Proceedings of 2001 IEEE International Symposium on
Computational Intelligence in Robotics and Automation
July 29 - August 1, 2001, Banff, Alberta, Canada

Dynamic Control of Curve-constrained Hyper-Redundant Manipulators

Shugen Ma, Mitsuru Watanabe, Hisashi Kondo

Department of Systems Engineering
Faculty of Engineering, Ibaraki University
4-12-1 Nakanarusawa-Cho, Hitachi-Shi 316-8511, JAPAN
Tel: +81-294-38-5209, Fax: +81-294-38-5272
E-mail: {shugen, kondo}@dse.ibaraki.ac.jp

Abstract

Hyper-redundant manipulators have large number of kinematic degrees of freedom, thus possessing some unconventional features such as the ability to enter a narrow space while avoiding obstacles. To control these hyper-redundant manipulators accurately, the manipulator dynamics must be included into consideration. This is however time-expansive and causes implementation of the real-time control difficult. In this paper, we propose a dynamic control scheme for hyper-redundant manipulators, which is based on analysis in the defined posture space where three parameters were used to determine the manipulator posture. The manipulator dynamics is modeled in the parameterized form with the parameter of the posture space path and the posture space path-tracking feedforward controller is formulated on the basis of the parameterized dynamic equation. Computer simulation was executed to show the validity of the proposed technique, where a hyper-redundant manipulator traces the posture space path well by the proposed feedforward controller. As a result, it is proven that the hyper-redundant manipulator can track the workspace path accurately.

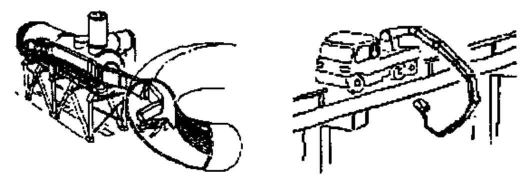
1 Introduction

Hyper-redundant manipulators have a very large relative degrees of kinematic redundancy. These robots, which are analogous in design and operation to the "Trunk of an Elephant", have some unconventional features such as the ability to enter a narrow space while avoiding obstacles. They, thus, can be not only used for applications of industrial robots but also for applications where a conventional industrial robot could never be introduced. The first task to make hyper-redundant manipulators usable must be the implementation of their real-time control system. To control the hyper-redundant manipulators in real-time is, however, difficult because of their large degrees of redundancy.

To solve the control problem of hyper redundant degrees of freedom, algorithms for hyper-redundant manipulators was introduced to resolve their kinematic redundancy in real-time [1, 2, 3, 4, 5]. The dynamics of hyper-redundant manipulators, however, was not included into consideration in these works. The dynamics of hyper-redundant manipulators was first formulated macroscopically by Chirikjian [6]. He used the principles of continuum mechanics to approximately represent the dynamics of hyper-redundant manipulators, where the dynamics of the continuum mechanics is first formulated and then projected onto the actual physical structure. This modeling technique, however, is only an approximation. In this paper, we formulate the dynamics of hyper-redundant manipulators accurately in a parameterized form and propose a dynamic control scheme for hyper-redundant manipulators where the manipulator dynamics is included into consideration. The technique is based on analysis in the defined posture space where three parameters were used to determine the hyper-redundant manipulator posture. The joint torques are formulated in the parameterized parameter of the posture space path and the posture space path-tracking feedforward controller is introduced based on the parameterized joint torque formulation. The proposed approach has the advantage over other existed ones that this scheme can be used for the real-time control of hyper-redundant manipulators. Computer simulation was executed to show that the hyper-redundant manipulator traces the path well by the proposed feedforward controller. Note that the discussion in this paper is limited to 2 dimensional work space problems in order to understand clearly the analysis technique of the posture space. Even though, while we consider the applications shown in figure 1, the scheme proposed in this paper can be directly used and should be useful.

The paper is organized as follows: Section 2 reviews the

0-7803-7203-4/01/$10.00 ©2001 IEEE



(a) Maintenance robot for nuclear reactor (b) Maintenance robot for highway/freeway

Fig. 1: Application examples of the 2 dimensional hyper-redundant manipulator

kinematic control of hyper-redundant manipulators. Section 3 introduces the posture space of hyper-redundant manipulators and the usage of the posture space. Section 4 represents the dynamics of hyper-redundant manipulators and introduces a posture space path-tracking control scheme for hyper-redundant manipulators. The computer simulation was performed and its results are given in section 5. Section 6 gives conclusion of the paper.

