## Contents

Abstract

1

I Eye-Vergence Visual Servoing with 6-Dof Pose

1 Introduction

5

2 Related Work

11

2.1 Object Recognition ................................. 11

2.2 Visual Servoing .................................. 12

3 Previous Work and Purpose of This Research

15

3.1 3-D Pose Recognition ............................. 15

3.2 Visual Servoing to A Moving Ball .................. 16

3.3 Visual Servoing to Human Face .................... 17

3.4 Proposal of Purpose of This Research ............. 18

4 Orientation Representation by Quaternion

21

4.1 Quaternion Definitions ............................. 21

4.2 Deduction of Quaternion Differential Equation \((\dot{\eta}, \dot{\epsilon})\) ......................... 23

4.2.1 Deduction of Angle/Axis Differential Equation \((\dot{\theta}, \dot{k})\) .................. 23

4.2.2 Deduction of Quaternion Differential Equation \((\dot{\eta}, \dot{\epsilon})\) ............... 28

5 Motion Feed-Forward Evolutionary Tracking

31

5.1 Motion-Feedforward (MFF) Compensation ............ 31
5.2 3-D Measurement Method Using Color Information .................. 34
   5.2.1 Kinematics of Stereo-Vision ................................. 34
   5.2.2 Model-based Matching .................................. 35
5.3 On-line Evolutionary Recognition “1-Step GA” .................... 37
5.4 “1-Step GA + MFF” Method ..................................... 43

6 On-line Pose Evolutionary Tracking with Hand-eye Configuration 47
   6.1 Robot Model and Control System ............................. 47
   6.2 Simulation Condition ...................................... 48
   6.3 Error Definition .......................................... 50
   6.4 Simulation Result ...................................... 51

7 6-DoF Visual Servoing Experiment 57
   7.1 Controller .................................................. 57
       7.1.1 Desired-trajectory Generation ...................... 57
       7.1.2 Servoing Controller ................................ 58
   7.2 Experimental Condition ..................................... 60
   7.3 Experimental Result ...................................... 60
       7.3.1 Step Response Experiment .......................... 60
       7.3.2 Time-varying Path Control Experiment ............. 62
       7.3.3 Visual Servoing to A Moving Object ............... 64

8 Hand & Eye-Vergence Visual Servoing System 69
   8.1 Advantage of Eye-Vergence dual Visual Servoing System .... 69
   8.2 Maximum Observability ..................................... 72
   8.3 Eye-vergence Simulation .................................... 73
       8.3.1 Simulator and Robot Dynamics ...................... 73
       8.3.2 Hand & Eye-Vergence Visual Servoing ............... 75
       8.3.3 Stability of Hand & Eye-vergence Motion .......... 78
       8.3.4 Differentiation of Jacobian matrix .................... 79
Control Methods of Robots for Eye-Vergence Visual Servoing and Sensorless Grinding

—Proposals of Control law and Stability Analyses by Lyapunov Method—

Abstract

This research comprise a control method of manipulator aiming at eye-vergence visual servoing system and sensorless grinding robot system as Part. In 3-D position/orientation measurement of an object and 6-DoF visual servoing. The proposed object tracking method is utilizing an evolutionary search technique of a genetic algorithm (GA) and a fitness evaluation based on stereo model matching whose pose is expressed by unit quaternion that has a merit of no representation singularities in an extent from $-\pi$ to $\pi$, unlikely other representations, the convergence of the recognizing method is proved by mathematics.

With a common sense of feedback control, we stress that improvement of the dynamics of the sensing unit is important for a stable visual servoing. We propose a method to improve dynamics in visual tracking, with fictional motions of a target in the camera images being compensated based on kinematics of the manipulator, by extracting real motion of the target. We named it as hand-eye motion feedforward (MFF) method. The enhanced dynamics of online tracking gave further stability and precision to the total visual servoing system, evaluated by full 6-DoF servoing experiment using 7-link manipulator. The convergence time in step response was about 10[s] and precise visual servoing to a moving target object has been achieved.

Moreover, we propose a new two-way visual servoing method, named as hand & eye-vergence visual servoing. This method includes two loops: an outer loop for conventional visual servoing that direct a manipulator toward a target object and an inner loop for active motion of binocular camera for accurate and broad observation of the target object. The
effectiveness of the hand & eye-vergence visual servoing is evaluated through simulations and experiments incorporated with actual dynamics of 7-DoF robot on the view points of how the new idea can improve the stability in visual servoing dynamics and the accuracy of hand pose.

The second part of the paper is concerned with a sensorless grinding robot system which is base on an analysis of the interaction between a manipulator for grinding process and a working object in the task space.

Motions of the constrained dynamics of the robot is modeled first, in the model, the constrained forces are expressed in surface constraint dynamics and hide constraint dynamics which are dual system of each other. Using this result, a new sensorless force control law is proposed by taking advantages of redundancy of the number of input generalized forces against the number of the constrained forces. A controller for a grinding robot is then constructed according to this control strategy which includes geometric constraint condition without involving any force sensors, then the convergence of the controller has been proved in mathematics and confirmed by simulation.
Part I

Eye-Vergence Visual Servoing with 6-Dof Pose


1 Introduction

In recent years, object recognition and visual tracking and servoing using a stereo camera system has been studied intensively in the field of robotics and in other research areas. For a robot to be much smarter than just a mechanical device, vision is required so that it can adapt itself to a changing working environment and recognize objects that exist in its surroundings. Tasks in which visual information are used to direct a manipulator toward a target object are referred to visual servoing \(^1\), \(^2\), \(^3\), \(^4\), as shown in Fig. 1.1. This field is the fusion of many areas, such as kinematics, dynamics, image recognition, and control theory. Generally, visual servoing can be described as a feedback control as shown in Fig. 1.2. The following things are well-known in a feedback control theory. Let \(dY\) denote the change of the output \(Y\), it gives

\[
\frac{dY}{Y} = \frac{1}{1 + CSH} \frac{dS}{S}.
\]

Usually \(CSH \gg 1\), the change of \(S\) will not affect the output a lot, which indicates that the influence from changing the dynamics of the system could be suppressed by the effect of feedback.

Let \(H\) be changed as \(dH\), then the change of the output \(Y\) is

\[
\frac{dY}{Y} = -\frac{CSH}{1 + CSH} \frac{dH}{H}.
\]

Giving \(CSH \gg 1\), we can get the following approximate expression

\[
\frac{dY}{Y} \approx -\frac{dH}{H}.
\]

Eq. (1.3) indicates that the change of \(H\) will affect the output directly even with the high controller gain. This analysis displays the uncertainty and time delay of the dynamics of \(H\) affects the output dynamics directly, and it descends the stability of visual servoing. Therefore, improvement of the dynamics of the sensing unit is essential for stable visual servoing. As shown in Fig. 1.2, hand-eye motion disturbs recognition in \(H\), and incorrect recognition will cause hand motion \(Y\) to be unstable, and the disturbed \(Y\) amplifies servoing
error. This repeating in feedback loop may lead to dangerous unstable motion. Such an undesirable circulation is preferably cut down by improving the recognition dynamics to make the system be robust against the hand-eye motion. However, it looks like that the researches concerning the sensing dynamics for visual servoing have not been concentrated energetically so far.

In this research, we use the word “recognition dynamics” as a time delay of the sensing variables of the 3D pose of the target object, stemming from the fact that sensing mechanism generally be governed by differential equations in time domain. Recently, several researches dealing with the problem of recognition dynamics have been presented. Hashimoto and Kimura 5) proposed a nonlinear controller and a nonlinear observer for the visual servo system to estimate the object’s velocity and predict the object’s motion. Theoretically, the prediction can guarantee the recognition error be reduced to zero when time would be infinite. However, the initial errors need some time to converge to nearly zero, which may cause the visual servoing system unstable. The same method to utilize observer is also discussed by Luca 6) to estimate the z-distance between the object to the camera whose measurement in real time is inherently difficult for image-based visual servoing. Also, correct pose recognition can not be achieved before the z-distance converges to actual one, accompanied by time-delay. And the convergence of z-distance is obtained by using the given fluctuated motion of the camera since it uses single camera to recognize, that is, the method does not work if the

---

Fig. 1.1: Visual servo system of PA-10

Fig. 1.2: Feedback system
camera motion is static. On the other hand, there is a big difference between the sampling rate of the camera 33[ms] and that of the joint servo controller 1[ms] of robots, which also cause the time delay of the sensing unit. Nakabou and Ishikawa 7) use a vision chip whose sampling rate is about 1[ms] to perform high-speed image processing. It has shown that high-speed moving object can be tracked by using the proposed vision chip without any prediction or compensation. However, such a high-speed vision chip system is so expensive that can not be applied popularly.

We use model-based method to recognize 3-D pose in real-time. The matching degree of the model to the target is estimated by a correlation function between the model and the target object, whose maximum value represents the best matching. Then the 3-D recognition of the target object can be solved by finding the valuables of position/orientation that give the maximum value of the correlation function through the on-line optimization method, “1-Step GA” 8). Unit quaternion is used to represent the orientation of the target object, which has an advantage that can represent the orientation of a rigid body without being annoyed by representation singularities. The singularities cause multisolusions for a single orientation, resulting in making the GA difficult to converge to the right variable.

Most visual servo systems use an hand-eye configuration, having the camera mounted on the robot’s end-effector to enhance the dexterity of the operation with visual information from the view point of hand-eye, so the dynamics of the manipulator will make the recognition dynamics deteriorating directly, since the oscillation produces a false motion of the target object in the camera image even though the target is stopping in the task space. We call the false motion as “fictional motion”. In this research, we proposed a motion-feedforward (MFF) method that is to improve the deterioration in recognition dynamics caused by the hand-eye camera’s dynamical motion. The target’s 3-D pose in the camera image made by the fictional motion is predicted through the camera motion calculated by the kinematics of the manipulator and the observation of joint angles and angular velocities. Since the fictional motions can be compensated during on-line recognition, it seems that the recognition were performed by using just fixed cameras in task space, then the recognition dynamics can be separated from the dynamics of the manipulator. Thus the recognition becomes easier and the recognition dynamics can be improved. Contrast to the nonlinear observer method, 5),
Fig. 1.3: Hand & Eye Visual servo system

6), the proposed MFF method can give effective prediction as soon as the camera starts to move without time-delay.

A lot of visual servoing researches focus attention on robot control problem, and simplify or omit the object measurement problem, which is thought to be dealt with in another research field: robot vision field. Visual servoing system is generally hand-eye configuration, in which the camera is fixed on the end-effector. The point is that we have to enhance both the camera observability and the robot stability simultaneously, because they affect each other in visual servoing. Keeping suitable viewpoint is important for object observation. It can provide more information of the object for fast and correct recognition. Unsuitable viewpoint may possibly cause a part of the target object or some feature points get out of the image, which will cause the robot unstable. Some methods are proposed to improve observability of the object, like using stereo camera 9), multiple cameras 10), and two cameras: one is fixed on the end-effector, the other is fixed in the workspace, 11). However, these methods only increase the number of cameras to give different views to observe the object, the cameras in there systems lack the adaptability to a changing environment, that is, the ability to change the viewpoint along with the moving object.
Since the cameras and the end-effector do different tasks - camera is used for observe the target object, and the end-effector is to move toward the target object - it is reasonable to separate the motion of the camera and the motion of the end-effector. In this research, we present a new hand & eye-vergence dual visual servoing system as shown in Fig. 1.3, in which the hand-visual servoing loop includes the active motion of binocular camera to maximize accurate and broad observation of the target object.
2 Related Work

2.1 Object Recognition

There is a variety of approaches for 3D target object’s pose estimation, and they can be classified into three general categories: feature-based, appearance-based, and model-based.

Feature-based approaches use local features like points, line segments, edges, or regions. The main idea of this method is to select a set of feature points, which are matched against the incoming video to update the estimation pose. Feature-based techniques are naturally less sensitive to occlusions, as they are based on local correspondences. Several researches apply this method to head pose estimation based on tracking of small facial features like the corners of the eyes or mouth. Yang and Zhang\textsuperscript{12} presented a head tracking algorithm using stereovision to overcome the occlusion problem. However, the tracker needs to know the initial head pose to start tracking which is determined by seven corresponding landmark points in each image, which are selected manually.

Appearance-based (also template-based) approaches take the template as a whole. The image is compared with various templates to determine which one most closely matches the image, resulting in wasting time to recognize. Some appearance-based methods include the work of Niyogi\textsuperscript{13} and Masson\textsuperscript{14}. In\textsuperscript{14}, the surface of the target 3D object is modeled by a set of small square patches, which needs a learning process to be determined from several key views.

The third method is to use a model to search a target object in the image, and the model is composed based on how the target object can be seen in the input image\textsuperscript{16}, \textsuperscript{17}. Our method is included in this category. The matching degree of the model to the target can be estimated by a fitness function, whose maximum value represents the best matching and can be solved by GA. An advantage of our method is that we use a 3D solid model which enables it possesses six-DOF (both the position and orientation). In other methods like feature-based recognition, the pose of the target object should be determined by a set of image points, which
Table 2.1: Position-based and Image-based Visual Servoing

<table>
<thead>
<tr>
<th></th>
<th>Advantage</th>
<th>Drawback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position-based</td>
<td>Trajectory planning is done in an intuitive Cartesian coordinate space</td>
<td>Require a model of the target object</td>
</tr>
<tr>
<td></td>
<td>There is a clear separation of the measurement problem and the control problem.</td>
<td>Require camera and robot calibration</td>
</tr>
<tr>
<td></td>
<td>Familiar robot control design is used</td>
<td></td>
</tr>
<tr>
<td>Image-based</td>
<td>do not require a model of the target object</td>
<td>difficult to do trajectory planning in the non-intuitive image plane,</td>
</tr>
<tr>
<td></td>
<td>Best suited to planar motion where the plane is parallel to the image plane</td>
<td>difficult to non-planar motion where the plane is not parallel to the image plane</td>
</tr>
<tr>
<td></td>
<td>Robust to camera and robot calibration errors</td>
<td></td>
</tr>
</tbody>
</table>

makes it need a very strict camera calibration. Moreover, searching the corresponding points in Stereo-vision camera images is also complicated and time consuming, e.g., \(^{15}\).

Additionally, some researches concerning scene recognition \(^{18},\,^{19}\) employ some image segmentation techniques that necessitates several processing levels. Such approaches seem to be computationally intensive. Also, in most of the researches that deal with object recognition \(^{20},\,^{21},\,^{22},\,^{23}\), a binary image is used for recognition purposes. However, in most cases, obtaining a binary image necessitates several preprocessing of an original gray-scale image, which may generate \(^{24}\) annoying noise due to thresholding problems.

2.2 Visual Servoing

Visual servoing can be classified into two major groups: position-based and image-based visual servoing. Position-based visual servoing is to determine the object pose in Cartesian coordinate frame and lead to Cartesian robot motion planning, \(^{25},\,^{26}\). On the other hand, in an image-based visual servoing, image features are measured in the 2-D image space, and the robot is controlled directly to servo the image features to a set of desired locations, \(^{28},\,^{29}\), without recognizing the target pose in 3-D space.

The advantages and drawbacks of each visual servoing method have been discussed by a significant amount of researches, listing in Table 2.1. Comparing image-based visual servoing
with position-based visual servoing, the latter is more understandable, since the way of the visual servo is more like human-being, who positions perceived pose in Cartesian space during dynamical action like sports. This 3-D space perception does not exist in the image-based servoing, and thus position-based method suits to the MFF compensation.
3 Previous Work and Purpose of This Research

In this chapter, we introduce a 3-D evolutionary pose measurement method in which the raw image is used directly for image recognition. Contrary to binary image processing, it does not increase the original noise of the threshold problem, and does not require any filtering time.

3.1 3-D Pose Recognition

We use a model-based matching method to recognize a target object in a 3-D searching area. A solid model is located in $\Sigma_E$, its position and orientation are determined by six parameters, $\psi = [x, y, z, \phi, \theta, \psi]^T$, where $\phi, \theta, \psi$ are Euler angles. The left and right input images from the stereo cameras are directly matched by the left and right searching models, which are projected from 3-D model onto 2-D image plane, as shown in Fig. 3.1. The matching degree of the model to the target can be estimated by a correlation function between the projected

![Fig. 3.1: Coordinate systems](image)

$\text{Solid Model}$

$\text{Searching Area}$

$\text{Camera L}$

$\text{Camera R}$

$\text{Image L}$

$\text{Image R}$

$\text{CR}_i$

$r_{1,\text{max}}$

$r_{2,\text{max}}$

$r_{3,\text{max}}$

$r_{1,\text{min}}$

$r_{2,\text{min}}$

$r_{3,\text{min}}$

$\sum M$

$\sum W$

$\sum I_R$

$\sum C_R$

$\sum I_L$

$\sum C_L$

$d$

$\psi_M = [r_1, r_2, r_3, \epsilon_1, \epsilon_2, \epsilon_3]^T$
models and input images, used as a fitness function $F(\psi)$ of the target. When the searching models fit to the target objects being imaged in the right and left images, $F(\psi)$ gives the maximum value. Therefore the 3-D object’s position/orientation measurement problem can be converted to a searching problem of $\psi$ that maximizes $F(\psi)$. We solve this optimization problem by a proposed “1-step GA” method.

### 3.2 Visual Servoing to A Moving Ball $^{30}$

For Real-Time visual control purposes, we employed the described method, ”1-Step GA”, to achieve Real-Time evolutionary recognition. This Real-Time recognition strategy works when the convergence speed of the individuals is higher than the moving speed of the target object in the successively input images. Considering the result of this experiment (6DOF) that calculation time for processing one generation of GA takes 150ms $^{30}$, we can say that it is impossible to recognize the position and orientation simultaneously in Real-Time unless the accuracy and speediness is improved. So in order to evaluate the effectiveness of the real time recognition of the proposed 3-D measurement method, the Visual-Servo experiment
using a ball as the target object have been performed, for it has only 3DOF expressing the ball’s x, y and z positions.

In the experiment, the target ball was put on a turning table and being tracked by a manipulator equipped with Hand-Eye (two cameras). The experimental system is shown in Fig.3.2, and the photograph of our experimental system is shown in Fig.3.3.

The true position of the ball in $\Sigma_W$ as the table turning is shown in Fig.3.4, and the experimental results of the hand position in $\Sigma_W$ is presented in Fig.3.5 under the speed of the turning table $\pi/60$ rad/sec. Here, in order to receive accurate GA searching result, we keep the distance from the origin of $\Sigma_{CR}$ to the center of ball as 600mm. It means that the position difference between the robot hand and the ball recognition result is 600mm. Then the value of Y,Z in Fig.3.5 could be directly compared with that in Fig.3.4, but the value of X in Fig.3.5 should minus 600mm to compare the one in Fig.3.4. It shows that the hand position of Y,Z directions in $\Sigma_W$ could be obtained accurately, but in X direction, the error of hand position is a little bigger. However, the error exists in the range of no bigger than 50mm, which can ensure that our tracking experiment can be conducted successfully.

3.3 Visual Servoing to Human Face

Firstly, we proposed a method to recognize the pose of a human head based on the model-based matching method introduced in 3.1. Here, we use roll, pitch, yaw angles to represent the target orientation. The head position is detected by an optimization of the computation of the brightness difference between an internal surface of head and a contour-strips, meanwhile, the head orientation is measured using color information about the skin region and the hair region of the input images. Moreover, the human’s eyes, as one of the facial features are also recognized to improve the robustness of head pose measurement. An objective function, which includes these three estimations, is defined to evaluate the extent how much the head model matches with the head being imaged, changing the recognition problem into an optimization problem, which is solved by GA.

We conducted the performance of 6-DOf visual servoing to a human face. The head of a human model is taken as the target head here. The visual servoing system is shown in Fig.3.6. As a fundamental research step, here we also did the step response to evaluate the
ability of the system. To evaluate the motion in all positions and orientations, the target head is moved a little, set as \((\phi, \theta, \psi) = (2, 4, 8)[\text{deg}]\). And the experimental result is shown in Figs. 3.7. As shown in the graphs, during the first 750 generations, the end-effector kept moving and rotating, then from the 750\(^{th}\) generation, both the position and orientation is unchanged. Since one generation of GA cost about 0.2\([s]\), the end-effector spent 150\([s]\) to reach the expected pose and keep stable. Even though the motion was slowly, the effectiveness of our method was verified by the performance of visual servo about 6DoF.

### 3.4 Proposal of Purpose of This Research

In the previous research, we have proposed a 3-D pose measurement method, and applied this method to 6-DoF visual servoing. These experimental results have shown the effectiveness of the proposed recognition method. However, it is still difficult to apply this method to a real-time visual servoing (6-DoF) task, because in GA process the convergence of the individuals cost too much time and the recognition accuracy is not satisfactory.

As we have stated in the introduction, the sensing unit is very important in the whole visual servoing system. A robust, correct recognition without long time-delay will increase the stability of visual servoing, and vice versa. Thus, in this research, we focus on how to improve the object observability to improve the stability of our visual servoing. And we
propose a MFF compensation method to make the recognition be robust to the ego motion of the hand-eye camera. Additionally, a new hand & eye-vergence dual visual servoing system is also proposed, in which stereo hand-eye cameras can change their poses to keep a suitable viewpoint to observe the target during the end-effector do the visual servoing task.
4 Orientation Representation by Quaternion

4.1 Quaternion Definitions

There are several representations used to describe the orientation of a rigid body. Euler angles is a well-known one that includes a set of three angles rotating around three coordinates, $z$, $y$, $z$ successively. A drawback of the Euler angles is the occurrence of representation singularities (for manipulator, the Jacobian matrix is singular for some orientation). An alternative representation is angle/axis, describing the general orientation of a rigid body as a displacement of an angle around an axis. A general angle/axis representation is not unique since a rotation by an angle $-\theta$ around an axis $-k$ can not be distinguished from a rotation by $\theta$ around $k$. Moreover, a representational singularity happens when $\theta = 0$, the rotation axis can not be determined. It does not satisfy our research since we use GA to search the best position/orientation, the problem appears when $\theta = 0$, where the rotation axis $k$ can not be converged since the above problem. Thus, it can not be used to detect 3-D pose and to apply it to visual servoing tasks in which the desired target pose is $\theta = 0$. A unit quaternion is a different angle/axis representation. It can represent the orientation of a rigid body without singularities.

