Thèse présentée pour obtenir le grade de
Docteur de l'Université Louis Pasteur
Strasbourg I

Discipline : Électronique,Électrotechnique, Automatique
Spécialité : Robotique

par Kanako Miura

Robot Hand Positioning and Grasping Using Vision

Positionnement et saisir par une main robotique à l'aide de la vision

Soutenue publiquement le 3 Février 2004

Membres du Jury

Directeur de Thèse :	M. Michel de Mathelin, Professeur, Université Louis Pasteur
Directeur de Thèse :	M. Hikaru Inooka, Professeur, Université de Tohoku
Examinateur :	M. Koichiro Deguchi, Professeur, Université de Tohoku
Examinateur :	M. Eiji Nakano, Professeur, Université de Tohoku
Invité :	M. Jacques Gangloff, Maître de conférences, Université Louis Pasteur
Rapporteur Externe :	M. Nicolas Chaillet, Professeur, Université de Franche-comté
Rapporteur Externe :	M. Koichi Hashimoto, Professeur, Université de Tokyo
Rapporteur Interne :	M. Ernest Hirsch, Professeur, Université Louis Pasteur

Acknowledgement

I wish to express my greatest gratitude to Professor Hikaru Inooka, Laboratory of Intelligent Control Systems, Graduate School of Information Sciences, Tohoku University, and Professor Michel de Mathelin, Equipe d'Automatique, Vision et Robotique, Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection, Strasbourg I University. Without their continuous encouragement, advice, and assistance, I would not have accomplished my doctor thesis.

Sincere thankfulness is also due to Professor Eiji Nakano and Professor Koichiro Deguchi, Grasuate School of Information Sciences, Tohoku University, for their serving on the graduate committee and precious comments on this thesis.

I would like to show my appreciation to Professor Nicolas Chaillet, Besançon University, Associate Professor Koichi Hashimoto, Tokyo University, and Professor Ernest Hirsch, Strasbourg I University, for their serving as referee on the precedent thesis committee and precious comments on this thesis.

I would especially like to thank Associate Professor Jacques Gangloff, Strasbourg I University, and Research Assistant Nobuaki Nakazawa, Gunma University, for their tireless guidance and assistance on my research work.

I am greatly indebted to Professor Tadashi Ishihara, Fukushima University, Research Assistant Takahiko Ono, and Research Assistant Yasuaki Ohtaki, Tohoku University. I learned much from them.

I am also grateful to secretary Ms. Noriko Gyoba, Tohoku University, Professor Joana Carvalho-Ostertag, Strasbourg I University, and fellow students and researchers in LICS and EAVR for their kindness and help.

I also owe Professor Danièle Alexandre, former vice-president of Robert Schuman University, who integrated me into Strasbourg consortium.

My stay in France (from April 2001 to July 2002) was supported by Renault Foundation, which is greatly acknowledged.

Contents

List of Figures

List of Tables

Chapter 1

General introduction

Recently, significant developments have been made in the design of practical robot manipulators and hands that can perform various manipulation tasks required in different fields. However, most industrial robots have been designed to perform only specific movements based on a priori knowledge of the object to be manipulated. Therefore they cannot accomplish tasks when the target object (e.g., object mass, shape, or position) is unknown, or when the relative position of the vision system with respect to the robot is unknown.

In this thesis, the total grasping task is investigated. The manipulator has an uncalibrated camera system and the simple precision gripper has two fingers on the end-effector. Then, the problem is divided into two tasks; the positioning task of the manipulator, and the grasping task of the robot hand.

1.1 Positioning task using vision

The positioning task of a robot with respect to an object is a main application of visual servoings. There exist different classification to describe visual servoing schemes. Firstly, when the vision system is used to provide set-point inputs to the joint-level controller, thus making use of joint feedback to internally stabilize the robot, it is referred to as *indirect visual servoing* or *dynamic look-and-move*. In contrast, *direct visual servoing* [76] eliminates the robot controller entirely, replacing it with a visual servo controller that directly computes joint inputs, thus using vision alone to stabilize the mechanism. Many visual servoing systems employ "indirect"

rather than "direct" visual servoing, because of low vision sample rates and high latencies. Secondly, most approaches can be categorized as either *position-based visual servoing* (PBVS [78] [75]) or *image-based visual servoing* (IBVS [4] [11]). In position-based control, features are extracted in the image and used in conjunction with a geometric model of the target and the known camera model to estimate the pose of the target with respect to the camera and the robot. In image-based servoing, control values are directly computed on the basis of image features. The image-based approach may reduce computational delay, eliminate the necessity for image interpretation and eliminate errors due to sensor modeling and camera calibration. Finely note that a third classification is *static-eye* and *hand-eye* configuration. In static-eye configuration, the camera is fixed in the environment and observes the robot. Position-based and image-based control laws are compared in [25], and it is clear that position-based control laws are more sensitive to calibration errors.

For our research problem, the relative pose between the end-effector of the robot and the object cannot be robustly computed. Therefore, image-based control laws are adopted. Most of the previous works on visual servoing assume that the kinematic model of the robot, a model the object, and the camera intrinsic parameters are known. They would fail if the robot and the vision system were not fully known. We employ an indirect (look-and-move) scheme for the versatility and stability brought by the internal joint controllers. A novel approach for uncalibrated and model-less visual servoing using a modified *simplex* iterative search method is proposed. The kinematic model of the robot may be unknown; however, we assume that the joint limits are known. The intrinsic parameters of the camera are also unknown, and a model of the object is not necessary.

The simplex method is an unconstrained optimization technique [67]. The basic idea behind this method is to compare the value of the objective function in several configurations and to move to the next configurations in order to decrease this value. In our approach, the objective function to be minimized is a cost function measuring the difference between the image of the object in the desired position of the robot and the image in the current position. The variables usually are the joint angles of the robot, but it is not a requirement. The main idea is to use the optimization algorithm in real-time to move the robot to the next position that makes the cost function decrease. Our approach may include cost functions that are not differentiable since it does not require the computation of Jacobian matrices. Demonstrations with a

6DOF industrial manipulator show the efficiency of this method.

1.2 Grasping task applying human features

Analytical studies of grasping and manipulation by robot hands have been done by many researchers. Yoshikawa and Nagai [79] divide the finger forces into two different forces: manipulation force and internal force. The manipulation force is defined as the force that generates the required force to compensate for gravity forces on the object. Salisbury (see [69], [70]) defines the internal force whose resultant is zero as the force that can be added when necessary to the manipulation force to prevent sliding or breaking contact. Shimoga and Goldenberg present in [71] the model and control of the impedance of a soft finger. They show experimentally how the presence of passive damping from soft materials can reduce the peak impact forces that occur when a rigid object is grasped by the fingers of a robotic hand. In [32], the authors analyze the stiffness models of human grasping and present the grasp stiffness model useful for modeling and controlling robotic manipulators. While the grasping, manipulation and internal forces are well understood their implementation is limited due to their complexity.

Another approach to the control of robot hands is a physiological approach motivated by the study on human hands. The study of human grasping has long been an area of interest for hand surgery to design prosthetic devices. The study also can lead to a grasp taxonomy that can assist robot hands in choosing an appropriate grasp to perform a specific task and can also be used to design efficient grippers [6] [7]. Flanagan and Wing [12] have investigated the stability of precision grasping forces behaviors. When a human handles and carries an object, the grasping force is delicately adjusted to the inertial force occurred by its movement [34]. This characteristic is applied as a control law in the grasping task of the robot hand in [57]. A hand-over motion from human to a robot has been realized in [36], emulating human's hand-over motions.

Though a lot of researches have been done about motions to lift-up or manipulate an object, the control of the motions to contact an object for grasping has not been investigated yet. Human have the ability to touch an object without inducing large displacements even if it is light and could easily fall. Though such skill is very important when the object is fragile, few investigations have been made so far on soft

grasping. Furthermore, it is not applied yet to control laws of robot hands. In this thesis, experimental studies are carried out on the human grasping with the index finger and the thumb (precision grasp). The contact force at the fingertip which contacts first the target object, and the displacement of the object are analyzed. Motions of human fingers are also investigated for objects of deformable width. The results reveal the function of the index finger and the thumb in grasping. The features of contact motions given by the measurement of human motions are applied to a robot grasping task. The "soft" contact motion is demonstrated with a robot hand with two fingers controlled individually. Each finger has two pairs of strain gauges as force sensors. A vision system is also available with a camera for real-time visual feedback.

1.3 Organization of this Dissertation

This thesis is structured as follows. Part I treats of the positioning task using vision while Part II treats of the soft grasping of fragile objects.

Chapter 2 in Part I is dedicated to a short summary of existing visual servoing algorithm. Existing uncalibrated visual servoing are described. The Gauss-Newton optimization technique is shown as an example of the existing methods. The main idea is to use the optimization algorithm in real-time to move the robot to the next position that makes the objective function decrease. The objective function to be minimized is a cost function measuring the difference between the image features of the object in the desired position of the robot and the image features in the current position. The variables are the joint angles of the robot. When the Jacobian of the cost function with respect to the joint variables of the robot is unknown, a real-time identification is performed. However, this type of adaptive algorithm cannot be stable when there is not enough informations for the identification.

In Chapter 3, PartI, a simplex optimization technique is presented for uncalibrated and model-less visual servoing. In this method, the kinematic model of the robot is not necessary; however, we assume that the joint limits are known. The intrinsic parameters of the camera are also unknown. Simulations are carried out for simple target centering tasks with 2DOF and 6DOF.

In Chapter 4, PartI, a positioning task to approach a target object is described. Assuming the kinematic model of the robot to be unknown, a modified simplex

optimization method is used to achieve the positioning task. The objective function to be minimized is a cost function measuring the difference between the ideal image in the desired position of the robot and the actual image in the current position. The variables are the joint angles of the robot. Modifications are brought to this visual servoing algorithm to improve convergence. Several configurations and objective functions are compared with simulations and experiments on 6DOF industrial robot.

In Chapter 5, Part II, the human flexible grasping control ability is investigated. Human grasping motions are measured changing the weights and sizes of the object, and the speed of the grasping motion. The human control features are analyzed paying attention to the displacement of the object and the applied force in the first contact.

Chapter 6 in Part II describes the application of the desirable human grasping features to robot hand control. Two fingers of the hand touch a target object almost at the same time like a human motion using visual feedback and synchronization of distance errors.

Part I

Positioning

Chapter 2

Existing approaches

2.1 Introduction

In this chapter we review the fundamentals of visual servoing with a classification of the different approaches. Then, image-based visual servoing is presented and existing works on uncalibrated and model-less visual servoing are reviewed. The simulations are done to show that the conventional optimization approach with on-line estimation of the Jacobian may become unstable.

2.2 Classification of visual servoing algorithms

Vision is a useful robotic sensor since it mimics the human sense of vision and allows non-contact measurements of the environment. Taken to the extreme, machine vision can provide closed-loop position control for a robot end-effector, which is referred to as *visual servoing*.

A good survey of early visual servoing research can be found in [28]. According to the classification by Sanderson and Weiss [76], there exist three main criteria to categorize visual servoing algorithms:

1. **Camera configurations**

    - The camera(s) is fixed in the workspace. It is often called *static-eye* configuration.
    - The camera(s) is mounted on the robot's end-effector. It is often called *hand-eye*, or *eye-in-hand* configuration.

2. **Control level**

   - The vision system is used to provide set-point inputs to the joint-level controller, thus making use of joint feedback to internally stabilize the robot. It is referred to as a *dynamic look-and-move*, or *indirect visual servoing*.
   - The robot controller is entirely replaced with a visual servo controller that directly computes joint inputs, thus using vision alone to stabilize the mechanism. It is referred to as a *direct visual servoing*.

3. **Feedback variables**

   - Features are extracted from the image and used to estimate the pose of the target with respect to the camera. Control values are computed using errors in the estimated pose (Cartesian) space. It is called *position-based visual servoing* (PBVS)
   - Control values are computed on the basis of image features directly. It is called *image-based visual servoing* (IBVS).
   - If part of the pose vector is estimated and part of the control values is computed directly from the image features, it is defined a *hybrid visual servoing* (HVS).

Note that pose vectors as well as other basic definitions are recalled in the appendix.

2.2.1 Camera configurations

Visual servoing systems typically use one of the two following camera configurations: end-effector mounted, or fixed in the workspace.

Static-eye configuration

This configuration has the camera(s) fixed in the workspace as shown in Figure 2.1. In this case, the camera image of the target is independent of the robot motion unless the target is the end-effector itself. So the camera(s) position can be defined relatively to the base of the robot, and to the object independently. A variant of this is for the camera to be agile, mounted on another robot or pan/tilt head in order to observe the visually controlled robot from the best vantage point. In the

static-eye configuration, the relative pose between the end-effector of the robot and the target must be measured or the relative pose between the camera and the base of the robot must be known.

Hand-eye configuration

The second, often called eye-in-hand configuration, has the camera mounted on the end-effector of the robot. Here, there exists a known, often constant, relationship between the pose of the camera(s) and the pose of the end-effector. The figure 2.2 illustrates this configuration.

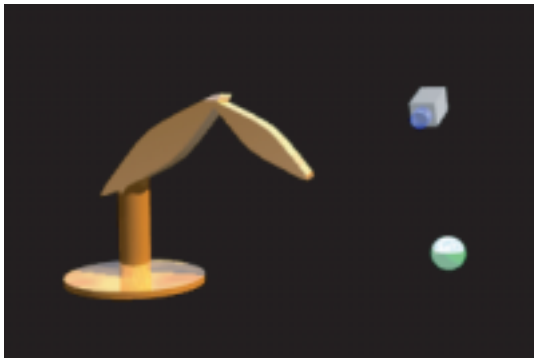


Figure 2.1: Static-eye configuration

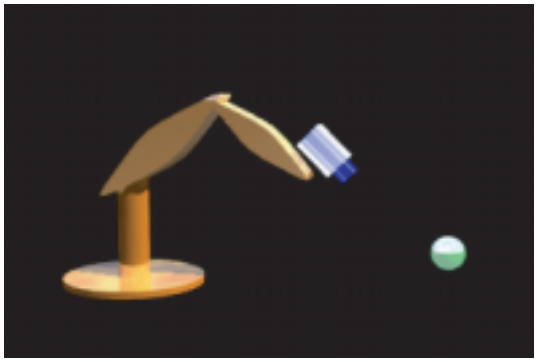


Figure 2.2: Hand-eye configuration

2.2.2 Control level

Indirect visual servoing

When the vision system is used to provide set-point inputs to the joint-level controller, thus making use of joint feedback to internally stabilize the robot, it is referred to as a *indirect visual servoing* system. The indirect visual servoing scheme is shown in Figure 2.3, where $\dot{r}^*$ is the Cartesian space reference velocity (velocity screw: see appendix). It is suitable for slow visual servoing loops ($< 50Hz$). The control low is easier to synthesize.

Direct visual servoing

Direct visual servoing [76] eliminates the robot controller entirely, replacing it with a visual servoing controller that directly computes joint inputs, thus using vision

alone to stabilize the mechanism. The direct visual servoing schema is shown in Figure 2.4, where $\dot{q}^*$ is the reference velocity of the joints. This scheme is suitable for fast visual servoing ($\gg 50Hz$). The control law is more complex since the robot dynamics must be taken into account.

Many visual servoing systems employ "indirect" rather than "direct" visual servoing because of low vision sample rates and high latencies. Using internal feedback loops with a high sampling rate generally presents idealized axis dynamics to the external controller. However, specialized hardware is nowadays developed for faster sampling rates [51]. An important reason is also that most robots already have an interface for Cartesian velocity reference signals or incremental position commands. Thus, the kinematic singularities of the mechanism can be separated from the visual controller, and the system design is greatly simplified in indirect visual servoing.

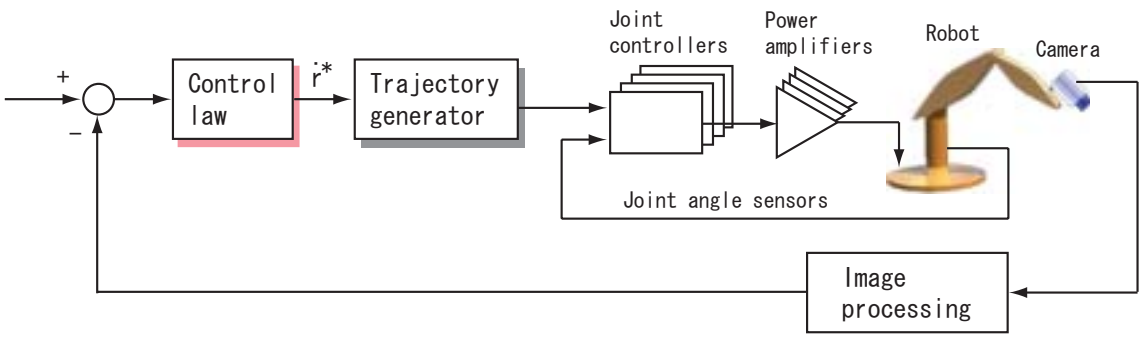


Figure 2.3: Indirect visual servoing

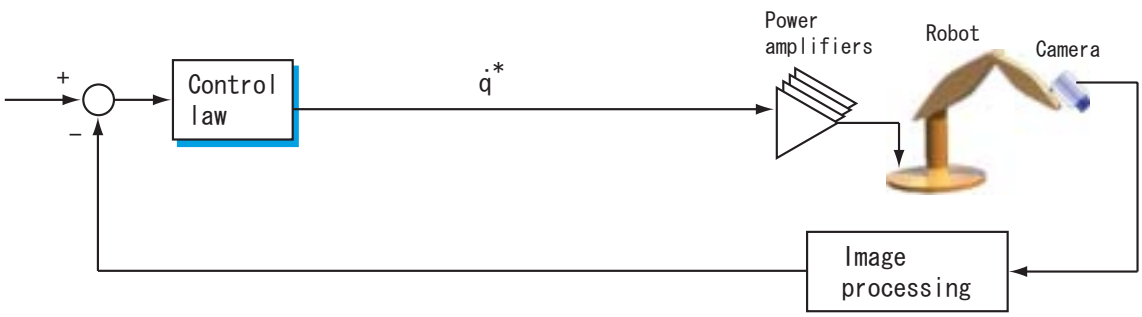


Figure 2.4: Direct visual servoing

2.2.3 Feedback variables

Position-based visual servoing

In position-based control, features are extracted in the image and used in conjunction with a geometric model of the target and the known camera model to estimate the pose of the target with respect to the camera. Using these values, an error between the current and the desired pose of the robot is defined in the Cartesian space. In this way, position-based control nearly separates the control issues, namely the computation of the feedback signal from the estimation problems involved in computing position or pose from visual data. The position-based visual servoing scheme is shown in Figure 2.5. The main inconvenience of this scheme is that calibration errors may induce large pose errors.

Image-based visual servoing

In image-based servoing, control values are directly computed on the basis of image features. The image-based approach may reduce computational delay, eliminate the necessity for image analysis and eliminate errors due to sensor modeling and camera calibration. Image-based visual servoing scheme is shown in Figure 2.6.

Hybrid visual servoing

These exist visual servoing schemes combining position based and image based visual servoing schemes in order to improve the robustness properties disadvantages of both schemes. This is called hybrid visual servoing.

Robots with visual servoing control loops run vision and motion in parallel, continuously updating visual estimates, and correcting the motion on the basis of these estimates. Position-based visual servoing refers to the use of a Cartesian coordinate system to describe the relative position between the end-effector and the target. Once the current pose of the robot is estimated from the images, the objective pose is immediately obtained. Assuming that the robot and camera parameters are well known. When their errors exist large pose errors, the robot may lose the vision of the target. Image-based visual servoing is more robust against errors in the robot parameter's identification. However, it minimizes the image features and does not optimize the motion of the robot in its workspace. Hence workspace limits could be reached. Hybrid visual servoing schemes try to improve the properties of the visual

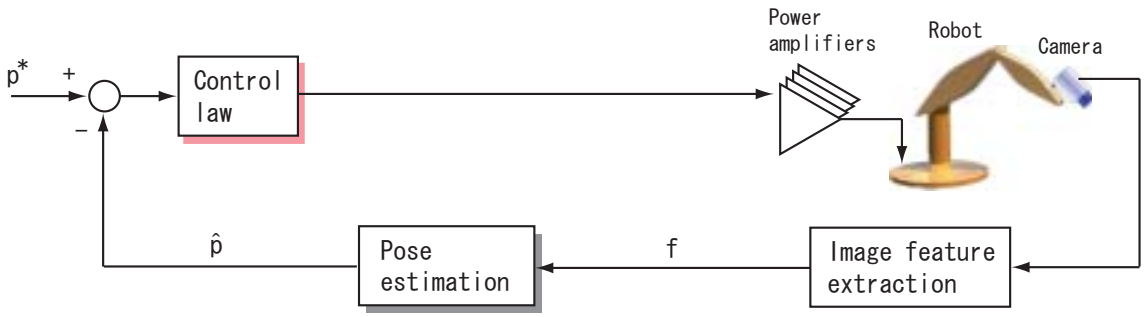


Figure 2.5: Position-based visual servoing

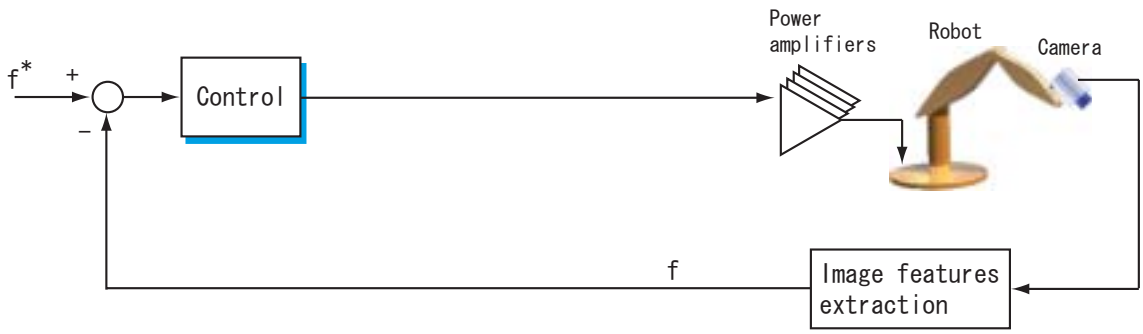


Figure 2.6: Image-based visual servoing

servoing loop in the case of large pose errors or large motions of the robot in its workspace.

For our application when the object to grasp has not necessary a good model and where the camera might be weakly calibrated or uncalibrated, relative pose reconstruction of the object with respect to end-effector is not realistic. Hence, hereafter we will consider only image-based (or hybrid) visual servoing approaches.

2.3 Basic image-based visual servoing control laws

2.3.1 Indirect image-based visual servoing control law

Figure 2.7 shows the indirect image-based visual servoing schema in the case of an eye-in-hand configuration. Images are acquired with the camera on the robot's end-effector, then image features f are extracted. These features are compared with the ones of the objective image, and the control law is selected in order to reduce the image feature errors e_f. The control law is usually selected as (proportional control law):

$$\dot{r}^* = kJ_I^+ e_f \tag{2.1}$$

where, J_I^+ is the pseudo inverse of the *image Jacobian* or interaction matrix J_I (see appendix), which relates the image features motions (e.g., the velocity of a point P on the image plane as explained in the appendix A.2) to the end-effector motion in Cartesian space, $\dot{r}$:

$$\dot{f} = J_I \dot{r} \tag{2.2}$$

In case of a low bandwidth feedback loop, the controlled $\dot{r}$ velocity screw can be considered to be almost the same as the control input $\dot{r}^*$ of the robot's end-effector:

$$\dot{r} \approx \dot{r}^* \tag{2.3}$$

Thus, from Equation 2.7 and Equation 2.2,

$$\dot{f} = kJ_I J_I^+ e_f = kJ_I J_I^+ (f^* - f) \tag{2.4}$$

Mostly, the image Jacobian J_I has to be estimated. Therefore, there is some uncertainty on the estimated Jacobian $\hat{J}_I$. However, Equation 2.4 converges exponentially to zero if:

$$kJ_I \hat{J}_I^+ > 0 \tag{2.5}$$

Espiau *et. al.* [11] proposed an approach to calculate the image Jacobian assuming a simple perspective projection model (see appendix). They use geometric figures such as points, lines, and shapes extracted from the target's contour in the image, and it is widely used today.

In image-based visual servoing, the geometric model of the target is not necessarily needed.

However, the image Jacobian J_I changes according to the camera position and the depth information which is normally unknown. Furthermore, some on-line estimation of the Jacobian might be needed (cf. for example Hosoda and Asada [26]).

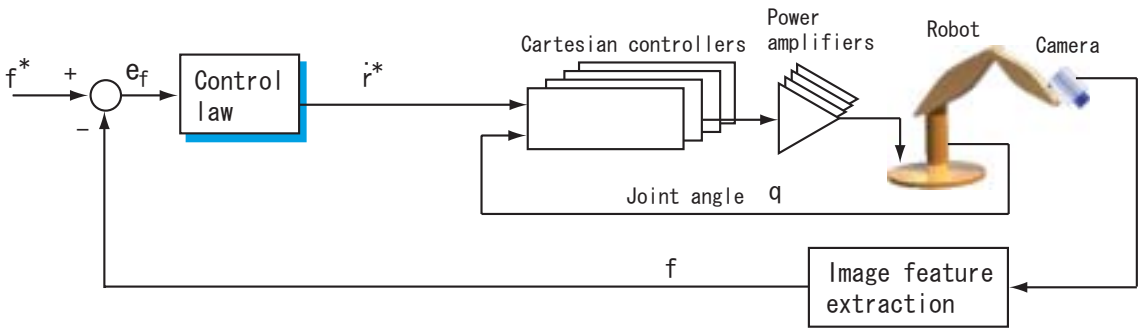


Figure 2.7: Indirect image-based visual servoing

2.3.2 Direct image-based visual servoing control law

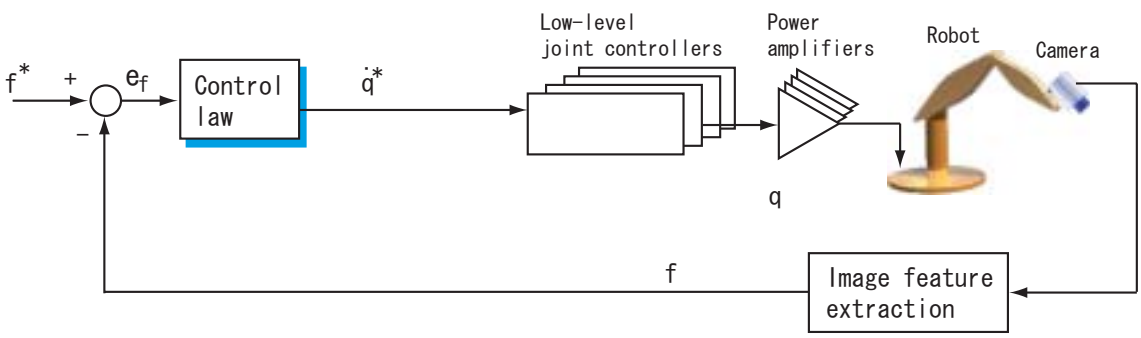


Figure 2.8: Direct image-based visual servoing

Figure 2.8 shows the direct image-based visual servoing scheme. It does not include Cartesian controller, and the visual servoing controller directly computes

joint inputs, thus using vision alone to stabilize the mechanism.

$$\dot{f} = J_I \dot{r} = J_I J_R(q)\dot{q} \tag{2.6}$$

where, J_R is a manipulator Jacobian that relates the joint velocity of the robot to the velocity screw of its end-effector. A stable control law is given by:

$$\dot{q}^* = k J_R^{-1} \hat{J}_I^+ e_f \tag{2.7}$$

Indeed, assuming that the visual servoing feedback loop has a smaller bandwidth , then the velocity controlled robot thus:

$$\dot{q} = \dot{q}^* \tag{2.8}$$

and it follows that:

$$\dot{f} = k J_I J_R J_R^{-1} \hat{J}_I^+ \tag{2.9}$$

which is exponentially stable as long as $k J_I \hat{J}_I^+ > 0$

2.4 Uncalibrated visual servoing

2.4.1 Existing approaches

To control the robot in image-based visual servoing, image Jacobian has to be known, or to be estimated. To compute the image Jacobian (see appendix), an information on the distance between camera and target, a model of the target, and camera intrinsic parameters such as focal length are generally needed. When these information are missing and the Jacobian needs to be estimated, we speak of "uncalibrated visual servoing".

Uncalibrated and model-less visual servoing has been addressed by Hosoda in [26], and Jägersand in [30]. They use an on-line estimation of the Jacobian between the joint velocities of the robot and the feature velocities in the image plane. In [27], Hosoda *et. al.* introduce a limited amount of image geometry constraints in order to do visual path planning with obstacle avoidance. Their approach assumes the on-line identification of a large number of parameters. In the presence of noise, this type of approach may lead to a badly estimated Jacobian matrix if the motions of the robot do not provide enough information in order to identify the parameters.

Another approach to Jacobian estimation is taken by Ritter Martinez and Schulten [68]. They use neural network learning techniques to organize a mapping of 3 DOF Jacobian parameters. They test the approach in simulation and find that about 20 000 test movements are required to learn the mapping (of 4 by 3 Jacobians). This is not a reasonable number of test movements to perform on a real manipulator. The visual-motor model changes typically several times during complex manipulations, for instance when grasping a new object. Since the exact grasp typically is not known it would be difficult to construct a simulation to train the system for the possible real world configurations.

Piepmeier *et al.* present in [65] and [66] a moving target tracking task based on a Newton-like iterative optimization method with an adaptive gain matrix. The Jacobian of the objective function is estimated on-line with a Broyden's update formula (equivalent to a LMS algorithm), or a recursive least-square algorithm with forgetting factor. This approach is adaptive, but cannot guarantee the stability of the visual servoing scheme in presence of large errors in the image. or large numbers of parameters to estimate.

In the next section, the Newton-like methods for uncalibrated and model-less visual servoing are detailed and simulations are carried out. In this method, the kinematic model of the robot is unknown; however, we assume that the joint limits are known. The intrinsic parameters of the camera are also unknown. The main idea is to use an optimization algorithm in real-time to move the robot to the next position that makes an objective function decrease. In these approaches, the objective function to be minimized is a cost function that penalizes the difference between the ideal image in the desired position of the robot and the actual image in the current position. The variables are the joint angles of the robot. The algorithm is based on the Gauss-Newton algorithm and requires the knowledge of the gradient of the objective function. The robot has to make small movements each time to calculate the gradient of the objective function. Large motions of the robot lead to bad estimation of the gradient and the robot may become worse. Further more, this is not best method with local convergence results.

2.4.2 Uncalibrated visual servoing using Newton-like methods

Assuming that m features are detected in the image, let define $f(\Theta)$ as the m-dimensional vector of the feature errors, where Θ is the vector of size n defining the position of the robot (e.g., Θ are the joint angles). Hence, the feature error vector $f(\Theta) = 0$ if $\Theta = \Theta^*$, the desired position of the robot. Therefore, the positioning task using visual servoing is achieved by minimizing the following objective function:

$$f(\Theta) = \begin{pmatrix} f_1(\Theta) \\ \vdots \\ f_m(\Theta) \end{pmatrix} \tag{2.10}$$

$$F(\Theta) = \frac{1}{2} f(\Theta)^\top f(\Theta) \tag{2.11}$$

Indeed, the feature error $f(\Theta^*) = 0$ minimizes $F(\Theta)$.

The gradient of the objective function $F(\Theta)$ is given by:

$$\nabla F(\Theta) = \begin{pmatrix} \frac{\partial f(\Theta)^\top}{\partial \theta_1} \\ \vdots \\ \frac{\partial f(\Theta)^\top}{\partial \theta_n} \end{pmatrix} f(\Theta) = J(\Theta) f(\Theta) \tag{2.12}$$

where, $J(\Theta) = (\nabla f_1(\Theta) \dots \nabla f_m(\Theta))$ is the $n \times m$ Jacobian matrix of the feature errors:

$$J_{ij}(\Theta) = \frac{\partial f_j(\Theta)}{\partial \theta_i} \tag{2.13}$$

The Hessian of the objective function $F(\Theta)$ is given by:

$$H(\Theta) = \nabla \left(\nabla^\top F(\Theta) \right) \tag{2.14}$$

$$= J(\Theta) J(\Theta)^\top + \sum_{i=1}^{m} f_i(\Theta) H_i(\Theta) \tag{2.15}$$

where $H(\Theta)$ is the Hessian of the ith feature error, i.e. $H_i(\Theta) = \nabla \left(\nabla^\top f_i(\Theta) \right)$.

Near the optimum value Θ^*, it can be assumed that $f_i(\Theta) \simeq 0$, therefore

$$H(\Theta) \simeq J(\Theta) J(\Theta)^\top \tag{2.16}$$

If the iterative Newton's algorithm is used for the minimization of $F(\Theta)$, then at each iteration:

$$\Theta_{k+1} = \Theta_k - \alpha_k H(\Theta_k)^{-1} F(\Theta_k) \tag{2.17}$$

Therefore, a Newton-like algorithm is given by:

$$\Theta_{k+1} = \Theta_k - \alpha_k (J_k J_k^\top)^{-1} J_k f(\Theta_k) \tag{2.18}$$

where $0 < \alpha_k$ is the step size in the descent direction and $J_k = J(\Theta_k)$. The Jacobian matrix is estimated on-line. At all time k, a first order approximation of the feature error is given by:

$$f(\Theta_k) \simeq f(\Theta_{k-1}) + J(\Theta_{k-1})^\top (\Theta_k - \Theta_{k-1}) \tag{2.19}$$

Therefore, an on-line estimation algorithm of the Jacobian matrix $J(\Theta^*)$ could be defined by:

$$J_k = J_{k-1} + \frac{\Gamma \Delta\Theta_k (\Delta f_k^\top - \Delta\Theta_k^\top J_{k-1})}{1 + \Delta\Theta_k^\top \Gamma \Delta\Theta_k}, \qquad \Gamma = \varsigma^\top > 0 \tag{2.20}$$

where Γ is constant gain positive matrix, J_k is the kth estimate of $J(\Theta^*)$, and where,

$$\begin{aligned} \Delta f_k &= f_k - f_{k-1} \\ \Delta\Theta_k &= \Theta_k - \Theta_{k-1} \end{aligned}$$

This algorithm is well known in adaptive filtering as the least means square (LMS) algorithm that converges toward $J(\Theta^*)$ if the θ_k are in a neighborhood of Θ^* where J is almost constant. The estimation of the Jacobian matrix will converge toward the true Jacobian matrix, if the feature errors provide enough information, i.e., if the displacements of the robot span locally all the possible directions. Furthermore, since the Jacobian matrix is depending on Θ, convergence will be possible only around a working position, i.e., assuming small displacements around its working position. Finally, it should be pointed out that the Newton optimization algorithm has local properties of convergence, i.e., it will converge toward the nearest local minimum of $F(\Theta)$.

Note that a recursive least squares estimation algorithm with modified forgetting factor can also be used to provide on-line estimates of the Jacobian $J(\Theta^*)$ (cf. [65] [41]).

$$J_k = J_{k-1} + \frac{\alpha P_{k-1} \Delta\Theta_k (\Delta f_k^\top - \Delta\Theta_k^\top J_{k-1})}{1 + \Delta\Theta_k^\top P_{k-1} \Delta\Theta_k} \tag{2.21}$$

$$P_k = \frac{1}{\lambda} \left(P_{k-1} - \frac{P_{k-1} \Delta\Theta_k \Delta\Theta_k^\top P_{k-1}}{1 + \Delta\Theta_k^\top P_{k-1} \Delta\Theta_k} \right) + \delta I - \beta P_{k-1} P_{k-1} \tag{2.22}$$

where λ is a forgetting factor and the extra terms multiplied by δ and β are stabilizing terms in case there is not enough excitation. Typical values for $\lambda, \alpha, \beta, \delta$ are:

$$\lambda \in [0.95\ 0.99]$$
$$\alpha \in [0.1\ 0.5]$$
$$0 < \beta \leq 0.01$$
$$0 < \delta \leq 0.01$$

2.4.3 Simulations

Simulation with 2DOF for a centering task

Simulations were carried out using the previous Gauss-Newton method. The Jacobian was estimated using a recursive least squares estimation with modified forgetting factor. The positioning processes was implemented under the graphic library OpenGL to model the robot SCEMI 6P01. Target is a cylindrical white cup of 11cm height without handle on a black background. Figure 2.9 and Figure 2.10 shows the starting image and goal image of the target.