2 Kinematic Control of Hyper-redundant Manipulators

Traditionally, kinematic analysis and motion planning for redundant manipulators have relied upon a pseudoinverse [9, 10], generalized inverse [11, 12], or extended inverse [13] of the manipulator Jacobian matrix. These schemes, however, are difficult to implement in the real-time control when including the dynamic effect of the arm into consideration, because of their computational inefficiency in manipulation of matrices and non-linear terms. The algorithms for the position-coordination of a hyper-redundant manipulator end-point [1, 2, 3, 4, 5, 7], and an algorithm for the shape control of its whole arm [8] have been introduced to resolve the kinematic redundancy. In this section, we reviews the techniques for describing the hyper-redundant manipulator kinematics previously-proposed [1, 2, 3, 4, 5], and the position-coordination control technique of a hyper-redundant manipulator [1, 2].

Curvilinear theory was used to represent the posture of kinematically hyper-redundant manipulators [1, 2, 3, 5, 7], with an assumption that a hyper-redundant manipulator can be captured by a continuous curve regardless of mechanical implementation. Therein, the arm posture of the hyper-redundant manipulator is modeled by a continuous curve with the curvature function $\kappa(s)$, where s is the distance along the curve measured from the base. As we explored, a serpenoid curve is better utilized to define the arm posture of hyper-redundant manipulators, because the solution from the given boundary position and the length of curve to the form of curve is easily-obtained [1].

Consider a serpeniod curve [14] or so-called Backbone curve [3], whose curvature is defined by*

$$\kappa(s) = \frac{2\pi}{\ell} a_1 \cos(\frac{2\pi}{\ell}s) + \frac{2\pi}{\ell} a_2 \sin(\frac{2\pi}{\ell}s) \qquad (1)$$

where a_1 and a_2 are the coefficients to define the curvature function, and ℓ is the curve length which is equal to the length of the manipulator arm that is to be configured.

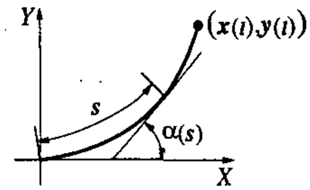


Fig. 2: Definition of the inclination angle $\alpha(s)$, and the (X,Y) coordinate

The angle $\alpha(s)$ as shown in figure 2, which represents the inclination angle of the vector with respect to x-axis on the curvilinear length s, and the end-point position $(x(\ell), y(\ell))$ can be derived through integration with respect to the curve length, and given by

$$\begin{aligned} \alpha(s) &= \alpha_0 + \int_0^s \kappa(u)du \qquad (2) \\ &= \alpha_0 + a_1 \sin\frac{2\pi}{\ell}s - a_2 \cos\frac{2\pi}{\ell}s + a_2, \\ x(\ell) &= \int_0^\ell \cos(\alpha(s))ds \qquad (3) \\ &= \cos(a_2+\alpha_0) J_0\left(\sqrt{a_1^2+a_2^2}\right)\ell, \\ y(\ell) &= \int_0^\ell \sin(\alpha(s))ds \qquad (4) \\ &= \sin(a_2+\alpha_0) J_0\left(\sqrt{a_1^2+a_2^2}\right)\ell \end{aligned}$$

where α_0 is the initial inclination angle of the vector with respect to x-axis at the start point or the arm base, and J_0 is the zero-order Bessel function [15].

