Experimental Study on Visual Servo Control of Robots

LAM Kin Kwan

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Automation and Computer-Aided Engineering

April 2005

Abstract

Visual servoing is a robot control method which uses visual information as control feedback. The research on visual servoing has been drawing extensive attention in recent years. To achieve high performance control, the camera intrinsic parameters and the homogeneous transformation matrix between the camera coordinate frame and the robot coordinate frame have to be calibrated accurately. However, the camera calibration process is tedious. Also, most research works focus only on the kinematics, neglecting the dynamic effects which limit the performance of controllers.

To target to the problems mentioned above, we propose a new image-based visual servo controller. The controller assumes that the camera intrinsic parameters and the homogeneous transformation matrix of the camera coordinate frame with respect to the robot coordinate frame are totally uncalibrated. Also, to achieve high performance of controller, full robot dynamics has been taken into account. The controller is proven to be asymptotically stable in image position error for set-point control. Also, the uncalibrated parameters can be estimated to the true values up to a scale by an adaptive approach. The performance of this new controller is mainly analyzed by Lyapunov theory, simulations and simple experiments. However, more and comprehensive experiments need to be conducted to verify the performance and the behavior of the controller.

The objective of this thesis is to conduct comprehensive experimental study on our dynamic adaptive controller for uncalibrated visual servoing. The experimental work is organized into two main parts. First, we compare the performance of the dynamic adaptive controller with a kinematic visual servo controller. Second, we investigate the properties of the new controller.

First, the properties of the proposed controller are investigated. Adjusting the proportional gain (P-gain) and the derivative gain (D-gain) of the controller has been experimented. Higher P-gain can shorten converging time, while higher D-gain can reduce oscillations. Varying adaptive gains are also experimented. To ensure fast convergence of adaptive parameters, non-unified adaptive gain should be implemented. Comparison between presence and absence of gravity compensator has been conducted. It is difficult to compare because the masses of robot links are too small. Comparison between controller using previous image positions and controller only using current position has been conducted. Theoretically, there should be at least seven image points to guarantee the adaptive parameters to be converged to the true values up to a scale. However, due to the large sampling period of the system, the parameters may not be converged to the true values up to a scale.

Then, we compare the dynamic adaptive controller with a kinematic controller. Converging path for dynamic controller is not smooth due to friction. Then, a kinematics controller with the same desired trajectory in different motor controllers is experimented. The performances differ greatly, which show that performance of kinematics controller is limited by the performance of motion controller.

摘　　要

視覺伺服是一種使用視覺訊息作爲反饋的機器人控制方法。視覺伺服的研究近年備受注目。攝影機的內部參數、攝影機坐標和機器人坐標之間的轉換矩陣皆需要校準，以達至高效能控制。視覺伺服具挑戰性的是繁複的校準過程。另外，大多數的研究只集中於運動學而忽略動力學效應，這會限制控制器的性能。

針對上述的問題，我們發展了新的圖像基視覺伺服控制器。這控制器是在攝影機的內部參數、攝影機坐標和機器人坐標之間的轉換矩陣皆沒有校準的情況下運作。另外，我們也考慮機器人動力學的影響，以達至高效能控制。這控制器已被證明圖像位置誤差可漸進收斂。同時，自適應算法可將沒有校準的參數估計至距離真實值有一比例。我們已透過 Lyapunov 理論、仿真和簡單的實驗來核實這控制器的效能。然而，這控制器需要更多實驗，以核實這控制器的性能和特性。

這篇論文的目標是要替我們的動力學自適應控制器進行進一步實驗。實驗的結果展示如下。同樣的運動學控制程式在不同效能的馬達控制器上，有相當不同的效果。增加比例比可以增加圖像位置誤差趨近速度，而增加微分比可以減少震盪。爲了讓自適應參數快速趨近，應該使用非統一性的自適應比。重力保償可以除去常態圖像位置誤差。理論上，有七個圖像位置點可以確保自適應參數趨近至距離真實值有一比例；然而，實際上自適應參數可能未能趨近至距離真實值有一比例，因爲系統抽樣的週期時間較長。

Acknowledgement

Most importantly, I would like to thank my supervisor Prof. Liu Yun-Hui, for his supervision in the past two and a half years. His guidance and encouragement were crucial in the completion of the research work. Through the comprehensive discussion with him, my knowledge and research skills have been enhanced a lot.

I would also like to thank Prof. Ronald Chung and Prof. Charlie Wang for their serving in the thesis committee members. Their comments are thoughtful and valuable. Special thanks also go to external member of the thesis committee: Prof. Shugen Ma from Ibaraki University, Japan, who joined us in my oral examination.

Moreover, I would like to thank my colleagues in Robot Control Laboratory for their help and support. Especially I would like to thank Mr. Wang Hesheng for discussion on the concepts of my work. Also, I would like to thank Mr. Martin Leung for his help in hardware setup for experiments.

Finally, I am grateful to my parents, my three sisters, and especially, my girl-friend Frances Tang for their endless support and love through my master studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Visual Servoing

Visual servoing is a robot control method that uses visual information as control feedback. Then, the pose (position and orientation) of the robot's end-effector can be controlled by visual information. With the enhancement of vision capability, robots can perform tasks where the work environment and object placement cannot be accurately controlled. [1]

The usage of vision with robots can be traced back in early 1970s. Traditionally, visual sensing and mechanical manipulation are arranged in open-loop method, "looking" and then "moving". [1] The performance of operation depends directly on the accuracy of the visual sensor and the manipulator. In 1973, Shirai and Inoue [2] proposed the use of vision as a control feedback to increase the accuracy of operating task, which is crucial for most applications. In 1979, the term "visual servoing" has been introduced by Hill and Park [3] to distinguish their approach from earlier "blocks world" experiments where the system alternated between photo taking and moving. The research progress of visual servoing has not been very fast

before 1990s. In 1990s, there has been a remarkable increase in published research works due to a rapid advancement in computational power.

1.1.1 System Architectures

The classification of visual servoing architectures has been introduced by Sanderson and Weiss [4], in 1980. Visual servoing architectures are mainly classified as position-based visual servoing and image-based visual servoing.

1.1.1.1 Position-based Visual Servoing

Position-based visual servoing is a visual servoing method which converts the extracted image information to estimated pose of the manipulator, in 3D Cartesian space. The estimated pose becomes the feedback signal for the controller [1]. The architecture is shown in Fig. 1.1.

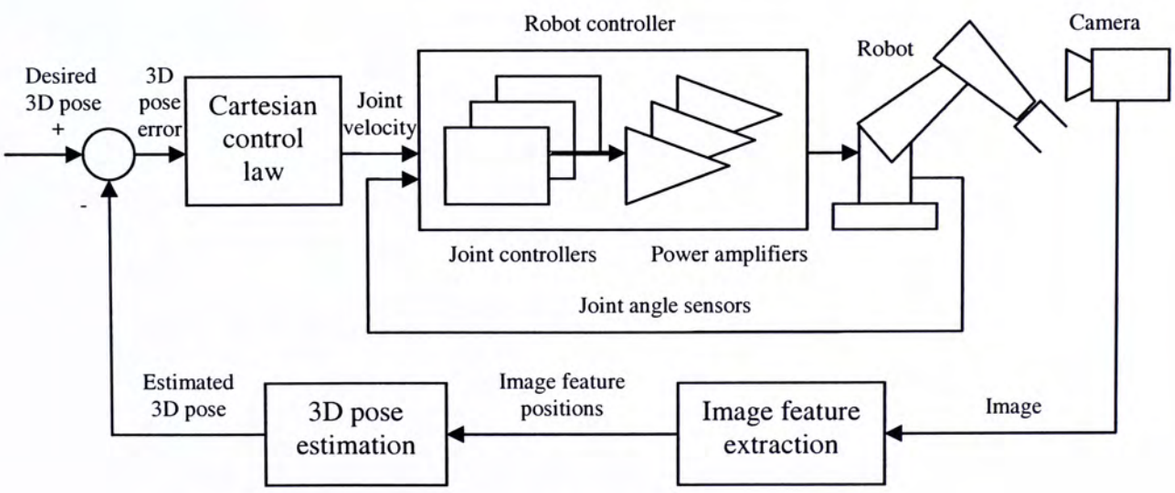


Figure 1.1: Architecture of position-based visual servoing.

The main advantage of position-based visual servoing is that the task information is expressed in terms of Cartesian space. Obstacle avoidance and singularities avoidance can be easily incorporated with the system [5]. The disadvantage is that the estimation of the pose depends greatly on calibration of cameras. The system could become very sensitive to the calibration error [1].

1.1.1.2 Image-based Visual Servoing

Image-based visual servoing is a visual servoing method that feedbacks directly from the 2D image information. The calculation of an image Jacobian or a composite Jacobian (the multiplication of image Jacobian and manipulator Jacobian) plays the crucial part of the controller. The architecture is shown in Fig. 1.2.

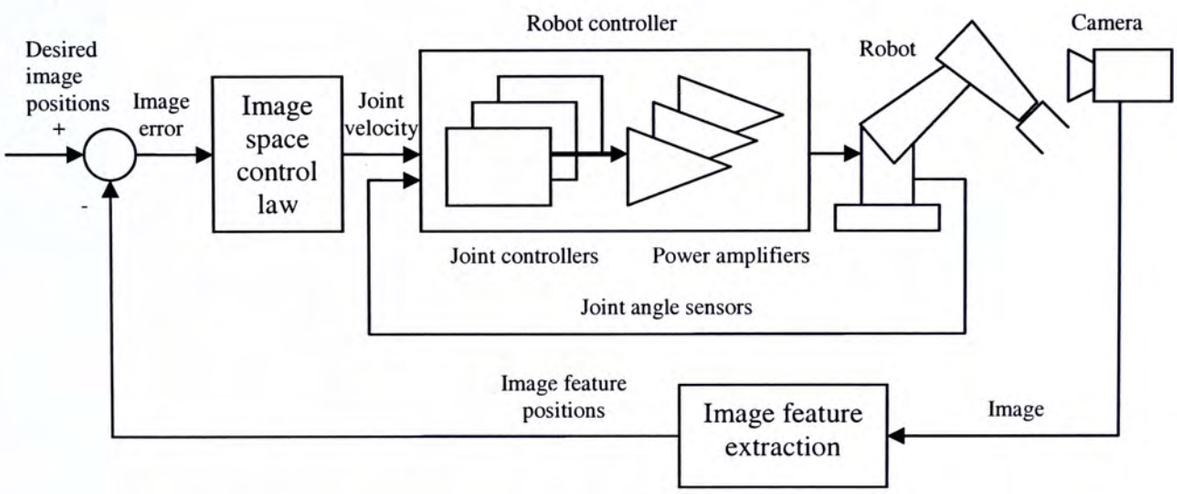


Figure 1.2: Architecture of image-based visual servoing.

The main advantage of image-based visual servoing is the positioning accuracy is more robust than in position-based visual servoing, as the image error is being used directly as the feedback of the controller. The main disadvantage is that

there may have singularities in calculating inverse Jacobian, which may cause the visual servoing system becoming unstable [1].

1.1.2 Camera Configurations

In visual servoing system, there are two types of camera configurations could be implemented. They are the "eye-in-hand" configuration and the "eye-and-hand" configuration.

"Eye-in-hand" configuration consists of the camera attached on the end-effector of the manipulator (Fig. 1.3). The objective task is to move the camera so that the target object is appeared in a particular position in the image. In this configuration, the relationship between the camera frame and the end-effector frame is known.