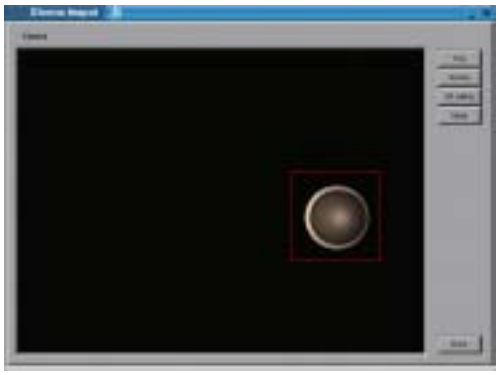


Figure 2.9: Image at the starting pose

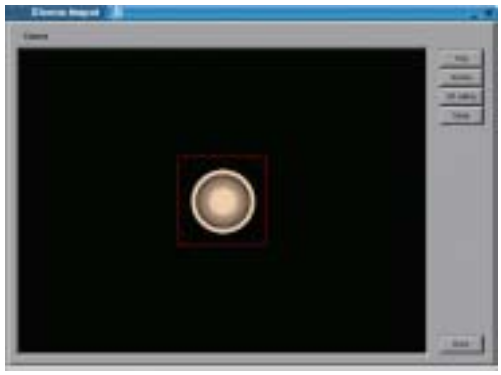


Figure 2.10: Image at the goal pose

The task consists in bringing the target object to the center of the image. The objective function was set as following:

$$F(\Theta) = \left(x - \frac{W}{2}\right)^2 + \left(y - \frac{H}{2}\right)^2 \tag{2.23}$$

$$f(\Theta) = \begin{pmatrix} f_1(\Theta) \\ f_2(\Theta) \end{pmatrix} \tag{2.24}$$

with

$$f_1 = \left(x - \frac{W}{2}\right) \ , \quad f_2 = \left(y - \frac{H}{2}\right) \tag{2.25}$$

where W and H are the width and the height of the image, x and y are the coordinates of the center of the object in the image. The number of image features to be optimized is $(m = 2)$.

In the first environment, only two degrees of freedom of the end-effector were controlled by the optimization process. Thus, $(m = n)$. Figure 2.13 shows that the optimization converges very well. Number of iterations is four, for an objective function with find value $F(\Theta) = 0.000289$ at convergence.

Simulation with 6DOF for a centering task

Next, six degrees of freedom of the manipulator are controlled by the optimization process. The same centering task was defined. The number of the system configurations to be optimized is larger than the objective function $(m < n)$ and the algorithm did not converge as shown in Figure 2.14.

Simulation with 6DOF

Six degrees of freedom of the manipulator were controlled by the optimization process. Figure 2.11 and Figure 2.12 shows the target's image at the starting pose and the goal pose. The number of image features to be minimized is six $(m = 6)$: x and y coordinates of the barycenter of the four dots, the distances between the barycenter and each dot. The process did not converge as shown in Figure 2.15.

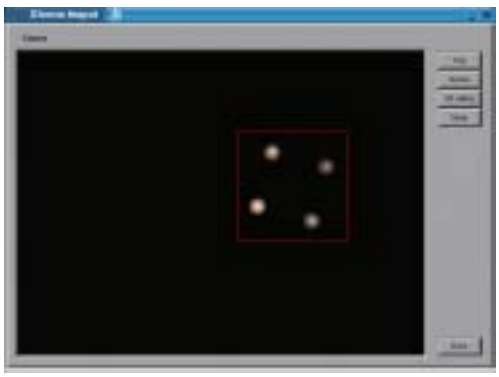


Figure 2.11: Image at the starting pose

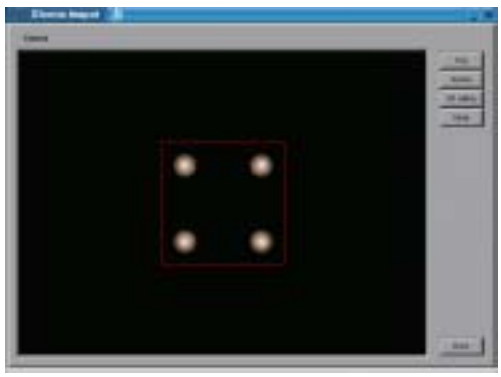
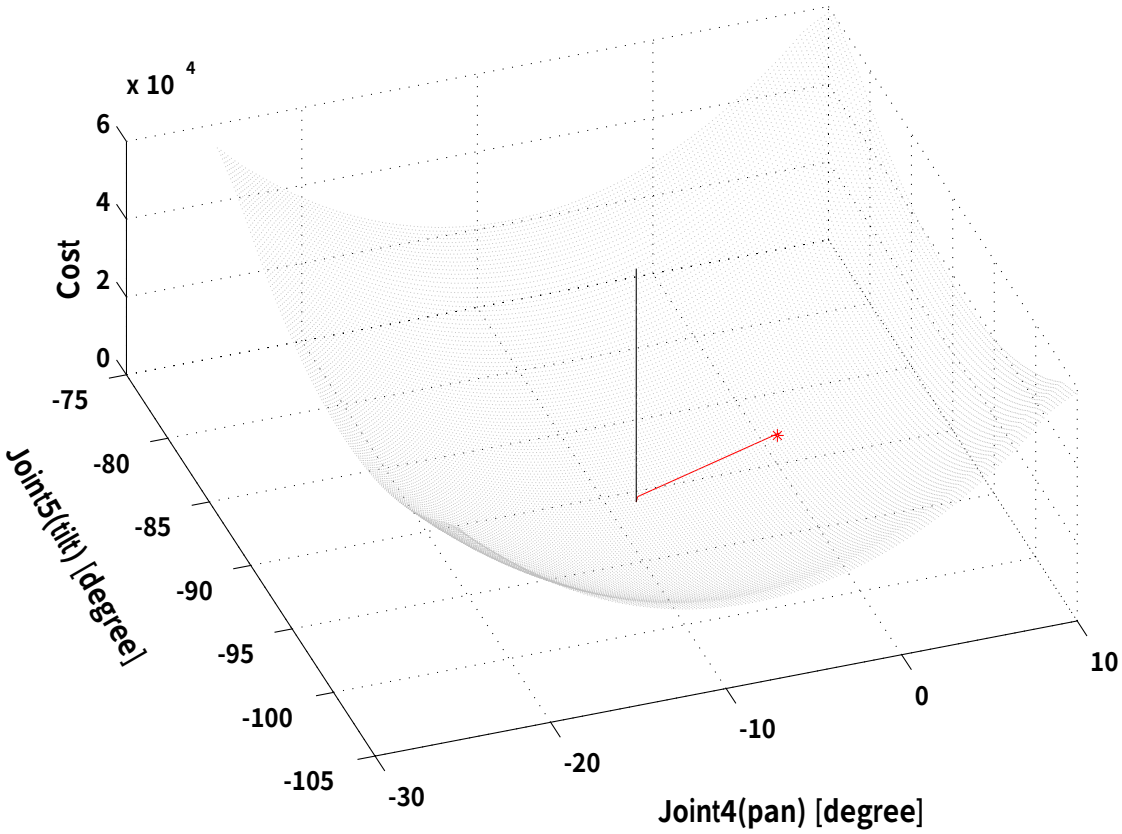
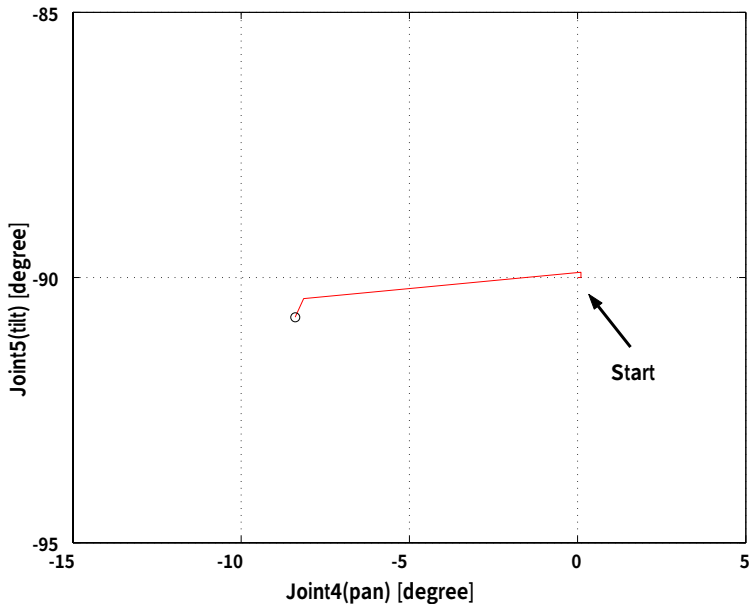


Figure 2.12: Image at the goal pose

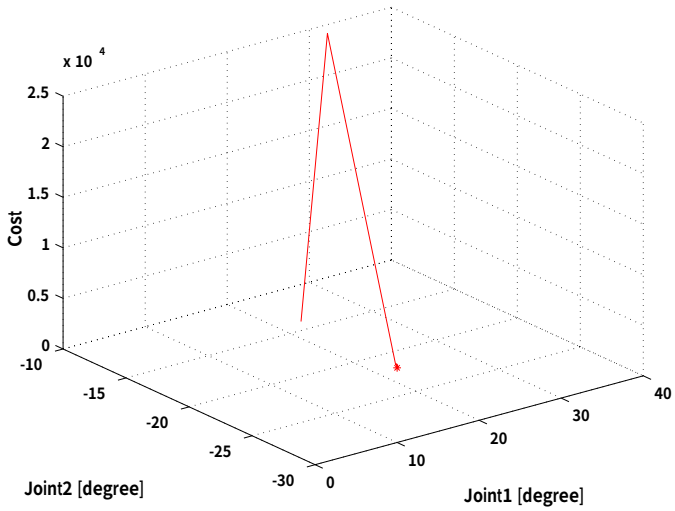


(a)Joint angles and their cost functions

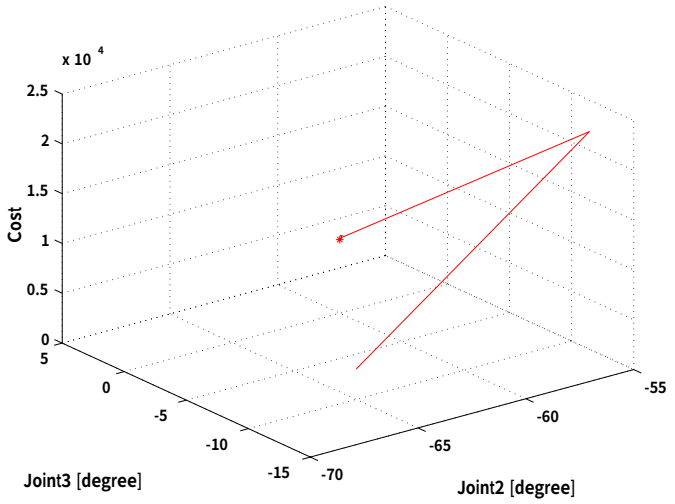


(b)Close-up of the trajectory of joint angles

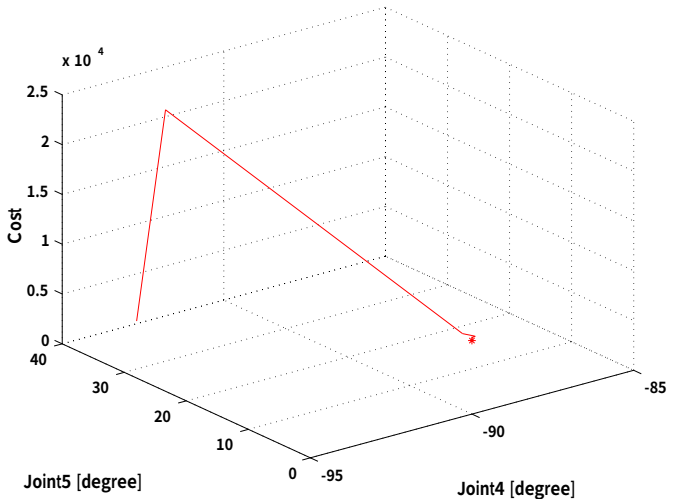
Figure 2.13: Result of adaptive Gauss-Newton optimization with 2DOF centering task



(a)Joint 1, 2, and Cost function

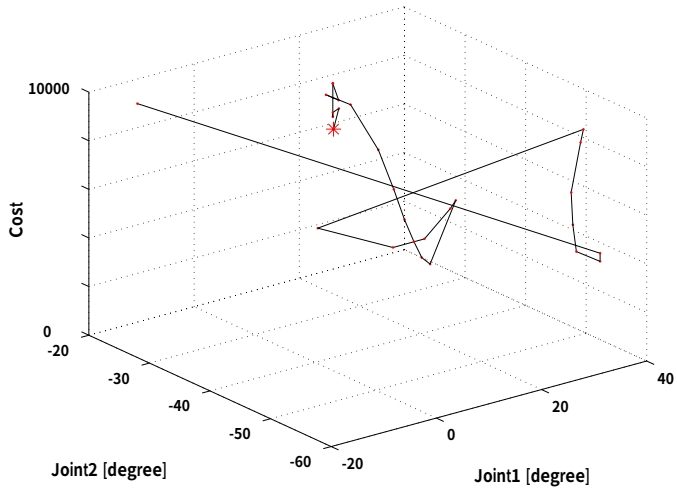


(b)Joint 3, 4, and Cost function

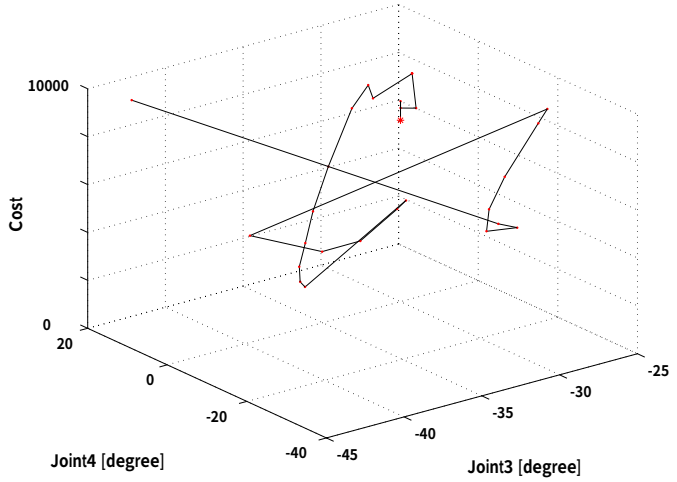


(c)Joint 5, 6, and Cost function

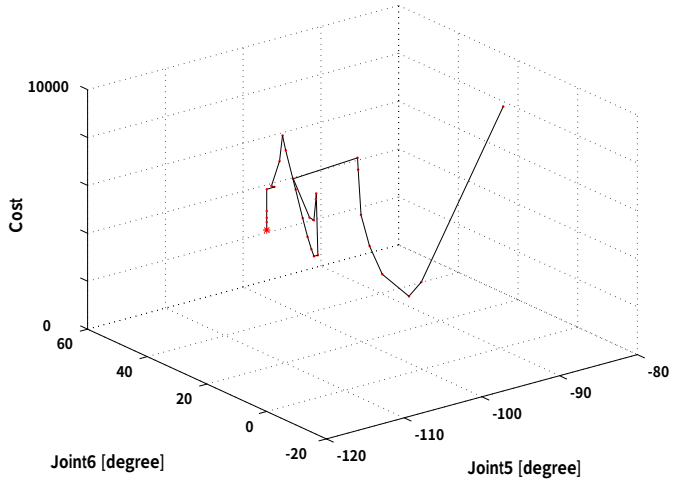
Figure 2.14: Result of adaptive Gauss-Newton optimization with 6DOF centering task



(a)Joint 1, 2, and Cost function



(b)Joint 3, 4, and Cost function



(c)Joint 5, 6, and Cost function

Figure 2.15: Result of adaptive Gauss-Newton optimization with 6DOF

Chapter 3

Modified simplex method

3.1 Introduction

Most of the previous works on visual servoing assume that the kinematic model of the robot and the camera intrinsic parameters are known. Most of these methods could work with weak calibration, but they would fail if the robot and the vision system were fully unknown.

Uncalibrated and model-less visual servoing has been addressed by Hosoda *et al.* in [26], and Jägersand in [30]. They use an on-line estimation of the Jacobian between the joint velocities of the robot and the feature velocities in the image plane. Their approach assumes the on-line identification of a large number of parameters. In the presence of noise, this type of approach may lead to a badly estimated Jacobian matrix if the motions of the robot do not guarantee sufficient excitation for the identification of the parameters. Piepmeier *et al.* present in [65] and [66] a moving target tracking task based on the Gauss-Newton optimization method. The Jacobian of the objective function is estimated on-line with a Broyden's update formula (equivalent to a LMS or gradient algorithm) in order to compute the control signal. This approach is adaptive, but cannot guarantee the stability of the visual servoing scheme in presence of large errors in the image, or if the motions of the robot do not guarantee sufficient excitation.

In this section, a novel approach for uncalibrated and model-less visual servoing using a modified *simplex* iterative search method is proposed. The kinematic model of the robot is not necessarily known; however, we assume that the bounds of the joint variables are known. The intrinsic parameters of the camera are also unknown.

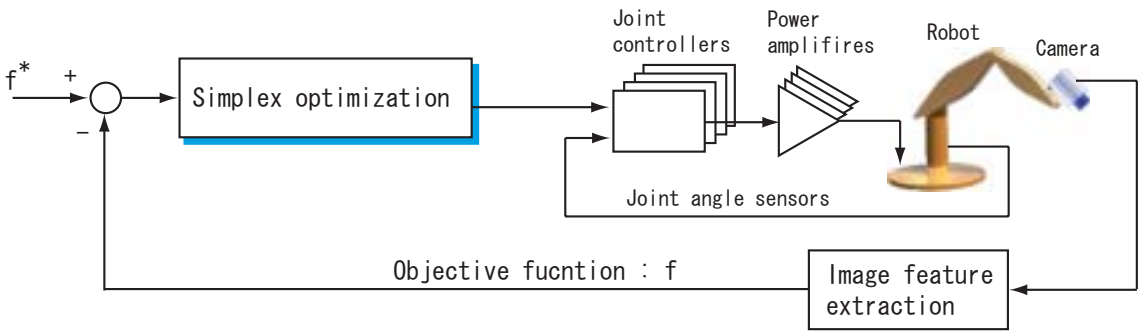


Figure 3.1: Block diagram with Nelder-Mead simplex method

The simplex method is an unconstrained optimization technique [67], that do not use gradient computations. Note that it is different from the linear programming technique method also called "simplex". This method was originally proposed by Spendley, Hext, and Himsworth [73] and was developed later by Nelder and Mead [64]. Box [2] extended the simplex method (Nelder and Mead algorithm) to solve constrained minimization problems, and this technique is called *complex method*.

The basic idea behind the simplex and complex methods is to compare the value of the objective function in several configurations and to move to the next position in order to decrease this value. For our problem, the objective function to be minimized is a cost function that penalizes the distance between the desired configuration of the robot and the current configuration. The explored configurations are usually defined parametrized by the joint angles of the robot. Note that other variables could be used as it will be shown in the next chapter. Block diagram of this method is illustrated in Figure 3.1

The main idea is to use the optimization algorithm in real-time to move the robot to the next pose that makes the cost function decrease. Our approach may include cost functions that are not differentiable since it does not require the computation of Jacobian matrices.

With our approach, the same control algorithm can be used, without any modification, to control any robot that belongs to the same class (*e.g.*, a 6 DOF anthropomorphic arm) fitted with the same type of vision system (*e.g.*, a black & white monocular system) and with the same type of target (*e.g.*, a cylindrical object).

The joint limits give constraints on the variables. The complex method can deal with these limits in order to avoid them. Furthermore, the method that we propose guarantees that the target will not be lost during the iterative search which is a well

known problem in visual servoing (see *e.g.*, E. Malis [39]).

The optimization algorithm provides admissible joint positions that make the cost function decrease. These joint positions are fed to a robot with stable internal joint-level position control loops and trajectory planning. Therefore, the approach guarantees stability and convergence.

The remaining of this chapter is structured as follows. Firstly, the fundamentals of the Nelder and Mead simplex method to minimize objective functions are presented in section 2. Then, the simplex method modified to handle the constrained problem is presented in section 3. In section 4, we describe the application of this optimization method to visual servoing. In the last section, simulations and experimental results are presented.

3.2 Fundamentals of Nelder and Mead simplex method

This section presents Nelder and Mead simplex method for solving the following n-dimensional unconstrained minimization problem:

$$\text{Find } \boldsymbol{X} = \{x_1, x_2, \ldots, x_n\} \text{ which minimizes } f(\boldsymbol{X}) \tag{3.1}$$

3.2.1 Simplex iterative process

The geometric figure formed by the convex hull of a set of $n+1$ points in an n-dimensional space is called a simplex. For example, in two dimensions, the simplex is a triangle, and in three dimensions, it is a tetrahedron.

The basic idea in the simplex method is to compare the value of the objective function f at the $n+1$ vertices $\{\boldsymbol{X}_i\}$ of a simplex and move the simplex gradually toward the optimum point during the iterative process. Starting from an initial simplex with $n+1$ known vertices $\{\boldsymbol{X}_i\}$, a new vertex will be computed to define a new simplex using reflection, expansion, or contraction.

Reflection

If $\boldsymbol{X}_h$ is the vertex corresponding to the highest value of the objective function among the vertices of a simplex, we can expect the point $\boldsymbol{X}_r$ obtained by reflecting

the point $\boldsymbol{X}_h$ in the opposite face of the simplex to have a smaller value. Mathematically, the reflected point $\boldsymbol{X}_r$ is given by:

$$\boldsymbol{X}_r = \alpha(\boldsymbol{X}_{os} - \boldsymbol{X}_h) + \boldsymbol{X}_{os} \tag{3.2}$$

where $\boldsymbol{X}_{os}$ is the centroid of all the points $\boldsymbol{X}_i$ except $\boldsymbol{X}_h$:

$$\boldsymbol{X}_{os} = \frac{1}{n} \sum_{i=1, i \neq h}^{n+1} \boldsymbol{X}_i \tag{3.3}$$

and α is the reflection coefficient ($\alpha > 0$), defined as

$$\alpha = \frac{\|\boldsymbol{X}_r - \boldsymbol{X}_{os}\|}{\|\boldsymbol{X}_h - \boldsymbol{X}_{os}\|} \tag{3.4}$$

where $\| \cdot \|$ denotes the Euclidian norm.

The reflected point $\boldsymbol{X}_r$ is then compared with the vertices $\boldsymbol{X}_l$ and $\boldsymbol{X}_s$ corresponding to the minimum and second maximum function value respectively. The next step depends on the value of $f(\boldsymbol{X}_r)$. The reflection process is illustrated in Figure 3.2.

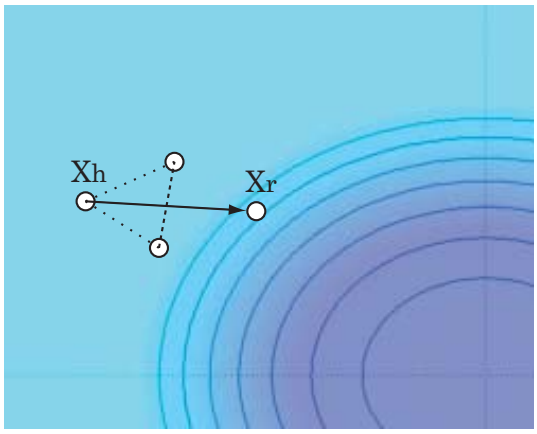


Figure 3.2: Reflection process of the simplex iteration

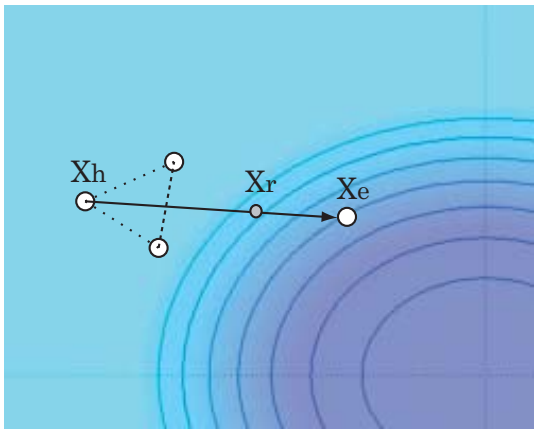


Figure 3.3: Expansion process of the simplex iteration

Expansion

In the case $f(\boldsymbol{X}_r) < f(\boldsymbol{X}_l)$ the minimum value of f, one can generally expect to see the function value decrease further by expanding $\boldsymbol{X}_r$ to $\boldsymbol{X}_e$ in the same direction using the relation

$$\boldsymbol{X}_e = \gamma(\boldsymbol{X}_r - \boldsymbol{X}_{os}) + \boldsymbol{X}_{os} \tag{3.5}$$

where γ is called the expansion coefficient ($\gamma > 1$), and is defined as

$$\gamma = \frac{\|\boldsymbol{X}_e - \boldsymbol{X}_{os}\|}{\|\boldsymbol{X}_r - \boldsymbol{X}_{os}\|} \tag{3.6}$$

Then, we replace the point $\boldsymbol{X}_h$ by $\boldsymbol{X}_e$ or $\boldsymbol{X}_r$ depending on the value of $f(\boldsymbol{X}_e)$, i.e.: $\boldsymbol{X}_h$ is replaced by $\boldsymbol{X}_e$ if $f(\boldsymbol{X}_e) < f(\boldsymbol{X}_r)$, or $\boldsymbol{X}_h$ is replaced by $\boldsymbol{X}_r$ otherwise. The expansion process is represented in Figure 3.3.

Contraction

If the reflection process gives a point $\boldsymbol{X}_r$ for which $f(\boldsymbol{X}_r) > f(\boldsymbol{X}_s)$ the second worst value of f, then obviously at the next iteration $\boldsymbol{X}_r$ would become the worst vertex, therefore we should contract the point $\boldsymbol{X}_r$ to $\boldsymbol{X}_c$ closer to the centroid of the vertex:

$$\boldsymbol{X}_c = \beta(\boldsymbol{X}_r - \boldsymbol{X}_{os}) + \boldsymbol{X}_{os} \tag{3.7}$$

where β is called the contraction coefficient ($0 \leq \beta \leq 1$) and is defined as:

$$\beta = \frac{\|\boldsymbol{X}_c - \boldsymbol{X}_{os}\|}{\|\boldsymbol{X}_r - \boldsymbol{X}_{os}\|} \tag{3.8}$$

If the contraction process produces a point $\boldsymbol{X}_c$ for which $f(\boldsymbol{X}_c) < f(\boldsymbol{X}_h)$, we replace the point $\boldsymbol{X}_h$ by $\boldsymbol{X}_c$ and proceed to the next iteration. This contraction process is illustrated at Figure 3.4

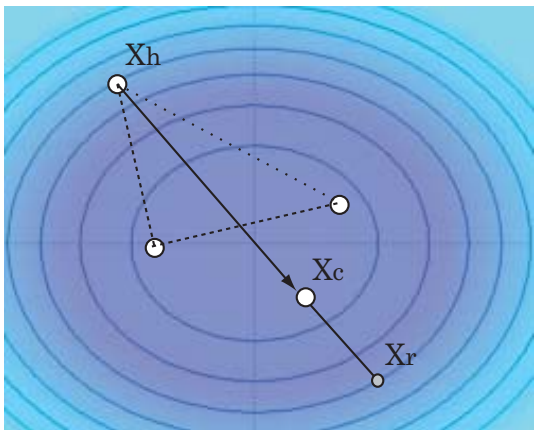


Figure 3.4: Contraction process of the simplex iteration

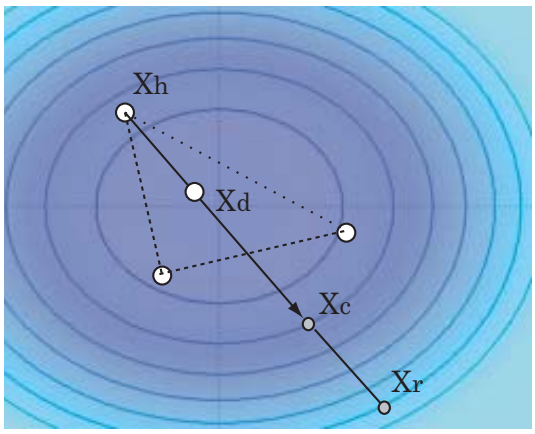


Figure 3.5: Reduction process of the simplex iteration

Reduction

However, if $f(\boldsymbol{X}_c) > f(\boldsymbol{X}_h)$ in the contraction process, then the contraction process will be a failure. In this case, we apply a reduction process to the simplex:

$$\boldsymbol{X}_d = \beta(\boldsymbol{X}_h - \boldsymbol{X}_{os}) + \boldsymbol{X}_{os} \tag{3.9}$$

before going to the next iteration with the point $\boldsymbol{X}_h$ replaced by $\boldsymbol{X}_d$. This reduction process is illustrated in Figure 3.5.

Convergence

The method is assumed to reach convergence whenever some stopping criteria have been met, e.g.:

$$\sqrt{\sum_{i=1}^{n+1} \frac{\{f(\boldsymbol{X}_i) - \bar{f}\}^2}{n+1}} \leq \varepsilon_1, \quad with \;\; \bar{f} \;=\; \sum_{i=1}^{n+1} \frac{f(\boldsymbol{X}_i)}{n+1} \tag{3.10}$$

$$\sqrt{\sum_{i=1}^{n+1} \frac{\{\boldsymbol{X}_i - \overline{\boldsymbol{X}}\}^2}{n+1}} \leq \varepsilon_2, \quad with \;\; \overline{\boldsymbol{X}} \;=\; \sum_{i=1}^{n+1} \frac{\boldsymbol{X}_i}{n+1} \tag{3.11}$$

The flow chart of the algorithm is shown at Figure 3.6.

The parameter α, β, γ are somewhat arbitrary but typical values are:

$$\alpha = 1 \tag{3.12}$$

$$\gamma = \frac{1}{2} \tag{3.13}$$

$$\beta = 2 \tag{3.14}$$

3.2.2 Constrained Problems

Complex Method

Any manipulator has a limited workspace and joint constraints. To generate the vertices of a simplex for a such manipulator, we propose the complex method, which is an extension of the simplex method to solve constrained minimization problems,

$$\text{Find } \boldsymbol{X} = \{x_1, x_2, \ldots, x_n\} \text{ which minimizes } f(\boldsymbol{X})$$

subject to:

$$g_v(\boldsymbol{X}) \leq 0, \quad v = 1, 2, \ldots, m \tag{3.15}$$

$$x_j^{(L)} \leq x_j \leq x_j^{(U)}, \quad j = 1, 2, \ldots, n \tag{3.16}$$

where, $x_j^{(L)}$ and $x_j^{(U)}$ are respectively the lower and upper bounds on the variables x_j. In general, the satisfaction of the side constraints given by Equation (3.16) may not correspond to the satisfaction of the constraints $g_v(\boldsymbol{X}) \leq 0$.

Initialization of the simplex

The method assumes that an initial feasible point $\boldsymbol{X}_1$ (i.e. such that Equation (3.15) and Equation (3.16) are verified) is available. This feasible point $\boldsymbol{X}_1$ will be the first vertex of the initial simplex. Thus, the remaining n points are computed as following:

$$x_{i,j} = x_j^{(L)} + r_{i,j}(x_j^{(U)} - x_j^{(L)}) \tag{3.17}$$
$$i = 2, 3, \ldots, n+1 \ , \quad j = 1, 2, \ldots, n$$

where $x_{i,j}$ is the jth component of the point $\boldsymbol{X}_i$, and $r_{i,j}$ is a random number lying in the interval $(0, 1)$. Note that the points generated according to Equation (3.17) satisfy the side constraints, Equation (3.16), but may not satisfy the constraints given by Equation (3.15).

As soon as a new point $\boldsymbol{X}_i$ is generated ($i = 2, 3, \ldots, k$), we find whether it satisfies all the constraints of Equation (3.15). If $\boldsymbol{X}_i$ violates any of them, the trial point $\boldsymbol{X}_i$ is moved toward the centroid of the remaining, already accepted points $(\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_{i-1})$, until it becomes feasible, that is:

$$\begin{aligned} (\boldsymbol{X}_i)_{new} &= \eta \boldsymbol{X}_{oi} + (1-\eta)\boldsymbol{X}_i \ \text{ with } \ 0 < \eta < 1 \\ \boldsymbol{X}_{oi} &= \frac{1}{i-1}\sum_{j=1}^{i-1} \boldsymbol{X}_j \end{aligned} \tag{3.18}$$

Iterative process

In order to take into the constraints, the simplex method must be modified. Firstly, an initial simplex must be defined in the feasible region (i.e., where the constraints are verified). Then, the reflection, expansion, contraction, and reduction procedures must be modified in order to stay in the feasible region. The same process can be applied. If the worst position $\boldsymbol{X}_h$ violates any of Equation (3.16) and Equation (3.15), the trial point $\boldsymbol{X}_h$ is moved toward the centroid of the remaining using Equation (3.18)

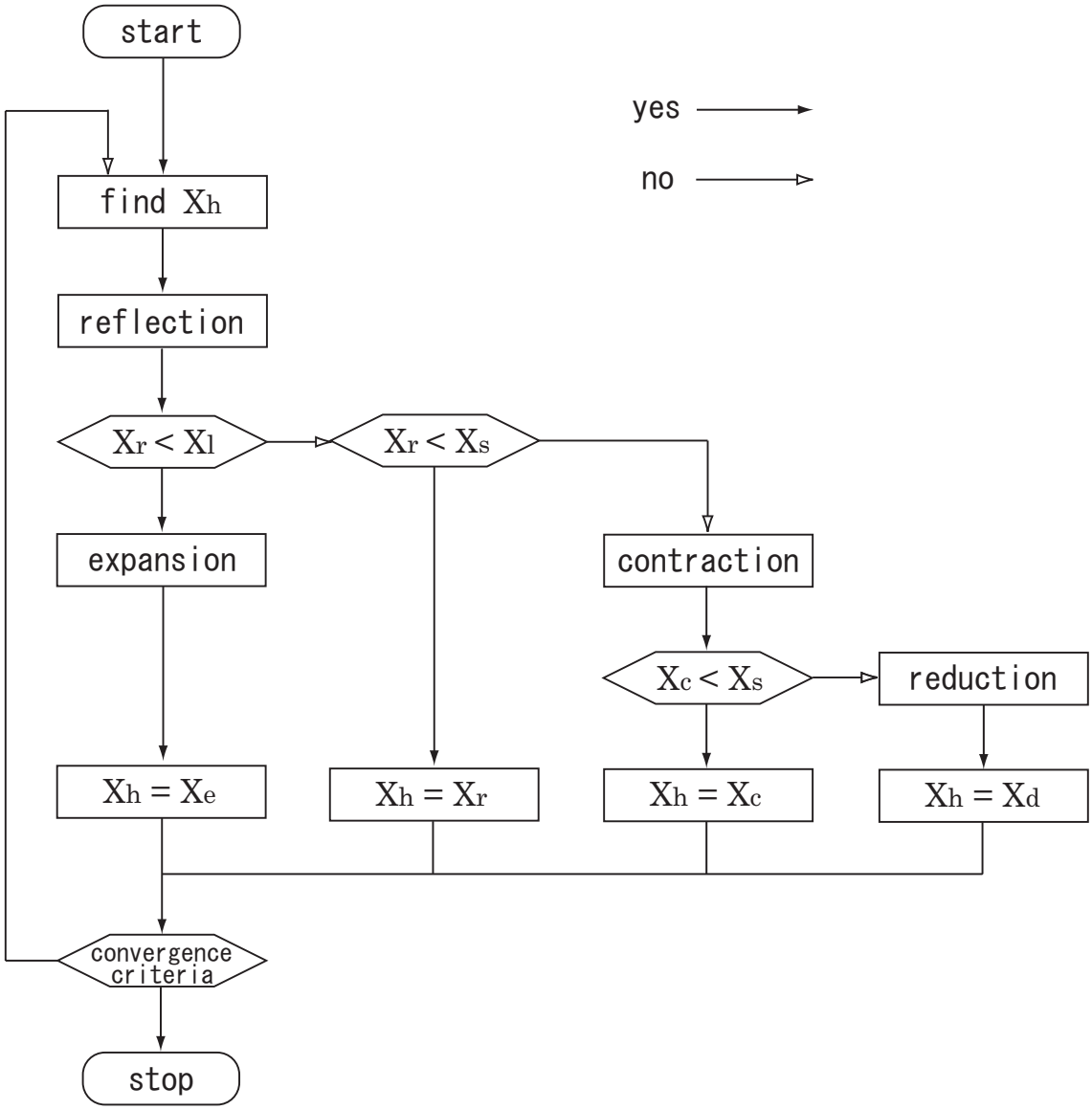


Figure 3.6: Flow chart of Nelder-Mead simplex method

3.3 Uncalibrated visual servoing task using the simplex method

This section deals with the modification of the optimization techniques described above, for application to visual servoing. The goal is to position the end-effector of the robot with respect to a static object. Our approach is generic and can be used with different type of visual servoing applications where the visual features used to build the objective function do not need to be differentiable or to have analytical expressions. After, we will illustrate our approach with a simple positioning task with respect to an unknown object for a robot in an eye-in-hand configuration.