While the end position of the curve is given, its form (*or posture*) given by the curvature $\kappa(s)$ corresponding to the given initial inclination angle α_0 can be defined by the coefficients a_1 and a_2. The coefficients a_1 and a_2 are derived by solving the equations (3) and (4), and given by

$$\begin{aligned} a_2 &= \tan^{-1}\left(\frac{y(\ell)}{x(\ell)}\right) - \alpha_0, \qquad (5) \\ a_1 &= \left(\left[J_0^{-1}\left(\left(x(\ell)^2+y(\ell)^2\right)^{\frac{1}{2}}/\ell\right)\right]^2 - a_2^2\right)^{\frac{1}{2}} \end{aligned}$$

*$\kappa(s) = A\sin(\frac{2\pi}{\ell}s+\phi)$; $A = \frac{2\pi}{\ell}\sqrt{a_1^2+a_2^2}$, $\phi = \tan^{-1}\frac{a_1}{a_2}$

0-7803-7203-4/01/$10.00 ©2001 IEEE

where J_0^{-1} is the restricted inverse zero-order Bessel function [15]. The inverse solution of the curve (*the from of the curve*) is, thus, derived and defined by the coefficients a_1 and a_2. Of course, if we changed the initial inclination angle α_0, the form of the curve would be changed too, as shown in equation (2). Thus, the form of the serpenoid curve is determined by three variables a_1, a_2, and α_0.

The arm posture of the hyper-redundant manipulator can be configured by restricting the arm on the defined serpenoid curve. In this case, the joint angles of the manipulator are derived from the grade of the tangent line, and can be expressed as

$$\begin{aligned} q_1 &= \alpha(\frac{L}{2}) = a_1 \sin(\frac{\pi}{n}) + a_2\left(1 - \cos(\frac{\pi}{n})\right) + \alpha_0, \\ q_i &= \alpha\left((i-1+\frac{1}{2})L\right) - \alpha\left((i-1-\frac{1}{2})L\right) \\ &= a_1\left[\sin\left(\frac{\pi}{n}(2i-1)\right) - \sin\left(\frac{\pi}{n}(2i-3)\right)\right] \\ &\quad -a_2\left[\cos\left(\frac{\pi}{n}(2i-1)\right) - \cos\left(\frac{\pi}{n}(2i-3)\right)\right] \end{aligned} \tag{6}$$

where $i = 2, 3, \cdots, n$ (*n is number of links of the manipulator*), and L is the length of the link, equal to ℓ/n. This discrete method through which the discrete manipulator arm (*its link length is* 0.5 *and number of links is* 7) is restricted onto the continuous serpenoid curve, makes the position error of the end position of the manipulator smaller than 10^{-6} [16].

As a result, we know that the inverse kinematic solution, in which the joint angles of the hyper-redundant manipulator arm are calculated from the given end position $(x(\ell), y(\ell))$, has been simply derived by equations (5) and (6). It must be understandable that the scheme is advanced in computational cost and makes the real-time position control of hyper-redundant manipulators possible.

It should be noted here that, the introduced technique highly restricts the working area and the flexibility of hyper-redundant manipulators by constraining the manipulator arm onto the serpenoid curve. The hyper-redundant manipulator, however, is not only used into highly-restricted working area, but also must be effectively used in an open space or simply-restricted space.

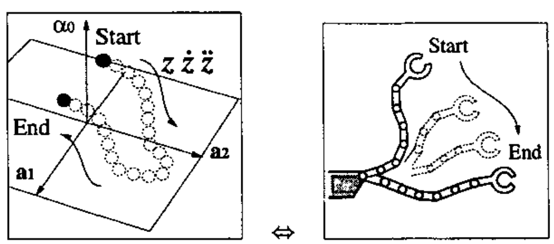
3 Posture Space of Hyper-redundant Manipulators

The configuration space of the manipulator was widely used for its motion planning. For hyper-redundant manipulators, however, since they have large number of kinematic degrees of freedom, the configuration space could not be still a possible tool for their motion planning. In this section, we introduce a new space for the motion planning of hyper-redundant manipulators.

As stated in section 2, we know that the arm posture of hyper-redundant manipulators can be determined through three variables a_1, a_2, and α_0 while the manipulator arm is restricted onto the serpenoid curve. That is, while giving the values of the a_1, a_2, and α_0, the posture of the manipulator arm is determined directly and simply through the equation (6). It should be noted that two parameters a_1, a_2, of course, can be also utilized to define the posture of the hyper-redundant manipulator, while α_0 is given as a constant value. That we use all three parameters a_1, a_2, and α_0 here is for increasing the possible working area in the work space that is reduced due to the fact that the arm is restricted on the serpenoid curve, as well as for giving the redundancy and/or flexibility of the curve-constrained hyper-redundant manipulator.