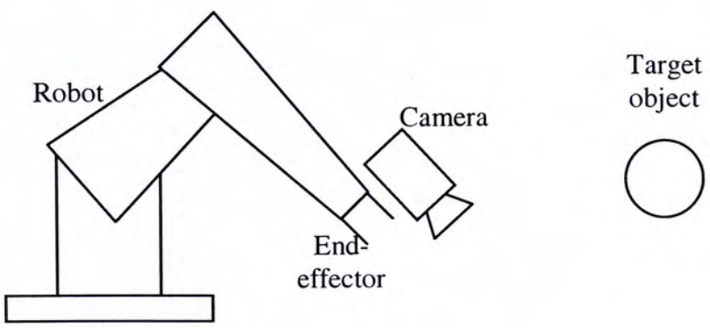


Figure 1.3: Eye-in-hand camera configuration.

"Eye-and-hand" configuration consists of the camera fixed in the working environment (Fig. 1.4). The task is to move the manipulator so that the pose of the manipulator approaches to the desire image position of the image plane. In this

configuration, the relationship between the camera frame and the manipulator base frame is known.

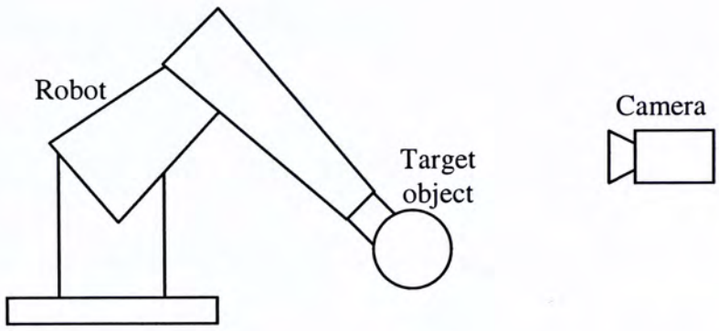


Figure 1.4: Eye-and-hand camera configuration.

For both types of camera configurations, calibrations must be preformed to determine the intrinsic camera parameters and the extrinsic parameters, before visual servoing task operations. Intrinsic parameters include focal length, pixel scaling factors, and the coordinate of the optical axis on the image plane. Extrinsic parameters include the relationship between camera frame and end-effector frame, for eye-in-hand configuration, or the relationship between camera frame and manipulator base frame, for eye-and-hand case [1].

1.2 Problem Definition

For both eye-in-hand and eye-and-hand camera configurations, calibration is very crucial for visual servoing performance. Although image-based visual servoing, as described in the above section, is less sensitive to the calibration error, the Jacobian,

which plays a crucial part in control calculation, still depends on the intrinsic parameters and the extrinsic parameters. If the intrinsic and extrinsic parameters have not been estimated properly, the stability and the performance may be unpredictable. Therefore, estimating for uncalibrated parameters is important of the work of this thesis.

Robot dynamics are often ignored in most development of visual servoing system. However, if only kinematics is considered, the performance would be limited due to neglecting the effects by nonlinear robot dynamics. Therefore, to achieve high performance for visual servo system, taking consideration for nonlinear robot dynamic effects is another focus of the work of this thesis.

The depth estimation is also needed for visual servoing control. For a monocular visual servoing system, depth cannot be measured directly. The depth is included in the Jacobian for image-based controller, which is crucial for control calculation. As depth varies with robot manipulator movement, the estimation of depth becomes challenging. The estimating of depth is also a focus of the work in this thesis.

1.3 Related Work

In early research of visual servoing, the intrinsic camera parameters, the extrinsic parameters, and the depth, are assumed to be known and accurately calibrated [6]. Nowadays, there has been significant amount of research works on image-based

uncalibrated visual servoing control. For those works, identification of the parameters has to be done in an off-line process [7]. However, visual servoing process with off-line parameter identification is not robust to noisy measurements of parameters. To solve this problem, some control methods with on-line parameter identification are proposed. Papanikolopoulos et al. [8] proposed an algorithm based on on-line estimation of relative distance of the target with respect to the camera. They also proposed an algorithm to estimate the depth related parameters [9]. Hosoda et al. [10] proposed the Broyden updating formula to estimate the composite Jacobian. Yoshimi et al. [11] proposed a method to estimate the image Jacobian by utilizing geometric constrains. Feddema et al. [12] proposed a method with the use of ARMAX model and estimated the parameters. However, the research works mentioned above do not consider robot dynamics effect.

To achieve high performance in visual servoing control, consideration of robot dynamics effect has been introduced by Corke and Good [13, 14]. Kelly et al. [15] developed an eye-in-hand system with set-point regulation controller. The approach assumes that the robot dynamics and the depth between camera and object are available. In [16], Kelly also designed an eye-and-hand set-point regulator which estimates the orientation of the camera. However, the controller requires the difference between the estimated orientation and the true orientation to be within $(-\frac{\pi}{2}, \frac{\pi}{2})$. Also, the knowledge of object depth and camera intrinsic parameters is required. In [17], Maruyama and Fujita developed position set-point controllers for eye-in-hand configuration. However, the controllers assume the camera orientation, the camera intrinsic parameters and the object depth are known. Zergeroglu et al. [18] proposed a uniformly ultimately bounded (UUB) set-point regulator for the eye-in-

hand camera configuration. However, the camera orientation is restricted in the range $(-90°,90°)$.

The unknown depths still play an important role in the image Jacobian matrix of visual servo control system. The visual servoing system may still achieve stability without the true knowledge of depth; however, depth is still important for object handling. Also, the depth observability determines the success of parameter estimation. Some researchers determine depth by using two cameras [19, 20]. Such method is called binocular stereo visual servoing. The variation of this approach is using different image positions (more than two), and then the depth is determined by least square technique. This is called active monocular stereo approach [21]. However, these methods estimate the depth off-line. To achieve high performance visual servoing, some researchers have proposed some methods to estimate the depth on-line [16, 22-23]. However, the proposed methods consider planar robot manipulators moving parallel to the image plane. The depth is considered as constant parameters. Malis et al. [24, 25] have proposed a new approach on visual servoing, which is called 2 1/2 D visual servoing. However, robot dynamics are ignored. Cheah et al [26, 27] have proposed feedback control laws for set-point control with uncertain kinematics, Jacobian matrix, and dynamics. In [24, 28-31], the researchers proposed adaptive methods to estimate depth on-line; however, the camera intrinsic parameters are assumed to be known. Kalman filter is another method to estimate depth on-line [1, 32-33]; however, only depth is considered unknown in these works.

As seen from above, most of the visual servoing methods do not consider totally uncalibrated conditions. Only few of these approaches estimate all camera

intrinsic parameters and extrinsic parameters online. Also, only few of these methods consider robot dynamics.

1.4 Contribution of This Work

The purpose of this thesis is to investigate the performance of a new dynamic adaptive visual servo controller by a series of experiments. The performance of the new proposed controller has been analyzed by Lyapunov theory, simulations and brief experiments. More comprehensive experiments need to be conducted to verify the performance and behaviour of the controller. The contributions of this thesis are summarized as the following:

1. Investigate the properties of our new dynamic adaptive visual servo controller through a series of experiments. The properties are studied by experiments comparing performances
   a) among different control gains (proportional gains and derivative gains);
   b) among different adaptive gains;
   c) presence and absence of gravity compensator;
   d) between system using seven previous image positions and current image position only.
2. We compared the performance of our new dynamic adaptive visual servo controller with a kinematic visual servo controller through experiments.

1.5 Organization of This Thesis

In Chapter 2, system model for visual servoing, including robot kinematics, robot dynamics and camera model are reviewed. In Chapter 3, the design of uncalibrated visual servoing controller is discussed. In Chapter 4, extensive experimental results are studied. In Chapter 5, the work done in this thesis is concluded and future work is suggested.

Chapter 2

System Modeling

In the computational process in visual servoing control system, several aspects, such as transformation between different coordinates, system kinematics, robot dynamics, and camera model are involved. In this chapter, the basics of the mentioned aspects are to be reviewed.

2.1 Coordinate Frames

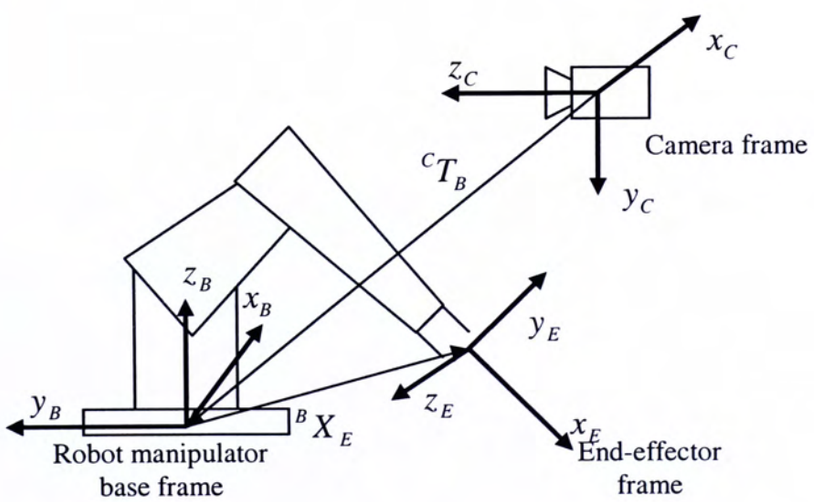


Figure 2.1: The coordinate in visual servoing system.

Fig. 2.1 shows a typical visual servoing setup with eye-and-hand configuration. The three coordinate frames are,

Σ_B : the robot manipulator base coordinate frame,

Σ_E : the end-effector coordinate frame,

Σ_C : the camera coordinate frame.

Generally, transformation relationship between coordinates is represented by a 4×4 homogeneous transformation matrix. The matrix ${}^{C}T_{B} \in \Re^{4\times4}$, which denotes the transformation of the robot manipulator base frame with respect to the camera frame, has the following form:

$$ {}^{C}T_{B} = \begin{bmatrix} {}^{C}R_{B} & {}^{C}p_{B} \\ 0_{1\times3} & 1 \end{bmatrix} $$

where ${}^{C}R_{B} \in \Re^{3\times3}$ denotes the rotational matrix of the robot base frame with respect to the camera frame, ${}^{C}p_{B} \in \Re^{3\times1}$ denotes the position of the origin of the robot base frame with respect to the camera frame. Position in camera coordinate frame can be expressed as:

$$ \begin{bmatrix} {}^{C}X \\ 1 \end{bmatrix} = {}^{C}T_{B} \begin{bmatrix} {}^{B}X \\ 1 \end{bmatrix} \qquad (2.1) $$

where ${}^{C}X \in \Re^{3\times1}$ and ${}^{B}X \in \Re^{3\times1}$ are the position vectors with respect to the camera coordinate frame and the robot manipulator base frame, respectively.

2.2 System Kinematics

Kinematics is the science of motion which refers to all the geometrical and time-based properties of the motion, without regarding to the forces and torques that cause the motion. The forward kinematics of a robot manipulator:

$$^{B}X_{E} = f(q) \tag{2.2}$$

where $^{B}X_{E} = \begin{bmatrix} ^{B}x_{E} & ^{B}y_{E} & ^{B}z_{E} & ^{B}\gamma_{E} & ^{B}\beta_{E} & ^{B}\alpha_{E} \end{bmatrix}^{T}$ represents the position and orientation of the end-effector with respect to the base frame, and $q \in \Re^{n\times 1}$ represents the joint-space coordinates of the robot manipulator with n joints. The first three components of $^{B}X_{E}$ represents the position vector, and the other three represents the orientation.

The velocity relationships between joint-space coordinates and Cartesian-space coordinates (with respect to robot manipulator base frame) can be obtained by taking derivative on both sides of (2.2):

$$^{B}\dot{X}_{E} = J(q)\dot{q} \tag{2.3}$$

where $^{B}\dot{X}_{E}$ is the velocity of the end-effector with respect to the robot manipulator base frame, $\dot{q}$ is the joint velocities, and $J(q)$ is the robot Jacobian matrix of the robot manipulator.