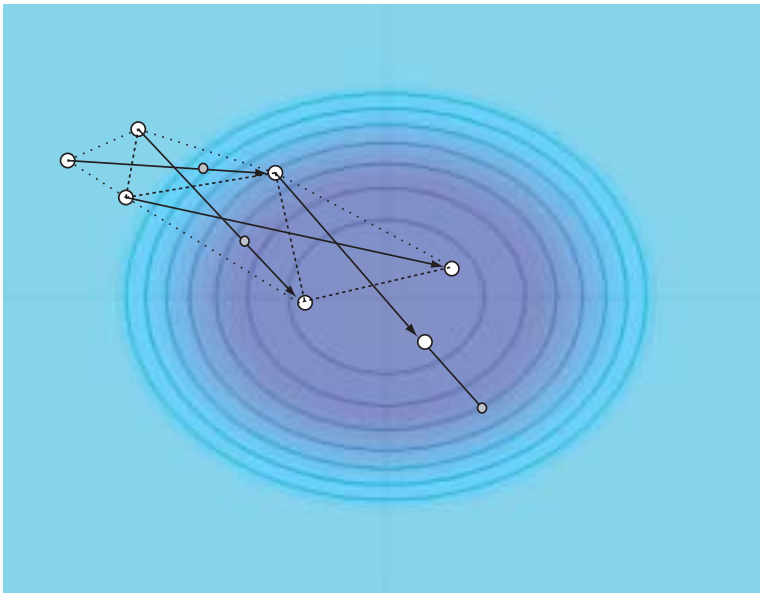
3.3.1 Simplex optimization process with a robot

The main idea is to use a simplex like optimization algorithm to move the robot from an initial position to a goal position. Since the vision system acquires images continuously and assuming that joint angles are also measured, the cost function can be computed along the trajectory of the robot while it is moving from a vertex to its reflection point. Therefore, the optimum could be selected along that vertex.

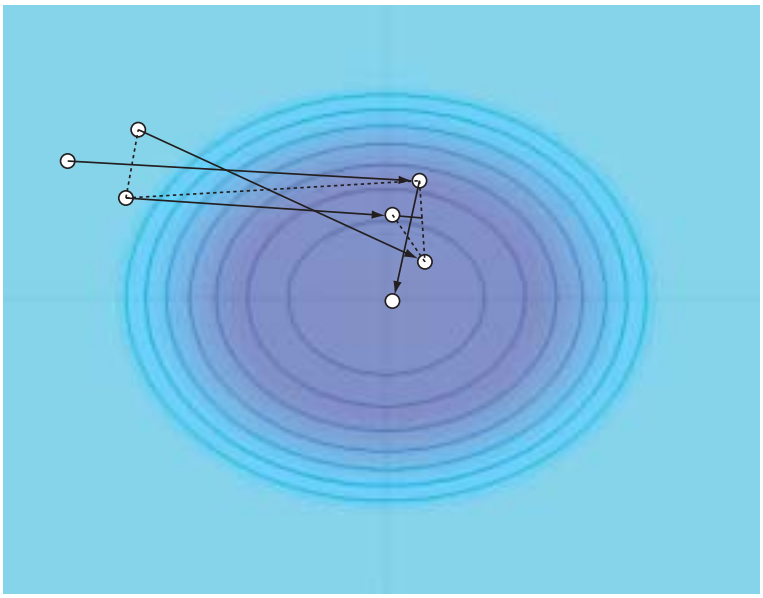
In real-time, in other words, this is equivalent to modifying the Nelder and Mead simplex method by adding a line search procedure during the reflection, expansion, or reduction process. While the manipulator moves from $\boldsymbol{X}_h$ to $\boldsymbol{X}_r$, images are acquired with the camera. Here $\boldsymbol{X}$ are coordinates that gives the position of the robot (e.g., joint angles, or Cartesian coordinates of the end-effector). Hence the robot can scan the segment $\boldsymbol{X}_h - \boldsymbol{X}_r$ in order to find the best point $\boldsymbol{X}_m$ corresponding the minimum value of f on this segment. Once $\boldsymbol{X}_m$ is found on the segment, $\boldsymbol{X}_h$ is replaced by $\boldsymbol{X}_m$ and we start a new simplex again. In the case where $\boldsymbol{X}_h$ corresponds to the minimum value on the segment, i.e. there are no smaller value, the reduction process different from Equation (3.9) is executed as follows:

$$\text{new}\boldsymbol{X}_i = \frac{1}{2}(\boldsymbol{X}_i + \boldsymbol{X}_l) \quad i = 1, 2, \ldots, n+1 \tag{3.19}$$

With this technique, the contraction process is omitted and faster convergence is expected.



(a)Classical simplex



(b)Modified simplex

Figure 3.7: Comparison between classical simplex optimization and modified simplex process with a robot

3.4 Simulation results

3.4.1 Simulations to compare modified method with original (Nelder and Mead) simplex method

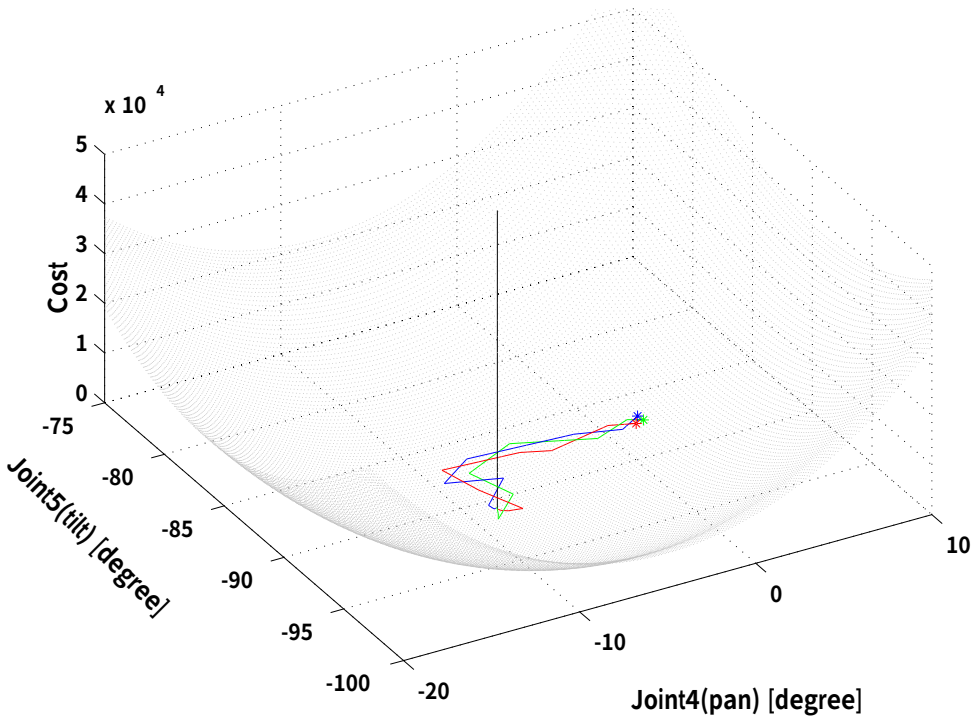
Simulations were carried out using the simplex and the modified simplex method presented in section 3.3.1. The task was the centering of the target object in the image in eye-in-hand visual servoing configuration, same as in section 2.4.3. The system configuration and objective function were also the same.

Simulation without redundancy

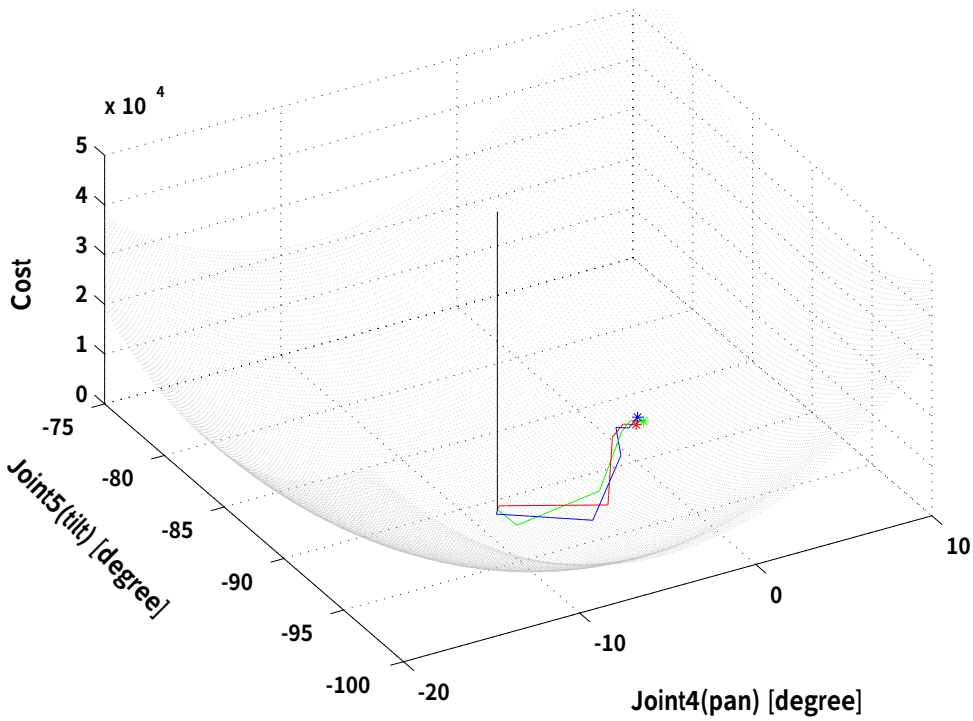
Two degrees of freedom of the end of manipulator were controlled by the optimization process. Thus the task has no redundancy ($m = n$). Figure 3.8 shows the trajectories of simplex on the surface of the objective function, and Figure 3.9 is close-up figure on the joint angles. The trajectories of classical and modified simplex seem to be same, but the main change is the number of iterations. Table 3.1 compares number of iterations and cost of the objective function at the position of convergence.

Simulation with redundant system configurations

Six degrees of freedom of the end of manipulator were controlled by the optimization process. The task with the redundant system configuration ($m < n$) was not able to converge as shown in Figure 3.11 and Figure 3.10. Table 3.1 also makes a comparison with the Gauss-Newton optimization. It is clear that the simplex methods are much efficient than Gauss-Newton for redundant problems.

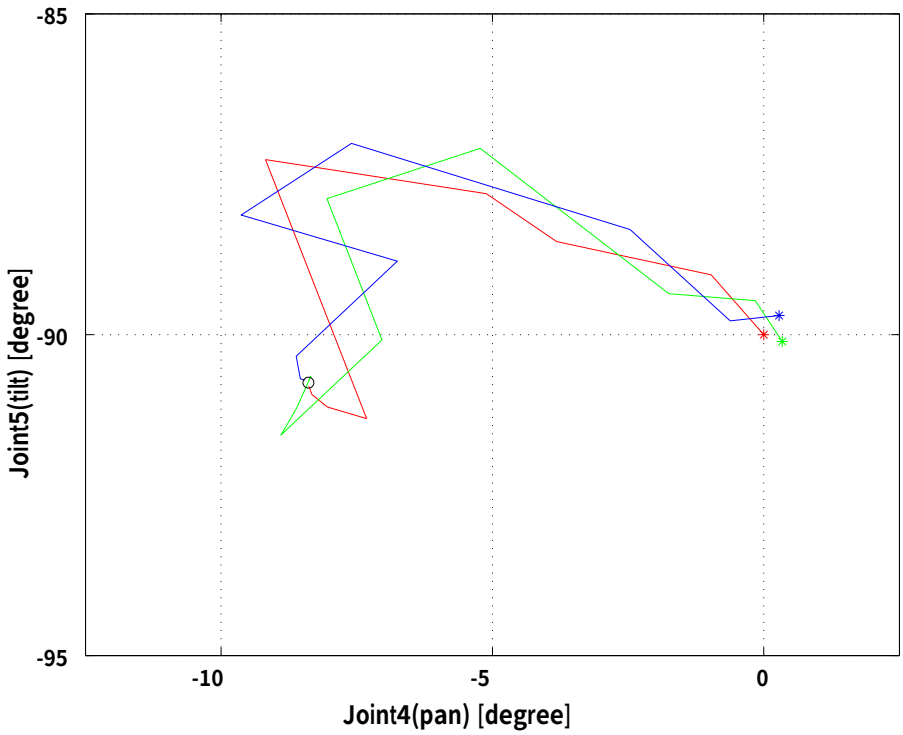


(a)Classical simplex method

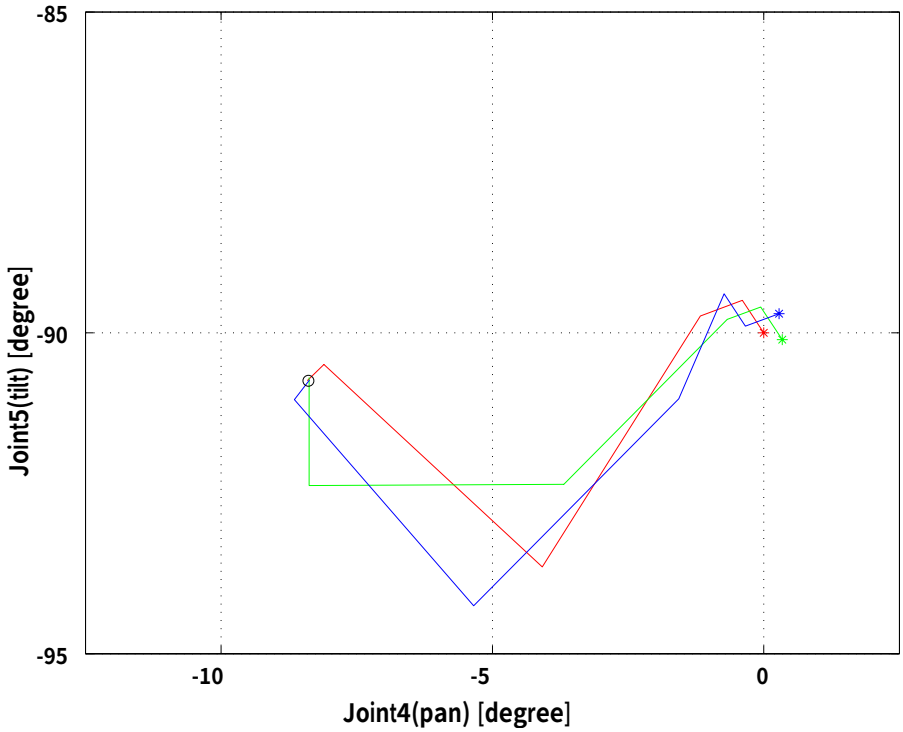


(b)Modified simplex method

Figure 3.8: Joint angles and their cost functions with 2DOF

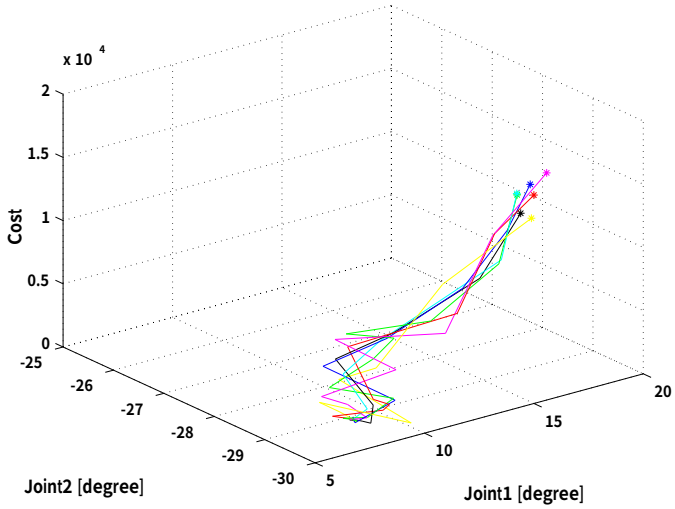


(a)Classical simplex method

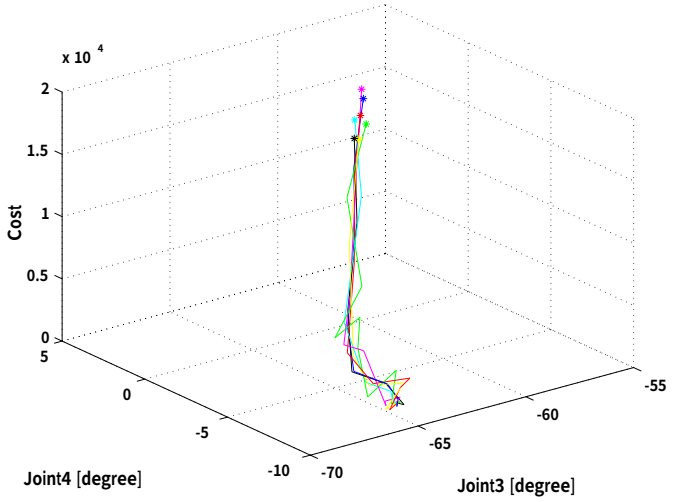


(b)Modified simplex method

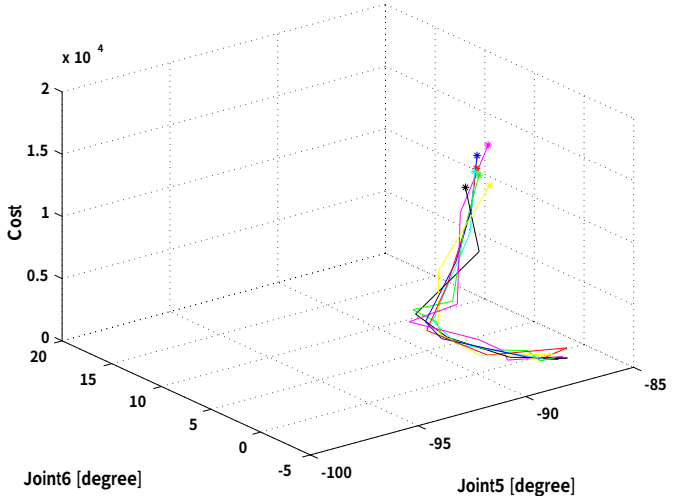
Figure 3.9: Trajectory of joint angles with 2DOF



(a)Joint 1, 2, and Cost function

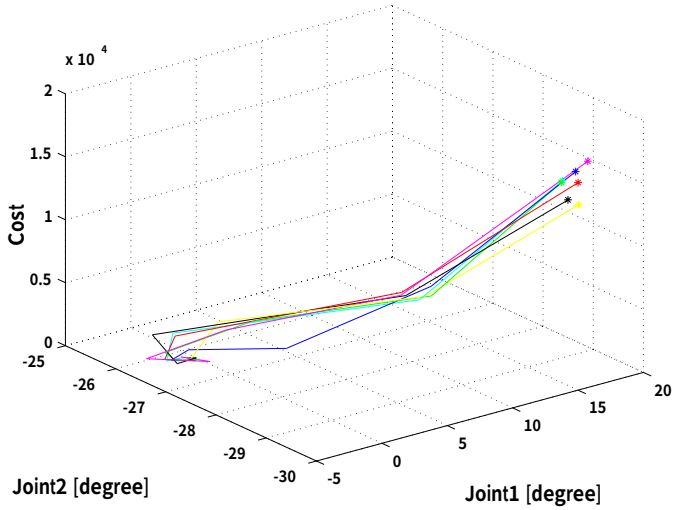


(b)Joint 3, 4, and Cost function

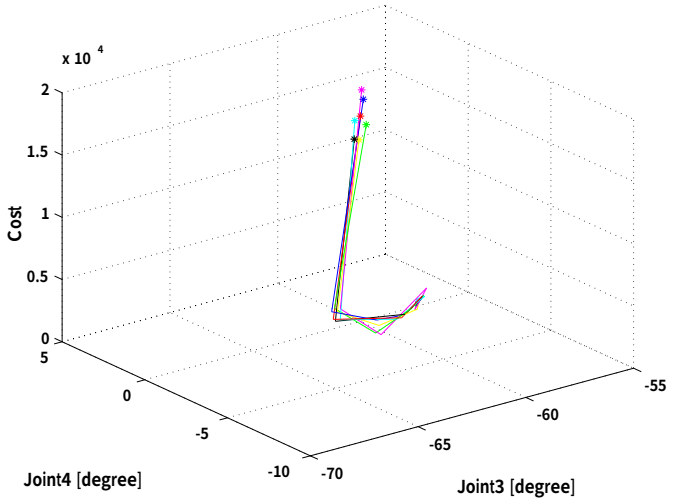


(c)Joint 5, 6, and Cost function

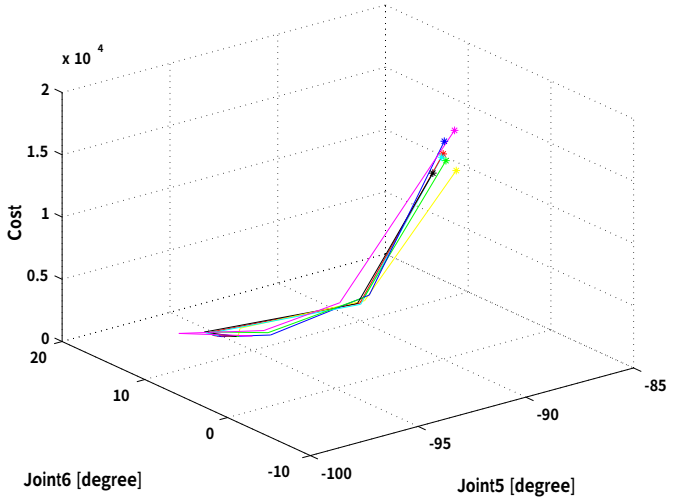
Figure 3.10: Result of the classical simplex optimization with 6DOF centering task



(a)Joint 1, 2, and Cost function

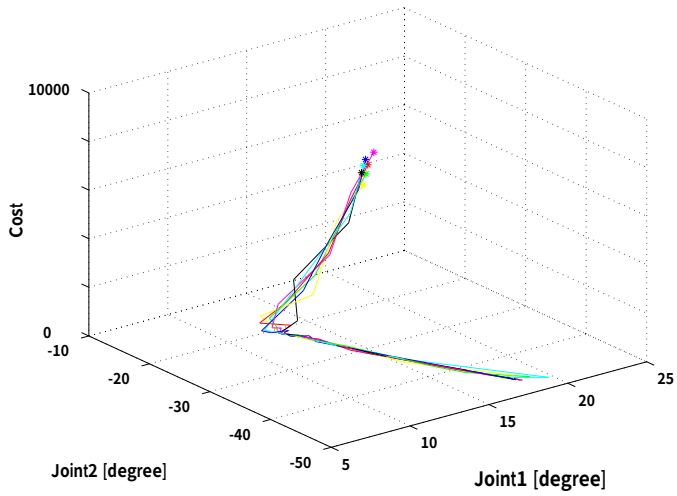


(b)Joint 3, 4, and Cost function

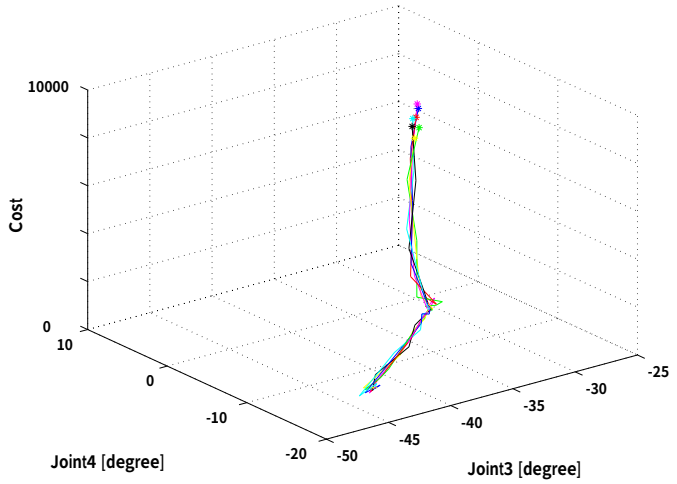


(c)Joint 5, 6, and Cost function

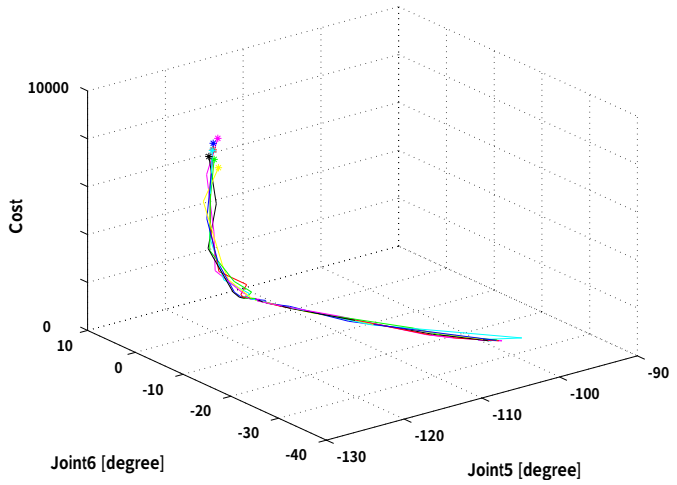
Figure 3.11: Result of the modified simplex optimization with 6DOF centering task



(a)Joint 1, 2, and Cost function

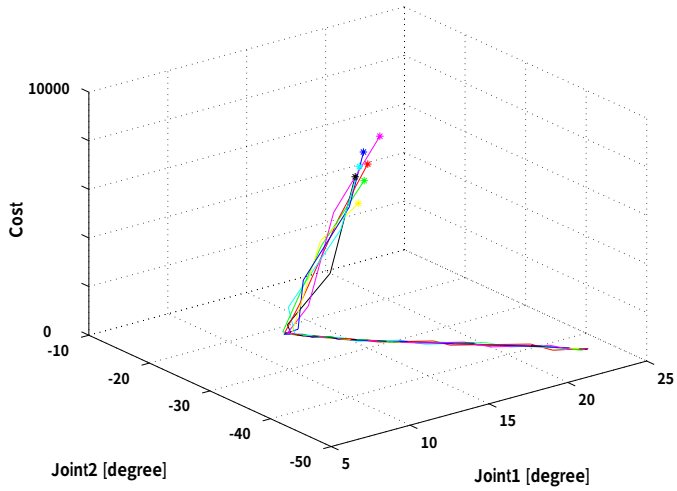


(b)Joint 3, 4, and Cost function

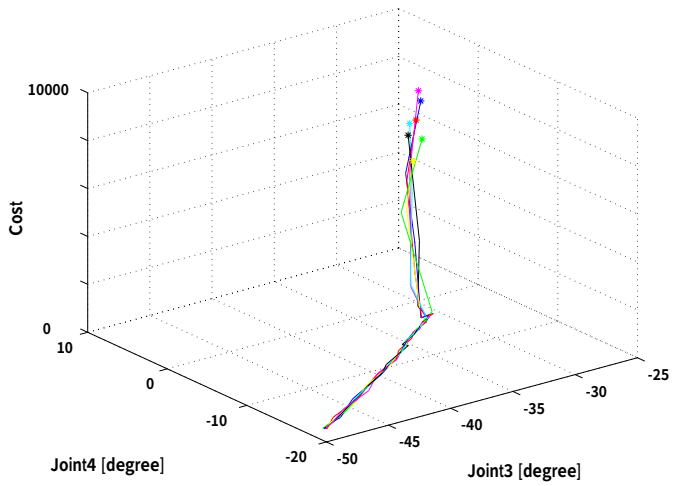


(c)Joint 5, 6, and Cost function

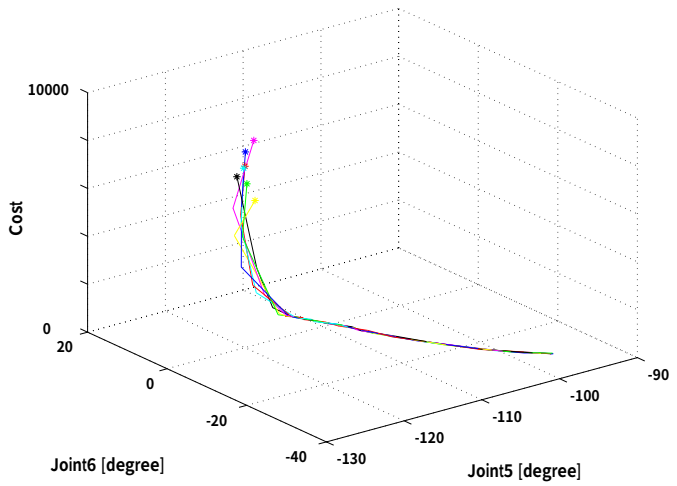
Figure 3.12: Result of classical simplex optimization with 6DOF



(a)Joint 1, 2, and Cost function



(b)Joint 3, 4, and Cost function



(c)Joint 5, 6, and Cost function

Figure 3.13: Result of modified simplex optimization with 6DOF

Table 3.1: Comparison with different optimization techniques

	number of iterations	Objective function
Gauss-Newton 2DOF centering	4	0.0003
classical classical Simplex 2DOF centering	26	0.2564
modified Simplex 2DOF centering	16	0.0388
Gauss-Newton 6DOF centering	not converged	
classical Simplex 6DOF centering	72	0.0832
modified Simplex 6DOF centering	37	0.2411
Gauss-Newton 6DOF positioning	not converged	
classical Simplex 6DOF positioning	119	2.2928
modified Simplex 6DOF positioning	112	2.3273

3.5 Conclusion

In this section, we investigated the simplex and modified simplex optimization techniques for a positioning task by visual servoing. This method does not need any model of the robot and does not require the estimation of Jacobian matrices. Constraints at the joint-level can be easily included. Since the simplex method does not need to calculate or to estimate the gradient of the objective function, a robot never goes in the wrong direction due to the bad estimation. Moreover, the objective function does not need to be differentiable. We successfully demonstrated the proposed scheme with simulations.

The convergence of simplex iteration relies on the probability and contingency. It is known that the convergence of simplex optimization technique is not so quick near the minimum. Quicker convergence could be expected by switching to Gauss-Newton iterative optimization technique with the LMS estimation algorithm, which is discussed in Chapter 2.

Chapter 4

Practical positioning task with simplex algorithm

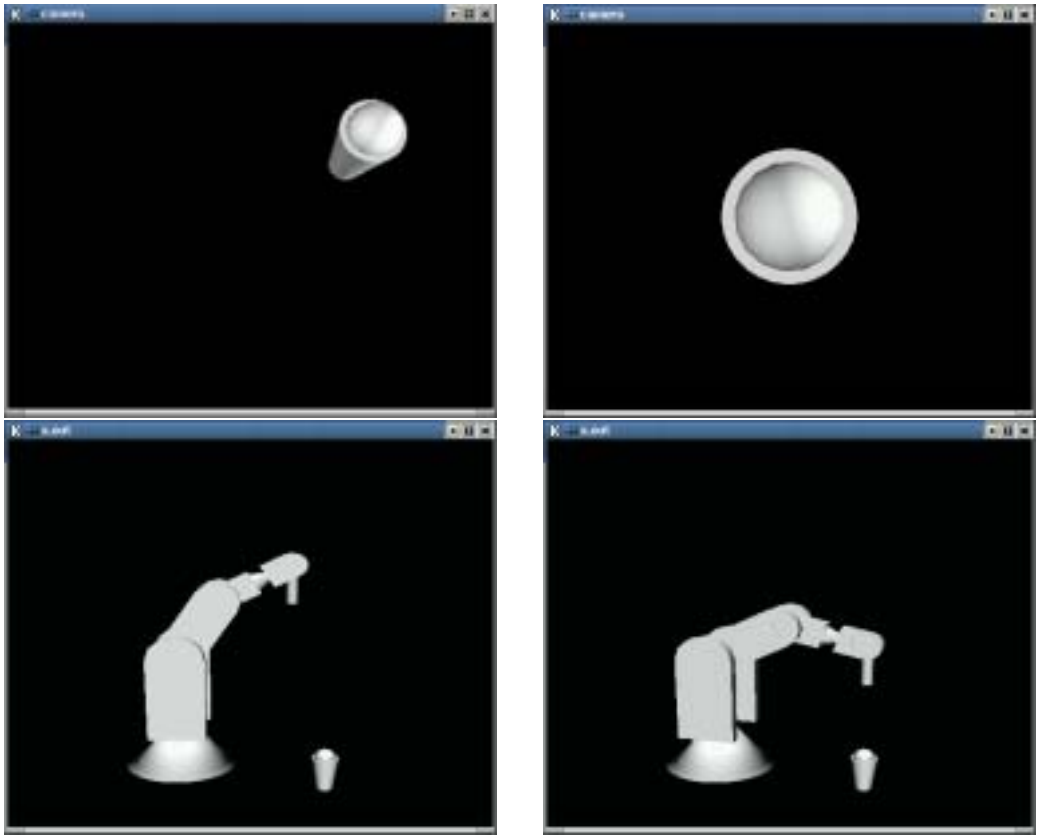
4.1 Introduction

In the previous section, a modified simplex method has been introduced. The method is efficient with complex objective functions due to redundant system configurations or could even weak with multiple vision systems. It is also compatible with non-smooth objective functions. We also found that the positioning task by the modified simplex algorithm takes time especially near the convergence. This chapter is devoted to practical applications to visual servoing tasks.

The goal is to position the end-effector fitted with a camera (eye-in-hand configuration) with respect to a static object. Our approach is generic and can be used with different type of visual servoing applications where the features used to build the objective function do not need to be differentiable nor to have analytical expansions. We will illustrate our approach with a simple positioning task with respect to an unknown object.

The task consists in bringing the end-effector at the vertical of a cylindrical object placed in the workspace of the robot as shown in Figure 4.1 (b).

Firstly, we set a task to be accomplished by using the simplex optimization technique. An industrial manipulator with 6DOF is considered as a typical example. It is desirable that the objective function scarcely has local minima. Existence of local minima may cause an irrelevant convergence. Several objective functions are proposed and compared. We also propose to restart initialization of the optimization



(a)start position (b)final position

Figure 4.1: Starting and goal position of the robot and their images taken with a camera at the end-effector

when the process falls in a local minimum.

We also present techniques for acceleration of the convergence. To accelerate the convergence, a hybrid scheme combining a simplex algorithm with an image-based partial visual servoing loop is proposed. The idea is to bring the target object to the center of the image, during the optimization process. Other modifications can be done to accelerate the convergence. Near the optimum, once a local Jacobian matrix can be accurately identified thanks to small local motions, the algorithm can switch to an iterative Gauss-Newton method.

4.2 Objective functions

The task consists in bringing the end-effector at the vertical of a cylindrical object placed in the workspace of the robot. Therefore, five degrees of freedom should be controlled by the optimization process.

4.2.1 Illustrative objective function

The objective functions are usually selected as sum of square of feature errors. A possible objective function could have the following features:

- The distance between the end-effector and the object is given by its size in the image.
- The angle between the object axis (cylinder axis) and the optical axis of the camera is given by the form factor of its image (form factor = 1 when this angle is zero). This angle can be decomposed in two elementary angles (*e.g.* roll and pitch).
- The position of the object in directions parallel to the image plane is given by the coordinates in the image of its center of mass.

For this task, we propose the following objective/cost function:

$$F = W_1(x - \frac{W}{2})^2 + W_2(y - \frac{H}{2})^2 + W_3(\frac{s}{S} - 1)^2 + W_4(\frac{l_{long}}{l_{short}} - 1)^2 \tag{4.1}$$

where W and H are the width and the height of the image, x and y are the coordinates of the center of the object in the image, s and S are the actual and the

desirable size of the object in the image, l_{long} is the longest distance from the center of the object to its contour, and l_{short} is the shortest one. Therefore $\frac{l_{long}}{l_{short}}$ is the form factor of the object image. Finally, W_1, W_2, W_3 and W_4 are the respective weights of feature errors in the cost function. Figure 4.2 shows an illustration of the features.

Clearly, even if a model of the object was known, some of these features would not have a simple analytical expression (like, e.g., the form factor). These are possible features that will allow for the control of the 5 DOF. However, there are no limitations on this type of features as long as they contribute to the definition of the desired position. This illustrative objective function requires the knowledge of the approximate shape of the object. It also needs different objective functions to reach different objective positions even with the same target object. A solution to this problem of modeling the object consists in using only the pixel values of the current and desired image as described in the next section.

