The Fourteenth International Symposium on Artificial Life and Robotics 2009 (AROB 14th '09),
B-Con Plaza, Beppu, Oita, Japan, February 5 - 7, 2009

Cooperative Manipulation of a Floating Object by Some Space Robots with Joint Velocity Controllers
– Application of a Tracking Control Method Using Transpose of Generalized Jacobian Matrix –

Shinichi Sagara† Yuichiro Taira‡

† Department of Control Engineering, Kyushu Institute of Technology
Tobata, Kitakyushu 804-8550, Japan
E-mail: sagara@cntl.kyutech.ac.jp

‡ Department of Ocean Mechanical Engineering, National Fisheries University
Nagata-Honmachi, Shimonoseki 759-6595, Japan
E-mail: taira@fish-u.ac.jp

Abstract

We have studied on a cooperative manipulation of a floating object by some space robots. Moreover, for the cooperative motions we have reported that a tracking control method using the transpose of the Generalized Jacobian Matrix (GJM) can be utilized for the robot having joint torque controllers. In this paper, for the cooperative motions by some space robots with joint velocity controllers, we proposed a tracking control method using the transpose of the GJM. Simulation results show the effectiveness of the proposed control method.

1 Introduction

Space exploration in future will demand robots such tasks that should be achieved by cooperative motions of some space robots. We have studied on control problems for realizing such cooperative manipulation and reported that a system consisting of some space robots having manipulators and a floating object can be treated as a kind of distributed system [1,2]. Using the distributed system representation each robot constituting the distributed system can be designed the control system individually.

For the robots having manipulators many control methods have been proposed [3]. Most of them, however, use the inverse of the generalized Jacobian matrix (GJM) [4] which is a coefficient matrix between the end-effector's velocity and the joint velocity of the manipulator. Therefore, if the robot becomes in a singular configuration, the manipulator is out of control because the inverse of the GJM does not exist. For this problem, we have proposed a joint torque input type discrete time trajectory tracking control method using the transpose of the GJM [5] and reported that the proposed control method can be used for the distributed system described above [6].

It is considered that joint velocity controllers are also used for space robot manipulators. So, we have proposed a control method for joint velocity controller [7]. In this paper, we propose a tracking control method using the transpose of the GJM for handling a floating object cooperatively by some space robots having joint velocity controllers. To validate the control method computer simulations are done. Simulation results show the effectiveness of the control method.

2 Modeling [1,2,6]

2.1 Robot system model

In this paper, we consider a space robot system consisting of M robots with manipulators and a floating object, as shown in Fig. 1. The h-th robot ($h = 1, \cdots, M$) consists of an uncontrolled base and n_h-DOF manipulator with revolute joints. Assumptions and symbols used in this paper are defined as follows:

[Assumptions]

- A1) All elements of the space robot are rigid.
- A2) The robot system is standing still in an initial state, i. e., the initial linear momentum and angular momentum of the space robots are zero.
- A3) No external force acts on the robot system.
- A4) Positions and attitude angles of the robots and an object in inertial coordinate frame can be measured.

©ISAROB 2009

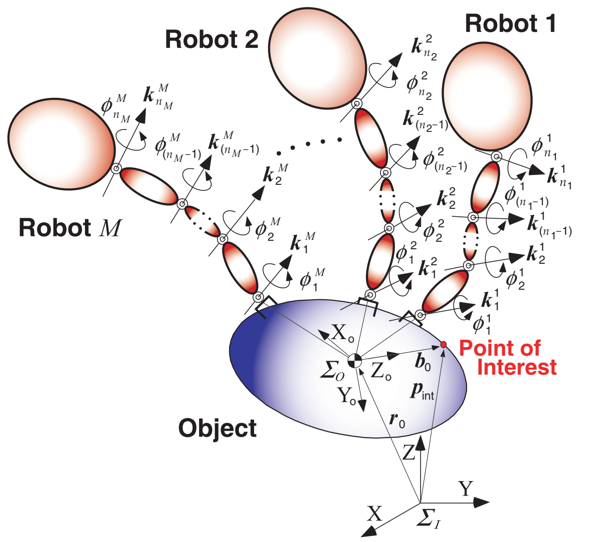


Fig. 1 Model of space robot system

[Symbols]