The posture space of hyper-redundant manipulators is thus defined as follows:

Definition: A **posture** $\lambda\,(a_1, a_2, \alpha_0)$ of a hyper-redundant manipulator is a specification of the configuration (*or form*) of the hyper-redundant manipulator. The **posture space** of the hyper-redundant manipulator is the space Æ of all possible postures λ of the hyper-redundant manipulator. An unique posture of Æ is arbitrarily selected ($a_1 = 0,\ a_2 = 0,\ \alpha_0 = 0$ *are generally selected, in the case that the arm posture is a straight line and the arm is on x-axis*) and is called as the **reference posture** of the hyper-redundant manipulator. It is denoted by ø.



(a) Posture Space (b) Work Space

Fig. 3: Mapping between the posture space and the work space

One of λ (*shown by "∘" in figure 3 (a)*) defines a posture of the arm as shown in figure 3 (b), and the path in the posture space defines a sequence of arm postures. Comparing the end position that is two degrees of freedom with the position in the posture space, one redundancy exists. This redundancy can be used to obstacle avoidance, torque minimization, and the like. As an example to show how the posture space is used, the obstacle collision-free path of a hyper-redundant manipulator was generated by means of the posture space analysis [7]. Figure 4 shows the obtained obstacle avoidance path in the posture space and figure 5 shows the corresponding arm postures of the hyper-redundant manipulator, respectively.

Note that the orientation of the manipulator end-effector can be controlled through a wrist-mounted actuator. We considered separately the position control and the orientation control of the manipulator end-effector and only discussed the position control in this study.

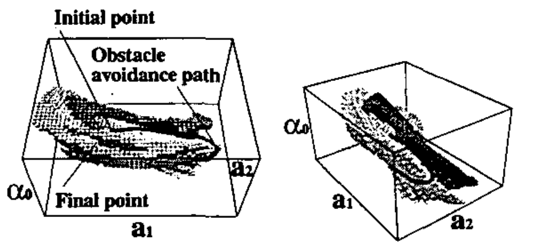


Fig. 4: The obstacle collision-free path in the posture space

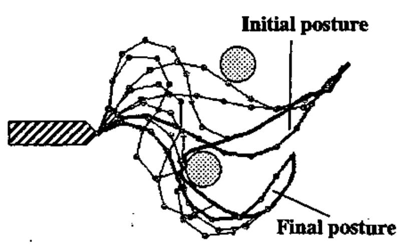


Fig. 5: Manipulator postures in the work space corresponding to figure 4

4 Dynamic Control of Hyper-redundant Manipulators

In this section, we first formulate the dynamics of a hyper-redundant manipulator and then introduce a feedforward controller for a hyper-redundant manipulator to track a posture space path.

4.1 Formulation of the parameterized path-tracking dynamics

Rewriting the equation (6) in vector form, we have

$$q(\boldsymbol{\lambda}) = \zeta(\boldsymbol{\lambda}) \tag{7}$$

where $\zeta \in \Re^n$ is the n-dimensional vector linear-function of the 3-dimensional vector $\boldsymbol{\lambda}$. Differentiating equation (7) with respect to time, we have

$$\dot{q}(\boldsymbol{\lambda}, \dot{\boldsymbol{\lambda}}) = \mathbf{J}_{\boldsymbol{\lambda}}\dot{\boldsymbol{\lambda}} \tag{8}$$

$$\ddot{q}(\boldsymbol{\lambda}, \dot{\boldsymbol{\lambda}}, \ddot{\boldsymbol{\lambda}}) = \mathbf{J}_{\boldsymbol{\lambda}}\ddot{\boldsymbol{\lambda}} + \dot{\mathbf{J}}_{\boldsymbol{\lambda}}\dot{\boldsymbol{\lambda}} \tag{9}$$

where $\mathbf{J}_{\boldsymbol{\lambda}} \in \Re^{n\times 3}$ is the Jacobian matrix, with elements given by $J_{\boldsymbol{\lambda}}(i,j) = \partial\zeta_i/\partial\lambda_j$. The elements of the Jacobian matrix $\mathbf{J}_{\boldsymbol{\lambda}}$ given by

