'91 KACC 1991. 10. 22~24

A STUDY ON THE RELATION BETWEEN CLOSED-FORM DESCRIPTION AND RECURSIVE-FORM REALIZATION OF ADAPTIVE CONTROL OF MANIPULATORS *

Takeshi KUBOZONO Masaki YAMAKITA and Katsuhisa FURUTA

Department of Control Engineering
Tokyo Institute of Technology
2-12-1,Oh-Okayama,Meguro-ku
Tokyo 152
JAPAN

ABSTRACT

Recently, a number of papers on adaptive control scheme of manipulators are proposed. Slotine and Li [1] showed an adaptive control scheme for robot manipulator. The controller was described in closed form. And later Niemeyer and Slotine discussed about a computational implementation of the controller in recursive form [2]. Walker proposed another adaptive control scheme which can be implemented by a recursive-form controller [4]. Closed-form description is used for the analysis or design of adaptive control systems while recursive-form realization is used for implementation of the controller. The relation between the closed-form realization and the recursive-form one seems to be inadequately referred. Hence, it makes sense to consider the relation between the closed-form description and the recursive-form one. In this paper, first, we make a simple derivation of an closed-form dynamics description of a robot arm from its recursive-form description. And then we derive the closed-form realization of Walker's scheme applied to manipulators having no kinematic loop. We clarify the difference between the Walker's scheme and Slotine's and evaluate the convergence under the controllers.

1 INTRODUCTION

A difficulty in considering a robot arm control system is the difference between the closed-form description of the system, which is effective for analysis or design of the system, and recursive-form one for the computational implementation. The closed-form description has a single equation,which is usually obtained from Lagrangian formulation and which is very hard to calculate completely in spite of the resultant simple structure, particularly when the degree-of-freedom of the robot arm is large. The recursive-form description has several iterative equations with respect to each link which is obtained directly from the Newton-Euler dynamic equation. It can be easily observed that closed-form description is obtained by composing the recursive-form equations.

Recently, to make the equations be simple, the spatial notation is used to describe the dynamics of the robot arm recursively [5]. We also use spatial notation in this paper to describe the velocity, inertia force and torque etc. A simple introduction of the spatial notation is given in Appendix.

In section 2, we show the robot arm dynamics in the recursive-form under the spatial notation and then reduce them into the closed-form. In section 3, we summarize some recent works on adaptive control of manipulators. In section 4, we derived the closed form of Walker's scheme and analyze it.

2 THE ROBOT ARM DYNAMICS

In this paper, we consider the robot arms which has n degree-of-freedom and no kinematic chain. The robot arm has $n+1$ links numbered from 0 (base link) to n (end effector) and n joints numbered so that i-th joint connects $(i-1)$-th link and i-th link.

2.1 CLOSED FORM AND RECURSIVE FORM

As discussed in a number of literature, the dynamic equation of a robot arm is represented in following form.

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + G(\mathbf{q}) = \tau - \eta(\dot{\mathbf{q}}) \tag{1}$$

where

$\mathbf{q} \in R^n$	:	vector of joint variables
$H(\mathbf{q}) \in R^{n\times n}$	:	positive definite pseudo inertia matrix
$C(\mathbf{q},\dot{\mathbf{q}}) \in R^{n\times n}$	:	Coriolis and centrifugal terms
$G(\mathbf{q}) \in R^{n\times 1}$	:	gravity terms
$\eta(\dot{\mathbf{q}}) \in R^{n\times 1}$	:	viscous friction terms

The matrix $C(\mathbf{q},\dot{\mathbf{q}})$ is not determined uniquely. But for adaptive control $C(\mathbf{q},\dot{\mathbf{q}})$ is taken so that $\dot{H}-2C$ is skew-symmetric; i.e.

$$\mathbf{x}^T(\dot{H}-2C)\mathbf{x} = 0 \quad \text{for } \forall \mathbf{x} \in R^{n\times 1} \tag{2}$$

The recursive representation of the equation (1) in spatial notation is as follows. [5]

