Time Optimal Control for Underwater Robot Manipulators Based on Iterative Learning Control and Time-Scale Transformation

Norimitsu Sakagami and Sadao Kawamura
Human Robotics Lab, Ritsumeikan University,
Kusatsu city, Shiga, 525-8577, JAPAN
sakanori@se.ritsumei.ac.jp, kawamura@se.ritsumei.ac.jp

Abstract- **For underwater robot manipulators, this paper presents a new time optimal control method without parameter estimation. The proposed method is based on iterative learning control and time-scale transformation. With this method, it is not necessary to estimate any physical parameters in order to form an input torque pattern that realizes a minimum time motion. Therefore, this method is very practical for control of the underwater robot manipulators with very complicated dynamics expressed by a lot of parameters including hydrodynamic parameters such as added-mass and drag coefficients. To realize time optimal control, we consider a class of motions whose spatial paths are fixed, but whose time trajectories (speed patterns) are changed. By using time-scale transformation, the feedforward input pattern is formed from the basic torque patterns obtained through iterative learning control. In this paper, we theoretically derive the time optical control method based on iterative learning control and time-scale transformation. The effectiveness of our proposed method is verified by sevral experiments and the results are presented.**

I. INTRODUCTION

For the development of ocean resources and the preservation of the marine environment, a number of ROVs and AUVs are utilized around the world. Their main operations are observation, survey, inspection, maintenance, repair, and construction, for scientific researches and offshore industry. These operations are frequently performed by manipulators mounted on the vehicles. Therefore, accurate control and planning for the manipulators are highly demanded to improve efficiency of their work. In fact, many researches on control and planning for manipulators, as in [1] – [4], are conducted so as to realize advanced manipulation tasks.

In this paper, especially, we focus on motion planning for the underwater manipulators. In general, motion planning is a useful method in many applications from the viewpoints of time optimal control and minimization of energy or torque. For motion planning, one conventional method is a model-based approach in which the input torque patterns are calculated by using the physical parameters of the manipulators. By the model-based approach, Sarker challenged drag optimization for a 3-DOF underwater manipulator in [5]. In this case, hydrodynamic parameters such as added-mass and drag coefficients need to be estimated. However, we consider that it is extremely difficult to determine these underwater manipulator parameters, because the parameters vary greatly with manipulator motion. In fact, Rock and his colleagues pointed out the change of the parameters via some experiments in [6]-[7]. In [7], the drag coefficients that change with 2-DOF manipulator motions were identified. However, they only considered the cases when the second joint was fixed, because it would be difficult to treat dynamics of general motions where the first and the second joints move simultaneously. Namely, it is expected that good results should not be obtained from the model-based approach.

Another method for motion planning is a neural network approach. In that method, parameter estimation is not necessary because the neural network model finds a solution by trial and error. By using the method, drag optimization for an actual 7 -DOF manipulator was attempted in [8] and the minimum energy consumption for a 2-DOF manipulator in simulation was challenged in [9]. However, their methods do not necessarily guarantee convergence of a solution. Even if the solution converges, it may take a large amount of time to solve.

To overcome the above difficulties, we apply a new method to motion planning for underwater manipulators in this paper. The method is based on iterative learning control and time-scale transformation. The original idea of this method is presented in [10]-[11], however, they did not discuss the specific problems of underwater manipulators with very complicated dynamics. We have applied the method described in [12]-[13] to the control of underwater manipulators and have confirmed its effectiveness. In that method, the spatial paths of the manipulators are fixed but the time trajectories (speed patterns) are varied. As a result, dynamics of the underwater manipulators can be easily treated for control, planning and hardware design.

In this paper, we propose a time optimal control method for underwater robot manipulators based on iterative learning control and time-scale transformation. In this method, we consider a class of motions whose spatial paths are fixed but the time trajectories (speed patterns) are changed in order to determine a minimum time motion under the condition on the limits of actuator torque or velocity. By using time-scale changing, the input torque patterns that realize the motions are formed from the basic torque patterns obtained by iterative learning control. Therefore, the proposed method does not need to estimate any physical parameters including the hydrodynamic parameters such as added-mass and drag coefficients.

The effectiveness of our method is verified by experiments in which a 3-DOF underwater manipulator is utilized, and the results are presented in this paper.

0-933957-31-9

II. NONLINEAR TIME-SCALE TRANSFORMATION

A. *Dynamics of underwater manipulators*

Suppose that dynamics on a n-DOF underwater manipulator as shown in Fig.1 is described by

$$M(q(t))\frac{d^2q(t)}{dt^2}+M_A(q(t))\frac{dv(t)}{dt}+\frac{dq(t)}{dt}^T S(q(t))\frac{dq(t)}{dt}$$
$$+v(t)^T S_A(q(t))v(t)+|v(t)|^T D(q(t))v(t)$$
$$+B\dot{q}+H\left\{\mathrm{sgn}.\left(\frac{dq(t)}{dt}\right)\right\}+g(q(t))+b(q(t))=u(t). \qquad (1)$$

The matrices and the vectors in (1) are given as follows:

$q(t)\in R^{n\times 1}$: joint angle coordinate vector,
$q(t)=[q_1(t),q_2(t),\cdots,q_n(t)]$,
$v(t)\in R^{nm\times 1}$: flow velocity vector $v(t)=[v_{11},v_{12},\cdots,v_{nm}]^T$
$v_{i,j}$: is a flow velocity on a segment of an manipulator
$M(q)\in R^{n\times n}$: inertia matrix,
$M_A(q)\in R^{n\times nm}$: added inertia matrix,
$S(q)\in R^{n\times n\times n}$: tensor of Coriolis and centripetal forces,
$S_A(q)\in R^{nm\times n\times nm}$: tensor of hydrodynamic Coriolis and centripetal forces,
$D(q)\in R^{nm\times n\times nm}$: hydrodynamic damping tensor,
$B\in R^{n\times n}$: coefficient matrix of viscous friction,
$H\in R^{n\times n}$: coefficient matrix of Coulomb friction,
$\mathrm{sgn}.(q)\in R^{n\times 1}$: vector given by
$\mathrm{sgn}.(q)=[\mathrm{sgn}.(q_1),\mathrm{sgn}.(q_2),\cdots\mathrm{sgn}.(q_3)]^T$
$\mathrm{sgn}.(q_i)$ $(i=1,\cdots,n)$ means the plus or minus sign of q_i ,
$g.(q)\in R^{n\times 1}$: vector of gravity force,
$b.(q)\in R^{n\times 1}$: vector of buoyancy force,
$u\in R^{n\times 1}$: input torque vector $u(t)=[u^1(t),u^2(t),\cdots,u^n(t)]$

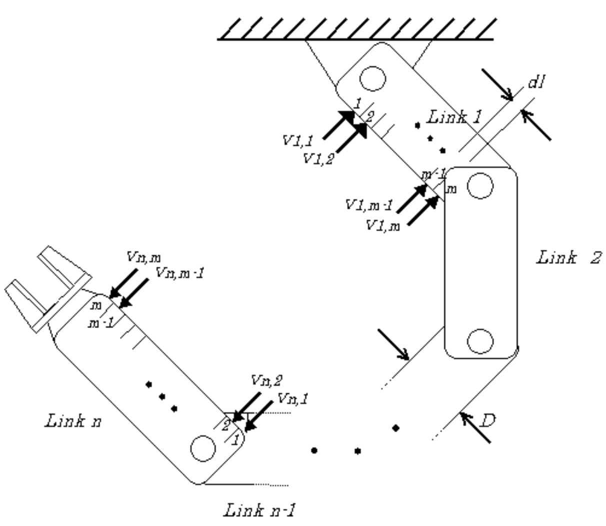


Fig. 1. An underwater manipulator

B. *Formation of The Desired Input Torque Pattern*

In this section, the method of forming an input torque pattern $\boldsymbol{u}_0(t)$ that realizes an arbitrary speed motion $\boldsymbol{q}_0(t)$ is stated ([12]-[13]). In the next section this concept will be utilized to search a minimum time motion.

At first, let us consider four different speed pattern motions $\boldsymbol{q}_1(r_1(t))$, $\boldsymbol{q}_2(r_2(t))$, $\boldsymbol{q}_3(r_3(t))$ and $\boldsymbol{q}_4(r_4(t))$ where $r_1(t)$, $r_2(t)$, $r_3(t)$ and $r_4(t)$ denote appropriate time-scale functions. Since all of the motions have the same space trajectories, we obtain

$$q_1(r_1(t))=q_2(r_2(t))=q_3(r_3(t))=q_4(r_4(t)) \ . \qquad (2)$$

Next, we consider a flow velocity vector $v_{i,j}$ on a segment of the underwater manipulator in Fig.1 and define the displacement $x(t)$ of $v_{i,j}$ as follows:

$$x_{i,j}(t)=\int_0^t v_{i,j}(\tau)d\tau \qquad (3)$$

where we can consider that the initial condition is given by v_{ij}=const. Moreover, we assume that the special trajectory of the flow velocity do not change if time-scale is changed. Under this assumption, we obtain the following same relations as in (2)

$$x_1(r_1(t))=x_2(r_2(t))=x_3(r_3(t))=x_4(r_4(t)) \ . \qquad (4)$$

Without loss of generality, the standard time t is given by $r_1(t)=t$. On time-scale functions, consider the following conditions:

I. For $r_i(t)$ (i=0,…,4),
(a) $r_i(0)=0$, $r_i(T_i)=T_1$
(T_i : Terminal time of control)
(b) $r_i(t)\in C^2$ $[0, T_i]$
(c) $0<dr_i(t)/dt<\infty$ $[0, T_i]$.

II. For $r_2(t)$, $r_3(t)$,
$r_2(t)=k_2t$, $r_3(t)=k_3t$.
where k_i (i=2,3) are positive constant scalars such that
$k_2\neq 1,\ k_3\neq 1,\ k_2\neq k_3$.

III. For $r_4(t)$,

$$1\Big/\frac{dr_4(t)}{dt}\neq 0,\quad \frac{d^2r_4(t)}{dt^2}\neq 0 \qquad t\in[0,T_4].$$

From condition II, $r_1(t)$, $r_2(t)$ and $r_3(t)$ be linearly proportional to one another. In other words, the velocities of the basic motions $\boldsymbol{q}_1$, $\boldsymbol{q}_2$ and $\boldsymbol{q}_3$ have to be linearly proportional to one another. On the basic motion $\boldsymbol{q}_4$ and the arbitrary motion $\boldsymbol{q}_0$, we only consider condition I.

In condition I, (a) means the condition of the terminal time of control for each time-scale function $r_i(t)$(i=0,…,4). This condition is necessary for the same spatial motions $\boldsymbol{q}_i$ (i=1,…,4) satisfying (2). Condition I(b) means that the functions $dr_i(t)/dt$ are continuously differentiable and condition I(c) means that $r_i(t)$ are monotone increasing functions. Here, it should be noted that the motion $\boldsymbol{q}_i(r_i)$ is faster than the motion $\boldsymbol{q}_1(t)$ if $dr_i(t)/dt < 1$ and $\boldsymbol{q}_i(r_i)$ is slower than $\boldsymbol{q}_1(t)$ if $dr_i(t)/dt > 1$.

