Real-Time Hand and Eye Coordination for Flexible Impedance Control of Robot Manipulator[*]

Mutsuhiro Terauchi[1], Yoshiyuki Tanaka[2], and Toshio Tsuji[2]

[1] Faculty of Psychological Sciences, Hiroshima International University
mucha@he.hirokoku-u.ac.jp
[2] Graduate school of Engineering, Hiroshima University

Abstract. In recent years a lot of versatile robots have been developed to work in environments with human. However they are not sufficiently flexible nor safe in terms of interaction with human. In our approach we focused on hand and eye coordination in order to establish a flexible robot control, in which a robot recognizes its environment from the input camera images and equips a soft contacting strategy by impedance control. To recognize the environment, we adopt a method to reconstruct motion from a sequence of monocular images by using a pair of parallel straight line segments, which enables us to obtain linear equations to solve the problem. On the other hand the impedance control strategy conveys a flexible interaction between robots and humans. The strategy can be considered as a passive force control, when something contacts the end-effector of the robot. In order to avoid a collision, we introduce a virtual impedance control which can generate force prior to the contact. Neural networks (hereafter: NN) learning is used to decide parameters for impedance control, in which NNs can obtain parameters during the motion (aka: online learning). The validity of the proposed method was verified through experiments with a multijoint robot manipulator.

1 Introduction

For autonomous robots moving in a workspace it is necessary to obtain and update information on their environments by using sensors. Among all types of sensors visual information is one of the most popular because of the abundance of its contents. The robots require geometric information of their environments to control the motion for interaction with them. In this paper we focus on a robot working in an artificial workspace where the environment and the object can be modelled as block shape (polyhedron). Furthermore we introduced impedance control to avoid damages caused by collision of the robot with environments or objects. The control system requires motion information such as position, velocity and acceleration, which can be obtained through input image sequences.

[*] This study was supported in part by Ministry of Culture, Science and Education of Japan.

G. Sommer and R. Klette (Eds.): RobVis 2008, LNCS 4931, pp. 303–318, 2008.

In approaches of *Motion from Image Sequence*, it has been key technology to find correspondence of primitives on subsequent images. [1].

Ullman studied motion estimation based on points correspondence between sequential images by using the assumption of rigidity of the objects. He showed it is possible to obtain motion and structure of an object from points correspondence of 4 points on 3 frames of the sequential images under the parallel projection, and 5 points on 3 frames under the perspective projection [2]. Tsai and Huang derived linear equations by using decomposition of the singular point of the matrix which contains medial parameters obtained from motion information [3]. The equation consists of 8 variables and is linear, and it enables us to obtain solutions from 8 points correspondence on 2 frames under the perspective projection. They realized to get linear solution, while it had been analyzed using nonlinear equations. It is, however impossible to solve the linear equations, if all the 8 points lie on 2 planes where one of two planes intersects the origin of 3D coordinates (at least 5 of 8 points satisfy the condition), or on the surface of the cone which intersects the origin (at least 6 of 8 points satisfy the condition).

On the other hand the impedance control has been proposed to equip robots an ability to behave flexibly against their environments[5].The control strategy is based on passive force generation from virtually set mechanical impedance such as stiffness, viscosity and inertia. However such a motion is invoked only after a contact between a robot and its environment occurs. To avoid collisions between them a virtual impedance scheme was proposed by Tsuji et al[6].It is possible to avoid a collision against an approaching object, or to set dynamic property of a robot adaptively before contacting the object, human or environment. However it is not easy to decide the impedance parameters beforehand.

In this paper we propose a linear algorithm to estimate motion of a rigid object utilizing the relative expression of coordinates by assuming the existence of a pair of parallel line segments, and also an online learning method with NNs to obtain impedance parameters during motion of tasks.

2 Motion Formulation

In this paper we focus on the motion reconstruction from two subsequent images onto which one rigid object in the scene is projected by the perspective projection. In this section we formulate the motion of one end-point of a line segment and the relative motion of another end-point. Motion of a line segment on the image is shown in Fig.1. We define (x, y) as one end-point of a line segment and (r, s) as the relative position of another one, and the positions after motion are defined as (x', y') and (r', s') respectively as well as ones before motion.

2.1 Motion of Terminal Points

We show the geometrical relationship between the motion of point P in the scene and the image plane in Fig.1. The origin of the world coordinate system $OXYZ$ is set at the lens center of camera, and the Z axis of the coordinate system is

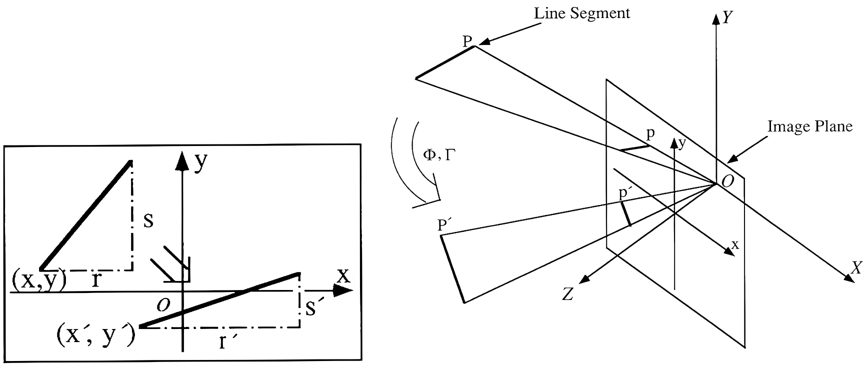


Fig. 1. Moving line segment and its projection onto an image

placed along the optical axis. Then the 3D motion of point P can be generally represented by the rotational component Φ and the translational component Γ as follows,

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \Phi \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \Gamma \tag{1}$$

where Φ and Γ are represented as follows,

$$\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \\ \phi_4 & \phi_5 & \phi_6 \\ \phi_7 & \phi_8 & \phi_9 \end{bmatrix}, \; \Gamma = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} \tag{2}$$

The elements of the rotation matrix Φ can be rewritten by using the rotation angle θ and the rotation axis $(\sigma_1, \sigma_2, \sigma_3\ )$.