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \mathbf{s}_i\dot{q}_i \quad (\mathbf{v}_0 = 0) \tag{3}$$

$$\dot{\mathbf{v}}_i = \dot{\mathbf{v}}_{i-1} + \mathbf{v}_i \times \mathbf{s}_i q_i + \mathbf{s}_i\ddot{q}_i \quad (\mathbf{a}_0 = 0) \tag{4}$$

$$\mathbf{f}_i = \mathbf{f}_{i+1} + I_i\mathbf{a}_i + \mathbf{v}_i \times I_i\mathbf{v}_i + I_i\mathbf{g} \tag{5}$$

$$\tau_i = \mathbf{s}_i'\mathbf{f}_i + \eta_i \tag{6}$$

*Send all correspondence to Prof. K.Furuta.Furuta Laboratory.Department of Control Engineering.Tokyo Institute of Technology.2-12-1.Oh-Okayama.Meguro-ku.Tokyo. 152.JAPAN

where

q_i	:	joint variable of joint i
$\mathbf{s}_i(\mathbf{q})$	:	spatial vector representing i-th joint axis
$I_i(\mathbf{q})$	:	spatial inertia of link i
$\mathbf{v}_i$	:	spatial velocity of link i
$\mathbf{f}_i$	:	spatial force acting on the joint i
τ_i	:	torque or force supplied by i-th actuator
$\mathbf{g}$	:	spatial gravitational acceleration $(= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -9.8 \end{bmatrix}^T)$
η_i	:	spatial force representing viscous friction of joint i

All the spatial vectors shown above is represented with respect to the base (link 0) coordinate frame. Note that there are several extraordinary operators such as '′' (spatial transpose) and '×' (spatial cross operator) in the above notation. For more details,see Appendix.

2.2 DERIVATION OF RELATION

Now, let us analyze the relation between the closed form (1) and recursive form (3)-(6). We assemble the equation (3)-(6) to derive the same structure as equation (1). The equation (5) has to be modified as follows so that the resultant matrix corresponding to $C(\mathbf{q},\dot{\mathbf{q}})$ satisfies the equation (2).

$$\mathbf{f}_i = \mathbf{f}_{i+1} + I_i\mathbf{a}_i + \frac{1}{2}\mathbf{v}_i \times I_i\mathbf{v}_i - \frac{1}{2}(I_i\mathbf{v}_i)\times\mathbf{v}_i - \frac{1}{2}I_i\mathbf{v}_i\times\mathbf{v}_i + I_i\mathbf{g} \quad (7)$$

Note that the cross operator × is not entirely equivalent to the ordinary vector product, so the third term and the forth term of the left-hand side of (7) make no cancellation. Note also that the fifth term of the right-hand side of the equation (7) turns out to be zero.

From equation (3) we obtain

$$\mathbf{v} = J(\mathbf{q})\dot{\mathbf{q}} \quad (8)$$

where

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{bmatrix} \in R^{6n\times 1},\ J(\mathbf{q}) = \begin{bmatrix} \mathbf{s}_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \mathbf{s}_1 & \cdots & \cdots & \mathbf{s}_n \end{bmatrix} \in R^{6n\times n}$$

From equation (4) we obtain

$$\dot{\mathbf{v}} = J(\mathbf{q})\ddot{\mathbf{q}} + V_s(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \quad (9)$$

where

$$V_s(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} \mathbf{v}_1\times\mathbf{s}_1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \mathbf{v}_1\times\mathbf{s}_1 & \cdots & \cdots & \mathbf{v}_n\times\mathbf{s}_n \end{bmatrix} \in R^{6n\times n}$$

Note that $V_s = \dot{J}$. Using equation (6),(7),(8), and (9) we get

$$\begin{aligned} \tau &= J_t(\mathbf{q})\{I(\mathbf{q})J(\mathbf{q})\ddot{\mathbf{q}} + [I(\mathbf{q})V_s(\mathbf{q},\dot{\mathbf{q}}) + \frac{1}{2}C_1(\mathbf{q},\dot{\mathbf{q}}) \\ &\quad -\frac{1}{2}C_2(\mathbf{q},\dot{\mathbf{q}}) - \frac{1}{2}C_3(\mathbf{q},\dot{\mathbf{q}})]\dot{\mathbf{q}} + G_1(\mathbf{q})\} + \eta \end{aligned} \quad (10)$$