When each motion $\boldsymbol{q}_i(r_i)$ satisfies (2), (3), (4) and the above conditions, an input torque pattern $\boldsymbol{u}_0(r_0)$ that realizes an arbitrary speed motion $\boldsymbol{q}_0(r_0)$ is calculated by

$$u_o(r_o(t)) = [I, I, I]A^{*-1}u^* \tag{5}$$

(see Appendix). The above matrix and the vector are given as follows:

$$A^* = \begin{bmatrix} \alpha_1^2 I & \alpha_1 I & I \\ (\alpha_1/k_2)^2 I & (\alpha_1/k_2)I & I \\ (\alpha_1/k_3)^2 I & (\alpha_1/k_3)I & I \end{bmatrix},$$

$$u^* = \begin{bmatrix} u_1(t) + \alpha_1^2\beta_1 f_{14}(t) \\ u_2(k_2 t) + (\alpha_1^2\beta_1/k_2^2) f_{14}(t) \\ u_3(k_3 t) + (\alpha_1^2\beta_1/k_3^2) f_{14}(t) \end{bmatrix} \tag{6}$$

where

$$\alpha_1 = 1\Big/\frac{dt}{dr_0}, \quad \beta_1 = \frac{d^2 t}{dr_0^2},$$

$$\begin{aligned} f_{14}(t) = {} & \frac{(a_4 k_2 - 1)(a_4 k_3 - 1)}{(k_3 - 1)(k_3 - 1)a_4^3 b_4} u_1(t) \\ & + \frac{(a_4 - 1)(a_4 k_3 - 1)k_2^2}{(k_2 - 1)(k_3 - k_2)a_4^3 b_4} u_2(k_2 t) \\ & + \frac{(a_4 - 1)(a_4 k_2 - 1)k_3^2}{(k_3 - 1)(k_3 - k_2)a_4^3 b_4} u_3(k_3 t) \\ & - \frac{1}{a_4^3 b_4} u_4(r_2(t)), \end{aligned}$$

$$a_4 = 1\Big/\frac{dr_4(t)}{dt}, \quad b_4 = \frac{d^2 r_4(t)}{dt^2}.$$

Namely, the desired input pattern $\boldsymbol{u}_0$ that can realize an arbitrary speed motion is formed from four basic torque patterns $\boldsymbol{u}_1(t)$, … , $\boldsymbol{u}_4(t)$ by using time-scale transformation. This method is very useful from a practical viewpoint because it is not necessary to estimate any physical parameters including hydrodynamic parameters.

Theoretically, there aren't any problems in obtaining the input torque pattern if the basic motions $\boldsymbol{q}_i$ (i=1,…,4) satisfy the above conditions on $r_i(t)$. However, the effect of the error of the input torque patterns obtained by iterative learning control should be considered. In (5), the inverse matrix of $\boldsymbol{A}^*$ is used for forming the torque pattern $\boldsymbol{u}_0$. In addition, the inverse matrix of $\boldsymbol{A}$ in (22) (see Appendix) is used for calculating $\boldsymbol{f}_{14}(t)$. Therefore, the accuracy of the torque pattern $\boldsymbol{u}_0$ becomes worse in the neighborhood of any singularity of the matrices. As a result, we can consider additional conditions as follows:

* Among 1, k_2 and k_3, the difference of any two numbers is large enough.

* α_1, a_4 and b_4, are large enough values during the motions.

III. TIME-OPTIMAL CONTROL METHOD

A. Problem of minimum-time motion

In this section, we propose a motion planning method that realizes a minimum-time motion solution for underwater robot manipulators without parameter estimation. The purposes of this section are to find out a minimum-time motion $\boldsymbol{q}_{opt}(t)$ under the limits of actuator torque as follows:

$$u_{\min}^j \le u_{opt}^j \le u_{\max}^j \quad (j = 1, \cdots, n) \tag{7}$$

and to form the input torque pattern $\boldsymbol{u}_{opt}(t)$ that realizes the motion $\boldsymbol{q}_{opt}(t)$. In (7), $u_{\min}^j$ and $u_{\max}^j$ are minimum and maximum bounds on the j-th element of the input pattern $\boldsymbol{u}_{opt}(t)$.

In this paper, we only treat the limits of actuator torque for motion planning, but this method can be also applied to motion planning under the limits of angular velocity of the manipulators.

B. Search for minimum-time motion

In the previous section, the input torque pattern $\boldsymbol{u}_0(t)$ with an arbitrary time-scale trajectory (arbitrary speed pattern) was obtained by the four basic torque patterns $\boldsymbol{u}_i(t)$. By using this method, we find a minimum-time motion $\boldsymbol{q}_{opt}(t)$ and form the input torque pattern $\boldsymbol{u}_{opt}(t)$ under condition (7) in this section.

Considering the definitions of $\boldsymbol{A}^*$ and $\boldsymbol{u}^*$, Equation (6) is rewritten as

$$\begin{aligned} u_o(r_o(t)) &= [I, I, I]A^{*-1}u^* \\ &= -\frac{f_{14}(t)}{\alpha_1^3}(-\alpha_1^3\beta_1) \\ &\quad + \frac{(\alpha_1 - k_2)(\alpha_1 - k_3)}{(1 - k_2)(1 - k_3)\alpha_1^2} u_1(t) \\ &\quad + \frac{(\alpha_1 - k_3)(\alpha_1 - 1)k_2^2}{(k_2 - k_3)(k_2 - 1)\alpha_1^2} u_2(k_2 t) \\ &\quad + \frac{(\alpha_1 - 1)(\alpha_1 - k_2)k_3^2}{(k_3 - 1)(k_3 - k_2)\alpha_1^2} u_3(k_3 t) \end{aligned} \tag{8}$$