4.2.2 Pixel matching by sum-of-square-difference

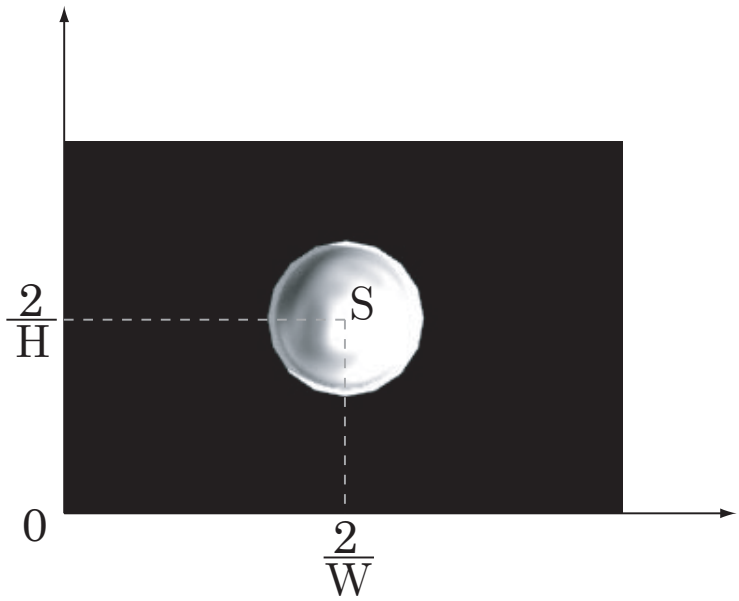
The following objective function does not need features. The sum-of-square-difference (SSD) of each pixels intensity between the current and the reference or goal image of the object.

$$F = \sum_{i,j} \left(P_{current}(i,j) - P_{reference}(i,j)\right)^2 \tag{4.2}$$

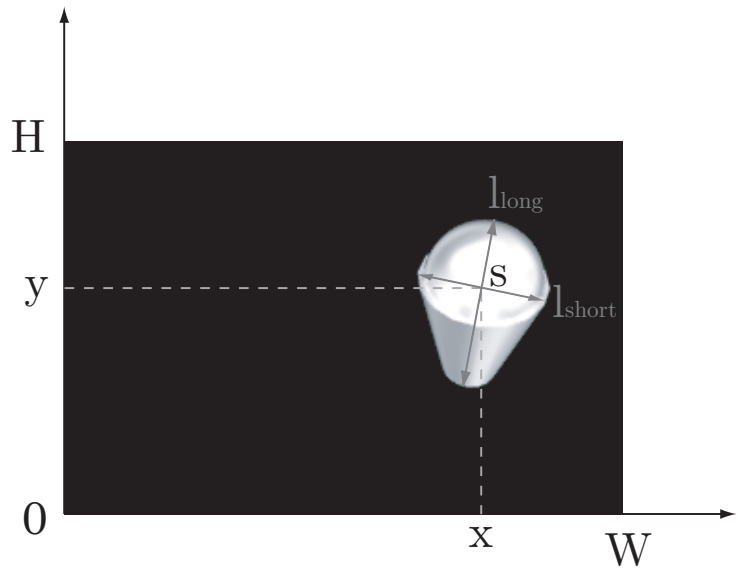
where i and j indicate the location of a pixel in the image, and $P_{current}(i,j)$ and $P_{reference}(i,j)$ are the intensity of (i,j)th pixel in the current and reference image, respectively. This objective function is independent of the target object or the objective position. The objective is easily learned, since it uses only in a view of the object at the right position. Therefore, this is no special objective function tuning.

4.2.3 Simulations

Simulations were carried out using the modified simplex method presented in section 3.3.1. The positioning process was implemented under the graphic library OpenGL to model the robot SCEMI 6P01. The target is a cylindrical white cup of 11cm height without handle on a black background. The task consists in bringing the end-effector at the vertical of a cylindrical object placed in the workspace of the robot as shown in Figure 4.1 (b). To compare several optimization runs, we use



(a)Reference image



(b)Current image

Figure 4.2: Concept of the illustrative objective function

the same starting positions as shown in Figure 4.1 (a). The target object is the cylindrical white cup used in the last section.

To compare the efficiency of the objective function, simulations with the illustrative objective function of Equation (4.1), the SSD pixel matching of Equation (4.2) are carried out. The combined objective function, that sums up these two objective functions, is also tested. For these simulations, different weights are chosen experimentally in order to have nearly the same cost at the starting and final positions for the two cost functions.

The tolerance of convergence ε is set to 1.0% of the mean cost between the starting $(n+1)$ positions, and the termination tolerance is 1.0%.

Figure 4.3 shows the convergence of the distance and the angle of the end-effector with respect to the object. The robot gradually approaches the optimal position.

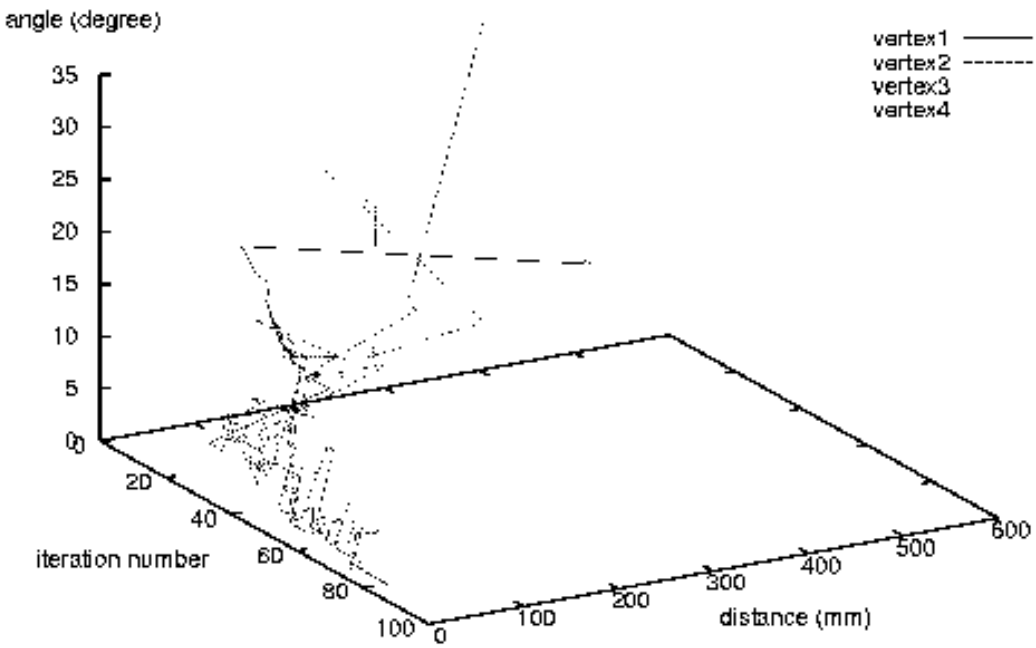


Figure 4.3: Convergence of the cost with the complex method. The operation takes 86 steps.

4.2.4 Influence of weightings for illustrative objective function

In order to converge faster, it is important that the cost function does not have local minima. Our illustrative function uses two image features: size and form factor. Each feature creates different local minima. Figure 4.4 and Figure 4.5 are the plots of size and form factor at each joint angle. The optimized joint angles of the robot are (Joint1, Joint2, Joint3) = $(0, 60, 70)$[degree]. In Figure 4.4, when it is far from the global minimum, the size of the object's image does not make difference of the objective function, and it makes large peak near the global minimum because it approaches the object too close. In Figure 4.5, the form factor makes a wide local minimum. The differences between global and local minimum are small. However, the positions where these features make local minimum are different. Thus the cost function that has small local minimum may be available by adjusting the weights of two image features. Figure 4.6 shows the objective function with $W_3 = 250, W_4 = 250$. Figure 4.6 shows the objective function with $W_3 = 150, W_4 = 450$. When we take the correct weights on the image features, the objective function without large local minimum is available.

4.2.5 The comparison of different cost functions

The comparison of the different cost functions is given in Table 4.1. The operations symbolized in the table are as follows:

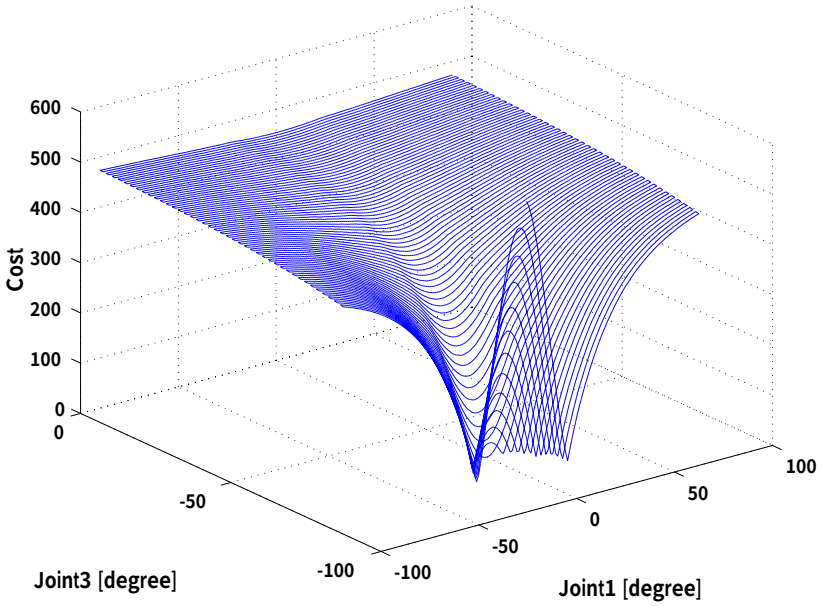
Ill: Simplex method with the illustrative objective function.

Ssd: Simplex method with the SSD pixel matching.

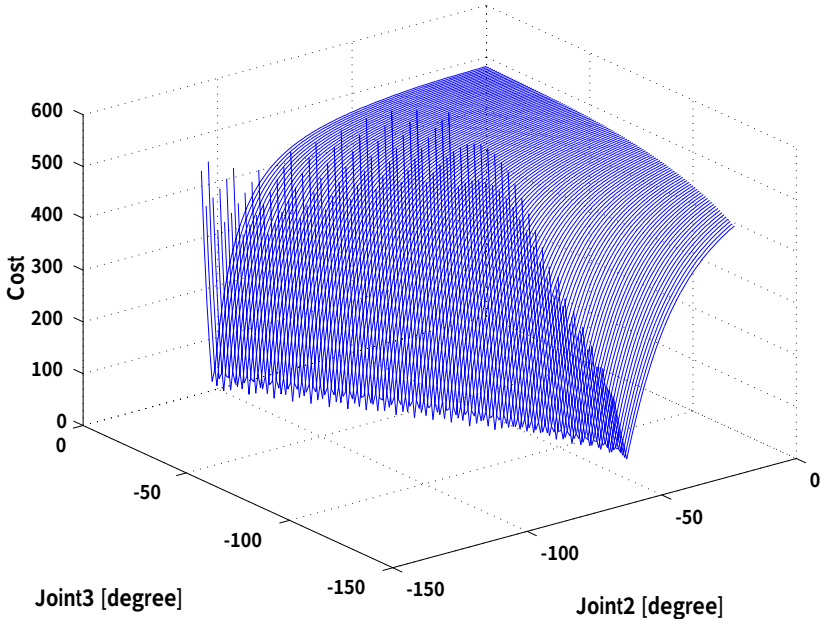
Cmb: Simplex method with the objective function combining **Ill** and **Ssd**.

The result on Table 4.1 are the mean between 5 successful runs. The operation over 50 iterations were stopped and omitted from the calculation.

From the resulting errors, it can be said that the combined objective function (**Cmb**) has the best convergence in this configuration. With cylindrical objects, the SSD objective function has local minima as shown in Figure 4.9. Far away from the global minimum, the objective function does not make large difference. The illustrative objective function was efficient because it contained size error. However the illustrative objective function had a wide local minimum as shown in Figure 4.8

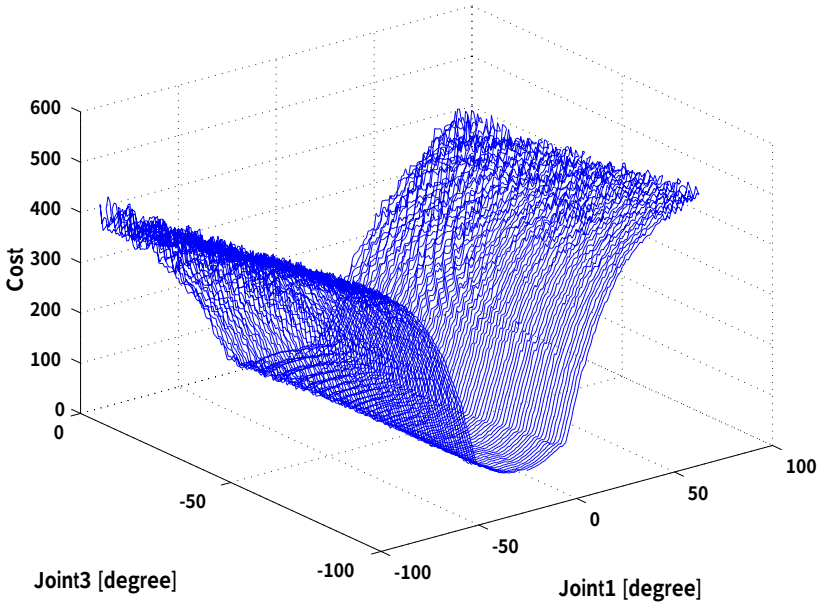


(a)Joint 1, 3, and Cost (Joint 2=60[$degree$])

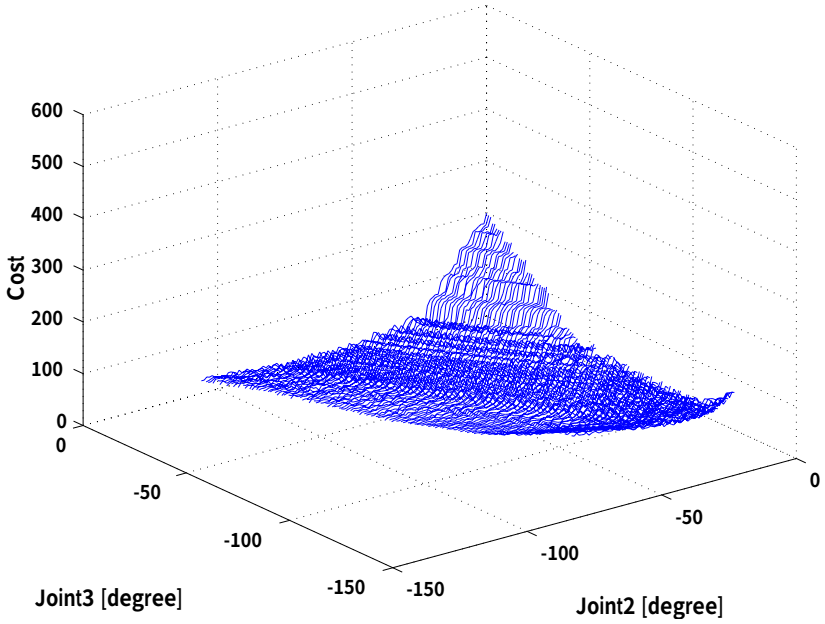


(b)Joint 2, 3, and Cost (Joint 1=0[$degree$])

Figure 4.4: Illustrative objective function $W_3 = 500, W_4 = 0$

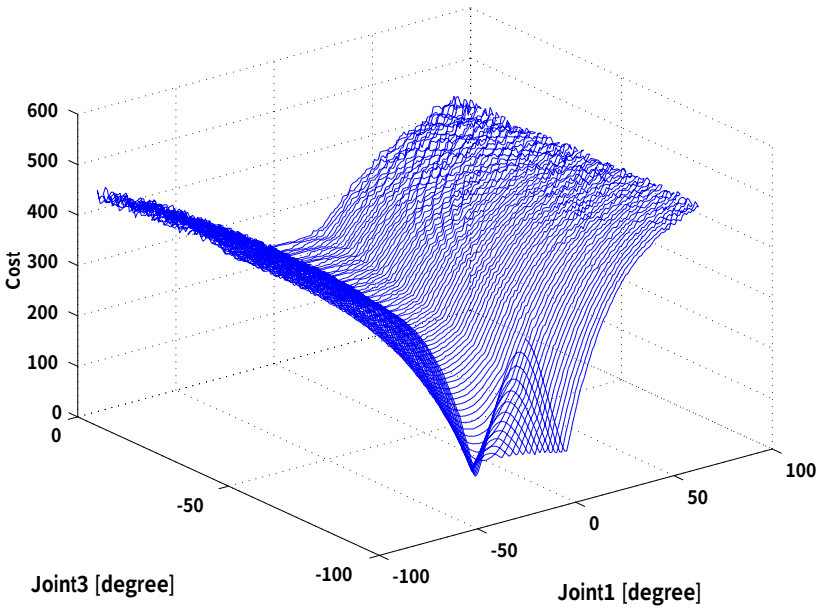


(a)Joint 1, 3, and Cost (Joint 2=60[$degree$])

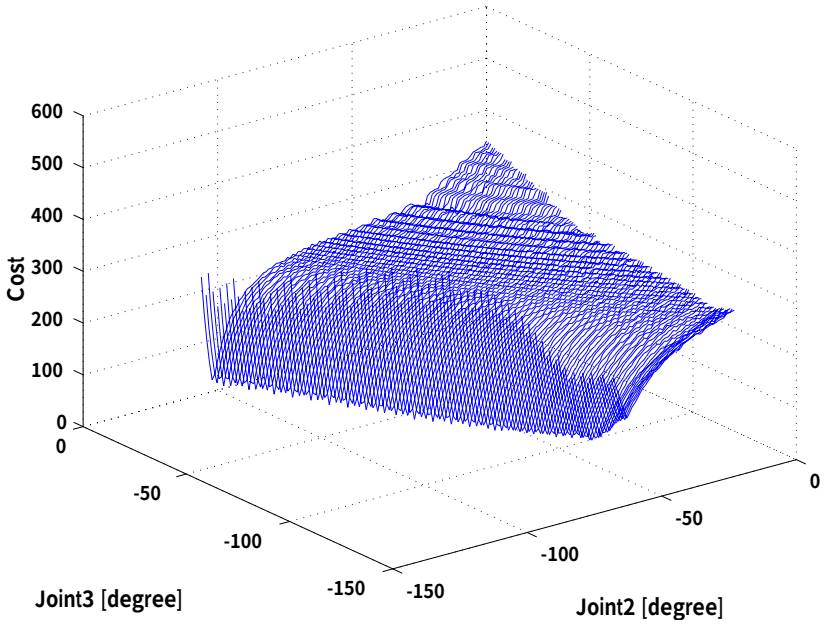


(b)Joint 2, 3, and Cost (Joint 1=0[$degree$])

Figure 4.5: Illustrative objective function $W_3 = 0, W_4 = 500$

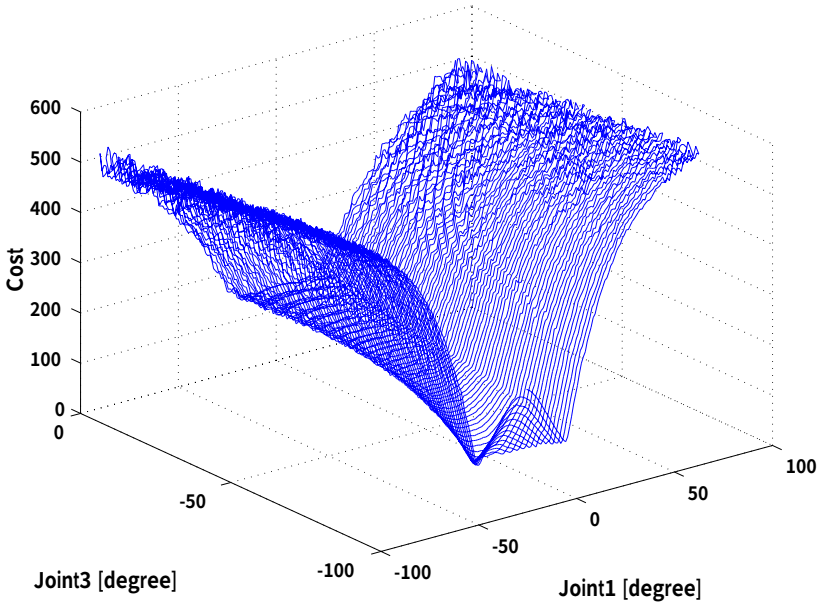


(a)Joint 1, 3, and Cost (Joint 2=60[$degree$])

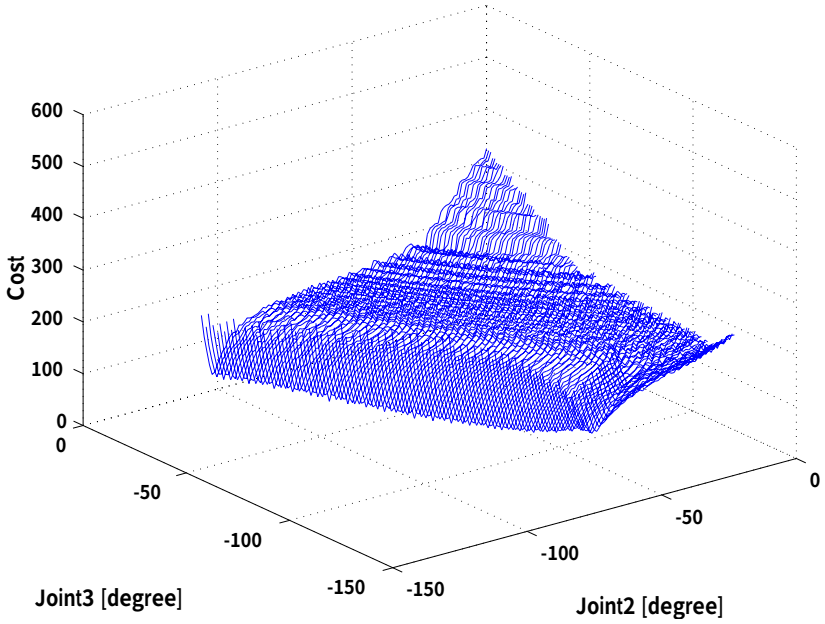


(b)Joint 2, 3, and Cost (Joint 1=0[$degree$])

Figure 4.6: Illustrative objective function $W_3 = 250, W_4 = 250$



(a)Joint 1, 3, and Cost (Joint 2=60[*degree*])



(b)Joint 2, 3, and Cost (Joint 1=0[*degree*])

Figure 4.7: Illustrative objective function $W_3 = 150, W_4 = 450$

	Ill	**Ssd**	**Cmb**
best result			
distance error (mm)	24.1	39.0	28.8
angle error (degree)	3.4	5.5	4.1
number of iterations	26	23	49
worst result			
distance error (mm)	159.7	65.9	64.5
angle error (degree)	22.5	9.3	9.1
number of iterations	9	31	25
average			
distance error (mm)	92.2	51.6	44.3
angle error (degree)	13.0	7.3	6.3
number of iterations	23.8	30.6	31.6
success rate	5/5	5/9	5/11

Table 4.1: Result of the Simulations

near the global minimum. Its form factor (the difference between the longest and the shortest strait lines intersecting at the center of the object image) is not large, and it does not give a great cost when the size of the object image is nearly correct. So the process always converges quickly, but often falls in local minimum.

The range of errors was the smallest with the combined objective function. The two functions seem to offset the value of local minima each other as shown in Figure 4.10. From the result on success rate, it seems to have poor convergence properties, however, it arrives at near the global minimum and wanders around there. It can be expected to give an accurate convergence with a larger iteration limit.

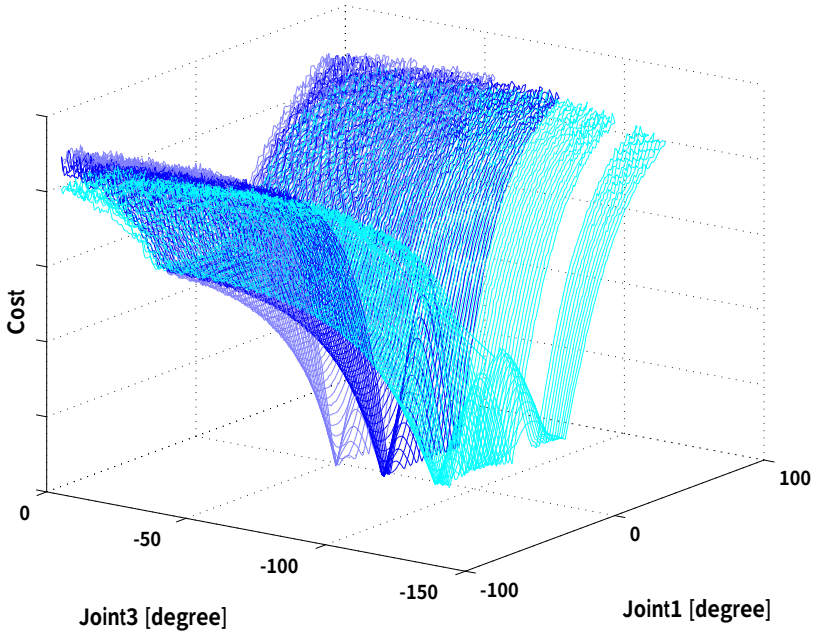


Figure 4.8: Illustrative objective function

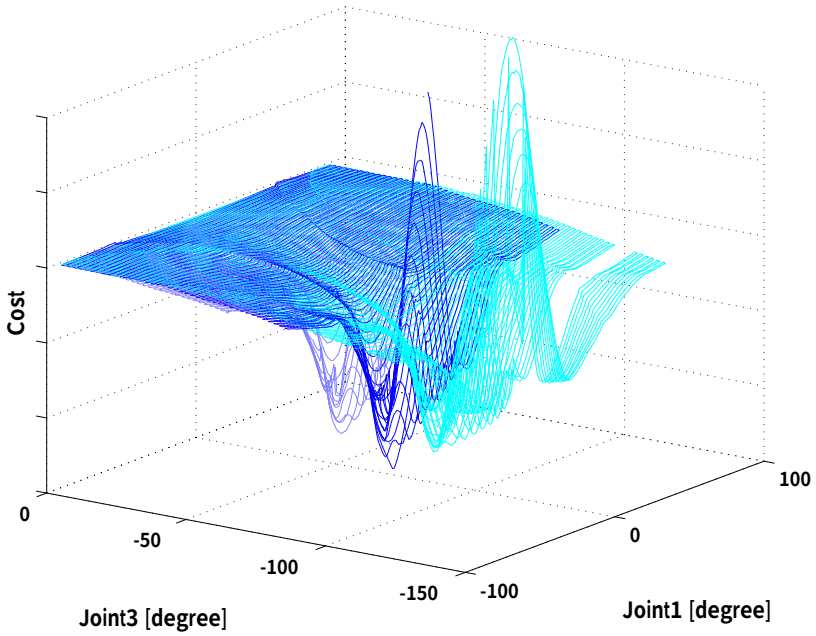


Figure 4.9: Sum-of-squared-difference pixel matching

Plots with different joint angle.
Joint 2= 70(obscure blue), 60(blue), 50(lightblue) [*degree*]

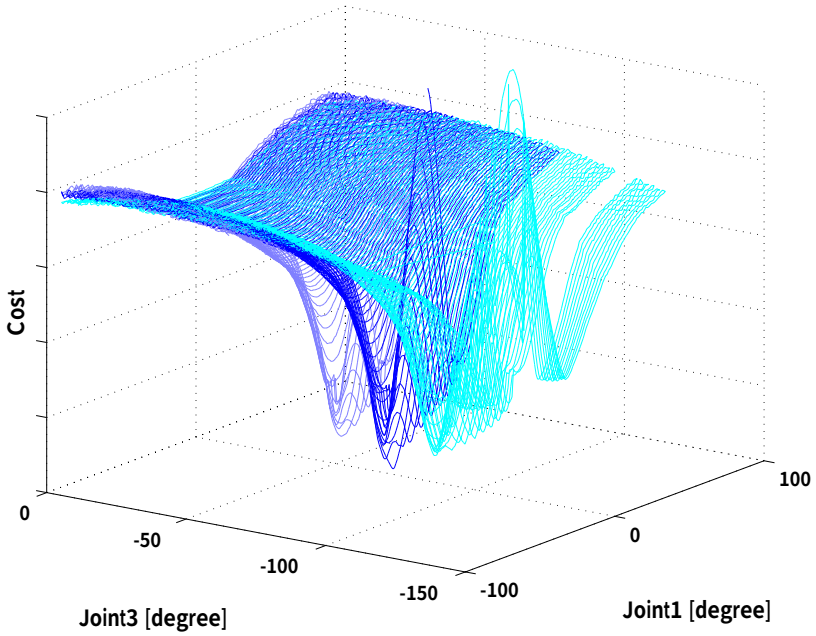


Figure 4.10: Combined objective function

Plots with different joint angle.
Joint 2= 70(obscure blue), 60(blue), 50(lightblue) [*degree*]

4.3 Improvement of the convergence

4.3.1 Hybrid scheme

Acceleration of the convergence can be achieved by reducing the dimension of the variable space. For example, to control some degree of freedom by image-based visual servo loop is one of the solutions. To do this, we add a constraint on the position of the robot end-effector. A simple image-based visual servo loop is used to maintain the position of the object in the center of the image. This visual servo loop guarantees that for each position of the other joints, the image remains centered. So the iterative search is limited to a reduced subspace in the initial variable space. It is an advantage of the simplex to be executed with other control method like this.

The illustrative objective function of Equation (4.1) is used on this simulations. Different weights (W_1 to W_4) for each runs are chosen experimentally for fast convergence. Since only five image DOF are controlled by vision and we use a 6 DOF robot, one DOF must be constrained. To do so, we simply constraint one of the

angle of the wrist to be equal to zero at any time.

To compare several optimization runs, we use the same starting positions as shown in Figure 4.1 (a). The tolerance of convergence ε in Equation (3.11) are set to 0.25% of the mean cost between the starting $n+1$ positions. To escape local minima, which explained in section 4.4.3, simplex iteration is terminated when it converge with the cost function lower than 1.0% of the mean cost between the starting $(n+1)$ positions.

The comparison of different optimization runs is given in Table 4.2. The operations symbolized in the table are as follows:

S: Simplex method without image-based centering.

M: Modified simplex method without image-based centering.

Si: Simplex method with image-based centering.

Mi: Modified simplex method with image-based centering.

	S	**M**	**Si**	**Mi**
number of iterations	144	108	98	86
distance error (mm)	23.9	33.3	9.6	4.0
angle error (degree)	7.5	10.3	3.1	1.3
image size error (%)	0.47	0.82	0.37	0.23
form factor	2.89	5.20	2.18	1.02

Table 4.2: Result of the Simulation

The method **Mi**(modified simplex method with image-based centering) has the best result among them. The use of an image-based visual loop clearly improves the optimization time and accuracy.

It is easy to guess that the number of iterations becomes generally small when the number of variables is small. Even if the end-effector comes to the good position, total value of the objective function becomes large when the object is far from the center of the image. This effects inaccuracy of the converged position.

4.3.2 Variable space

Joint space

The joint angles of a manipulator can be used as the variables for simplex process. Since only joint-level controller is used, the process doesn't need any Jacobian matrix calculation that needs kinematic model of the manipulator. It is also easy to include joint-space constraints with complex method explained in section 3.2.2. Cartesian-space constraints are also available in the same way if a direct kinematic model is known. However, the cost function in a joint space may have complex local minima caused by the structure of the robot. Depending on the structure of the robot, it may also be difficult to control some joints by image-based visual servo loop for the purpose of keeping the target object in the image.

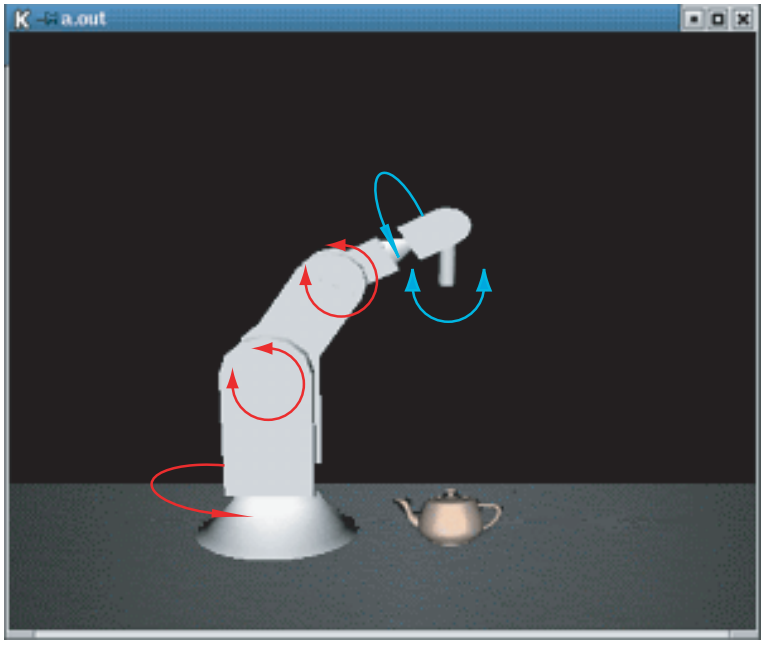
Cartesian space

When the inverse kinematics model of the manipulator is known, the pose of the camera on the end-effector is available as variable space for the simplex algorithm. It could be used to reduce the dimension of the variable space, since the position and size of the object in the image can be adjusted easily with the visual servo loop, which was described in section 4.3.1. For example, translations in x, y, z directions are applied to maintain the position of an object in the center of the image and its size in a number of pixels, then the dimension of simplex variables vector is three. Furthermore, when rotated images are generated off-line, then all the rotated images at this position can be compared with the reference image, then the optimization for the rotation angle can be neglected and then the dimension of simplex variables vector becomes two.

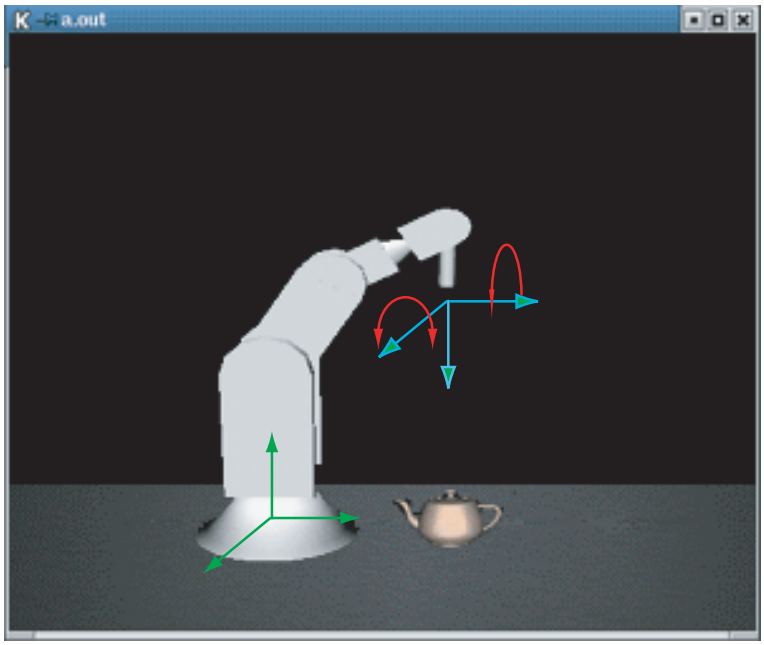
Simulations with Joint space and Cartesian space

Simulations are carried out, using the pose of the camera as variables. We use translations of the camera in x, y, z directions to maintain the position of an object in the center of the image and its size in certain number of pixels with an image servo loop. Rotated reference images are generated off-line before the optimization process. Then the number of variable for simplex operation is only two: pitch and yaw of the camera.

The simulation is executed with the modified simplex method, using SSD pixel



(a)Joint angles



(b)Camera pose

Figure 4.11: Different variables; the variable controlled by simplex operation is signified with red, by visual feedback with blue

matching as objective function. The target object is a teapot as shown in Figure 4.12. The start position of the camera is just above the object; pitch= $\frac{2}{\pi}$, yaw= 0. Different 24 goal images are tested. The tolerance of convergence ε and the termination tolerance are set to the same values for any goal images. The operation is stopped when the number of iterations exceeds 50.

The results are shown in Figure 4.13. For the purpose of comparison, the results with joint space are shown in Figure 4.14.

The process with Cartesian space has better convergence. With joint space, it is sometimes difficult to exceed local minima because of the structural restriction of the robot. The symmetry images can be local minima of each other, and the images obtained at same angle of pitch and same absolute angle of yaw, too. The process with Cartesian space sometimes falls into these local minima.