where

$$\begin{aligned} J_t(\mathbf{q}) &= \begin{bmatrix} \mathbf{s}_1' & \cdots & \cdots & \mathbf{s}_1' \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{s}_n' \end{bmatrix} \in R^{6n\times 6n} \\ I(\mathbf{q}) &= \mathrm{diag}(I_1,\cdots,I_n) \in R^{6n\times 6n} \\ C_1(\mathbf{q},\dot{\mathbf{q}}) &= \mathrm{diag}(\mathbf{v}_1\times I_1,\cdots,\mathbf{v}_1\times I_n) \in R^{6n\times 6n} \\ C_2(\mathbf{q},\dot{\mathbf{q}}) &= \mathrm{diag}((I_1\mathbf{v}_1)\times,\cdots,(I_n\mathbf{v}_n)\times) \in R^{6n\times 6n} \\ C_3(\mathbf{q},\dot{\mathbf{q}}) &= \mathrm{diag}(I_1\mathbf{v}_1\times,\cdots,I_n\mathbf{v}_n\times) \in R^{6n\times 6n} \\ G_1(\mathbf{q}) &= \begin{bmatrix} (I_1\mathbf{g})^T & \cdots & (I_n\mathbf{g})^T \end{bmatrix}^T \in R^{6n\times 1} \\ \eta &= \mathrm{diag}(\eta_1,\cdots,\eta_n) \end{aligned}$$

Note that J_t is equal to the transpose of J except for the spatial transpose operation to the spatial vectors.

Comparing equation (1) with (10) we can conclude that

$$\begin{aligned} H(\mathbf{q}) &= J_t(\mathbf{q})I(\mathbf{q})J(\mathbf{q}) & (11) \\ C(\mathbf{q},\dot{\mathbf{q}}) &= J_t(\mathbf{q})[I(\mathbf{q})V_s(\mathbf{q},\dot{\mathbf{q}}) + \frac{1}{2}C_1(\mathbf{q},\dot{\mathbf{q}})J(\mathbf{q}) \\ &\quad -\frac{1}{2}C_2(\mathbf{q},\dot{\mathbf{q}})J(\mathbf{q}) - \frac{1}{2}C_3(\mathbf{q},\dot{\mathbf{q}})J(\mathbf{q})] & (12) \\ G(\mathbf{q}) &= J_t(\mathbf{q})G_1(\mathbf{q}) & (13) \end{aligned}$$

The matrices H and C given by equation (11) and (12) satisfy the property (2). To see this, we can calculate the matrix $\dot{H} - 2C$ from (11) and (12) as

$$\dot{H} - 2C = J_tC_2J \quad (14)$$

To compute the entries of the matrix we obtain

$$[\dot{H} - 2C]_{ij} = \sum_{k=max(i,j)}^{n} \mathbf{s}_i'(I_k\mathbf{v}_k)\times\mathbf{s}_j \quad (15)$$

As $\mathbf{s}_i'(I_k\mathbf{v}_k)\times\mathbf{s}_j$ is a scalar, it is invariant with respect to the spatial transpose operation.

$$\mathbf{s}_i'(I_k\mathbf{v}_k)\times\mathbf{s}_j = (\mathbf{s}_i'(I_k\mathbf{v}_k)\times\mathbf{s}_j)' = -\mathbf{s}_j'(I_k\mathbf{v}_k)\times\mathbf{s}_i \quad (16)$$

It can be simply observed from equation (15),(16) that the matrix $\dot{H} - 2C$ is indeed skew-symmetric.

The coefficients of equation (1) H, C, G has following important property.

$$H(\mathbf{q})\dot{\mathbf{r}} + C(\mathbf{q},\dot{\mathbf{q}})\mathbf{r} + G(\mathbf{q}) = Y(\mathbf{q},\dot{\mathbf{q}},\mathbf{r},\dot{\mathbf{r}})\mathbf{a} \quad (17)$$

where $\mathbf{r}$ is arbitrary vector which have the same size as $\mathbf{q}$ and $\mathbf{a}$ is a constant vector in terms of the robot arm parameters. Now, let us find the entries of Y of (17) for the convenience of considering the recursive form. The spatial inertia of one link has ten parameters, which are six true inertia, three position components of the center of mass, and mass. let $\mathbf{a}$ be composed of these parameters. We can write

$$I_i = \sum_{j=1}^{10} R_j a_{ij} \quad (18)$$

where a_{ij} is the j-th entry of $\mathbf{a}$ with respect to the inertia of the i-th link and $R_j \in R^{6n\times 6n}$ consist of ones and zeros giving the location of the appropriate parameter within the inertia.