For the reader’s convenience, a few basic concepts regarding the use of a unit quaternion to describe the orientation of a rigid body are summarized hereafter \(^{(35)}\). The unit quaternion is defined as

$$Q = \{ \eta, \epsilon \},$$  \hspace{1cm} (4.1)

where

$$\eta = \cos \frac{\theta}{2}$$  \hspace{1cm} (4.2)

$$\epsilon = \sin \frac{\theta}{2} k,$$  \hspace{1cm} (4.3)
where \( \mathbf{k}(\|\mathbf{k}\| = 1) \) is the rotation axis and \( \theta \) is the rotation angle around \( \mathbf{k} \). \( \eta \) is called the scalar part of the quaternion while \( \epsilon \) is called the vector part of the quaternion. They are constrained by

\[
\eta^2 + \epsilon^T \epsilon = 1 \tag{4.4}
\]

hence it is named unit quaternion. It is worth to remark that, differently from the angle/axis representation, a rotation by an angle \(-\theta\) about an axis \(-\mathbf{k}\) gives the same quaternion as that associated with a rotation by \(\theta\) about \(\mathbf{k}\), and \(\theta = 0\) gives \(\eta = 1, \epsilon = 0\), which has solved the nonuniqueness problem from singularity.

The rotation matrix corresponding to a given quaternion is

\[
\mathbf{R}(\eta, \epsilon) = (\eta^2 - \epsilon^T \epsilon) \mathbf{I} + 2\epsilon \epsilon^T + 2\eta \mathbf{S}(\epsilon) \tag{4.5}
\]

where \(\mathbf{S}(\cdot)\) is the operator performing the cross product between two \((3 \times 1)\) vectors. Given \(\mathbf{a} = [a_x, a_y, a_z]^T\), \(\mathbf{S}(\mathbf{a})\) takes on the form

\[
\mathbf{S}(\mathbf{a}) = \begin{bmatrix}
0 & -a_z & a_y \\
 a_z & 0 & -a_x \\
- a_y & a_x & 0
\end{bmatrix} \tag{4.6}
\]

For arbitrary vectors \(\mathbf{a}\) and \(\mathbf{b}\), we have

\[
\mathbf{S}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}, \quad \mathbf{S}(\mathbf{a})\mathbf{b} = -\mathbf{S}(\mathbf{b})\mathbf{a}. \tag{4.7}
\]

On the other hand, the quaternion corresponding to a given rotation matrix \(\mathbf{R} = \{R_{ij}\}(i, j = 1, 2, 3)\) is

\[
\eta = \frac{1}{2} \sqrt{1 + R_{11} + R_{22} + R_{33}} \tag{4.8}
\]

\[
\epsilon = \frac{1}{4} \begin{bmatrix}
R_{32} - R_{23} \\
R_{13} - R_{31} \\
R_{12} - R_{21}
\end{bmatrix} \tag{4.9}
\]

The relations between the time derivative of the Quaternion parameters and the body angular velocity \(\omega\) are established as:

\[
\dot{\eta} = -\frac{1}{2} \epsilon^T \omega \tag{4.10}
\]

\[
\dot{\epsilon} = \frac{1}{2}(\eta \mathbf{I} - \mathbf{S}(\epsilon)) \omega \tag{4.11}
\]

Deductions of (4.10) and (4.11) are written in the next Section.
4.2 Deduction of Quaternion Differential Equation (\(\dot{\eta}, \dot{e}\))

4.2.1 Deduction of Angle/Axis Differential Equation (\(\dot{\theta}, k\))

Angle/Axis can be obtained in terms of a rotation angle \(\theta\) around an axis \(k\). Then, the angular velocity vector \(\omega\) is given by

\[
\omega = k \dot{\theta},
\]

where \(k = [k_1, k_2, k_3]^T\) is a \(3 \times 1\) unit vector, that is, \(k^T k = I\). By using \(k^T k = I\), (4.12) can be rewritten as

\[
\dot{\theta} = k^T \omega.
\]

In Fig. 4.1 the vector \(r\) rotate around the axis \(k\) angle \(\theta\) and get a new vector \(r'\) so

\[
r' = R(\theta, k)r
\]

And from Fig. 4.1,

\[
r' = \overrightarrow{ON} + \overrightarrow{NV} + \overrightarrow{VQ}
\]

\[
= \overrightarrow{ON} + \overrightarrow{NP} \cdot \cos \theta - \overrightarrow{QV}
\]

\[
= \overrightarrow{ON} + (\overrightarrow{OP} - \overrightarrow{ON}) \cdot \cos \theta - \overrightarrow{QV}
\]

\[
= k(k^T \cdot r) + [r - k(k^T \cdot r)] \cos \theta - (r \times k) \sin \theta
\]

\[
= (kk^T + \cos \theta(I - kk^T) + \sin \theta S(k))r
\]

From (4.13) and (4.15) the rotation matrix corresponding to an angle/axis representation is

\[
R = kk^T + \cos \theta(I - kk^T) + \sin \theta S(k).
\]

Here, we define

\[
R_a = kk^T.
\]

\[
R_b = \cos \theta(I - kk^T).
\]

\[
R_c = \sin \theta S(k).
\]
Substituting (4.17), (4.18), (4.19) into (4.16), then (4.16) and the differential of (4.16) can be described as

\[
R = R_a + R_b + R_c, \quad (4.20)
\]

and

\[
\dot{R} = \dot{R}_a + \dot{R}_b + \dot{R}_c. \quad (4.21)
\]

Multiplying \( k \) to both left and right side of (4.21), we have

\[
\dot{R}k = \dot{R}_a k + \dot{R}_b k + \dot{R}_c k, \quad (4.22)
\]

where

\[
\dot{R}_a k = \frac{d}{dt} \langle kk^T \rangle k = (\dot{k}k^T + kk^T \dot{k})k + kk^T \dot{k} = (I + kk^T) \dot{k} \quad (4.23)
\]
\[
\dot{R}_k = \frac{d}{dt}(\cos\theta(I - kk^T))k \\
= \left(- \sin\theta \dot{\theta}(I - kk^T) + \cos\theta(\dot{k}k^T - k\dot{k}^T)\right)k \\
= -(\sin\theta)\dot{\theta}(I - kk^T)k + \cos\theta(\dot{k}k^T - k\dot{k}^T)k \\
= -(\sin\theta)\dot{\theta}(k - k) - \cos\theta(k + kk^T\dot{k}) \\
= -\cos\theta(I + kk^T)\dot{k} 
\]

(4.24)

By using \( \dot{S}(r) = S(\dot{r}) \) and \( S(\dot{r})r = -S(r)\dot{r} \), we get

\[
\dot{R}_c = \frac{d}{dt}(\sin\theta S(k))k \\
= \left((\cos\theta)\dot{\theta}S(k) + \sin\theta S(\dot{k})\right)k \\
= (\cos\theta)\dot{\theta}S(k)k + \sin\theta S(\dot{k})k \\
= (\cos\theta)\dot{\theta}(k \times k) - \sin\theta S(k)\dot{k} \\
= -\sin\theta S(k)\dot{k} 
\]

(4.25)

Substituting (4.23), (4.24), (4.25) into (4.22), we get

\[
\dot{R}_k = \dot{R}_s k + \dot{R}_b k + \dot{R}_c k \\
= (I + kk^T)\dot{k} - \cos\theta(I + kk^T)\dot{k} - \sin\theta S(k)\dot{k} \\
= [(1 - \cos\theta)(I + kk^T) - \sin\theta S(k)] \dot{k} 
\]

(4.26)

On the other hand,

\[
\dot{R} = [\dot{x}, \dot{y}, \dot{z}] \\
= [\omega \times x, \omega \times y, \omega \times z] \\
= [S(\omega)x, S(\omega)y, S(\omega)z] \\
= S(\omega)R 
\]

(4.27)

Multiplying \( k \) to both left and right side of (4.27) and Substituting (4.17), (4.18), (4.19) into
\[
\dot{R}_k = S(\omega)R_k
\]
\[
= S(\omega)R_\alpha k + S(\omega)R_\beta k + S(\omega)R_\gamma k
\]
\[
= S(\omega)kk^T k + S(\omega)(\cos(\theta(I - kk^T)))k + S(\omega)(\sin(\theta)S(k))k
\]
\[
= S(\omega)k + \cos(\theta)S(\omega)(k - k) + \sin(\theta)S(\omega)k \times k
\]
\[
= S(\omega)k
\]
\[
= -S(k)\omega
\]  

(4.28)

From (4.26) and (4.28), we have

\[((1 - \cos(\theta))(I + kk^T) - \sin(\theta)S(k))k = -S(k)\omega.\]  

(4.29)

To move the matrix before \( k \) to the right side of (4.29), we consider the following equation.

\[
\frac{1}{1 - \cos(\theta)}(I - kk^T) - \frac{1}{\sin(\theta)}S(k)\frac{1}{(1 - \cos(\theta))(I + kk^T) - \sin(\theta)S(k)}
\]
\[
= (I - kk^T)(I + kk^T) - \frac{\sin(\theta)}{1 - \cos(\theta)}(I - kk^T)S(k) - \frac{1 - \cos(\theta)}{\sin(\theta)}S(k)(I + kk^T) + S(k)S(k)
\]
\[
= (I - kk^T) - \frac{\sin(\theta)}{1 - \cos(\theta)}S(k) - \frac{1 - \cos(\theta)}{\sin(\theta)}S(k) + (kk^T - I)
\]
\[
= -\frac{2}{\sin(\theta)}S(k).
\]  

(4.30)

where

\[(I - kk^T)(I + kk^T)\]
\[
= I - kk^T + kk^T - kk^T kk^T
\]
\[
= I - kk^T,
\]  

(4.31)

\[(I - kk^T)S(k)\]
\[
= S(k) - kk^T S(k)
\]
\[
= S(k) - k(S^T(k)k)^T
\]
\[
= S(k) - k(-S(k)k)^T
\]
\[
= S(k),
\]  

(4.32)
\[ S(k)(I + kk^T) \]
\[ = S(k) + S(k)kk^T \]
\[ = S(k), \quad (4.33) \]

and

\[ S(k)S(k) \]
\[ = kk^T - k^T kI \]
\[ = kk^T - I, \quad (4.34) \]

since \( S(a)S(b) = ba^T - a^T bI. \)

Multiplying \( \left[ \frac{1}{1 - \cos \theta} (I - kk^T) - \frac{1}{\sin \theta} S(k) \right] \) to both left and right side of (4.29), and using (4.30), we get

\[ -\frac{2}{\sin \theta} S(k) \dot{k} = -\left[ \frac{1}{1 - \cos \theta} (I - kk^T) - \frac{1}{\sin \theta} S(k) \right] S(k) \omega. \quad (4.35) \]

Then,

\[ S(k) \dot{k} = \frac{1}{2} \left[ \cot \frac{\theta}{2} (I - kk^T) - S(k) \right] S(k) \omega \]
\[ = \frac{1}{2} S(k) \left[ \cot \frac{\theta}{2} (I - kk^T) - S(k) \right] \omega, \quad (4.36) \]

That is

\[ k \times \dot{k} = \frac{1}{2} k \times \left[ \cot \frac{\theta}{2} (I - kk^T) - S(k) \right] \omega. \quad (4.37) \]

Since \( \dot{k} \) is the velocity of \( k \), \( k \) and \( \dot{k} \) are orthogonal vectors. And because of the following equation,

\[ k^T \left( \left[ \cot \frac{\theta}{2} (I - kk^T) - S(k) \right] \omega \right) \]
\[ = \frac{1}{2} \left[ \cot \frac{\theta}{2} (k^T - k^T kk^T) - k^T S(k) \right] \omega \]
\[ = \frac{1}{2} \left[ 0 - \left( S^T(k)k \right)^T \right] \omega \]
\[ = \frac{1}{2} \left[ -\left( -S(k)k \right)^T \right] \omega \]
\[ = 0 \quad (4.38) \]

\( k \) and \( \left[ \cot \frac{\theta}{2} (I - kk^T) - S(k) \right] \omega \) are orthogonal vectors.
From (4.37), and the orthogonal relations of $k$ and $\dot{k}$, and $k$ and $[\cot \frac{\theta}{2}(I - kk^T) - S(k)]\omega$, we can get

$$\dot{k} = \frac{1}{2}[\cot \frac{\theta}{2}(I - kk^T) - S(k)]\omega,$$  

(4.39)

which can be explained by the following example:

Let vectors $r$, $a$, $b$, $c$ in Fig. 4.2 satisfy

$$r \times a = r \times b = c,$$  

(4.40)

And we have

$$||r \times a|| = ||r \times b|| = ||c||,$$  

(4.41)

$$||r|| \ ||a|| \sin \theta_a = ||r|| \ ||b|| \sin \theta_b.$$  

(4.42)

If $\theta_a = \theta_b$, then from (4.42), we can get $||a|| = ||b||$, that is $a = b$.

In the case of (4.37), $k$ and $\dot{k}$, $k$ and $[\cot \theta (I - kk^T) - S(k)]\omega$ are orthogonal vectors, that is $\theta_a = \theta_b = 90[deg]$, so we can get the result of (4.39).

Put together (4.13) and (4.39), Angle/Axis differential equations of $\dot{\theta}$, $\dot{k}$ are shown as

$$\dot{\theta} = k^T \omega,$$  

(4.43)

$$\dot{k} = \frac{1}{2}[\cot \frac{\theta}{2}(I - kk^T) - S(k)]\omega.$$  

(4.44)

### 4.2.2 Deduction of Quaternion Differential Equation ($\dot{\eta}$, $\dot{\epsilon}$)

The unit quaternion is defined as

$$\eta = \cos \frac{\theta}{2}$$  

(4.45)
\[ \epsilon = \sin \frac{\theta}{2} k, \]  
(4.46)

where \( k(||k|| = 1) \) is the rotation axis and \( \theta \) is the rotation angle around \( k \).

Differentiating (4.45) with respect to time, we have

\[ \dot{\eta} = -\frac{1}{2} \sin \frac{\theta}{2} \dot{\theta}. \]  
(4.47)

Substituting \( \dot{\theta} \) expressed in (4.43) to (4.47), we have

\[ \dot{\eta} = -\frac{1}{2} \sin \frac{\theta}{2} \dot{\theta} \]

\[ = -\frac{1}{2} \sin \frac{\theta}{2} k^T \omega \]

\[ = -\frac{1}{2} \epsilon^T \omega \]  
(4.48)

Differentiating (4.46) with respect to time, we have

\[ \dot{\epsilon} = \frac{1}{2} \cos \frac{\theta}{2} \dot{k} + \sin \frac{\theta}{2} \dot{k}. \]  
(4.49)

Substituting \( \dot{\theta} \) and \( \dot{k} \) expressed in (4.43) to (4.49), we have

\[ \dot{\epsilon} = \frac{1}{2} \cos \frac{\theta}{2} \dot{k} + \sin \frac{\theta}{2} \dot{k} \]

\[ = \frac{1}{2} \cos \frac{\theta}{2} k^T \omega + \sin \frac{\theta}{2} \frac{\cos \frac{\theta}{2}}{2} (I - kk^T) - S(k) \omega \]

\[ = \frac{1}{2} \eta kk^T \omega + \frac{1}{2} \eta[I - kk^T] - \sin \frac{\theta}{2} S(k) \omega \]

\[ = \frac{1}{2} \eta kk^T \omega + \frac{1}{2} \eta[I - \eta kk^T - S(\epsilon)] \omega \]

\[ = \frac{1}{2} \eta[I - S(\epsilon)] \omega \]  
(4.50)

Put together (4.47) and (4.50), Quaternion differential equations of \( \dot{\eta}, \dot{\epsilon} \) are shown as

\[ \dot{\eta} = -\frac{1}{2} \sin \frac{\theta}{2} \dot{\theta}, \]  
(4.51)

\[ \dot{\epsilon} = \frac{1}{2} [\eta I - S(\epsilon)] \omega. \]  
(4.52)
5  Motion Feed-Forward Evolutionary Tracking

5.1  Motion-Feedforward (MFF) Compensation

The motion of the target being seen from the hand-eye camera is affected by both the motion of the target in the real world and the ego motion of the hand-eye camera. Here we describe such a relationship, which can distinguish these two motions in mathematical formulation.

The target coordinate system is represented as $\Sigma_M$ (see Fig. 5.1). Take $\Sigma_W$ as the reference frame. Denote the vector from the origin of $\Sigma_W$ ($O_W$) to the origin of $\Sigma_{CR}$ ($O_{CR}$) expressed in $\Sigma_W$ as $Wr_{CR}$, the vector from $O_W$ to the origin of $\Sigma_M$ ($O_M$) expressed in $\Sigma_W$ as $Wr_M$, and the vector from $\Sigma_{CR}$ to $\Sigma_M$ expressed in $\Sigma_{CR}$ as $CRr_M$. The following relations hold:

$$CRr_M = CRR_W(q)(Wr_M - Wr_{CR}(q)),$$

(5.1)

where $CRR_W$ is a rotation matrix determined by $q$. Differentiating (5.1) with respect to time, we have

$$CRr_M = CRR_W(q)(Wr_M - Wr_{CR}) + S(CR\omega_W)CRR_W(q)(Wr_M - Wr_{CR}(q)).$$

(5.2)

The angular velocities of $\Sigma_{CR}$ and $\Sigma_M$ with respect to $\Sigma_W$ are represented by $W\omega_{CR}$ and $W\omega_M$, and the angular velocity of $\Sigma_M$ with respect to $\Sigma_{CR}$ is represented by $CR\omega_M$. Then the following relation holds:

$$CR\omega_M = CRR_W(q)(W\omega_M - W\omega_{CR}).$$

(5.3)

The 3-D pose of the target is defined as

$$CR\psi_M = \begin{bmatrix} CRr_M \\ CR\epsilon_M \end{bmatrix},$$

(5.4)

where $CRr_M = [r_1, r_2, r_3]^T$, $CR\epsilon_M = [\epsilon_1, \epsilon_2, \epsilon_3]^T$. 


Then the velocity of the target’s 3-D pose is defined as

\[ CR\dot{\psi}_M = \begin{bmatrix} CR\dot{r}_M \\ CR\dot{\epsilon}_M \end{bmatrix}, \]  

(5.5)

where the time derivation of target’s position \( CR\dot{r}_M \) is given by (5.2). The relation between the time derivative of \( CR\dot{\epsilon}_M \) and the body angular velocity \( CR\omega_M \) is given by (4.11) and is rewritten as

\[ CR\dot{\epsilon}_M = \frac{1}{2}(CR\eta_M I - S(CR\epsilon_M))CR\omega_M, \]  

(5.6)

where \( CR\omega_M \) is given by (5.3).

Moreover, the camera velocity, which is considered as the end-effector velocity, can be expressed using the Jacobian matrix \( J(q) = [JP_T(q), JO_T(q)]^T \),

\[ W\dot{r}_{CR} = JP(q)\dot{q}, \]  

(5.7)

\[ W\omega_{CR} = JO(q)\dot{q}, \]  

(5.8)

\[ S^{(CR}\omega_W) = -CRR_W(q)S^{(W}\omega_{CR})WR_{CR}(q) \]  

\[ = -CRR_W(q)S(J_O(q)\dot{q})WR_{CR}(q). \]  

(5.9)

Obviously, \( CR\psi_M \) is expressed based on \( \Sigma_{CR} \) and the value is different when it is described by the left camera coordinate \( \Sigma_{CL} \). However, \( CR\psi_M \) and \( CL\psi_M \) can be converted each other by multiplying constant matrix, so we express them as \( \psi_M \) without distinction, and the object’s velocity is expressed by \( \dot{\psi}_M \).

Substituting (5.7), (5.8), (5.9) to (5.2), (5.6), and rewriting \( CR\psi_M \) as \( \psi_M \), \( CR\dot{\psi}_M \) as \( \dot{\psi}_M \),
the target velocity $\dot{\psi}_M$ can be described as (see Appendix 2):

$$
\dot{\psi}_M = \begin{bmatrix}
\dot{r}_M \\
\dot{\epsilon}_M
\end{bmatrix}
= \begin{bmatrix}
-CR_{W}(q)J_P(q) + CR_{W}(q)S(WCR(q)r_M)J_D(q) \\
-\frac{1}{2}(\eta_M I - S(e_M))CR_{W}(q)J_D(q)
\end{bmatrix} \dot{q} \\
+ \begin{bmatrix}
CR_{W}(q) \\
0
\end{bmatrix}
\frac{1}{2}[\eta_M I - S(e_M)]CR_{W}(q)W \dot{\omega}_M \\
+ \begin{bmatrix}
W \dot{r}_M \\
W \omega_M
\end{bmatrix}
J_M(q, \dot{\psi}_M)\dot{q} + J_N(q)W \dot{\phi}_M
$$

(5.10)

The matrix $J_M$ in (5.10) describes how target pose change in camera coordinate with respect to the joint velocity of the manipulator $\dot{q}$. The matrix $J_N$ describes how target pose change in camera coordinate with respect to the changing pose of the target itself in $\Sigma_{CR}$.