On the other hand a point in the 3D space $P(X, Y, Z)$ is projected onto an image plane $p(x, y)$ as follows, The relations between two points in the 3D world $P(X, Y, Z)$ and $P'(X', Y', Z')$, and their projection onto image plane $p(x, y)$ and $p'(x', y')$ are given as follows,

$$x = \frac{X}{Z},\; y = \frac{Y}{Z},\; x' = \frac{X'}{Z'},\; y' = \frac{Y'}{Z'}. \tag{3}$$

where $P'(X', Y', Z')$ and $p'(x', y')$ are the points after motion in 3D space and on the image respectively.

2.2 Relative Expression of Motion

Here we introduce a new Cartesian coordinate system $\hat{O}\hat{X}\hat{Y}\hat{Z}$ whose origin $\hat{O}$ is set at the end-point of line segment P as shown in Fig.2. Then another end-point can be represented as $M(m_1, m_2, m_3)$ in this coordinate system. The corresponding point after motion can be represented as $M'(m_1{}', m_2{}', m_3{}')$, where M and

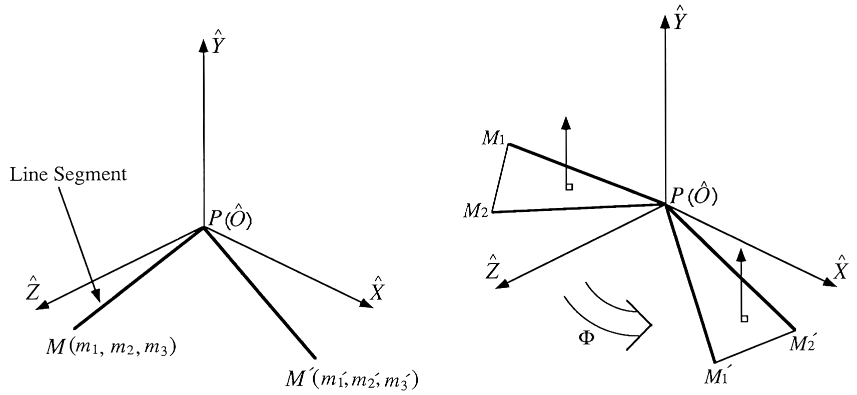


Fig. 2. Local coordinate system $\hat{O}\hat{X}\hat{Y}\hat{Z}$ and motion of two line segments

M' are the vectors along the line-segments. The relations between the vectors M, M' and their projection onto the image plane (r, s), (r', s') are given as follows,

$$r = \frac{m_1 - m_3 x}{Z + m_3}, \quad s = \frac{m_2 - m_3 y}{Z + m_3}, \tag{4}$$

$$r' = \frac{m_1^{'} - m_3^{'} x'}{Z' + m_3^{'}}, \quad s' = \frac{m_2^{'} - m_3^{'} y'}{Z' + m_3^{'}}. \tag{5}$$

In Fig.2, the motion between M and M' is represented only by the rotation matrix Φ as follows,

$$M' = \Phi M \tag{6}$$

Here we introduce parameters t and t' defined as follows,

$$t = \frac{m_3}{Z + m_3}, \; t' = \frac{m_3^{'}}{Z' + m_3^{'}}, \tag{7}$$

These parameters t, t' denote ratios of both depths of end-points of line segments. Therefore, we obtain the following equations,

$$M = (Z + m_3)N, \quad N = \begin{bmatrix} r + tx \\ s + ty \\ t \end{bmatrix}, \tag{8}$$

$$M' = (Z' + m_3^{'})N', \quad N' = \begin{bmatrix} r' + t'x' \\ s' + t'y' \\ t' \end{bmatrix}. \tag{9}$$

For each line segment before and after motion we define unit vectors along the line segments as I and I' respectively,

$$I = \frac{N}{|N|}, \; I' = \frac{N'}{|N'|}. \tag{10}$$

Substituting Eq.(10) into Eq.(8) and Eq.(9), we get the following equations,

$$M = (Z + m_3)\,|N|\,I, \tag{11}$$

$$M' = (Z' + m_3^{'})\,|N'|\,I'. \tag{12}$$

M and M' have the same length, because they are relatively represented and denote the same line segment. As the Z components of the coordinates of the end-points $Z + m_3$ and $Z' + m_3{}'$ are positive, we can define the following value K from Eq.(11) and Eq.(12),

$$K = \frac{Z' + m_3^{'}}{Z + m_3} = \frac{|N|}{|N'|}. \tag{13}$$

Here it is possible to compute the value K, if t and t' defined in Eq.(9) could be obtained.

After Eq.(6) representing the rotation in the 3D space can be described by using the coordinates on the image as follows,

$$KN' = \Phi N. \tag{14}$$

In this equation the unknown parameters are Φ, t and t'.

3 Motion Estimation

3.1 Motion Parameters

When the correspondence between two line segments in the 3D space is found from an image, the parameters in Eq.(14) can be obtained. In the equation Φ is a unitary matrix, therefore we obtain the following equation,

$$\left|K_1 N_1^{'}\right|^2 = |N_1|^2, \tag{15}$$

$$\left|K_2 N_2^{'}\right|^2 = |N_2|^2, \tag{16}$$

$$K_1 K_2 N_1^{'\mathrm{T}} N_2^{'} = N_1{}^{\mathrm{T}} N_2. \tag{17}$$

These equations include unknown parameters t_1, $t_1{}'$, t_2, $t_2{}'$, hence, we need more than three equations shown above.

3.2 Linearization by Using Parallel Relation

Once the relative lengths t and t' could be obtained, it is possible to get the rotation matrix Φ from the relations of two line segments using Eq.(14). Therefore we assume a pair of line segments are parallel in 3D space. This assumption is obviously the constraint for obtaining a unique solution for the parameters t and t'.