Compared (7) with (10), we can see that the replacement of $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ with $\mathbf{r}$ and $\dot{\mathbf{r}}$ between the left-hand side of (1) and that of (17) is equivalent to following replacement in recursive-form description. We introduce a new spatial velocity $\mathbf{w}_i$ as

$$\begin{aligned} \mathbf{w}_i &= \mathbf{w}_{i-1} + \mathbf{s}_i r_i \ (\mathbf{w}_0 = 0) & (19) \\ \dot{\mathbf{w}}_i &= \dot{\mathbf{w}}_{i-1} + \mathbf{v}_i \times \mathbf{s}_i r_i + \mathbf{s}_i \dot{r}_i \ (\dot{\mathbf{w}}_0 = 0) & (20) \end{aligned}$$

Instead of (7) we take

$$\mathbf{f}_i = \mathbf{f}_{i+1} + I_i \dot{\mathbf{w}}_i + \frac{1}{2}\mathbf{v}_i \times I_i w_i - \frac{1}{2}(I_i \mathbf{v}_i) \times w_i - \frac{1}{2} I_i \mathbf{v}_i \times w_i + I_i g \quad (21)$$

It can be easily observed that by constructing the closed-form as above from equations (3), (4), (6), (19), (20), and (21), the left-hand side of (17) appears at that of the resultant closed-form equation. From (18) and (21) we obtain

$$\begin{aligned} \mathbf{f}_i &= \mathbf{f}_{i+1} + \sum_{j=1}^{10} [R_j \dot{\mathbf{w}}_i + \frac{1}{2}\mathbf{v}_i \times R_j \mathbf{w}_i \\ & -\frac{1}{2}(R_j \mathbf{v}_i) \times \mathbf{w}_i - \frac{1}{2} R_j \mathbf{v}_i \times \mathbf{w}_i + R_j \mathbf{g}] a_{ij} \quad (22) \end{aligned}$$

Note that a_{ij} is scalar. By composing equations (3), (4), (6), (19), (20), and (22) $Y(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{r}, \dot{\mathbf{r}})$ and $\mathbf{a}$ can be expressed as

$$Y(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{r}, \dot{\mathbf{r}}) = J_t Y_1 \quad (23)$$

$$\mathbf{a} = \begin{bmatrix} a_{11} & \cdots & a_{1\,10} & \cdots & a_{n\,1} & \cdots & a_{n\,10} \end{bmatrix}^T \quad (24)$$

where Y_1 is very complicated $10 \times 6n$ matrix:

$$\begin{aligned} Y_1 &= \begin{bmatrix} \mathbf{y}_{11} & \cdots & \mathbf{y}_{1n} \\ \vdots & & \vdots \\ \mathbf{y}_{10\,1} & \cdots & \mathbf{y}_{10\,n} \end{bmatrix} \\ \mathbf{y}_{ij} &= R_j \dot{\mathbf{w}}_i + \frac{1}{2}\mathbf{v}_i \times R_j \mathbf{w}_i - \frac{1}{2}(R_j \mathbf{v}_i) \times \mathbf{w}_i \\ & -\frac{1}{2} R_j \mathbf{v}_i \times \mathbf{w}_i + R_j \mathbf{g} \end{aligned}$$

3 DIRECT ADAPTIVE CONTROL SCHEMES

In this section we summarize several works on the adaptive control of the robot arms relevant to this paper.

3.1 CLOSED-FORM CONTROLLER

Slotine and Li [1] presents a globally stable adaptive controller. This controller is designed in closed form as follows.

Let $\hat{\mathbf{a}}$ denote the estimate of $\mathbf{a}$ of (17). We define $\hat{H}, \hat{C}, \hat{G}$ as the estimates of H, C, G respectively by substitute $\mathbf{a}$ for $\hat{\mathbf{a}}$, namely

$$Y(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{r}, \dot{\mathbf{r}})\hat{\mathbf{a}} = \hat{H}(\mathbf{q})\dot{\mathbf{r}} + \hat{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{r} + \hat{G}(\mathbf{q}) \quad (25)$$

let $\mathbf{q}_d$ denote the desired trajectory of $\mathbf{q}$. Note that from the discussion in Section 2, $\hat{H}(\mathbf{q}), \hat{C}(\mathbf{q}, \dot{\mathbf{q}}), \hat{G}(\mathbf{q})$ can be made from $H(\mathbf{q}), C(\mathbf{q}, \dot{\mathbf{q}}), G(\mathbf{q})$ in the equations (11),(12),(13) respectively by replacing I with $\hat{I}$ where $\hat{I}$ is block diagonal matrix of $\hat{I}_i$. Thus,