The velocity of end-effector with respect to the camera frame can be expressed as:

$$^{C}\dot{X}_{E} = \begin{bmatrix} ^{C}R_{B} & 0 \\ 0 & ^{C}R_{B} \end{bmatrix} J(q)\dot{q} = AJ(q)\dot{q} \tag{2.4}$$

Then

$$\dot{q} = J^{-1}(q) A^{T\,C} \dot{X}_E \tag{2.5}$$

where $J(q)$ is assumed to be square and nonsingular.

2.3 System Dynamics

Dynamics is the science on the forces and torques which cause the motion. The dynamics of a frictionless serial n-link manipulator is expressed as:

$$\tau = H(q)\ddot{q} + C(q,\dot{q}) + G(q) \tag{2.6}$$

where $\tau \in \Re^{n\times 1}$ represents input joint torques, $H(q) \in \Re^{n\times n}$ represents the inertia matrix, $C(q,\dot{q}) \in \Re^{n\times 1}$ represents the vector of centripetal and Coriolis torques, and $G(q) \in \Re^{n\times 1}$ represents the torques caused by gravitational force.

The visual servoing controller presenting in the later chapters is based on some properties of robot manipulator dynamic equation in (2.6). The properties are as follows:

Property 1 *The inertia matrix $H(q)$ is symmetric and positive definite which is both upper bounded and below bounded:*

$$\alpha I_n \le H(q) \le \beta I_n$$

where α and β are any scalars satisfying $0 < \alpha < \beta$.

Property 2 *The centripetal and Coriolis torque vector $C(q,\dot{q})$ can be rewritten as:*

$$C(q,\dot{q}) = \{\tfrac{1}{2}\dot{H}(q) + S(q,\dot{q})\}\dot{q}$$

where $\dot{H}(q)$ *is the time derivative of the inertia matrix and* $S(q,\dot{q}) \in \Re^{n\times n}$ *is a skew-symmetric matrix.*

Property 3 *The gravitational torque* $G(q)$ *verifies*

$$\|G(q)\| \leq g_0$$

for some bounded constant $g_0 \leq 0$.

Property 4 *The dynamic equation in (2.6) can be linearized as follows:*

$$H(q)\ddot{q} + C(q,\dot{q}) + G(q) = W_1(q,\dot{q},\ddot{q})\Phi$$

where $\Phi \in \Re^{m\times 1}$ *represents a vector of m uncertain physical dynamics parameters and* $W_1(q,\dot{q},\ddot{q}) \in \Re^{n\times m}$ *represents the regressor matrix that contains known functions which depend on* q, $\dot{q}$ *and* $\ddot{q}$.

2.4 Camera model

The study of the relationships between the robot frame and the camera image frame is important for performing visual servoing. In computer vision communities, variety camera projection models are being used. For example, perspective projection is the model which is most commonly used. Affine projection model is an approximation of perspective projection model. Orthographic, parallel, weak-perspective and paraperspective projection models are derived from affine projection model. The

visual servoing controller presenting in later part of this thesis is based on the perspective projection, which is presented in the following.

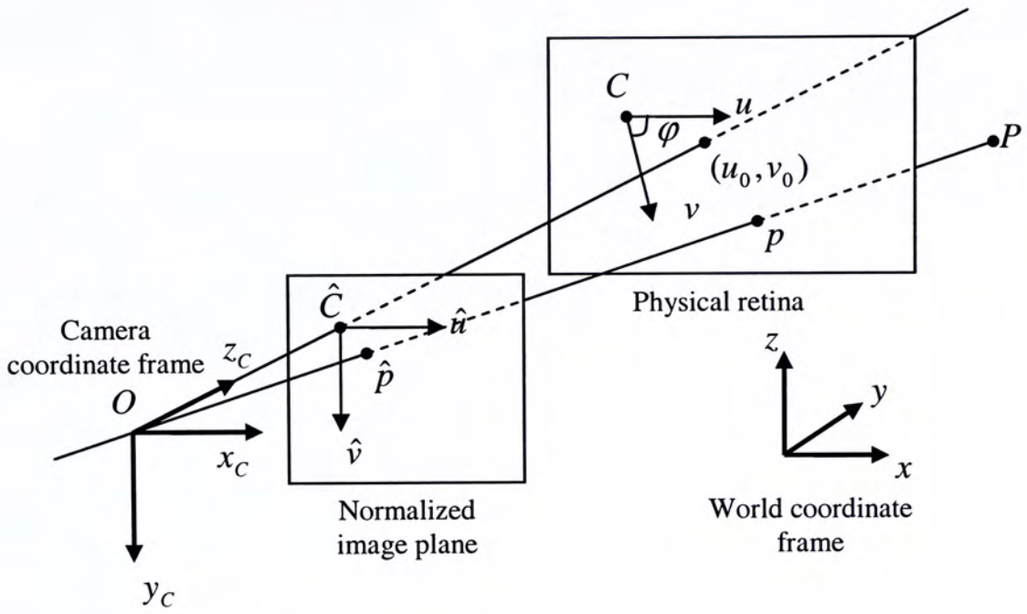


Figure 2.2: Perspective projection

Define position P in the real world with respect to the camera frame as ${}^{C}X = [x_C \quad y_C \quad z_C]$. The relationship between position P and its projected position in the image plane as described by the perspective projection model (Fig. 2.2) can be expressed as the following:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z_C} \begin{bmatrix} fk_u & -fk_u \cos\varphi & u_0 & 0 \\ 0 & fk_v / \sin\varphi & v_0 & 0 \end{bmatrix} \begin{bmatrix} x_C \\ y_C \\ z_C \\ 1 \end{bmatrix} \tag{2.7}$$

where the vector $[u \quad v]^T$ represents the projected image position, u_0 and v_0 are the coordinates of the principal axis, f is the focal length, k_u and k_v are the

magnification factors in $\vec{u}$ and $\vec{v}$ directions respectively, and φ is the angle between the image axes $\vec{u}$ and $\vec{v}$. Re-arrange equation (2.7) becomes:

$$\begin{bmatrix} B_u \\ B_v \end{bmatrix} = \begin{bmatrix} u - u_0 \\ v - v_0 \end{bmatrix} = \begin{bmatrix} fk_u & -fk_u \cos\varphi \\ 0 & fk_v / \sin\varphi \end{bmatrix} \begin{bmatrix} \frac{x_C}{z_C} \\ \frac{y_C}{z_C} \end{bmatrix} \tag{2.8}$$

Define $a_u = fk_u$, $a_v = fk_v / \sin\varphi$, and $\gamma = -fk_v \cos\varphi$, for simplicity. Usually, the image axes can be assumed to be perpendicular with each other. Then (2.8) becomes:

$$\begin{bmatrix} B_u \\ B_v \end{bmatrix} = \begin{bmatrix} a_u & 0 \\ 0 & a_v \end{bmatrix} \begin{bmatrix} \frac{x_C}{z_C} \\ \frac{y_C}{z_C} \end{bmatrix} \tag{2.9}$$

Take time derivative on both sides of equation (2.9):

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} a_u & 0 \\ 0 & a_v \end{bmatrix} \begin{bmatrix} \frac{\dot{x}_C z_C - x_C \dot{z}_C}{z_C^2} \\ \frac{\dot{y}_C z_C - y_C \dot{z}_C}{z_C^2} \end{bmatrix} = \begin{bmatrix} a_u & 0 \\ 0 & a_v \end{bmatrix} \begin{bmatrix} \frac{1}{z_C} & 0 & -\frac{x_C}{z_C^2} \\ 0 & \frac{1}{z_C} & -\frac{y_C}{z_C^2} \end{bmatrix} \begin{bmatrix} \dot{x}_C \\ \dot{y}_C \\ \dot{z}_C \end{bmatrix} \tag{2.10}$$

Equation (2.10) describes the relationships between velocity on image plane and velocity on real world, with respect to the camera coordinate.

2.4.1 Eye-in-hand Configuration

In eye-in-hand camera configuration, the camera frame moves with the end-effector as the environment is static. The velocity of a point with respect to the camera frame is expressed as [8]:

$$^{C}\dot{X} = -T - \Omega \times {}^{C}X \tag{2.11}$$

where T and Ω are the translational velocity and the angular velocity of the camera, respectively, with respect to the camera frame. $\Omega \times {}^{C}X$ can be rewritten as:

$$\Omega \times {}^{C}X = k({}^{C}X)\Omega = \begin{bmatrix} 0 & z_C & -y_C \\ -z_C & 0 & x_C \\ y_C & -x_C & 0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.12}$$

where $k(\cdot)$ is a function which performs cross product. Then equation (2.11) can be re-written as:

$$\begin{bmatrix} \dot{x}_C \\ \dot{y}_C \\ \dot{z}_C \end{bmatrix} = -\begin{bmatrix} T_x + z_C\omega_y - y_C\omega_z \\ T_y - z_C\omega_x + x_C\omega_z \\ T_z + y_C\omega_x - x_C\omega_y \end{bmatrix} \tag{2.13}$$

Substitute equation (2.13) into equation (2.10) and re-arrange becomes:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = -\begin{bmatrix} a_u & 0 \\ 0 & a_v \end{bmatrix} \begin{bmatrix} \frac{1}{z_C} & 0 & -\frac{x_C}{z_C^2} \\ 0 & \frac{1}{z_C} & -\frac{y_C}{z_C^2} \end{bmatrix} \begin{bmatrix} T_x + z_C\omega_y - y_C\omega_z \\ T_y - z_C\omega_x + x_C\omega_z \\ T_z + y_C\omega_x - x_C\omega_y \end{bmatrix}$$

$$= -\begin{bmatrix} a_u & 0 \\ 0 & a_v \end{bmatrix} \begin{bmatrix} \frac{1}{z_C} & 0 & -\frac{x_C}{z_C^2} & -\frac{x_C y_C}{z_C^2} & 1+\frac{x_C^2}{z_C^2} & -\frac{y_C}{z_C} \\ 0 & \frac{1}{z_C} & -\frac{y_C}{z_C^2} & -\left(1+\frac{y_C^2}{z_C^2}\right) & \frac{x_C y_C}{z_C^2} & \frac{x_C}{z_C} \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.14}$$

Then, equation (2.14) can be re-written as:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} -\frac{a_u}{z_C} & 0 & -\frac{B_u}{z_C} & -\frac{B_u B_v}{a_v} & -a_u - \frac{B_u^2}{a_u} & -\frac{a_u B_v}{a_v} \\ 0 & -\frac{a_v}{z_C} & -\frac{B_v}{z_C} & a_v + \frac{B_v^2}{a_v} & \frac{B_u B_v}{a_u} & \frac{a_v B_u}{a_u} \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = J_C \begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (2.15)$$

Now transform the camera velocity from with respect to camera frame to with respect to the manipulator base frame. The transformation can be expressed as:

$$\begin{bmatrix} T_x \\ T_y \\ T_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} {}^C R_B & 0_{3\times3} \\ 0_{3\times3} & {}^C R_B \end{bmatrix} \begin{bmatrix} I_3 & -k({}^B R_E \, {}^B X_E) \\ 0_{3\times3} & I_3 \end{bmatrix} \begin{bmatrix} T_E \\ \Omega_E \end{bmatrix} \quad (2.16)$$

where $T_E = [T_{Ex} \quad T_{Ey} \quad T_{Ez}]^T$ and $\Omega_E = [\Omega_{Ex} \quad \Omega_{Ey} \quad \Omega_{Ez}]^T$. Substitute equation (2.16) into equation (2.15):

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = J_C \begin{bmatrix} {}^C R_B & 0_{3\times3} \\ 0_{3\times3} & {}^C R_B \end{bmatrix} \begin{bmatrix} I_3 & -k({}^B R_E \, {}^B X_E) \\ 0_{3\times3} & I_3 \end{bmatrix} \begin{bmatrix} T_E \\ \Omega_E \end{bmatrix} = J_s \begin{bmatrix} T_E \\ \Omega_E \end{bmatrix} \quad (2.17)$$

where J_s is the image Jacobian matrix.