If the two line segments l_1 and l_2 are parallel, we can obtain the following equations,

$$\begin{bmatrix} r_1 + t_1 x_1 \\ s_1 + t_1 y_1 \\ t_1 \end{bmatrix} = \alpha \begin{bmatrix} r_2 + t_2 x_2 \\ s_2 + t_2 y_2 \\ t_2 \end{bmatrix} . \tag{18}$$

The relation of line segments holds as well as one after the motion.

$$\begin{bmatrix} r'_1 + t'_1 x'_1 \\ s'_1 + t'_1 y'_1 \\ t'_1 \end{bmatrix} = \alpha' \begin{bmatrix} r'_2 + t'_2 x'_2 \\ s'_2 + t'_2 y'_2 \\ t'_2 \end{bmatrix} , \tag{19}$$

where, α and α' are constant value and denote the ratio of two vectors along two parallel line segments. Here, we can eliminate α and α' in Eq.(26) and (27), and can solve in terms of t_1, t_2, t'_1 and t'_2. Then we get,

$$t_1 = \frac{r_2 s_1 - r_1 s_2}{s_2 x_1 - s_2 x_2 + r_2 y_2 - r_1 y_1}, \tag{20}$$

$$t_2 = \frac{r_2 s_1 - r_1 s_2}{s_1 x_1 - s_1 x_2 + r_1 y_2 - r_1 y_1}, \tag{21}$$

$$t'_1 = \frac{r'_2 s'_1 - r'_1 s'_2}{s'_2 x'_1 - s'_2 x'_2 + r'_2 y'_2 - r'_1 y'_1}, \tag{22}$$

$$t'_2 = \frac{r'_2 s'_1 - r'_1 s'_2}{s'_1 x'_1 - s'_1 x'_2 + r'_1 y'_2 - r'_1 y'_1}. \tag{23}$$

Thus we can obtain the relative lengths t_1, t_2, t'_1 and t'_2 by using the coordinates on an image of end points of line segments which are parallel in the space.

3.3 Motion Estimation by Using Parallelism

As mentioned above, the parameters t_1, t_2, t'_1 and t'_2 can be obtained assuming two line segments are parallel. It is, however, impossible to compute motion parameters, i.e. the rotation matrix, by using just the relationship between parallel two line segments, because the equations derived from them is not independent. Therefore, we also introduce virtual lines which are the line segments obtained by connecting each end-point of two line segments not as to make diagonals as shown in Fig.3 a).

In Fig.3 a), line segments l_1 and l_2 are real ones, and l_3 , l_4 virtual ones. Z coordinates of end points of these four line segments are defined as Z_1, Z_2, Z_3, Z_4, then vectors toward another end points of each line segments: μ_1, μ_2, μ_3, μ_4. These definition of coordinates can be seen in Fig.3 b).

STEP 1. In Fig.3a the line segments l_1 and l_2 denote the real line, and l_3 and l_4 the virtual line. The parameters t_1, ${t_1}'$, t_2 and ${t_2}'$ can be obtained, because line segments l_1 and l_2 are parallel.

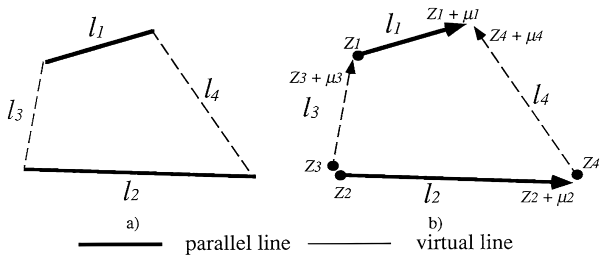


Fig. 3. Virtual lines are added to form a quadrilateral

STEP 2. From the parameters t_1, $t_1{}'$, t_2 and $t_2{}'$, we will have the vectors N_1, $N_1{}'$, N_2 and $N_2{}'$. In the next step we derive the parameter K_3 and K_4 for the virtual line segments by using the obtained parameters K_1 and K_2 for the real line segments.

STEP 3. As well as Eq.(7) for real line segments, we can define t_3 for l_3, and rewritten as,

$$Z_3 + \mu_3 = \frac{Z_3}{1 - t_3}. \tag{24}$$

Here we define the coordinates along Z axis of each base-point (the origin of local coordinates system) of the line segments as Z_1, Z_2, Z_3 and Z_4, and the relative ones of the end-points as m_1, m_2, m_3 and m_4 paying attention to depth (Z axis) component of all points. As four line segments correspond their base/end-points each other, we have following relationships,

$$Z_1 = Z_3 + \mu_3, \tag{25}$$

$$Z_1 + \mu_1 = Z_4 + \mu_4, \tag{26}$$

$$Z_2 + \mu_2 = Z_4, \tag{27}$$

$$Z_2 = Z_3. \tag{28}$$

Here we define the ratios of vector lengths for the virtual line segment l_3 as K_3, as following,

$$K_3 = \frac{Z_3{}' + \mu_3{}'}{Z_3 + \mu_3}. \tag{29}$$

Substituting Eq.(25) into Eq.(29) we have,

$$K_3 = \frac{Z_1{}'}{Z_1}. \tag{30}$$

Substituting Eq.(24) into Eq.(30) we obtain the following equation,

$$\begin{aligned} K_3 = \frac{Z_1{}'}{Z_1} &= \frac{1 - t_1{}'}{1 - t_1} \cdot \frac{Z_1{}' + \mu_1{}'}{Z_1 + \mu_1} \\ &= \frac{1 - t_1{}'}{1 - t_1} \cdot K_1. \end{aligned} \tag{31}$$

Thus the parameter K_3 can be obtained from K_1. In the same way, the ratios of vector lengths K_4 for the virtual line segment l_4 is solved as follows,

$$K_4 = \frac{Z_4{}' + \mu_4{}'}{Z_4 + \mu_4}. \tag{32}$$

Substituting Eq.(26) into Eq.(32) yields,

$$K_4 = \frac{Z_1{}' + \mu_1{}'}{Z_1 + \mu_1} = K_1. \tag{33}$$

Then, the parameter K for the virtual line segment can be acquired by using K for the real line segments. It is self-evident that we can solve them even for replacing the end-point and the base-point of the line segment. Finally we can get components of a vector which represents a direction of a virtual line segment l_3 or l_4 in the next step.