- Σ_I: inertial coordinate frame
- $\Sigma_{\rm int}$: point of interest coordinate frame
- Σ_T: target coordinate frame
- i^h: number of link or joint i of robot h
- $\boldsymbol{p}_{\rm int}$: position vector of point of interest
- $\boldsymbol{p}_T$: position vector of origin of Σ_T
- $\boldsymbol{r}_0$: position vector of mass center of object
- $\boldsymbol{v}_*$: linear velocity vector of point of interest ($*=$ int) or mass center of object ($*=0$)
- $\boldsymbol{\omega}_*$: angular velocity vector of point of interest ($*=$ int) or mass center of object ($*=0$)
- $\boldsymbol{p}_i^h$: position vector of joint i^h
- $\boldsymbol{r}_i^h$: position vector of mass center of link i^h
- $\boldsymbol{k}_i^h$: unit vector indicating joint axis direction of joint i^h
- $\boldsymbol{r}_g$: position vector of mass center of system
- $\boldsymbol{r}_g^h$: position vector of mass center of robot h
- $\boldsymbol{q}$: joint angle vector
- ${}^I\boldsymbol{A}_*$: rotation matrix from Σ_* ($*=$ int, T) to Σ_I
- ϕ_i^h: relative angle of joint i^h
- $\boldsymbol{\phi}^h$: joint angle vector of robot h
- m_0: mass of object
- m_i^h: mass of link i^h
- $\boldsymbol{I}_0$: inertia tensor of object
- $\boldsymbol{I}_i^h$: inertia tensor of link i^h
- $\boldsymbol{E}$: identity matrix

The tilde operator stands for a cross product such that $\tilde{\boldsymbol{r}}\boldsymbol{a} = \boldsymbol{r}\times\boldsymbol{a}$. All position and velocity vectors are defined with respect to the inertial reference frame.

2.2 Kinematic model

The robot system shown in Fig. 1 can be understood as one robot with M manipulators by regarding the object as a robot body, and M robot arms and robot bodies as M manipulators. The kinematic formulation of such space system has been derived by Yoshida et al. [8]. The relation obtained from its geometrical relationships, and the conservation laws of linear momentum and angular momentum under the above assumptions as follows:

$$\boldsymbol{\nu}_{\rm int} = \begin{bmatrix}\dot{\boldsymbol{p}}_{\rm int}\\ \boldsymbol{\omega}_{\rm int}\end{bmatrix} = \boldsymbol{J}_s\begin{bmatrix}\boldsymbol{v}_0\\ \boldsymbol{\omega}_0\end{bmatrix},\quad \boldsymbol{H}_s\begin{bmatrix}\boldsymbol{v}_o\\ \boldsymbol{\omega}_o\end{bmatrix} + \boldsymbol{H}_m\dot{\boldsymbol{\phi}} = \boldsymbol{0} \qquad (1)$$

where

$$\boldsymbol{J}_s = \begin{bmatrix}\boldsymbol{E} & \tilde{\boldsymbol{r}}_0 - \tilde{\boldsymbol{p}}_{\rm int}\\ \boldsymbol{0} & \boldsymbol{E}\end{bmatrix},\quad \boldsymbol{H}_s = \begin{bmatrix}w\boldsymbol{E} & w(\tilde{\boldsymbol{r}}_0 - \tilde{\boldsymbol{r}}_g)\\ w\tilde{\boldsymbol{r}}_g & \boldsymbol{I}_w\end{bmatrix},$$

$$\boldsymbol{H}_m = \begin{bmatrix}\boldsymbol{J}_{T_w}\\ \boldsymbol{I}_\phi\end{bmatrix},\quad \boldsymbol{\phi} = \left[(\boldsymbol{\phi}^1)^T,\ (\boldsymbol{\phi}^2)^T,\ \cdots,\ (\boldsymbol{\phi}^M)^T\right]^T,$$

