inject the modified reference $\boldsymbol{x}_r(\tau)$ in stead of $\boldsymbol{x}_r(t)$ to the closed loop system. For example, if we set $s \equiv 2$, the new reference $\boldsymbol{x}_r(\tau)$ becomes $\boldsymbol{x}_r(t/2)$. This implies that the reference trajectory moves slowly (i.e., it takes double time to do the same job). For notational convenience, let $\boldsymbol{\eta}$ denote the transformation;

$$\boldsymbol{x}_{ref}(t) = \boldsymbol{\eta}(\boldsymbol{x}_r(t)) \tag{17}$$

where $\boldsymbol{x}_{ref}(t) = \boldsymbol{x}_r(\tau)$.

Now we show how to choose the function s so that the maximum joint rate does not exceed the specified value $\bar{V}$. First, we calculate

$$\boldsymbol{\theta}_r = \begin{bmatrix} \theta_{r1} \\ \theta_{r2} \end{bmatrix} = \boldsymbol{\psi}^{-1}(\boldsymbol{x}_r)$$

which is the obstacle avoidance trajectory to be actually tracked by the manipulator joint $\boldsymbol{\theta}$. Let V be

$$V = \max_i \left| \frac{\theta_{ri}(\tau)}{d\tau} \right| \tag{18}$$

then we propose to choose the following positive function.

$$s = \begin{cases} 1 & (V \le V_{max}) \\ \frac{V}{V_{max}} & (V > V_{max}) \end{cases} \tag{19}$$

where V_{max} is a parameter to be designed subject to $V \ge V_{max} > 0$. This implies that the modified reference $\boldsymbol{x}_r(\tau)$ moves slowly when $V > V_{max}$ holds, and $\boldsymbol{x}_r(\tau) = \boldsymbol{x}_r(t)$ holds otherwise.

In the ideal case, that is, if $\boldsymbol{x}(t)$ tracks $\boldsymbol{x}_r(\tau)$ exactly, the angular velocity $\dot{\theta}_i$ in the actual time scale t is always no greater than V_{max}. This can be checked as follows:

$$\left| \frac{d\theta_i(t)}{dt} \right| = \left| \frac{d\theta_{ri}(\tau)}{d\tau} \right| \frac{d\tau}{dt} \le V_{max}$$

3.2 Time scale transformation in the feedback loop

(A) Linearization in new time scales

As well as the reference trajectory, we introduce the time scale transformation in the feedback loop. Here we consider the two time scales τ_1 and τ_2 which are defined by

$$\frac{dt}{d\tau_i} = s_i(\boldsymbol{x}, \dot{\boldsymbol{x}}) > 0, \qquad i = 1, 2 \tag{20}$$

where s_i's are state-dependent positive function to be designed. By straightforward calculation, the system (7) with $\boldsymbol{x} = (x_1, x_2)^T = (\varphi, \nu)^T$ is described in the new time scales by

$$\begin{bmatrix} \frac{dx_1}{d\tau_1} \\ \frac{dx_2}{d\tau_2} \end{bmatrix} = \begin{bmatrix} \frac{d\varphi}{d\tau_1} \\ \frac{d\nu}{d\tau_2} \end{bmatrix} = \begin{bmatrix} s_1\dot{\varphi} \\ s_2\dot{\nu} \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} \frac{d^2x_1}{d\tau_1^2} \\ \frac{d^2x_2}{d\tau_2^2} \end{bmatrix} = \boldsymbol{a} + \boldsymbol{B} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} \tag{22}$$

where $\boldsymbol{a}$ and $\boldsymbol{B}$ are given as follows:

$$\boldsymbol{a} = \begin{bmatrix} s_1(\frac{\partial s_1}{\partial \varphi}\dot{\varphi} + \frac{\partial s_1}{\partial \nu}\dot{\nu})\dot{\varphi} \\ s_2(\frac{\partial s_2}{\partial \varphi}\dot{\varphi} + \frac{\partial s_2}{\partial \nu}\dot{\nu})\dot{\nu} \end{bmatrix} \tag{23}$$