STEP 4. Eq.(16) and Eq.(17) hold for the virtual line segments as well as for the real line segments. These are written as follows,

$$\left|K_3 N_3'\right|^2 = |N_3|^2, \tag{34}$$

$$K_1 K_3 N_1'^{\mathrm{T}} N_3' = N_1{}^{\mathrm{T}} N_3. \tag{35}$$

where N_1 and K_1 are given from Eq.(8), Eq.(9) and Eq.(13) etc. As the equation: $K_3 = \beta K_1$, derived from Eq.(31). Hence, Eq.(34) and Eq.(35) have unknown parameters t_3 and $t_3{}'$.

$$N_3 = \begin{bmatrix} r_3 + t_3 x_3 \\ s_3 + t_3 y_3 \\ t_3 \end{bmatrix}, \quad N_3' = \begin{bmatrix} r_3' + t_3' x_3' \\ s_3' + t_3' y_3' \\ t_3' \end{bmatrix}. \tag{36}$$

The unknowns N_3 and $N_3{}'$ can be expressed by t_3 and $t_3{}'$ using Eq.(8), and Eq.(9), so Eq.(34) and Eq.(35) are rewritten as follows,

$$\begin{aligned} & \beta^2 k_1{}^2 \left\{(r_3{}' + t_3{}' x_3{}')^2 + (s_3{}' + t_3{}' y_3{}')^2 + t_3{}'^2\right\} \\ = \quad & \left\{(r_3 + t_3 x_3)^2 + (s_3 + t_3 y_3)^2 + t_3{}^2\right\} \end{aligned} \tag{37}$$

$$\begin{aligned} & \beta k_1{}^2 \{(r_1{}' + t_1{}' x_1{}')(r_3{}' + t_3{}' x_3{}') + (s_1{}' + t_1{}' y_1{}')(s_3{}' + t_3{}' y_3{}') + t_1{}' t_3{}'\} \\ = \quad & \{(r_1 + t_1 x_1)(r_3 + t_3 x_3) + (s_1 + t_1 y_1)(s_3 + t_3 y_3) + t_1 t_3\} \end{aligned} \tag{38}$$

Then we get two solutions for the above simultaneous equations. Furthermore we can select the unique solution using coefficients of variation, if the image data contains less error or noises. From these obtained parameters t_1, $t_1{}'$, t_2, $t_2{}'$, t_3 and $t_3{}'$, we can have the rotation matrix Φ using Eq.(14).

4 Non-contact Impedance Control

4.1 Impedance Control

In general, a motion equation of an m-joint manipulator in the m-dimensional task space can be written as

$$\boldsymbol{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = \boldsymbol{\tau} + \boldsymbol{J}^T(\boldsymbol{\theta})\boldsymbol{F}_{int} \tag{39}$$

where $\boldsymbol{\theta} \in \Re^m$ is the joint angle vector; $\boldsymbol{M}(\boldsymbol{\theta}) \in \Re^{m\times m}$ is the nonsigular inertia matrix (hereafter, denoted by $\boldsymbol{M}$); $\boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \Re^m$ is the nonlinear term including the joint torque due to the centrifugal, Coriolis, gravity and friction forces; $\boldsymbol{\tau} \in \Re^m$ is the joint torque vector; $\boldsymbol{F}_{int} \in \Re^l$ is the external force exerted on the en-effector; and $\boldsymbol{J} \in \Re^{l\times m}$ is the Jacobian matrix (hereafter, denoted by J).

The desired impedance properties of the end-effector can be expressed as

$$\boldsymbol{M}_e \boldsymbol{d}\ddot{\boldsymbol{X}} + \boldsymbol{B}_e \boldsymbol{d}\dot{\boldsymbol{X}} + \boldsymbol{K}_e \boldsymbol{dX} = \boldsymbol{F}_{int} \tag{40}$$

where $\boldsymbol{M}_e, \boldsymbol{B}_e, \boldsymbol{K}_e \in \Re^{l\times l}$ are the desired inertia, viscosity and stiffness matrices of the end-effector, respectively; and $\boldsymbol{dX} = \boldsymbol{X}_e - \boldsymbol{X}_d \in \Re^l$ is the displacement vector between the current position of the end-effector $\boldsymbol{X}_e$ and the desired one $\boldsymbol{X}_d$. The impedance control law does not use an inverse of the Jacobian matrix and is given as follows:

$$\begin{aligned}
\boldsymbol{\tau} &= \boldsymbol{\tau}_{effector} + \boldsymbol{\tau}_{comp} && (41)\\
\boldsymbol{\tau}_{effector} &= \boldsymbol{J}^T\{\boldsymbol{M}_x(\boldsymbol{\theta})[\boldsymbol{M}_e^{-1}(-\boldsymbol{K}_e\boldsymbol{dX} - \boldsymbol{B}_e\boldsymbol{d}\dot{\boldsymbol{X}}) \\
&\quad + \ddot{\boldsymbol{X}}_d - \dot{\boldsymbol{J}}\dot{\boldsymbol{\theta}}] - [\boldsymbol{I} - \boldsymbol{M}_x(\boldsymbol{\theta})\boldsymbol{M}_e^{-1}]\boldsymbol{F}_{int}\} && (42)\\
\boldsymbol{\tau}_{comp} &= (\boldsymbol{M}^{-1}\boldsymbol{J}^T\boldsymbol{M}_x(\boldsymbol{\theta})\boldsymbol{J})^T\hat{\boldsymbol{h}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) && (43)\\
\boldsymbol{M}_x(\boldsymbol{\theta}) &= (\boldsymbol{J}\hat{\boldsymbol{M}}^{-1}(\boldsymbol{\theta})\boldsymbol{J}^T)^{-1} && (44)
\end{aligned}$$

where $\boldsymbol{M}_x(\boldsymbol{\theta}) = (J\hat{\boldsymbol{M}}\ -1J^T)^{-1} \in \Re^{l\times l}$ indicates the operational space kinetic energy matrix that is proper as long as the joint configuration θ is not singular; $\boldsymbol{\tau}_{effector} \in \Re^m$ is the joint torque vector necessary to realize the desired end-effector impedance; $\boldsymbol{\tau}_{comp} \in \Re^m$ is the joint torque vector for nonlinear compensation; $\hat{\boldsymbol{h}}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ and $\hat{\boldsymbol{M}}$ denote the estimated values of $\boldsymbol{h}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$ and $\boldsymbol{M}(\boldsymbol{\theta})$, respectively, and I is the $l \times l$ unit matrix.