where scalars k_2, k_3 and vectors $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{u}_3$ and $\boldsymbol{f}_{14}$ are already known from the basic motions. Moreover, substituting the following equation

$$\beta_1 = \frac{d^2 t}{dr_0^2} = \frac{dt}{dr_0}\frac{d(dt/dr_0)}{dt} = \frac{1}{\alpha_1}\frac{d\alpha_1^{-1}}{dt} = \frac{1}{\alpha_1^3}\frac{d\alpha_1}{dt} \tag{9}$$

into (8) yields

$$u_0(r_0) = \mathrm{X}_1(t, \alpha_1)\frac{d\alpha_1}{dt} + \mathrm{X}_2(t, \alpha_1)$$

where

$$\mathrm{X}_1(t, \alpha_1) = -\frac{f_{14}(t)}{\alpha_1^3},$$

$$\begin{aligned} \mathrm{X}_2(t, \alpha_1) = {} & \frac{(\alpha_1 - k_2)(\alpha_1 - k_3)}{(1 - k_2)(1 - k_3)\alpha_1^2} u_1(t) \\ & + \frac{(\alpha_1 - k_3)(\alpha_1 - 1)k_2^2}{(k_2 - k_3)(k_2 - 1)\alpha_1^2} u_2(k_2 t) \\ & + \frac{(\alpha_1 - 1)(\alpha_1 - k_2)k_3^2}{(k_3 - 1)(k_3 - k_2)\alpha_1^2} u_3(k_3 t). \end{aligned} \tag{10}$$

Namely, each element of $\boldsymbol{u}_0(r_0)$ is a linear function of $d\alpha_1/dt$.

Now, $\boldsymbol{u}_0(r_0)$ is given by expanding or shrinking the time-scale of $\boldsymbol{u}_0(t)$. Therefore, the time-scale function r_0 is expanded or shrunk so that the input torque pattern $\boldsymbol{u}_0(r_0)$ satisfies the limits of actuator torque (7) as follows:

$$u^j_{\min} \le X^j_1(t,\alpha_1)\frac{d\alpha_1}{dt} + X^j_2(t,\alpha_1) \le u^j_{\max} \quad (j=1,\cdots,n) \tag{11}$$

where $X^j_i(t,\alpha_1)(i=1,2)$ are the elements of $X_i(t,\alpha_1)$.

Since the elements $X^j_i(t,\alpha_1)$ are formed by the input torque patterns obtained from iterative learning control without parameter estimation, this method is very effective from the practical point of view.

Hereafter, we analyze the paths of $d\alpha_1/dt$ in (11) to obtain the minimum time motion on the $t-\alpha_1$ phase plane.

From (11), the range of $d\alpha_1/dt$ on the j-th joint is

$$L^j(t,\alpha_1) \le \frac{d\alpha_1}{dt} \le H^j(t,\alpha_1) \quad (j=1,\cdots,n) \tag{12}$$

where we define

$$L^j(t,\alpha_1) = \begin{cases} (u^j_{\min} - X^j_2)/X^j_1 & (X^j_1 > 0) \\ (u^j_{\max} - X^j_2)/X^j_1 & (X^j_1 < 0) \end{cases},$$

$$H^j(t,\alpha_1) = \begin{cases} (u^j_{\max} - X^j_2)/X^j_1 & (X^j_1 > 0) \\ (u^j_{\min} - X^j_2)/X^j_1 & (X^j_1 < 0) \end{cases}.$$

These inequalities give upper limit and lower limit of $d\alpha_1/dt$ at the j-th joint. In the case $X^j_1 = 0$, equation (11) is satisfied over the whole range of $d\alpha_1/dt$ when $u^j_{\min} < X^j_2 < u^j_{\max}$. On the other hand, when $u^j_{\max} < X^j_2$ or $X^j_2 < u^j_{\min}$, the inequality is not satisfied over the whole range of $d\alpha_1/dt$.

Let us define the maximum value of $L^j(t,\alpha_1)$ as $ML(t,\alpha_1)$ and the minimum value of $H^j(t,\alpha_1)$ as $MH(t,\alpha_1)$ among all the joints. When $MH(t,\alpha_1)$ is larger than $ML(t,\alpha_1)$, we obtain an inequality as follows:

$$ML(t,\alpha_1) \le \frac{d\alpha_1}{dt} \le MH(t,\alpha_1). \tag{13}$$

This inequality represents an enabled bound of direction at any point on the $t-\alpha_1$ phase plane. Fig.2 illustrates an example of the $t-\alpha_1$ phase plane. On the contrary, if the inequality

$$ML(t,\alpha_1) > MH(t,\alpha_1) \tag{14}$$

holds on the $t-\alpha_1$ plane, no direction is enabled.

From the conditions on time-scale function $r_0(t)$, α_1 must satisfy $\alpha_1 \in C^1$ and $\alpha_1 > 0$. On the phase plane, the motion of the manipulators is faster as the value of α_1 becomes smaller.