In this paper, we do not deal with the prediction of the target’s motion $W \dot{\phi}_M$ in the real world, and we take account of the prediction of the target velocity in $\Sigma_{CR}$ based on the joint velocity $\dot{q}$, so we can rewrite (5.10) as

$$
\dot{\psi}_M = J_M(q, \psi_M)\dot{q}.
$$

(5.11)

Then the 3-D pose of the target at future time $t + \Delta t$ can be predicted based on the motion of the end-effector at time $t$, presented by

$$
\hat{\psi}_M(t + \Delta t) = \hat{\psi}_M(t) + \dot{\psi}_M \Delta t
= \hat{\psi}_M(t) + J_M(q, \dot{\psi}_M)\dot{q} \Delta t.
$$

(5.12)

(5.12) shows $J_M$ is a function of $q$ and $\dot{\psi}_M$ (we use $\dot{\psi}_M$ since the quaternion parameters include $\eta_M$ and $e_M$ that have to be estimated by measuring the target’s pose using “1-Step GA”). $q$ can be observable correctly from the robot manipulator, while $\hat{\psi}_M(t)$ is the result of recognition at time $t$ by using model-based matching (explained in 5.2.2) in which errors exist, derived from recognition dynamics. Then the errors are included in $J_M$ from $\dot{\psi}_M(t)$ will lead to incorrect prediction and cause the recognition errors at the time $t + \Delta t$. It seems as a difficulty in 3-D pose prediction since the errors may increase drastically due to such a vicious circle as we can see in (5.12) that may amplify the errors in $\dot{\psi}_M(t)$ to those of $\psi_M(t + \Delta t)$. However, an on-line optimization method, “1-Step GA” combined with the above prediction will limit the increasing errors by correcting the recognition result through exploring nature of GA in heuristic searching behavior, which will be explained in detail in Section 5.3.
5.2 3-D Measurement Method Using Color Information

Here, we take a rectangular solid block as an example of the target object to explain the 3-D Measurement Method. The shape of the solid block is assumed to be known. Other different kinds of shape targets can also be measured by model-based matching strategy if their shape is given, for example, in 8) a model of fish is used to recognize fish in real time, and in 33) a model of human face is used for face detection.

5.2.1 Kinematics of Stereo-Vision

We utilize perspective projection as projection transformation. Figure 3.1 shows the coordinate system of our stereo vision system. The target object’s coordinate system is represented by $\Sigma_M$ and image coordinate systems of the left and right cameras are represented by $\Sigma_{IL}$ and $\Sigma_{IR}$. An $i$−th point on the target can be described using these coordinates and homogeneous transformation matrices. At first, a homogeneous transformation matrix from $\Sigma_{CR}$ to $\Sigma_M$ is defined as $^{CR}T_M$. And a arbitrary point $i$ on the target object is defined as $^{CR}r_i$ in $\Sigma_{CR}$ and as $^M r_i$ in $\Sigma_M$. Then $^{CR}r_i$ is given as,

$$^{CR}r_i = ^{CR}T_M ^M r_i. \quad (5.13)$$

Where $^M r_i$ is predetermined fixed vectors being defined by the knowledge of the object’s shape. The position vector of $i$−th point in the right image coordinate is defined as $^{IR}r_i$, and can be described by using projection matrix $P$ of camera as,

$$^{IR}r_i = P^{CR}r_i. \quad (5.14)$$

In the same way as above, using a homogeneous transformation matrix of fixed values defining the kinematical relation from $\Sigma_{CL}$ to $\Sigma_{CR}$, $^{CL}T_{CR}$, $^{CL}r_i$ is given by

$$^{CL}r_i = ^{CL}T_{CR} ^{CR}r_i. \quad (5.15)$$

As we have obtained $^{IR}r_i$, the position vector of $i$ − th point in the left image coordinate $^{IL}r_i$ is calculated by,

$$^{IL}r_i = P^{CL}r_i \quad (5.16)$$
Fig. 5.2: Definition of a solid model and left/right searching models

The pose of $\Sigma_M$ based on $\Sigma_{CR}$ has been defined as $\psi_M$, which means $^{CR}T_M$ in (5.13) is determined by $\psi_M$. Thus (5.14), (5.16) are rewritten as

$$\begin{align*}
IR_{r_i} &= f_R(\psi_M, M_{r_i}) \\
IL_{r_i} &= f_L(\psi_M, M_{r_i})
\end{align*}$$

(5.17)

The relations in (5.17) connect the arbitrary points on the object and projected points on the left and right images through a 3-D pose $\psi_M$ of the object. The measurement of $\psi_M(t)$ in real time will be solved by consistent convergence of a matching model to the target object by “1-Step GA” which will be explained in Section 5.3.

5.2.2 Model-based Matching

The 3-D model named $S$ to search for the target object of a solid block whose surfaces have different color with each other is shown on the top of Fig. 5.2. The space of coordinates on the surface of the block model is depicted as $S_{in}$, and the $k-th$ surface in $S_{in}$ is defined as a $k-th$ set $S_{in,k}(k = 1, 2, \cdots, n)$. Further, we define the projected surface of $S_{in,k}$ in 2-D left image coordinates $\Sigma_{IL}$ as $S_{L,in,k}$, described as

$$S_{L,in,k}(\psi_M) = \{^{IL}r_i \in \mathbb{R}^2 \mid ^{IL}r_i = f_L(\psi_M, M_{r_i}), M_{r_i} \in S_{in,k} \in \mathbb{R}^3\}$$

(5.18)

Let $m$ ($m<n$) denotes the number of the visible surfaces. Then the whole projected
visible surface space in $\Sigma_{IL}$ is defined as $S_{L,in}$, which include every projected visible surface, described by

$$S_{L,in}(\psi_M) = \bigcup_{k=1}^{m} S_{L,in,k}(\psi_M)$$

(5.19)

The outside space enveloping $S_{in}$ is denoted as $S_{out}$, and the $k-th$ space in $S_{out}$ is defined as a $k-th$ set $S_{out,k}(k = 1, 2, \cdots, n)$. Similarly, we define the projected space of $S_{out,k}$ in $\Sigma_{IL}$ as $S_{L,out,k}$, described as

$$S_{L,out,k}(\psi_M) = \{ II_{r_i} \in \mathbb{R}^2 | II_{r_i} = f_L(\psi_M, M_{r_i}), M_{r_i} \in S_{out,k} \in \mathbb{R}^3 \}$$

(5.20)

Then the whole projected outside visible space in $\Sigma_{IL}$ is defined as $S_{L,out}$, given by

$$S_{L,out}(\psi_M) = \bigcup_{k=1}^{m} S_{L,out,k}(\psi_M)$$

(5.21)

The left 2-D searching model named $S_L$, including $S_{L,in}$ and $S_{L,out}$, is shown on the left bottom of Fig. 5.2, where visible surfaces have three different colors of red, blue, green. The projection onto the right image coordinates $\Sigma_{IR}$ is in the same way. And the right 2-D searching model named $S_R$ is shown on the right bottom of Fig. 5.2.

We suppose there are many solid models in the searching area, each has its own pose $\psi_M$. To determine which solid model is most close to the real target, a fitness function is defined for the evaluation of correlation between them. The input images will be directly matched by the projected moving models $S_L$ and $S_R$, which are located by only $\psi_M$ as described in (5.20) that includes the kinematic relations of the left and right camera coordinates. Therefore, if the camera parameters and kinematic relations are completely accurate, and the solid searching model describes precisely the target object shape, then the $S_{L,in}$ and $S_{R,in}$ will completely lie on the target reflected on the left and right images, provided that the model’s pose coincide to the true value of $\psi_M$ of the target object.

Here, we use color information to search for the target object in the images. Let $b_k, (k = 1, 2, \cdots, n)$ denote the hue value of the color in $S_{in,k}$. Let $h(II_{r_i})$ denote the hue value of the searching models at the image position $II_{r_i}$. Then the evaluation function of the left moving surface-strips model is given as,

$$F_L(\psi_M) = \frac{1}{\lambda} \sum_{k=1}^{m} \left( \sum_{II_{r_i} \in S_{L,in,k}(\psi_M)} \delta(h(II_{r_i}) - b_k) - \sum_{II_{r_i} \in S_{L,out}(\psi_M)} \delta(h(II_{r_i}) - b_k) \right)$$

(5.22)
where \( \delta \) is the Kronecker delta function defined as

\[
\delta(n) = \begin{cases} 
1 & n = 0 \\
0 & n \neq 0.
\end{cases}
\] (5.23)

Here \( \Lambda = \sum_{k=1}^{m} n_k \), where \( n_k \) represents the number of the searching points in \( S_{L,in,k} \). \( \Lambda \) is a scaling factor to normalize as \( F_L(\psi_M) \leq 1 \). In the case of \( F_L(\psi_M) < 0 \), \( F_L(\psi_M) \) is given to zero. The first part of this function (5.22) expresses how much each color area of \( S_{L,in} \) defined by \( \psi_M \) lies on the target being imaged on the left camera. And the second part is the matching degree of its contour-strips. The difference between the internal surface and the contour-strips of the surface-strips model can make the estimation more sensible, especially in recognition of distance between the target to the cameras that determine the size of the projected models to left and right images.

The evaluation function of the right moving surface-strips model is defined in the same way. We define the whole evaluation function as

\[
F(\psi_M) = F_L(\psi_M) + F_R(\psi_M) \quad \text{2}
\] (5.24)

Equation (5.24) is used as a fitness function in GA process. When the moving searching model fits to the target object being imaged in the right and left images, then the fitness function \( F(\psi_M) \) gives maximum value.

Therefore the problem of finding a target object and detecting its pose can be converted to searching \( \psi_M \) that maximizes \( F(\psi_M) \). We solve this optimization problem by GA, which will be explained in the next section. The genes of GA representing possible pose solution is defined as

\[
\begin{align*}
& t_x \quad t_y \quad t_z \quad \epsilon_1 \quad \epsilon_2 \quad \epsilon_3 \\
& \quad \text{12bit} \quad \text{12bit} \quad \text{12bit} \quad \text{12bit} \quad \text{12bit} \quad \text{12bit}
\end{align*}
\]

The 72 bits of gene refers to the range of the searching area: \(-150 \leq t_x \leq 150[mm] \), \( 0 \leq t_y \leq 300[mm] \), \( 650 \leq t_z \leq 950[mm] \), and \(-0.5 \leq \epsilon_1, \epsilon_2, \epsilon_3 \leq 0.5 \) which represents almost the same range of \(-60 \leq \text{roll, pitch, yaw} \leq 60[\text{deg}] \).

5.3 On-line Evolutionary Recognition “1-Step GA”

Theoretically optimal pose \( \psi_{max}(t) \) that gives the highest peak of \( F(\psi(t)) \) is defined as

\[
\psi_{max}(t) = \{ \psi(t) \mid \max_{\psi \in L} F(\psi(t)) \}
\] (5.25)
where $L$ represents 6-DoF searching space of $x, y, z, \epsilon_1, \epsilon_2, \epsilon_3$.

An individual of GA is defined as $\psi_i^j(t)$, which means the $i$-th gene ($i = 1, 2, \cdots, p$) in the $j$-th generation, to search $\psi_{max}(t)$. Denote $\psi_{GA}^{max}(t)$ to be the maximum among the $p$ genes of $\psi_i^j(t)$ in GA process,

$$
\psi_{max}^{GA}(t) = \{ \psi_i^j(t) \mid \max_{\psi_i^j \in L} F(\psi_i^j(t)) \}. \quad (5.26)
$$

In fact we cannot always guarantee the best individual of GA $\psi_{max}^{GA}(t)$ should coincide with the theoretically optimal pose $\psi_{max}(t)$, because the number of GA’s individuals is not infinite. The difference between $\psi_{max}(t)$ and $\psi_{max}^{GA}(t)$ is denoted as

$$
\delta\psi(t) = \psi_{max}(t) - \psi_{max}^{GA}(t). \quad (5.27)
$$

And the difference between $F(\psi_{max}(t))$ and $F(\psi_{max}^{GA}(t))$ is denoted as

$$
\Delta F(\delta\psi(t)) = F(\psi_{max}(t)) - F(\psi_{max}^{GA}(t)), \quad (5.28)
$$

Since $F(\psi_{max}(t)) \geq F(\psi_{max}^{GA}(t))$, we have

$$
\Delta F(\delta\psi(t)) \geq 0. \quad (5.29)
$$

Based on the definition of $\Delta F(\delta\psi(t))$ in (5.28), in this research, we let GA work in the following way:
(a) GA evolves to minimize $\Delta F(\delta \psi(t))$.

(b) The elitist individual of GA is preserved at every generation (elitist gene preservation strategy).

(c) $\psi_{\text{max}}^{GA}(t)$ does keep the same value in the evolving when the evolved new gene with different value gives the same value of $\Delta F$.

Here, we present two assumptions.

[Assumption 1] $\Delta F(\delta \psi(t))$ is positive definite.

This means the distribution of $F(\psi(t))$ satisfies $\Delta F(\delta \psi(t)) = 0$ if and only if $\delta \psi(t) = 0$, which indicates $\Delta F(\delta \psi(t)) = 0$ has a sole minimum at $\delta \psi(t) = 0$ over the searching space $L$, even though $\Delta F$ is multi-peak distribution having peaks and bottoms with limited number. When the model overlap the target object in the image, then the situation can make $\Delta F$ have the sole minimum in $L$. $0 \leq F(\psi(t)) \leq 1$, since $F(\psi(t))$ is normalized to be less than 1 and negative value to be set as zero by a definition of correlation function $F(\psi(t))$ (38).

So the fitness function is always less than 1 except only one point which means the $\psi_{\text{max}}^{GA}(t)$ can express the target object’s pose, as shown in Fig. 5.3(a). From (5.28), we can see when $\psi_{\text{max}}^{GA}(t) = 1$, $\Delta F(\delta \psi(t)) = 0$ (Fig. 5.3(b)), which means that only in this case, $\psi_{\text{max}}^{GA}(t)$ can express the actual pose of the target object.

[Assumption 2] $\dot{F}(\psi_{\text{max}}^{GA}(t)) \geq 0$.

This means GA evolves itself to get a bigger fitness function value ($\dot{F}(\psi_{\text{max}}^{GA}(t)) > 0$) or keep a same value ($\dot{F}(\psi_{\text{max}}^{GA}(t)) = 0$). It is not only an assumption but also the character of GA if the target object is static, because the elitist individual is preserved in every generation of GA. However, when the target object is moving, $\dot{F}(\psi_{\text{max}}^{GA}(t)) \geq 0$ will indicate that the convergence speed to the target in the dynamical images should be faster than the moving speed of the target object. Furthermore, with the pose tracking in dynamic scene being input at a certain video rate, this assumption means that $F(\psi_{\text{max}}^{GA}(t))$ have the tendency of approaching to $F(\psi_{\text{max}}^{GA}(t))$, and $\psi_{\text{max}}^{GA}(t)$ moves toward $\psi_{\text{max}}^{GA}(t)$ in each period of the input image, or keeps a distance to $\psi_{\text{max}}^{GA}(t)$. Since in this paper we think that the object’s motion is enough slow comparing the calculation speed of GA’s evolving to find $F(\psi_{\text{max}}^{GA}(t))$ from the view point that the one image be input every input video period and evolving iterations
in input video period are enough to catch up with the $F(\psi_{\text{max}}(t))$ being stationary during the input video period.

Differentiating (5.28) by time $t$, we have

$$\Delta \dot{F}(\delta\psi(t)) = \dot{F}(\psi_{\text{max}}(t)) - \dot{F}(\psi_{\text{max}}^{GA}(t)).$$

(5.30)

We defined $F(\psi_{\text{max}}(t)) = 1$ representing that the true pose of the target object gives the highest peak. Therefore, the time differentiation of $F(\psi_{\text{max}}(t))$ will be $\dot{F}(\psi_{\text{max}}(t)) = 0$. Thus, from (5.30) and [Assumption 2], we have

$$\Delta \dot{F}(\delta\psi(t)) = -\dot{F}(\psi_{\text{max}}^{GA}(t)) \leq 0.$$  

(5.31)

$\psi_{\text{max}}^{GA}(t)$ represents current best GA solution. [Assumption 2] means GA can change its best gene $\psi_{\text{max}}^{GA}(t)$ to always reduce the value of $\Delta F$ regardless of dynamic image or static one, which indicates that the convergence speed to the target in the dynamically continuous images should be faster than the moving speed of the target object.

We cannot guarantee that the above two assumptions always hold, since they depend on some factors such as object’s shape, object’s speed, definition of $F(\psi(t))$, parameters of GA and viewpoint for observing, lightening environment, computer’s performance et al. However, we can make efforts to improve the environment and correlation function and so on. Providing above two assumptions be satisfied, (5.29) and (5.31) hold, then $\Delta F(\delta\psi(t))$ is so-called Lyapunov function. The objective here is to verify that $\delta\psi(t)$ asymptotically stable, resulting in it converges to 0 by using the Lyapunov function of $\Delta F(\delta\psi(t))$, meaning $\psi_{\text{max}}^{GA}(t) \rightarrow \psi_{\text{max}}(t), \ (t \rightarrow \infty)$, and the following shows how to verify it.

Since $\Delta \dot{F}(\delta\psi(t))$ is only negative semi-definite, in the view of LaSalle theorem, $\delta\psi(t)$ asymptotically converges to the invariant set of the solutions $\delta\psi$ satisfying $\Delta \dot{F}(\delta\psi(t)) = 0$.

Considering the following expression,

$$\Delta \dot{F}(\delta\psi(t)) = \frac{\partial \Delta F}{\partial \delta\psi} \cdot \delta\dot{\psi} = 0,$$

(5.32)
Equation (5.32) shows the invariant set of the solutions of $\Delta \dot{F}(\delta \psi(t)) = 0$ includes (1): $P_1$, the solution set of $\partial \Delta F / \partial \delta \psi = 0$; (2): $P_2$, the solution set of $\delta \dot{\psi} = 0$; and (3): $P_3$, the solution set satisfying $\partial \Delta F / \partial \delta \psi \neq 0$, $\delta \dot{\psi} \neq 0$, but their inner product is 0.

As shown in Fig. 5.4, $P_1$ includes the points of $\delta \psi$ that give the local maximum or minimum values of the function $\Delta F$ including $0$. The number of these points is finite by [Assumption 1] denoted by $p$, that is

$$P_1 = \{0, \delta \psi_1, \delta \psi_2, \cdots, \delta \psi_{p-1}\}. \quad (5.33)$$

Concerning (4.1), the evolving process of GA may stay temporarily at the same $\Delta F$ value. If the target object is static, it means the best gene of GA stops at some moments for the reason that the limited individuals of GA could not improve a current solution that gives a smaller fitness function value $\Delta F$ during some generations. And when the target object is moving, $\delta \dot{\psi} = 0$ means at these moments that the evolution speed of the best gene of GA is equal to the moving speed of the target object, by (5.27). The number of these points is
assumed to be possibly finite, denoted by \( q \). Thus, we describe the set of \( P_2 \) as

\[
P_2 = \{0, \delta \psi_{G1}, \delta \psi_{G2}, \ldots \delta \psi_{G(q-1)}\}.
\] (5.34)

Notice that there is another solution set of \( \delta \psi \): \( P_3 \). In this case, the vector of \( \partial \Delta F / \partial \delta \psi \) is vertical to the vector of \( \delta \dot{\psi} \) since the calculation \( (\Delta F / \delta \psi) \cdot \delta \dot{\psi} \) in (5.32) means inner cross product, which means GA evolves in the direction that keeps a same fitness function value \( \Delta F \). This GA’s evolution way is forbidden in this research for the GA work rule \((c)\) that we have stated above. Then, \( P_3 \) is null. So the invariant set that \( \delta \psi(t) \) asymptotically converges to is

\[
P = P_1 \bigcup P_2.
\] (5.35)

Here, \( \delta \psi_1, \delta \psi_2, \ldots, \delta \psi_{p-1} \) in \( P_1 \) are all unstable since \( \delta F(\delta \psi_i) > 0 \) \((i = 1, 2, \ldots, p - 1)\), and only \( \delta \psi = 0 \) is stable from [Assumption 1], since when \( t \to \infty \) there should always remain the possibility to get out of local maximum/minimum of \( \delta \psi_1 \cdots \delta \psi_{p-1} \) And all the points in \( P_2 \) except the point \( 0 \) are unstable because GA has possibility to get out of these points by its evolving nature. Therefore, \( 0 \) is the only stable point in the invariant set of \( P \), that is, \( \delta \psi(t) \) will finally converges to \( 0 \). The image of the changing of \( \Delta F(\delta \psi(t)) \) with respect to time \( t \) in the whole GA’s evolution is shown in Fig.5.5.

The above verification shows \( \delta \psi(t) \to 0 \), which means

\[
\psi_{\text{max}}^G(t) \longrightarrow \psi_{\text{max}}(t), \quad (t \to \infty)
\] (5.36)

Let \( t_e \) denotes a convergence time, then

\[
|\delta \psi(t)| = |\psi_{\text{max}}(t) - \psi_{\text{max}}^G(t)| \leq \epsilon, \quad (\epsilon > 0, t \geq t_e)
\] (5.37)

In (5.37), \( \epsilon \) is tolerable extent that can be considered as a observing error. Thus, it is possible to realize real-time optimization, because \( \psi_{\text{max}}^G(t) \) can be assumed to be in the vicinity of the theoretically optimal \( \psi_{\text{max}}(t) \) after \( t_e \).