Impedance properties of the end-effector can be regulated by the designed controller.

4.2 Non-contact Impedance Control

Fig.4 (left) schematically shows the non-contact impedance control. Let us consider the case in which an object approaches a manipulator, and set a virtual sphere with radius r at the center of the end-effector. When the object enters

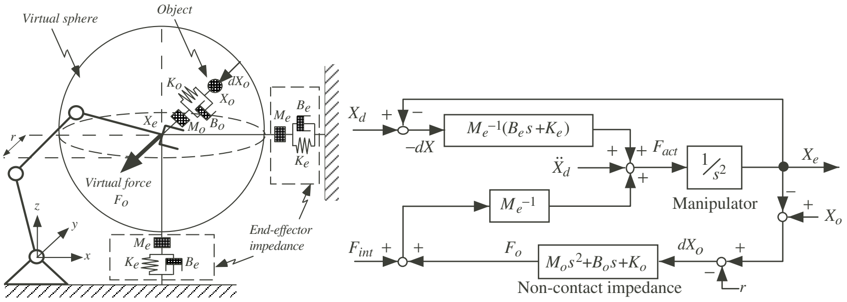


Fig. 4. Scheme and block diagram of non-contact impedance control

the virtual sphere, the normal vector on the surface of the sphere toward the object $\boldsymbol{dX}_o \in \Re^l$ can be represented as

$$\boldsymbol{dX}_o = \boldsymbol{X}_r - r\boldsymbol{n} \tag{45}$$

where $\boldsymbol{X}_r = \boldsymbol{X}_o - \boldsymbol{X}_e$ is the displacement vector from the center of the sphere (namely, the end-effector position) to the object; and the vector $\boldsymbol{n} \in \Re^l$ is given by

$$\boldsymbol{n} = \begin{cases} \dfrac{\boldsymbol{X}_r}{|\boldsymbol{X}_r|} & (\boldsymbol{X}_r \neq \boldsymbol{0}) \\ \boldsymbol{0} & (\boldsymbol{X}_r = \boldsymbol{0}). \end{cases} \tag{46}$$

When the object is in the virtual sphere ($|\boldsymbol{X}_r| < r$), the virtual impedance works between the end-effector and the object so that the virtual external force $\boldsymbol{F}_o \in \Re^l$ is exerted on the end-effector by

$$\boldsymbol{F}_o = \begin{cases} \boldsymbol{M}_o \boldsymbol{d}\ddot{\boldsymbol{X}}_o + \boldsymbol{B}_o \boldsymbol{d}\dot{\boldsymbol{X}}_o + \boldsymbol{K}_o \boldsymbol{dX}_o & (|\boldsymbol{X}_r| \leq r) \\ \boldsymbol{0} & (|\boldsymbol{X}_r| > r) \end{cases} \tag{47}$$

where $\boldsymbol{M}_o, \boldsymbol{B}_o and \boldsymbol{K}_o \in \Re^{l \times l}$ represent the virtual inertia, viscosity and stiffness matrices. It should be noted that $\boldsymbol{F}_o$ becomes zero when the object is outside the virtual sphere or at the center of the sphere. Thus, the dynamic equation of the end-effector for non-contact impedance control can be expressed as

$$\boldsymbol{M}_e \boldsymbol{d}\ddot{\boldsymbol{X}} + \boldsymbol{B}_e \boldsymbol{d}\dot{\boldsymbol{X}} + \boldsymbol{K}_e \boldsymbol{dX} = \boldsymbol{F}_{int} + \boldsymbol{F}_o. \tag{48}$$

Fig. 4 (right) depicts a block diagram of the non-contact impedance control.

The motion equation of the end-effector $\boldsymbol{X}_e(s)$ for the external forces depending on the object position $\boldsymbol{X}_o(s)$ and the desired end-effector position $\boldsymbol{X}_d(s)$ yields

$$\begin{aligned} \boldsymbol{X}_e(s) = {} & \frac{\boldsymbol{M}_o s^2 + \boldsymbol{B}_o s + \boldsymbol{K}_o}{\boldsymbol{M} s^2 + \boldsymbol{B} s + \boldsymbol{K}} \boldsymbol{X}_o(s) + \frac{\boldsymbol{M}_e s^2 + \boldsymbol{B}_e s + \boldsymbol{K}_e}{\boldsymbol{M} s^2 + \boldsymbol{B} s + \boldsymbol{K}} \boldsymbol{X}_d(s) \\ & + \frac{-(\boldsymbol{M}_o s^2 + \boldsymbol{B}_o s + \boldsymbol{K}_o) r\boldsymbol{n} + \boldsymbol{F}_{int}(s)}{\boldsymbol{M} s^2 + \boldsymbol{B} s + \boldsymbol{K}} \end{aligned} \tag{49}$$

where $\boldsymbol{M} = \boldsymbol{M}_o + \boldsymbol{M}_e$, $\boldsymbol{B} = \boldsymbol{B}_o + \boldsymbol{B}_e$, $\boldsymbol{K} = \boldsymbol{K}_o + \boldsymbol{K}_e$. Therefore the stable condition for this system is $\boldsymbol{M}_o \geq -\boldsymbol{M}_e$, $\boldsymbol{B}_o \geq -\boldsymbol{B}_e$, $\boldsymbol{K}_o \geq -\boldsymbol{K}_e$, where equals do not hold simultaneously. In the non-contact impedance control, the relative motion between the end-effector and the object can be regulated by the virtual impedance parameters during non-contact movements. These virtual impedance parameters $(\boldsymbol{M}_o, \boldsymbol{B}_o, \boldsymbol{K}_o)$ are decided using NNs to be described in the next section.