$$J_{\boldsymbol{\lambda}}(i,1) = \begin{cases} \sin(\frac{\pi}{n}), & i=1 \\ \sin\left(\frac{\pi}{n}(2i-1)\right) & \\ \quad -\sin\left(\frac{\pi}{n}(2i-3)\right), & i=2,..,n \end{cases}$$

$$J_{\boldsymbol{\lambda}}(i,2) = \begin{cases} \left(1-\cos(\frac{\pi}{n})\right), & i=1 \\ \cos\left(\frac{\pi}{n}(2i-3)\right) & \\ \quad -\cos\left(\frac{\pi}{n}(2i-1)\right), & i=2,..,n \end{cases}$$

$$J_{\boldsymbol{\lambda}}(i,3) = \begin{cases} 1, & i=1 \\ 0, & i=2,\cdots,n \end{cases}$$

are constant and time-independent. Its time-derivative $\dot{\mathbf{J}}_{\boldsymbol{\lambda}} \in \Re^{n\times 3}$ is thus the zero matrix, and the joint accelerations are become into

$$\ddot{q}(\boldsymbol{\lambda}, \ddot{\boldsymbol{\lambda}}) = \mathbf{J}_{\boldsymbol{\lambda}}\ddot{\boldsymbol{\lambda}} \tag{10}$$

and are velocity-independent or not the function of $\dot{\boldsymbol{\lambda}}$.

Assume that the geometric path in the posture space is given in the parameterized form by a vector function $\boldsymbol{\eta}(z) \in \Re^3$ of the scalar path parameter $z \in \Re$, $z_0 \leq z \leq z_T$, where $\boldsymbol{\eta}(z_0)$ is the start point, and $\boldsymbol{\eta}(z_T)$ is the end point of the path. Thus, we have

$$\begin{aligned} q(z) &= \zeta(\boldsymbol{\eta}(z)) = \hat{\zeta}(z) \\ \dot{q}(z,\dot{z}) &= \mathbf{J}_{\boldsymbol{\lambda}}\boldsymbol{\eta}'\dot{z} = \hat{\boldsymbol{J}}_z\dot{z} \\ \ddot{q}(z,\dot{z},\ddot{z}) &= \mathbf{J}_{\boldsymbol{\lambda}}\left(\boldsymbol{\eta}'\ddot{z} + \boldsymbol{\eta}''\dot{z}^2\right) = \hat{\boldsymbol{J}}_z\ddot{z} + \dot{\hat{\boldsymbol{J}}}_z\dot{z} \end{aligned} \tag{11}$$

where $\boldsymbol{\eta}'(z) = d\boldsymbol{\eta}(z)/dz \in \Re^3$, $\boldsymbol{\eta}''(z) = d^2\boldsymbol{\eta}(z)/dz^2 \in \Re^3$, $\hat{\zeta}(z) = \zeta(\boldsymbol{\eta}(z)) \in \Re^n$, $\hat{\boldsymbol{J}}_z(z) = \mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta}(z))\boldsymbol{\eta}'(z) \in \Re^n$, and $\dot{\hat{\boldsymbol{J}}}_z(z,\dot{z}) = \mathbf{J}_{\boldsymbol{\lambda}}(\boldsymbol{\eta}(z))\boldsymbol{\eta}''(z)\dot{z} \in \Re^n$, respectively. Same as joint angles, velocities, and accelerations, the joint torques can also be represented in the parameterized form. As known, the joint torques $\boldsymbol{\tau} \in \Re^n$ of the manipulator can be given in a well-known closed form

$$\boldsymbol{\tau} = \mathbf{M}_{(\boldsymbol{q})}\ddot{\boldsymbol{q}} + \boldsymbol{c}_{(\boldsymbol{q},\dot{\boldsymbol{q}})} + \boldsymbol{g}_{(\boldsymbol{q})} \tag{12}$$

where $\mathbf{M} \in \Re^{n\times n}$ is the inertia matrix, $\boldsymbol{c} \in \Re^n$ is the torque vector of Coriolis and centrifugal forces, and $\boldsymbol{g} \in \Re^n$ is the torque vector of gravity force, respectively. Substituting equation (11) for equation (12), we have

$$\boldsymbol{\tau}(z,\dot{z},\ddot{z}) = \boldsymbol{\mu}(z)\ddot{z} + \boldsymbol{\nu}(z,\dot{z}) \tag{13}$$