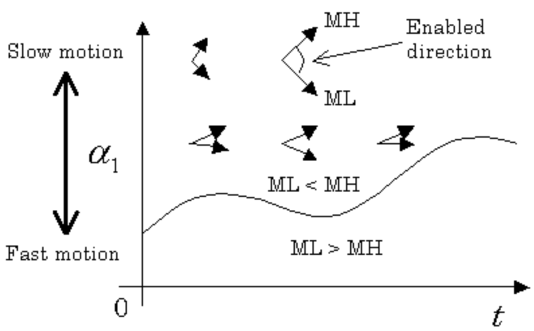


Fig. 2 An example of the $t-\alpha_1$ phase plane

Now, the time-optimal motion $\boldsymbol{q}_{\mathrm{opt}}(t)$ is determined from a path of α_1 which surrounds a minimum region on the phase plane, because

$$\int_0^t \alpha_1 d\tau = \int_0^t \frac{dr_0(\tau)}{d\tau} d\tau = r_0(t) - r_0(0) = r_0(t). \tag{15}$$

Since the time-scale function $r_{\mathrm{opt}}(t)$ is obtained by integration of α_1, $\boldsymbol{q}_{\mathrm{opt}}(t)$ is obtained from $\boldsymbol{q}_1(t)$ by path planning on the phase plane. Namely, minimum-time motion $\boldsymbol{q}_{\mathrm{opt}}(t)$ with torque pattern bounded by (11) is formed from the four basic motions $\boldsymbol{q}_i(t)$ and the ideal input torque patterns $\boldsymbol{u}_i(t)$ (i=1,…,4) without parameter estimation.

IV. EXPERIMENTAL RESULT

To investigate the performance of our time optimal control method, an experiment was conducted by using a 3-DOF underwater manipulator as shown in Fig. 2. Table I shows the parameters of the manipulator. In this experiment, input electric currents to the actuators of the manipulator are considered to be the input patterns.

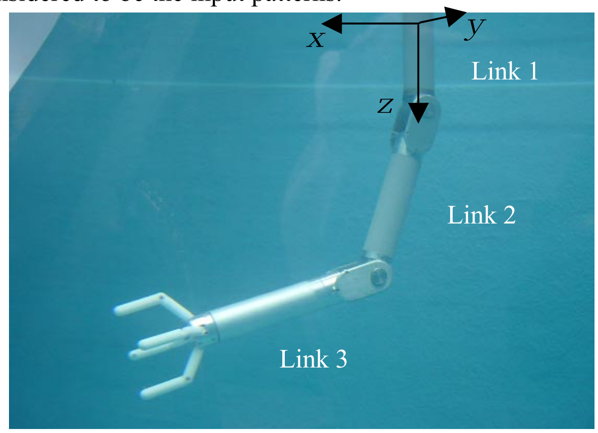


Fig. 3 Three degree-of-freedom underwater manipulator

TABLE I
PARAMETERS OF THE UNDERWATER MANIPULATOR

	Link 1	Link 2	Link 3
Length [m]	0.328	0.216	0.200
Width [m]	0.050	0.050	0.050
Mass [kg]	1.69	1.56	1.56
Actuator Power [W]	DC 11	DC 11	DC 11
Gear ratio	1:50	1:50	1:50

At first, four basic motions $\boldsymbol{q}_i(r_i)$ have to be decided in order to obtain the basic torque patterns $\boldsymbol{u}_i(r_i)$. In this experiment, $\boldsymbol{q}_1(t)$ is defined as the joint angle coordinate vector when the position vector $\boldsymbol{p}_1(t)=[x_d, y_d, z_d]$ of the end-effector was set as follows:

$$p_1(t) = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} = \begin{bmatrix} 0.35 \\ -0.25 + 0.5\sin\theta(t) \\ 0.75 \end{bmatrix},$$

$$\theta(t) = \frac{\pi}{2}\left(\frac{6}{T_1^5}t^5 - \frac{15}{T_1^4}t^4 + \frac{10}{T_1^3}t^3\right), \quad T_1 = 1.8[s]. \tag{16}$$

The motion $\boldsymbol{q}_1(t)$ corresponding to $\boldsymbol{p}_1(t)$ is solved by inverse kinematics. The time-scale functions $r_i(t)$ and the terminal time of control T_i for each motions $\boldsymbol{p}_i(r_i)$ (i=2,3,4) are set as

$$\begin{aligned} &r_2(t) = 3t && T_2 = 5.4[\mathrm{sec}], \\ &r_3(t) = 1.5t && T_3 = 2.7[\mathrm{sec}], \\ &r_4(t) = (5t^2 + 10)/5 && T_4 = 2.45[\mathrm{sec}]. \end{aligned} \tag{17}$$

The motions $\boldsymbol{q}_1(t)$, $\boldsymbol{q}_2(t)$ and $\boldsymbol{q}_3(t)$ was calculated by (2) and (17). The torque patterns $\boldsymbol{u}_i(r_i)$ (i=1,…,4) that realize the basic motions $\boldsymbol{q}_i(r_i)$ are formed by PD-type iterative learning control in this experiment. The iteration number was 15 for the each motion $\boldsymbol{q}_i(r_i)$. If we employ a performance index as follows:

$$\sqrt{\frac{1}{T_i}\int_0^{T_i} e_i(t)^2\,dt} \quad [mm] \tag{18}$$

where $e = \sqrt{(x_d - x)^2 + (y_d - y)^2 + (z_d - z)^2}$, the experimental results of the motions $\boldsymbol{q}_1(r_1)$, $\boldsymbol{q}_2(r_2)$, $\boldsymbol{q}_3(r_3)$ and $\boldsymbol{q}_4(r_4)$ were 2.9 [mm], 2.5[mm], 2.1[mm] and 2.3[mm], respectively.

Next, the time optimal motion is planned under the conditions of the limits of the input pattern as follows:

$$-1.0 < u_{opt}(r_{opt}) < 1.0 \quad [\mathrm{A}] \tag{19}$$

Under this condition, the $t-\alpha_1$ phase plane is analyzed to find out the path of α_1. Here, we do not take into account computation time and adopt a simple iterative counting to determine the path. At first, the $t-\alpha_1$ plane is divided into the small segments $\Delta\alpha_1$=0.002, Δt=0.0025 and α_1 that satisfies (13) is searched by the iterative counting on the plane. In this experiment, the path of α_1 is obtained as shown in Fig.4. After finding out α_1, integration of α_1 leads to the time-scale function $r_0(t)$ that realizes the minimum-time motion $\boldsymbol{q}_{opt}(t)$. From this experimental result, the terminal time of control for the motion $\boldsymbol{q}_{opt}(t)$ is determined on 1.28[sec]. This time is shorter than the other times of the basic motion patterns $\boldsymbol{q}_i(r_i)$ (i=1,…,4).