5 Learning of Virtual Impedance by NNs

5.1 Structure of Control System

In the proposed control system, the virtual impedance part in Fig. 4 is composed of three multilayered NNs: a virtual stiffness network (VSN) at $\boldsymbol{K}_o$, a virtual viscosity network (VVN) at $\boldsymbol{B}_o$ and a virtual inertia network (VIN) at $\boldsymbol{M}_o$. The NNs input the relative motion between end-effector and object $(\boldsymbol{X}_r, \dot{\boldsymbol{X}}_r, \ddot{\boldsymbol{X}}_r)$ and interaction force $\boldsymbol{F}_{int}$, while each NN outputs the corresponding impedance parameter: $\boldsymbol{K}_o$ from VSN, $\boldsymbol{B}_o$ from VVN and $\boldsymbol{M}_o$ from VIN. The NNs utilize a linear function in the input units and a sigmoid function in the hidden and output units. Therefore, the input and output of each unit in the i-th layer, x_i and y_i can be derived as

$$x_i = \begin{cases} I_i & \text{(input layer)} \\ \sum w_{ij} y_j & \text{(middle and output layers)} \end{cases} \tag{50}$$

$$y_i = \begin{cases} x_i & \text{(input layer)} \\ \frac{1}{1+e^{-x_i}} & \text{(middle layer)} \\ \frac{U}{2}\left(\frac{1-e^{-x_i+\theta}}{1+e^{-x_i+\theta}}\right) & \text{(output layer)} \end{cases} \tag{51}$$

where w_{ij} indicates the weight coefficient from the unit j to i; and U and θ are positive constants for the maximum output and the threshold of NN, respectively.

5.2 On-Line Learning

The learning of NNs is performed by updating synaptic weights in the NNs on-line so as to minimize an energy function $E(t)$ depending on tasks under the stable condition. The synaptic weights in the VSN, the VVN and VIN are updated in the direction of the gradient descent to reduce the energy function $E(t)$ as

$$\varDelta w_{ij} = -\eta \frac{\partial E(t)}{\partial w_{ij}} \tag{52}$$

$$\frac{\partial E(t)}{\partial w_{ij}} = \frac{\partial E(t)}{\partial \boldsymbol{F}_{act}(t)} \frac{\partial \boldsymbol{F}_{act}(t)}{\partial \boldsymbol{O}(t)} \frac{\partial \boldsymbol{O}(t)}{\partial w_{ij}} \tag{53}$$

where η is the learning rate of each NN, $\boldsymbol{F}_{act}(t) \in \Re^{l}$ is the control input and $\boldsymbol{O}(t) \in \Re^{l\times 3}$ is the NN output. The term $\partial \boldsymbol{F}_{act}(t)/\partial \boldsymbol{O}(t)$ can be computed from Eq.(47), and $\partial \boldsymbol{O}(t)/\partial w_{ij}$ by the error back-propagation learning method. However, the term $\partial E(t)/\partial \boldsymbol{F}_{act}(t)$ cannot obtained directly because of the nonlinear dynamics of the manipulator. In our method the term $\partial E(t)/\partial \boldsymbol{F}_{act}(t)$ is approximated in the discrete-time system so that Δw_{ij} can be calculated in real time by using the change of $E(t)$ for a slight variation of $\boldsymbol{F}_{act}(t)$.

Defining the energy function $E(t)$ depending on end-point position and velocity, $\boldsymbol{X}_e(t)$ and $\dot{\boldsymbol{X}}_e(t)$, the term $\partial E(t)/\partial \boldsymbol{F}_{act}(t)$ can be expanded as

$$\frac{\partial E(t)}{\partial \boldsymbol{F}_{act}(t)} = \frac{\partial E(t)}{\partial \boldsymbol{X}_e(t)}\frac{\partial \boldsymbol{X}_e(t)}{\partial \boldsymbol{F}_{act}(t)} + \frac{\partial E(t)}{\partial \dot{\boldsymbol{X}}_e(t)}\frac{\partial \dot{\boldsymbol{X}}_e(t)}{\partial \boldsymbol{F}_{act}(t)} \,. \tag{54}$$

The slight change of control input $\Delta \boldsymbol{F}_{act}(t)$ within short time yields the following approximations:

$$\Delta \boldsymbol{X}_e(t) \approx \Delta \boldsymbol{F}_{act}(t)\Delta t_s^2 \tag{55}$$

$$\Delta \dot{\boldsymbol{X}}_e(t) \approx \Delta \boldsymbol{F}_{act}^{(k)}(t)\Delta t_s \tag{56}$$

so that $\partial X(t)/\partial \boldsymbol{F}_{act}(t)$ and $\partial \dot{X}(t)/\partial \boldsymbol{F}_{act}(t)$ can be expressed [8] as follows:

$$\frac{\partial \boldsymbol{X}_e(t)}{\partial \boldsymbol{F}_{act}(t)} = \frac{\Delta \boldsymbol{X}_e(t)}{\Delta \boldsymbol{F}_{act}(t)} = \Delta t_s^2 \boldsymbol{I} \tag{57}$$

$$\frac{\partial \dot{\boldsymbol{X}}_e(t)}{\partial \boldsymbol{F}_{act}(t)} = \frac{\Delta \dot{\boldsymbol{X}}_e(t)}{\Delta \boldsymbol{F}_{act}(t)} = \Delta t_s \boldsymbol{I} \tag{58}$$

where Δt_s is a sampling interval, $\boldsymbol{I} \in \Re^{l\times l}$ an identity matrix. Consequently the term $\partial E(t)/\partial \boldsymbol{F}_{act}(t)$ can be approximately computed. With the designed learning rules, an online learning can be held so that the output of NNs, $\boldsymbol{O}(t)$, will be regulated to the optimal virtual impedance parameters for tasks.