where $\boldsymbol{\mu}(z) = \mathbf{M}_{(\boldsymbol{q}(z))}\hat{\boldsymbol{J}}_z = \hat{\mathbf{M}}_{(z)}\hat{\boldsymbol{J}}_z \in \Re^n$, $\boldsymbol{\nu}(z,\dot{z}) = \hat{\mathbf{M}}_{(z)}\dot{\hat{\boldsymbol{J}}}_z\dot{z} + \hat{\boldsymbol{c}} + \hat{\boldsymbol{g}} \in \Re^n$, $\hat{\boldsymbol{c}}_{(z,\dot{z})} = \boldsymbol{c}_{(\boldsymbol{q}(z),\dot{\boldsymbol{q}}(z,\dot{z}))} \in \Re^n$, and $\hat{\boldsymbol{g}}_{(z)} = \boldsymbol{g}_{(\boldsymbol{q}(z))} \in \Re^n$. Equation (13) shows the joint torques for the hyper-redundant manipulator to track the geometric path in the posture space.

4.2 Feedforward path-tracking controller

The parameterized path-tracking dynamics in the posture space, given in equation (13), can provide a feedforward path-tracking controller design strategy. Since the inverse solution of equation (11) does not exist and the feedforward controller shows almost same accuracy to track the path as the computed torque controller [17], we utilized the feedforward controller for the hyper-redundant manipulator to track the posture space path.

The feedforward controller augments the basic PD controller by providing a set of nominal torques $\boldsymbol{\tau}_{ff}$:

$$\boldsymbol{\tau}_{ff}(z_d, \dot{z}_d, \ddot{z}_d) = \boldsymbol{\mu}(z_d)\ddot{z}_d + \boldsymbol{\nu}(z_d, \dot{z}_d) \tag{14}$$

and then the feedforward controller for the hyper-redundant manipulator to track the posture space path can be given by

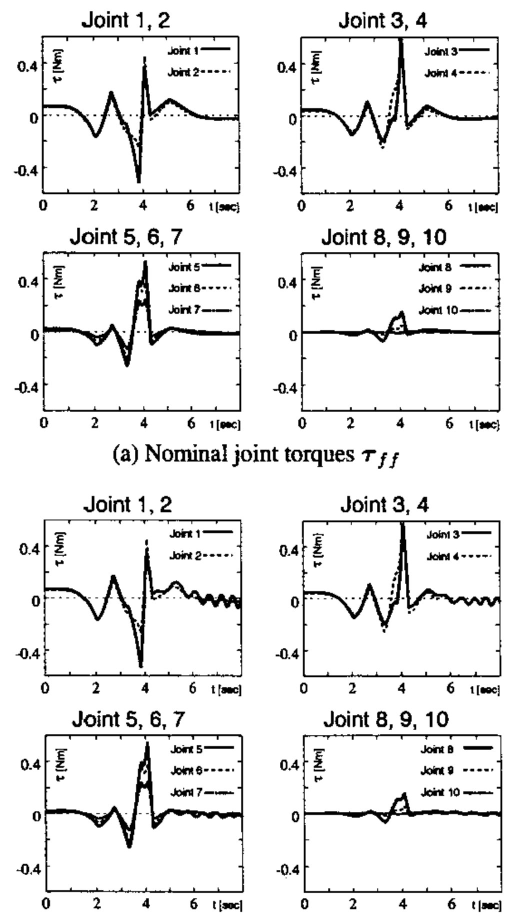
$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{\tau}_{ff}(z_d, \dot{z}_d, \ddot{z}_d) \\ &\quad + \boldsymbol{K}_v\left(\hat{\boldsymbol{J}}_z(z_d)\dot{z}_d - \dot{\boldsymbol{q}}\right) + \boldsymbol{K}_p\left(\hat{\zeta}(z_d) - \boldsymbol{q}\right)\end{aligned} \tag{15}$$