$$\boldsymbol{I}_w = \sum_{h=1}^{M}\boldsymbol{I}_w^h + \boldsymbol{I}_o,\quad \boldsymbol{J}_{T_w} = \sum_{h=1}^{M}\boldsymbol{J}_{T_w}^h,\quad \boldsymbol{I}_\phi = \sum_{h=1}^{M}\boldsymbol{I}_\phi^h,$$

$$\boldsymbol{I}_w^h = \sum_{i=1}^{n_h}\{\boldsymbol{I}_i^h - m_i^h\tilde{\boldsymbol{r}}_i^h(\tilde{\boldsymbol{r}}_i^h - \tilde{\boldsymbol{r}}_0^h)\},\quad \boldsymbol{J}_{T_w}^h = \sum_{i=1}^{n_h} m_i^h\boldsymbol{J}_{T_i}^h,$$

$$\boldsymbol{I}_\phi^h = \sum_{i=1}^{n_h}(\boldsymbol{I}_i^h\boldsymbol{J}_{R_i}^h + m_i^h\tilde{\boldsymbol{r}}_i^h\boldsymbol{J}_{T_i}^h),$$

$$\boldsymbol{J}_{T_i}^h = \left[\boldsymbol{O}_a\quad \bar{\boldsymbol{J}}_{T_i}^h\quad \boldsymbol{O}_b\right],\quad \boldsymbol{J}_{R_i}^h = \left[\boldsymbol{O}_a\quad \bar{\boldsymbol{J}}_{R_i}^h\quad \boldsymbol{O}_b\right],$$

$$\bar{\boldsymbol{J}}_{T_i}^h = \left[\tilde{\boldsymbol{k}}_1^h(\boldsymbol{r}_i^h - \boldsymbol{p}_1^h),\ \cdots,\ \tilde{\boldsymbol{k}}_i^h(\boldsymbol{r}_i^h - \boldsymbol{p}_i^h),\ \boldsymbol{0},\ \cdots,\ \boldsymbol{0}\right],$$

$$\bar{\boldsymbol{J}}_{R_i}^h = \left[\boldsymbol{k}_1^h,\ \cdots,\ \boldsymbol{k}_i^h,\ \boldsymbol{0},\ \cdots,\ \boldsymbol{0}\right],$$

and $\boldsymbol{O}_a \in \boldsymbol{R}^{3\times n_a}$ ($n_a = \sum_{i=1}^{h-1} n_i$) and $\boldsymbol{O}_b \in \boldsymbol{R}^{3\times n_b}$ ($n_b = \sum_{i=h+1}^{M} n_i$) are zero matrices.

Form Eq. (1), the relation between velocity $\boldsymbol{\nu}_{\rm int}$ of the object and joint angular velocity $\dot{\boldsymbol{\phi}}$ of the manipulator can be derived as

$$\boldsymbol{\nu}_{\rm int} = \boldsymbol{J}^*\dot{\boldsymbol{\phi}} \qquad (2)$$

where $\boldsymbol{J}^* = -\boldsymbol{J}_s(\boldsymbol{H}_s)^{-1}\boldsymbol{H}_m$ is a GJM of the system shown in Fig. 1.

2.3 System partition

For the system shown in Fig. 1 control systems can be easily constructed by using Eq. (2). However, if the number of robots is changed, Eq. (2) must be recalculated. Furthermore, if the number of robots becomes increased, a large amount of calculation for the system is necessary. To solve the problems described above, This total robot system is regarded as a distributed system.