To ensure the vector $[T_E \quad \Omega_E]^T$ can be fully determined, at least three image feature points are required. With n image feature points, redefine $J_s = [J_{s1} \quad J_{s2} \quad J_{s3} \quad \dots \quad J_{sn}]$. Then, equation (2.17) becomes:

$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \\ \dot{u}_3 \\ \dot{v}_3 \\ \vdots \\ \dot{u}_n \\ \dot{v}_n \end{bmatrix} = J_s \begin{bmatrix} T_{Ex} \\ T_{Ey} \\ T_{Ez} \\ \omega_{Ex} \\ \omega_{Ey} \\ \omega_{Ez} \end{bmatrix} = \begin{bmatrix} J_{s1} \\ J_{s2} \\ J_{s3} \\ \vdots \\ J_{sn} \end{bmatrix} \begin{bmatrix} T_{Ex} \\ T_{Ey} \\ T_{Ey} \\ \omega_{Ex} \\ \omega_{Ey} \\ \omega_{Ez} \end{bmatrix} \tag{2.18}$$

Combine equation (2.18) and equation (2.3), becomes:

$$\dot{\xi} = J_s {}^B\dot{X}_E = J_s J(q)\dot{q} \tag{2.19}$$

where $\dot{\xi} = [\dot{u}_1 \quad \dot{v}_1 \quad \dots \quad \dot{u}_n \quad \dot{v}_n]$ denotes velocities on image feature space. Joint-space velocities can be obtained by taking pseudo-inverse of the image Jacobian J_s and the robot Jacobian $J(q)$:

$$\dot{q} = J^+(q) J_s^+ \dot{\xi} \tag{2.20}$$

where J_s^+ and $J^+(q)$ denote pseudo-inverse of J_s and $J(q)$, respectively.

The image Jacobian matrix J_s consists of the unknown camera intrinsic parameters such as focal length, magnification factors, principal axis position, distortion coefficients, and consists of the extrinsic parameters such as the translation and rotation between the camera coordinate frame and the robot base coordinate frame.

Equations (2.19) and (2.20) show the velocity relationships between image feature space and the joint space. The image Jacobian matrix J_s is highly nonlinear due to perspectivity, radial lens distortion, and quantization between camera and end-effector frame. The robot Jacobian $J(q)$ is a highly coupled and nonlinear function

of joint angles. Some researchers have proposed to estimate a constant matrix on-line to approximate the composite Jacobian matrix by adaptive schemes.

2.4.2 Eye-and-hand Configuration

Assume the end-effector moves in a translational velocity T_E and an angular velocity Ω_E with respect to the robot base frame. Suppose there is a point P in the end-effector, where ${}^E r$ is the coordinate of point P with respect to the origin of the end-effector frame. The velocity of point P with respect to the base frame can be expressed as:

$$ {}^B\dot{P} = \Omega_E \times {}^B R_E \, {}^E r + T_E \tag{2.21} $$

Substitute equation (2.21) into equation (2.10):

$$ \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{1}{z_C} \begin{bmatrix} a_u & 0 & B_u \\ 0 & a_v & B_v \end{bmatrix} {}^C R_B \begin{bmatrix} {}^B\dot{x} \\ {}^B\dot{y} \\ {}^B\dot{z} \end{bmatrix} \tag{2.22} $$

Similar as the previous sub-section, equation (2.22) can be re-arranged as:

$$ \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \frac{1}{z_C} \begin{bmatrix} a_u & 0 & B_u \\ 0 & a_v & B_v \end{bmatrix} {}^C R_B \begin{bmatrix} I & k({}^B R_E \, {}^E r) \end{bmatrix} \begin{bmatrix} T_E \\ \Omega_E \end{bmatrix} = J_s \begin{bmatrix} T_E \\ \Omega_E \end{bmatrix} \tag{2.23} $$

To have the velocity vector $[T_E \quad \Omega_E]^T$ being observed fully, three or more image feature points are required to attach on the end-effector. For visual servoing system with multiple image feature points, equation (2.23) is expressed as follows:

$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \\ \dot{u}_3 \\ \dot{v}_3 \\ \vdots \\ \dot{u}_n \\ \dot{v}_n \end{bmatrix} = J_s \begin{bmatrix} T_E \\ \Omega_E \end{bmatrix} = \begin{bmatrix} J_{s1} \\ J_{s2} \\ J_{s3} \\ \vdots \\ J_{sn} \end{bmatrix} \begin{bmatrix} T_E \\ \Omega_E \end{bmatrix} \tag{2.24}$$

Define:

$$S_i = {}^C R_B \left[I \quad k({}^B R_E \, {}^E r_i) \right], \tag{2.25}$$

Then, the image Jacobian matrix J_s becomes:

$$J_s = \begin{bmatrix} \frac{1}{z_{C1}} \begin{bmatrix} a_u & 0 & u_1 - u_0 \\ 0 & a_v & v_1 - v_0 \end{bmatrix} S_1 \\ \frac{1}{z_{C2}} \begin{bmatrix} a_u & 0 & u_2 - u_0 \\ 0 & a_v & v_2 - v_0 \end{bmatrix} S_2 \\ \frac{1}{z_{C3}} \begin{bmatrix} a_u & 0 & u_3 - u_0 \\ 0 & a_v & v_3 - v_0 \end{bmatrix} S_3 \\ \vdots \\ \frac{1}{z_{Cn}} \begin{bmatrix} a_u & 0 & u_n - u_0 \\ 0 & a_v & v_n - v_0 \end{bmatrix} S_n \end{bmatrix} \tag{2.26}$$

Combine equations (2.3) and (2.24), velocities in image feature space can be expressed in terms of joint velocities:

$$\dot{\xi} = J_s \, {}^B \dot{X}_E = J_s J(q) \dot{q} \tag{2.27}$$

where $\dot{\xi} = \begin{bmatrix} \dot{u}_1 & \dot{v}_1 & \dots & \dot{u}_n & \dot{v}_n \end{bmatrix}^T$ denotes velocities on image feature space. The joint velocities can be obtained by taking pseudo-inverse of the image Jacobian matrix and the robot Jacobian matrix:

$$\dot{q} = J^+(q) J_s^+ \dot{\xi} \tag{2.28}$$

where J_s^+ and $J^+(q)$ denote pseudo-inverse of J_s and $J(q)$, respectively.

Similar as the previous sub-section, the image Jacobian matrix J_s consists of uncalibrated camera intrinsic parameters and extrinsic parameters, which are to be estimated by some adaptive schemes.

Chapter 3

Adaptive Visual Servoing Control

3.1 Controller Design

The controller design in this thesis is based on position control. The objective of the robot task is to move the robot manipulator so that the feature points attached with the end-effector approach to the given desired points, in image plane. In this work, feedback information mainly relies on the image error, which is the difference between the current image feature position and the desired image feature position.

Let $\xi_d = [u_d \quad v_d]^T$ be the desired position with respect to the image frame, the controller is to compute and generate torques for robot joints to move the end-effector, so that the image points attached to the end-effector $\xi = [u \quad v]^T$ reach the desired image position. Notice that the desired joint positions are not available in image-based visual servoing.

The following control law is inspired by the concept of transpose Jacobian control [34, 35]. For a robot manipulator with m degree of freedom and n image feature points, the following control law is proposed:

$$\tau = G(q) - K_v\dot{q} - K_p J^T(q)\left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2}\hat{J}_e^T \right)\Delta\xi \qquad (3.1)$$

where the hat sign "^" denotes estimated parameter or the matrix contains estimated parameter, $G(q)$ is the gravity compensator, K_p and K_v are $m \times m$ symmetric positive definite matrix or positive scalars for proportional (position) gain and derivative (velocity) gain, $J^T(q)$ is the transposed robot Jacobian matrix, $\Delta\xi$ is the feature image position error where $\Delta\xi = \xi - \xi_d$, $D(\hat{z})$ is a $2n \times 2n$ diagonal matrix which contains the estimated depth of each image feature point in the following form:

$$D(\hat{z}) = \begin{bmatrix} \hat{z}_{C1} & & & & & & & & \\ & \hat{z}_{C1} & & & & & & & \\ & & \hat{z}_{C2} & & & & & & \\ & & & \hat{z}_{C2} & & & & & \\ & & & & \hat{z}_{C3} & & & & \\ & & & & & \hat{z}_{C3} & & & \\ & & & & & & \ddots & & \\ & & & & & & & \hat{z}_{Cn} & \\ & & & & & & & & \hat{z}_{Cn} \end{bmatrix} \qquad (3.2)$$

then recall equation (2.26):

$$\hat{J}_s D(\hat{z}) = \begin{bmatrix} \begin{bmatrix} \hat{a}_u & 0 & u_1 - \hat{u}_0 \\ 0 & \hat{a}_v & v_1 - \hat{v}_0 \end{bmatrix} \hat{S}_1 \\ \begin{bmatrix} \hat{a}_u & 0 & u_2 - \hat{u}_0 \\ 0 & \hat{a}_v & v_2 - \hat{v}_0 \end{bmatrix} \hat{S}_2 \\ \begin{bmatrix} \hat{a}_u & 0 & u_3 - \hat{u}_0 \\ 0 & \hat{a}_v & v_3 - \hat{v}_0 \end{bmatrix} \hat{S}_3 \\ \vdots \\ \begin{bmatrix} \hat{a}_u & 0 & u_n - \hat{u}_0 \\ 0 & \hat{a}_v & v_n - \hat{v}_0 \end{bmatrix} \hat{S}_n \end{bmatrix} \qquad (3.3)$$

where $\hat{S}_i = {}^C\hat{R}_B \begin{bmatrix} I & k({}^B R_E \, {}^E r_i) \end{bmatrix}$. The diagonal matrix $D(\hat{z})$ cancels the perspective scaling effect of the image Jacobian matrix $\hat{J}_s$. Also,

$$\hat{J}_e = \begin{bmatrix} \begin{bmatrix} 0 & 0 & u_1 - u_{d1} \\ 0 & 0 & v_1 - v_{d1} \end{bmatrix} \hat{S}_1 \\ \begin{bmatrix} 0 & 0 & u_2 - u_{d2} \\ 0 & 0 & v_2 - v_{d2} \end{bmatrix} \hat{S}_2 \\ \begin{bmatrix} 0 & 0 & u_3 - u_{d3} \\ 0 & 0 & v_3 - v_{d3} \end{bmatrix} \hat{S}_3 \\ \vdots \\ \begin{bmatrix} 0 & 0 & u_n - u_{dn} \\ 0 & 0 & v_n - v_{dn} \end{bmatrix} \hat{S}_n \end{bmatrix} \tag{3.4}$$

where u_{di} and v_{di} are the desired image position of feature image point i in u and v coordinates, respectively.

It is worth noticing that the proposed controller computes the transpose of the robot Jacobian matrix and the transpose of image Jacobian matrix, and its variations. The advantage of such controller over the traditional inverse Jacobian controller is the transpose Jacobian controller can avoid computing inverse of Jacobian matrices in singularity configurations.

Besides the image error $\Delta\xi$, the controller also requires the measurement of joint positions and velocities, the gravitional torque vector $G(q)$, and the estimated non-scaled image Jacobian matrix $\left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2} \hat{J}_e^T \right)$.

Substitute the control torque vector τ of the controller in equation (3.1) into the robot manipulator dynamic equation (2.6), then the closed-loop dynamic equation becomes:

$$H(q)\ddot{q} + C(q,\dot{q}) = -K_v \dot{q} - K_p J^T(q) \left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2} \hat{J}_e^T \right) \Delta\xi \tag{3.5}$$

The above robot dynamics equation contains unknown parameters which are to be determined online.