$$\begin{aligned} \hat{H}(\mathbf{q}) &= J_t \hat{I} J & (26) \\ \hat{C}(\mathbf{q}, \dot{\mathbf{q}}) &= J_t[\hat{I} V_s + \frac{1}{2}\hat{C}_1 J - \frac{1}{2}\hat{C}_2 J - \frac{1}{2}\hat{C}_3 J] & (27) \\ \hat{G} &= J_t \hat{G}_1 & (28) \end{aligned}$$

where $\hat{C}_1, \hat{C}_2, \hat{C}_3$, and $\hat{G}_1$ are made from C_1, C_2, C_3 and G_1 by replacing I with $\hat{I}$.

We further define the virtual reference trajectory $\mathbf{q}_r$ and the virtual error $\mathbf{s}$ as

$$\begin{aligned} \dot{\mathbf{q}}_r &= \dot{\mathbf{q}}_d - \Lambda \tilde{\mathbf{q}} & (29) \\ \mathbf{s} &= \dot{\mathbf{q}} - \dot{\mathbf{q}}_r & (30) \end{aligned}$$

where $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$ and Λ is positive definite symmetric, usually diagonal, design parameter matrix with main respect to the rate of convergence of $\tilde{\mathbf{q}}$. Note that

$$\begin{aligned} H(\mathbf{q})\ddot{\mathbf{q}}_r + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + G(\mathbf{q}) &= Y(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{a} & (31) \\ \hat{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \hat{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \hat{G}(\mathbf{q}) &= Y(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\hat{\mathbf{a}} & (32) \end{aligned}$$

From the above equations we obtain

$$\tilde{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \tilde{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \tilde{G}(\mathbf{q}) = Y(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\tilde{\mathbf{a}} \quad (33)$$

where

$$\begin{aligned} \tilde{H}(\mathbf{q}) &= \hat{H}(\mathbf{q}) - H(\mathbf{q}) \\ \tilde{C}(\mathbf{q}, \dot{\mathbf{q}}) &= \hat{C}(\mathbf{q}, \dot{\mathbf{q}}) - C(\mathbf{q}, \dot{\mathbf{q}}) \\ \tilde{G}(\mathbf{q}) &= \hat{G}(\mathbf{q}) - G(\mathbf{q}) \\ \tilde{\mathbf{a}} &= \hat{\mathbf{a}} - \mathbf{a} \end{aligned}$$

Now the control law and adaptation law are taken as follows

$$\begin{aligned} \tau &= \hat{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \hat{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \hat{G}(\mathbf{q}) - K_D \mathbf{s} & (34) \\ \dot{\hat{a}} &= -\Gamma^{-1} Y^T(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{s} & (35) \end{aligned}$$

where K_D and Γ are both positive definite symmetric, usually diagonal, design parameter matrices with main respect to the rate of convergence of $\mathbf{s}$ and that of $\tilde{\mathbf{a}}$ respectively. Note that this controller is designed in closed-form description.

To prove the global stability of this controller, we consider Lyapunov function candidate

$$V(t) = \frac{1}{2}(\mathbf{s}^T H \mathbf{s} + \tilde{\mathbf{a}}^T \Gamma \tilde{\mathbf{a}}) \quad (36)$$

The time derivative of $V(t)$ is obtained using the equations (1), (33), (34), and (35) as follows

$$\dot{V}(t) = -\mathbf{s}^T K_D \mathbf{s} \le 0 \quad (37)$$

The equation (37) indicates the global convergence of s,in other words,s $\to 0$ as $t \to \infty$. From the definition of s, $\tilde{\mathbf{q}}$ converge to the sliding surface

$$\dot{\tilde{\mathbf{q}}} + \Lambda \tilde{\mathbf{q}} = 0 \quad (38)$$

Thus, the adaptive controller defined by (34) and (35) is globally asymptotically stable for joint positions. Note that the equation (37) gives no guarantee of the convergence of $\tilde{\mathbf{a}}$ because $\dot{V}(t)$ is not a function of $\tilde{\mathbf{a}}$. Hence,this control scheme is a kind of direct adaptive control scheme.