©ISAROB 2009

By examining parameters and variables included in the matrix $\boldsymbol{H}_s$ and vector $\boldsymbol{H}_m\dot{\boldsymbol{\phi}}$ in Eq. (1), the matrix and vector can be rewritten as

$$\boldsymbol{H}_s = \boldsymbol{H}_s^0 + \sum_{h=1}^{M} \boldsymbol{H}_s^h, \quad \boldsymbol{H}_m\dot{\boldsymbol{\phi}} = \sum_{h=1}^{M} \boldsymbol{H}_m^h \dot{\boldsymbol{\phi}}^h \tag{3}$$

where

$$\boldsymbol{H}_s^0 = \begin{bmatrix} m_0\boldsymbol{E} & \boldsymbol{0} \\ m_0\tilde{\boldsymbol{r}}_0 & \boldsymbol{I}_0 \end{bmatrix}, \quad \boldsymbol{H}_s^h = \begin{bmatrix} m^h\boldsymbol{E} & m^h(\tilde{\boldsymbol{r}}_0^h - \tilde{\boldsymbol{r}}_g^h) \\ m^h\tilde{\boldsymbol{r}}_g^h & \boldsymbol{I}_w^h \end{bmatrix},$$

$$\boldsymbol{H}_m^h = \begin{bmatrix} \bar{\boldsymbol{J}}_{T_w}^h \\ \bar{\boldsymbol{I}}_\phi^h \end{bmatrix}, \quad m^h = \sum_{i=1}^{n_h} m_i^h,$$

$$\bar{\boldsymbol{J}}_{T_w}^h = \sum_{i=1}^{n_h} m_i^h \bar{\boldsymbol{J}}_{T_i}^h, \quad \bar{\boldsymbol{I}}_\phi^h = \sum_{i=1}^{n_h} \left(\boldsymbol{I}_i^h \bar{\boldsymbol{J}}_{R_i}^h + m_i^h \tilde{\boldsymbol{r}}_i^h \bar{\boldsymbol{J}}_{T_i}^h \right).$$

$\boldsymbol{H}_s^h$ and $\boldsymbol{H}_m^h$ are matrices including parameters the h-th robot only, and $\boldsymbol{H}_s^0$ is a matrix including parameters of the object only.

Eqs. (1) and (3) make the following relation:

$$\left(\boldsymbol{H}_s^0 + \sum_{h=1}^{M} \boldsymbol{H}_s^h \right) \boldsymbol{J}_s^{-1} \boldsymbol{\nu}_{\rm int} + \sum_{h=1}^{M} \boldsymbol{H}_m^h \dot{\boldsymbol{\phi}}^h = \boldsymbol{0}. \tag{4}$$

It is clear that the following set of equations is one of solutions of Eq. (4), when a constant and diagonal matrix $\boldsymbol{A}_h$ is introduced.

$$\bar{\boldsymbol{H}}_s^h \boldsymbol{J}_s^{-1} \boldsymbol{\nu}_{\rm int} + \boldsymbol{H}_m^h \dot{\boldsymbol{\phi}}^h = \boldsymbol{0} \quad (h=1,\cdots,M) \tag{5}$$

where

$$\bar{\boldsymbol{H}}_s^h = \boldsymbol{H}_s^h + \boldsymbol{A}_h \boldsymbol{H}_s^0, \qquad \sum_{h=1}^{M} \boldsymbol{A}_h = \boldsymbol{E}.$$

Then, the following relation can be derived from Eq. (5).

$$\boldsymbol{\nu}_{\rm int} = -\boldsymbol{J}_s (\bar{\boldsymbol{H}}_s^h)^{-1} \boldsymbol{H}_m^h \dot{\boldsymbol{\phi}}^h \quad (h=1,\cdots,M). \tag{6}$$

Therefore, for each robot of the system the control system can be designed individually.

3 Digital Control

3.1 Torque input type tracking control law [5]

Eq. (6) can be rewritten as

$$\begin{bmatrix} \boldsymbol{v}_{\rm int}(k) \\ \boldsymbol{\omega}_{\rm int}(k) \end{bmatrix} = -\boldsymbol{J}_s (\bar{\boldsymbol{H}}_s^h)^{-1} \boldsymbol{H}_m^h \dot{\boldsymbol{\phi}}^h(k) = \begin{bmatrix} \boldsymbol{J}_L^h \\ \boldsymbol{J}_A^h \end{bmatrix} \dot{\boldsymbol{\phi}}^h(k) \tag{7}$$