3.2 Parameter Estimation

As the image Jacobian matrix $\left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2}\hat{J}_e^T\right)$ contains unknown camera intrinsic parameters, and the unknown extrinsic parameters, which must be determined. Then the following proposition is proposed:

Proposition 1: *Arrange the m unknown elements of the product of the image Jacobian matrix and the depth matrix by a* $m\times 1$ *vector* Θ *. The term* $\left(J_s^T D(z) + \frac{1}{2}J_e^T\right)\Delta\xi$ *can be represented as:*

$$\left(J_s^T D(z) + \frac{1}{2}J_e^T\right)\Delta\xi = W(\Delta\xi, u, v, q)\Theta \tag{3.6}$$

where $W(\Delta\xi, u, v, q)$ *is a regressor matrix with elements independent by the unknown parameters. Then*

$$\left(J_s^T D(z) + \frac{1}{2}J_e^T\right)\Delta\xi - \left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2}\hat{J}_e^T\right)\Delta\xi = W(\Delta\xi, u, v, q)\Delta\Theta \tag{3.7}$$

where Θ *is the vector of real parameter,* $\hat{\Theta}$ *is the vector of estimated parameter, and* $\Delta\Theta = \Theta - \hat{\Theta}$ *is the vector of parameter error.*

The transformation relationship between the camera coordinate frame and the robot base coordinate can be expressed as follows:

$$\begin{bmatrix} {}^C x \\ {}^C y \\ {}^C z \end{bmatrix} = {}^C R_B \begin{bmatrix} {}^B x \\ {}^B y \\ {}^B z \end{bmatrix} + {}^C p_B = \begin{bmatrix} {}^C R_{B11} & {}^C R_{B12} & {}^C R_{B13} \\ {}^C R_{B21} & {}^C R_{B22} & {}^C R_{B23} \\ {}^C R_{B31} & {}^C R_{B32} & {}^C R_{B33} \end{bmatrix} \begin{bmatrix} {}^B x \\ {}^B y \\ {}^B z \end{bmatrix} + \begin{bmatrix} {}^C p_{Bx} \\ {}^C p_{By} \\ {}^C p_{Bz} \end{bmatrix} \tag{3.8}$$

Define ${}^{C}R_{B1} = \begin{bmatrix} {}^{C}R_{B11} & {}^{C}R_{B12} \\ {}^{C}R_{B21} & {}^{C}R_{B22} \end{bmatrix}$, ${}^{C}R_{B2} = \begin{bmatrix} {}^{C}R_{B13} \\ {}^{C}R_{B23} \end{bmatrix}$, ${}^{C}R_{B3} = \begin{bmatrix} {}^{C}R_{B31} & {}^{C}R_{B32} \end{bmatrix}$, and ${}^{C}R_{B4} = {}^{C}R_{B33}$. Rearrange equation (3.8) and becomes:

$$\begin{bmatrix} x_C \\ y_C \end{bmatrix} = {}^{C}R_{B1} \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} + {}^{C}R_{B2}\,{}^{B}z + \begin{bmatrix} {}^{C}p_{Bx} \\ {}^{C}p_{By} \end{bmatrix} \tag{3.9}$$

$$z_C = {}^{C}R_{B3} \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} + {}^{C}R_{B4}\,{}^{B}z + {}^{C}p_{Bz} \tag{3.10}$$

From the perspective projection model in equation (2.9), the following approach of estimating unknown parameters to reach to the actual parameters up to a scale is proposed. The equation is defined as the following:

$$\hat{z}_C \begin{bmatrix} u \\ v \end{bmatrix} - \left[\begin{bmatrix} \hat{a}_u & 0 \\ 0 & \hat{a}_v \end{bmatrix} \begin{bmatrix} x_C \\ y_C \end{bmatrix} + \hat{z}_C \begin{bmatrix} \hat{u}_0 \\ \hat{v}_0 \end{bmatrix} \right]$$

$$= \hat{z}_C \begin{bmatrix} u \\ v \end{bmatrix} - \left[\hat{H} \begin{bmatrix} x_C \\ y_C \end{bmatrix} + \hat{z}_C \begin{bmatrix} \hat{u}_0 \\ \hat{v}_0 \end{bmatrix} \right]$$

$$= -[z_C - \hat{z}_C] \begin{bmatrix} u \\ v \end{bmatrix} + z_C \begin{bmatrix} u \\ v \end{bmatrix} - \left[\hat{H} \left[{}^{C}\hat{R}_{B1} \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} + {}^{C}\hat{R}_{B2}\,{}^{B}z + \begin{bmatrix} {}^{C}\hat{p}_{Bx} \\ {}^{C}\hat{p}_{By} \end{bmatrix} \right] + \hat{z}_C \begin{bmatrix} \hat{u}_0 \\ \hat{v}_0 \end{bmatrix} \right]$$

$$= -[z_C - \hat{z}_C] \begin{bmatrix} u \\ v \end{bmatrix} + \left[H \left[{}^{C}R_{B1} \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} + {}^{C}R_{B2}\,{}^{B}z + \begin{bmatrix} {}^{C}p_{Bx} \\ {}^{C}p_{By} \end{bmatrix} \right] + z_C \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \right]$$
$$- \left[\hat{H} \left[{}^{C}\hat{R}_{B1} \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} + {}^{C}\hat{R}_{B2}\,{}^{B}z + \begin{bmatrix} {}^{C}\hat{p}_{Bx} \\ {}^{C}\hat{p}_{By} \end{bmatrix} \right] + \hat{z}_C \begin{bmatrix} \hat{u}_0 \\ \hat{v}_0 \end{bmatrix} \right]$$

$$= -[z_C - \hat{z}_C] \begin{bmatrix} u \\ v \end{bmatrix} + \left[H\,{}^{C}R_{B1} - \hat{H}\,{}^{C}\hat{R}_{B1} \right] \begin{bmatrix} {}^{B}x \\ {}^{B}y \end{bmatrix} + \left[H\,{}^{C}R_{B2} - \hat{H}\,{}^{C}\hat{R}_{B2} \right] {}^{B}z$$
$$+ \left[H \begin{bmatrix} {}^{C}p_{Bx} \\ {}^{C}p_{By} \end{bmatrix} - \hat{H} \begin{bmatrix} {}^{C}\hat{p}_{Bx} \\ {}^{C}\hat{p}_{By} \end{bmatrix} \right] + \left[z_C \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} - \hat{z}_C \begin{bmatrix} \hat{u}_0 \\ \hat{v}_0 \end{bmatrix} \right]$$

$$= Y_1(u,v)\Delta\Theta_1 + Y_2(q)\Delta\Theta_2 + Y_3(q)\Delta\Theta_3 + Y_4\Delta\Theta_4 + Y_5(q)\Delta\Theta_5$$

$$= Y_6(u,v,q)\Delta\Theta \tag{3.11}$$

Here, $\Delta\Theta$ can be arranged as follows:

$$\Delta\Theta = \begin{bmatrix} (a_u {}^C R_{B11} + u_0 {}^C R_{B31}) - (\hat{a}_u {}^C \hat{R}_{B11} + \hat{u}_0 {}^C \hat{R}_{B31}) \\ (a_u {}^C R_{B12} + u_0 {}^C R_{B32}) - (\hat{a}_u {}^C \hat{R}_{B12} + \hat{u}_0 {}^C \hat{R}_{B32}) \\ (a_u {}^C R_{B13} + u_0 {}^C R_{B33}) - (\hat{a}_u {}^C \hat{R}_{B13} + \hat{u}_0 {}^C \hat{R}_{B33}) \\ (a_u {}^C p_{Bx} + u_0 {}^C p_{Bz}) - (\hat{a}_u {}^C \hat{p}_{Bx} + \hat{u}_0 {}^C \hat{p}_{Bz}) \\ (a_v {}^C R_{B21} + v_0 {}^C R_{B31}) - (\hat{a}_v {}^C \hat{R}_{B21} + \hat{v}_0 {}^C \hat{R}_{B31}) \\ (a_v {}^C R_{B22} + v_0 {}^C R_{B31}) - (\hat{a}_v {}^C \hat{R}_{B22} + \hat{v}_0 {}^C \hat{R}_{B31}) \\ (a_v {}^C R_{B23} + v_0 {}^C R_{B31}) - (\hat{a}_v {}^C \hat{R}_{B23} + \hat{v}_0 {}^C \hat{R}_{B31}) \\ (a_v {}^C p_{By} + v_0 {}^C p_{Bz}) - (\hat{a}_v {}^C \hat{p}_{By} + \hat{v}_0 {}^C \hat{p}_{Bz}) \\ {}^C R_{B31} - {}^C \hat{R}_{B31} \\ {}^C R_{B32} - {}^C \hat{R}_{B32} \\ {}^C R_{B33} - {}^C \hat{R}_{B33} \\ {}^C p_{Bz} - {}^C \hat{p}_{Bz} \end{bmatrix} \tag{3.12}$$

For the visual servoing system with n feature points and m unknown parameters, equation (3.11) can be extended to:

$$\begin{bmatrix} \hat{z}_{C1}u_1 \\ \hat{z}_{C1}v_1 \\ \hat{z}_{C2}u_2 \\ \hat{z}_{C2}v_2 \\ \hat{z}_{C3}u_3 \\ \hat{z}_{C3}v_3 \\ \vdots \\ \hat{z}_{Cn}u_n \\ \hat{z}_{Cn}v_n \end{bmatrix} - \begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} \hat{H} & 0 & 0 & \cdots & 0 \\ 0 & \hat{H} & 0 & \cdots & 0 \\ 0 & 0 & \hat{H} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \hat{H} \end{bmatrix} \begin{bmatrix} x_{C1} \\ y_{C1} \\ x_{C2} \\ y_{C2} \\ x_{C3} \\ y_{C3} \\ \vdots \\ x_{Cn} \\ y_{Cn} \end{bmatrix} + \begin{bmatrix} \hat{z}_{C1}\hat{u}_0 \\ \hat{z}_{C1}\hat{v}_0 \\ \hat{z}_{C2}\hat{u}_0 \\ \hat{z}_{C2}\hat{v}_0 \\ \hat{z}_{C3}\hat{u}_0 \\ \hat{z}_{C3}\hat{v}_0 \\ \vdots \\ \hat{z}_{Cn}\hat{u}_0 \\ \hat{z}_{Cn}\hat{v}_0 \end{bmatrix} = Y(u,v,q)\Delta\Theta \tag{3.13}$$

where $= Y(u,v,q)$ is a $2n \times m$ regressor matrix which is independent of the unknown parameters. In order to guarantee the parameter can be fully estimated, $n \geq m/2$ image feature points are required.

Proposition 2: *If seven points, which are not coplanar in space, are selected, the equation*

$$Y(u,v,q)\Delta\Theta = 0$$

is equivalent to that the parameters can be estimated up to a scale

$$\hat{\Theta} = \lambda\Theta$$

where λ *is a scalar.*

The above result is well known in computer vision. For details, please refer to the book [36].

Here, the adaptive estimation scheme is proposed as the following:

$$\Delta\dot{\Theta}^T = -\dot{q}^T K_p J^T(q) W(\Delta\xi,u,v,q) K_1^{-1} - \Delta\Theta^T Y^T(u,v,q) Y(u,v,q) K_2 \tag{3.14}$$

where K_1 and K_2 are either positive scalars or positive definite diagonal matrices.

3.3 Stability Analysis

The following section presents the stability analysis of the proposed controller (3.1) and the adaptive estimation method (3.14). Details of the control theories used in stability analysis can be referred to Appendix A.