Above discussion is under the condition of continuous time. Here, when we consider evolution time of each generation of GA denoted by \( \Delta t \). The GA’s evolving process is described as

\[
\psi_i^j(t) \xrightarrow{\text{evolve}} \psi_i^{j+1}(t + \Delta t).
\] (5.38)
Obviously, this time-discrete evolution with the interval of time $\Delta t$ may enlarge the recognition error $\delta \psi(t)$. Should this undesirable influence of $\Delta t$ be considered, the tolerable pose error $\epsilon$ will expand to $\epsilon'$ as,

$$|\delta \psi(t)| \leq \epsilon', \quad (\epsilon' > \epsilon > 0). \quad (5.39)$$

Since the GA process to recognize the target’s pose at the current time is executed at least one time with the period of $\Delta t$ as the current quasi-optimal pose $\psi_{\text{max}}^{GA}(t)$ is output synchronously, we named this on-line recognition method as “1-step GA”. We have confirmed that the above real-time optimization problem could be solved by “1-step GA” through several experiments to recognize swimming fish $^{32}$ and human face $^{33}$.

### 5.4 “1-Step GA + MFF” Method

Here, we use $p$ individuals for searching. The best one in $p$ individuals in $j$-th generation at time $j\Delta t$ represented by $\psi_{\text{max}}^j(j\Delta t)$, is denoted as $\hat{\psi}_M(j\Delta t)$ from here, which represents
the measured pose of the target object, \( \hat{\psi}_M^j \) is described by

\[
\hat{\psi}_M^j = \{ \psi_i^j(j \Delta t) | \max_{i=1,2,\cdots,p} F(\psi_i^j(j \Delta t)) \}, \quad (5.40)
\]

We define the individual of GA as \( \psi_i^{*j} \) in the case of using MFF method to predict to distinguish from \( \psi_i^j \) used in the case of not using MFF method. And the best gene in \( p \) individuals of \( \psi_i^{*j} \) is defined by \( \hat{\psi}_M^{*j} \) to make a difference to \( \hat{\psi}_M^j \). \( \hat{\psi}_M^{*j} \) is described by

\[
\hat{\psi}_M^{*j} = \{ \psi_i^{*j}(j \Delta t) | \max_{i=1,2,\cdots,p} F(\psi_i^{*j}(j \Delta t)) \}. \quad (5.41)
\]

Using the prediction of (5.12), the pose of the individuals \( \psi_i^{*j+1} \) in the next generation can be predicted based on the current pose \( \hat{\psi}_M^{*j} \), presented by

\[
\psi_i^{*j+1} = \psi_i^{*j} + J_M(q, \hat{\psi}_M^{*j}) \dot{q} \Delta t, \quad (i = 1, 2, \cdots, p). \quad (5.42)
\]

Here, \( q \) can be considered to be observable correctly from the robot manipulator, while \( \hat{\psi}_M^{*j} \) is the result of recognition at time \( t \) in which errors exist, stemming from recognition dynamics. Then the errors included in \( J_M \) through \( \hat{\psi}_M^{*j} \) will lead to incorrect prediction. However, the proposed “1-Step GA” combined with the above prediction will limit the increasing errors with the evolution process exploring possible solution space in heuristic and predictive manner and correcting the recognition result.

Through “1-step GA”, the measured pose of the target object in the next time at \((j+1)\Delta t\) is given by \( \psi_M^{*j+1} \). The recognition system of the proposed method is shown in Fig. 5.7. The
proposed MFF method can predict the motion of the target projected to the cameras based on the ego motion of the robot. So when the individuals of GA got converged, the whole group of genes $\psi^j_i$ will move together with the motion of the target in the image, never loose it even under a camera’s ego motion of robot manipulator. Thus, recognition by hand-eye cameras will be independent of the dynamical motion of the manipulator, then robust recognition can be expected as the same performance as using fixed cameras.
6  On-line Pose Evolutionary Tracking with Hand-eye Configuration

6.1 Robot Model and Control System

Mitsubishi PA-10 robot arm is a 7-DoF robot arm manufactured by Mitsubishi Heavy Industries. In this paper, we constructed a simulator that includes full model of the PA-10 (see Fig. 1.1) to discuss the effectiveness of the MFF method that explained in Chapter 5. The following is the explanation about the dynamics of manipulator, and the parameters of the PA-10 included in the following dynamical model are given in Table 6.1.

The general equation of motion of manipulator is

$$M(q)\ddot{q} + h(q, \dot{q}) + g(q) = \tau,$$

(6.1)

where, $q$: the joint displacement and $q = [q_1, q_2, \cdots, q_7]^T$, $\tau$: the joint driving force and $\tau = [\tau_1, \tau_2, \cdots, \tau_7]^T$, $M(q)$: the inertia matrix, $h(q, \dot{q})$: the vector representing the centrifugal and coriolis forces, $g(q)$: the vector representing the gravity load.

The $\tau$ is decided by PD control,

$$\tau = D_p(q_d - q) + D_d(\dot{q}_d - \dot{q}),$$

(6.2)

where $D_p$ and $D_d$ are gains of controller and are positive definition diagonal matrices. The PD controller is simple but effective, so it has usually been used in industrial robot such as PA-10 we used in this paper. The effectiveness has been proved by S. Arimoto\(^{36}\) that for any initial state $q(0), \dot{q}(0)$, the PD control system will converge globally and asymptotically to constant $q_d$ as $t \to \infty$.

We establish world coordinate system $\Sigma_W$, end-effector coordinate system $\Sigma_E$ and camera coordinate system $\Sigma_{CR}$(see Fig. 1.1). $\Sigma_E$ is assumed the same as $\Sigma_{CR}$ since the camera is mounted on the robot’s end-effector.

Here, the orientation of the end-effector is represented by unit quaternion $\{W\eta_E, W\epsilon_E\}$. Since $W\eta_E$ can be determined by $W\epsilon_E$ through (4.4), we use only three parameters $W\epsilon_E$ to
express the end-effector’s orientation. So the position/orientation of end-effector is \( W\psi_E = \left[ W r_E; W \epsilon_E \right]^T \), where \( W r_E = [r_{1E}, r_{2E}, r_{3E}]^T \), \( W \epsilon_E = [\epsilon_{1E}, \epsilon_{2E}, \epsilon_{3E}]^T \).

Given a desired trajectory of end-effector, \( W\psi_{Ed}(t), \ t \in [0, T] \), desired joint displacement \( q_d \) in (6.2) that realizes the trajectory \( W\psi_{Ed} \) can be determined by solving the inverse kinematics problem,

\[
q_d = f^{-1}(W\psi_{Ed}).
\]  

(6.3)

The desired joint velocity \( \dot{q}_d \) in (6.2) is determined by using the pseudo-inverse Jacobian matrix \( J^+(q) \) since PA-10 is a 7-link redundant manipulator.

\[
\dot{q}_d(t) = J^+(q)W\dot{\psi}_{Ed}(t).
\]  

(6.4)

where \( J^+(q) = J^T(JJ^T)^{-1} \).

### 6.2 Simulation Condition

To verify the effectiveness of the proposed MFF recognition method, we conduct the simulation experiments to recognize a rectangular solid block \((100mm \times 150mm \times 200mm)\) with symmetrical colored surfaces, shown in Fig. 6.1(a). We compare the recognition result when using just “1-step GA” with that using “1-step GA + MFF” under a given trajectory of the end-effector with dynamical oscillation. To see clearly the result of proposed method, here, we keep the target object static, so the target motion in the camera view is purely generated by the motion of the camera, whose angular frequency of the camera motion in Fig. 6.1(b) is given by \( \omega \).

<table>
<thead>
<tr>
<th>Joint</th>
<th>Base</th>
<th>Link1</th>
<th>Link2</th>
<th>Link3</th>
<th>Link4</th>
<th>Link5</th>
<th>Link6</th>
<th>Link7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length [m]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.200</td>
<td>0.115</td>
<td>0.307</td>
<td>0.143</td>
<td>0.225</td>
<td>0.245</td>
<td>0.080</td>
<td>0.020</td>
<td></td>
</tr>
<tr>
<td>Center of mass (m)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0750</td>
<td>−0.0518</td>
<td>0.0633</td>
<td>0.0536</td>
<td>0.0461</td>
<td>0.0803</td>
<td>−0.0186</td>
<td>0.0040</td>
<td></td>
</tr>
<tr>
<td>Inertia moment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I_{xx} [Kgm^2]</td>
<td>N/A</td>
<td>0.0123</td>
<td>0.0686</td>
<td>0.0370</td>
<td>0.0279</td>
<td>0.0407</td>
<td>0.0109</td>
<td>0.00250</td>
</tr>
<tr>
<td>Inertia moment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I_{yy} [Kgm^2]</td>
<td>N/A</td>
<td>0.0636</td>
<td>0.0686</td>
<td>0.0262</td>
<td>0.0279</td>
<td>0.00583</td>
<td>0.0109</td>
<td>0.00250</td>
</tr>
<tr>
<td>Inertia moment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I_{zz} [Kgm^2]</td>
<td>N/A</td>
<td>0.123</td>
<td>0.0119</td>
<td>0.0370</td>
<td>0.00648</td>
<td>0.0407</td>
<td>0.000697</td>
<td>0.000174</td>
</tr>
</tbody>
</table>
Simulation experiments are performed using a software “Open GL” as a display tool in our simulation system. Here, a manipulator’s model of the actual 7-link “PA-10” is used, and two cameras are mounted on its end-effector, as shown in Fig. 6.1. The dynamics of the PA-10 simulator is the same with the actual manipulator as shown in Section 3 by using the real physical parameters of the PA-10 robot in Table 6.1.

A trajectory of end-effector is given as a circle with a fixed distance to the target and keeping the eye-line (z axis of Σ_E) passes the center of the target. The initial hand pose is defined as Σ_{E_0} as shown in Fig. 6.1(b) and the homogeneous transformation matrix from Σ_{E_0} to Σ_W is set as,

\[
W_0 T_{E_0} = \begin{bmatrix}
0 & 0 & 1 & 918 \\
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 455 \\
0 & 0 & 0 & 1 \\
\end{bmatrix},
\]

(6.5)

The desired hand trajectory expressed in Σ_{E_0} is

\[
E_0 \psi_{Ed} = \begin{cases}
E_0 x_{Ed}(t) = d * sin(\theta_d(t)) \\
E_0 y_{Ed}(t) = 0 \\
E_0 z_{Ed}(t) = d - d * cos(\theta_d(t)) \\
E_0 e_1_{Ed}(t) = 0 \\
E_0 e_2_{Ed}(t) = sin(\theta_d(t)/2) \\
E_0 e_3_{Ed}(t) = 0
\end{cases}
\]

(6.6)

where \( d = 800[mm] \), \( \theta_d(t) = 15sin(\omega t)[deg] \), \( \omega \) represents the frequency of end-effector’s motion. By using (6.6), (6.5), the desired hand trajectory can be expressed in Σ_W as

\[
W T_{Ed}(t) = W_0 T_{E_0} E_0 T_{Ed}(t).
\]

(6.7)
Fig. 6.2: Desired hand trajectory $E_0\psi_{Ed}(t)$ and the actual trajectory $E_0\psi_E(t)$ with dynamics, $\omega = 0.105[rad/s]$. The desired hand pose and the actual hand pose in $\Sigma_{E_0}$ obtained by PD control are shown in Fig. 6.2 when ($\omega = 0.105[rad/s]$). It can be found that there is oscillation in the actual hand trajectory at the first 30 seconds due to transient response made by the dynamics of the manipulator, which would make the recognition of the object more difficult because the hand-eye camera is moving together with the manipulator. The effectiveness of MFF method to solve this kind of problem will be evaluated in the following simulation, in which the target object is assumed to be static in $\Sigma_W$, set as $E\psi_M(0) = [0, 155[mm], 800[mm], 0, 0, 0]^T$.

### 6.3 Error Definition

The maximum value of the fitness function $F$ in $j$ generation is defined as $F^j_M$, then $\bar{F}$ is given by

$$\bar{F} = \frac{1}{n} \sum_{j=1}^{n} F^j_M(\hat{\psi}^j_M(t_j)) \quad (6.8)$$

Let $\Delta \psi_M$ describes the pose error, which is defined as the difference between the desired value and the recognized value, $\Delta \psi_M = [\psi_M - \hat{\psi}_M] = [\Delta x_M, \Delta y_M, \Delta z_M, \Delta \varepsilon_1 M, \Delta \varepsilon_2 M, \Delta \varepsilon_3 M]^T$. Then the root-mean-square (rms) value of the pose error is given by

$$\Delta \tilde{\psi}_M = [\Delta \tilde{x}_M, \Delta \tilde{y}_M, \Delta \tilde{z}_M, \Delta \tilde{\varepsilon}_1 M, \Delta \tilde{\varepsilon}_2 M, \Delta \tilde{\varepsilon}_3 M]^T, \quad (6.9)$$
We compare the methods of “1-step GA” and “1-step GA + MFF” by changing the moving speed of hand-eye through \( \omega = 0.105 [rad/s] \). The true values of the 3-D pose of the target object in \( \Sigma_E \) are \( \psi_M \). The recognition results using only “1-step GA” without MFF method are represented by \( \hat{\psi}_M \), and the recognition results using both “1-step GA” and MFF are denoted by \( \hat{\psi}_M^* \). Due to

where

\[
\Delta \hat{p}_M = \sqrt{\frac{1}{n} \sum_{j=1}^{n} (\Delta p_M(t_j))^2}, (p = x, y, z, \epsilon_1, \epsilon_2, \epsilon_3). \tag{6.10}
\]

It is obvious that high value of \( F \) and small value of \( \Delta \hat{\psi}_M \) represent good recognition. Here, we use millimeter to measure position. When using quaternion to express the orientation of an object, no unit, just values. The object rotation of 1[deg] around \( x \) axis corresponding to quaternion representation as \( \epsilon_1 = 0.008, \epsilon_2 = 0, \epsilon_3 = 0 \).

### 6.4 Simulation Result

We compare the methods of “1-step GA” and “1-step GA + MFF” by changing the moving speed of hand-eye through \( \omega = 0.105, 0.157, 0.314 [rad/s] \), corresponding to the periods of hand-eye, \( T = 60, 40, 20 [s] \). As you see in Fig. 6.2 the camera motion with the condition of \( \omega = 0.105 [rad/s] \) has oscillations, which disturbs the recognition. Please notice that bigger \( \omega \) generate shorter motion period \( T \), and the motion in Fig. 6.2 is smallest.

Fig. 6.3 shows comparison of the methods “1-step GA” and “1-step GA + MFF” under \( \omega = 0.105 [rad/s] \). The true values of the 3-D pose of the target object in \( \Sigma_E \) are \( \psi_M \). The recognition results using only “1-step GA” without MFF method are represented by \( \hat{\psi}_M \), and the recognition results using both “1-step GA” and MFF are denoted by \( \hat{\psi}_M^* \). Due to
the dynamics of the manipulator, the target object in the images includes the motion caused by hand oscillation. Fig. 6.3(a), (b) show the obvious transient oscillation during the first 30 seconds. Such oscillations surely bring difficulty to object recognition and the oscillations are increasing when the speed of the end-effector becomes faster (see later simulations in this section).

\( \dot{\psi}_M(t) \) is true value calculated from \( E_0 \dot{\psi}_{Ed} \) in (6.6), and all components of \( \psi_M(t) \) are depicted by white lines in Fig. 6.3. Comparing the recognition results using “1-step GA” and “1-step GA + MFF” in Fig. 6.3, we find that “1-Step GA” method can not recognize precisely, especially during the first 30[s] oscillation period, which can be seen from the result that the solid lines representing \( \dot{\psi}_M(t) \) without MFF is not overlapping the true values of \( \psi_M(t) \). On the other hand, “1-step GA + MFF” method gives more correct result despite the oscillation since the dotted lines representing \( \dot{\psi}_M^*(t) \) using MFF overlaps the true values of \( \psi_M(t) \).

Table 6.2 shows the mean value \( \bar{F} \) of the fitness function \( F \) and \( \Delta \dot{\psi}_M \) defined by (6.9) indicating rms value of all components of \( \Delta \dot{\psi}_M \) with separating “1-step GA” in the upper half and “1-step GA + MFF” in the lower half and with categorizing \( \omega = 0.105, 0.157, 0.314 \text{[rad/s]} \) in three columns. The results in Fig. 6.3 with the condition of \( \omega = 0.105 \text{[rad/s]} \) are listed in

---

Fig. 6.4: Comparison of the recognition by “1-step GA” \( \dot{\psi}_M \) and by “MFF + 1-step GA” \( \dot{\psi}_M^* \) under \( \omega = 0.157 \text{[rad/s]} \).
Fig. 6.5: Comparison of the recognition by “1-step GA” \( \hat{\psi}_M \) and by “MFF + 1-step GA” \( \hat{\psi}^*_M \) under \( \omega = 0.314 \text{[rad/s]} \).

We can see that \( \bar{F} \) is lower and \( \Delta \hat{\psi}_M \) is bigger in the case of using only “1-step GA” method than that of “1-step GA + MFF” method. Especially, \( \Delta \bar{z}_M \) is about 14[mm] by using just “1-step GA”, which is much bigger than using “1-step GA + MFF” method of only about 3[mm]. Such a big position error, 14[mm], indicates the “1-step GA” could not track the motion of the target object because of the oscillation of the hand-eye. And the small error in the case of using “1-step GA + MFF” has confirmed that MFF method can compensate the fictional target motions in the camera view induced by the hand-eye’s motion, then the recognition became robust to the dynamics of the manipulator. In other words, recognition by hand-eye cameras can be independent of the dynamical motion of the manipulator, and robustness in recognition of our method is confirmed to be just like using fixed cameras by MFF.

Simulation results of comparisons of the methods “1-step GA” and “1-step GA + MFF” under \( \omega = 0.157 \text{[rad/s]} \) are shown in Fig. 6.4. Tracking of the target by “1-step GA” became more difficult when the speed of the end-effector has gotten faster, which caused the decrease of GA’s convergence speed, stemming from the increase of the target speed relative to the camera. However, using MFF recognition method, the data shown in Fig. 6.4 indicates the
models have kept matching the target well. It was verified the proposed method leads to robustly accurate recognition under end-effector motion.

When the velocity of the end-effector gets more quicker, \( \omega = 0.314 [\text{rad/s}] \), “1-step GA” lost the target soon since there are so large amplitudes of oscillation of solid lines, as shown in Fig. 6.5. On the other hand, it can be found that the recognition results of “1-step GA + MFF” always overlap the real position/orientation even under such a high-speed motion of the hand-eye.

Table 6.2 shows \( \bar{F} \) and \( \Delta \tilde{\psi}_M \) of each situation we have discussed above. We can see that using only “1-step GA ”, \( \bar{F} \) gets lower from 0[s] to 40[s], and \( \Delta \tilde{\psi}_M \) gets bigger to about 33[mm] in \( \Delta \tilde{z}_M \), and 0.0656 in \( \Delta \tilde{\psi}_{2M} \) (corresponding to 12[deg] in this situation) when \( \omega \) is 0.3214[rad/s], which means the “1-step GA” became more difficult to track the motion of the target object when the oscillation of the end-effector being harder. On the other hand, by using “1-step GA + MFF”, both \( \bar{F} \) and \( \Delta \tilde{\psi}_M \) are not changed much, about 5[mm] in \( \Delta \tilde{z}_M \) and 0.0098 in \( \Delta \tilde{\psi}_{2M} \) (corresponding to 2[deg] in this situation), which indicates the “1-step GA” was tracking the target without time delay, because the end-effector’s motion has been compensated, even the hand-eye cameras move faster and faster. Furthermore, \( \bar{F} \) in
case of $\omega = 0[rad/s]$, the hand-eye does not move, i.e., stationary recognition experiment, is 0.9863, representing that the recognition performance of “1-step GA + MFF” during hand-eye motion is almost equal to the performance under the stationary recognition condition. It has confirmed that by using MFF recognition method, the recognition in a dynamic hand-eye system is invariant, just like using a fixed camera system.
7 6-DoF Visual Servoing Experiment

To verify the effectiveness of the proposed visual servoing system, we conduct the experiment of visual servoing to a moving a 3D marker that is composed of a red ball, a green ball and a blue ball. The radiuses of these three balls are set as 30[mm].

7.1 Controller

7.1.1 Desired-trajectory Generation

As shown in Fig. 7.1, the world coordinate frame is denoted by $\Sigma_W$, the target coordinate frame is denoted by $\Sigma_M$, and the desired and actual end-effector coordinate frame is denoted by $\Sigma_{Ed}$, $\Sigma_E$ respectively. The desired relative relation between the target and the end-effector is given by Homogeneous Transformation as $^{Ed}T_M$, the relation between the target and the actual end-effector is given by $^ET_M$, then the difference between the desired end-effector pose $\Sigma_{Ed}$ and the actual end-effector pose $\Sigma_E$ is denoted as $^ET_{Ed}$, and calculated by:

$$^ET_{Ed}(t) = ^ET_M(t)^{Ed}T_M^{-1}(t) \quad (7.1)$$

$(7.1)$ is a general deduction that satisfies arbitrary object motion $^WT_M(t)$ and arbitrary visual servoing objective $^{Ed}T_M(t)$. However, the relation $^ET_M(t)$ is only observed by cameras using the on-line model-based recognition method and 1-step GA $^9)$. Let $\Sigma_M$ denote the detected object, there always exist an error between the actual object $\Sigma_M$ and the detected one $\Sigma_M$. However, in visual servoing we use different methods to decrease this error. for example, we can limit the error inside 5[mm] in $^{34)}$. So in visual servoing, $(7.1)$ will be rewritten based on $\Sigma_M$ that includes the error $^MT_M$, as

$$^ET_{Ed}(t) = ^ET_M(t)^{Ed}T_M^{-1}(t), \quad (7.2)$$

where $^ET_M = ^ET_M$ determined by the given visual servoing objective.