For Eq. (7) the following digital tracking control law using the transpose of the GJM [5] is utilized:

$$\begin{aligned} \boldsymbol{\tau}_d^h(k) = (\boldsymbol{J}_L^h)^T(k) \left[\hat{k}_L(k)\boldsymbol{e}_L(k) - \hat{\boldsymbol{K}}_L(k)\boldsymbol{v}_{\rm int}(k) \right] \\ + (\boldsymbol{J}_A^h)^T(k) \left[\hat{k}_A(k)\boldsymbol{e}_A(k) - \hat{\boldsymbol{K}}_A(k)\boldsymbol{\omega}_{\rm int}(k) \right] \end{aligned} \tag{8}$$

where $\boldsymbol{\tau}_d^h(k)$ is the joint torque input vector and

$$\boldsymbol{e}_L(k) = \boldsymbol{p}_T(k) - \boldsymbol{p}_{\rm int}(k), \ \boldsymbol{e}_A(k) = -\frac{1}{2}\boldsymbol{E}_X^T(k)\boldsymbol{E}_A(k),$$

$$\boldsymbol{E}_A(k) = \begin{bmatrix} \boldsymbol{n}_T(k) - \boldsymbol{n}_{\rm int}(k) \\ \boldsymbol{s}_T(k) - \boldsymbol{s}_{\rm int}(k) \\ \boldsymbol{a}_T(k) - \boldsymbol{a}_{\rm int}(k) \end{bmatrix}, \quad \boldsymbol{E}_X(k) = \begin{bmatrix} \tilde{\boldsymbol{n}}_{\rm int}(k) \\ \tilde{\boldsymbol{s}}_{\rm int}(k) \\ \tilde{\boldsymbol{a}}_{\rm int}(k) \end{bmatrix},$$

$$\hat{k}_\dagger(k) = k_\dagger\{1 + \alpha_\dagger \nu_\dagger(k)\} \ (\dagger = L, \ A),$$

$$\hat{\boldsymbol{K}}_\dagger(k) = \boldsymbol{K}_\dagger\{1 - \beta_\dagger \nu_\dagger(k)\} \ (\dagger = L, \ A),$$

$$\nu_L(k) = \frac{||\boldsymbol{v}_{{\rm int}_d}(k)||}{v_{d_{\max}}}, \quad \nu_A(k) = \frac{||\boldsymbol{\omega}_{{\rm int}_d}(k)||}{\omega_{d_{\max}}}.$$

The vectors $\boldsymbol{n}_*$, $\boldsymbol{s}_*$ and $\boldsymbol{a}_*$ ($* = T$, int) are unit vectors along the axes of Σ_* with respect to Σ_I, i. e., ${}^I\!\boldsymbol{A}_* = [\boldsymbol{n}_*(k) \ \boldsymbol{s}_*(k) \ \boldsymbol{a}_*(k)]$. $*_{{\rm int}_d}(k)$ ($* = \boldsymbol{v}, \ \boldsymbol{\omega}$) is the desired velocity of $*_{\rm int}(k)$, $*_{d_{\max}}$ ($* = v, \ \omega$) is the maximum values of the norm of $*_{{\rm int}_d}(k)$, $\alpha_\dagger$ ($\alpha_\dagger \geq 0$) and $\beta_\dagger$ ($0 \leq \beta_\dagger \leq 1$) are setting parameters. Furthermore, $k_\dagger$ is a positive scalar gain, and $\boldsymbol{K}_\dagger$ is a symmetric and positive definite gain matrix.