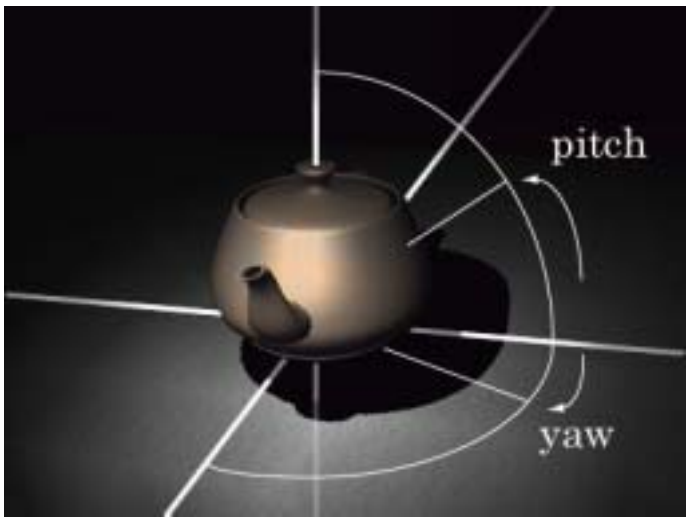


Figure 4.12: Target object

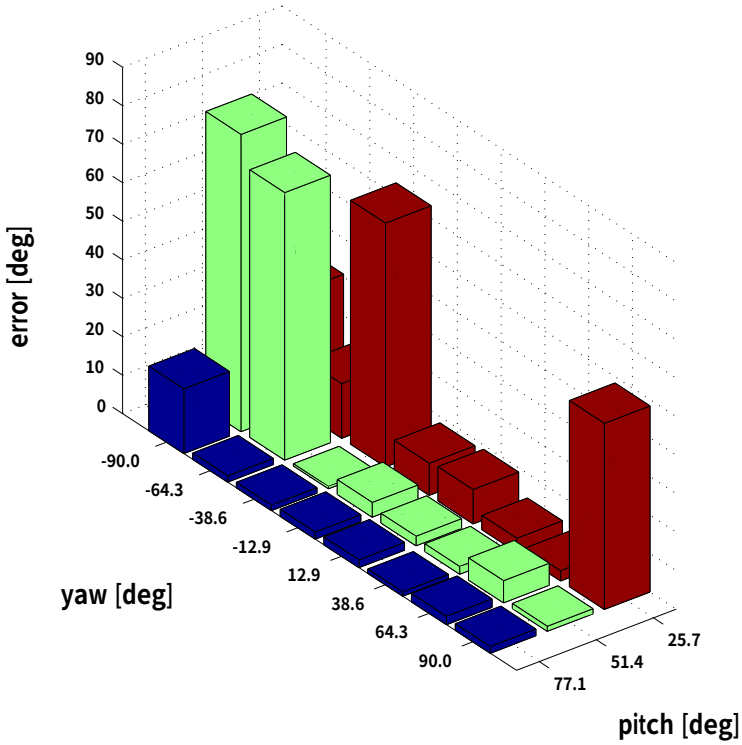


Figure 4.13: Pose error for simplex in Cartesian space

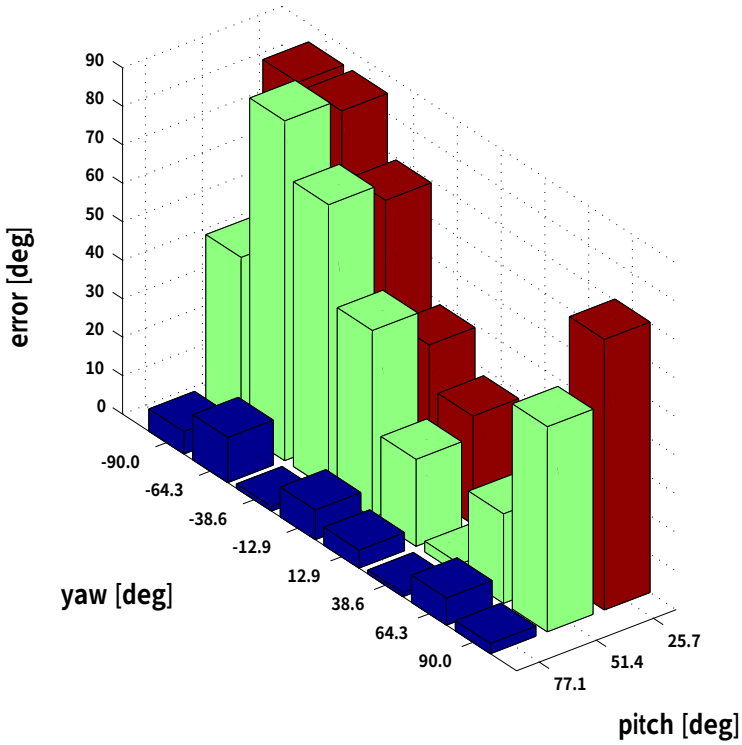


Figure 4.14: Pose error for simplex in Joint space

4.4 Solution for local minimum

4.4.1 Switching to Newton-like iterative methods

The classical and the modified simplex algorithm take time especially near the convergence. On the other hand, the simulations in the previous chapter showed that the process using Gauss-Newton method has converged very well near the local minimum. Therefore, it is naturally to think of switching algorithms depending on the optimization progress. The main idea is to estimate the image Jacobian during the simplex optimization search, and once the optimization process arrives near the optimal position, the optimization methods switch to the Gauss-Newton iterative method. Faster convergence is expected with this technique near the optimum.

4.4.2 Multiple cameras

Since the objective function for the simplex algorithm has multiple image features, it is also useful to employ several cameras. When the camera is set to observe the workspace of the robot, it is possible to give a penalty objective function not to exceed the workspace limit, e.g. not to touch other obstacles. The information about the distance between the end-effector of the robot and the object is also available, as shown in Figure 4.15.

4.4.3 Going out of a local minimum

When the cost function at $n + 1$ vertices converges and still has a large value, it means that the algorithm has reached a local minimum. So, in this case, a simplex process with new vertices is started. In this new simplex, the base point X_1 is the last minimum point X_l.

Since the simplex process is restarted each time when it falls in a local minimum, if the cost function F has many local minima, the simplex process may need many iterations and so takes a lot of time. So it is better to find a relevant cost function and proper weights to avoid local minima.

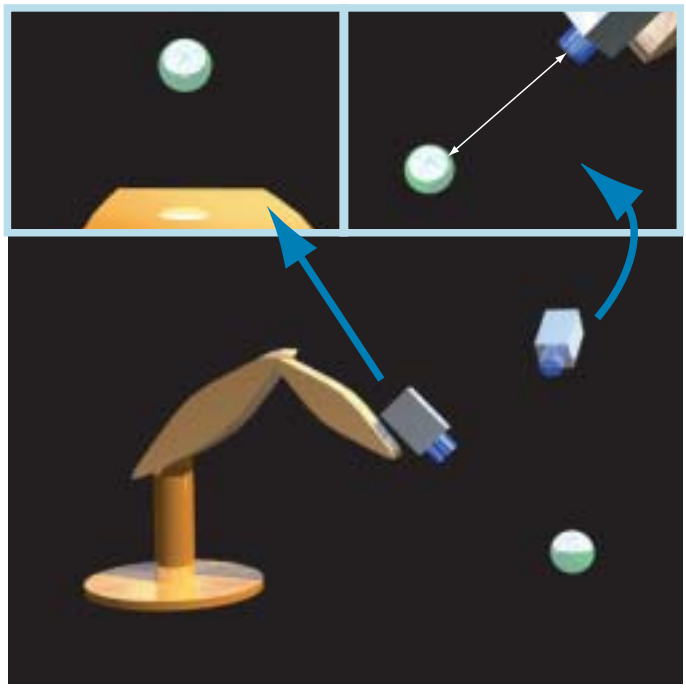


Figure 4.15: Using multiple cameras

4.5 Experiments with an industrial robot manipulator

The robot is a SCEMI 6P01 industrial manipulator with 6 rotational joints. The camera is a PAL CCD black & white camera. The frame grabber is an Imaging Technologies PCI board (PCVISION). Its host is a PC (bi-Pentium III 800 MHz) running the Linux operating system. It performs image processing and controls the optimization process. The target that we use in the experiment is a 12cm long black screw placed on a white table as shown in figure 4.16. The desired position of the end-effector with respect to this screw is at an approximate distance 40 cm.

base position	1	2	3	4
joint 1	-0.2085	-0.1163	0.6638	0.1509
joint 2	0.4945	0.8548	0.7770	0.1155
joint 3	1.4900	1.1452	1.0119	1.0804
joint 4	0.0000	0.0000	0.0000	0.0000
joint 5	-0.2257	-0.0835	0.1019	-0.2640
joint 6	0.2993	0.4465	-0.4206	0.5537

Table 4.3: Initial base positions X_1(rad)

Table 4.3 shows the four initial robot joint positions (X_1) that are used in the experiments. We use the modified simplex method with image-based centering (that we call **Mi**). The first three joints are the variables of the cost function (x_1, x_2, x_3), joints 5 and 6 are used for the centering and joint 4 is set to 0.

Figure 4.17 shows the cost function evolution with initial position 1. The cost function gradually decreases, and the target projection in the image also changes toward the desired image. The function converges at the 7th iteration and restarts with new simplex again to go out of the local minimum.

Figure 4.18 shows the cost function for the 3 other initial positions. The number of iterations needed to converge depends highly on the initial position.

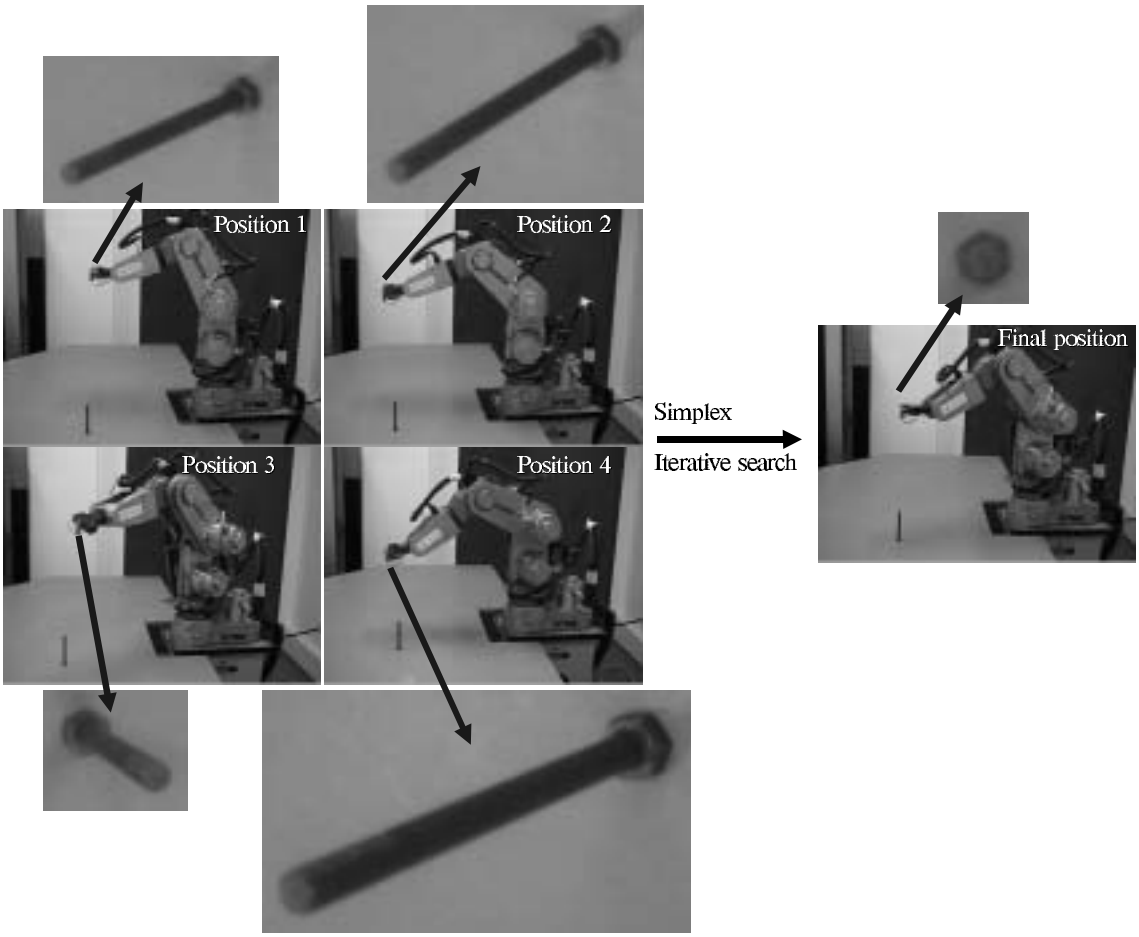


Figure 4.16: Starting position and final position of the manipulator and corresponding images acquired by the camera.

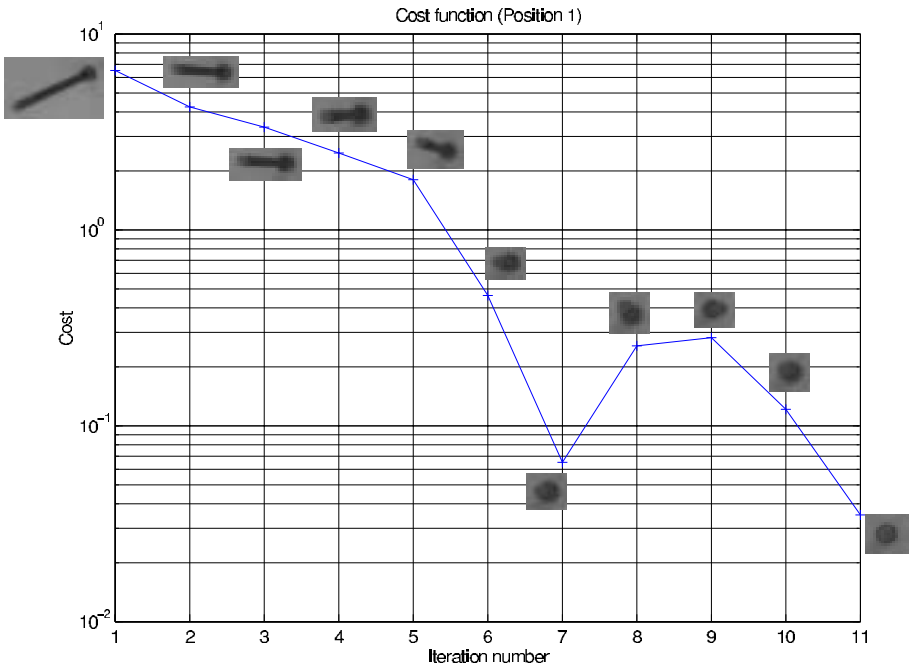


Figure 4.17: Evolution of the cost function with initial position 1

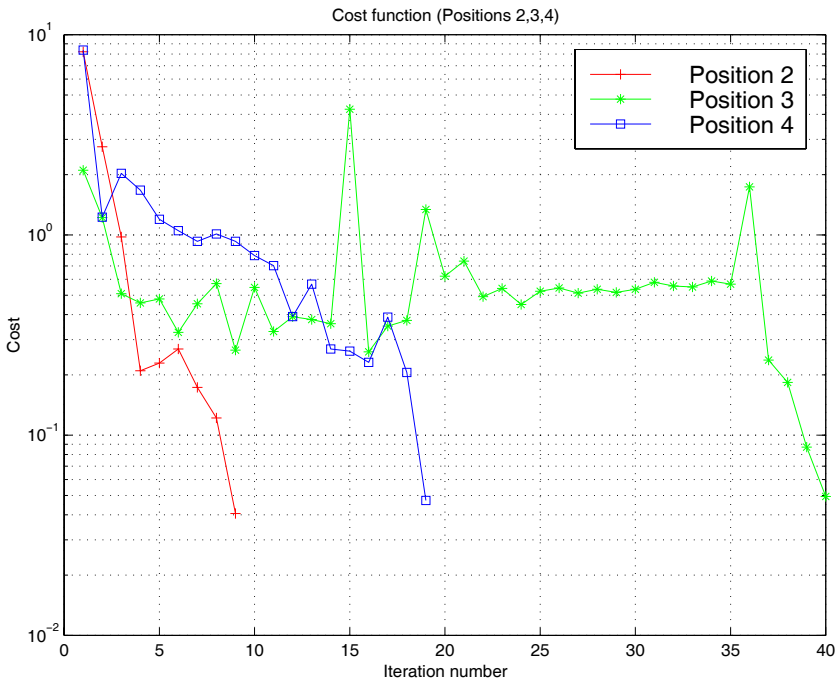


Figure 4.18: Evolution of the cost function with initial positions 2,3, and 4

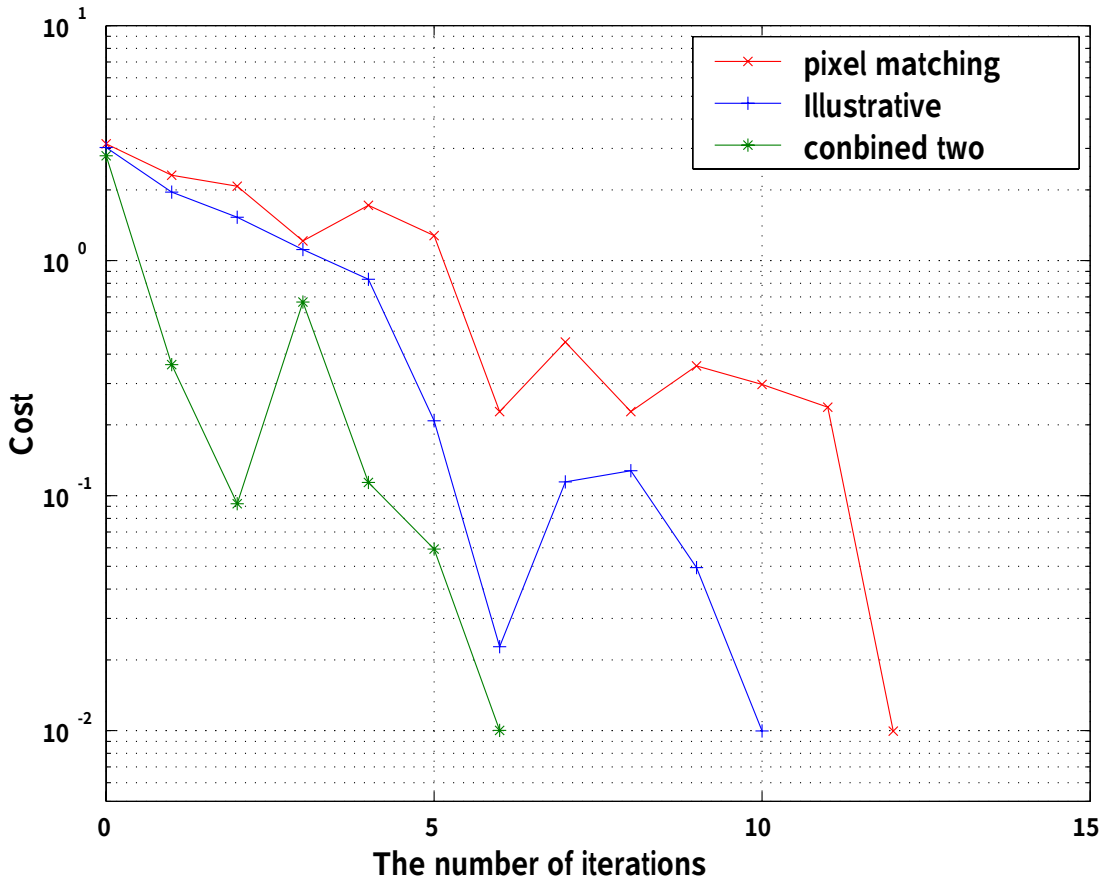


Figure 4.19: Comparison of the objective functions

4.6 Conclusion

Focusing on the convergence, the process with the illustrative function, SSD pixel matching, and the combining cost function of these two were simulated. It was difficult to converge with the pragmatic limit of only 50 iterations, the positioning errors were not so small (2-4cm). Stable convergences were realized with the combined objective function. Although the number of iterations was not improved with these cost functions, the accuracy of the converged position can be improved by combining several cost functions.

The comparison between joint angles and camera pose as the optimization variables was also done. The process has better convergence with smaller dimensions of the variable vectors. Joint-level simplex operation also has difficulty to go out from local minimum due to the structural constraints of the manipulator, even if the global minimum is inside its workspace limits. With Cartesian space optimization, the problem arises only from the local minimum of the objective function. When Cartesian space is used for simplex operation with such manipulator, workspace has to be considered by complex method.

We found that the cost function has local minima, and that the number of iterations needed to converge depends on the initial position. The positions where local minima exist depends on the target object and the objective function. Further work could focus on improving the cost functions for rapid convergence and robustness to initial positions. For example, switching cost functions during the process is a possible solution for this problem.

Part II

Grasping

Chapter 5

Measurement of human grasping motion

5.1 Introduction

Recently, a lot of research has been done on humanoid robots aiming to coexist with humans. It is necessary that such robots have high-functional end-effector in order to carry out complex task in the human living space. To realize such a dexterous end-effector, analysis and application of human skills can be a solution.

It is known that humans have excellent skill in manipulating and grasping objects. The analysis of various features of human grasping has been made until today. About the motion to lift up a target object, Westling [77] and Jenmalm [31] reported the effect of changing friction coefficient and curvature of the contact surface on the behavior of human grasping. The development of the ability to control the grasping force along the age is investigated by Forssberg [16] and Kinoshita [37]. It has been observed that when a human handles and carries a target object, the grasping force is delicately adjusted to the inertial force created by its movement [34]. This characteristic has been applied for the control law of the robot hand in the grasping task [57]. The analysis of the motions necessary to put an object on a table has been made by Nakazawa [52], and it follows that humans unconsciously decrease the grasping force to the minimum slip, in order to reduce the impact force, while avoiding that the object slips.

Humans are able to touch an object without large displacements even if it is light and could easily fall. Although such skill is very important when the object is

fragile, little research has been made so far on this type of grasping.

In this chapter, experimental studies are carried out on the human grasping with the index finger and the thumb (precision grasp). The contact force and the displacement of the object are analyzed depending on the fingertip which contacts first to the target object. Motions of human fingertips are investigated with objects of deformable width. The experimental results reveal the function of the index finger and the thumb during grasping.

5.2 displacement of the target object

5.2.1 Experimental set-up

The system arrangement for the experimental set-up is shown in Figure 5.6. The target object was put on a table. A digital video camera was set $600mm$ above the table. It recorded the displacement of the markers on the target object at $30Hz$ (i.e. $\frac{1}{30}sec$ sampling interval), and the displacement of the object raised by the contact with the human fingertips was analyzed. Light emitting diodes were also taken by the camera for the detection of the contact time instant between the target object and the table, and between the fingertip of the subject and the target object. They turned off when the object had no contact with the table nor one of the fingertips of the subject.

The view taken by the camera is shown in Figure 5.2. The target object was an empty can. Fiducial marks were put on the upper face of the object to measure the displacement with a digital video camera. The weight of the object was successively $30, 120, 210$, and $300g$, and it was adjusted by putting calibrated weights into the can.

A subject sat down on a chair $400mm$ height, and grasped the target object at a distance of $350mm$, then lifted it up to around $50mm$ high. The subject pinched the object with the index finger and the thumb. He was asked to make it quickly or slowly. The grasping was demonstrated five times for each weights and speed conditions.

The experiment was made by three subjects(S_1, S_2, and S_3), who were male and female students in their twenties. Practices have been carried out several times in order to accustom the subjects to the handling of the target object.

5.2.2 Results and discussion

Figure 5.5 indicates the average of distance of five markers from the position of the target object before the contact by a fingertip to the position at the time instant of lift from the table. It is clear that the heavier the target object, the smaller the displacement distance, and that the quicker motion, the larger the displacement distance. Note that subjects moved the target object around 1.0cm maximum, but never dropped it. It can be observed that the two fingertips hold the object in the early phase and that they take some time to lift it up.

Figure 5.4 shows the typical pattern of displacement of the target object depending on the fingertip that first made the contact. The starting position of the target object is colored in red. The position where one fingertip touches the object is colored in blue, and when both fingertips touch same time, it is colored in green. In the case when the index finger touches first, as shown in Figure 5.4 (a), the displacement distance made by the index finger was as large as the one during the phase of lift-up. The direction of displacement made by the first contact was the composition of the movement of the index finger and the wrist. In the case that the thumb touches first, as shown in (b), Most of displacement has been made by the thumb. That indicates that the movement of the arm in horizontal direction has been almost stopped when the thumb touched the object.

Subject		first touched finger		
		index finger	thumb	simultaneity
S_1	fast motion	13	0	7
	slow motion	14	3	3
S_2	fast motion	9	1	10
	slow motion	8	4	8
S_3	fast motion	7	3	10
	slow motion	4	15	1

Table 5.1: Count numbers patterned on the first touched finger

The frequency of first contact by different fingertip is indicated on Table 5.2.2. The pattern of this frequency was largely depended on the subjects. The frequency of the first contact by both fingers simultaneously tended to increase with quicker motion. It was caused by the sampling rate of the video camera that cannot detect

small time difference of the contact.

The bar diagram in Figure 5.5 indicates the distance of displacement during first contact and lift-up motions. When the thumb contact first, it tends to move the object more than during lift-up motion. When the index finger touches the target object first, it moves it both during the first contact phase and the lift-up phase. This trend agrees with Figure 5.4.

The following conclusions can be drawn from these experimental results:

- human slightly moves the target object, but never let it fall down.
- When the index finger touched the target object first, the arm continued on moving horizontally.
- When the thumb touched first, the arm stopped moving in the horizontal direction, then perpendicularly lifted the target object.

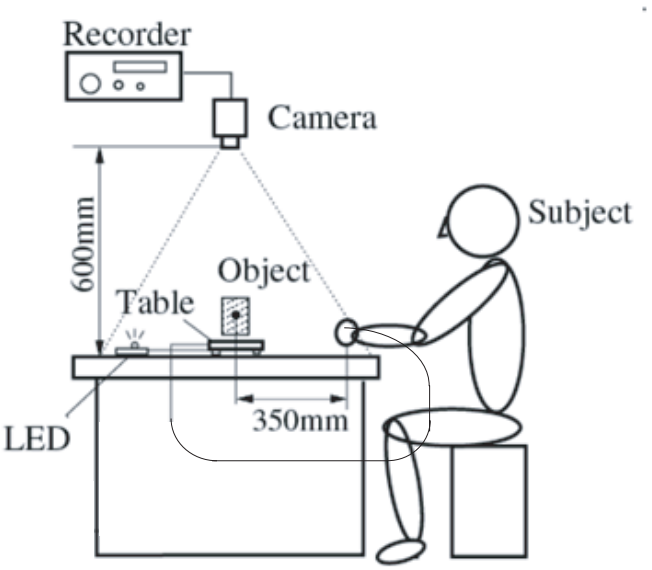


Figure 5.1: System arrangement for the experimental set-up

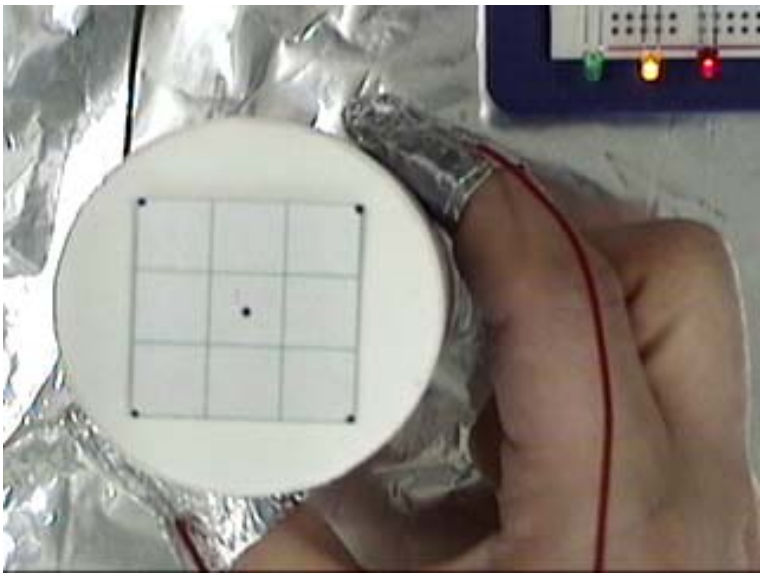


Figure 5.2: Camera view of experimental object and markers

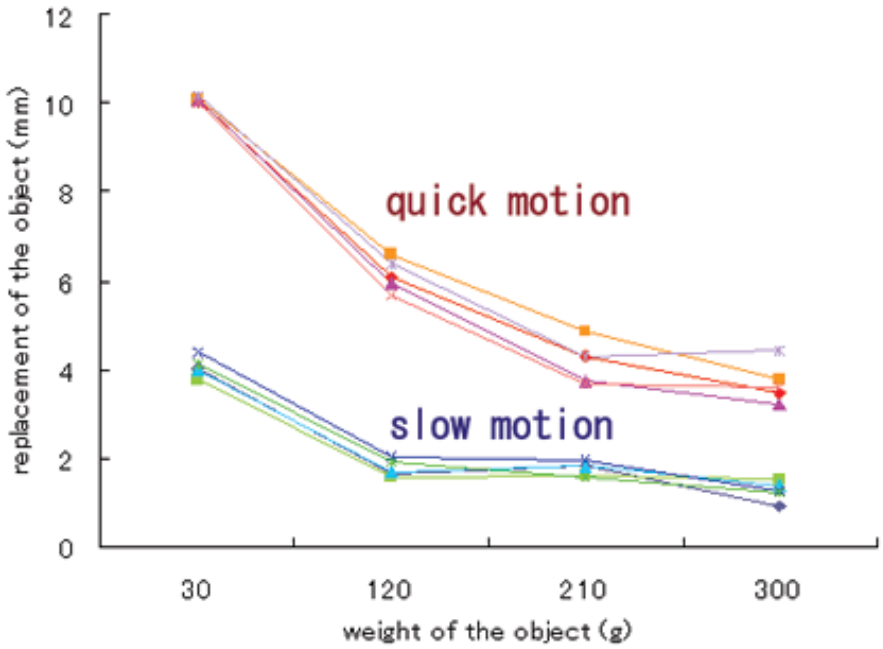
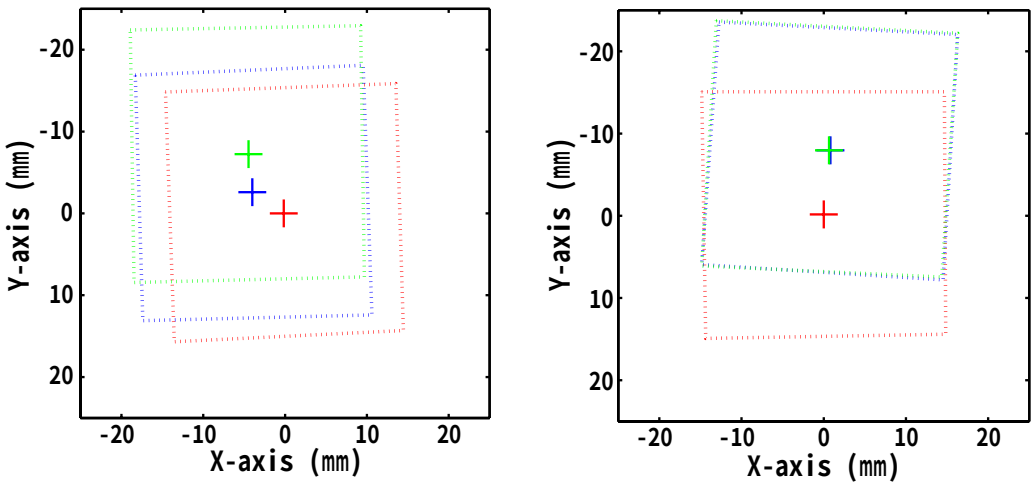


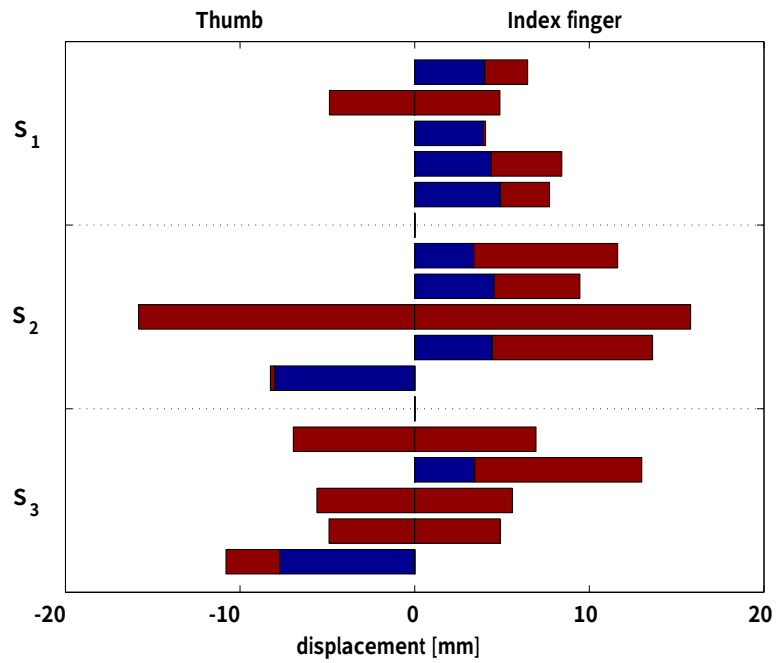
Figure 5.3: Average of displacement distance of the markers



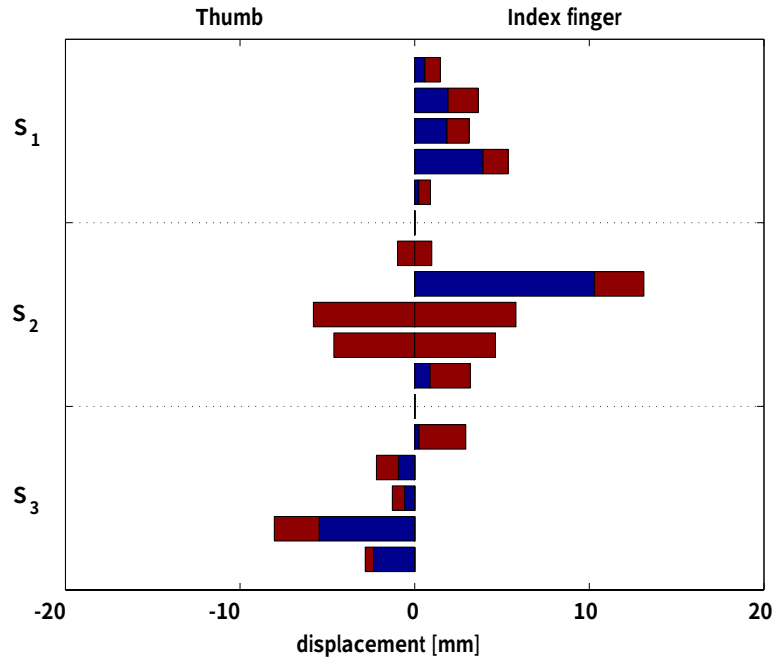
(a)Trajectory of the object markers when the index finger touched first

(b)Trajectory of the object markers when the thumb touched first

Figure 5.4: Trajectory of the markers put on the target object



(a)Quick motion



(b)Slow motion

Figure 5.5: displacement of the target object of 30g weight. The results in the case when the thumb contacted first are on the left side, and the index finger first on the right side. The blue bars are the displacement during the first contact, and the red ones the displacement during the lift-up motion

5.3 Contact force in human grasping

5.3.1 Experimental set-up

The same system arrangement as in Figure 5.6 was used for these experiments.

The target object of Figure 5.7 was used in order to measure the contact force and the displacement. Subjects pinched the object with the index finger and the thumb at the part A and B of the object. The force applied by the index finger and the thumb, F_{index} and F_{thumb}, were measured using two pairs of strain gauges. The weight of this object was $230g$. Fiducial marks were also put on the upper face of the object to measure the displacement with a digital video camera.

Three load cells were set under the target object. The force applied to the desk while the holding the object was measured with these load cells. Forces applied to the object and the table were recorded in the computer through the gauge amplifier with a $5ms$ sampling interval.

The experiment was made by five subjects(S_1 to S_5), who were all male students in their twenties. Practices were carried out several times in order to accustom the subject to the handling of the target object.

5.3.2 Results and discussion

The typical time trajectory of the measured forces is shown in Figure 5.8. First, one of the fingertips touched the target object at time $t = 0$. Then, once the two fingertips touched the object, their contact force F_{index} and F_{thumb} increased simultaneously. Then, the target object lost the contact with the table at $t = t_u$ and this is the end of the grasping motion. At the time instant of contact, the force which was applied to the table through the object F_{floor} changed, but the change was quite small. It was confirmed that a human could grasp the target object flexibly without pressing the table.

The enlarged view on the contact force is shown in Figure 5.9. It is found from this figure that one of the fingertips has touched the target object earlier than the other, and there is a difference in the contact time. We call it *first contact* from now on. Hence, we will focus here after on the force applied just after the first contact.

Figure 5.10 indicates the average of the first contact force and its standard deviations. The contact force of the fingertip during grasping (until the other fingertip touches the object, from t_1 to t_2 in Figure 5.9) was averaged to every subject, in

time and trials. independently of the fingertip which reached the object first, the first contact force is about 0.1 to $0.2N$. It can be said that a human can touch the object with very small contact forces. The standard deviation shows the variation in first contact force. All the subjects had larger variations with the thumb than with the index finger. So the index finger can apply the contact force more stably than the thumb.

The displacement of the target object from the initial state with no contact are shown in Table 5.3.2. Their values are averaged on five experiments. The displacement of the target object was less than $3mm$. Subjects touched it without large displacement.

Subject	S_1	S_2	S_3	S_4	S_5
Displacement[mm]	1.8	3.1	1.9	2.5	2.7

Table 5.2: Object displacements due to finger grasping

Figure 5.11 shows the time derivative of the first contact force. The grasping motion can be classified into three patterns; (a) (b) and (c) in Figure 5.11. Their characteristics are described as following;

Pattern I : While the first contact force is increasing without moving the target object, the other fingertip touches it.

Pattern II : The contact force of both fingertips increase simultaneously.

Pattern III: While the first contact force is oscillating (the time derivative of the first contact force is zero or negative) from the contact time instant, the other fingertip touches the target object.

Figure 5.12 indicates the occurrence of these patterns for ten trials by each of the five subjects. The right and left half plane in the figure are the frequency of Patterns I, II, and III according to the finger making the first contact, i.e. the index finger and the thumb, respectively. In the case of a grasping task by three fingers, results on the frequency of first contact by the middle finger, the thumb, and the index finger is reported in [14]. It has also been reported that there is a difference between subjects [37]. Our measurement data had also some bias with respect to each human subject.