6 Experiments with the Robotic Manipulator

To verify the validity of the proposed method, we had some experiments for tasks, catching and hitting an object with a 6-DOF multijoint robotic manipulator as shown in Fig.5 (right).

6.1 Experimental System

The system for the experiments is outlined in Fig.5 (left). Motions of the robotic arm and the object are restricted on the vertical 2D plane in the experiments. The object is a wooden cubic, has several pairs of parallel edges on its surface, is hung from a ceiling by a metal stick and swung like a pendulum. An image sequence of motion of the object is captured with a conventional TV camera, which has been calibrated in advance.

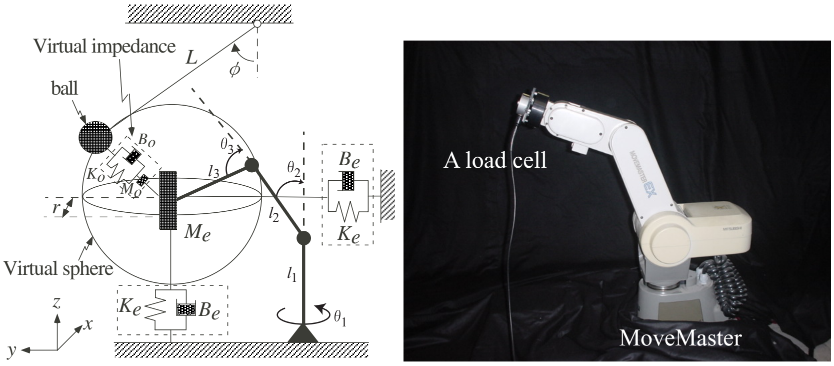


Fig. 5. A model and a manipulator of experimental system

6.2 Catching Task

In catching task, the interaction force between the end-effector and the object should converge to the desired value without overshooting to avoid exerting a large interaction force on them. To this end, relative velocity between the end-effector and the object should be reduced before contact, and the end-point force after contact with the object should be controlled. Accordingly an energy function for the learning of NNs can be defined as

$$E_c = E_{cv}(t) + \mu E_{cf}(t) \tag{59}$$

$$E_{cv}(t) = \frac{1}{2}\left|\alpha(\left|\boldsymbol{X}_r\right|)\dot{\boldsymbol{X}}_r(t_c) - \dot{\boldsymbol{X}}_r(t)\right|^2 \tag{60}$$

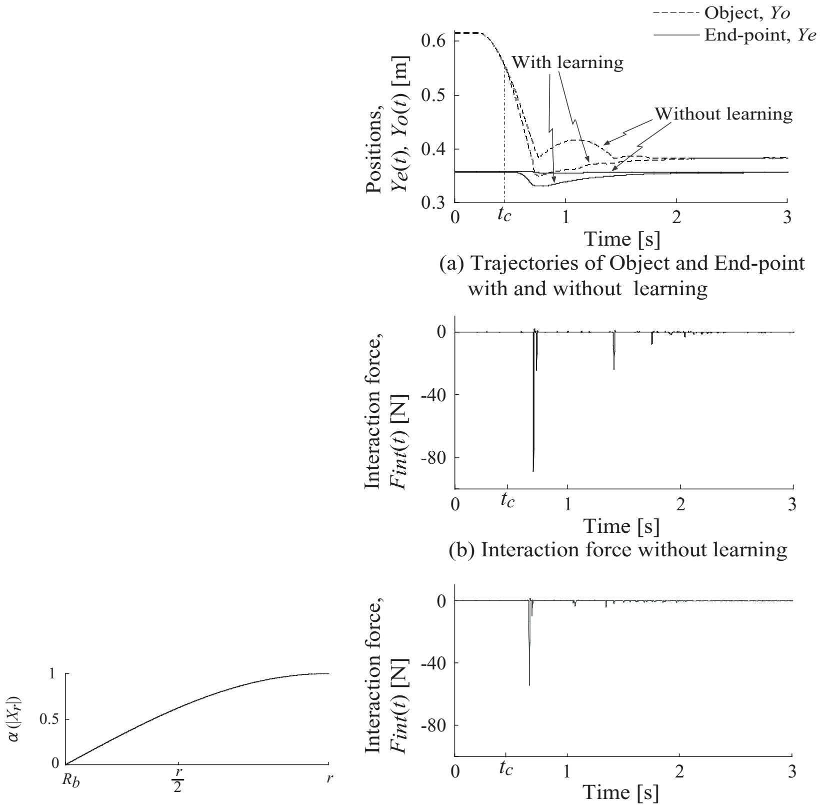
$$E_{cf}(t) = \frac{1}{2}\int_0^t \left|\boldsymbol{F}_{cd}(u) - \boldsymbol{F}_{int}(u)\right|^2 du \tag{61}$$

where t_c is the time when the virtual sphere just contacts the object. In this task, the end-effector should move with the same direction as the approaching direction of the object at the first phase and then slow down gradually in order to catch the object. $\alpha(\left|\boldsymbol{X}_r\right|)$ is the time-varying gain function that should be designed according to contact tasks so as to avoid generating an excessive interaction force while performing the stable learning of NNs immediately after the object enters the virtual sphere. This function plays a role to smooth out the effects from the change of velocity of the object. Consequently the gain function was designed as follows

$$\alpha(\left|\boldsymbol{X}_r\right|) = \begin{cases} \sin\left(\frac{\left(\left|\boldsymbol{X}_r\right| - R_b\right)\pi}{2\left(r - R_b\right)}\right) & (\left|\boldsymbol{X}_r\right| \geq R_b) \\ 0 & (\left|\boldsymbol{X}_r\right| < R_b) \ . \end{cases} \tag{62}$$

where r is a radius of the virtual sphere. $\alpha(|\boldsymbol{X}_r|)$ is shown in Fig.6.