3.2 Control input of joint velocity

For manipulators with joint velocity controllers the control law (8) cannot be applied directly. To obtain similar control performance to the case of the joint torque controllers, we use the dynamic equation of the robot. Equation of motion of each space robot shown in Fig. 1 can be described as follows [3]:

$$\boldsymbol{M}^h \dot{\boldsymbol{\chi}}(t) + \boldsymbol{C}^h = \boldsymbol{u}^h(t) \tag{9}$$

where

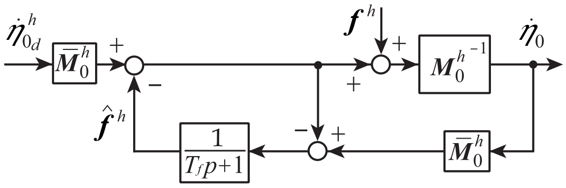
$$\boldsymbol{\chi} = \begin{bmatrix} \boldsymbol{\eta}_0 \\ \dot{\boldsymbol{\phi}}^h \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{f}^h \\ \boldsymbol{\tau}^h \end{bmatrix},$$

and $\boldsymbol{M}^h$ is the symmetric and positive definite inertia matrix, and $\boldsymbol{C}^h$ is the vector of Coliolis and centrifugal forces, $\boldsymbol{\eta}_0 = [\boldsymbol{v}_0{}^T, \ \boldsymbol{\omega}_0{}^T]^T$ is the velocity of the mass center of object, $\boldsymbol{f}^h$ is the external force affected by other robots.

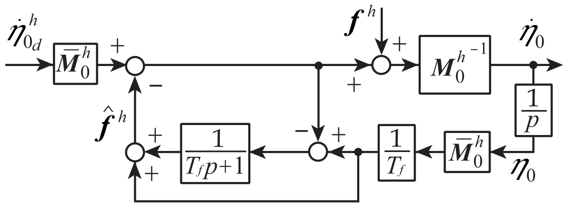
Discretizing Eq. (9) by the sampling period T_1 ($T = nT_1$, n is positive integer) and applying the Euler approximation to $\ddot{\boldsymbol{q}}(k_1)$, we have [7]

$$\boldsymbol{\chi}^h(k_1) = \boldsymbol{\chi}^h(k_1-1) - T_1 {\boldsymbol{M}^h}^{-1}(k_1) \left\{ \boldsymbol{C}^h(k_1) - \boldsymbol{u}^h(k_1) \right\}. \tag{10}$$

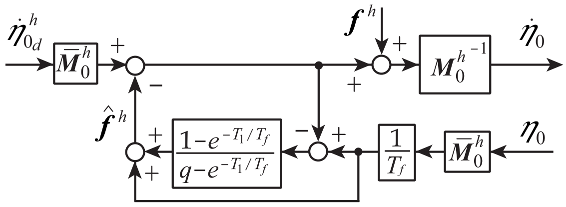
©ISAROB 2009



(a) continuous time block diagram



(b) equivalent transformation of (a)



(c) discrete time block diagram

Fig. 2 Disturbance observer

For Eq. (10) the actual joint velocity control input $\dot{\phi}_d^h(k_1)$ is determined as

$$\chi_d^h(k_1) = \chi^h(k_1-1) - T_1 M^{h^{-1}}(k)\left\{C^h(k_1) - u_d^h(k_1)\right\} \tag{11}$$

where

$$\chi_d(k_1) = \begin{bmatrix} \eta_{0_d}^h(k_1) \\ \dot{\phi}_d^h(k_1) \end{bmatrix}, \quad u_d(k_1) = \begin{bmatrix} f^h(k_1) \\ \tau_d^h(k) \end{bmatrix}.$$

Since the value of the external force f_h affected by other robots cannot obtained directly, a disturbance observer in discrete time [9] is used to estimate f_h.

The equation of motion of the floating object with respect to the h-th robot is

$$M_0^h(k_1)\dot{\eta}_0(k_1) = f^h(k_1) + \bar{M}_0^h \eta_{0_d}^h(k_1) \tag{12}$$

where $M_0^h(k_1)$ is the inertia matrix of the floating object and

$$\bar{M}_0^h = A_h \begin{bmatrix} m_0 E & 0 \\ 0 & I_0 \end{bmatrix}$$

is the nominal model of $M_0^h(k_1)$.