Differentiating $(7.2)$ with respect to time yields

$$^ET_{Ed}(t) = ^ET_M(t)\dot{^MT}_M + ^ET_M(t)\dot{^MT}_{Ed}(t). \quad (7.3)$$
Differentiating Eq. (7.3),
\[ E\ddot{T}_{Ed}(t) = E\dddot{T}_{M}(t)\dot{M}T_{Ed}(t) + 2E\ddot{T}_{M}(t)\dot{M}T_{Ed}(t) + E\dot{T}_{M}(t)\dddot{T}_{Ed}(t), \]
(7.4)
where \( \dot{M}T_{Ed}, \ddot{M}T_{Ed}, \dddot{M}T_{Ed} \) are given as the desired visual servoing objective. \( E\ddot{T}_{M}, E\dddot{T}_{M}, E\dddot{T}_{Ed} \) can be observed by cameras. From these preparations, we can calculate the variables in the controllers of the system in the next subsection, such as \( \Delta \dot{p}_E \) and so on, leaving detail explanation for next subsection. As shown in Fig. 7.1, there are two errors that we should decrease to the value as small as possible in the visual servoing process. First one is the error between the actual object and the detected one, \( \dot{M}T_{Ed} \), and the other is the error between the desired end-effector and the actual one, \( E\ddot{T}_{Ed} \). In our research, the error of \( \dot{M}T_{Ed} \) is decreased by on-line recognition method of 1-step GA, MFF compensation method and the eye-vergence camera system, and the error of \( E\ddot{T}_{Ed} \) can be decreased by the hand visual servoing controller.

### 7.1.2 Servoing Controller

The aforementioned real-time recognition system is depicted at the lower side of the block diagram of the visual servoing system in Fig. 7.2. Based on the above analysis of the desired-trajectory generation, the desired hand velocity \( W\dot{r}_{Ed} \) is calculated as,
\[ W\dot{r}_{Ed} = K_p W\dot{r}_{E,Ed} + K_v W\dot{r}_{E,Ed}, \]
(7.5)
where $W_{r,E,Ed}, W_{\dot{r},E,Ed}$ are given by transforming $E_T Ed$ and $E_{\dot{T}} Ed$ from $\Sigma_E$ to $\Sigma_W$. $K_{P_\varphi}$ and $K_{V_\varphi}$ are positive definite matrix to determine PD gain.

The desired hand velocity $W_\omega d$ is calculated as,

$$W_\omega d = K_{P_\omega} W_{R E} E \Delta \epsilon + K_{V_\omega} W_{E,Ed}, \quad (7.6)$$

where $E \Delta \epsilon$ is the quaternion error that from the recognition result directly, and $W_{E,Ed}$ can be calculated by transforming $E_T Ed$ and $E_{\dot{T}} Ed$ from $\Sigma_E$ to $\Sigma_W$. Also, $K_{P_\omega}$ and $K_{V_\omega}$ are suitable feedback matrix gains.

The desired joint variable $\dot{q}_d$ is obtained by

$$\dot{q}_d = J^+(q) \begin{bmatrix} W_\dot{P}_d \\ W_\omega_d \end{bmatrix}, \quad (7.7)$$

where $J^+(q)$ is the pseudo-inverse matrix of $J(q)$, and $J^+(q) = J^T(JJ^T)^{-1}$. The control system, based on a PI control is expressed as

$$\tau = K_{SD}(\dot{q}_d - \dot{q}) + K_{SP}(q_d - q), \quad (7.8)$$

where $\dot{q}_d - \dot{q}$ is the velocity error of the joint angle, $K_{SD}$ and $K_{SP}$ are symmetric positive definite matrix to determine PD gain.
Robot PA10
Stereo Camera
3-D Marker

(a) (b)

Fig. 7.3: Step response experiment. (a) Initial pose of Pa10. (b) Visual servoing to a static 3-D marker.

7.2 Experimental Condition

A photograph of our experimental system is shown in Fig. 7.9. The robot used in this experimental system is a 7-Link manipulator, Mitsubishi Heavy Industries PA-10 robot. Two cameras are mounted on the robot manipulator’s end-effector. The frame frequency of stereo cameras is set as 33fps. The image processing board, CT-3001, receiving the image from the CCD camera is connected to the DELL WORKSTATION PWS650 (CPU: Xeon, 2.00 GHz) host computer.

Fig. 7.9 shows the coordinate system corresponding to Fig. 7.10. The initial pose of the end-effector is defined as $\Sigma_{E_0}$, and given by

$$
W_{E_0} = \begin{bmatrix}
0 & 0 & 1 & -918 \\
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 455 \\
0 & 0 & 0 & 1
\end{bmatrix},
$$

(7.9)

position unit: [mm].

7.3 Experimental Result

7.3.1 Step Response Experiment

Here, a static object is set as $^E\psi_M = [-70\text{[mm]}, 70\text{[mm]}, 1000\text{[mm]}, 0.1, -0.2, 0.12]^T$, where the value of orientation 0.1 in quaternion expression is about 12[deg]. The objective of visual
Fig. 7.4: Hand pose error of step response without using MFF

Fig. 7.5: Hand pose error of step response by using MFF

 servoing is given by a fixed relation between $\Sigma_E$ and $\Sigma_M$, as

$$E\psi_{Md} = [0[mm], 10[mm], 900[mm], 0, 0, 0]^T.$$ (7.10)

The initial pose of the robot manipulator is shown in Fig.7.3(a), and the moved robot manipulator to satisfy $E\psi_{Md}$ is shown in Fig.7.3(b).

To show the effectiveness of the proposed MFF method, we perform the step response experiment with MFF method and without MFF method separately. Fig 7.4 shows the difference of the desired hand pose and the actual hand pose in $\Sigma_{E0}$ without using MFF method. Fig 7.5 shows the hand difference with using MFF method. In Fig 7.4, the end-effector is unstable from 6[s] to 28[s]. Since the hand began to move, the object in camera frame was moving together with the end-effector, then the recognition dynamics became worse, which cause the vibration in this period. The end-effector cost 30[s] to be converged to the desired pose in the case of not using MFF compensation.
Table 7.1: Review of Literature

<table>
<thead>
<tr>
<th>Reference</th>
<th>Convergence time of step response</th>
</tr>
</thead>
<tbody>
<tr>
<td>28)</td>
<td>about 9.9[s] when the desired position is parallel to the image plane, else, about 49.5[s].</td>
</tr>
<tr>
<td>29)</td>
<td>in x, y, roll, pitch, yaw 30s, in z position about 70s</td>
</tr>
<tr>
<td>37)</td>
<td>about 60s.</td>
</tr>
<tr>
<td>38)</td>
<td>about 150s.</td>
</tr>
<tr>
<td>39)</td>
<td>is about 200s.</td>
</tr>
</tbody>
</table>

On the other hand, as shown in Fig 7.5, such vibrations existing in Fig 7.4 had been suppressed, and the end-effector cost about 10[s] to converge to the desired pose by using MFF method.

Step response is usually used to evaluate the ability of a visual servoing system. Here, we list some researches on visual servoing in Table 7.1. By comparing the convergence speed with these researches, our system shows a good ability in visual servoing task.

7.3.2 Time-varying Path Control Experiment

The visual servoing described in this section is that the object remains stationary and the robot is commanded to move through a reference path with respect to it. Such a visual servoing has been performed by William J. Wilson etc. in 26, and they named it as relative path control experiment.
Fig. 7.7: Hand pose error of time-varying visual servoing without MFF method when $t = 60s$

Here, a static object is set as $E_0 = [0[mm], 70[mm], 1300[mm], 0, 0, 0]^T$. The desired end-effector’s time-varying trajectory is given by

\[
\begin{align*}
Ed_{xM}(t) &= 100 \times \sin(\frac{\pi}{T} t)[mm] \\
Ed_{yM}(t) &= 70[mm] \\
Ed_{zM}(t) &= 1300[mm] \\
Ed_{\epsilon_1}(t) &= 0 \\
Ed_{\epsilon_2}(t) &= 0 \\
Ed_{\epsilon_3}(t) &= 0
\end{align*}
\]

The desired motion of the end-effector with respect to a static object is shown in Fig. 7.6.

Here, we set the motion period of the manipulator $T$ as $60[s]$.

Fig 7.7 shows the errors of the desired hand pose and the actual hand pose in $\Sigma_{E_0}$ without using MFF method. Fig 7.8 shows the hand errors with using MFF method. The stereo cameras were shaking because of the dynamics of the robot manipulator. Thus, the fictional motion of the target object coming from the moving camera was difficult to recognize. As
Fig. 7.8: Hand pose error of time-varying visual servoing with MFF method when $t = 60s$ shown in Fig.1.2, the incorrect recognition affects the hand motion directly, and will cause the feedback system unstable. The increased errors shown in Fig 7.7 indicated the system became unstable as time passing. Compared with Fig.7.7, the errors in Fig 7.8 were limited in a range, which means the system was under a stable control.

This time-varying path control experiment has confirmed the effectiveness of the proposed MFF method. By using MFF method, the affect on recognition from the motion of the camera itself is compensated and the recognition dynamics is improved, therefore, the stability of the visual servoing system is increased.

### 7.3.3 Visual Servoing to A Moving Object

In this experiment, the target object is fixed on a mobile robot, as shown in Fig. 7.9. The coordinate system of the mobile robot is represented as $\Sigma_R$. Here, the motion of the mobile
robot is rotation around the $z$ axis of $\Sigma_R$ by
\[
\theta_d[deg] = a\sin\left(\frac{2\pi}{T}\right)t,
\]  
where we set $a = 8[deg], T = 40[s]$. The voltage of the right and left wheel is given by
\[
V_R = kp(\theta_d - \theta) + kv(\dot{\theta}_d - \dot{\theta}),
\]
\[
V_L = -V_R,
\]  
where $kp$ and $kv$ are suitable feedback PD control gains.

The effectiveness of the proposed visual servoing are evaluated by comparing the actual hand pose with the desired hand pose through visual servoing to the moving target object.
We also do the same experiment in the case of without using MFF method and with MFF method separately. Here, the objective of visual servoing is a fixed relative pose between $\Sigma_M$ and $\Sigma_E$, defined as $E_d\psi_M = [0, 10, 700, 0, 0, 0]^T$.

Figs. 7.11(a) to (f) is the experimental results in the case of not using MFF method, which show the actual motion of the end-effector with respect to the fixed frame of $\Sigma_{E_0}$, defined as $E_0\psi_E$, compared with the desired hand pose $E_0\psi_{Ed}$. Figs. 7.12(a) to (f) show the experimental results of $E_0\psi_E$ and $E_0\psi_{Ed}$ in the case of using MFF method. In the period of the trajectory of $E_0\psi_{Ed}$ is a straight line, the mobile robot did not move, visual servoing to a static object was performed firstly. Then the desired trajectory in Fig. 7.11 and Fig.

![Fig. 7.11: Hand pose error of visual servoing without MFF method](image)

![Fig. 7.12: Hand pose error of visual servoing with MFF method](image)
7.12(a),(e) began to turn to curved line of sin/cos function, the mobile robot started to move. Comparing Figs. 7.11(a),(e) with Fig. 7.12(a),(e), the time-delay of hand motion in the case of using MFF method was smaller than that without using MFF method. The errors of hand motion in the other (b),(c),(d),(f) figures were also smaller in the case of using MFF method. This experiment of visual servoing to a moving object has also confirmed the effectiveness of the proposed MFF method and the stability improvement of the visual servoing system.
8 Hand & Eye-Vergence Visual Servoing System

8.1 Advantage of Eye-Vergence dual Visual Servoing System

Fig. 8.1 (a) shows that when people keep tracking a moving object, they may keep up to the object in case of the object moving slowly, but when the object become to move faster and faster, human’s face cannot be kept positioned squarely to the object, while human’s eye can still keep staring at the object because of its small mass and inertial moment. Another example is that when people shoot an arrow to some object, while people is running or riding on a horse as shown in Fig. 8.1 (b), the eyes can always keep staring at the object even when the body and head cannot. Inspired on these actions of human, we have proposed the eye-vergence system.

Eye-vergence function bears two fundamental fruits as kinematic merit and dynamical one. Firstly kinematic merit is described. Observing ability of a fixed-hand-eye configuration may be deteriorated by relative geometry of the camera and the target, as the robot cannot observe the object well when it is too near the cameras (Fig. 8.2 (a)), small intersection space of the possible sight space of the two cameras (Fig. 8.2 (b)), and the image of the

![Fig. 8.1: Eye-vergence in Humanoid](image-url)
object cannot appear in the center of both cameras as Fig. 8.2 (c), so we could not get clear image information of target because of camera lens aberrations at peripheral part of the lens, resulting in the pose measurement accuracy with fixed camera being deteriorated.

Eye-vergence system can solve the above problems since it can adapt its orientation to see the target object at the center of cameras, as it is shown in Fig. 8.3 (a)-(c), enhancing the measurement accuracy in trigonometric calculation and averting peripheral distortion of camera lens by observing target at the center of lens. Recent researches on visual servoing are limited generally in a swath of tracking an object while keeping a certain constant distance \cite{9, 28, 29}. But the final objective of visual servoing lies in approaching the end-effector to a target and then work on it, like grasping. In this case, the desired relation between the cameras and the object is time varying, so such eye-vergence camera system in Fig. 8.3 is indispensable to keep suitable viewpoint all the time during the approaching visual servoing.

The other merit of eye-vergence is concerning dynamical effects to keep tracking a moving target in the camera’s view. Needless to say in visual servoing application, keeping closed loop
of visual feedback is vital from a view point of closed loop control stability. As shown in Fig. 8.4, cameras fixed at the hand of manipulator are keeping staring at the object at first in (a), but when the target moves so fast that the manipulator cannot catch up with the speed of the target because of whole manipulator’s dynamics, resulting in the visual feedback cut as shown in (b). To improve this pose tracking difficulty of fixed hand eye system, the eye-vergence function seems dynamically effective, because of small mass and inertia moment of the eye ball comparing those of full manipulator’s structure. Therefore tracking ability of eye-vergence can be better than fixed, as in Fig. 8.4 (c), like animals tracks target with eye motion before rotate their heads to the target. Some researcher may say that dynamical couplings between the eyes’ motions and manipulator’s body stand against utilizing the dynamical effect of eye-vergence, however the small mass of eye makes the coupling less than the other coupling among links constituting the manipulator’s body. This helps above dynamical effect usable.

To verify above dynamical superiority of eye-vergence system by real eye-vergence robot system, in this report frequency response experiments where target object moves with sinusoidal time profile have been conducted. In our research before 27), we have evaluated the effects of eye-vergence visual servoing system as shown in Fig. 1.3 through approaching motion to a target, in which the desired time-varying pose relation between hand and
target was given. But on the viewpoint of above mentioned dynamical merit of hand &
eye-vergence, it has yet to be affirmed experimentally. Then we discuss the performance of
the eye-vergence system on the viewpoints of how the system improve dynamical stability
and trackability against sinusoidal motion of the target by frequency analyses, clarifying that
the eye-vergence system has superior stability and trackability performances in pose tracking
dynamical motions.

8.2 Maximum Observability

Here, we show an investigation into the influence of viewpoint of observing on 3-D posi-
tion/orientation measurement. The distance between the origin of one camera to the origin
of $\Sigma_E$ is defined as $L_l / L_R$, and the angle rotating around y axis of $\Sigma_{CL} / \Sigma_{CR}$ is defined as
$\theta_l / \theta_R$.

We compare two stereo camera systems: parallel stereo camera system (Fig.8.5(a)), where
$L_l = L_R = 75[mm]$, and cross stereo camera system (Fig.8.6(a)), where $L_l = L_R = 250[mm]$,
$\theta_l = \theta_R = 20[deg]$. A fitness value distribution of $F(\psi)$ by scanning the x and z position of of
$\Sigma_E$ using the moving model with fixed true values of $(y, \epsilon_1, \epsilon_2, \epsilon_3)$, is shown in Fig.8.5(b) and
Fig.8.6(b) corresponding to each stereo camera system, Fig.8.5(b) and Fig.8.6(b). It can be
seen that when the position of the model near to the true position $(x, z) = (0, 700)[mm]$, the
fitness function has maximum values in both Fig.8.5(b) and Fig.8.6(b). However, it shows
that the peak in Fig.8.6(b) is sharper than the peak in Fig.8.5(b). As we know, GA find
the maximum value fast in sharp mountain. Thus, the real-time recognition is easier to
be performed in the case of cross stereo camera system than that of parallel stereo camera
system.

We have also done some investigations by comparing cross stereo camera systems with
different $\theta_l / \theta_R$ with fixed $L_l$ and $L_R$, from which we found that observing the object through
both centers of left and right cameras gave the sharpest mountain. Based on the this, we
consider that the stereo cameras had better to keep changing their placement toward a moving
target object in order to prepare a good viewpoint for a better measurement of a target object,
which is the concept of eye-visual servoing.

Since the maximum observability can be achieved in eye-visual servoing, the object recog-
8.3 Eye-vergence Simulation

8.3.1 Simulator and Robot Dynamics

As manipulators the dynamics equation of the eye-vergence visual servoing system is also

\[ M(q) \ddot{q} + h(q, \dot{q}) + g(q) + d(q, \dot{q}) = \tau \]

but here \( q \) is a 10 \( \times \) 1 vector express the angle of each joint, \( \tau \) is the input torque, we define

\[ q_E = \begin{bmatrix} q_1 \\ \vdots \\ q_{10} \end{bmatrix}, \quad q_c = \begin{bmatrix} q_8 \\ q_9 \\ q_{10} \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_{10} \end{bmatrix}, \]

and, \( M(q) \): the inertia matrix, \( h(q, \dot{q}) \): the vector representing the centrifugal and coriolis forces, \( g(q) \): the vector representing the gravity load, \( d(q, \dot{q}) \): the vector representing the frictional force. Here, we assume \( d(q, \dot{q}) = 0 \).
Two cameras are mounted on the end-effector, FCB-1X11A manufactured by Sony Industries. The frame frequency of stereo cameras is set as 33fps. The structure of the manipulator and the cameras are shown in Fig. 8.7. To evaluate the trackability of a visual servoing system, we should have the whole model of the system first, in the research before we always ignore the mass and moment inertia of the camera system. In this paper we also consider the mass and moment inertia of the two cameras.

The motion equations of PA-10 can be calculated by normal Newton-Euler formulation:

\[
\omega_i = R^T_i \omega_{i-1} + R_i r_i \dot{q}_i \tag{8.2}
\]

\[
\dot{\omega}_i = R^T_i (\dot{\omega}_{i-1} + R_i r_i \ddot{q}_i + \omega_{i-1} \times (R_i r_i \dot{q}_i)) \tag{8.3}
\]

\[
\ddot{p}_i = R^T_i (\ddot{p}_{i-1} + \dot{\omega}_{i-1} \times l_{i-1} + \omega_{i-1} \times (\omega_{i-1} \times l_{i-1})) \tag{8.4}
\]

\[
\ddot{p}_g = \ddot{p}_i + \dot{\omega}_i \times a_i + \omega_i \times (\omega_i \times a_i) \tag{8.5}
\]

where \(i\) represents the link number and \(1 \leq i \leq 7\) in (8.2) to (8.5).

Denote the tilt angle of the camera system as link 8, the motion equations of this camera link also follows (8.2) to (8.5), only let \(i=8\).

Denote the right pan angle of the camera system as link 9, and the left one as link 10,
the motion equations of these two links are

\[\omega_i = R_i^T (\omega_8 + R_i r_i \dot{q}_i) \quad (8.6)\]
\[\dot{\omega}_i = R_i^T (\dot{\omega}_8 + R_i r_i \ddot{q}_i + \omega_8 \times (R_i r_i \dot{q}_i)) \quad (8.7)\]
\[\ddot{p}_i = R_i^T (\ddot{p}_8 + \omega_8 \times l_8 + \omega_8 \times (\omega_8 \times l_8)) \quad (8.8)\]
\[\ddot{p}_{gi} = \ddot{p}_i + \dot{\omega}_i \times a_i + \omega_i \times (\dot{\omega}_i \times a_i), \quad (8.9)\]

where \(i = 9, 10\).