We confirm that the input pattern $\boldsymbol{u}_{opt}(t)$ are bounded by the limits (19) as shown in Fig.5. As the results, the input pattern of the second joint is bounded by the limitation.

To check the control performance with the input pattern, trajectory control is performed. Feedback control with $\boldsymbol{u}_{opt}(t)$ and only feedback control are adopted for the sake of comparison. Fig.6 illustrates the result of the position errors of end-effector. As shown in Fig.6, the position error for only feedback control is large between 0.7[s] and 0.9[s] when the manipulator motion is fast. On the contrary, the position error for feedback control with $\boldsymbol{u}_{opt}(t)$ is small during the fast motion. The error norms defined by (18) are 19.0[mm] and 94.4[mm], respectively. From the results, the input pattern formed by the proposed method can realize more accurate control of the underwater manipulator. The effectiveness of our method was verified by this experiment.

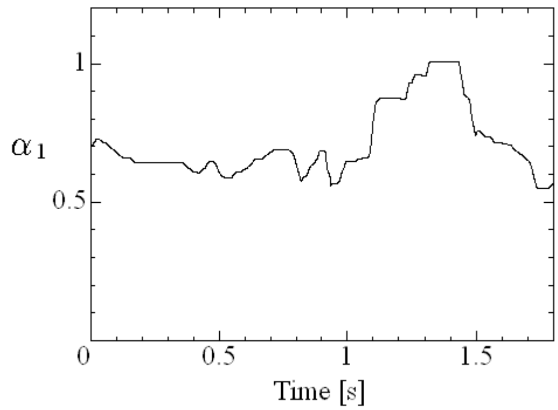
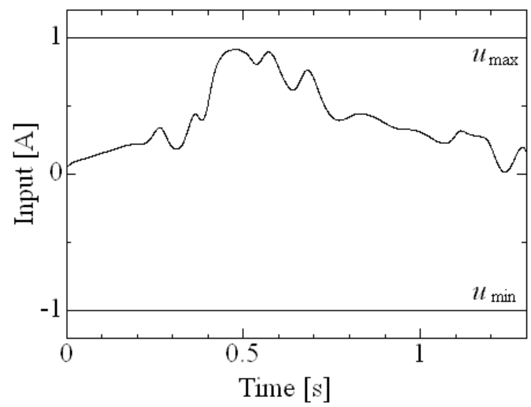
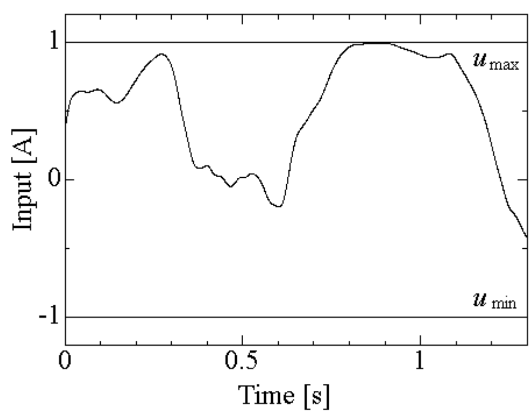


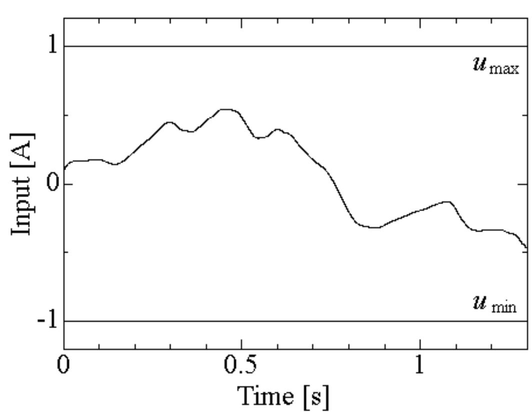
Fig. 4 $t-\alpha_1$ phase plane in this experiment



(a) Joint 1



(b) Joint 2



(c) Joint 3

Fig.5 Input pattern for minimum-time motion

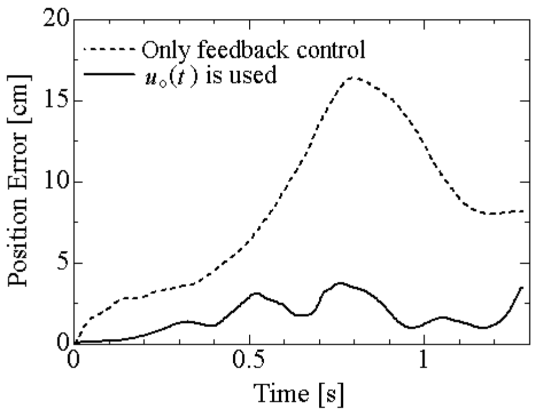


Fig. 6 Comparison of the position errors

V. CONCLUSION

We proposed the time optical control method for underwater manipulators based on iterative learning control and time-scale transformation. In this method, the input torque pattern that realizes the minimum-time motion can be formed from the basic torque patterns by using time-scale changing. For motion planning, it is not necessary to estimate any physical parameters including hydrodynamic parameters. Therefore, this method is very practical for control of underwater manipulators that have complicated dynamics. The effectiveness of the proposed method was demonstrated through experiments in which a 3-DOF manipulator was used. As a result, the input pattern formed by the proposed method can realize more accurate control of the underwater manipulator. However, the position error is not negligible even when the proposed method is used. The reason may be that manipulator dynamics defined by (1) changes when the manipulator motion is highly fast. Therefore, we will reconsider manipulator dynamics during fast motion in order to realize more accurate control.