For Eq. (12) the estimated value of f_h, $\hat{f}_h$, can be obtained from the disturbance observer as shown in

Table 1 Physical parameters of robots and object

	Length m	Mass kg	Moment of inertia kg·m^2
Base	3.5	2000	3587.9
Link 2	2.5	50	26.2
Link 1	2.5	50	26.2
hand	0.5	5	0.23
Object	4.0	1200	2400.0

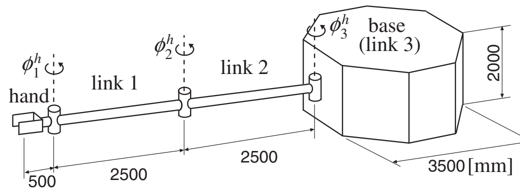


Fig. 3 Space robot

Fig. 2. In this figure, (a), (b) and (c) show the basic configuration in continuous time, the equivalent transformation of (a) and the discrete time version, respectively, T_f is a time constant of a low-pass filter, p and q are the differential and shift operators. From Fig. 2(c), the estimated force can be obtained as follows:

$$\hat{f}^h(k_1) = \hat{f}^h(k_1-1) + \left(1 - e^{-T_1/T_f}\right)\bar{M}_0^h \dot{\eta}_{0_d}(k_1-1) + \frac{1}{T_f}\bar{M}_0^h\left\{\eta_0(k_1) + \left(1 - 2e^{-T_1/T_f}\right)\eta_0(k_1-1)\right\}. \tag{13}$$

4 Simulation

To examine the performance of the proposed control method described in Section 3, simulations are performed by using three of the horizontal planar 3-DOF robots shown in Fig. 3 and an object.

The physical parameters of the robots and object are shown in Table 1. Simulations are carried out under the following condition. A point of interest on the object moves along a straight path from the initial position to the target position, and the object angle is set up as the initial value. The sampling periods are $T = 0.01$s and $T_1 = 0.001$s ($n = 10$). The coefficient matrices are $A_1 = A_2 = 0.33E$ and $A_3 = 0.34E$. The feedback gains are $k_L = k_A = 2 \times 10^5$, $K_L = \mathrm{diag}\{2 \times 10^4,\ 2 \times 10^4\}$ and $K_A = 2 \times 10^4$. The setting parameters are $\alpha_\dagger = 2$ and $\beta_\dagger = 0.2$ ($\dagger = L,\ A$). The time constant for the low-pass filter is $T_f = 1$s.

Fig. 4 shows the motion of the robot system. From this figure, the object is successfully moved by three robots. Fig. 5 shows the time history of the simulation. This figure also shows the case of constant gains, i. e., $\alpha_\dagger = \beta_\dagger = 0$ except estimated values. From Fig. 5,

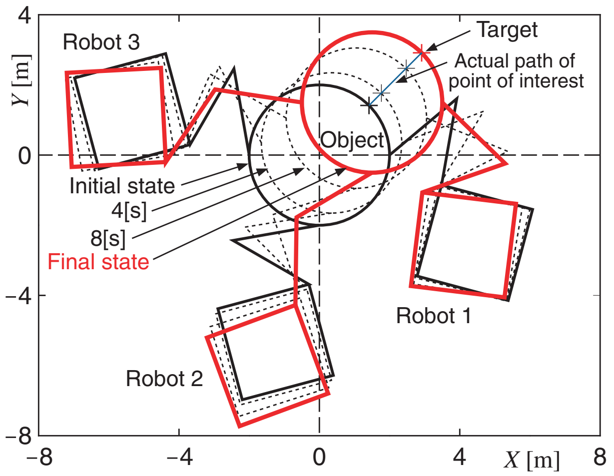


Fig. 4 Motion of the robot system

it can be seen that good control performance can be achieved using the proposed control method.

5 Conclusion

In this paper, we proposed a tracking control method using the transpose of the GJM for handling a floating object cooperatively by some space robots having joint velocity controllers. Simulation results show the effectiveness of the control method.