$$\boldsymbol{B} = \begin{bmatrix} s_1^2 + s_1\frac{\partial s_1}{\partial \dot{\varphi}}\dot{\varphi} & s_1\frac{\partial s_1}{\partial \dot{\nu}}\dot{\varphi} \\ s_2\frac{\partial s_2}{\partial \dot{\varphi}}\dot{\nu} & s_2^2 + s_2\frac{\partial s_2}{\partial \dot{\nu}}\dot{\nu} \end{bmatrix} \tag{24}$$

Therefore, from (22) and (7), choosing $\boldsymbol{u}_x$ with the new input $\boldsymbol{v}$ as

$$\boldsymbol{u}_x = \boldsymbol{B}^{-1}(\boldsymbol{v} - \boldsymbol{a}) \tag{25}$$

we obtain the following linear system in the new time scales:

$$\begin{bmatrix} \frac{dx_1}{d\tau_1} \\ \frac{dx_2}{d\tau_2} \\ \frac{d^2x_1}{d\tau_1^2} \\ \frac{d^2x_2}{d\tau_2^2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \frac{dx_1}{d\tau_1} \\ \frac{dx_2}{d\tau_2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{v} \tag{26}$$

Note that the input torque τ is given by

$$\boldsymbol{\tau} = \boldsymbol{k} + \boldsymbol{M}\boldsymbol{J}^{-1}\{-\boldsymbol{h} + \boldsymbol{B}^{-1}\{\boldsymbol{v} - \boldsymbol{a}\}\} \tag{27}$$

from (2), (5) and (25).

(B) Choice of the time scales for rate compression

What we have to do now is to choose the positive function s_i's in (20) in such a way that the rate $\dot{\boldsymbol{\theta}}$ remains within the admissible range. If we compress the magnitude of $\dot{\varphi}$ and $\dot{\nu}$, we may expect to suppress the joint angle rate $\dot{\boldsymbol{\theta}}$. Therefore we choose s_i's from the viewpoint of rate suppression in the (φ, ν) coordinate. In particular, the functions s_i's canceling the complexity of $\boldsymbol{J}^{-1}$ are selected. One of such a choice is given by

$$s_1 = \alpha_1 \frac{\sqrt{r_\varphi^2 + 1}}{r_\nu} + \beta_1 (|\dot{\varphi}| - 1)^2 \quad (28)$$

$$s_2 = \alpha_2 \frac{1}{r_\nu} + \beta_2 (|\dot{\nu}| - 1)^2 \quad (29)$$

where $r_\varphi := \frac{\partial r}{\partial \varphi}$, $r_\nu := \frac{\partial r}{\partial \nu}$, the weighting scalars α_i and β_i $(i = 1, 2)$ are defined as follows:

$$\beta_1 = \begin{cases} \gamma_1 & (|\dot{\varphi}| > 1) \\ 0 & (|\dot{\varphi}| \leq 1) \end{cases} \quad (30)$$

$$\beta_2 = \begin{cases} \gamma_2 & (|\dot{\nu}| > 1) \\ 0 & (|\dot{\nu}| \leq 1) \end{cases} \quad (31)$$

$$\begin{aligned} \alpha_1 &= \exp(-1.2 \frac{k\mu - \nu}{\mu} \sqrt{\frac{\frac{\partial r_\nu}{\partial \nu}}{r_\nu}}) \\ \alpha_2 &= \exp(-\frac{k\mu - \nu}{k\mu} \sqrt{\frac{\frac{\partial r_\nu}{\partial \nu}}{r_\nu}}) \end{aligned} \quad (32)$$

In the above equations, γ_i's are positive scalars to be chosen by the designer. If γ_1 is large, φ tends to behave slowly in the actual time because s_1 becomes large. The behavior of μ corresponds to the choice of γ_2 similarly.