Then with the motion of each joint we can calculate the force and the moment of the links interacting. The force and moment of the pan angles of the cameras (link 9 and link 10) can be calculated by:

\[f_i = m_i \ddot{p}_{gi} \quad (8.10)\]
\[n_i = I_i \dot{\omega}_i + \omega_i \times I_i \omega_i + a_8 \times (m_8 \ddot{p}_{gi}) \quad (8.11)\]
\[\tau_i = q_i^T n_i \quad (8.12)\]

where \(i = 9, 10\). Add the forth and moment of both left and right pan links onto the tilt link of the camera system, we have the force and moment of tilt link (link 8) as:

\[f_8 = R_9 f_9 + R_{10} f_{10} + m_8 \ddot{p}_8 \quad (8.13)\]
\[n_8 = R_{10} n_{10} + R_9 n_9 + I_8 \dot{\omega}_8 + \omega_8 \times I_8 \omega_8 + a_8 \times (m_8 \ddot{p}_{gs}) + l_{8L} \times R_{10} f_{10} \]
\[+ l_{8R} \times R_9 f_9 \quad (8.14)\]
\[\tau_8 = q_8^T n_8 \quad (8.15)\]

Then the force and moment of the manipulator are as follows:

\[f_i = R_i f_i m_i + \ddot{p}_{g,i+1} \quad (8.16)\]
\[n_i = R_{i+1} n_{i+1} + I_i \dot{\omega}_i + \omega_i \times I_i \omega_i + a_i \times (m_i \ddot{p}_{gi}) + l_{i+1} \times R_{i+1} f_{i+1} \quad (8.17)\]
\[\tau_i = q_i^T n_i \quad (8.18)\]

8.3.2 Hand & Eye-Vergence Visual Servoing

Hand Desired Acceleration

The block diagram of our proposed hand & eye-vergence visual servoing controller is shown in Fig. 1.3. The hand-visual servoing is the outer loop. The controller used for hand-visual
servoing is proposed by B. Siciliano \(^{35}\). First we will introduce the variables defined in the system:

\[
\Delta p_E = p_d - p_E
\]  

(8.19)

here \(d\) and \(E\) in the bottom right corner of the \(p\) means the desired position and actual position of the end-effector. There is no letter in the top left corner, it means the vector or the matrix is expressed in the world frame.

A special type of angle/axis representation of the orientation error is obtained with the quaternion, i.e

\[
E_{\eta} = \cos \frac{\theta_{Ed}}{2}
\]

(8.20)

\[
\Delta E = \sin \frac{\theta_{Ed}}{2} E_{kEd}
\]

(8.21)

Here \(\theta\) and \(k\) are the rotation angle and the rotation axis of the object. The letter in the top left corner express the coordinate where the vector or the rotation matrix is expressed in.

While the angle velocity error between the desired and the actual angle velocity is defined as

\[
\Delta \omega_E = \omega_d - \omega_E
\]

(8.22)

With the variables defined above, we just show main equations of the hand visual servoing controller that are used to calculate desired acceleration and joint compensation

\[
a_{pE} = \ddot{p}_d + K_{Dp} \Delta \dot{p}_E + K_{Pp} \Delta p_E
\]

(8.23)

\[
a_{oE} = \dot{\omega}_d + K_{Do} \Delta \omega_E + K_{Po} R_{EE} \Delta \epsilon
\]

(8.24)

\[
s_E = J_E^T(q_E) \{ a_E - \dot{J}_E(q_E, \dot{q}_E) \dot{q}_E \} + \{ I - J_E^T(q_E) \dot{J}_E(q_E) \} \{ E_p(q_{Ed} - q_E) + E_d(\dot{q}_{Ed} - \dot{q}_E) \}
\]

(8.25)

Here, \(\dot{q}_E\) is a 7 \(\times\) 1 vector representing the angular accelerations of the first 7 links of the PA-10 manipulator. \(a\) can be written short for \([a_{pE}^T, a_{oE}^T]^T\). The quaternion error from the actual orientation to the desired orientation of the end effector \(E \Delta \epsilon\) can be extracted from the transformation \(E T_{Ed}\), and the other error variables in (8.23), (8.24) are described in \(\Sigma_W\), which can be calculated by the transformation \(E T_{Ed}, E \dot{T}_{Ed}, E \ddot{T}_{Ed}\) in (7.2), (7.4), using the rotational matrix \(W R_E(q)\) through coordinate transformation.
\( \mathbf{J}_E(q_E) \) is the Jacobian matrix from the world coordinate to the end effector, which means that \( \omega_E = \mathbf{J}_E(q_E) \dot{q}_E \), and \( \mathbf{J}_E^+(q_E) \) in (8.25) is the pseudo-inverse of \( \mathbf{J}_E(q_E) \) given by \( \mathbf{J}_E^+(q_E) = \mathbf{J}_E^T (\mathbf{J}_E \mathbf{J}_E^T)^{-1} \). \( K_{D_p}, K_{F_p}, K_{D_o}, K_{P_o} \) are positive control gains.

**Eye-vergence Desired Acceleration**

The eye-vergence visual servoing is the inner loop of the visual servoing system shown in Fig. 1.3. In this paper, we use two pan-tilt cameras for eye-vergence visual servoing. Here, the positions of cameras are supposed to be fixed on the end-effector. For camera system, \( q_8 \) is tilt angle, \( q_9 \) and \( q_{10} \) are pan angles, and \( q_8 \) is common for both cameras. As it is shown in Fig. 8.8, \( E_{x_M}, E_{y_M}, E_{z_M} \) express position of the detected object in the end-effector coordinate.

The desired angle of the camera joints \( q_{cd}^T = [q_{sd}, q_{9d}, q_{10d}]^T \) can be calculated by:

\[
q_{sd} = \text{atan2}(E_{y_M}, E_{z_M})
\]

\[
q_{9d} = \text{atan2}(l_{SR} + E_{x_M}, E_{z_M})
\]

\[
q_{10d} = \text{atan2}(l_{SL} - E_{x_M}, E_{z_M})
\]

where \( l_{SL} = l_{SR} = 150[mm] \) is the camera location. We set the center line of a camera as the \( z \) axis of each camera coordinate, so the object will be in the center of the sight of the right camera when \( R_{x_M} = 0 \) and \( R_{y_M} = 0 \). Here \( R_{x_M}, R_{y_M}, R_{z_M} \) express the position of the detected object in the right camera coordinate.

\[
\Delta q_C = q_{Cd} - q_C
\]
\[ s_C = \ddot{q}_{Cd} + K_{DC} \Delta \dot{q}_C + K_{PC} \Delta q_C \]  
(8.30)

here, \( K_{DC}, K_{PC} \) are positive definite diagonal matrix.

**Hand/Eye-vergence Controller**

By the desired accelerations from (8.25) and (8.30), input \( \tau \) is calculated by:

\[ s = \begin{bmatrix} s_E \\ s_C \end{bmatrix} \]  
(8.31)

\[ \tau = M(q)s + h(q, \dot{q})\dot{q} + g(q) \]  
(8.32)

**8.3.3 Stability of Hand & Eye-vergence Motion**

**Manipulator Dynamics**

First, we discuss about the convergence of our proposed hand visual servoing system. From the input torque of each joint in (8.32) and the dynamics equation of the system (8.1)

\[ \ddot{q} = s \]  
(8.33)

so

\[ \ddot{q}_E = s_E. \]  
(8.34)

Take (8.25) (here we do not consider the second item in the right side which is the controller of the redundance) into (8.34) we have

\[ a_E = J_E(q_E)\ddot{q}_E + \dot{J}_E(q_E, \dot{q}_E)\dot{q}_E \]  
(8.35)

and, from the definition of Jacobian matrix,

\[ \begin{bmatrix} \dot{p}_E \\ \omega_E \end{bmatrix} = J_E(q_E)\dot{q}_E \]  
(8.36)

so

\[ \begin{bmatrix} \dot{p}_E \\ \dot{\omega}_E \end{bmatrix} = J_E(q_E)\ddot{q}_E + \dot{J}_E(q_E, \dot{q}_E)\dot{q}_E \]  
(8.37)

which means:

\[ \begin{bmatrix} \dot{p}_E \\ \dot{\omega}_E \end{bmatrix} = \begin{bmatrix} a_{pE} \\ a_{\omega E} \end{bmatrix}. \]  
(8.38)
Submit \( \mathbf{a}_{pE}, \mathbf{a}_{oE} \) in (8.38) into (8.23) and (8.24),

\[
\Delta \dot{p}_E + \mathbf{K}_{Dp} \Delta \dot{p}_E + \mathbf{K}_{Pp} \Delta p_E = 0 \quad (8.39)
\]

\[
\Delta \dot{\omega}_E + \mathbf{K}_{D\omega} \Delta \dot{\omega}_E + \mathbf{K}_{P\omega} \mathbf{R}_E \Delta \mathbf{E} = 0 \quad (8.40)
\]

### 8.3.4 Differentiation of Jacobian matrix

Here we will introduce the method to differentiate Jacobian matrix. As we know the Jacobian matrix can be calculated as follow:

\[
\mathbf{J} = \begin{bmatrix}
0 z_1 \times 0 p_{E,1} & 0 z_2 \times 0 p_{E,2} & \ldots & 0 z_n \times 0 p_{E,n}
\end{bmatrix} \quad (8.41)
\]

Here \( z \times p \) is the outproduct of vector \( z \) and vector \( p \), it also can be calculated by \([z \times]p\), where

\[
[z \times] = \begin{bmatrix}
0 & -z_z & z_y \\
z_z & 0 & -z_x \\
-z_y & z_x & 0
\end{bmatrix} \quad (8.42)
\]

And

\[
0 z_i = 0 R_i^i e_z \quad (8.43)
\]

Here \( ^i e_z = (0, 0, 1)^T \)

Differentiate the double side of equation (8.43) by time,

\[
0 \dot{z}_i = 0 \dot{R}_i e_z + 0 R_i \dot{e}_z
\]

\[
= 0 \dot{R}_i e_z
\]

\[
= ([0 \omega_i] \times 0 R_i e_z)
\]

(8.44)

For \( z \times p \) can be expressed as \([z \times]p\),

\[
\frac{d[0 z_i \times 0 p_{E,i}]}{dt} = \frac{d[0 z_i \times]}{dt} 0 p_{E,i} + [0 z_i \times] \frac{d0 p_{E,i}}{dt}
\]

(8.45)

According to (8.42) \( d[0 z_i \times]/dt \) can be made up by the elements of \( \dot{z}_i \), which can be calculated by equation (8.44) and

\[
0 \dot{p}_{E,i} = \dot{p}_E - \dot{p}_i
\]

(8.46)
\( \dot{p}_i \) can be calculated by

\[
\dot{p}_i = J_{pi} \dot{q}
\]

(8.47)

here \( J_{pi} \) is the matrix which is made up by the first three lines of the Jacobian matrix of the \( i-th \) link.

Substitute (8.46) and \( d[0 z_i] \times /dt \) into (8.45) we can calculate the differential Jacobian matrix:

\[
\dot{J} = \begin{bmatrix}
[0^T \omega_1 \times]^0 R_1 e_z \times]^0 p_{E,1} + [0^T z_1 \times](J p \dot{q} - J_{p1} \dot{q}) \\
[0^T \omega_1 \times]^0 R_1 e_z \\
[0^T \omega_n \times]^0 R_n e_z \times]^0 p_{E,n} + [0^T z_n \times](J p \dot{q} - J_{pn} \dot{q})
\end{bmatrix}.
\]

(8.48)

### 8.3.5 Camera Dynamics

For the cameras, from (8.33),

\[
\ddot{q}_C = s_C.
\]

(8.49)

From (8.30), The close loop becomes:

\[
\Delta \ddot{q}_C + K_{D_C} \Delta \dot{q}_C + K_{P_C} \Delta q_C = 0
\]

(8.50)

#### Stability Analysis

We invoke a Lyapunov argument, the feed back gains are taken as scalar matrices, i.e. \( K_{D_p} = K_{D_p} I, K_{P_p} = K_{P_p} I, K_{D_o} = K_{D_o} I \) and \( K_{P_o} = K_{P_o} I \). Here we assume that the feedback gains of the links are the same.

\[
V = \Delta p_T \dot{E} K_{P_p} \Delta p + (\Delta \dot{p}_E)^T \Delta \dot{p}_E + K_{P_p} \{ (\eta - 1)^2 + (\Delta \epsilon)^T \Delta \epsilon \}
\]

\[
+ \frac{1}{2} (\Delta \omega_E)^T \Delta \omega_E + \Delta q_T \dot{C} K_{P_C} \Delta q_C + (\Delta \dot{q}_C)^T \Delta \dot{q}_C \geq 0
\]

(8.51)

so

\[
\dot{V} = 2 \Delta \dot{p}_E^T (\Delta \dot{p}_E + K_{P_p} \Delta p) + 2 K_{P_p} \{ (\eta - 1) \eta + (\Delta \epsilon)^T \Delta \epsilon \} + \Delta \omega_E^T \Delta \omega_E
\]

\[
+ 2 \Delta \dot{q}_C^T (\Delta \dot{q}_C + K_{P_C} \Delta q_C)
\]

(8.52)

from (8.39) we can know that

\[
\Delta \ddot{p}_E + K_{P_p} \Delta p_E = -K_{D_p} \Delta \dot{p}_E
\]

(8.53)
from the quaternion definition we can know that \( \dot{\eta} = -\frac{1}{2}(\Delta \epsilon)^T \Delta \omega_E \) (8.54)

and

\[
\Delta \dot{\epsilon} = \frac{1}{2} \mathbf{E}(\eta, \Delta \epsilon) \Delta \omega_E
\]

where \( \mathbf{E}(\eta, \epsilon) = \eta \mathbf{I} - \mathbf{S}(\epsilon) \), \( \mathbf{S}(a) \) is a antisymmetric matrix that satisfies \( \mathbf{S}(a)b = a \times b \).

From (8.50) we can get:

\[
\Delta \ddot{\eta} \mathbf{C} + \mathbf{K}_D C \Delta \dot{\eta} \mathbf{C} = -\mathbf{K}_D \Delta \dot{\eta} \mathbf{C}
\]

Substitute (8.40), (8.53), (8.54), (8.55) and (8.56) into (8.52) we can get:

\[
\dot{V} = 2\Delta \dot{p}_E^T (\Delta \dot{p}_E + \mathbf{K}_D \Delta p_E) + \mathbf{K}_P \{ - (\eta - 1)(\Delta \epsilon)^T \Delta \omega_E + (\Delta \epsilon)^T \mathbf{E}(\eta, \Delta \epsilon) \Delta \omega_E \}
+ \Delta \omega_E^T \Delta \dot{\omega}_E + 2\Delta \dot{q}_C^T (\Delta \dot{q}_C + \mathbf{K}_P \Delta q_C)
\]

\[
= -2\Delta \dot{p}_E^T K_{D_p} \Delta \dot{p}_E + \mathbf{K}_P \{ - (\eta - 1)(\Delta \epsilon)^T \Delta \omega_E + \eta \Delta \epsilon^T \Delta \omega_E \} - \Delta \omega_E^T K_{D_o} \Delta \omega_E
\]

\[
- \mathbf{K}_P \Delta \dot{\omega}_E \mathbf{R}_E \Delta \epsilon - 2\Delta \dot{q}_C^T K_{D_C} \Delta \dot{q}_C
\]

\[
\leq -2\Delta \dot{p}_E^T K_{D_p} \Delta \dot{p}_E - \Delta \omega_E^T K_{D_o} \Delta \omega_E - 2\Delta \dot{q}_C^T K_{D_C} \Delta \dot{q}_C \leq 0
\]

(8.57)

here, because \( K_{D_p}, K_{D_o} \leq K_{D_C} \) are positive definite, only if \( \Delta \dot{p}_E = 0, \Delta \omega_E = 0 \) and \( \Delta q_C = 0, \dot{V} = 0 \). \( \Delta \dot{p}_E = 0 \) and \( \Delta \dot{p}_E = 0 \), from (8.39), \( \Delta p_E = 0 \). For the same reason, when \( \Delta q_C = 0 \), from (8.50), \( \Delta q_C = 0 \). when \( \Delta \omega_E = 0, \Delta \dot{\omega}_E = 0 \) and from (8.40) \( \Delta \epsilon = 0 \).

The definition domain of \( \theta \) is \((-\pi, \pi)\), so the manipulator and the cameras asymptotically converge to the invariant sets \( s \).

\[
s = \{ \Delta p_E = 0, \Delta \dot{p}_E = 0, \eta = 1, \Delta \epsilon = 0, \Delta \omega_E = 0, \Delta q_C = 0, \Delta \dot{q}_C = 0 \}
\]

(8.58)

so,

\[
\lim_{t \to \infty} \Delta p_E = 0, \lim_{t \to \infty} \Delta \dot{p}_E = 0, \lim_{t \to \infty} \Delta \epsilon = 0
\]

(8.59)

then

\[
\lim_{t \to \infty} E T_{Ed} = I, \quad \lim_{t \to \infty} E \dot{T}_{Ed} = 0
\]

(8.60)

substitute (8.60) to (7.2),

\[
\lim_{t \to \infty} E T = \lim_{t \to \infty} E \dot{T}_{M} = 0
\]

(8.61)
In (8.61), $\Sigma_E$ converge to $\Sigma_{Ed}$. so, from (8.58)

$$\lim_{t \to \infty} q_C = q_{Cd}$$

and

$$\lim_{t \to \infty} R_z^M = 0, \quad \lim_{t \to \infty} R_y^M = 0, \quad \lim_{t \to \infty} R_y^M = 0$$

so the object will become on the center line of the cameras, which means that the object will always keep in the center of the sight of the cameras.

### 8.4 Simulation of Hand & Eye-vergence Visual Servoing on Known Object

To verify the effectiveness of the proposed hand & eye visual servoing system, we conduct the experiment of visual servoing to the same 3D marker in chapter 7.

#### 8.4.1 Simulation Condition

The recognition error does not affect the dynamic error, so we assume that $^M T_M = I$. The position and orientation of the target object are given to the robot directly in the simulation. The initial hand pose is defined as $\Sigma_{E_0}$, while the initial object pose is defined as $\Sigma_{M_0}$, and the homogeneous transformation matrix from $\Sigma_W$ to $\Sigma_{M_0}$ is:

---

Fig. 8.9: Object and the visual-servoing system

---

82
The target object move according to the following time function:

\[ \begin{bmatrix} 0 & 0 & -1 & -1410[mm] \\ 1 & 0 & 0 & 0[mm] \\ 0 & -1 & 0 & 355[mm] \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot (8.64) \]

The controller gain of the system \( K_{D_p}, K_{P_p}, K_{D_o}, K_{P_o} \) are affected by the mass, initial moment, the amplifier output and many other conditions. From the common sense because the camera mass is smaller than the manipulator, \( K_{D_c} \) and \( K_{P_c} \) can be set bigger than \( K_{D_o} \) etc. Here \( K_{P_c} = diag\{5, 5, 5\}, K_{D_c} = diag\{3, 3, 3\}, K_{P_p} = K_{P_o} = K_{D_p} = K_{D_o} = diag\{1, 1, 1\}. \)

### 8.4.2 Definition of Trackability

Camera trackability: Here, to compare the trackability of the eye-vergence system and fixed camera system, we define a concept of gazing point. As it is shown in Fig. 8.10 the intersection
Fig. 8.11: Comparison of Cameras’ and End-effector’s Trackabilities by Frequency Response of the gazing line of right camera and the $y_{M_0}z_{M_0}$ plane is defined as the gazing point. The relative relation between $\Sigma_{M_0}$ and $\Sigma_R$ is given by Homogeneous Transformation as $M_0T_R$, $M_0T_R$ conclude the rotation matrix $M_0R_R$ and the position vector $M_0p_R$, and the rotation matrix $M_0R_R$ can be written as $[M_0x_R, M_0y_R, M_0z_R]$. The direction of $M_0l_R$ in Fig. 8.10 is same to the direction of $x_R$, and $M_0l_R$ can be expressed as:

$$M_0l_R = M_0p_R + k_R M_0x_R \quad (8.67)$$

here $k_R$ is a scalar variable. The gazing point of the right camera expressed in $\Sigma_{M_0}$ is $M_0p_{GR} = [0, M_0y_{GR}, M_0z_{GR}]^T$. For $M_0l_R = M_0p_{GR}$ in $x$ direction, $(M_0p_R)_x + k_R(M_0x_R)_x = 0$. And usually $(M_0x_R)_x \neq 0$, $k_R$ can be calculated by $k_R = -(M_0p_R)_x/(M_0x_R)_x$, and the $y$, $z$ coordinate of the gazing point in $\Sigma_{M_0}$ can be calculated by:

$$M_0y_{GR} = (M_0p_R)_y + k_R(M_0x_R)_y \quad (8.68)$$

$$M_0z_{GR} = (M_0p_R)_z + k_R(M_0x_R)_z \quad (8.69)$$

The target object’s motion is given by (8.65), because the motion of the target object $M$ is parallel to the $y_{M_0}$, we take $M_0y_M(t)$ as the input, and the gazing point of the right camera $M_0y_{GR}(t)$ as the response. And define the concept of trackability by the frequency response of $M_0y_{GR}(t)$, the trackability of the left camera can be defined in the same way. End-effector trackability: To compare with the trackability of the camera, it is necessary to define the End-effector trackability. Here the gazing line direction of normal static hand eye system is
same to the $x$ direction of $\Sigma_E$, so the gazing point of the static hand-eye system is same to the gazing point of the end-effector. As shown in Fig. 8.10, the gazing line equation of the end-effector is:

$$M_0 l_E = M_0 p_E + k_E M_0 x_E$$  \hspace{1cm} (8.70)

here $k_R$ is a scalar variable. The gazing point of the end-effector is in $y M_0$-$z M_0$ plane. $k_E = -(M_0 p_E)_x/(M_0 x_E)_x$ the $y$, $z$ coordinate of the end-effector is:

$$M_0 y_{GE} = (M_0 p_E)_y + k_E (M_0 x_E)_y$$  \hspace{1cm} (8.71)

$$M_0 z_{GE} = (M_0 p_E)_z + k_E (M_0 x_E)_z$$  \hspace{1cm} (8.72)

The end-effector gazing point coordinate is $[0, M_0 y_{GE}, M_0 z_{GE}] \in \Sigma_{M_0}$.