REFERENCES

[1] I. Scholberg, and T. Fossen, "Modeling and control of underwater vehicle-manipulator system," *Proc. of the 3rd Conference on MCMC'94*, pp. 45-57, 1994.

[2] C. Canudas-de-Wit, E.O. Diaz, and M. Perrier, "Nonlinear control of an underwater vehicle/manipulator with composite dynamics," *Proc. of the IEEE ICRA*, pp. 452-457.

[3] G. Antonelli, and F. Caccavale, S. Chiaverini, and L. Villani, "Tracking control for underwater vehicle-manipulator systems with velocity estimation," IEEE Journal of Oceanic Engineering, Vol. 25, No. 3, pp.399-413, 2000.

[4] N. Kato, and D.M. Lane, "Coordinated control of multiple manipulators in underwater robots," *Proc. of the IEEE ICRA,* pp.2505-2510, 1996.

[5] N. Sarkar and T.K. Podder, "Motion coordination of underwater vehicle-manipulator systems subject to drag optimization," *Proc. of the IEEE International Conference on Robotics and Automation,* pp. 387-391, 1999.

[6] T.W. McLain, and S.M Rock, "Experiments in the coordinated control of underwater manipulator and vehicle control," *Proc. of OCEANS '95 MTS/IEEE*, pp. 1208-1215, 1995.

[7] K.N. Leabourne, and S.M. Rock, "Model development of an underwater manipulator for coordinated arm-vehicle control," *Proc. of OCEANS '98 MTS/IEEE,* Vol. 2, pp. 941-946, 1998.

[8] S. Ishibashi, E. Shimizu, and M. Ito, "The motion planning for underwater manipulators depend on genetic algorithm," *Proc. of IFAC 15th world congress*, pp. 305-310, 2002.

[9] E. Shintaku, "Minimum energy trajectory for underwater manipulator and its simple planning method by using a genetic algorithm," *Advanced Robotics,* vol. 13, No. 2, pp.115-133, 1999.

[10] S. kawamura, and N. Fukao, "Use of feedforward input patterns obtained through learning control," *Proc. of the 2nd asian control conference*, pp.255-259, 1997.

[11] S. Kawamura, N. Fukao, and H. Ichii, "Planning and control of robot mation based on time-scale transformation and iterative learning control," *Springer-Verlag London Limited 2000*, pp.214-220, 2002.

[12] S. Kawamura, and N. Sakagami, "Analysis on dynamics of underwater robot manipulators basing on iterative learning control," *Proc. of the IEEE ICRA,* pp. 1088-1094, 2002.

[13] N. Sakagami, and S. Kawamura, "Analysis on dynamics of 3-dof underwater robot manipulators based on time-scale transformation," *2002 Japan-USA Symposium on Flexible Automation,* vol. 3, pp.1139-1145, 2002.

Appendix

Equation (1) can hold even though time t is replaced with the functions $r_i(t)$ as follows:

$$
\begin{aligned}
&M(q_i(r_i))\frac{d^2 q_i(r_i)}{dr_i^2} + M_A(q_i(r_i))\frac{dv_i(r_i)}{dr_i} + \frac{dq_i(r_i)^T}{dr_i} S(q_i(r_i))\frac{dq_i(r_i)}{dr_i} \\
&+ v_i(r_i)^T S_A(q_i(r_i))v_i(r_i) + |v_i(r_i)|^T D(q_i(r_i))v_i(r_i) \\
&+ B\dot{q}_i + H\left\{\operatorname{sgn}\left(\frac{dq_i(r_i)}{dr_i}\right)\right\} + g(q_i(r_i)) + b(q_i(r_i)) = u_i(r_i)
\end{aligned}
$$

$$(i = 0,2,3,4). \qquad (19)$$

Now, the following equations

$$
\begin{aligned}
&\frac{dq_i(r_i)}{dr_i} = a_i \frac{dq_1(t)}{dt}, \\
&\frac{d^2 q_i(r_i)}{dr_i^2} = a_i^2 \frac{d^2 q_1(t)}{dt^2} - a_i^3 b_i \frac{dq_1(t)}{dt}, \\
&v_i(r_i) = a_i v_1(t), \\
&\frac{dv_i(r_i)}{dr_i^2} = a_i^2 \frac{dv_1(t)}{dt} - a_i^3 b_i v_1(t)
\end{aligned}
\qquad (20)
$$

where

$$a_i = \frac{dr_i}{dt}, \qquad b_i = \frac{d^2 r_i(t)}{dt^2}$$

are satisfied from (2) and (4). Substituting (2), (4) and (20) into (19) yields

$$-a_i^3 b_i f_{14}(t) + a_i^2 f_{13}(t) + a_i f_{12}(t) + f_{13}(t) = u_i(r_i) \qquad (21)$$

where

$$f_{11}(t) = M(q_1(t))\frac{d^2 q_1(t)}{dt^2} + M_A(q_1(t))\frac{dv_1(t)}{dt} + \frac{dq_1(t)^T}{dt} S(q_1(t))\frac{dq_1(t)}{dt} + v_1(t)^T S_A(q_1(t))v_1(t) + |v_1(t)|^T D(q_1(t))v_1(t),$$

$$f_{12} = B\frac{dq_1(t)}{dt},$$

$$f_{13} = H\left\{\mathrm{sgn}\left(\frac{dq_1(t)}{dt}\right)\right\} + g(q_1(t)) + b(q_1(t)),$$

$$f_{14} = M(q_1(t))\frac{dq_1(t)}{dt}.$$

In practice, the number of $\boldsymbol{u}_i(\boldsymbol{r}_i)$ data is different if the time-scale is changed. Therefore, it is necessary to complete the number of data before time-scale transformation. The number N of data in time-scale $r_i(t)=t$ is given by

$$N = T_i / \Delta t$$

where T_1 is the terminal time, Δt is the sampling time. In the same manner, the number of data in another time-scale function r_i(t) equals the number N when the terminal time T_i of each time-scale divided by $\Delta r_i(t)(= r_i(t+\Delta t) - r_i(t))$.