3.3 Design procedure of control system

Combining the coordinate transformation in Section 2 and the time scale transformation shown in this section, we can construct the control system for autonomous obstacle avoidance.

The design procedure is summarized as follows:

Step1. Define μ to determine the boundary ∂S_1 of the obstacle in (11).

Step2. Choose $k(\varphi)$ to define the boundary ∂S_2 of the deformation domain in (12).

Step3. Define the function $\nu(r, \varphi)$ in (13).

Step4. Choose V_{max} in (19) to determine the new time scale for the reference trajectory.

Step5. Choose γ_i $(i = 1, 2)$ in (20) to determine the new time scales for the closed loop.

Step6. Design a linear controller K

$$\boldsymbol{v} = \boldsymbol{K}(\boldsymbol{x}_{ref} - \boldsymbol{x}) \quad (33)$$

The resultant control system is shown in Figure 2.

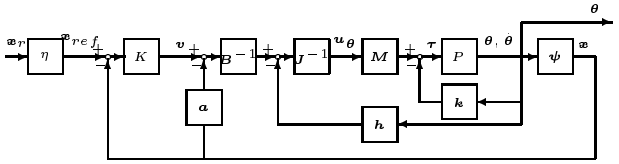

Figure 2: Proposed control system

4 Experimental Evaluation

The most important part of this paper is the experimental evaluation shown in this section.

4.1 Description of the experimental setup

The procedure given in the previous section is now applied to a 2-link robot manipulator. Each joint is driven by a direct drive motor, and each link rotates in the horizontal plane. The dynamics (1) of this system is

$$\begin{aligned} \boldsymbol{\tau} = &\begin{pmatrix} \rho_1 + \rho_2 + 2l_1\rho_3 \cos\theta_2 & \rho_2 + l_1\rho_3 \cos\theta_2 \\ \rho_2 + l_1\rho_3 \cos\theta_2 & \rho_2 \end{pmatrix} \ddot{\boldsymbol{\theta}} \\ &+ \begin{pmatrix} -l_1\rho_3 \sin\theta_2 \dot{\theta}_2^2 - 2l_1\rho_3 \sin\theta_2 \theta_1 \theta_2 + d_1\dot{\theta}_1 \\ l_1\rho_3 \sin\theta_2 \dot{\theta}_1^2 + d_2\dot{\theta}_2 \end{pmatrix} \end{aligned}$$

where $\rho_1 := I_1 + m_1 l_{g1}^2 + m_2 l_1^2$, $\rho_2 := I_2 + m_2 l_{g2}^2$, $\rho_3 := m_2 l_{g2}$, $\boldsymbol{\tau} = (\tau_1, \tau_2)^{\mathrm{T}}$ and $\boldsymbol{\theta} = (\theta_1, \theta_2)^{\mathrm{T}}$. The parameters are defined as follows: τ_i [Nm] denotes the input torque for joint i, m_i [kg] denotes the mass of link i l_i [m] denotes the length of link i l_{gi} [m] denotes the length from the center to joint of link i, I_i [kgm^2] denotes the inertia of link i, d_i [Nms/rad] denotes the friction coefficient of joint i and θ_i [rad] denotes the rotation angle of link i. The concrete values of parameters are $l_1 = 0.25$, $l_2 = 0.30$, $\rho_1 = 2.55$, $\rho_2 = 0.72$, $\rho_3 = 2.60$, $d_1 = 0.2415 d_2 = 0.2457$.

As for the obstacle, we put the cylindrical obstacle in front of the manipulator as shown in Figure 3. Cartesian coordinate (x, y) is set on the horizontal plane, whose origin is located at the center of the first joint. The obstacle is located at 0.7[m] above from the origin, and its diameter is 0.18 [m].