Pattern II occurred less than the other two, and appeared when both fingertips contacted almost at the same time. Pattern I has appeared only when the index finger touched the object first, therefore the contact force by the index finger tends to increase without moving the target object, before the thumb touches it. For pattern III, it is often in the case when the thumb made the first contact, so the contact force tends to maintain, until the index finger touches the object. Based on these results, it can be said that the characteristics of the first contact force were different depending on the fingertip which contacted the target object first. The number of subjects was five, and each of them did the grasping motions ten times.

Another experiment was carried out in order to confirm the above tendency. The subjects were asked to touch the object first with the index finger or the thumb. The number of subjects was five, and each of them made ten trials. Figure 5.14 shows the typical contact force data of the experiments. From these figures, the trend discussed in Figure 5.12 can be confirmed. When the index finger touched the object first (Figure 5.14 (a)), its contact force increased before the thumb arrived. On the other hand, when the thumb touched the object first (Figure 5.14 (b)), its contact force increased but was not stable. Figure 5.13 shows the occurrence of patterns I, II, and III for these experiments. When the designated finger made first contact consciously, the same trend with unconscious motions has appeared. Pattern I obviously occurred more frequently with the first contact by the index finger, while Pattern III with the thumb, similarly to the unconscious grasping motion. The subject S_5 who did not make the first contact with the index finger as shown in Figure 5.12, usually showed pattern I when he was asked to touch with the index finger first.

From these results, it can be concluded that the grasping movement of the fingers are related strongly to the first contact force characteristics.

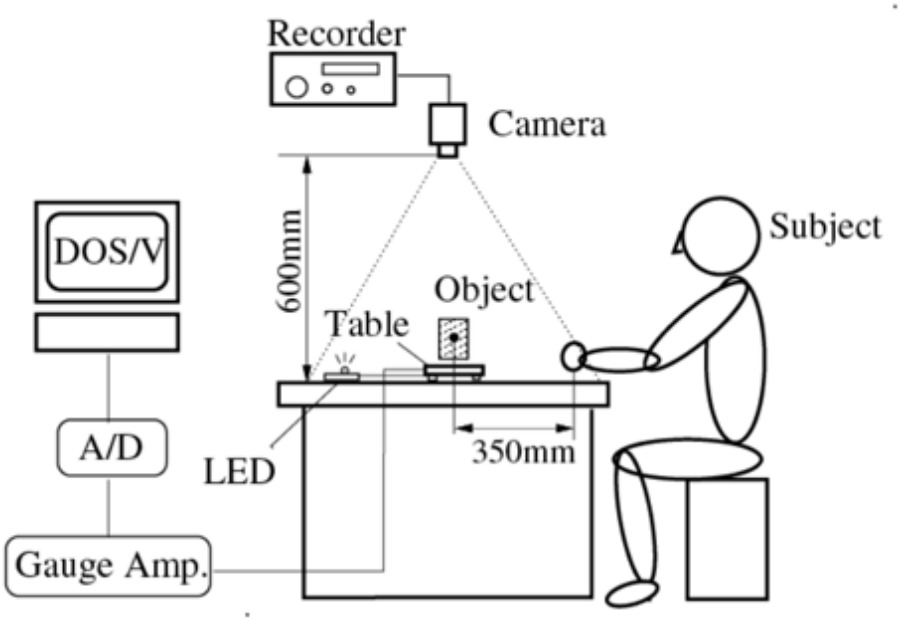


Figure 5.6: System arrangement for the experimental set-up

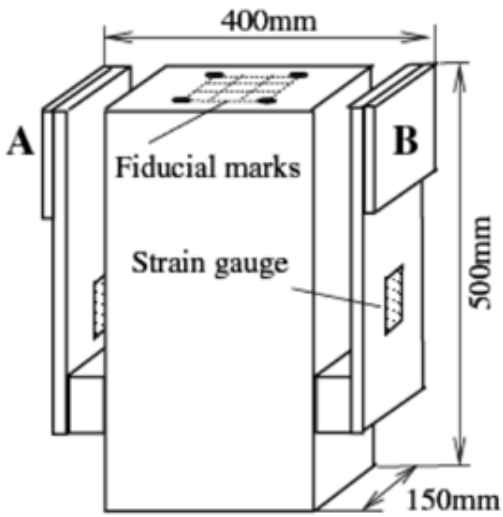


Figure 5.7: Experimental object and markers

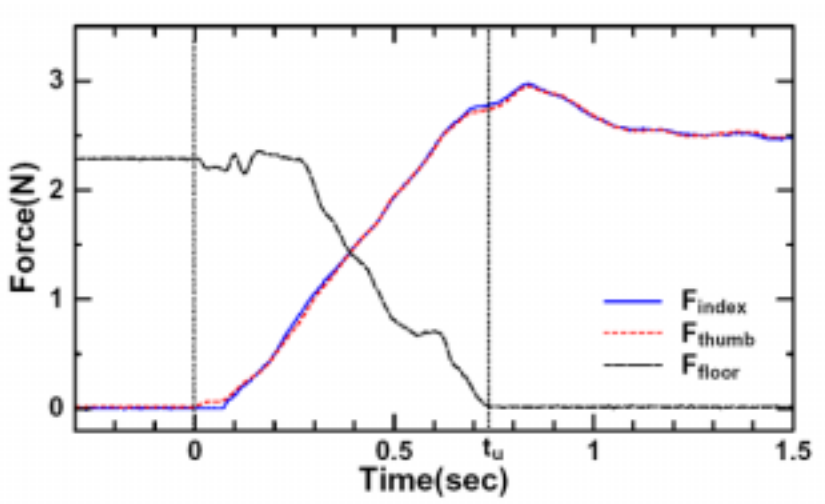


Figure 5.8: Typical time trajectories in picking-up motion

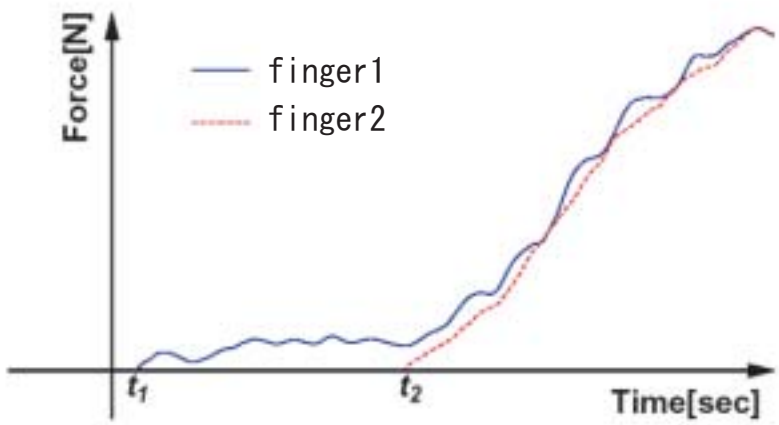


Figure 5.9: Close-up of time trajectories of contact force by the two fingers

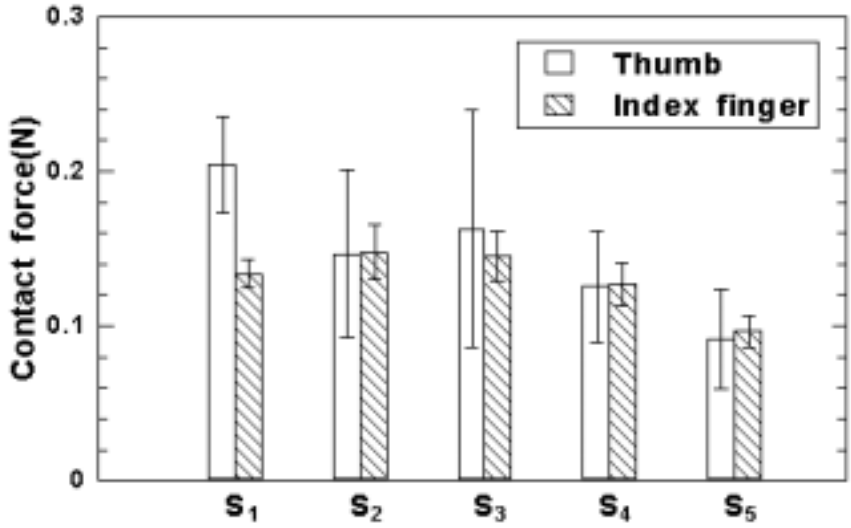


Figure 5.10: Average of contact force and its standard deviation

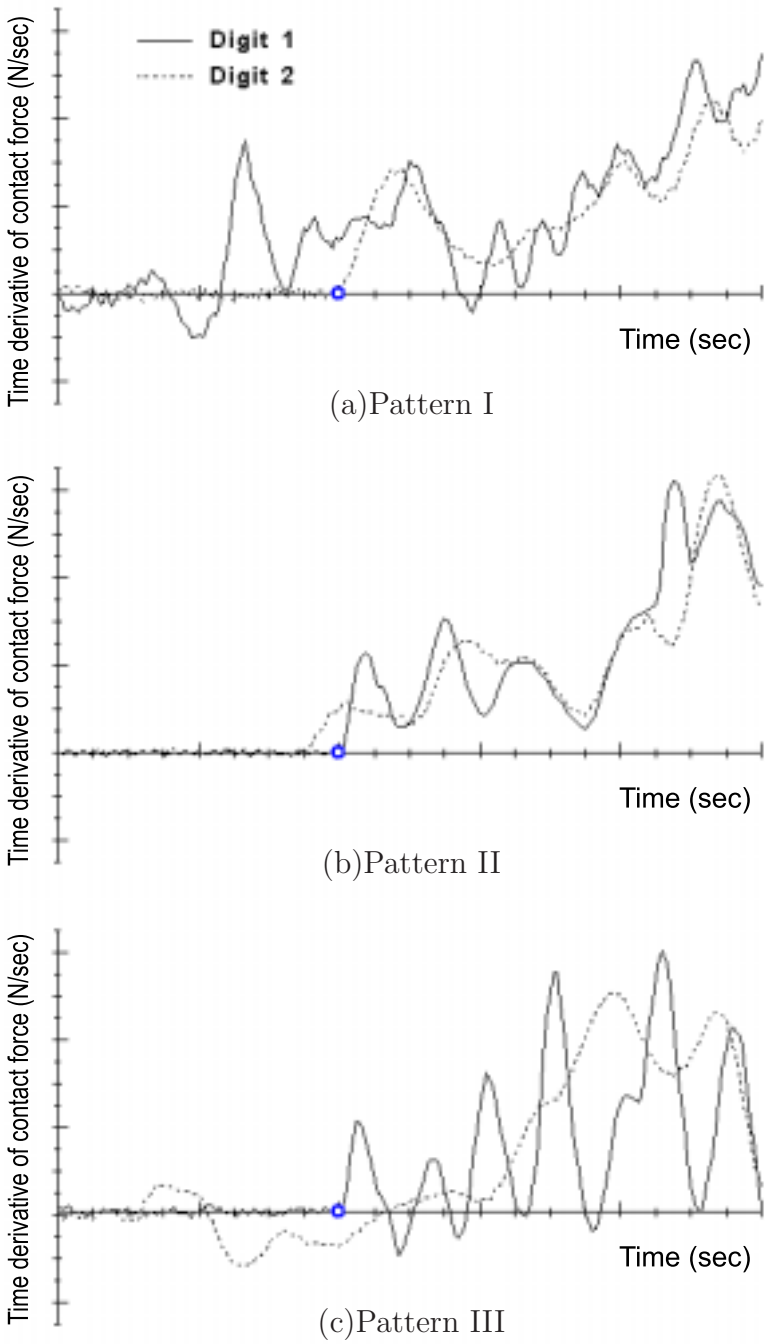


Figure 5.11: Typical patterns of time derivative of the contact force

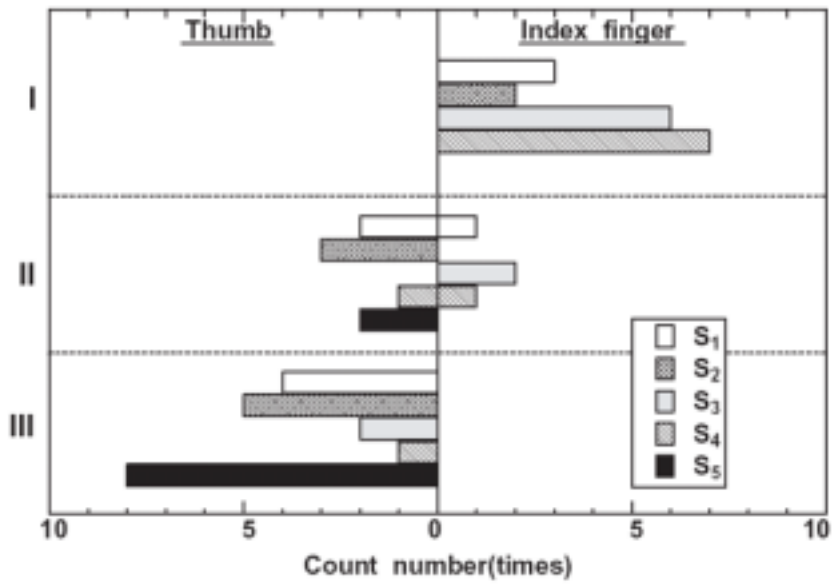


Figure 5.12: Occurrence of a grasping speed patterns in picking up motions

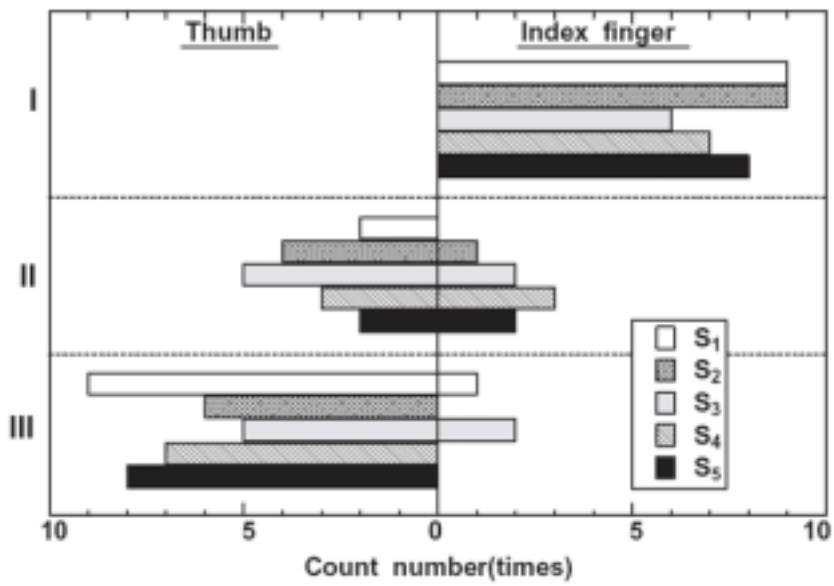
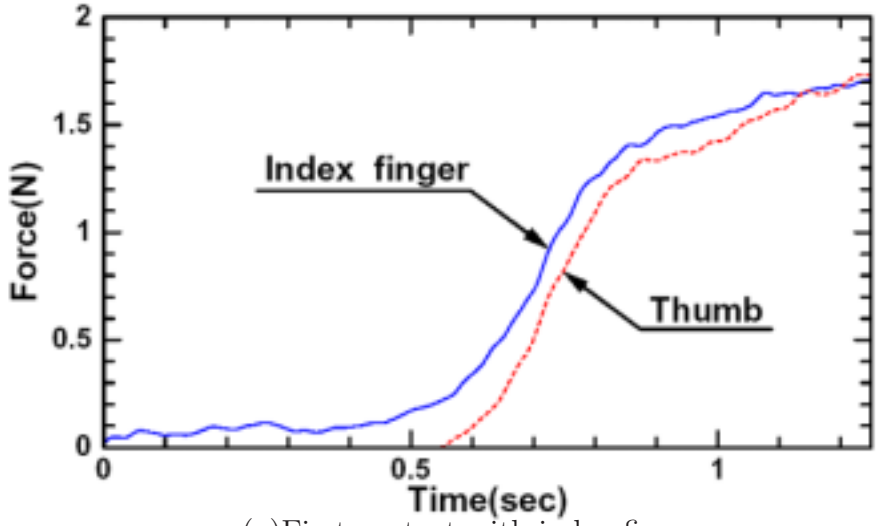
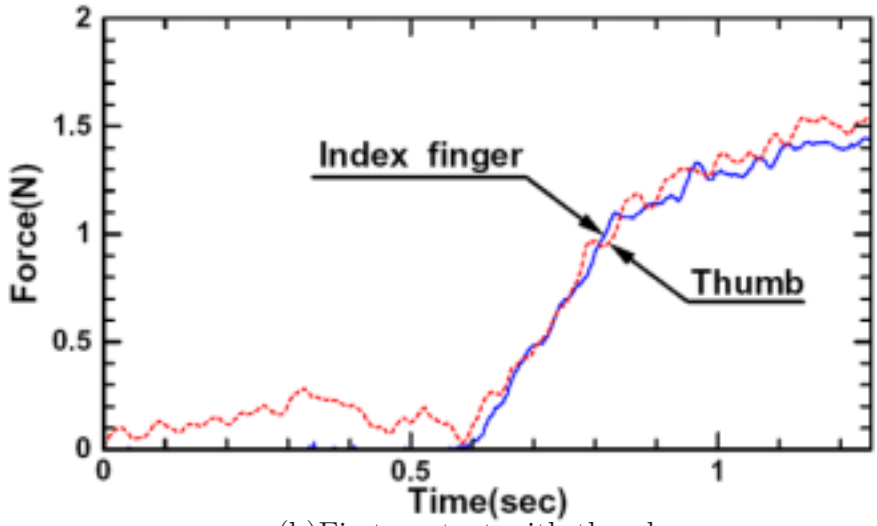


Figure 5.13: Occurrence of a time derivative of the contact force patterns when the first contact finger is designated



(a)First contact with index finger



(b)First contact with thumb

Figure 5.14: Typical time trajectories of contact force in case the first contact is made by the thumb or the index finger

5.4 Motion of human fingertips and their functions in grasping

5.4.1 Experimental set-up

The motion of the fingertips was measured. Experiments have been done using a cylindrical object whose width is deformable due to the finger force as shown in Figure 5.15. The object has a spring inside ($15g$ weight, $55mm$ width, $12mm$ side diameter, $0.23N/mm$ spring coefficient, $14mm$ deformable width), and the object width is shortened proportionally to the applied force. Since the target object is light, it is possible to hold it without shortening its width at the beginning of the experiment. A subject pinched it with the index finger and the thumb. After the stable holding condition was confirmed, the subject was asked to apply then reduce the force to the target object. Markers were put on the 5 finger joints and on the 2 fingertips. The movement of the fingertips has been measured using the digital video camera. At first, visual information was restricted for the subjects in order to avoid its effect, and the grasping action was done only with the sense of the fingertips. Then, the measurements were also carried out the visual information. Subjects were five male subjects(S_1 to S_5) in their twenties.

5.4.2 Results and discussion

Figure 5.16 shows a typical motion of the fingertips obtained by the experimental measurement. The polygonal lines connect the joints, thus they represent the posture of the hand at every time instant. It shows the motion of the fingertips applying the force in the direction indicated by the arrows. From the figure, it was found that the thumb hardly moved, and that the index finger largely moved.

The displacement magnitude of the markers on the fingers is shown in Figure 5.17 and Figure 5.18. This is the ratio of the displacement of the index finger and the thumb for a complete displacement of the target object. The result of the experiment without visual information in Figure 5.17 tells us that the displacement of the index finger was clearly larger than the one of the thumb. The same trend was obtained for the experiments with visual information as in Figure 5.18. The subject S_2 moved the thumb comparatively when he applied the force, but the ratio was still 40%. It seems reasonable to suppose that the index finger applies the force actively and that

the thumb hardly moves to support it. Next, the fine motions of each joint were investigated. The back of the finger was considered as a straight line, and the joint angles were calculated by the gradient of the lines as explained in Figure 5.19. The calculated results are presented in Figure 5.20 and Figure 5.21. The vertical line in the figure is the range of each joint angle that moved from initial state to the final state where the action was completed. It is positive in the direction indicated by the arrows in Figure 5.19. It can be said that the index finger moved more than the thumb, especially the joint angles θ_2 and θ_3. Paying attention to the direction of the joints motions, these θ_2 and θ_3 of the index finger and θ_4 of the thumb moved in the positive direction when the force was applied (Figure 5.20 (a) and Figure 5.21 (a)), and negative motions when the force was reduced (Figure 5.20 (b) and Figure 5.21 (b)). Conversely, the direction of motion of θ_1 and θ_5 for both fingertips were different depending on the subject and the trial. It can be explained by the fact that the joints of both fingertips have passive motions when they moved in the opposite direction of the other three joints, and that they have active motions when it moved in the same direction. With respect to the rotating stiffness of the finger joints, it has been reported in [55] that the rigidity of the joint of the fingertips is low. From this fact, it is likely that the passive motions of the fingertips is important for the posture of the fingers when applying the force and holding the object.

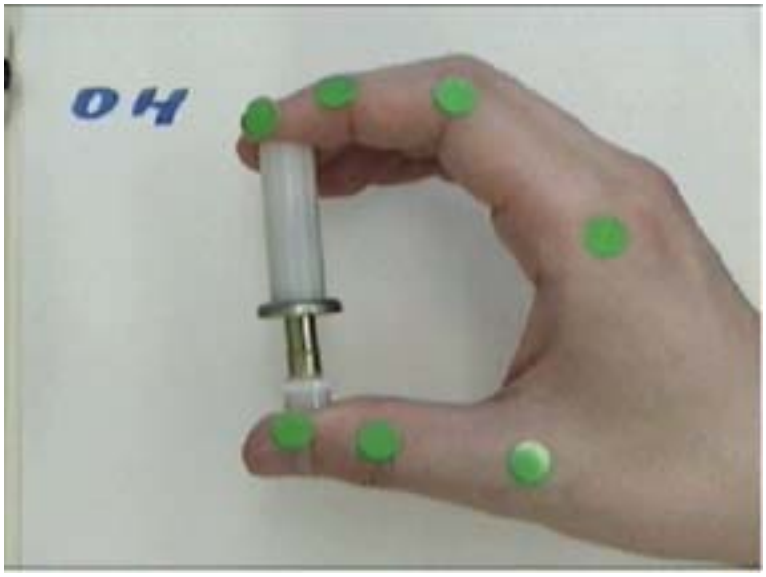


Figure 5.15: Experiment for grasping a cylindrical object whose width is deformable due to the finger force

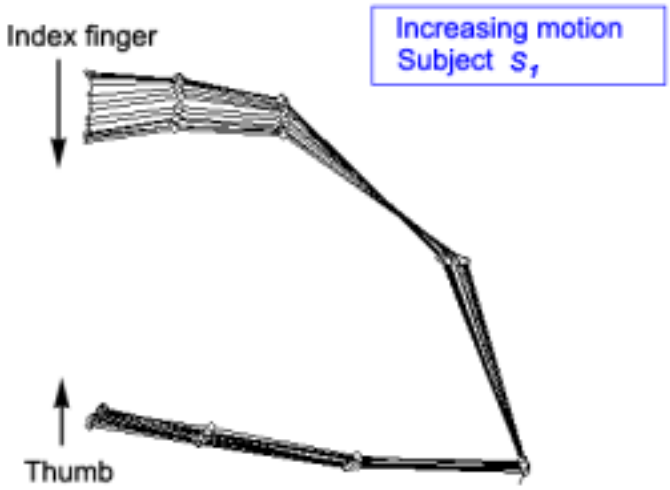
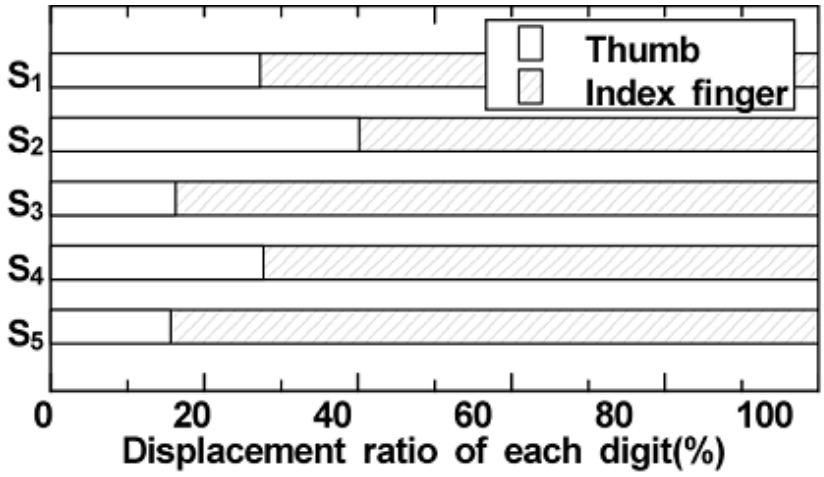
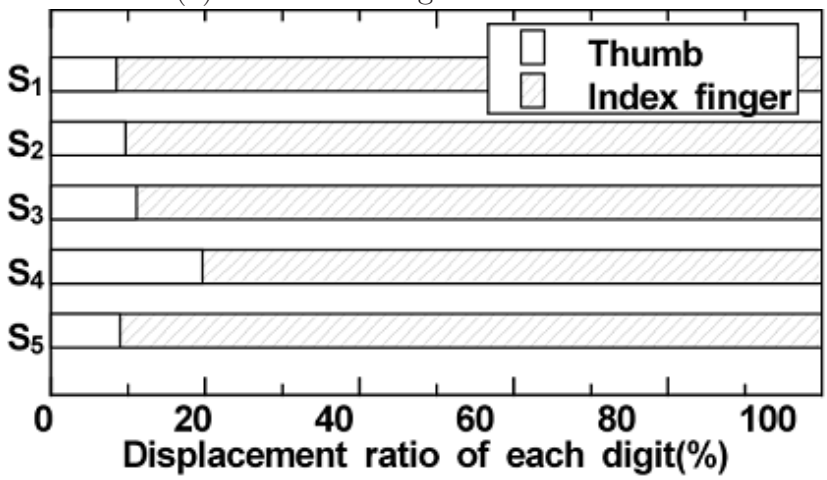


Figure 5.16: Typical trajectories of the fingers when the subject gives grasping force to an object whose width may be deformable

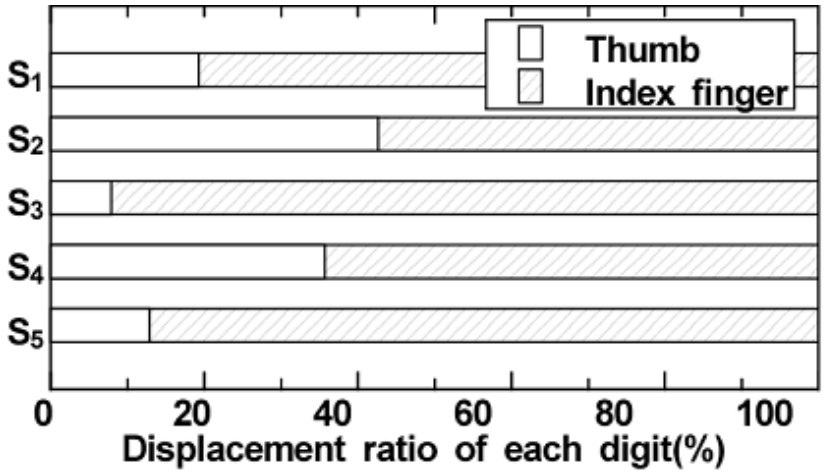


(a)Force-increasing motions without vision

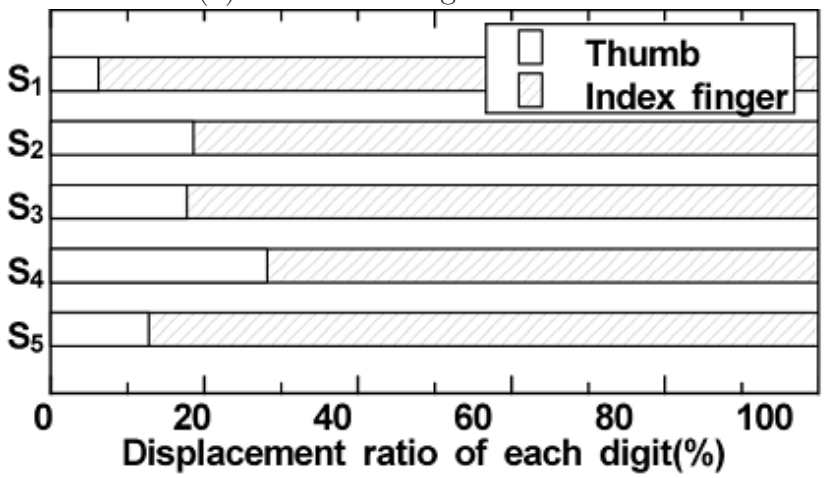


(b)Force-decreasing motions without vision

Figure 5.17: Ratio of displacement between the two fingers without vision



(a)Force-increasing motions with vision



(b)Force-decreasing motions with vision

Figure 5.18: Ratio of displacement between the two fingers with vision

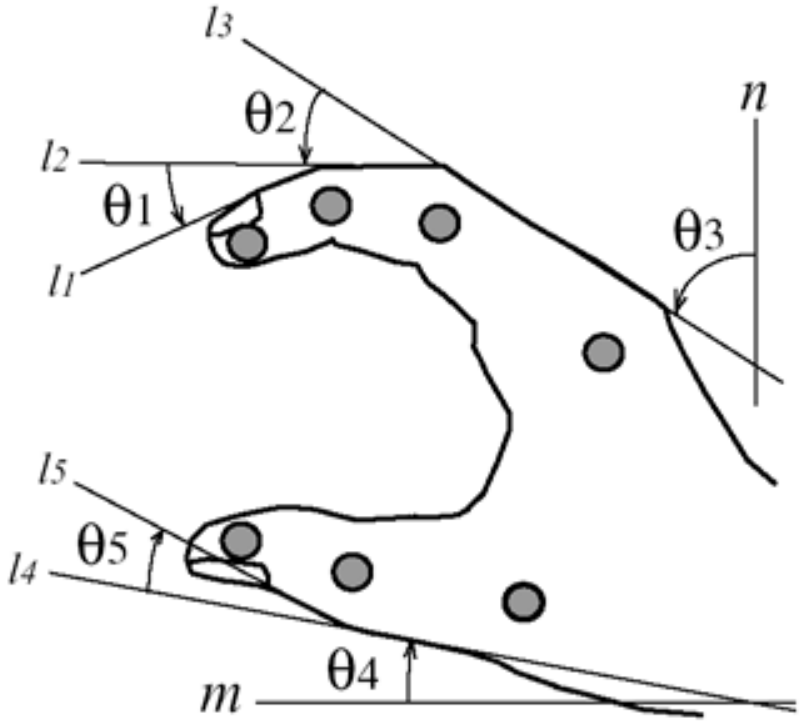
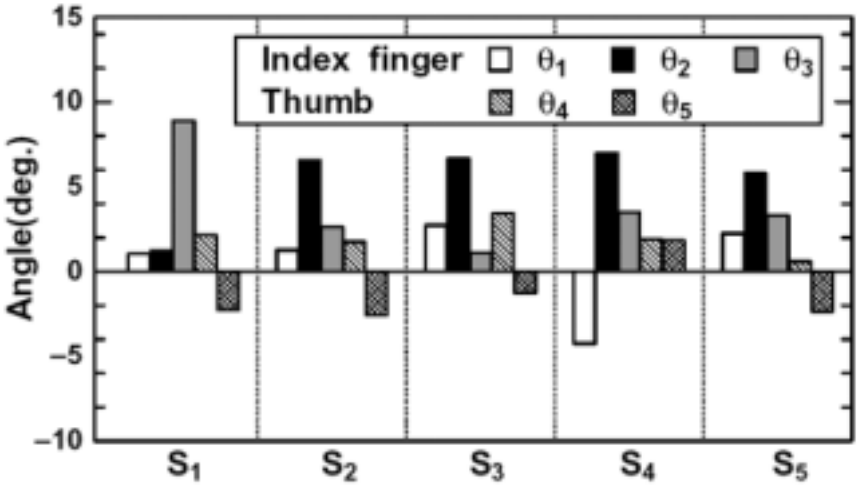
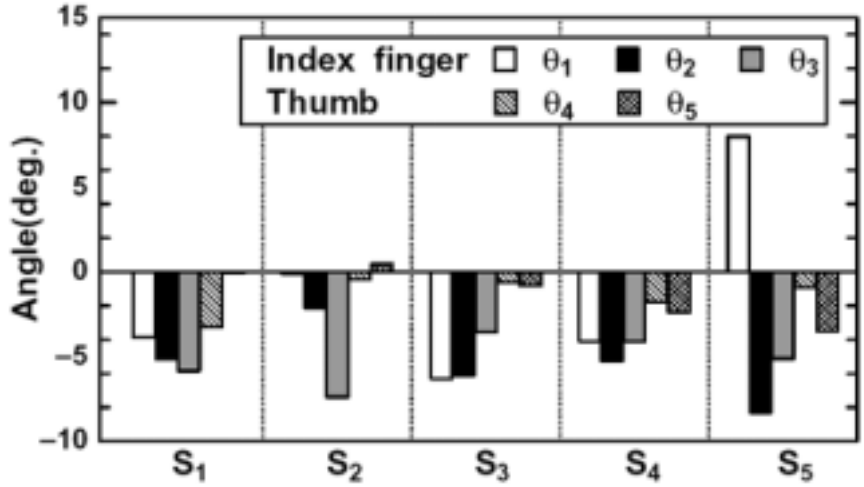


Figure 5.19: Average of contact force and its standard deviations

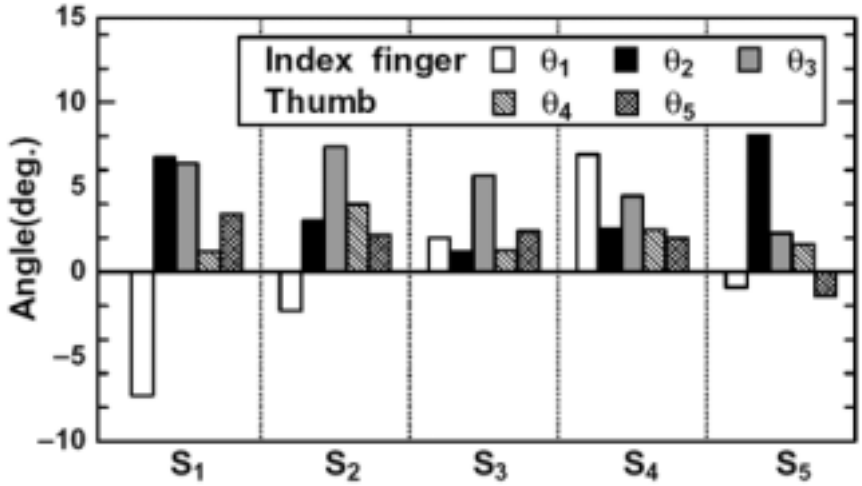


(a)Force-increasing motions without vision

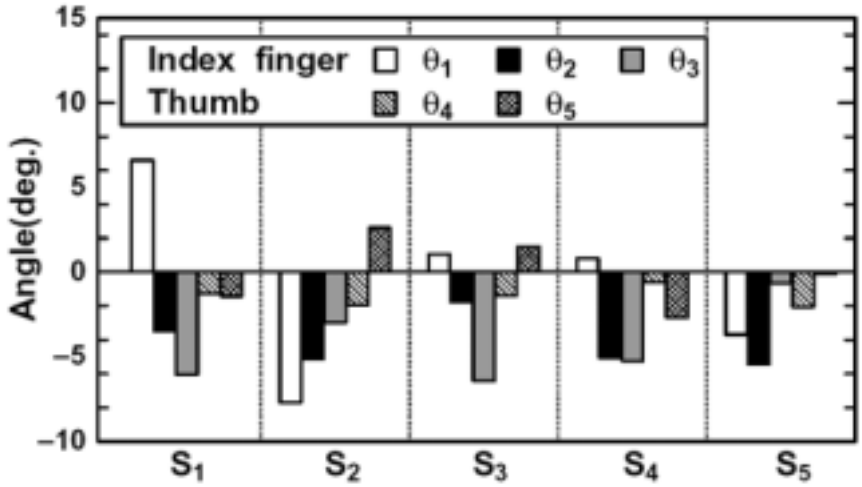


(b)Force-decreasing motions without vision

Figure 5.20: Angle variation of each links without vision



(a)Force-increasing motions with vision



(b)Force-decreasing motions with vision

Figure 5.21: Angle variation of each links with vision

5.5 Conclusion

In this chapter, the contact force and the displacements of the object have been analyzed. Motion of human fingertips has been investigated for objects with deformable width.

The motion of the markers on the target object has been observed by a video camera. Human slightly moves the target object, but knock it down. When the thumb touched the object first, the markers were moved more than during the lift-up phase.

The contact force has been measured paying attention to the fingertip which touched the target object first. The time derivative of the first contact force has been classified into three patterns, and the patterns had a close relationship with the fingertip which made the first contact. When the index finger touched the object first, the contact force stably increased before the thumb arrived. On the other hand, when the thumb touched the object first, its contact force was not stable.