By using the matrix form of (21), we obtain

$$Af = u \qquad (22)$$

where

$$A = \left[\begin{array}{ccc|c} & & & -a_1^2 b_1 I \\ & B & & -a_2^2 b_2 I \\ & & & -a_3^2 b_3 I \\ \hline a_4^2 I & a_4 I & I & -a_4^2 b_4 I \end{array}\right],$$

$$B = \begin{bmatrix} I & I & I \\ a_2^2 I & a_2 I & I \\ a_3^2 I & a_3 I & I \end{bmatrix},$$

$$f = \begin{bmatrix} f_{11}(t) \\ f_{12}(t) \\ f_{13}(t) \\ f_{14}(t) \end{bmatrix}, \qquad u = \begin{bmatrix} u_1(t) \\ u_2(k_2 t) \\ u_3(k_3 t) \\ u_4(r_4) \end{bmatrix}.$$

From $r_1(t)=t$ and the second condition of time-scale functions, $b_1= b_2= b_3=0$. Since the matrix **B** is a block Vandermonde matrix and neither the character a_4 nor b_4 becomes zero, the matrix **A** is always nonsingular. Therefore, we can estimate the term $f_{14}(t)$ in (22) as follows:

$$\begin{aligned} f_{14}(t) &= [0,0,0,I]A^{-1}u \\ &= \frac{(a_4 k_2 - 1)(a_4 k_3 - 1)}{(k_2 - 1)(k_3 - 1)a_4^2 b_4} u_1(t) \\ &+ \frac{(a_4 - 1)(a_4 k_3 - 1)k_2^2}{(k_2 - 1)(k_3 - k_2)a_4^2 b_4} u_2(k_2 t) \\ &+ \frac{(a_4 - 1)(a_4 k_2 - 1)k_3^2}{(k_3 - 1)(k_3 - k_2)a_4^2 b_4} u_3(k_3 t) \\ &- \frac{1}{a_4^3 b_4} u_4(r_4). \end{aligned} \qquad (23)$$

Now, we form an input torque pattern that realizes a new motion by using $\boldsymbol{u}_1(t)$, $\boldsymbol{u}_2(r_2)$, $\boldsymbol{u}_3(r_3)$ and $\boldsymbol{f}_{14}(t)$. If we consider r_i(t) as a function of r_0(t) and

$$\alpha_i = 1\Big/\frac{dr_i}{dr_0}, \beta_i = d^2 r_i / dr_0^2$$

in the same manner as (21), we can obtain

$$\alpha_i^2 f_{03}(t) + \alpha_i f_{02}(t) + f_{03}(t) = u_i(r_i) + a_i^3 b_i f_{04}(t) \qquad \text{(i=1,…,3).} \quad (24)$$

where

$$f_{11}(t) = M(q_0(r_0))\frac{d^2 q_0(r_0)}{dr_0^{\,2}} + M_A(q_0(r_0))\frac{dv_0(r_0)}{dr_0} + \frac{dq_0(r_0)^T}{dr_0} S(q_1(r_0))\frac{dq_0(r_0)}{dr_0} + v_0(r_0)^T S_A(q_0(r_0))v_0(r_0) + |v_0(r_0)|^T D(q_0(r_0))v_0(r_0),$$

$$f_{12} = B\frac{dq_0(r_0)}{dr_0},$$

$$f_{13} = H\left\{\mathrm{sgn}\left(\frac{dq_0(r_0)}{dr_0}\right)\right\} + g(q_0(r_0)) + b(q_0(r_0)),$$

$$f_{14} = M(q_0(r_0))\frac{dq_0(r_0)}{dr_0}.$$

Since

$$\alpha_2 = dr_0/d(k_2 t). = \alpha_1 / k_2, \quad \alpha_3 = dr_0/d(k_3 t). = \alpha_1 / k_3,$$
$$\beta_2 = d^2(k_2 t)/dr_0^2. = k_2 / \beta_1, \quad \beta_3 = d^2(k_3 t)/dr_0^2. = k_3 / \beta_1,$$

and

$$\begin{aligned} f_{11}(t) &= M(q_0(r_0))\frac{dq_0(r_0)}{dr_0} + M_A(q_0(r_0))v_0 \\ &= M(q_1(t))\frac{dq_1(t)}{dt}\frac{dt}{dr_0} + M_A(q_1(t))\frac{dx_1(t)}{dt}\frac{dt}{dr_0}, \end{aligned}$$

by using matrix form, we can rewrite (24) as

$$A^* \begin{bmatrix} f_{01}(r_0) \\ f_{02}(r_0) \\ f_{03}(r_0) \end{bmatrix} = u^*.$$

Since the matrix $\mathbf{A}^*$ is a block Vandermonde matrix, the matrix $\mathbf{A}^*$ is always nonsingular. Therefore, we finally obtain the result as follows:

$$\begin{aligned} &[I \quad I \quad I]A^{*-1}u^* \\ &= f_{01}(r_0) + f_{02}(r_0) + f_{03}(r_0) \\ &= u_0(r_0). \end{aligned}$$