4.2 Design of control system

We construct a control system according to the proposed procedure.

(Step 1) First, we determine the obstacle domain S_1. Taking account of the link width, we set the region is described by

$$x^2 + (y - 0.70)^2 = 0.30^2 \quad (34)$$

in the (x, y) coordinate. This domain is depicted as the dark colored region in Figure 4 in the (φ, r) coordinate.

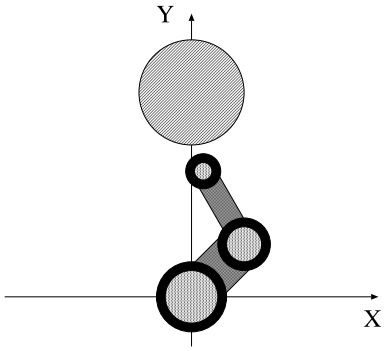

Figure 3: Location of the obstacle and the manipulator

The smooth function $\mu(\varphi)$ which determines the boundary ∂S_1 is given by

$$\mu(\varphi) = \frac{ab}{\sqrt{a^2 \sin^2(\varphi - \varphi_0) + b^2 \cos^2(\varphi - \varphi_0)}} \qquad (35)$$

where $a = 1.78$, $b = 0.42$ and $\varphi_0 = -1.03$.

(Step 2) We choose $k(\varphi)$ as

$$k(\varphi) \equiv 2. \qquad (36)$$

The boundary ∂S_2 is denoted as the thin broken line in Figure 4.

(Step 3) We choose the following C^2 function ν which satisfies (13).

$$\nu(r, \varphi) := \begin{cases} a_1 - \frac{a_1}{a_2} \tan(a_2 - \frac{a_2}{a_1} \frac{r}{\mu(\varphi)}) & (\mu(\varphi) \le r < k\mu(\varphi)) \\ r & (r \ge k\mu(\varphi)) \end{cases}$$

where $a_1 = 3.56$ and $a_2 = 2.33$.

(Step 4) As for the admissible rate upper bound, the manipulator moves stably when the angular velocity of each joint is below 5.0 [rad/s]. So we take

$$V_{max} = 5.0.$$

(Step 5) Concerning to the time scales in the feedback loop, we use the parameter

$$(\gamma_1, \gamma_2) = (10, 5).$$

(Step 5) The PD type control law is adopted here.

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 200(\varphi - \varphi_{ref}) + 30 s_1(\dot{\varphi} - \dot{\varphi}_{ref}) \\ 200(\nu - r_{ref}) + 30 s_2(\dot{\nu} - \dot{r}_{ref}) \end{bmatrix} \qquad (37)$$

where $\boldsymbol{x}_{ref} = (\varphi_{ref}, r_{ref})^T$ denotes the reference of the closed loop. Note that we have to use the derivative with respect to the new time scales (τ_1, τ_2) in the closed loop. This explains why s_i's appear in the above control scheme.

4.3 Experimental results

In the experiment, the reference trajectory is given by

$$\begin{cases} \theta_{1r} = 0 \\ \theta_{2r} = 0.3t + 0.1 \end{cases} \quad (0 \le t < 1)$$

$$\begin{cases} \theta_{1r} = \frac{3.14(t-1)}{3} \\ \theta_{2r} = 0.4 \end{cases} \quad (1 \le t < 4) \qquad (38)$$

$$\begin{cases} \theta_{1r} = 3.14 \\ \theta_{2r} = 0.4 \end{cases} \quad (t \ge 4)$$

Figure 4 shows the trajectory in the (φ, r) coordinate and in the (x, y) coordinate, respectively. If the manipulator follows the given trajectory precisely, it collides with the obstacle as shown in the figures. Note that the manipulator moves from right to left in the figures.