3.2 RECURSIVE-FORM CONTROLLER

Niemayer and Slotine [2] presented a recursive-form implementation of the adaptive controller shown in the previous subsection. The controller is realized as

$$\mathbf{v}_{r_i} = \mathbf{v}_{r_{i-1}} + \mathbf{s}_i \dot{q}_{r_i} \quad (39)$$

$$\dot{\mathbf{v}}_{r_i} = \dot{\mathbf{v}}_{r_{i-1}} + \mathbf{s}_i \ddot{q}_{r_i} + \mathbf{v}_i \times \mathbf{s}_i \dot{q}_{r_i} \quad (40)$$

$$\mathbf{f}_{r_i} = \mathbf{f}_{r_{i+1}} + \frac{1}{2}\mathbf{v}_i \times \hat{I}_i \mathbf{v}_{r_i} + \frac{1}{2}\mathbf{v}_{r_i} \times \hat{I}_i \mathbf{v}_i + \frac{1}{2}\hat{I}_i \mathbf{v}_{r_i} \times \mathbf{v}_i + \hat{I}_i \dot{\mathbf{v}}_{r_i} \quad (41)$$

$$\tau_i = \mathbf{s}_i' \mathbf{f}_{r_i} + \hat{\eta}_i - [K_D \mathbf{s}]_i \quad (42)$$

where $\hat{I}_i$ is the estimate of the spatial inertia of link i defined as

$$\hat{I}_i = \sum_{j=1}^{10} R_j \hat{a}_{ij}$$

The equivalence between this and the controller in Section 3.1 can be easily derived by using the results in Section 2.

Walker presented another recursive-form controller [4]. The control scheme is expressed as

$$\tilde{\mathbf{v}} = \mathbf{v}_i - \mathbf{v}_{r_i} \quad (43)$$

$$\mathbf{f}_{r_i} = \mathbf{f}_{r_{i+1}} + \hat{I}_i(\dot{v}_r - \gamma \mathbf{v}_j + \mathbf{g}) + \mathbf{v}_{r_i} \times \hat{I}_i \mathbf{v}_{r_i} \quad (44)$$

$$\tau_i = \mathbf{s}_i' \mathbf{f}_{r_i} + \hat{\eta}_i \quad (45)$$

where γ is a control gain parameter and the rest is same notation as Niemayer and Slotine's. In the next section, this recursive adaptive control scheme is translated into the closed-form and then evaluated.

4 DERIVATION AND ANALYSIS

In this section, we present the closed-form realization of the Walker's controller presented in the previous section. The contorl scheme is analyzed and evaluated using the closed form.

By composing the equations (39), (40), (43), (44), (45), the closed-form controller of Walker's is derived as

$$\tau = \hat{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \hat{C}_w(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r)\dot{\mathbf{q}}_r + \hat{G}(\mathbf{q}) + \hat{\eta} - \gamma \hat{H}(\mathbf{q})\mathbf{s} \quad (46)$$

where

$$\hat{C}_w(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r) = J_t[\hat{I}V_s(\mathbf{q}, \dot{\mathbf{q}}) + \frac{1}{2}\hat{C}_1(\mathbf{q}, \dot{\mathbf{q}}_r)J(\mathbf{q}) - \frac{1}{2}\hat{C}_2(\mathbf{q}, \dot{\mathbf{q}}_r)J(\mathbf{q}) - \frac{1}{2}\hat{C}_3(\mathbf{q}, \dot{\mathbf{q}}_r)J(\mathbf{q})]$$

Note that this control scheme is different from Slotine and Li's in the dependence of Colioris and centrifugal term matrix on $\dot{\mathbf{q}}_r$. Note also that the feedback gain K_D of this control depends on $\mathbf{q}$, therefore it is time-varying.

Now, let us evaluate the rate of the convergence of s with respect to γ using following Lyapunov function.

$$V_1(t) = \mathbf{s}^T \hat{H} \mathbf{s} \quad (47)$$

The derivative of $V_1(t)$ is expressed as

$$\dot{V}_1(t) = -\gamma \mathbf{s}^T \hat{H} \mathbf{s} \quad (48)$$

(47) and (48) yields

$$\dot{V}_1 = -\gamma V_1 \quad (49)$$

Thus, the rate of convergence of $V_1(t)$ is directly dominated by γ. Suppose that we has chosen a time-invariant matrix for K_D. let k_d be the minimum singular value of K_D and h be the maximun singular value of $\hat{H}(\mathbf{q})$. Then following inequalities are satisfied.

$$V_1(t) \leq h||\mathbf{s}||^2 \quad (50)$$

$$\dot{V}_1(t) = -\mathbf{s}^T K_D \mathbf{s} \leq -k_d||\mathbf{s}||^2 \quad (51)$$

(52)

We can evaluate the rate of convergence of V_1 by

$$\dot{V}_1 \leq -\frac{k_d}{h} V_1 \quad (53)$$

In this case, the rate of convergence is bounded by k_d/h. If constant matrix is chosen as the feedback gain, we can design the rate of convergence only in the worst case.