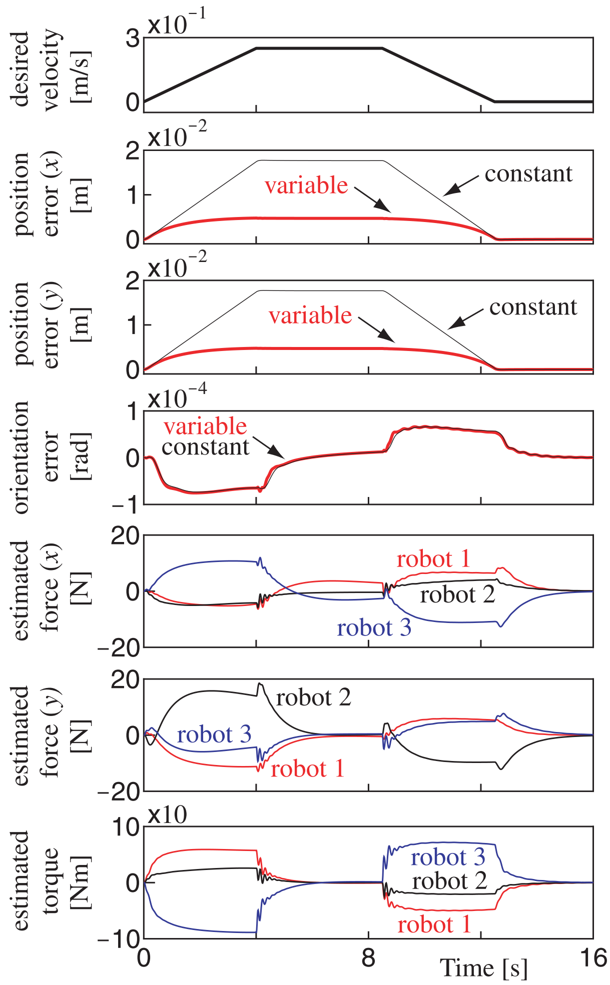


Fig. 5 Simulation result

References

[1] R. Katoh, K. Nakatsuka, S. Sagara, J. Kobayashi, F. Ohkawa, "Manipulation of a Floating Object by Two Space Manipulators", *Proc. IECON'97*, pp. 1397 – 1402, 1997.

[2] S. Sagara, M. Hideura,R. Katoh, T. Yamashita, J. Kobayashi, F. Ohkawa, "Adaptive RMRC for Cooperative Manipulation of a Floating Object by Two Free-Based Space Robots", *Proc. IROS'98*, pp. 1467 – 1472, 1998.

[3] Y. Xu and T. Kanade (eds), *Space Robotics: Dynamics and Control*, Kluwer Academic Publishers, 1993.

[4] K. Yoshida and Y. Umetani, "Control of Space Robot Manipulators with Generalized Jacobian Matrix", In Y. Xu and T. Kanade (eds), *Space Robotics: Dynamics and Control*, Kluwer Academic Publishers, pp. 165–204, 1993.

[5] S. Sagara, Y. Taira, "Digital Tracking Control of Space Robots using Transpose of Generalized Jacobian Matrix", *J. Artif Life Robotics*, Vol. 11, pp. 82 – 86, 2007.

[6] S. Sagara, Y. Taira, "Cooperative Manipulation of a Floating Object by Some Space Robots - Application of a Tracking Control Method Using Transpose of GJM -", *J. Artif Life Robotics*, Vol. 12, pp. 138 – 141, 2008.

[7] S. Sagara, Y. Taira, "Digital Control of Space Robot Manipulators with Velocity Type Joint Controller Using Transpose of Generalized Jacobian Matrix", *Proc. 13th AROB*, pp. 127 – 130, 2008.

[8] K. Yoshida, R. Kurazume, Y. Umetani, "Coordinated Control of Multiple Manipulators in Space Robots", *J. Robotics Society of Japan*, Vol. 9, No. 6, (in Japanese) pp. 718–726, 1991.

[9] I. Godler, H. Honda and K. Ohnishi, "Design Guidelines for Disturbance Observer's Filter in Discrete Time", *Proc. 7th Int. Workshop on Advanced Motion Control*, pp. 390 – 395, 2002.