Define the function as the following:

$$V = \frac{1}{2}\dot{q}^T H(q)\dot{q} + \frac{1}{2}\Delta\xi^T K_p D(z)\Delta\xi + \frac{1}{2}\Delta\Theta^T K_1 \Delta\Theta \tag{3.15}$$

From Property 1 of robot dynamics, $H(q)$ is positive definite, $D(z)$ is positive definite for $z > 0$, and also $K_1 > 0$; therefore, V is a positive definite function. Differentiate the above equation (3.15) becomes:

$$\dot{V} = \dot{q}^T H(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{H}(q)\dot{q} + \Delta\dot{\xi}^T K_p D(z)\Delta\xi + \frac{1}{2}\Delta\xi^T K_p D(\dot{z})\Delta\xi + \Delta\dot{\Theta}^T K_1 \Delta\Theta \tag{3.16}$$

From Property 2 of robot dynamics:

$$\dot{q}^T H(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{H}(q)\dot{q} = \dot{q}^T H(q)\ddot{q} + \dot{q}^T C(q,\dot{q}) \tag{3.17}$$

Then, substitute the closed-loop dynamic equation (3.5) into equation (3.17) becomes:

$$\dot{q}^T H(q)\ddot{q} + \frac{1}{2}\dot{q}^T \dot{H}(q)\dot{q} = -\dot{q}^T K_v \dot{q} - \dot{q}^T K_p J^T(q)\left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2}\hat{J}_e^T\right)\Delta\xi \tag{3.18}$$

From equation (2.27) and $\dot{\xi}_d = 0$, the followings are obtained:

$$\Delta\dot{\xi}^T K_p D(z)\Delta\xi = \dot{\xi}^T K_p D(z)\Delta\xi = \dot{q}^T K_p J^T(q) J_s^T D(z)\Delta\xi \tag{3.19}$$

and

$$\frac{1}{2}\Delta\xi^T K_p D(\dot{z})\Delta\xi = \frac{1}{2}K_p \dot{z} D(\Delta\xi)\Delta\xi = \frac{1}{2}\dot{q}^T K_p J^T(q) J_e^T \Delta\xi \tag{3.20}$$

From the adaptive updating formula (3.14), the following is obtained:

$$\Delta\dot{\Theta}^T K_1 \Delta\Theta = -\dot{q}^T K_p J^T(q) W(\Delta\xi, u, v, q)\Delta\Theta - \Delta\Theta^T Y^T(u,v,q) Y(u,v,q) K_2 K_1 \Delta\Theta \tag{3.21}$$

Substitute equations (3.18), (3.19), (3.20), and (3.21) into (3.16):

$$\begin{aligned}\dot{V} = {} & -\dot{q}^T K_v \dot{q} - \dot{q}^T K_p J^T(q)\left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2}\hat{J}_e^T\right)\Delta\xi \\ & + K_p \dot{q}^T J^T(q) J_s^T D(z)\Delta\xi + \frac{1}{2}\dot{q}^T K_p J^T(q) J_e^T \Delta\xi \\ & - \dot{q} K_p J^T(q) W(\Delta\xi, u, v, q)\Delta\Theta - \Delta\Theta^T Y^T(u,v,q) Y(u,v,q) K_2 K_1 \Delta\Theta\end{aligned}$$

$$= -\dot{q}^T K_v \dot{q} - \dot{q}^T K_p J^T(q)\left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2}\hat{J}_e^T \right)\Delta\xi + \dot{q}^T K_p J^T(q)\left(J_s^T D(z) + \frac{1}{2} J_e^T \right)\Delta\xi$$

$$- \dot{q} K_p J^T(q) W(\Delta\xi, u, v, q)\Delta\Theta - \Delta\Theta^T Y^T(u, v, q) Y(u, v, q) K_2 K_1 \Delta\Theta \tag{3.22}$$

From equation (3.7), equation (3.22) becomes:

$$\dot{V} = -\dot{q}^T K_v \dot{q} + \dot{q}^T K_p J^T(q) W(\Delta\xi, u, v, q)\Delta\Theta - \dot{q}^T K_p J^T(q) W(\Delta\xi, u, v, q)\Delta\Theta$$
$$- \Delta\Theta^T Y^T(u, v, q) Y(u, v, q) K_2 K_1 \Delta\Theta$$

$$= -\dot{q}^T K_v \dot{q} - \Delta\Theta^T Y^T(u, v, q) Y(u, v, q) K_2 K_1 \Delta\Theta \tag{3.23}$$

Then, $\dot{V}$ is a non-positive function, and V is a positive function. Therefore, V is said to be a Lyapunov function. From LaSalle theorem, it is known that $\lim_{t\to\infty} \dot{q} \to 0$ and $\lim_{t\to\infty} Y(u, v, q)\Delta\Theta \to 0$. It also implies that the estimated parameters will be asymptotically converged to the actual values up to a scale. Also, the image error $\Delta\xi$ will asymptotically converge to zero if the matrix $\left(\hat{J}_s^T D(\hat{z}) + \frac{1}{2}\hat{J}_e^T \right)$ is non-singular.

Chapter 4

Experimental Studies

This chapter presents the extensive experimental results of the controller proposed in Chapter 3. The presentation of experimental work is arranged in the following sections:

a) Setup of experimental platform;

b) Comparison of different proportional gains and derivative gains;

c) Comparison among different adaptive gains;

d) Comparison between the presence and absence of gravity compensator;

e) Comparison between current image position only and including previous image positions.

f) Comparison between the proposed controller and the kinematic controllers;

4.1 Experimental Setup

4.1.1 Hardware Setup

An experimental platform of visual servoing system consists of visual image processing part and motion control part. Both parts are cooperated by a visual servoing program in a PC which contains an Intel Pentium III 733 MHz processor.

The visual image processing part is implemented by placing a camera approximately 1.8 metres away from the robot manipulator (Fig. 4.1), in order to obtain visual information to control the manipulator. The camera is a Pulnix TMC-76 high-resolution 2/3" CCD camera. The image is in PAL format with $768(H)\times576(V)$ pixel array. Then the image is captured by Matrox Pulsar frame grabber which has a frame rate of 25 frames per second (the sampling period is 40 milliseconds). The captured image is then processed and control signal is generated.

In the motion control part, the control output register (16-bit) from the controller program is converted to DAC voltage signal by a motion control board, which is made by Googol Technology Limited. The DAC voltage signal is then converted to motor current by Maxon servo-amplifiers. The current then drives the motors of the robot manipulator. The robot manipulator is developed by the Robot Control Laboratory in the Chinese University of Hong Kong. It has three degree-of-freedom in Puma configuration. The motors are Maxon DC motors with encoders attached.

Figure 4.1: Setup of experimental platform.

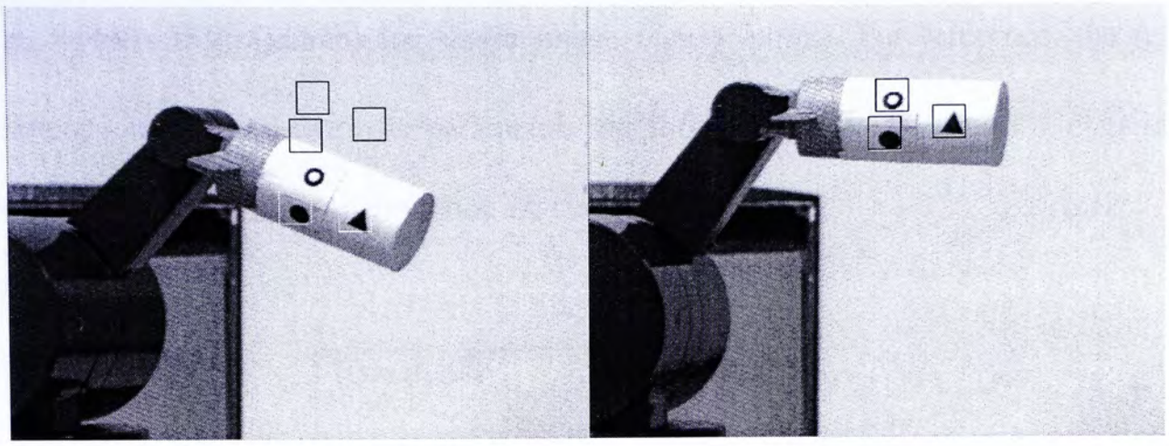
4.1.2 Image Pattern Recognition

The image feature position can be extracted by an image pattern recognition algorithm, which is implemented using an image pattern recognition toolbox from the Matrox Imaging Library (MIL).

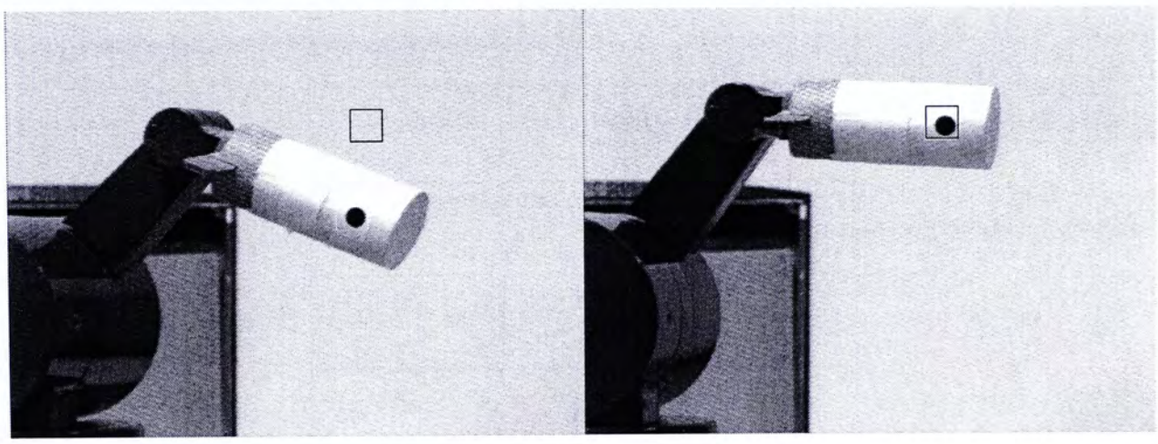
First, the image feature point is placed under a black square. Then, the feature point is recorded as a reference image template. When the manipulator is moving, the image recognition function would search the corresponding image feature point. Assume the moving speed of manipulator is not very fast, the searching range can be narrowed down as some neighbourhood around the previous image position. As a hundred percent match of the moving image feature with the feature in the template is almost impossible, a threshold match score is set for comparing the similarity between both moving image on the end-effector and the reference image template. If the similarity is higher than the threshold match score, a match is considered and the position of the image feature point is extracted. The extracted image feature position will feedback to the visual servoing system.

4.1.3 Experimental Task

The task in the following experiments is to move the manipulator so that the feature points attached at the end-effector reach the desired positions (Fig. 4.2(a) and 4.2(b)). First, move the manipulator to a particular position. Then, mark the position of feature points in the image screen. These image positions are desired image positions. Then, move the manipulator to another position, which will be the initial position for experiments. Finally, run the controller program to evaluate the performance of the controller.



(a) (b)



(c) (d)

Figure 4.2: The robot manipulator move from the initial position (a & c) to the desired position (b & d).