In most cases, it should be better for us to choose the time-varying feedback gain as Walker's [4], according to the rate of convergence of V_1 as the performance index.

5 CONCLUSION

We have utilized the relation between the closed-form description and recursive-form one of a robot arm dynamics and the direct adaptive controller. We believe that the advanced control will be progressed further and further, then the observation in this paper would reduce the gap between analysis/design and implementation.

References

[1] J.J.E.Slotine,W.Li,"On The Adaptive Control of Robot Manipulators",Int. J. Robotic Research vol.6 No.3 ,1987

[2] G.Niemeyer,J.J.E.Slotine,"COMPUTATIONAL ALGORITHMS FOR ADAPTIVE COMPLIANT MOTION", IEEE trans. on Robotics and Automation pp 566-571,1989

[3] G.Niemayer,J.J.E.Slotine,"PERFORMANCE IN ADAPTIVE MANIPULATOR CONTROL", IEEE Proc. of Conf. on Decision and Control pp 1585-1591,1988

[4] M.W.Walker,"Adaptive Control of Manipulators Containing Closed Kinematic Loops",IEEE trans. on Robotics and Automation pp 10-19,1990

[5] Roy Featherstone,"Robot Dynamics Algorithms", Kluwer Academic Publishers 1987

APPENDIX: THE SPATIAL NOTATION

The spatial notation is much convenient to simplify the dynamic equation of rigid bodies. Now, we make a brief tutorial on the concept of spatial notation, which appears in this paper, such as spatial velocity, spatial force, spatial inertia ,and so forth. The extraordinary operation such as spatial transpose and spatial cross operator is also referred to.

In general, spatial notation consists of spatial vectors and spatial tensors. A spatial vector is 6-dimensional vector which corresponds to two 3-dimensional vectors. A spatial tensor is 6×6 matrix which expresses a transformation between spatial vectors.

Now, suppose a 3-dimensional space which has origin and coordinate axes. Let us consider a rigid body in the coordinate frame which has the translational velocity $\mathbf{v}_P$ and the rotational velocity ω at the point P in the body which is usually taken at the center of mass. Let $\mathbf{r}$ denote the position vector of P. Then the translational velocity at the origin can be expressed as

$$\mathbf{v}_O = \mathbf{v}_P + \mathbf{r} \times \omega$$

where the operator "$\times$" for 3-dimensional vector is defined as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} \times = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

The rotational velocity at the origin is also ω. The spatial velocity of the rigid body is define as $\begin{bmatrix} \omega^T & \mathbf{v}_O \end{bmatrix}^T$. Let the body has the translational momentum $\mathbf{p}$ at P ($\mathbf{p} = m\mathbf{v}_P$) and the rotational momentum L. The spatial momentum of the body is defined as $\begin{bmatrix} \mathbf{p}^T & (\mathbf{r} \times \mathbf{p})^T + L^T \end{bmatrix}^T$.

Consider the force $\mathbf{f}$ acting on the point P of the rigid body considered above. Consider also the torque τ acting on the body. Then the spatial force acting on the body is defined as $\begin{bmatrix} \mathbf{f}^T & (\mathbf{r} \times \mathbf{f})^T + \tau^T \end{bmatrix}^T$

The spatial inertia of the rigid body is defined as the transformation from the spatial velocity to spatial momentum; i.e.

$$\mathbf{m} = I\mathbf{v}$$

where $\mathbf{m}$ is the spatial momentum, I is the spatial inertia and $\mathbf{v}$ is the spatial velocity of the rigid body.

The dynamic equation of the body which is equivalent to the Newton-Euler formulation is expressed as

$$\frac{d}{dt}(I\mathbf{v}) = \mathbf{f} \tag{54}$$

The time derivative of spatial vectors is not its componentwise time derivative.