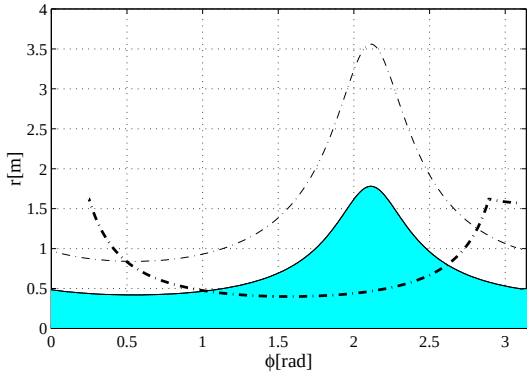

Figure 4: Reference trajectory in the (φ, r) coordinate

Finally, we perform the experiment. Figures 5 and 6 show the time responses of $\dot{\theta}_1(t)$ and $\theta_1(t)$, respectively, by solid lines. The dot and dashed lines correspond to the case where no time scaling is used. From these figures, we can see that the velocity remains within the safety zone, and the manipulator reaches the destination without delay.

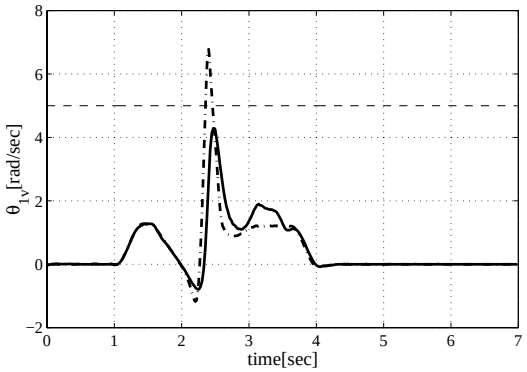

Figure 5: Responses of $\dot{\theta}_1(t)$ with ($V_{max} = 5.0, \gamma_1 = 10, \gamma_2 = 5$)

The solid lines in Figures 7 and 8 show the trajectory of the manipulator in this case. The other line in each figure corresponds to the obstacle avoidance trajectory

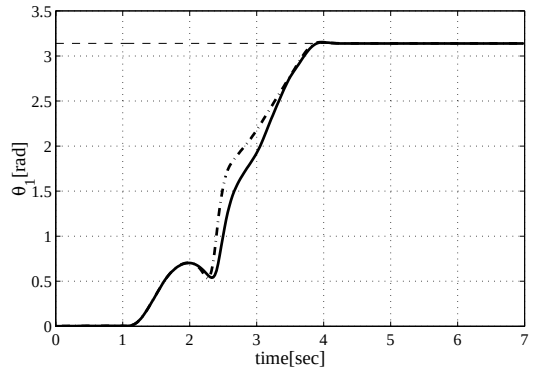

Figure 6: Responses of $\theta_1(t)$ with ($V_{max} = 5.0$,$\gamma_1 = 10$,$\gamma_2 = 5$)

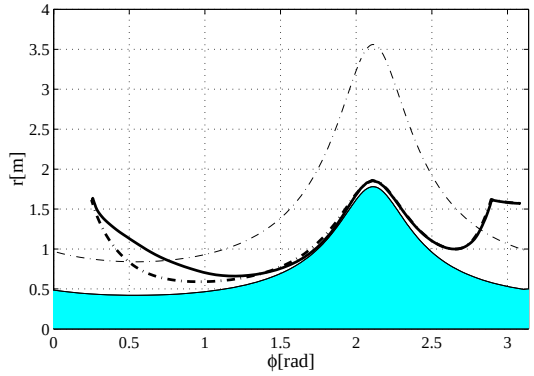

Figure 7: Manipulator configuration in the (φ, r) coordinate with ($V_{max} = 5.0$,$\gamma_1 = 10$,$\gamma_2 = 5$)

produced by the coordinate transformation $\boldsymbol{\psi}$. The manipulator trace the obstacle avoidance trajectory reasonably well.

These experimental results show the effectiveness of the proposed method.