where $\boldsymbol{K}_v$ and $\boldsymbol{K}_p$ are $n \times n$ diagonal matrices of velocity and position gains, and $z_d, \dot{z}_d, \ddot{z}_d$ are the commanded path-tracking inputs that are given by a user. As seen, the feedforward controller term $\boldsymbol{\tau}_{ff}$ can be thought of as a set of nominal torques which allow the dynamics (13) to be linearized about the operating points z_d, $\dot{z}_d$, and $\ddot{z}_d$. Therefore, it is reasonable to add the linear feedback terms $\boldsymbol{K}_v\left(\hat{\boldsymbol{J}}_z(z_d)\dot{z}_d - \dot{\boldsymbol{q}}\right) + \boldsymbol{K}_p\left(\hat{\zeta}(z_d) - \boldsymbol{q}\right)$ as the control for the linearized system. These feedforward terms can be computed off-line, in contract to the computed torque controller where the dynamics must be computed on-line.

5 Computer simulation

We use a 10-DOF (*Degrees of Freedom*) hyper-redundant manipulator to evaluate the validity of the proposed dynamic control scheme. The length of each link of the arm is 0.08 $[m]$, the mass of each link is set as $m = 0.3[kg]$, and inertia parameter is derived by seeing the link as an uniform beam. The posture space path shown in figure 4, that was generated by the obstacle collision-free generation algorithm proposed in [7], is tracked by the constant *bang-bang* acceleration in the motion time $T = 8$ $[s]$. The manipulator dynamics is integrated by Euler integration at the time interval of 5 $[ms]$, where gravity is neglected. The position and velocity error feedback-gains of the PD control term are set as 4.0 and 0.02, respectively.

Figure 6 shows the joint torques of the feedforward controller for the hyper-redundant manipulator to track the posture space path shown in figure 4, while figure 7 shows the path-tracking errors of the feedforward controller (15), the PD controller $\boldsymbol{K}_v\left(\hat{\boldsymbol{J}}_z(z_d)\dot{z}_d - \dot{\boldsymbol{q}}\right) + \boldsymbol{K}_p\left(\hat{\zeta}(z_d) - \boldsymbol{q}\right)$, and the open-loop controller where only the nominal torques $\boldsymbol{\tau}_{ff}$ are the input of system. The path-tracking errors for the open-loop controller are extremely larger, the PD controller reduced the errors, but still has large errors. The path-tracking errors are significantly reduced by the feedforward controller. The feedforward controller showed good performance and made the posture space path well traced. As a result, we know that the hyper-redundant manipulator can track the work space path accurately by the proposed posture space feedforward controller.



(a) Nominal joint torques $\boldsymbol{\tau}_{ff}$

(b) Joint torque input $\boldsymbol{\tau}$

Fig. 6: Nominal joint torques $\boldsymbol{\tau}_{ff}$ and joint torque inputs $\boldsymbol{\tau}$ of the feedforward controller for the hyper-redundant manipulator to track the posture space path shown in figure 4

It should be noted here that, since there has no solution from the joint angle q to the posture space parameter $\boldsymbol{\lambda}$, the path-tracking errors in the posture space are derived through the pseudoinverse of the matrix $\mathbf{J}_{\boldsymbol{\lambda}}$. The points in the posture space were obtained by solving the equation $\boldsymbol{\lambda} = \mathbf{J}^{+}_{\boldsymbol{\lambda}} q$. The position error shown in figure 7 is approximately given by $||\lambda_d - \lambda|| = ||\eta(z_d) - \lambda||$, the distance between the commanded position point and the current point of the manipulator posture in the posture space.

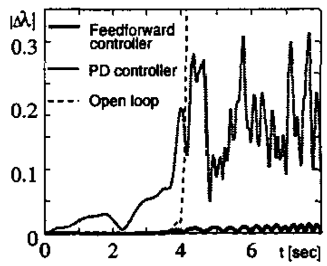


Fig. 7: Path-tracking errors of the feedforward controller (15), the PD controller $K_v\left(\hat{J}_z(z_d)\dot{z}_d - \dot{q}\right) + K_p\left(\hat{\zeta}(z_d) - q\right)$, and the open-loop controller where only the nominal torques τ_{ff} are the input of system

At the last of this section, we make an issue of the computation time. In the case of using a PC Computer (*Pentum-333MHz*), the dynamic path-tracking control of the hyper-redundant manipulator takes less than 2.19 seconds' calculation time, even when the feedforward terms are calculated on-line. If we calculate the the feedforward terms off-line, more little calculation time is required. The implementation of the proposed control scheme to a real hyper-redundant manipulator where the calculation time is much smaller that the operation time, thus, must be possible.