### 8.4.3 Simulation Results

The original position of the target object $W T_{M_0}$ is given by (8.64), the target object motion function is (8.65), the desired relation between the end-effector and the target object is given in (8.66). The $\omega$ in (8.65) changes from 0.01 to 2.00. In Fig. 8.11, we show the result of our experiment. The amplitude-frequency curve and the delay frequency curve are shown in Fig.8.11 (a) and Fig.8.11 (b). Here, for the fixed camera $A = M_0 y_M(t)$, $B = M_0 y_{GE}(t)$. For the right camera of Eye-Vergence system $A = M_0 y_M(t)$, $B = M_0 y_{GR}(t)$, for the left camera $A = M_0 y_M(t)$, $B = M_0 y_{GL}(t)$. In this two figures the abscissa axes are logarithmic scalar of $\omega$. In (a), (b), we sign the angular velocity when $\omega = 0.1256, 0.5024, 1.256$, and show the position of the gazing point of the cameras in eye-vergence simulation and the position of the gazing point of the end-effector in fixed camera experiment in (c), (d), (e). We can see that both the fixed-camera system and eye-vergence system can track the target object when $\omega = 0.1256$ while the fixed-camera system cannot track the target object when $\omega$ is faster than 0.5024 so in (e), and the eye-vergence system can track the target object even when $\omega = 1.256$.

From Fig. 8.11 (a) we can see the data of the cameras and the end-effector all become bigger as $\omega$ increases for the reason of resonance, but the curve of the fixed camera system is always below the curves of the cameras, we can see that the amplitude of the eye-vergence system is more closed to the target object than the fixed camera system, the fixed camera
system cannot track the target object when $\omega$ is bigger than 0.5024, so the point line disappear near $\omega = 0.5024$, while in eye-vergence system the fastest velocity of the target object under which the system can catch up with is 1.6956. From (b) the the curve of the fixed camera system is also below the curves of the cameras, which means that delay of the fixed camera system is bigger than the eye-vergence system. To be understood easily, we show the position of the gazing point of the cameras in eye-vergence experiment and the position of the gazing point of the end-effector in fixed camera experiment in (c), (d), (e). From the figures it is also easily to see that comparing with the fixed camera system, the eye-vergence system can track the target object better.

8.5 Simulation of Hand & Eye-Vergence Dual Visual Servoing on Unknown Object

To verify the effectiveness of the proposed hand & eye visual servoing system, we conduct the simulation of visual servoing to the same 3D marker in section 7.
Fig. 8.13: Results of hand & eye visual servoing by using MFF method in Σ_{E_0}.

8.5.1 Simulation Condition

Visual servoing is usually performed to keep a fixed relation with respect to a static or moving object. The visual servoing described in this paper is that the object remains stationary and the robot is commanded to move through a reference path with respect to it. Such a visual servoing has been performed by William J. Wilson etc. in \cite{26}, and they named it as relative path control visual servoing.

The coordinates of the system is set as Fig.8.12. The initial hand pose is defined as Σ_{E_0},
and the homogeneous transformation matrix from $\Sigma_{E_0}$ to $\Sigma_W$ is
\[
WTE_0 = \begin{bmatrix}
0 & 0 & 1 & 918[mm] \\
-1 & 0 & 0 & 0[mm] \\
0 & -1 & 0 & 455[mm] \\
0 & 0 & 0 & 1
\end{bmatrix}.
\] (8.73)

The desired hand time-varying trajectory expressed in $\Sigma_{E_0}$ is
\[
\begin{align*}
E_0x_{ed}(t) &= r_0 \ast \sin \frac{2\pi}{T} t \\
E_0y_{ed}(t) &= 0 \\
E_0z_{ed}(t) &= r_0 \ast \cos \frac{2\pi}{T} t \\
E_0\epsilon_1_{ed}(t) &= 0 \\
E_0\epsilon_2_{ed}(t) &= 0 \\
E_0\epsilon_3_{ed}(t) &= 0
\end{align*}
\] (8.74)

where the radius $r_0 = 100[mm]$, period $T = 60[s]$.  

### 8.5.2 Simulation Result

Firstly, we compare the hand & eye visual servoing with the proposed MFF method and without MFF method separately.

Fig. 8.13 shows the results of hand & eye visual servoing by using MFF method, all these results are represented in $\Sigma_{E_0}$. Fig. 8.13(a) is the actual end-effector in x and y position compared with the desired x and y. Fig. 8.13(b) is the end-effector’s motion in y and z plane of $\Sigma_{E_0}$. Fig. 8.13(c) is the end-effector’s motion in x and z plane of $\Sigma_{E_0}$. Fig. 8.13(d) is the end-effector’s motion in the orientation $\epsilon_1$ and $\epsilon_2$. Fig. 8.13(e) is the end-effector’s motion in the orientation $\epsilon_2$ and $\epsilon_3$. Fig. 8.13(f) is the end-effector’s motion in the orientation $\epsilon_1$ and $\epsilon_3$. Fig. 8.13(g) shows changing of the left camera’s angle and the right camera’s angle. Fig. 8.13(h) shows the changing of fitness value of the target’s recognition during visual servoing.

Fig. 8.13(a) to (f) show us the stable control of the robot manipulator. The errors between the actual and desired position in Fig. 8.13(a) and (b) is small, less than 10[mm]. As Fig. 8.13(c) shown, the desired motion of end-effector in x and z plane is a circle, and the actual position is close to it. The distribution of the end-effector’s actual orientation $\epsilon_1, \epsilon_2, \epsilon_3$ is converged to the desired value 0, error is less than 0.02 (about 3[deg]). Fig. 8.13(g) shows the placement of the left and right cameras keep changing to recognize the object easily. It confirmed the adaptability of the mobile stereo cameras. Fig. 8.13(h) shows the fitness value keep high, which means precise recognition of the object during visual servoing. It verified that once “1-step GA” finds the closeness model of the target object, the model will keep
Fig. 8.14: Results of hand & eye visual servoing without using MFF method in $\Sigma_{E_0}$.

overlapping the target object, never lose it, because of the MFF method that can compensate the target’s fictional motion coming from the robot itself.

On the other hand, as shown in Fig. 8.14, the visual servoing could not be performed (even in the first 3 seconds) in the case of without using MFF method. When the robot starts to moving, the pose of target object in $\Sigma_E$ is changed due to the dynamics of the robot manipulator. Without MFF method’s compensation, the “1-Step GA” can not recognize precisely, wrong recognition result will lead to wrong control of the robot, which makes the recognition more difficult. As shown in Fig. 8.14(h), the fitness value of the recognition is

89
decreasing to about 0.2, in such case, the target object is considered as to be lost, and the robot can not be normally controlled. The reason of target object lost is considered as GA’s convergence speed was not faster than the target speed relative to the camera.

Secondly, we compare the visual servoing by using the hand & eye visual servoing system and fixed parallel stereo cameras separately. Fig. 8.15 shows the results of time-varying visual servoing with using MFF method, in the case of using parallel stereo cameras. Compare with Fig. 8.13, we can find that the errors between the actual and desired position is bigger, especially in Fig. 8.15(c), the radius of the circle of the actual position in x and z plane is about
20[mm] smaller than the desired one. The distributions of the end-effector’s actual orientation $\epsilon_1, \epsilon_2, \epsilon_3$ in Fig. 8.13(d) to (f) are around the desired value 0, error are bigger than using mobile stereo cameras, about 0.04 (6[deg]). Fitness value of the target’s recognition during visual servoing is shown in Fig. 8.13(g), from 10[s] to 50[s] the fitness value is decreasing to 0.6, where a part of the object got out of the camera view because the stereo cameras are fixed to be parallel, they do not have the adaptability for recognition.

### 8.6 Hand & Eye-Vergence Experiment

#### 8.6.1 Hand & Eye-Vergence Visual Experiment Circumstance and Servoing Controller

The Eye-Vergence system also utilizes in experiment. The Mitsubishi PA-10 robot arm is a 7-DoF robot arm manufactured by Mitsubishi Heavy Industries. Two rotatable cameras with two pan angles and one sharing tilt angle mounted on the end-effector are FCB-1X11A manufactured by Sony Industries (Fig. 8.16). The frame frequency of stereo cameras is set as 33fps. The image processing board, CT-3001, receiving the image from the CCD camera is connected to the DELL WORKSTATION PWS650 (CPU: Xeon, 2.00 GHz) host computer. Here in the outer loop in Fig.1.3 which control the manipulator motion in 7.1.2 is also controls the manipulator in Hand & Eye-Vergence Experiment. The camera controller are shown as follow:
\[ \dot{q}_8 = K_{P_T}(q_{8d} - q_8) + K_{D_T}(\dot{q}_8 - \dot{q}_8), \]  \( (8.75) \)

\[ \dot{q}_9 = K_{P_C}(q_{9d} - q_9) + K_{D_C}(\dot{q}_9 - \dot{q}_9), \]  \( (8.76) \)

\[ \dot{q}_{10} = K_{P_C}(q_{10d} - q_{10}) + K_{D_C}(\dot{q}_{10d} - \dot{q}_{10}). \]  \( (8.77) \)

where \( K_{P_T}, K_{D_T}, K_{P_C}, K_{D_C} \) are positive control gain. And \( q_{8d}, q_{9d}, q_{10d} \) are calculated by 8.26 - 8.28 in 8.3.2 To verify the effectiveness of the proposed hand & eye visual servoing system, we conduct the experiment of visual servoing to the 3D marker in chapter 7

8.6.2 Experiment Condition

The initial hand pose is defined as \( \Sigma_{E_0} \), while the initial object pose is defined as \( \Sigma_{M_0} \), and the homogeneous transformation matrix from \( \Sigma_W \) to \( \Sigma_{M_0} \) is:

\[ W T_{M_0} = \begin{bmatrix} 0 & 0 & -1 & -1410[mm] \\ 0 & 1 & 0 & 0[mm] \\ 0 & -1 & 0 & 355[mm] \\ 0 & 0 & 0 & 1 \end{bmatrix}. \]  \( (8.78) \)

The target object move according to the following time function

\[ M_0 \vec{y}_M(t) = [0, M_0 y_M(t), 0, 0, 0]^T \]

\[ M_0 y_M(t) = -200 \sin(\omega t)[mm] \]  \( (8.79) \)
Fig. 8.18: Comparison of Cameras’ and End-effector’s Trackabilities by Frequency Response
here, $\omega$ is the angular velocity of the motion of the object.

The relation between the object and the desired end-effector is set as:

$$E_d\psi_M = [800[mm], 0, 0, 0, 0]$$  \hspace{1cm} (8.80)

Here, to compare the trackability of the eye-vergence system and fixed camera system, we also use the concept of gazing point introduced in 8.4.2

8.6.3 Experiment Results

In Fig. 8.18, we show the result of our experiment, we change the $\omega$ in (8.79) from 0.01 to 1.256 and get the data of the gazing point of the cameras of eye-vergence system and the gazing point of the end-effector of the fixed camera system separately, we do the experiment 10 times at every $\omega$ we selected, and use the average delay time and the amplitude to draw the frequency response curve. The amplitude-frequency curve and the delay frequency curve are shown in Fig.8.18 (a) and Fig.8.18 (b). Here, for the fixed camera $A = M_0 y_M(t)$, $B = M_0 y_{GE}(t)$. For the right camera of Eye-Vergence system $A = M_0 y_M(t)$, $B = M_0 y_{GR}(t)$, for the left camera $A = M_0 y_M(t)$, $B = M_0 y_{GL}(t)$. In this two figures the abscissa axes are $\omega$. In (a), (b), we sign the angular velocity when $\omega = 0.314, 0.628, 1.256$, and show the position of the gazing point of the cameras in eye-vergence experiment and the position of the gazing point of the end-effector in fixed camera experiment in (c), (d), (e). From (a), (b) we can see that the
fixed-camera system cannot track the target object when \( \omega \) is faster than 0.628 so in (e), there is only the data of the cameras and the target object. From Fig. 8.18 (a) we can see the data of the cameras and the end-effector all become smaller as \( \omega \) increases but the curve of the fixed camera system is always below the curves of the cameras, which means that delay of the fixed camera system is bigger than the eye-vergence system, from (b) the curve of the fixed camera system is also below the curves of the cameras, we can see that the amplitude of the eye-vergence system is more closed to the target object than the fixed camera system, so from (a) and (b) we can get the conclusion that the eye-vergence system has the better trackability than the fixed-camera system. To be understood easily, we show the position of the gazing point of the cameras in eye-vergence experiment and the position of the gazing point of the end-effector in fixed camera experiment in (c), (d), (e). and \( M_0 \dot{y}_M(0) = -200[mm/s] \), while the target object moved from static, so it cannot move stably at first, we use the data when the target object’s motion became stable. From the figures it is also easily to see that comparing with the fixed camera system, the eye-vergence system can track the target object better.
Part II

Lyapunov-stable Constraint-combined Force/Position Control Exploiting Constraint Redundancy and Dual Nature
9 Introduction

9.1 Background of Robot Constraint Control

It is well known that robots, particularly articulated types, are very dexterous and have large operable space. Hence, it will have a promising future to introduce such kinds of robots more extensively into manufacturing. For example, the tasks of the grinding or cutting of the deeply located surfaces within a cabinet might be too difficult to machine. Furthermore, for some auxiliary machining operations, it may cost too much for an expensive machining center to do. Therefore, employing robots in such areas will be a satisfactory alternative.

On the other hand, comparing with a machine tool, the characteristics of robots on stiffness, damping and vibration-proofing are somewhat poor. In order to take the advantage of the dexterity of robots, much sophisticated design and control strategies have to be developed.

We think that a paper \(^\text{40}\) had classified contacting tasks of robots practically. The following classification is along with the statements in \(^\text{40}\). Robot force control method can be largely classified into impedance control and hybrid control. In impedance control, a prescribed dynamic relation is sought to be maintained between the robot end-effector's force exerting to an object constraining the end-effector and position displacement toward the direction vertical to the object’s surface \(^\text{50}\). In hybrid control, the end-effector’s force is explicitly controlled in selected directions and the end-effector’s position is controlled in the remaining (complementary) directions \(^\text{52}\).

The hybrid control approaches can be further classified into three main categories \(^\text{40}\); (A) explicit (model based) hybrid control of rigid robots in elastic contact with a compliant environment, e.g., \(^\text{53}-\text{54}\), in which the end-effector force is controlled by directly commanding the joint torques of the robot based on the sensed force error; (B) implicit (position/velocity based) hybrid control of rigid robots in elastic contact with a compliant environment, e.g., \(^\text{55}\), in which the end-effector force is controlled indirectly by modifying the reference trajectory given into an inner loop joint position/velocity controller based on the sensed force error;
and (C) explicit (model based) hybrid control of rigid robots in hard contact with a rigid environment, e.g., \cite{52, 56}.

Nakamura have used the following matrix equation below in \cite{45}, so both constraint condition and the dynamics can be represented simultaneously. This equation means that determining the constraint force $F_n$ does not include time integration like $\ddot{q}$. In this equation $\ddot{q}$ express the angular acceleration, and $F_n$ express the constraint force. Furthermore, he extended the dynamics representation method into the concept of dynamics filter \cite{46, 47}.

\[
\begin{bmatrix}
M & -J^T_c \\
\frac{\partial C}{\partial q} & 0
\end{bmatrix}
\begin{bmatrix}
\ddot{q} \\
F_n
\end{bmatrix}
= 
\begin{bmatrix}
\tau - \frac{1}{2}M\ddot{q} - N\dot{q} - G - J^T_t F_t \\
-\ddot{q}^T [\frac{\partial C}{\partial q} (\frac{\partial C}{\partial q})] \ddot{q}
\end{bmatrix}
\]  \tag{9.1}

Furthermore, many researches have discussed on the constraint-combined force/position hybrid control method. To ensure the stabilities of the constrained motion, those force and position control methods have utilized Lyapunov's stability analysis under the inverse dynamic compensation where force control strategies have been explained intelligibly in papers \cite{40, 41, 42}. But these stability proofs are trying to divide the procedure into two different parts \cite{48, 49}: $\lim_{t \to \infty} F_n = F_{nd}$ and $\lim_{t \to \infty} \tau = \tau_d$, here $F_n$ and $F_{nd}$ are the actual constraint force and the desired constraint force, while $\tau$ and $\tau_d$ are the actual hand position of the manipulator and the desired one.

In \cite{51} it is written that “If contact is modeled by means of geometric constraints, then the contact forces cannot be expressed as algebraic functions of the state variables $q, \dot{q}$.” The $q, \dot{q}$ express the angle and angular velocity of the joints. We do not think it is right, because the contact force has been calculated in (9.2). (9.2) is derived from (9.1), which has been pointed out by Hemami \cite{57} in the analysis of biped walking robot, denotes clearly the algebraic relation between the input torque $\tau$ of the robot and exerting force to the working object $F_n$ have algebraic relation, when robot’s end-effector being in touch with a surface in 3-D space:

\[F_n = a(q, \dot{q}) + A(q)J^T_t F_t - A(q)\tau.\]  \tag{9.2}

Where $a(q, \dot{q})$ and $A(q)$, $J_t$ are scalar function and vectors defined in following section. (9.2) exhibits vector $\tau$ determining $F_n$ has a redundancy against constraint force $F_n$ since $F_n$ is
In this paper, the third category (C) of contacting situation that assumes rigid link manipulator and hard contacting with nonelastic environment. From (9.2) we can know that the force transmission process is an immediately finished process for a rigidly structured manipulator just as the acceleration being determined immediately by state variables and input generalized forces. Exploiting (9.2), we design a new controller whose stability is guaranteed by Lyapunov method, which exerting force \( F_n(t) = F_{nd} \) and \( \lim_{t \to \infty} r = r_d \). The effectiveness has been confirmed by a 2-link grinding robot model in simulation.

9.2 Analysis of Grinding Task

There are four kinds of grinding processes in common use, called respectively vertical surface grinding, horizontal surface grinding, internal grinding and cylindrical grinding. A grinding machine usually can only perform one or two of these processes because of kinematic limitation. However, all of the four kinds of tasks can be finished by a single robot manipulator for its dexterity in movement. To do so, the grinding wheel has to contact with the workpiece. A set of contacting surfaces, especially the surfaces being machined, will form constraints to the motions of the grinding wheel. As for vertical surface grinding operation shown in Fig.9.1(a), the grinding wheel in contact with a surface of the work-piece is not free to move through that surface, which forms a position constraint. And also, the wheel cannot freely apply arbitrary force tangent to the surface in case of no disturbing force like friction existing, which forms a set of force constraint. Situations of constraints for other kinds of grinding tasks are shown in Fig.9.1(b), (c) and (d).

In general, the desired grinding position trajectory is given by processing drawings for each grinding procedure, which the grinding allowance is considered. As for grinding forces \( F_n, F_t \) and \( F_s \), the desired values should be determined carefully for different grinding conditions. Generally speaking, the grinding power is related to the metal removal rate (weight of metal being removed within unit time), which is determined by the depth of cut, the width of cut, the linear velocity of the grinding wheel, the feed rate and so on. There are many empirical formula available for the determination of grinding power, and the desired force trajectory can then be planned according to the power. The normal grinding force \( F_n \) is exerted in the
perpendicular direction of the surface. It is a significant factor that affects ground accuracy and surface roughness of workpiece. The value of it is also related to the grinding power or directly to the tangential grinding force as

$$F_t = K_t F_n,$$  \hspace{1cm} (9.3)

where, \(K_t\) is an empirical coefficient, \(F_t\) is the tangential grinding force. This relation gives us an estimated value of \(F_t\) given that \(K_t\) and \(F_n\) are known.

The axial grinding force \(F_s\) is proportional with the feed rate, and is much smaller than the former force.

Eq. (9.3) is based on the situation that position of the grinding cutter is controlled like currently used machining center. But when a robot is used for the grinding task, the exerting...
force to the object and the position of the grinding cutter should be controlled simultaneously. The $F_n$ is generally determined by the constrained situation, and it is not suitable to apply (9.3) to grinding motion by the robots.
10 Modeling of Constraint Control System

10.1 Dual Constrained Dynamic Systems

Hemami and Wyman have addressed the issue of control of a moving robot according to constraint condition and examined the problem of the control of the biped locomotion constrained in the frontal plane. Their purpose was to control the position coordinates of the biped locomotion rather than generalized forces of constrained dynamic equation involved the item of generalized forces of constraints. And the constrained force is used as a determining condition to change the dynamic model from constrained motion to free motion of the manipulators. In this paper, the grinding manipulator whose end-point is in contact with the constrained surface, is modeled according with Lagrangian equations of motion in term of the constraint forces, referring to what Hemami\(^\text{57}\) and Arimoto\(^\text{48}\) have done:

\[
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial q}\right) = \tau + \left(\frac{\partial C}{\partial q}\right)^T \frac{1}{\left\| F_n - \left(\frac{\partial r}{\partial q}\right)^T \hat{r}\right\|} \hat{r} - \left(\frac{\partial C}{\partial \dot{r}}\right)^T \frac{1}{\left\| F_t \right\|} F_t
\]

(10.1)

Fig. 10.1: Model of Constraint Dynamic System
where, $J_c$ and $J_r$ are defined as:

$$J_c = \frac{\partial C}{\partial q^T} \| \frac{\partial C}{\partial r} \|, \quad J_r = \frac{\partial r}{\partial q^T}, \quad J_r^T = J_r^T \frac{\dot{r}}{\| \dot{r} \|},$$

$r$ is the position vector of the hand and can be expressed as a kinematic equation,

$$r = r(q).$$  \hfill (10.2)

$q$ is $n(\geq 2)$ generalized coordinates. Then this manipulator does not have kinematic redundancy. In this research we only discuss the problem under only one constraint condition, so $C$ is a scalar function of the constraint, and is expressed as an equation of constraints,

$$C(r(q)) = 0.$$  \hfill (10.3)

$F_n$ is the scalar express the value of the constrained force associated with $C$ and $F_t$ is the scalar express the value of tangential friction force.