5 Conclusion

In this paper, we have presented a control method of autonomous obstacle avoidance for manipulators with rate constraints. The basic idea behind the method is to utilize the extra freedom in time scale and coordinate transformation for exact linearization. First, we have achieved the obstacle avoidance by using suitably chosen coordinate transformation. Then, introducing the time scale transformation to the feedforward and the feedback loop, we have succeeded in suppressing the velocity of manipulator joints to meet with the rate constraints. Finally, we have shown the effectiveness of the proposed method by extensive experiments, which is most important contribution of the paper.

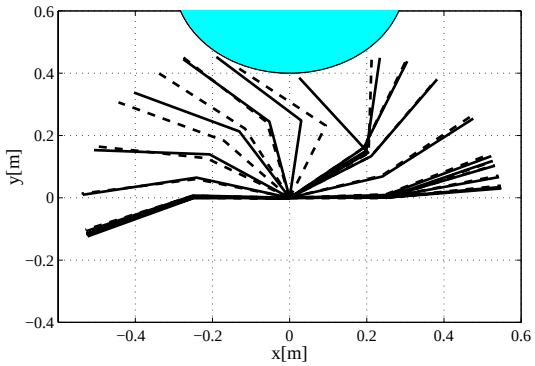

Figure 8: Manipulator configuration in the (x, y) coordinate with ($V_{max} = 5.0$,$\gamma_1 = 10$,$\gamma_2 = 5$)

References

[1] S. Akishita, et al. Velocity potential approach for path planning to avoid moving obstacles. *J. of Advanced Robotics*, 7(5):463, 1993.

[2] K. Fujimoto, K. Kimura, and T. Sugie. Obstacle avoidance of manipulators by using freedom in coordinate transformation for exact linearization. In A. Beghi, L Finesso, and G. Picci, editors, *Mathematical Theory of Networks and Systems*, pages 1007–1010. Il Poligrafo, Padova, 1998.

[3] K. Fujimoto and T. Sugie. Freedom in coordinate transformation for exact linearization and its application to transient behavior improvement. *Proc. 35th IEEE Conf. on Decision and Control*, pages 84–89, 1996.

[4] J. M. Hollerbach. Dynamic scaling of manipulator trajectories. *Trans. ASME*, 106:102–106, 1984.

[5] B. Jakubczyk and W. Respondek. On linearization of control systems. *Bulletin de l'Academie Polonaise des Sciences, Serie des Sciences Mathematiques*, 18:517–522, 1980.

[6] N. Kawarazaki and K. Taguchi. Path planning for a manipulator using differential geometry. *IEEE Int. Conf. on Intelligent Robots and Systems*, pages 130–137, 1995.

[7] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. of Robotics Research*, pages 90–98, 1986.

[8] P. Khosla and R. Volpe. Super-quadratic potential for obstacle avoidance and approach. *IEEE Conf. on Robotics and Automation*, pages 1778–1784, 1988.

[9] T. Lozano-Perez. Automatic planning of manipulator transfer movements. *IEEE Trans. SMC*, 11(10):681–698, 1981.

[10] T. Lozano-Perez. A simple motion-planning algorithm for general robot manipulators. *IEEE Trans. Robotics and Automation*, 3(3):224–238, 1987.

[11] M. Sampei and K. Furuta. On time scaling for nonlinear systems –application to linearization–. *IEEE Trans. Autom. Contr.*, AC-31(5):459–462, 1986.

[12] M. Sampei and K. Furuta. On linearization and control with transformation of the time scale. *Proc. IFAC 10th World Congress*, pages 97–102, 1987.

[13] M. Sampei and K. Furuta. Robot control in the neighborhood of singular points. *IEEE J. of Robotics and Automation*, 4(3):303–309, 1988.

[14] Y. Tanaka, T. Tsuji, M. Kaneko, and P. G. Morrasso. Trajectory generation using time scaled artificial potential field. *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 223–228, 1998.