For the following experiments, the initial joint position vector is $q_0 = [1.2 \quad 0.8 \quad 0.4]^T$ (radian) which gives the initial image position vector at $\xi_0 = [456.75 \quad 276.50 \quad 400.27 \quad 222.56 \quad 376.05 \quad 269.08]^T$ (pixels) ($\xi_0 = [456.75 \quad 276.50]^T$ (pixels) for single image feature point); and the desired joint position vector is $q_d = [1.5 \quad 1.0 \quad 0.5]^T$ (radian) which gives the desired image position vector at $\xi_d = [480 \quad 155 \quad 405 \quad 120 \quad 395 \quad 170]^T$ (pixels)

($\xi_d = [480 \quad 155]^T$ (radian) for single image feature point). For reference, the real camera intrinsic and extrinsic parameters are shown in Table 4.1; and the dynamic parameters of the robot manipulator are shown in Table 4.2:

Parameter	Real Value
${}^{C}R_B$	$\begin{bmatrix} 0.997 & -0.083 & 0 \\ 0 & 0 & -1 \\ 0.083 & 0.997 & 0 \end{bmatrix}$
${}^{C}p_B$	$[-0.15 \quad 0 \quad 1.8]^T$
a_u	4763
a_v	4515
u_0	550
v_0	300

Table 4.1: Real values of the camera intrinsic and extrinsic parameters

Link	Length (m)	Mass (kg)
2	0.145	0.167
3	0.1285	0.096

Table 4.2: Dynamic parameters of the robot manipulator

The true values of adaptive parameter vector then become:

$$\Theta = [4792.27 \quad 151.17 \quad 0 \quad -166.36 \quad 25 \quad 298.96 \quad -4515 \quad 540 \quad 0.083 \quad 0.997 \quad 0 \quad 1.8]^T$$

4.2 Control Performance with Different Proportional Gains and Derivative Gains

For the experiment of different proportional gain, the control gains are set as the following: $K_v = 50N \cdot m \cdot s / rad$, $K_1 = 20$ and $K_2 = 0.025$. The initial estimated parameters are set as follows: ${}^C\hat{R}_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, ${}^C\hat{p}_B = [0 \quad 0 \quad 3]^T$, $\hat{a}_u = 4763$, $\hat{a}_v = 4515$, $\hat{u}_0 = 550$, and $\hat{v}_0 = 300$. Using two different K_p , which are $0.0001N \cdot m / pixels$ and $0.00005N \cdot m / pixels$, the results are shown in Figure 4.3.

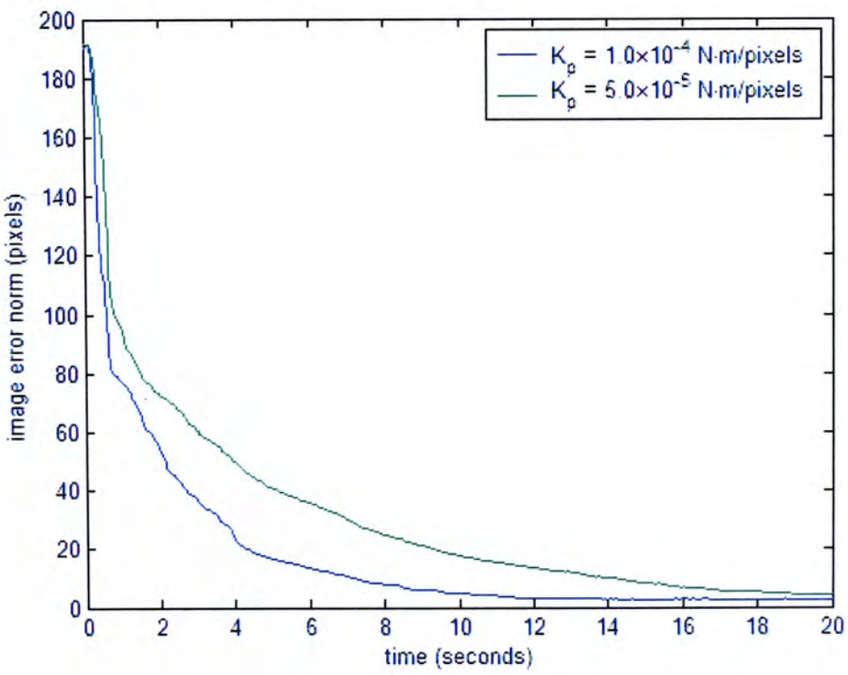


Figure 4.3: Experimental result with two different K_p

From Figure 4.4, it is shown that with larger K_p , the image position error $\Delta\xi$ converges faster.

For the experiment with different derivative gain K_v, the control gain and the parameters are almost the similar as in the proportional gain experiment, except $K_p = 0.0001N \cdot m / pixels$, and K_v is set as $50N \cdot m \cdot s / rad$ and $35N \cdot m \cdot s / rad$. The result is shown in Figure 4.4.

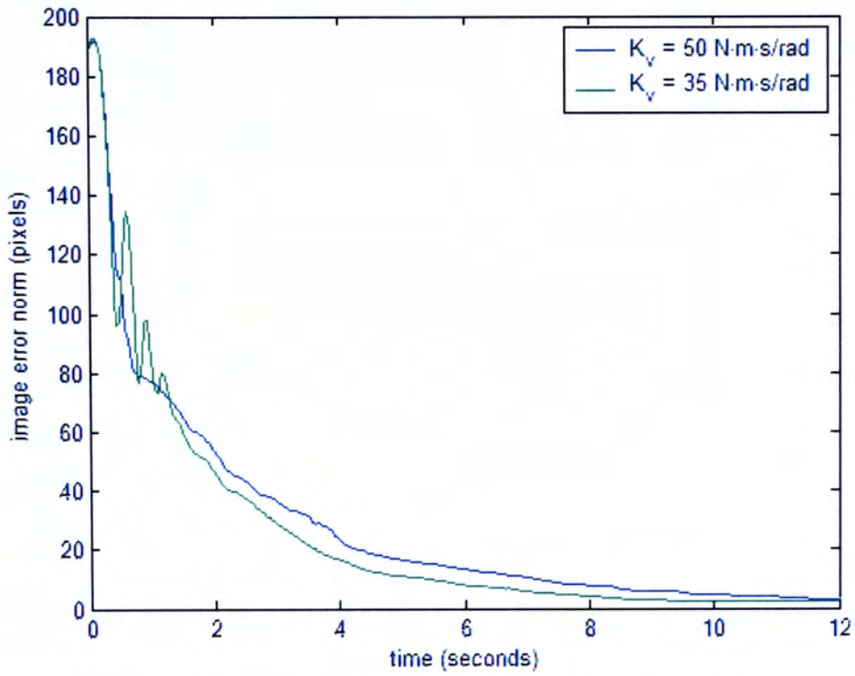


Figure 4.4: Experimental result with two different K_v.

From the above figure, it is shown that a smaller K_v gain will give more oscillation in the trajectory converging to the desired image position.

4.3 Control Performance with Different Adaptive Gains

For the following experiment using single image feature point, the control gains are set as the following: $K_p = 0.0003N \cdot m / pixels$, $K_v = 90N \cdot m \cdot s / rad$, and $K_2 = 0.05$. The initial estimated parameters are set as follows: ${}^{C}\hat{R}_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, ${}^{C}\hat{p}_B = [0 \quad 0 \quad 3]^T$, $\hat{a}_u = 4763$, $\hat{a}_v = 4515$, $\hat{u}_0 = 550$, and $\hat{v}_0 = 300$. With varying K_1, which has values of 12, 16, and 20; the controller performance is shown as follows (Fig. 4.5):

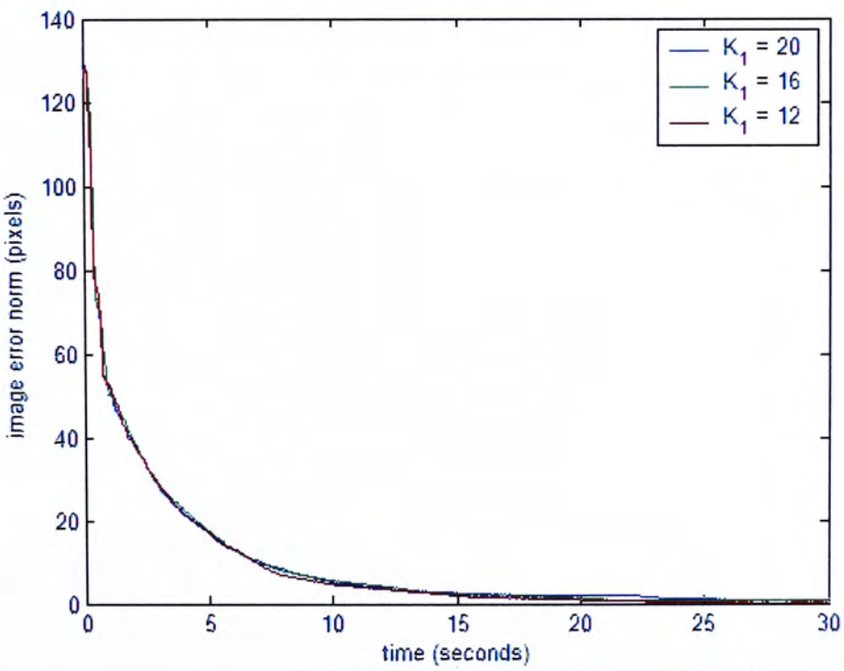


Figure 4.5: Control performances with varying K_1

As seen from the above figure, the performance does not differ greatly. However, there could be difference in the converging response of adaptive parameters. The convergences of adaptive parameters are shown in Figure 4.6 (a) and (b).