The motion of the finger has been measured using a deformable object. When the force was applied and reduced, the index finger moved largely and the thumb hardly moved regardless of visual information. The index finger applied the force actively, and the thumb supported it.

These features of human grasping motion are related to each other, and reflect grasping motions which do not move the target object.

Chapter 6

Application to robot hand grasping

6.1 Introduction

Huge number of studies on grasping and manipulation by robot hands have been done by many researchers. Salisbury and Craig [70] defined the internal force whose resultant is zero as the force that can be added when necessary to the manipulation force to prevent sliding or breaking contact. Yoshikawa and Nagai [79] proposed a new definition of grasping and manipulating forces for multifingered robot hands. Even though the grasping, manipulation and interval forces are well understood, methods for determination of these forces are still inadequate for real-time implementation, mainly because of their complexity.

Another approach to the control of robot hands is a physiological approach motivated by the study on human hands. For an approach task in order to grasp a target object, Nakazawa observed that the path of two human fingertips is similar to the obstacle avoidance against the target object, and proposed a path-planning model build with a complex potential field [58]. When a human handles and carries a target object, the grasping force is delicately adjusted to the inertial force created by its movement [34], and this characteristic is applied as a control law in the grasping task of the robot hand in [57]. Furthermore, a hand-over motion from human to a robot has been realized in [36], emulating human's hand-over motion.

Human have an ability to touch an object without large displacement even if it is light and could easily fall. It is not applied yet to control laws of a robot hand.

Our aim is to apply the model of human's behavior to a robot hand.

We assume a robot hand with two fingers controlled individually. Each finger has two pairs of strain gauges as force sensors. A vision system is also available with a camera on the end-effector of the manipulator.

Human can keep the first contact force very small. It contributes to a "soft" grasping without rolling over the target object. The first contact force is always almost the same and does not depend on the shape or weight of the target object. Not to move the target object while waiting for the other finger to arrive, the characteristic of human grasping motion is applied to robot hand control.

However, human fingers slightly move the target object during the first contact. The displacement of the target object can be neglected when the time-lag between the contacts by both fingers is very small. The two fingers of a robot hand do not always have actuators with the same performance. When the characteristics of the motors are different, they cannot always contact the target object with a small time-lag between both fingertips, even if the object is exactly in the middle of the distance between both fingers after the initial positioning phase. Therefore, position synchronization is required to control the position and the velocity of two fingers so that they can touch a target object at the same time and with the same velocity. Visual feedback is used for this synchronization.

After both fingers touch the target object, the position and the force are controlled simultaneously. The position synchronization is also used, and the force error and position error are added with different weighting factors for the feedback to the control of the fingers.

The remaining of this chapter is structured as follows. Firstly, maintenance of the first contact force is proposed as an application of the human grasping features. Strain gauges are used to detect the contact force, and experiments show the efficiency. For reduction of the impact force arising from the first contact, several materials are attached on the fingertips and tested. Then, the fundamentals of the synchronization method are presented. The synchronization is used for control of two fingers in order to touch a target object simultaneously. Simulations with synchronization are carried out. The simulation results prove the efficiency of the method. Finally, an experiment of total grasping is demonstrated on an experimental set-up with position synchronization through visual feedback and force control.

6.2 Control of the first contact force

6.2.1 Detection of touch and the first contact force

Contact time instants of two fingers may have small time-lag even with the position synchronization if the resolution of the vision system is not high. It is undesirable that the finger oscillates between contacting and no-contacting conditions due to measurement noise, or that it moves the target object with the large first contact force.

Human keeps the first contact force small. It contributes to a "soft" grasping without rolling over the target object. The human subjects exerted forces around 0.1 to $0.2N$, as found in the measurement of human grasping as shown in Figure 5.10. This force does not depend on the shape and weight of the target object.

We apply this feature to the robot hand control. When one of the fingertips touches the target object, the contact force is kept around $0.16N$, which is the average human force taken from the data obtained by the measurement of human grasping in Chapter 5.

6.2.2 Reduction of impact force by soft attachments

6.2.3 Experiments with a robot hand

Experimental set-up

The grasping demonstration was carried out using a robot hand with two fingers as shown in Figure 6.2. The system arrangement for the experimental set-up is shown in Figure 6.1.

The fingers were made from acrylic plate with $5mm$ in thickness, $2cm$ width, $5cm$ length. Two fingers were moved by DC servo motors through a ball screw, and each finger could be shifted in one direction. The contact forces were measured by strain gauges through a low-pass filter (cut off frequency: 10Hz) and an amplifier.

The threshold of the force to detect the contact was $0.16N$, and the gripping force was $1.6N$. Command voltages of two motors were constant until the fingertip touches the target object. The finger that detects the contact first, stops and keeps the first contact force at $0.16N$. After both fingers contacted, the contact forces are simultaneously increased until $1.6N$. PID controller was used to control the force, and its parameters have been set experimentally. The period was $0.01sec$.

Several materials were attached on the fingertip, and tested for the detection of contact. The target object was an aluminium can of $50g$ on a table.

Results and discussion

Experimental results were obtained by using the robot gripper. Figure 6.3 shows the contact force of two fingers with rubber attachment. The first contact force has been kept at $0.16N$. The strain gauges were sensitive enough to measure the contact force.

The impact force was also detected. Since strain gauges measure the force by the strain on the fingers, it is unavoidable to have a little impact force. It can be a problem if a finger makes the target object slip over. Thus a "soft" attachment was put on the fingertips in order to dissipate the impact force. Figure 6.4 shows the result comparing the force over the threshold for detection of the contact. The results are compared for a fingertip without attachment (acrylic plate), the case of a block of rubber, and of a fingersack filled with gel. The result shows that the impact forces were well reduced by the fingersack of gel. They were also affected by the velocity of the finger motion controlled by the input voltage.

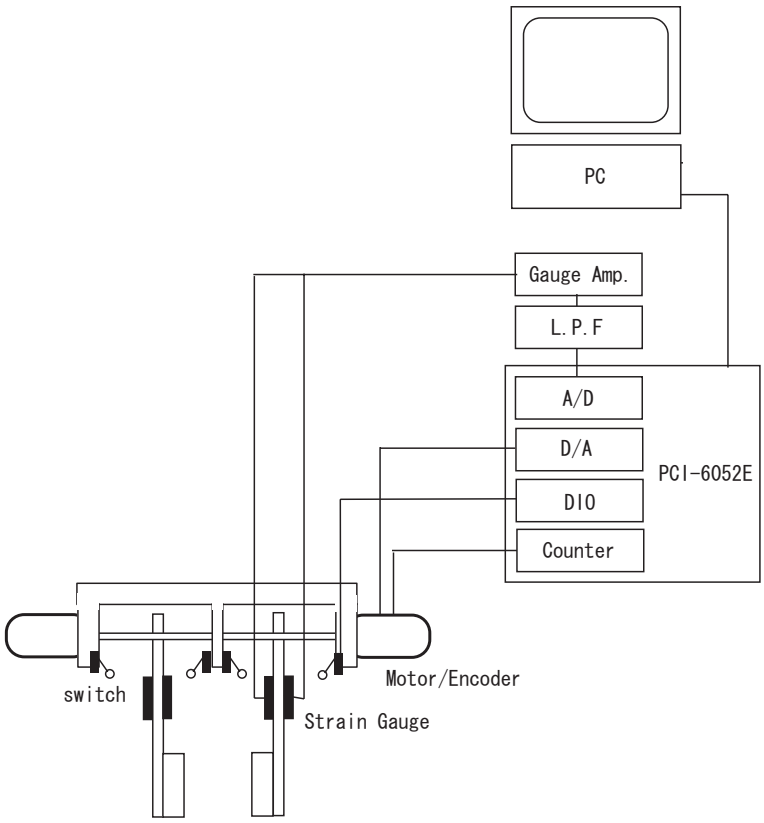


Figure 6.1: Experimental set-up

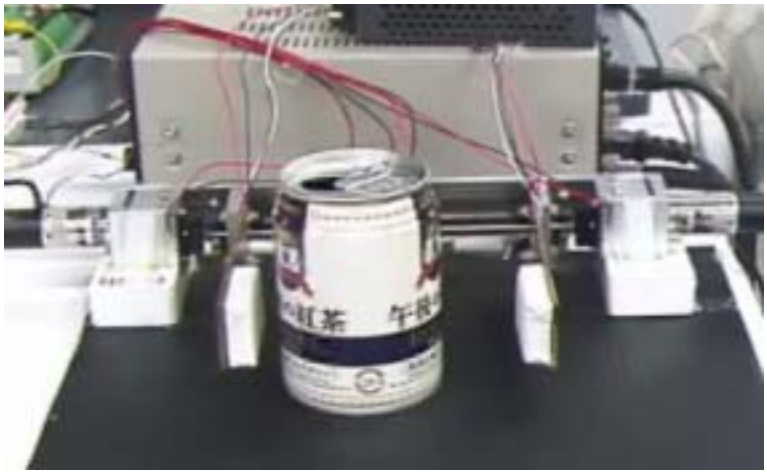


Figure 6.2: Robot hand with two fingertips

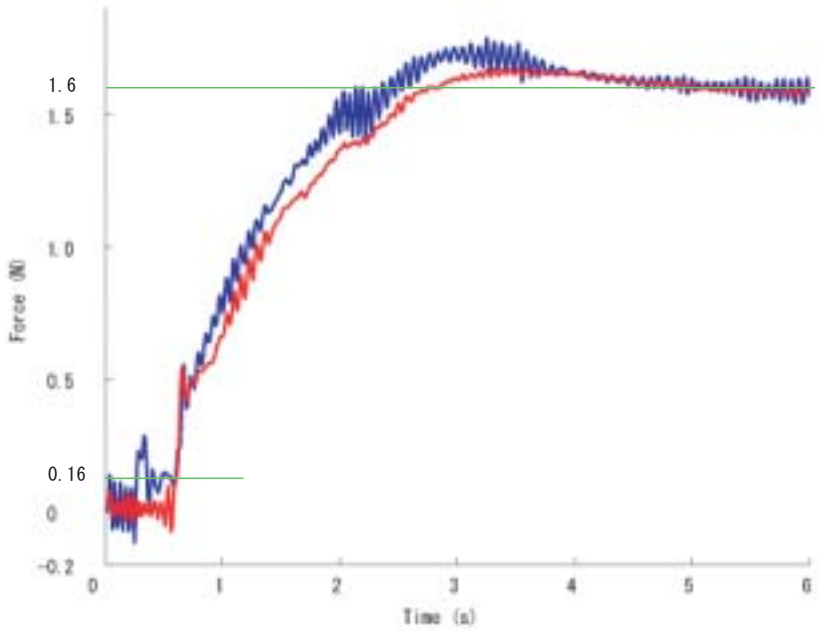


Figure 6.3: Time trajectory of the contact force

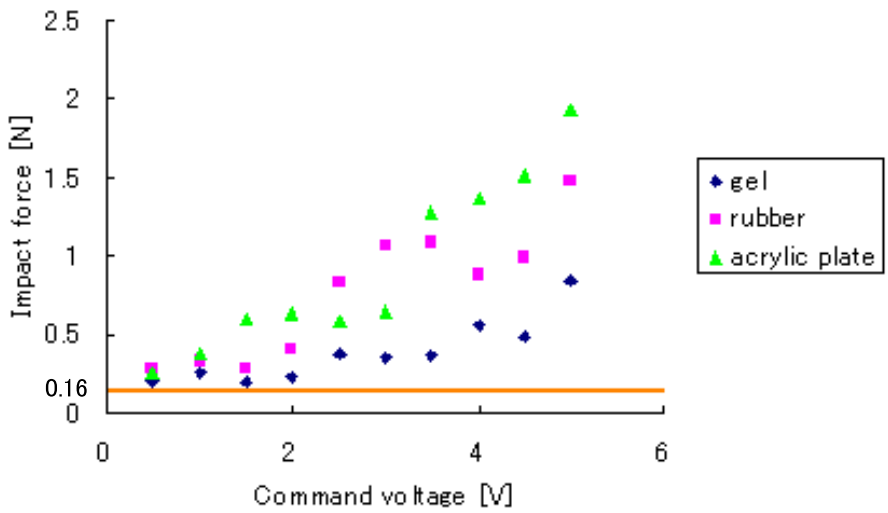


Figure 6.4: Time trajectory of the contact force

6.3 Position synchronization

6.3.1 Position synchronization to touch a target object

Human can position the index finger and the thumb to have almost the same distance between each finger and the target object [54], and thus the two fingertips can touch the object almost at the same time. From the initial positioning task (using the simplex algorithm), the end-effector converged to a position that could be different from the ideal position with an error up to $10mm$. The target object is not always put just in the middle of the fingertips of the robot hand, and thus they cannot touch the target object simultaneously without any motion synchronization. Therefore, position synchronization is applied to control the position and the velocity both fingertips so that they can touch a target object at the same time and with the same velocity.

Consider a motion control system with n axes. Define the position tracking error of the ith axis as

$$d_i(t) \;=\; r_i(t) \;-\; x_i(t) \tag{6.1}$$

where $r_i(t)$ denotes the desired position of the ith axis and $x_i(t)$ is current position. In synchronized motion, besides $d_i(t) = 0$, we would like to regulate motions amongst axes during the approach so that

$$d_1(t) \;=\; d_2(t) \;=\; \ldots \;=\; d_n(t) \quad \forall t \tag{6.2}$$

The goal of maintaining kinematic relationships amongst n axes, as given by Equation (6.2), can be expressed as n sub-goals such as

$$d_1(t) = d_2(t),\; d_2(t) = d_3(t),\; \ldots,\; d_n(t) = d_1(t) \tag{6.3}$$

Then, let synchronization errors of a subset of all possible pairs of two axes from the total of n axes in the following way:

$$\begin{aligned} \epsilon_1^d(t) &= d_1(t) - d_2(t) \\ \epsilon_2^d(t) &= d_2(t) - d_3(t) \\ &\vdots \\ \epsilon_n^d(t) &= d_n(t) - d_1(t) \end{aligned} \tag{6.4}$$

Obviously, if $\epsilon_i^d(t) = 0$ for all $i = 1, \cdots, n$, the goal of multi-axis synchronization of Equation (6.2) is achieved. Here, considering $n = 2$, the distances between each fingertip and the target object d_1 and d_2 in Figure 6.5 are measured. Position synchronization errors as in Equation (6.4) are also defined.

$$\begin{aligned} \epsilon_1^d(t) &= d_1(t) - d_2(t) \\ \epsilon_2^d(t) &= d_2(t) - d_1(t) \end{aligned} \tag{6.5}$$

Velocities of two fingertips should also be the same at the time of contact. Therefore, if $v_1(t)$ and $v_2(t)$ are the velocities of both fingers, then:

$$\begin{aligned} v_1(t) &= \dot{x}_1(t) = \dot{d}_1(t) \\ v_2(t) &= \dot{x}_2(t) = \dot{d}_2(t) \\ \epsilon_1^v(t) &= -\epsilon_2^v(t) = v_1(t) - v_2(t) \end{aligned} \tag{6.6}$$

Accordingly, a total feedback error denoted by $e_s(t)$, is defined. This total error includes the position errors $d_i(t)$, and synchronization errors $\epsilon_i^d(t)$, and the velocity errors $\epsilon_i^v(t)$, i.e.:

$$e_s(t) = \begin{bmatrix} e_{s1}(t) \\ e_{s2}(t) \end{bmatrix} = \begin{bmatrix} d_1(t) + \alpha_d \epsilon_1^d(t) + \alpha_v \epsilon_1^v \\ d_2(t) + \alpha_d \epsilon_2^d(t) + \alpha_v \epsilon_2^v \end{bmatrix} \tag{6.7}$$

where α_d and α_v are positive coupling weights.

6.3.2 Position and force control

After both fingers touched the target object, they increase the contact force until it is sufficient to lift up and hold the object. When only the forces of the fingers are controlled, the target object usually moves since its position is not controlled anymore. It is necessary to have both position and force controlled simultaneously.

Total feedback errors denoted by $e_i(t)$ are defined that contain the position errors $d_i(t)$ and the force errors $f_i(t)$. The position errors $d_i(t)$ are the distance between current positions of the fingertips and the position where the fingertip touched the target object first. The force errors $f_i(t)$ are the difference between current and objective forces.

$$e_f(t) = \begin{bmatrix} e_{f1}(t) \\ e_{f2}(t) \end{bmatrix} = \begin{bmatrix} \beta_d d_1(t) + \beta_f f_1(t) \\ \beta_d d_2(t) + \beta_f f_2(t) \end{bmatrix} \tag{6.8}$$

where β_d and β_f are positive coupling weights.

6.3.3 Simulation

Synchronization

The proposed algorithm has been tested by simulations using Matlab simulink toolbox. The model of the fingertip is shown in Figure 6.7. The robot hand model had two fingers controlled individually. A PID controller was selected and the gains α_d, α_v ware decided experimentally.

Starting positions for the two fingertips d_1 and d_2 were $30mm$ and $10mm$, respectively. The object width was $50mm$.

Simulation results are shown in Figure 6.8. The time trajectories in Figure 6.8 (a) indicate the positions of both fingertips, and Figure 6.8 (b) indicate their velocity. Dotted lines indicate the results without synchronization($\alpha_d = 0,\ \alpha_v = 0$), and continuous lines indicate the results with position and velocity synchronization($\alpha_d = 0.75,\ \alpha_v = 0.1$). Different starting positions are indicated by different colors.

The position and velocity synchronization had the best result. The time difference between contact time instants was smaller than that of human even without synchronization. Nevertheless, the time-lag with synchronization was less than half of the one without synchronization, moreover, the velocity difference was much smaller. Since we assume that the rigidity of the finger of the robot hand is stiffer than the one of a human finger, the effect of synchronization can be important for a grasping task by a robot hand.

Position and force control

The simulation of force and position control has been done. Assuming that the two fingers have been just touched the target object, start conditions were set to $d_1 = 0,\ d_2 = 0$, and $f_1 = f_2 = 0$. The objective force was set to $1[N]$. The coupling weights β_p and β_f have been determined experimentally.

Simulation results are shown in Figure 6.9. The time trajectories in Figure 6.9 (a) indicate the positions of two fingertips, and (b) indicate their force. The colors indicate if position are controlled with the forces or not (red indicates force feedback alone, and blue force and position feedback).

It is clear from the simulation that position and force control had the best result.

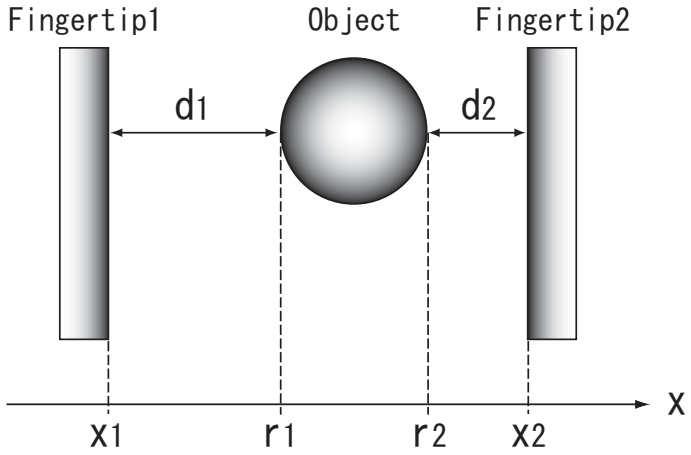


Figure 6.5: Model of the system

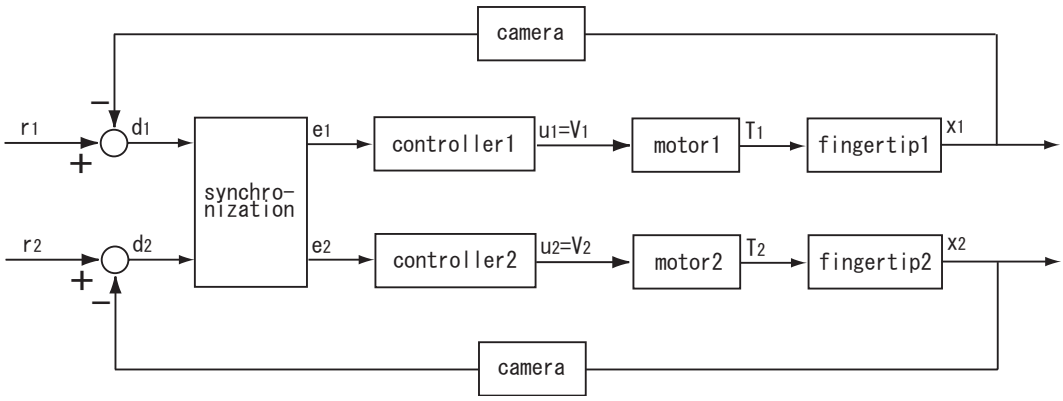


Figure 6.6: Block diagram of position synchronization

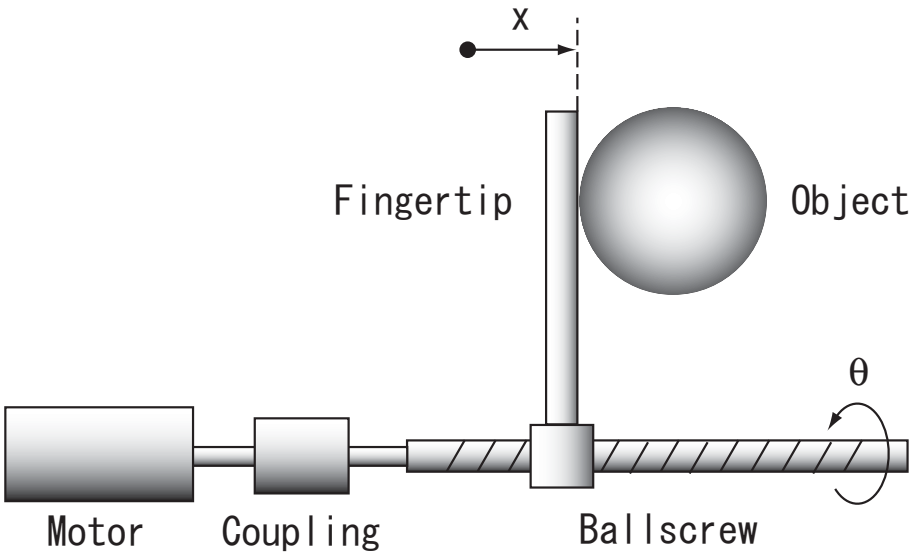
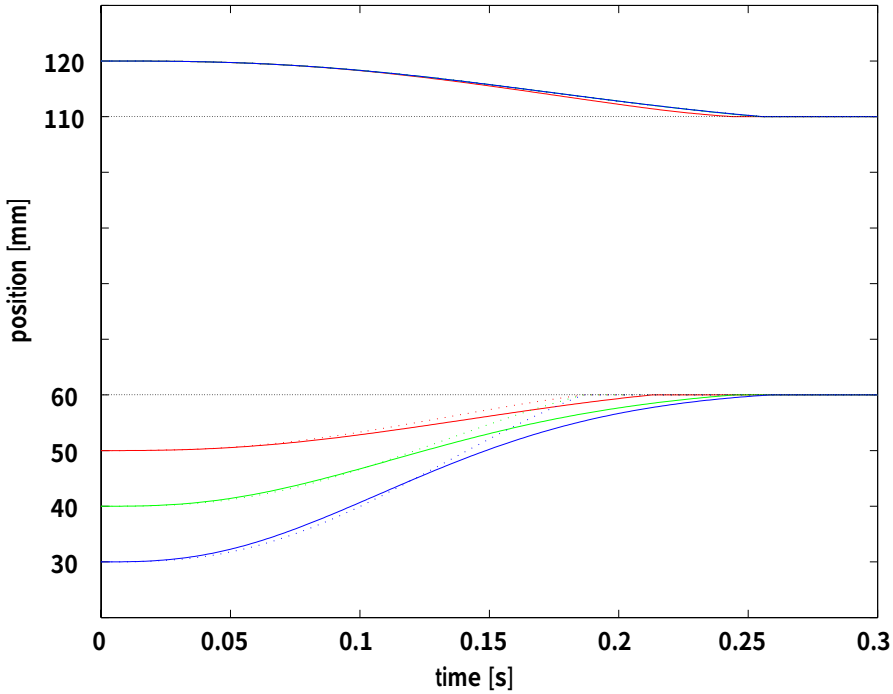
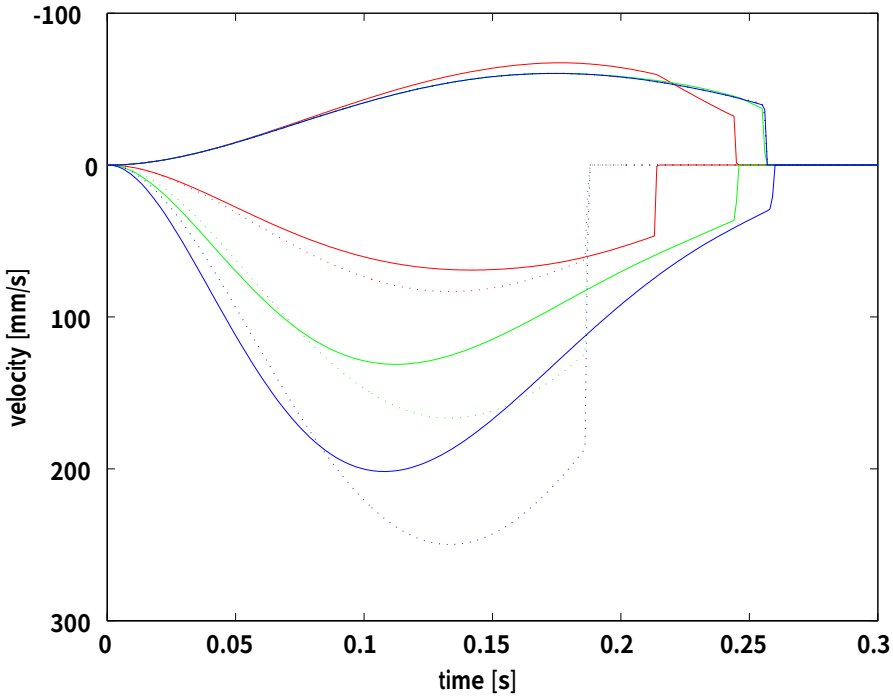


Figure 6.7: The finger model of robot hand with motor, coupling, and ballscrew

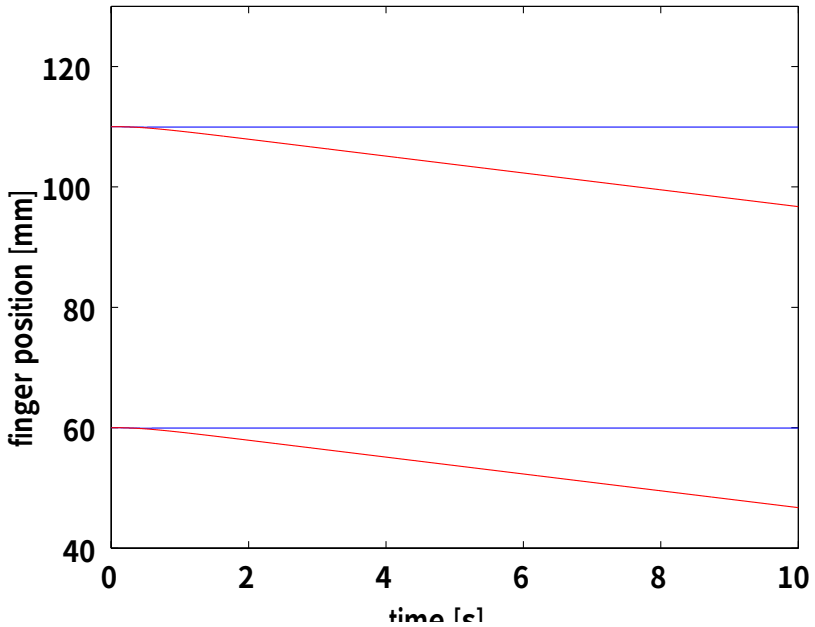


(a)Time trajectories of the finger position

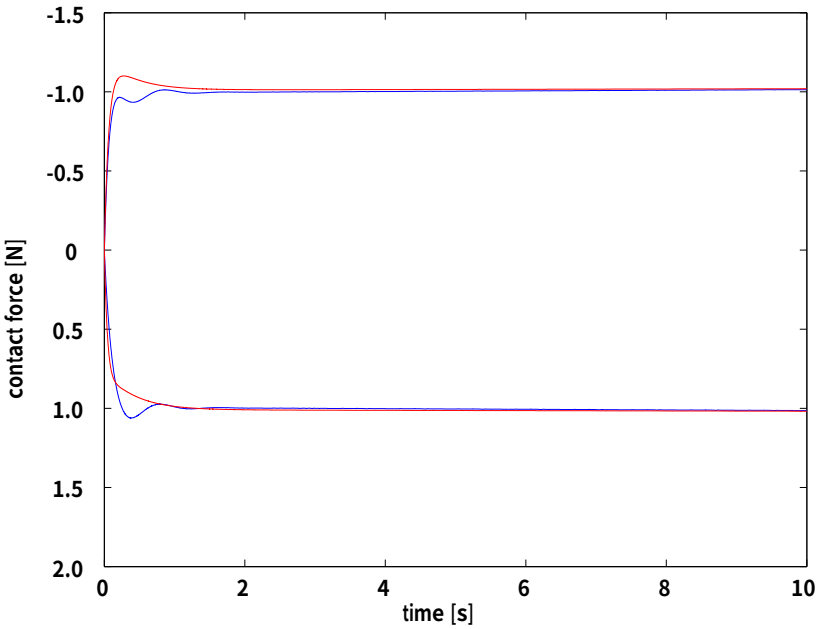


(b)Time trajectories of the finger velocity

Figure 6.8: Simulation results of the synchronization



(a)Time trajectories of the finger position



(b)Time trajectories of the finger force

Figure 6.9: Simulation results of the force and position control

6.3.4 Experiments with a robot hand

Experimental set-up

The system arrangement for the experimental set-up is shown in Figure 6.10. The target object was a plastic cup on the table. A digital video camera was set $600mm$ above the table. The positions of the two fingertips and the target object were taken with a $\frac{1}{25}sec$ sampling interval.

Total grasping demonstration was carried out using a robot hand with two fingers, same as the last section. First, the two fingers approached the target object. When both errors became zero, the gripping phase was started. The objective gripping force was $1.0N$. The distance between the image center of the target object and each fingertip were used to compute the synchronized error, because the shape of the plastic object could be changed by the gripping force.

Result and discussion

Grasping task was successfully demonstrated. Figure 6.12 shows the positions of both fingertips. The two fingers gradually compensated their difference of position errors, and arrived at the target object at the same time with the same trajectories. In the gripping phase, the position of the target object in the image did not move largely as shown in Figure 6.13 (a). Figure 6.13 (b) indicates the contact force of both fingertips detected by the strain gauges. They maintained the contact force without moving the target object.

At the time instant when the second finger touches the object $d_1(t) = d_2(t) = 0$ and the controller switch from the control law (6.3.1) to the control law (**??**). This instantaneous switch does not create bumps in the experiment. However, at high velocities we could use the control law (6.2) between both control laws.

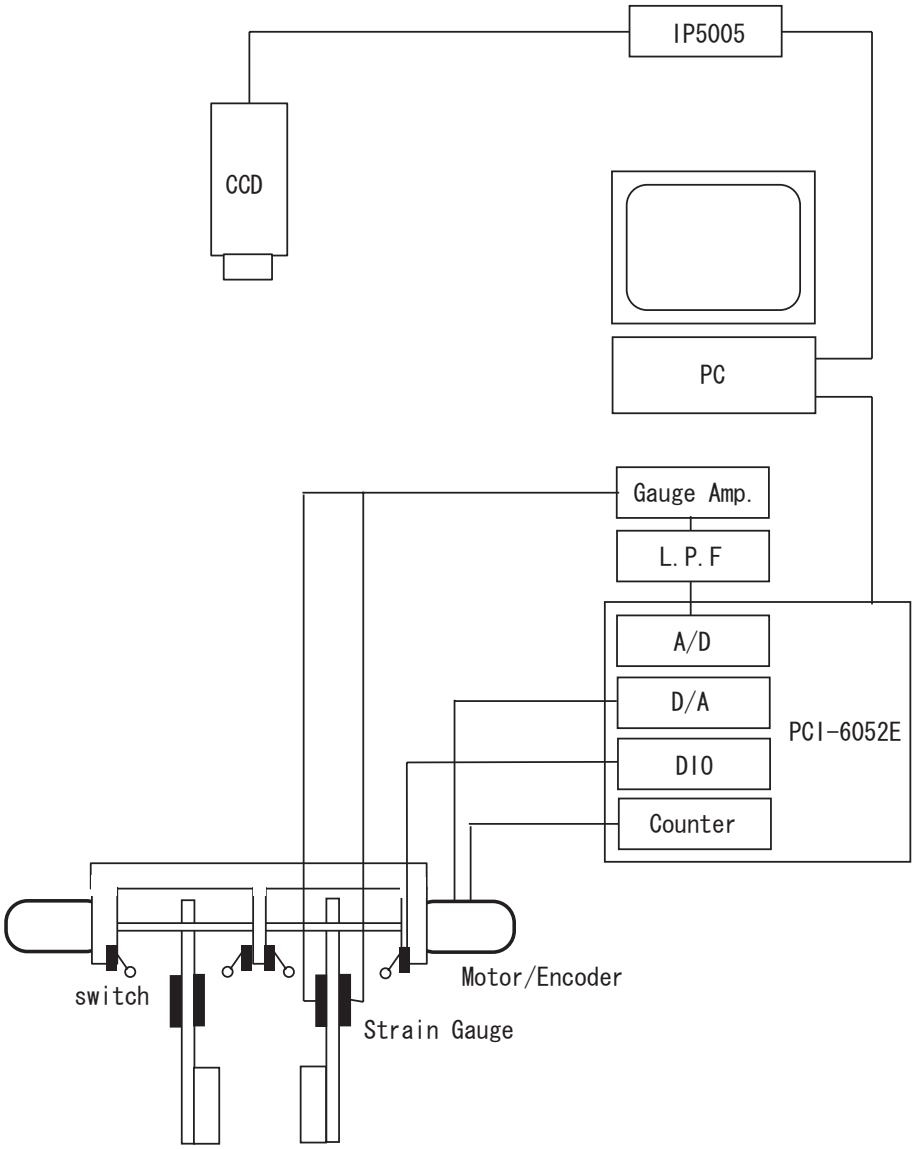


Figure 6.10: Experimental set-up

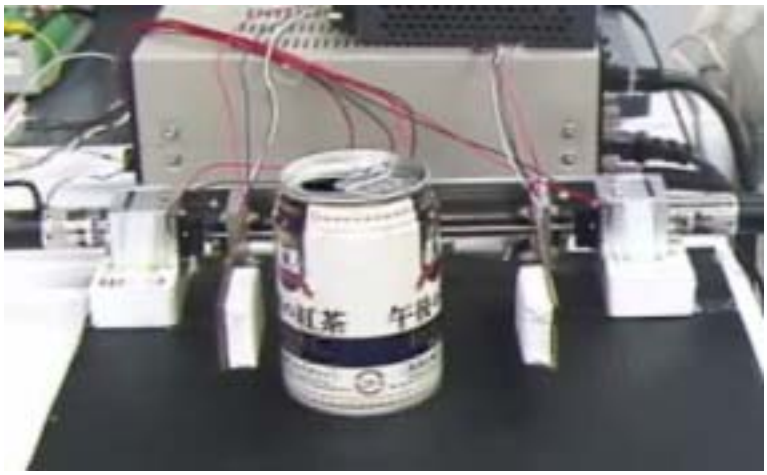


Figure 6.11: Robot hand with two fingertips

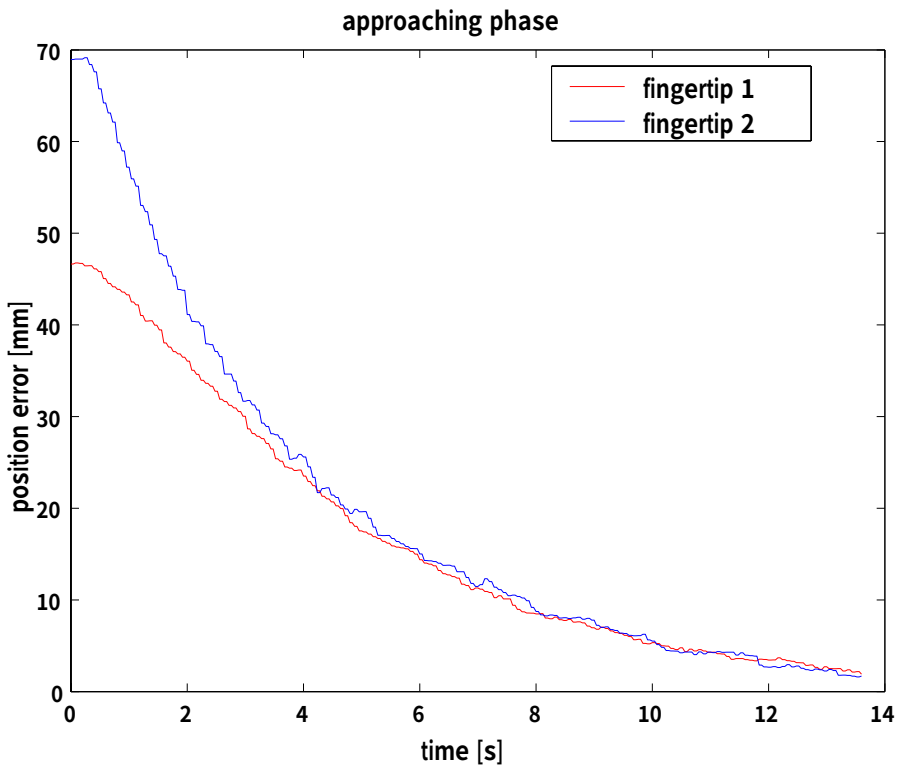
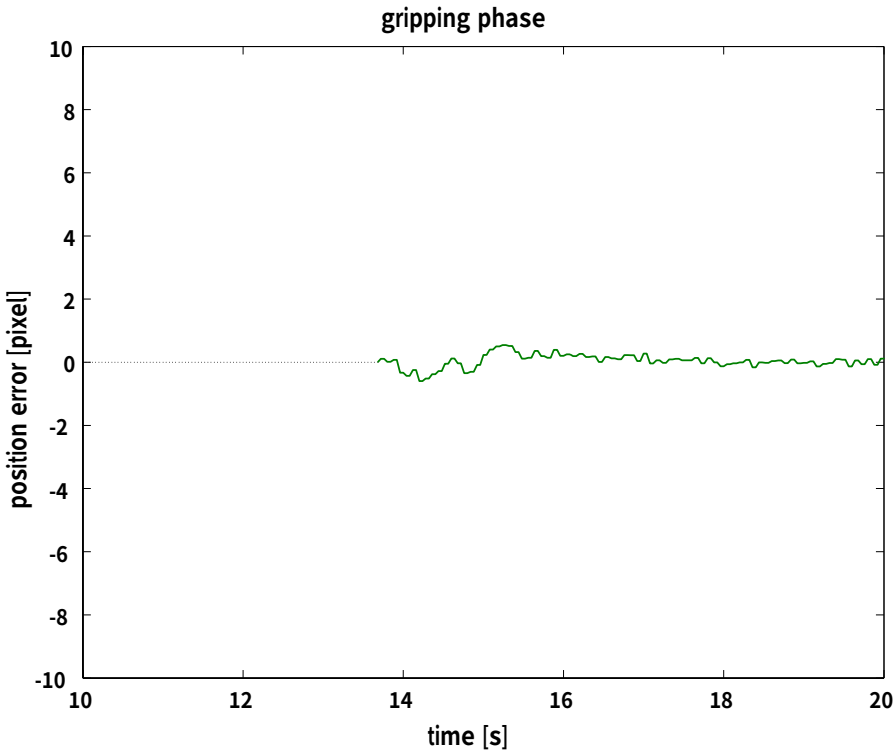
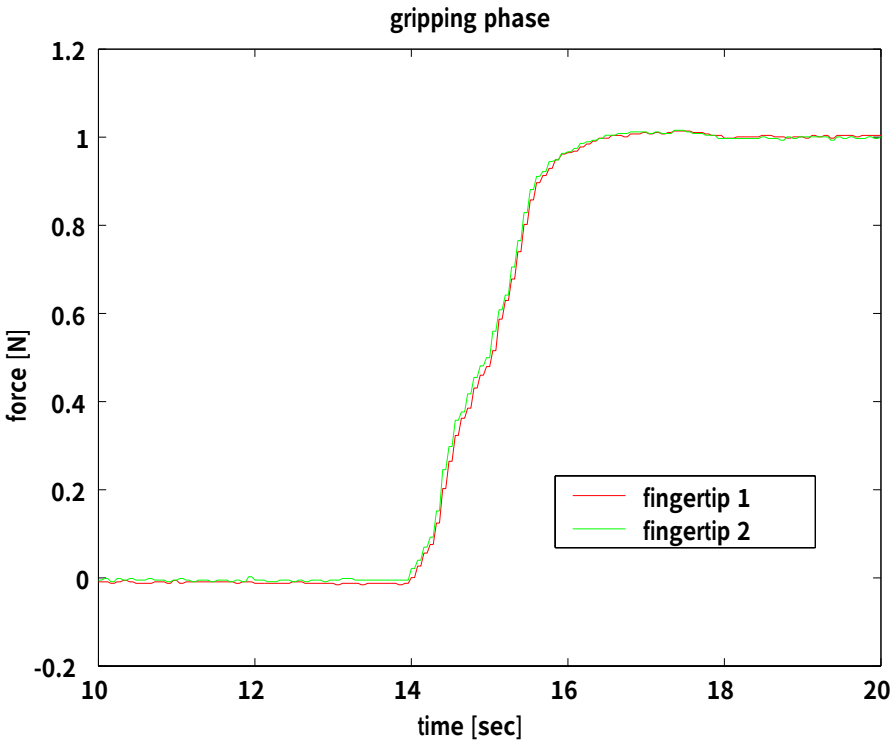


Figure 6.12: Time trajectory of fingertips



(a)Time trajectories of the finger position detected by camera



(b)Time trajectories of the contact force detected by strain gauges

Figure 6.13: Experimental results for the gripping phase

6.4 Conclusion

The principle of not moving the target object has been developed applying the human features of grasping.