In (10.1) can be derived into:

$$M(q)\ddot{q} + \frac{1}{2} \dot{M}(q)\dot{q} + N(q, \dot{q})\dot{q} + G(q) = \tau + J_c^T(q)F_n - J_r^T(q)F_t,$$  \hfill (10.4)

here we express $M(q)$ as $M$ and $N(q, \dot{q})$ as $N$ for short. $M$ is an $n \times n$ matrix, $N$ is a $n \times n$ skew-symmetric matrix. $G$ is a $n$ row vectors. $\tau$ is $n$ inputs.

From the constraint condition (10.3) we can get

$$\frac{\partial C}{\partial q^T} \dddot{q} = -\frac{q^T}{\| \dot{q} \|} \left[ \frac{\partial C}{\partial q^T} \dot{q} \right] \| \dot{q} \|.$$  \hfill (10.5)

The equation (10.4) and (10.5) can be combined as follows which is the same equation of (9.1):

$$\begin{bmatrix} M & -J_c^T \\ \frac{\partial C}{\partial q^T} & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_n \end{bmatrix} = \begin{bmatrix} \tau - \frac{1}{2} \dot{M}q - N\dot{q} - G - J_c^TF_n \\ -\dot{q}^T \left[ \frac{\partial C}{\partial q^T} \dot{q} \right] \| \dot{q} \| \end{bmatrix}.$$  \hfill (10.6)

This equation is also used by Nakamura in (45), (46) and (47), it is easy to see that when the matrix on the left side is invertible, there exist a $\tau$ which determines the $\ddot{q}$ and $F_n$ satisfying (10.4) and (10.5) respectively. And as below, the inertia matrix combined with constraint condition is guaranteed to be non-singular.
The inverse matrix can be calculated as follow:

\[
\frac{\partial C}{\partial q^T} \ddot{q} = -\dot{q} \left[ \frac{\partial C}{\partial q} \frac{\partial C}{\partial \dot{q}^T} \right] \dot{q}
\]  

(10.5)

From (10.6) and (10.9) the constraint force \( F_n \) being identical to (9.2) can be solved like:

\[
F_n = \left( \frac{\partial C}{\partial q^T} \right) M^{-1} \left( \frac{\partial C}{\partial q^T} \right)^T \left( -\dot{q} - \left( \frac{\partial C}{\partial q^T} \right) \ddot{q} \right) + \left( \frac{\partial C}{\partial q^T} \right) M^{-1} \left( \frac{1}{2} \dot{M} \ddot{q} + N \ddot{q} + G(q) + J^T F_i \right)
\]

\[
F_n = a(q, \dot{q}) + A(q) J^T F_i - A(q) \tau,
\]  

(10.10)

where, \( a(q, \dot{q}) \) is a scalar representing the first term in the expression of \( F_n \), and \( A(q) \) is an \( n \) line vector. As shown clearly in (10.10) that dimension of \( \tau \) is larger than the dimension...
of $F_n$, and $F_n$ can be realized in the range space of $A(q)$. This means $\tau$ has a kind of redundancy against $F_n$. We named this redundancy appearing always in constraint dynamics of manipulator as constraint redundancy. $a(q, \dot{q})$ and $A(q)$ are defined concretely as follow:

$$a(q, \dot{q}) \triangleq m_c^{-1} \frac{\partial C}{\partial r^T} \{ -\frac{\partial}{\partial q^T} \frac{\partial C}{\partial q} \dot{q} \dot{q} + (\frac{\partial C}{\partial q^T}) M^{-1} \left( \frac{1}{2} M \ddot{q} + N \dot{q} + G \right) \}$$

(10.11)

$$A(q) \triangleq m_c^{-1} \frac{\partial C}{\partial r^T} \{ (\frac{\partial C}{\partial q^T}) M^{-1} \}$$

(10.12)

(10.10) is written as follow for short:

$$F_n = F_n(q, \dot{q}, \tau, F_t).$$

(10.13)

From (10.6) and (8.48), we can get that:

$$\ddot{q} = M^{-1} (\tau - \frac{1}{2} M \ddot{q} - N \dot{q} - G - J_r^T F_t + J_e^T F_n).$$

(10.14)

Inserting $F_n$, (10.6) into (10.14), the state equation of the system excluding the constrained force (as $F_n > 0$) can be rewritten as

$$M(q) \ddot{q} + \frac{1}{2} M(q) \dot{q} + N(q, \dot{q}) \dot{q} + G(q)$$

$$= J_e^T(q) a(q, \dot{q}) + (I - J_e^T A) \tau + (J_e^T A - I) J_r^T F_t$$

(10.15)

which is denoted as a model of the constraint dynamic system in Fig. 10.1.

Solutions of these dynamic equation always satisfy the constrained condition (10.3). The forward description of contacting dynamics has been represented by (10.4) and (10.3). The fact that the solution $q$ of (10.4) have to satisfy (10.3) make us anticipate that $F_d$ should be satisfied simultaneously and instantly regardless of the motion $q, \dot{q}$, and any $\tau$. The algebraic solution has been derived as (10.10). Then the dynamics of the manipulator whose solution $q, \dot{q}$ always satisfy the constraint condition (10.5) derived from (10.3) has been translated into (10.15). In the Fig. 3, the backward relation of (10.10) and (10.15) are described in the right hand half. (10.10) exhibits clearly the comment “If contact is modeled by means of geometric constraints, then the contact forces cannot be expressed as algebraic functions of the state variables $q, \dot{q}$,” in pp.55 51) is not correct and contradicting to. The backward description of constraint dynamics has been long ignored by robotic researchers, but we had proposed the force sensorless position/force control based on using this backward description
Fig. 10.3: Grinding position / force control system

directly \(^{(43)}\). Here two descriptions on left and right side are equivalent, then it can be called a “dual system.” In this paper, we propose a new controller with Lyapunov-stability over non-constraint motion and with instantaneous achievement of desired contacting force.

10.2 Robot Controller under Constraint Condition

Let \( S \) be a column full rank matrix spanning the null space of \( \partial C / \partial q \), we can get \( S^T \left( \partial C / \partial q \right)^T = 0 \), i.e.

\[
S^T J_c^T = 0.
\] (10.16)

It is possible to find an auxiliary vector \( p \) satisfies

\[
\dot{q} = S \dot{p}.
\] (10.17)

in the manipulator situation, \( p \) is the end-effector position except the constraint direction, and

\[
\ddot{q} = \dot{S} \dot{p} + S \ddot{p}.
\] (10.18)

From the definition of \( A \) in (10.12) and \( S \) in (10.16) we know that \( [A^T, S]^T \) is reversible, i.e. there certainly exist \( B \) \((n \times 1 \) vector\) and \( D \) \((n \times (n - 1) \) matrix\) satisfies

\[
\begin{bmatrix}
A \\
S^T
\end{bmatrix} B = \begin{bmatrix} 1 \\
0
\end{bmatrix}
\] (10.19)

\[
\begin{bmatrix}
A \\
S^T
\end{bmatrix} D = \begin{bmatrix} 0^T \\
I_{n-1}
\end{bmatrix}
\] (10.20)
respectively. Here $I_{n-1}$ is a $(n-1) \times (n-1)$ identify matrix. $B$ means selection matrix of range space of $\partial C/\partial q$, which corresponds directly to the range space of $A(q)$ as shown in (10.12) and null space of $\partial C/\partial q^T$. And $D$ is vice versa.

Before propose the controller we will put forward three assumptions:

(a) The constraint condition is known and expressed by $C(r(q)) = 0$.
(b) The tangential grinding force can be calculated by (9.3).
(c) The dynamic parameters of the system are known.

The following is a controller guaranteeing that the closed loop satisfies the exerted constrained force $F_n$ be identical to the desired force $F_{nd}$ regardless of time and the robot’s motion along with the free motion directions.

$$\tau = B(F_{nd} - a) + D[k_p(p_d - p) + k_d(\dot{p}_d - \dot{p})] + J_t^T F_t$$

(10.21)

Here on the right side, the first term is to realize the desired constrained force, the second term is to control the pose of the manipulator, while the third item is to compensate the friction, with an assumption of $F_t$ being able to be gotten correctly. This assumption can be materialized by using $F_n$ and (9.3). The block diagram of the system is given in Fig. 10.3.

Because (9.2) is a algebraic function of the input torque, when we substitute (10.21) into (10.10), we can get

$$F_n = a(q, \dot{q}) + AB(F_{nd} - a) + AD[k_p(p_d - p) + k_d(\dot{p}_d - \dot{p})]$$

(10.22)

from the definition we know that $AB = 1$ and $AD = 0$ so

$$F_n = F_{nd},$$

(10.23)

here (10.23) does not include the variable of time $t$ as a time differential manner, meaning the output force always equals the desired one.

### 10.3 Calculation of the Variables

From definition of Jacobian matrix $J$ we can get

$$\dot{r} = J\dot{q}$$

(10.24)

there exists $P$ which satisfies:

$$\dot{p} = PJ\dot{q} = \ddot{J}q$$

(10.25)
Define $PJ$ as $\tilde{J}$, taking (10.25) into (10.17) we can get

$$\tilde{J}S\dot{p} = \dot{p}. \quad (10.26)$$

$\tilde{J}$ is a row full rank matrix of $(n - 1) \times n$, so it is possible to find a $S$ which satisfying $\tilde{J}S = I_{n-1}$ is a solution of (10.26) i.e. $\tilde{J}S = I$ satisfies (10.17), and $\tilde{J}S = I_{n-1}$ and (10.16) can be combined into one matrix equation:

$$\begin{bmatrix} \tilde{J} \\
J_c \end{bmatrix} S = \begin{bmatrix} I_{n-1} \\
0 \end{bmatrix} \quad (10.27)$$

Here we define $\hat{J} = [\tilde{J}^T, J_c^T]^T$. $\tilde{J}$ expresses range space of robot’s hand motion not being constrained by object, and $J_c$ represents constraint direction, which is orthogonal to row vectors of $\tilde{J}$. Then $\hat{J}$ is reversible, so $S$ can be calculated as follow:

$$S = \hat{J}^{-1} \begin{bmatrix} I_{n-1} \\
0 \end{bmatrix} . \quad (10.28)$$

By using $S$, we can calculate $B$ and $D$ in (10.19) and (10.20), and calculate the input $\tau$ in (10.21).

### 10.4 Stability Analysis

Taking (10.17) and (10.18) into (10.4) we get:

$$M\ddot{S}\dot{p} + MS\ddot{p} + \left(\frac{1}{2}\ddot{M} + N\right)S\dot{p} = \tau + J_c^TF_n - J_r^T(q)F_t \quad (10.29)$$

Premultiply $S^T$ we can get

$$S^T M\ddot{S}\dot{p} + S^T MS\ddot{p} + S^T \left(\frac{1}{2}\ddot{M} + N\right)S\dot{p} = S^T \tau + S^T J_c^TF_n - S^T J_r^TF_t \quad (10.30)$$

Substituting (10.21) into (10.30) we can get

$$S^T M\ddot{S}\dot{p} + S^T MS\ddot{p} + S^T \left(\frac{1}{2}\ddot{M} + N\right)S\dot{p} = k_p(p_d - p) + k_d(\dot{p}_d - \dot{p}) \quad (10.31)$$

here we set the desired end-effector is static which means $\dot{p}_d = 0$, so closed loop dynamics is

$$S^T M\ddot{S}\dot{p} + S^T MS\ddot{p} + S^T \left(\frac{1}{2}\ddot{M}\right)S\dot{p} - k_p(p_d - p) = -S^T NS\dot{p} - k_d\dot{p} \quad (10.32)$$
set Lyapunov argument as:

\[ V = \frac{1}{2} \dot{p}^T S^T M S \dot{p} + \frac{1}{2} (p_d - p)^T k_p (p_d - p) \]  

(10.33)

so

\[ \dot{V} = \dot{p}^T S^T M \dot{S} \dot{p} + \dot{p}^T S^T M S \dot{p} + \dot{p}^T S^T \frac{1}{2} M S \dot{p} - k_p \dot{p}^T (p_d - p) \]  

(10.34)

from (10.32), (10.34) can be transformed to

\[ \dot{V} = -\dot{p}^T S^T N S \dot{p} - k_d \dot{p}^T \dot{p} \]  

(10.35)

because \( N \) is a skew symmetrical matrix, \( \dot{p}^T S^T N S \dot{p} = 0 \), so

\[ \dot{V} = -k_d \dot{p}^T \dot{p} \]  

(10.36)

because \( \dot{V} \leq 0 \) and \( V \geq 0 \), from (10.36) we can see if and only if \( \dot{p} = 0 \), \( \dot{V} = 0 \), which means that submitting \( \dot{p} = 0 \) into (10.33) we can get \( p_d - p = 0 \) from Lasalle theorem we know that

\[ \lim_{t \to \infty} p = p_d, \lim_{t \to \infty} \dot{p} = 0 \]  

(10.37)

Because the constraint system satisfies that \( C(r(q)) = 0 \),

\[ \lim_{t \to \infty} r = r_d. \]  

(10.38)

So the system will converge to the desired pose at last.
11 Simulation

11.1 2-link Grinding Robot Model

In this section I will introduce some simulations have been done to check the controller in 2-link condition as Fig. 11.1. To a 2-link manipulator the variables in (10.15) can be calculated as follow:

\[
M = \begin{bmatrix} J_1 + J_2 + 2\beta \cos q_2 & J_2 + 2\beta \cos q_2 \\ J_2 + 2\beta \cos q_2 & J_2 \end{bmatrix}
\]  

(11.1)

\[
\frac{1}{2} \dot{M} \dot{q} + N \dot{q} = \begin{bmatrix} -(2\dot{q}_1 \dot{q}_2 + \dot{q}_1)\beta \sin q_2 \\ \dot{q}_1^2 \beta \sin q_2 \end{bmatrix}
\]  

(11.2)

Here \(J_1 = I_1 + (m_1 + 4m_2)l_1^2\), \(J_2 = I_2 + m_2l_2^2\) and \(\beta = 2m_2l_1l_2\) and \(I, m, l\) are the initial moment, mass and length of the links. Jacobian matrix \(J\) is.

\[
J = \begin{bmatrix} -\sin q_1 - \sin(q_1 + q_2) & -\sin(q_1 + q_2) \\ \cos q_1 + \cos(q_1 + q_2) & \cos(q_1 + q_2) \end{bmatrix}
\]  

(11.3)

In the simulation we set the constraint condition as:

\[
C(r(q)) = y - 0.6 = 0
\]  

(11.4)

So

\[
\frac{\partial C}{\partial r} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \dot{r} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]  

(11.5)

So

\[
J_c = \begin{bmatrix} \cos(q_1 + q_2) \\ \cos q_1 \end{bmatrix}, \quad J_r = \begin{bmatrix} -\sin(q_1 + q_2) \\ -\sin q_1 \end{bmatrix}
\]  

(11.6)

From the variables above we can calculate \(a, A\) and \(m_c\) defined in (10.11) (10.12) and (10.8) and calculate \(F_n\) by (10.10).

For 2-link manipulator, \(S\) and \(\partial C/\partial q\) in (10.16) are both \(2 \times 1\) vectors, we can get

\[
S^T \left( \frac{\partial C}{\partial q^T} \right) = 0, \quad \text{i.e.}
\]

\[
S^T J_c^T = 0
\]  

(11.7)
This $S$ also satisfies the following equation,

$$\dot{q} = S\dot{p}_x$$  \hspace{1cm} (11.8)

where $\dot{p}_x$ is the end-effector position on the x-axis, here we define the two elements of $S$ as $[S_1, S_2]$ and $J_c = [J_{c1}, J_{c2}]$, also the two angles of the joints are defined as $q_1$ and $q_2$ respectively, so (11.8) can be written as

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} \dot{p}_x$$  \hspace{1cm} (11.9)

from the definition of Jacobian matrix we can know

$$[J_{11}, J_{12}] \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \dot{p}_x$$  \hspace{1cm} (11.10)

here $[J_{11}, J_{12}]$ is the first line of Jacobian matrix and also the $\dot{J}$ in (10.25). To get $S$ satisfies $\dot{J}S = I$, take (11.9) into (11.10), we can get

$$[J_{11}, J_{12}] \begin{bmatrix} S_1 \\ S_2 \end{bmatrix} = 1$$  \hspace{1cm} (11.11)

so $\dot{J}$ in (10.28) is

$$\dot{J} = \begin{bmatrix} J_{11} & J_{12} \\ J_{c1} & J_{c2} \end{bmatrix}$$  \hspace{1cm} (11.12)

and

$$S = \dot{J}^{-1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{J_{c2}}{J_{c2}J_{11} - J_{c1}J_{12}} \\ -\frac{J_{c1}}{J_{c2}J_{11} - J_{c1}J_{12}} \end{bmatrix}.$$  \hspace{1cm} (11.13)

Because there are only 2 links, $B$ and $D$ in (10.21) can be simplified as follow

$$\begin{bmatrix} A \\ S^T \end{bmatrix} B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$  \hspace{1cm} (11.14)

$$\begin{bmatrix} A \\ S^T \end{bmatrix} D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$  \hspace{1cm} (11.15)

### 11.2 Simulation Result

Then we did some simulations to check the controller, first we set the desired as a step input, in this simulation $x_d = 0.5[m]$, $F_{nd} = 5[N]$ $k_p = 1000$ and $k_d = 300, 100, 30$ respectively, the
Fig. 11.1: 2-link grinding robot

(a) Actual and desired constraint force
(b) Actual and desired trajectory on x-axis
(c) Actual and desired trajectory on y-axis

Fig. 11.2: Simulation result when \( k_p = 1000, x_d = 0.5[m] \) and \( F_{nd} = 5[N] \)

result is shown in Fig. 11.2, the constraint force and \( y \)-position are coincide and all \( x \)-position can converge to the desired position.

In the second simulation I will show the data of the simulation when \( k_p = 1000, k_d = 300, x_d = 0.06t[m] \) and \( F_{nd} = 5[N] \). From Fig. 11.3 (a) we can see that the system will output the desired force. And from (b) and (c) we can see that the controller can control the end-effector move along the desired trajectory.

In the third simulation we make the desired force as a function of time, in this simulation \( k_p = 1000, k_d = 300, x_d = 0.06t[m] \) and \( F_{nd} = 6 + sint[N] \) the result is shown in Fig. 11.4.
Fig. 11.3: Simulation result when $k_p = 1000$, $k_d = 300$, $x_d = 0.06t[m]$ and $F_{nd} = 5[N]$.

Fig. 11.4: Simulation result when $k_p = 1000$, $k_d = 300$, $x_d = 0.06t[m]$ and $F_{nd} = 5 + \sin t[N]$.
12 Conclusion

The first part of the paper is concerned with 3-D position/orientation measurement of an object and 6-DoF visual servoing. We use model-based method to recognize 3-D pose in real-time. The maximum matching degree represents the model matching the target best. Then the 3-D recognition of the target object can be solved by finding the valuables of position/orientation that give the maximum value of the correlation function through the online optimization method, "1-Step GA". Unit quaternion is used to represent the orientation of the target object, which has an advantage that can represent the orientation of a rigid body without being annoyed by representation singularities. The singularities cause multi-solutions for a single orientation, resulting in making the GA difficult to converge to the right variable.

We stress that improvement of the dynamics of the sensing unit is important for a stable visual servoing, and proposed a MFF method to improve dynamics in visual recognition, with compensating the fictional motion of the target in the camera images based on kinematics of the manipulator, by extracting the real motion of the target. The enhanced dynamics of recognition gave further stability and precision to the total visual servoing system, evaluated by full 6-DoF servoing experiment using 7-link manipulator. The convergence time in step response was about 10[s] and precise visual servoing to a moving target object has been achieved.

Moreover, we proposed a new two-way visual servoing method: hand & eye-vergence visual servoing. Our proposed method includes two loops: an outer loop for conventional visual servoing that direct a manipulator toward a target object and an inner loop for active motion of binocular camera for accurate and broad observation of the target object. The effectiveness of the hand & eye-vergence visual servoing is evaluated through simulations incorporated with actual dynamics of 7-DoF robot and considering the dynamics of the rotatable cameras which are installed on the same link on the view points of how the new
idea improved the trackability and stability in visual servoing dynamics and the accuracy of hand pose.

As future research, we will consider about using the redundancy of the manipulator during the controlling like obstacle avoidance, force control will be combined to our visual servoing system, we are looking forward to building a human-like smart robot that can help us, communicate with us and live with us.

The second part of paper is concerned with the manipulator control under constraint condition. Constraint dynamic model of manipulator can be expressed in two ways separately which are dual system each other. Manipulator's hand tip is used as a position sensor, to supply those necessary information for this proposed force and position control methodology. Hence, the system is controlled without any force or torque sensor.

We designed control law which presented is constructed by using the dynamical redundancy of constraint systems. The controller designed by this control law can be used for simultaneous force and position control. and prove the convergence of the controller in by Lyapunov method, the output force of the system can always equal to the desired one.

Then we confirmed the controller in a 2-link grinding robot in simulation. In the future we will use spline function to predict the outline of the constraint surface and apply it into experiment.
REFERENCE