6 Conclusions

In this paper, we proposed a new dynamic control scheme for hyper-redundant manipulators that is possible to implement to real-time control. The new control scheme is based on analysis in the posture space where three parameters were used to determine the hyper-redundant manipulator posture. The dynamics of hyper-redundant manipulator was modeled in the parameterized form with the parameter of the posture space path and the posture space path-tracking feedforward controller was formulated on the basis of the parameterized dynamic equation. Computer simulations showed that the posture space path-tracking errors for the traditional PD controller are large, but they are significantly reduced by the proposed path-tracking feedforward controller. The proposed posture space path-tracking feedforward controller showed good performance and made the hyper-redundant manipulator track the work space path accurately.

References

[1] S. Ma, S. Hirose, and H. Yoshinada, "Development of a hyper-redundant manipulator for maintenance of nuclear reactors", Int. J. of Advanced Robotics, vol.9, no.3, pp.281—300, 1995

[2] G. S. Chirikjian and J. W. Burdick, "An obstacle avoidance algorithm for hyper-redundant manipulators", in Proc. IEEE Int. Conf. on Robotics and Automation, vol.1, pp.625—631, 1990

[3] G. S. Chirikjian and J.W. Burdick, "A modal approach to hyper-redundant manipulator kinematics", IEEE Trans. on Robotics and Automation, vol.10, no.3, pp.343–354, 1994

[4] Chirikjian, G.S., J.W. Burdick, "A geometric approach to hyper-redundant manipulator obstacle avoidance," ASME Journal of Mechanical Design, vol.114, December, pp.580—585, 1992

[5] G. S. Chirikjian and J.W. Burdick, "Kinematically optimal hyper-redundant manipulator configurations", IEEE Trans. on Robotics and Automation, vol.11, no.6, pp.794–806, 1995

[6] G. S. Chirikjian, "Hyper-redundant manipulator dynamics: A continuum approximation", Advanced Robotics, vol.9, no. 3, pp.217—243, 1995

[7] S. Ma and M. Konno, "An obstacle avoidance scheme for hyper-redundant manipulators — Global motion planning in posture space", in Proc. 1997 IEEE Int. Conf. on Robotics and Automation, vol.1, pp.161–167, 1997

[8] H. Mochiyama, *Shape control of manipulators with hyper degrees of freedoms*, Ph.D. thesis, (Japan Advanced Institute of Science and Technology, School of Information Science), 1998

[9] C. A. Klein and C. H. Huang, "Review of the pseudoinverse for control of kinematically redundant manipulators", IEEE Trans. on Syst., Man, Cyber., vol.13, no.2, pp.245—250, 1983

[10] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multi-body mechanisms", IEEE Trans. on System, Man, and Cybernetics, vol.7, no.12, pp.868—871, 1977

[11] D. N. Nenchev, "Redundancy resolution through local optimization: a review", Journal of Robotic Systems, vol.6, no.6, pp.769—798, 1989

[12] T. Shamir and Y. Yomdin, "Repeatability of redundant manipulators: Mathematical solution of the problem", IEEE Trans. on Automatic Control, vol.33, no.11, pp.1004—1009, 1988

[13] J. Baillieul, "Kinematic programming alternatives for redundant manipulators", in Proc. IEEE Int. Conf. on Robotics and Automation, vol.2, 1985

[14] S. Hirose, *Biologically Inspired Robots*, Kogyo Chosakai Publishing Co., Ltd., Tokyo, 1987 (in Japanese); also Oxford, Oxford University Press, 1993

[15] I. N. Sneddon, *Special Functions of Mathematical – Physics and Chemistry –*, London: Oliver & Boyd, 1961

[16] K. Yokoshima, *Study on the control of the Moray-drive type multijoint manipulators with hyper degrees*, ME dissertation, (Dept. of Mechanical Engineering Science, Tokyo Institute of Technology), 1992

[17] C. H. An, C. G. Atkeson, and J. M. Hollerbach, *Model-Based Control of a Robot Manipulator*, The MIT Press, 1988