It is important to maintain the first contact force stable and small so that target object does not to fall over. Only strain gauges were used for detection of the contact force, and experiments were successfully carried out with a light can. Since strain gauges could not detect the peak of the impact force arising at the first contact time instant, a soft attachment was put on the fingertips, and it reduced the impact force efficiently. Several materials for the attachment have been also tested, and the gel in a sack had the best result.

Considering the results from a human hand, the motions of two fingertips of robot hand have been synchronized using visual feedback in order to touch the target object simultaneously. The total grasping phase by a robot hand was also demonstrated through simulations and experiments using position and force synchronization, and it showed very good results.

Chapter 7

Conclusion

7.1 Conclusion of this dissertation

In this thesis, research on the total grasping task has been investigated. The manipulator had an uncalibrated camera system and the simple precision gripper with two fingers on the end-effector. Positioning and grasping tasks are investigated individually, assuming that the positioning test is defined precisely enough to allow for grasping of the object.

For the positioning task, a novel approach for uncalibrated and model-less visual servoing using a modified *simplex* iterative search method has been proposed. Most of the previous works on visual servoing assume that the kinematic model of the robot, the camera intrinsic parameters, and a model of the object are known. They would fail if the robot and the vision system were fully unknown. Since our aim is to accomplish a robust positioning task even with unknown system, the proposed method did not need any kinematic model of the robot, or intrinsic parameters of the camera. The method can work also without precise model of the object, only an image of the object is needed. We just assumed that the joint limits were known. Thus this approach can be easily applied to positioning task by any kind of robot.

To grasp an object, it is important to contact it "softly", especially if the target object is fragile or may easily fall down. Human ability to grasp an object without large displacements has been measured, and the main features of human grasping motions have been applied to a robot hand control law.

7.2 Specific contributions of this thesis

In the second chapter of this thesis, existing approaches for uncalibrated visual servoing are reviewed. These methods use an iterative Newton-like optimization algorithm that requires the knowledge of the Jacobian matrix of the objective function. In uncalibrated visual servoing, the Jacobian matrix which relates objective function and system configuration has to be estimated on-line. Once the dimension of configuration variables vector becomes large, this adaptive scheme may diverge. Simulation results with 2DOF and 6DOF exhibit this potential stability problem. However, it should be noted that when the optimization process started near the global minimum, convergence could be very quick with high accuracy.

Chapter 3 is dedicated to the presentation of simplex optimization technique. This method required neither a model of the robot nor the estimation of Jacobian matrices. Constraints at the joint-level can be easily included. Since the simplex method does not need to calculate or to estimate the gradient of the objective function, a robot never goes to irrelevant directions due to a poorly estimated Jacobian. Moreover, the objective function does not need to be differentiable. We successfully demonstrated the proposed scheme with simulations, and the results have been compared with that of Newton-like methods.

In Chapter 4, the application of the simplex method to the robot positioning task is presented. Using different types of cost functions, attempts to accelerate the convergence and to exit from local minima are discussed. Objective functions do not need to be differentiable, therefore, it is even possible to use pixel matching. The comparison between joint angles and camera pose as the variables were also done. The proposed scheme is successfully demonstrated. Successful experiments with industrial manipulator are also presented.

In Chapter 5, the contact force and the displacement of the object are analyzed. Motions of human fingers are investigated with objects of variable width. We found that human slightly move light objects while grasping them. The contact force has been measured by looking at the finger that touches the target object first. The time derivative of the first contact force is classified into three patterns, and they are related to the fingertip which makes the first contact. When the index finger touches the object first, the contact force increases in a stable manner before the thumb makes also contact. On the other hand, when the thumb touches the object first, its contact force increases, but not stable. The motion of the finger has been

measured using an object of variable width. When a force is applied, the index finger makes large motions while the thumb hardly moves, regardless of visual information. The index finger applies the force actively, and the thumb supports it.

In Chapter 6, the human features of soft grasping are applied to a robot hand with two fingers. It is important to maintain the first contact force stable and small in order to avoid the target object would not to fall over. Only strain gauges were used for detection of the contact force, and the experiment is successfully carried on a light can. Strain gauges could not detect the impact force arising from the first contact. A soft attachment is put on the fingertips, and it efficiently reduces the impact force. The comparison between several types of material for the attachment has been performed, and the gel sack has the best result. Considering the difference from a human hand, the motion of two fingers of the robot hand have to be synchronized in order to touch the target object simultaneously using visual feedback. The total grasping phase by a robot hand is demonstrated using a position and force synchronization scheme, and it works very well.

7.3 Future works

The purpose of the methods we have developed is to make manipulation of object even with minimum information about the model of the robot, the model of the object, and the camera intrinsic parameters.

For positioning task, simplex algorithm only uses the current image features. It is also interesting to use the image features taken during the simplex iterations for recognition of the object, learning the environment, and reproducing the learned motions even if the object has moved position. For example, to construct a subspace can be a solution. Eigenspace methods relate the robot position to the feature images as a compressed representation in a subspace. Eigenspace methods have been extensively explored for object recognition [61] [62] [50], visual positioning and tracking [60] [63] [8]. Normally the construction of an eigenspace needs intensive computation, and many sample images are needed to construct the subspace with enough reliability. Simplex algorithm can be used for an unknown object, and at the same time the subspace can be calculated. It may help to make faster convergence and also it is useful for faster positioning task with the same target object when it is done a second time.

The simplex algorithm may also integrate multiple tasks. In the case of a multiple camera system like when a camera is mounted on the robot's end-effector and another camera is fixed in the environment, both the positioning task with respect to an object and obstacles (e.g., the floor, other objects) avoidance task observed by the fixed camera can be take into account simultaneously. Not only for vision systems, this method can be used with other type of informations, e.g., vision and laser scan.

For grasping task, we realized a "soft" touch motion applying human grasping features. Human finger stops when it touches the object, and keep the small contact force until the other finger touches it. It is applied to the robot using the strain gauges as the force sensors, but position synchronization is employed for more reliability. The structure of human soft fingers helps reducing the impact force applied to the object, and it is also important to apply a robot hand. In future works we would like to complete experiments with fast positioning and grasping at the same time by mounting a hand on a robot arm.

Appendix A

Basic definitions

A.1 Coordinates and pose

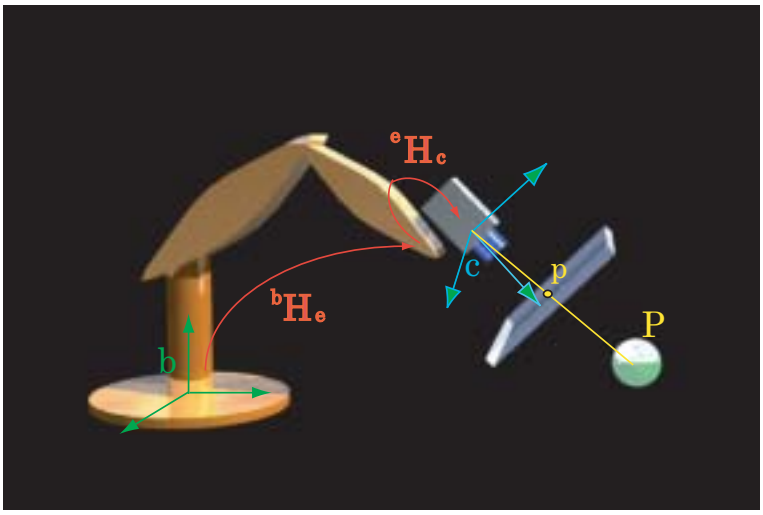


Figure A.1: coordinate frames and pose

Figure A.1 shows coordinate frames of reference the robot base and the camera (in the case of an eye-in-hand configuration). When the camera on the robot's end-effector is moved, the pose of the robot has to be determined, and it can be described using coordinate frames. Coordinates of a point P with respect to coordinate frame i is expressed as ${}^{i}P$:

$$ {}^{i}P = [{}^{i}P_x \; {}^{i}P_y \; {}^{i}P_z]^{\top} \tag{A.1} $$

Position and orientation of a frame i with respect to frame j is defined as a *pose*

vector:

$$^{j}p_i = [T_x\ T_y\ T_z\ \alpha\ \beta\ \gamma]^\top \tag{A.2}$$

where $(T_x\ T_y\ T_z)^\top = {}^{j}O_i$ is a translation vector that gives the coordinates of the origine of frame i with respect to frame j, and $(\alpha\ \beta\ \gamma)^\top$ are rotation angles (e.g., like the Euler angles, role, pitch, and yaw) that allow to define a rotation matrix ${}^{j}R_i$ describing the rotation between frame i and frame j. Coordinates of point ${}^{i}P$ with respect to frame j can be calculated using:

$$^{j}P = {}^{j}O_i + {}^{j}R_i{}^{i}P \tag{A.3}$$

The coordinates of vector ${}^{i}V$ with respect to frame j can be calculated using:

$$^{j}V = {}^{j}R_i{}^{i}V \tag{A.4}$$

This can be also expressed with a *homogeneous transformation* ${}^{j}H_i$:

$$^{j}H_i = \begin{bmatrix} {}^{j}O_i & {}^{j}R_i \\ 0 & 1 \end{bmatrix} \tag{A.5}$$

$$\begin{bmatrix} {}^{j}P \\ 1 \end{bmatrix} = {}^{j}H_i \begin{bmatrix} {}^{i}P \\ 1 \end{bmatrix} \tag{A.6}$$

$$\begin{bmatrix} {}^{j}V \\ 0 \end{bmatrix} = {}^{j}H_i \begin{bmatrix} {}^{i}V \\ 0 \end{bmatrix} \tag{A.7}$$

Homogeneous transformations express change of coordinate frame and can be combined:

$$^{j}H_i = {}^{j}H_k{}^{k}H_i \tag{A.8}$$

When the camera pose with respect to the coordinate frame of the robot's end-effector, and the pose of the end-effector with respect t othe coordinate frame of the robot's base are known, the pose of the camera with respect to the base of the robot can be easily calculated.

The velocity screw of frame i with respect to frame j in frame j coordinates is described by:

$$^{j}({}^{j}\dot{r}_i) = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^\top \tag{A.9}$$

$$\begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = {}^j({}^jV_i) \quad \text{is a translational velocity} \tag{A.10}$$

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = {}^j({}^j\Omega_i) \quad \text{is a translational velocity} \tag{A.11}$$

Then, the velocity of point P rigidly attached to frame i with respect to frame j expressed in frame j coordinates is given by:

$$\begin{aligned} {}^j\dot{P} &= {}^j({}^j\Omega_i) \times {}^jP + {}^j({}^jV_i) \\ &= {}^j({}^j\Omega_i) \times ({}^jR_iP + {}^jO_i) + {}^j({}^jV_i) \end{aligned} \tag{A.12}$$

Note that the velocity screw is generally not equal to the derivative of the pose.

A.2 Camera projection model

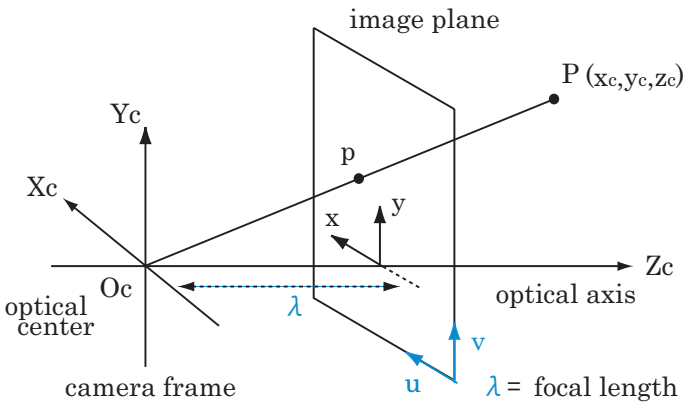


Figure A.2: Camera projection model

Figure A.2 shows the perspective transformation from a point in the camera coordinates to point in the image plane. The optical center is taken as the origin of the camera frame, and where λ is the focal length. If the coordinates of the target are ${}^cP = (x_c, y_c, z_c)^\top$, and its coordinates in the image plane are $(x, y)^\top$, then:

$$ {}^cP = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \lambda \frac{x_c}{z_c} \\ \lambda \frac{y_c}{z_c} \end{bmatrix} \tag{A.13} $$

The center of the image plane frame of coordinate is generally not on the optical axis, and the scale of the image plane is not same as that of the camera coordinate frame. Thus, the coordinates of the projection of P on the image plane is given by:

$$ \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 + \lambda k_u \frac{x_c}{z_c} \\ v_0 + \lambda k_v \frac{y_c}{z_c} \end{bmatrix} \tag{A.14} $$

where u_0 and v_0 are the coordinates of the position of intersection of the optical axis Z_c with the image plane expressed in the image frame, k_u and k_v are scaling factors between the image plane and the camera frame coordinates. These intrinsic camera parameters are obtained by calibration.

A.3 Image Jacobian

The image Jacobian or interaction matrix expresses an relatopnship between the velocity of the features and the velocity screw of the target:

$$\dot{f} = J_I{}^c(\dot{r}) \tag{A.15}$$

For example, the motion of the projection $[uv]^\top$ in the image plane ${}^cP = [x\ y\ z]^\top$ expressed in the camera frame can be described using the velocity screw of a fome rigidly attached to P in the camera frame ${}^c({}^c\dot{r}_p) = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^\top$.

Therefore,

$$\begin{aligned} {}^c\dot{P} &= {}^c({}^c\Omega_p) \times {}^cP + {}^c({}^cV_p) \\ {}^c\dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} &= \begin{bmatrix} z\omega_y - \frac{vz}{\lambda}\omega_z + v_x \\ \frac{uz}{\lambda}\omega_z - z\omega_x + v_x \\ \frac{z}{\lambda}(v\omega_y - u\omega_x) + v_z \end{bmatrix} \end{aligned} \tag{A.16}$$

where $u_0 = 0$, $v_0 = 0$, and $k_x = 1$, $k_y = 1$ in Equation A.14. Then, the velocity of the projection of ${}^cP$ in the image plane is given by:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{z} & 0 & -\frac{u}{z} & -\frac{uv}{\lambda} & \frac{\lambda^2+u^2}{\lambda} & -v \\ 0 & \frac{\lambda}{z} & \frac{v}{z} & -\frac{\lambda^2+v^2}{\lambda} & \frac{uv}{\lambda} & u \end{bmatrix} {}^c({}^c\dot{r}_e) \tag{A.17}$$

Which implies that:

$$J_I = \begin{bmatrix} \frac{\lambda}{z} & 0 & -\frac{u}{z} & -\frac{uv}{\lambda} & \frac{\lambda^2+u^2}{\lambda} & -v \\ 0 & \frac{\lambda}{z} & \frac{v}{z} & -\frac{\lambda^2+v^2}{\lambda} & \frac{uv}{\lambda} & u \end{bmatrix} \tag{A.18}$$

Bibliography

[1] D. P. Bertsekas, "Nonlinear Programming, 2nd edition," *Athena Scientific*, Belmont, 1999.

[2] M.J. Box, "A new method of constrained optimization and a comparison with other methods," *Computer Journal*, vol. 8, no. 1, pp. 42-52, April 1965.

[3] F. Chaumette, and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, April 2000.

[4] P. I. Corke and S. A. Hutchinson, "Real-time Vision, Tracking and Control," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, April 2000.

[5] P. I. Corke and M. C. Good, "Dynamic Effects in Visual Closed-loop Systems," *IEEE Transactions on Robotics and Automation*, vol.12, no.5, pp.671-683, October 1996.

[6] M. R. Cutkosky and P. K. Wright, "Friction, Stability and the Design of Robotic Fingers," *International Journal of Robotics Research*, vol. 5, no. 4, pp. 20-37, 1986.

[7] M. R. Cutkosky, "On Grasp Choice, Grasp Mdels, and the Design of Hands for Manufacturing Tasks,"*IEEE Transactions on Robotics and Automation*, vol.5, no.3, pp. 269-279, 1989.

[8] K. Deguchi, and T. Noguchi, "Visual Servoing Using Eigenspace Method and Dynamic Calculation of Interaction Matrices," *Proceedings of the 1996 IEEE International Conference on Pattern Recognition*, Vienna, August 1996.

[9] K. Deguchi, "A Direct Interpretation of Dynamic Images with Camera and Object Motions for Vision Guided Robot Control," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 7-20, August 2000.

[10] Z. Doulgeri, A. Simeonidis, and S. Arimoto, "A position/force control for a soft tip robot finger under kinematic uncertainties," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, April 2000.

[11] B. Espiau, F. Chaumette, and P. Rives, "A New Approach to Visual Servoing in Robotics," *IEEE Tranactions on Robotics and Automation*, vol. 8, no. 3, pp. 313-326, June 1992.

[12] J. R. Flanagan, and A. M. Wing, "The Stability of of Precision Grip Forces During Cyclic Arm Movements with a Hand-held Load," *Exp Brain Res*, vol. 105, pp. 455-464, 1995.

[13] J. R. Flanagan, and A. M. Wing, "The role of internal models in motion planning and control: evidence from grip force adjustments during movements of hand-held loads," *Journal of Neurophysiology*, vol. 71, pp. 1519-1528, 1997.

[14] J. R. Flanagan, M. K. O. Burstedt, and R. S. Johansson, "Control of Fingertip Forces in Multidigit Manipulation," *Journal of Neurophysiology*, vol. 81, pp. 1706-1717, 1999.

[15] T. Flash and N. Hogan, "The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model," *Journal of Neurosciences*, vol. 5, no. 7, pp. 1688-1703, 1985.

[16] H. Forssberg et al. "Development of human precision grip I -Basic coordination of force-," *Exp Brain Res*, vol. 85, pp. 451-457, 1991.

[17] J. Gangloff and M. de Mathelin, "High speed visual servoing of a 6 DOF manipulator using multivariable predictive control," *Advanced Robotics*, vol. 17, no. 10, pp. 993-1021, December 2003.

[18] J. Gangloff and M. de Mathelin, "Visual servoing of a 6 DOF manipulator for unknown 3D profile following," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 511-520, August 2002.

[19] R. Ginhoux, J. A. Gangloff, M. F. de Mathelin, L. Soler, J. Leroy, J. Marescaux, "A 500 Hz Predictive Visual Servoing Scheme to Mechanically Filter Complex Repetitive Organ Motions in Robotized Surgery," *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, October, 2003.

[20] Graziano Chesi, Koichi Hashimoto, D. Prattichizzo and A. Vicino, "A Switching Control Law for Keeping Features in the Field of View in Eye-in-Hand Visual Servoing," *Proceedings of 2003 IEEE International Conferende on Robotics and Automation*, Taipei, September, 2003.

[21] Graziano Chesi and Koichi Hashimoto, "Improving Camera Displacement Estimation in Eye-in-Hand Visual Servoing: A Simple Strategy," *Proceedings of 2003 IEEE International Conferende on Robotics and Automation*, Taipei, September, 2003.

[22] Graziano Chesi and Koichi Hashimoto, "Static-Eye Against Hand-Eye Visual Servoing," *Proceedings of 41st IEEE Conference on Decision and Control*, Las Vegas, December, 2002.

[23] K. Hashimoto and T. Noritsugu, "Visual Servoing with Linearized Observer," *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, Detroit, May 1995.

[24] K. Hashimoto and H. Kimura, "Visual Servoing with Nonlinear Observer," *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, May 1999.

[25] K. Hashimoto, "Configuration and Robustness in Visual Servo," *Proceedings of the JSME Conference on Robotics and Mechatronics*, Hakodate, May 2003.

[26] K. Hosoda, and M. Asada, "Versatile Visual Servoing without Knowledge of True Jacobian," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Munchen, September 1994.

[27] K. Hosoda, K. Sakamoto and M. Asada, "Trajectory Generation for Obstacle Avoidance of Uncalibrated Stereo Visual Servoing without 3D Reconstruction," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Pittsburgh, August, 1995.

[28] S. Hutchinson, G. D. Hager, and P. I. Corke, "A Tutorial on Visual Servo Control," *IEEE Tranactions on Robotics and Automation*, vol. 12, no. 5, pp. 651-670, October 1996.

[29] S. C. Jacobsen et al., "The UTAH/M.I.T. Dextrous Hand: Work in Prigress," *International Journal of Robotics Research*, vol. 3, no. 4, pp. 21-50, 1984.

[30] M. Jägersand, O. Fuentes, and R. Nelson, "Experimental Evaluation of Uncalibrated Visual Servoing for Precision Manipulation," *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, New Mexico, April 1997.

[31] P. Jenmalm et al. "Control of Grasp Stability When Humans Lift Objects With Different Surface Curvatures," *Journal of Neurophysiology*, vol. 79, pp. 1643-1652, 1998.

[32] I. Kao, et al., "Robotic Stiffness control and Calibration as Applied to Human Graspin Tasks," *IEEE Transactions on Robotics and Automation*, vol.13, no.4, pp.557-566, 1997.

[33] J. Kerr and B. Roth, "Analysis of Multifingered Hands," *Journal of MTAD Transactions of ASME*, vol. 4, no. 4, pp. 3-17, 1985.

[34] I. Kim, H. Shiire, and H. Inooka, "Experimental study on grasp force control characteristics of a human," *The Japanese Journal of Ergonomics*, vol. 29, no. 2, pp. 95-102, 1993.

[35] I. Kim, and H. Inooka, "Determination of Grasp Forces for Robot Hands Based on Human Capabilities," *Control Engineering Practice (Elsevier Science)*, vol. 2, no. 3, pp. 415-420, 1994.

[36] I. Kim, N. Nakazawa, and H. Inooka, "Control of a robot hand emulating human's hand-over motion," *Mechatronics (Elsevier Science)*, vol. 12, no. 1, pp. 55-69, 2002.

[37] Kinoshita et al. "The development of the grasp force adjustment function under pluck motion of the small object," *Biomechanics*, vol. 12, pp. 199-209, 1994.

[38] A. Krupa, J. Gangloff, C. Doignon, M. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux, "Autonomous 3-D positioning of surgical instruments

in robotized laparoscopic surgery using visual servoing," *IEEE transactions on robotics and automation*, vol. 19, no. 5, pp. 842-853, October 2003.

[39] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D Visual Servoing," *IEEE Transactions on Robotics and Automation*, vol.15, no.2, pp.238-250, April 1999.

[40] E. Malis, F. Chaumette, and S. Boudet, "Multi-cameras Visual Servoing," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, April 2000.

[41] M. de Mathelin, and R. Lozano, "Robust adaptive identification of slowly time varying parameters with bounded disturbances," *Automatica*, vol. 35, pp. 1291-1305, 1999.

[42] B. Mishra and N. Silver, "Some Discussion of Static Gripping and Its Stability," *IEEE Transaction SMC*, vol.19, no.4, pp. 783-796, 1989.

[43] K. Miura, K. Sagawa, and H. Inooka, "On the Measurement of Walking Velocity to Derive Consumed Energy," *Proceedings of the 38th SICE Annual Conference Domestic Session Papers*, vol. II, pp. 763-764, Morioka, 1999.

[44] K. Miura, Y. Ohtaki, K. Sagawa, and H. Inooka, "Generation of Walking Pattern and Kansei Evaluations," *The 190th SICE Tohoku Local Research Meeting*, reference no. 190-9, Yonezawa, 2000.

[45] N. Nakazawa, K. Miura, and H. Inooka, "A Study on the First Contact Force of Fingertips in Grasping," *The 3rd SICE System Integration Division Annual Conference*, Proceedings(II), pp. 439-440, Kobe, 2002.

[46] K. Miura and H. Inooka, "Uncalibrated Visual Servoing for Soft Grasping," *Proceedings of the 45th Automatic Control Association Annual Conference*, pp. 203-204, Sendai, 2002.

[47] K. Miura, N. Nakazawa, and H. Inooka, "A study on the First Contact Force and Control of Fingertips in Grasping," *Proceedings of 2003 JSME Conference on Robotics and Mechatronics*, 1P1-3F-D6, Hakodate, 2003.

[48] K. Miura, Y. Ohtaki, and H. Inooka, "Impression Analysis of Various Human-like Gait Patterns," *Proceedings of IEEE International Workshop on Robot and*

Human Interactive Communication, pp. 568-573, Bordeaux and Paris, France, 2001.

[49] K. Miura, J. A. Gangloff, M. F. de Mathelin, "Robust and Uncalibrated Visual Servoing without Jacobian using a Simplex Method," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 311-316, Lausanne, Switzerland, 2002.

[50] H. Murase, and S. K. Nayar, "Visual learning and recognition of 3D objects from appearance," *International Journal of Computer Vision*, vol. 14, no. 1, pp. 5-24, January 1995.

[51] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno, "1ms Column Parallel Vision System and It's Application to High Speed Target Tracking," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, April 2000.

[52] N. Nakazawa, R. Ikeura, and H. Inooka, "Experimental study on grasp force control characteristics of a human," *The Japanese Journal of Ergonomics*, vol. 32, no. 5, pp. 223-229, 1996.

[53] N. Nakazawa, R. Ikeura, and H. Inooka, "Characteristics of Human Fingertips in Shearing Direction," *Transactions of the Japan Society of Mechanical Engineers*, vol. 64, no. 624, pp. 292-298, 1998.

[54] N. Nakazawa, S. Kajikawa, H. Inooka, and R. Ikeura, "Experimental study on path planning of fingertips in human grapping motion," *The Japanese Journal of Ergonomics*, vol. 36, no. 1, pp. 19-27, 2000.

[55] N. Nakazawa, Y. Uekita, R. Ikeura, and H. Inooka, "Impedance Characteristics of Human Finger Joints," *Transactions of the Japan Society of Mechanical Engineers*, vol. 66, no. 641, pp. 214-219, 2000.

[56] N. Nakazawa, S. Kajikawa, H. Inooka, and R. Ikeura, "Characteristics of human fingertips in the shearing direction," *Biological Cybernetics (Springer-Verlag)*, vol. 83, no. 3, pp. 207-214, 2000.

[57] N. Nakazawa, I. Kim, H. Inooka, and R. Ikeura, "Force control of a robot gripper based on human grasping schemes," *Control Engineering Practice (Elsevier Science)*, vol. 9, no. 7, pp. 735-742, July 2001.

[58] N. Nakazawa, S. Kajikawa, H. Inooka, and R. Ikeura, "Simulations of Path Planning Emulating Human's Grasping Motions," *Transactions of the Japan Society of Mechanical Engineers*, vol. 68, no. 669, pp. 1474-1480, 2002.

[59] N. Nakazawa, K. Miura, and H. Inooka, "Behavior and role of the fingertips in the beginning of human grasping motion," *The Japanese Journal of Ergonomics*, Accepted for publication (April, 2004).

[60] S. K. Nayar, H. Murase, and S. A. Nene, "Learning, Positioning, and Tracking Visual Appearance," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, May 1994.

[61] S. K. Nayar, H. Murase, and S. A. Nene, "Parametric appearance representation," *Early Visual Learning, Oxford University Press*, pp. 131-160, February 1996.

[62] S. K. Nayar, H. Murase, and S. A. Nene, "Real-Time 100 Object Recognition System," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, Minneapolis, April 1996.

[63] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace Methods for Robot Vision," *IEEE Tranactions on Robotics and Automation*, vol. 12, no. 5, pp. 750-758, October 1996.

[64] J. A. Nelder, and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, no. 4, pp. 308-313, January 1965.

[65] J. A. Piepmeier, G. V. MacMurray, and A. Pfeiffer, "Dynamic Quasi-Newton Method for Uncalibrated Visual Servoing," *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, May 1999.

[66] J. A. Piepmeier, G. V. MacMurray, A. Pfeiffer, and H. Lipkin, "Uncalibrated Target Tracking with Obstacle Avoidance," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, April 2000.

[67] S. S. Rao, "Engineering Optimization: Theory and Practice, 3rd edition," *Wiley Eastern Ltd.*, New Delhi, 1996.

[68] H. Ritter, T. Martinetz, and K. Schulten, "Neural Computation and Self-Organizing Maps," *Addison-Wesley*, 1992.

[69] J. K. Salisbury and B. Roth, "Kinematic and Force Analysis of Articulated Mechanical Hands," *Journal of MTAD Transactions of ASME*, vol. 105, pp. 35-41, 1983.

[70] J. K. Salisbury and J. J. Craig, "Articulated Hands: Force Control and Kinematic Issues," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 4-17, 1982.

[71] K. B. Shimoga and A. A. Goldenberg, "Soft Robotic Fingertips Part B: Modeling and Impedance Regulation," *International Journal of Robotics Research*, vol. 15, no. 4, pp. 335-350, 1996.

[72] J. Son, R. Howe, J. Wang, and G. Hager, "Preliminary results on grasping with vision and touch," *Proceedings of the 1996 IEEE International Conference on Intelligent Robots and Systems*, 1996.

[73] W. Spendley, G. R. Hext, and F. R. Himsworth, "Sequential application of simplex designs in optimization and evolutionary operation," *Technometrics*, vol. 4, pp. 441-461, November 1962.

[74] Y. Uno, M. Kawato, and R. Suzuki, "Formation and Control of Optimal Trajectory in Human Multi-Joint Arm Movement - Minimum Torque Change Model," *Biological Cybernetics*, vol. 61, pp. 89-101, 1989.

[75] J. Wang and W. J. Wilson, "3D Relative Position and Orientation Estimation Using Karman Filtering for Robot Control," *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, May 1992.

[76] A.C. Sanderson and L.E. Weiss. "Image-based visual servo control using relational graph error signals," *Proceedings of the 1980 IEEE International Conference on Cybernetics and Society*, pp. 1074-1077, October 1980.

[77] G. Westling and R. S. Johansson, "Factors Influencing the Force Control During Precision Grip," *Exp Brain Res*, vol. 53, pp. 277-284, 1984.

[78] W. J. Wilson, C. C. Williams, and G. S. Bell, "Relative End-effector Control Using Cartesian Position Based Visual Servoing," *IEEE Transactions on Robotics and Automation*, vol.12, no.5, pp.684-696, October 1996.

[79] T. Yoshikawa and K. Nagai, "Manipulating and Grasping Forces in Manipulation by Multifingered Robot Hands," *IEEE Tranactions on Robotics and Automation*, vol. 7, no. 1, pp. 67-77, 1991.

Publications

K. Miura, K. Sagawa, and H. Inooka, "On the Measurement of Walking Velocity to Derive Consumed Energy," *Proceedings of the 38th SICE Annual Conference Domestic Session Papers*, vol. II, pp. 763-764, Morioka, 1999.

K. Miura, Y. Ohtaki, K. Sagawa, and H. Inooka, "Generation of Walking Pattern and Kansei Evaluations," *The 190th SICE Tohoku Local Research Meeting*, reference no. 190-9, Yonezawa, 2000.

N. Nakazawa, K. Miura, and H. Inooka, "A Study on the First Contact Force of Fingertips in Grasping," *The 3rd SICE System Integration Division Annual Conference*, Proceedings(II), pp. 439-440, Kobe, 2002.

K. Miura and H. Inooka, "Uncalibrated Visual Servoing for Soft Grasping," *Proceedings of the 45th Automatic Control Association Annual Conference*, pp. 203-204, Sendai, 2002.

K. Miura, N. Nakazawa, and H. Inooka, "A study on the First Contact Force and Control of Fingertips in Grasping," *Proceedings of 2003 JSME Conference on Robotics and Mechatronics*, 1P1-3F-D6, Hakodate, 2003.

K. Miura, Y. Ohtaki, and H. Inooka, "Impression Analysis of Various Human-like Gait Patterns," *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*, pp. 568-573, Bordeaux and Paris, France, 2001.

K. Miura, J. A. Gangloff, M. F. de Mathelin, "Robust and Uncalibrated Visual Servoing without Jacobian using a Simplex Method," *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 311-316, Lausanne, Switzerland, 2002.

N. Nakazawa, K. Miura, and H. Inooka, "Behavior and role of the fingertips in the beginning of human grasping motion," *The Japanese Journal of Ergonomics*, Accepted for publication (April, 2004).