Suppose a spatial vector $\mathbf{a}$ is moving at the velocity expressed by the spatial velocity $\mathbf{v}$, the time derivative of $\mathbf{a}$ is expressed as

$$\frac{d}{dt}\mathbf{a} = \mathbf{v} \times \mathbf{a}$$

where the spatial cross operator "$\times$", which is for spatial vectors, is defined as follows. Suppose a spatial vector $\mathbf{b} = \begin{bmatrix} \mathbf{b}_l^T & \mathbf{b}_f^T \end{bmatrix}^T$ where both $\mathbf{b}_l$ and $\mathbf{b}_f$ are 3-dimensional vectors. The cross operator is defined as

$$\mathbf{b}\times = \begin{bmatrix} \mathbf{b}_l \\ \mathbf{b}_f \end{bmatrix} \times = \begin{bmatrix} \mathbf{b}_l\times & 0 \\ \mathbf{b}_f\times & \mathbf{b}_l\times \end{bmatrix}$$

The time derivative of spatial inertia of the rigid body moving at the velocity $\mathbf{v}$ is expressed as

$$\frac{d}{dt}I = \mathbf{v} \times I - I\mathbf{v}\times$$

Thus, the dynamic equation (54) becomes

$$\mathbf{f} = I\dot{\mathbf{v}} + \mathbf{v} \times I\mathbf{v} \tag{55}$$

The joint axes can be expressed by spatial vectors. In case the joint type is rotational, if the direction of the axis is represented by a 3-dimensional unit vector $\mathbf{d}$ and the line of axis pass through a point P, the spatial vector which represents the joint axis is expressed as $\begin{bmatrix} \mathbf{d}^T & (\mathbf{r} \times \mathbf{d})^T \end{bmatrix}^T$. For the case such that the joint type is translational, if the direction of the axis is represented by a 3-dimensional unit vector $\mathbf{d}$, the spatial vector which represents the joint axis is expressed as $\begin{bmatrix} 0^T & \mathbf{d}^T \end{bmatrix}^T$. Once the joint axis is represented by the spatial vector $\mathbf{s}$, the relative spatial velocity of the rigid bodies connected by the joint is expressed as $\mathbf{s}\dot{q}$ where q is called a joint variable, which is relative angle in rotational case and which is relative displacement in translational case.

In the spatial notation, there is an extraordinary operation of spatial transpose. In this paper, we have expressed this operation as " $'$ ". The spatial transpose of spatial vector is defined as

$$\begin{bmatrix} \mathbf{a} \\ \mathbf{a}_0 \end{bmatrix}' = \begin{bmatrix} \mathbf{a}_0^T & \mathbf{a}^T \end{bmatrix}$$

where both $\mathbf{a}$ and $\mathbf{a}_0$ are 3-dimensional vectors. The spatial transpose of spatial inertia is defined as

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}' = \begin{bmatrix} D^T & B^T \\ C^T & A^T \end{bmatrix}$$

where A, B, C, and D are all 3×3 matrices. The spatial transpose of scalar is an invariant operation.

The important properties of spatial cross operator and spatial transpose are shown below.

$\mathbf{a}, \mathbf{b}$	:	spatial vector
λ	:	scalar
X, Y	:	spatial tensor
E	:	spatial tensor which is orthogonal, ($E'E$ is identity matrix)

$$\begin{aligned}
\lambda\mathbf{a}\times &= (\lambda\mathbf{a})\times \\
\mathbf{a}\times + \mathbf{b}\times &= (\mathbf{a}+\mathbf{b})\times \\
(\mathbf{a}\times)' &= -\mathbf{a}\times \\
\mathbf{a}\times\mathbf{b} &= -\mathbf{b}\times\mathbf{a} \\
(E\mathbf{a})\times &= E\mathbf{a}\times E' \\
(\mathbf{a}\times\mathbf{b})\times &= \mathbf{a}\times\mathbf{b}\times - \mathbf{b}\times\mathbf{a}\times
\end{aligned}$$

$$\begin{aligned}
(\mathbf{a}')' &= \mathbf{a} \\
(X')' &= X \\
(X\mathbf{a})' &= \mathbf{a}'X' \\
(XY)' &= Y'X' \\
\mathbf{a}'X\mathbf{b} &= (\mathbf{a}'X\mathbf{b})' = \mathbf{b}'X'\mathbf{a} \\
\frac{d}{dt}\mathbf{a}' &= (\frac{d}{dt}\mathbf{a})' \\
\frac{d}{dt}X' &= (\frac{d}{dt}X)'
\end{aligned}$$