The NNs consist of four layered networks with four input units, two hidden layers with twenty units in each, and one output unit. Fig.6 shows typical experimental results with and without online learning. In Fig.6, (a) illustrates the



(a) Trajectories of Object and End-point with and without learning

(b) Interaction force without learning

(c) Interaction force with learning

Fig. 6. A gain function and experimental results for catching task

time histories of the end-effector position along the x-axis (solid line) and the object (broken line), while (b) and (c) show the time histories of the interaction force without and with real-time learning. t_c indicates the time when the object came into the virtual sphere. It can be seen that the manipulator moves its end-effector according to the object movements after the learning of NNs, so that the robot catches the object smoothly by reducing the impact force between the hand and the object.

6.3 Hitting Task

In hitting task, we need to control the direction of the end-effector motion opposite to that of the object. We define the evaluation function for the task as follows,

$$E_h = E_{hv}(t) + \mu E_{hf}(t) \tag{63}$$

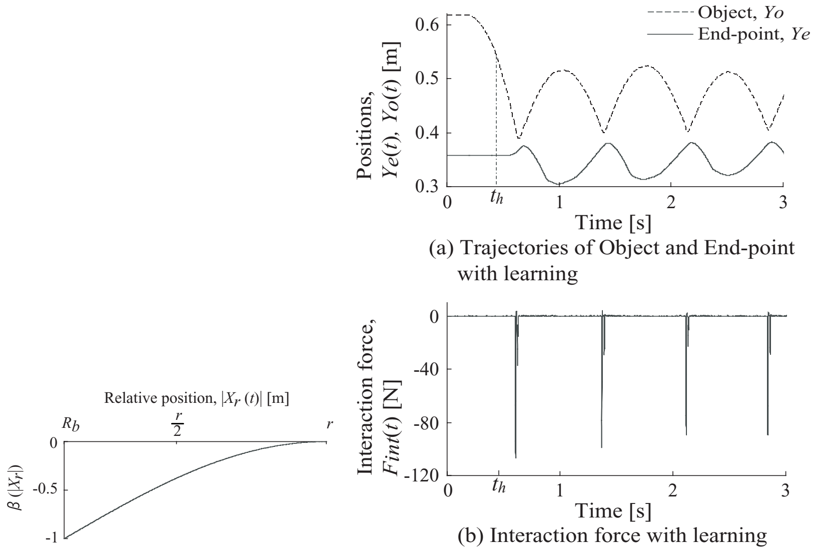
$$E_{hv}(t) = \frac{1}{2}\left|\beta(\left|\boldsymbol{X}_r\right|)\dot{\boldsymbol{X}}_o(t) - \dot{\boldsymbol{X}}_e(t)\right|^2 \tag{64}$$

$$E_{hf}(t) = \frac{1}{2}\int_0^t \left|\boldsymbol{F}_{hd}(u) - \boldsymbol{F}_{int}(u)\right|^2 du \tag{65}$$

where, $\boldsymbol{F}_{hd} \in \Re^l$ depicts the desired interaction force, μ a positive constant which regulates the effect of Eq.(65). The difference of the velocities and the forces between the object and the end-effector is evaluated using $E_{hv}(t)$ and $E_{hf}(t)$ respectively. The function $\beta(\left|\boldsymbol{X}_r\right|)$ in Eq.(64) plays a role to smooth out the effect of the velocity change right after the object enters into the virtual sphere, and defined as

$$\beta(\left|\boldsymbol{X}_r\right|) = \begin{cases} \sin\left(\frac{\left(\left|\boldsymbol{X}_r\right| - R_b\right)\pi}{2\left(r - R_b\right)}\right) - 1 & (\left|\boldsymbol{X}_r\right| \geq R_b) \\ 0 & (\left|\boldsymbol{X}_r\right| < R_b) \end{cases} \tag{66}$$

The function $\beta(\left|\boldsymbol{X}_r\right|)$ is shown in Fig.7. The structure of NNs is the same as that used for the catching task.



(a) Trajectories of Object and End-point with learning

(b) Interaction force with learning

Fig. 7. A gain function and experimental results for hitting task

In Fig.7, (a) illustrates the time histories of the positions of the end-effector (solid line) and the object (broken line) along y-axis, while (b) shows the interaction force between the end-effector and the object with online learning of NNs parameters. It can be seen that the manipulator succesfully hits the object periodically and repeatedly.

7 Conclusion

In this paper we proposed a recognition and a control methods for hand and eye coordination in order to establish a flexible robot system which can interact with artificial environments. From experimental results, real time control of a robot manipulator was achieved in which the robot could acquire parameters for impedance control online using NNs. We utilized a pair of parallel line segments on the object surface. However it is necessary to incorporate much more visual information in order to raise the reliability of reconstructed parameters from images.

References

1. Aggarwal, J.K., Nandhakumar, N.: On the Computation of Motion from Sequence of Images - A Review. Proc. of IEEE 76(8), 917–935 (1988)
2. Ullman, S.: The Interpretation of Visual Motion. MIT Press, Boston (1979)
3. Tsai, R.Y., Huang, T.S.: Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Object with Curved Surface. IEEE Trans. of PAMI PAMI-6(1), 13–27 (1984)
4. Kanatani, K.: Constraints on Length and Angle. Computer Vision, Graphics and Image Processing 41, 28–42 (1988)
5. Hogan, N.: Impedance Control: An Approach to Manipulation, Parts I, II, III. ASME Journal of Dynamic Systems, Measurement, and Control 107(1), 1–24 (1985)
6. Tsuji, T., Kaneko, M.: Non-contact Impedance Control for Redundant Manipulator. IEEE Transaction on Systems, Man, and Cybernetics - Part A 29(2), 184–193 (1999)
7. Cohen, M., Flash, T.: Learning Impedance Parameters for Robot Control Using an Associative Search Networks. IEEE Trans. on Robotics and Automation 7(3), 382–390 (1991)
8. Tsuji, T., Tanaka, Y.: On-line Learning of Robot Arm Impedance Using Neural Networks. Robotics and Autonomous Systems 52(4), 257–271 (2005)