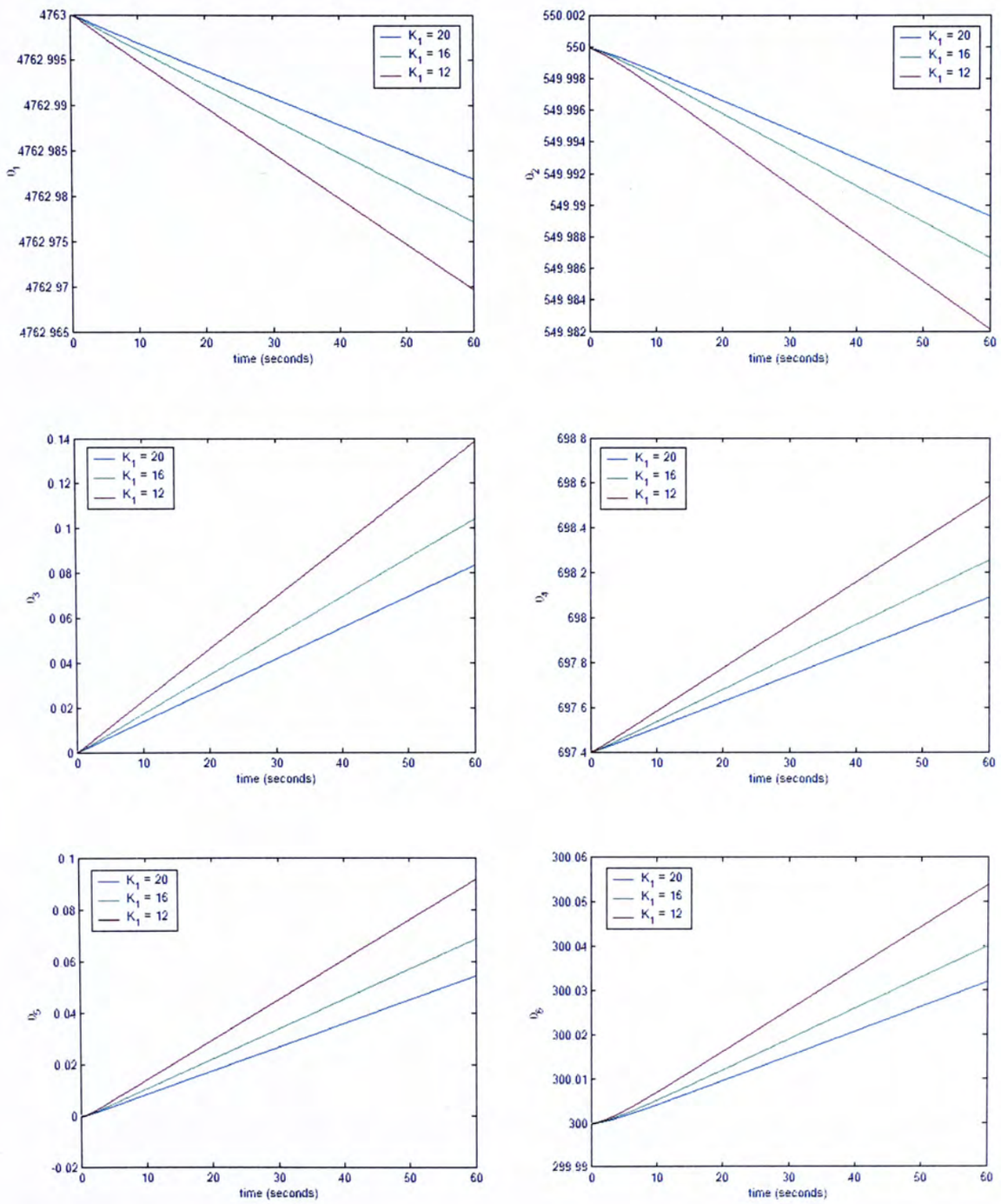


Figure 4.6a: Updating behavoir of first six adaptive parameters of varying K_1

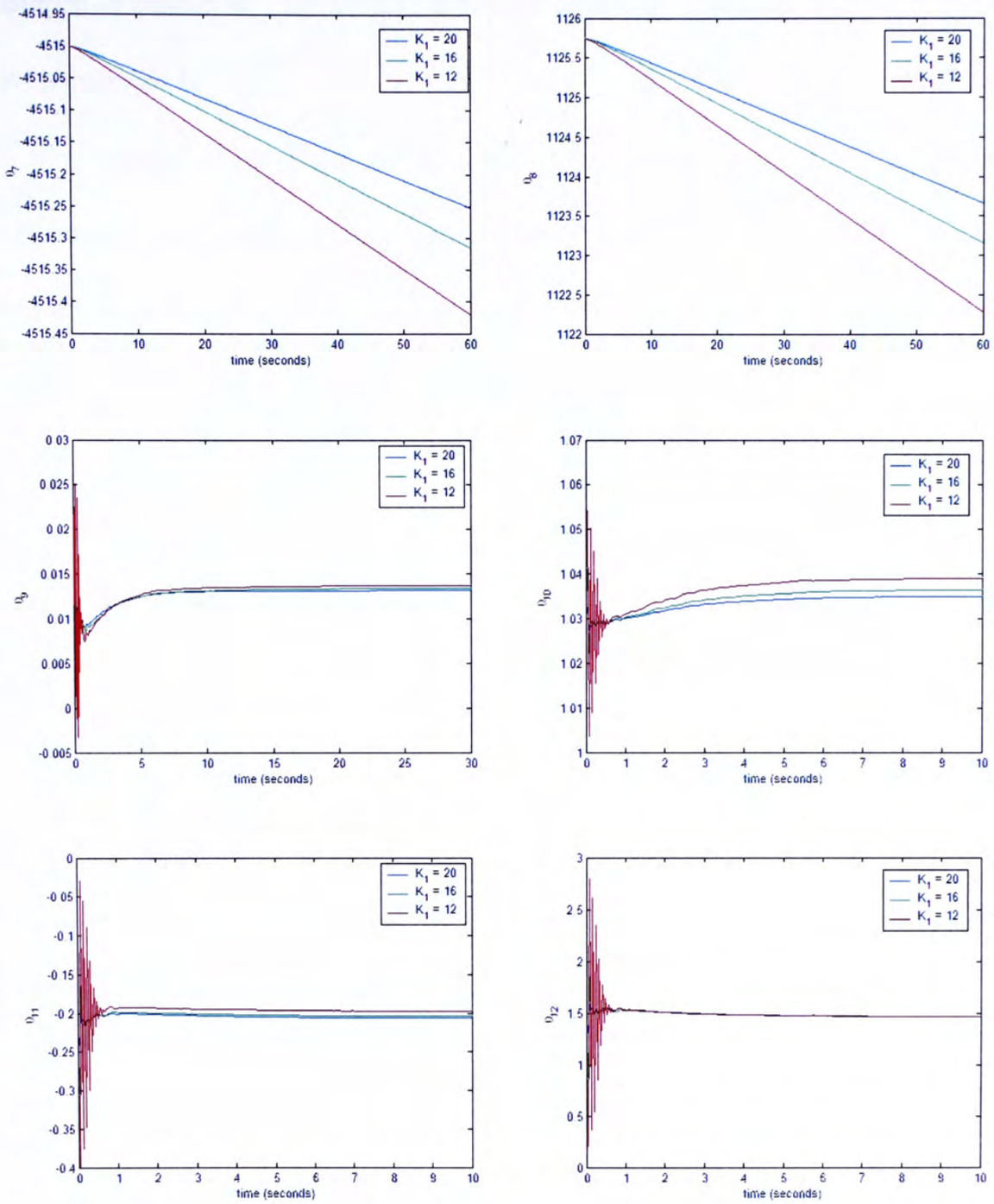


Figure 4.6b: Updating behavoir of last six adaptive parameters of varying K_1

From Figure 4.11, it is shown that different K_1 could result in different converging response in adaptive parameters. Faster converging rate of adaptive parameters is obtained by smaller K_1. Oscillation in adaptive parameters may also

happen in smaller K_1. Eventually, the parameters may converge to different values with different K_1.

Similar result could be obtained in the experiment of K_2 in adaptive algorithm. The control gains and adaptive parameters are similar as above except $K_1 = 20$, $K_p = 0.0004N \cdot m / pixels$. With K_2 equals to 0.025, 0.0025, 0.00025, and 0.000025, the result is shown as follows (figure 4.7):

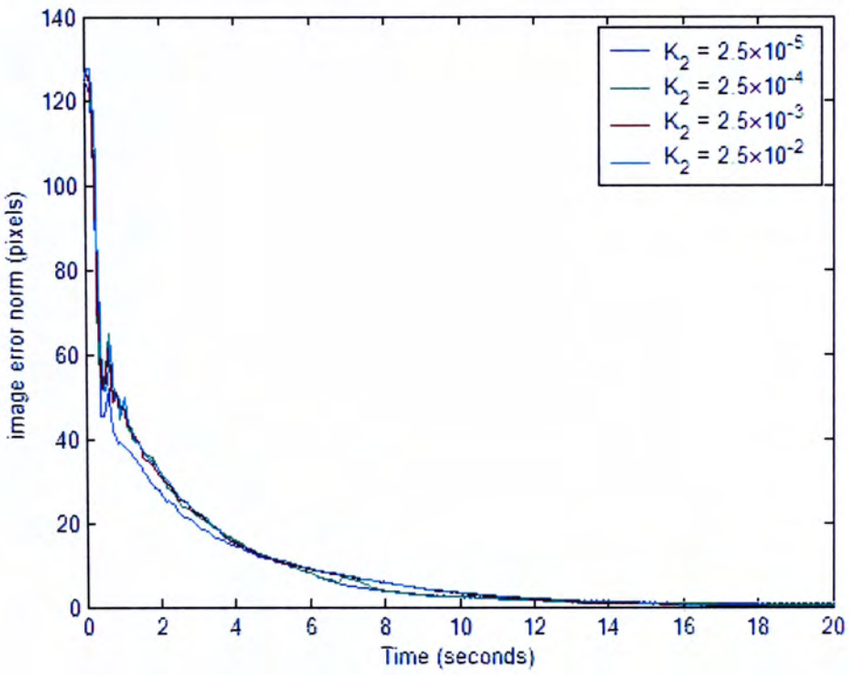


Figure 4.7: Control performances with varying K_2

Similar as the experiment on adaptive gain K_1, from the above figure, the control performance does not affect much by different K_2. Also, similar as the experiment on K_1, different K_2 can make different converging rate of the adaptive parameters, the result is shown in Figure 4.8 (a) and (b).

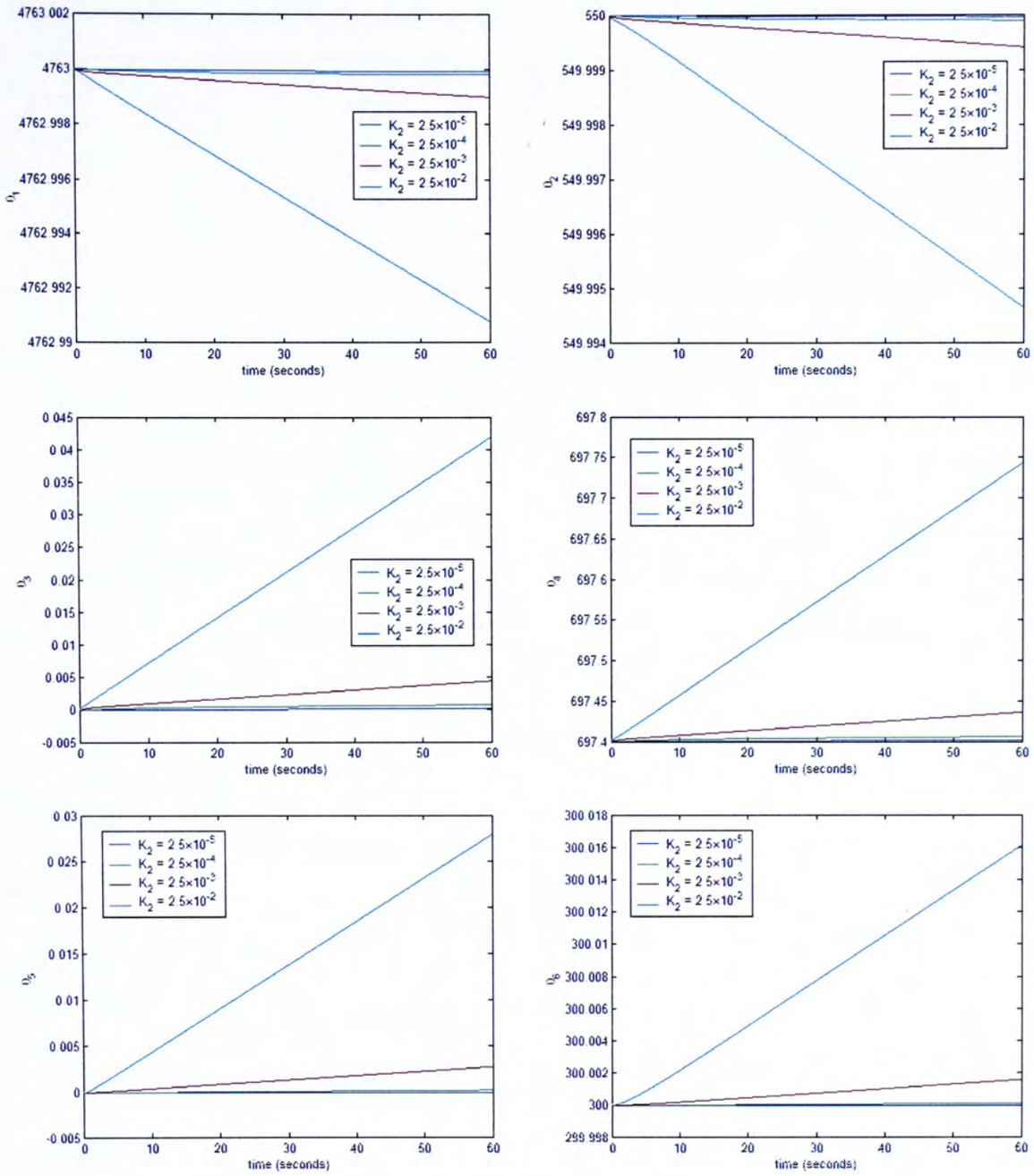


Figure 4.8a: Updating behavoir of first six adaptive parameters of varying K_2

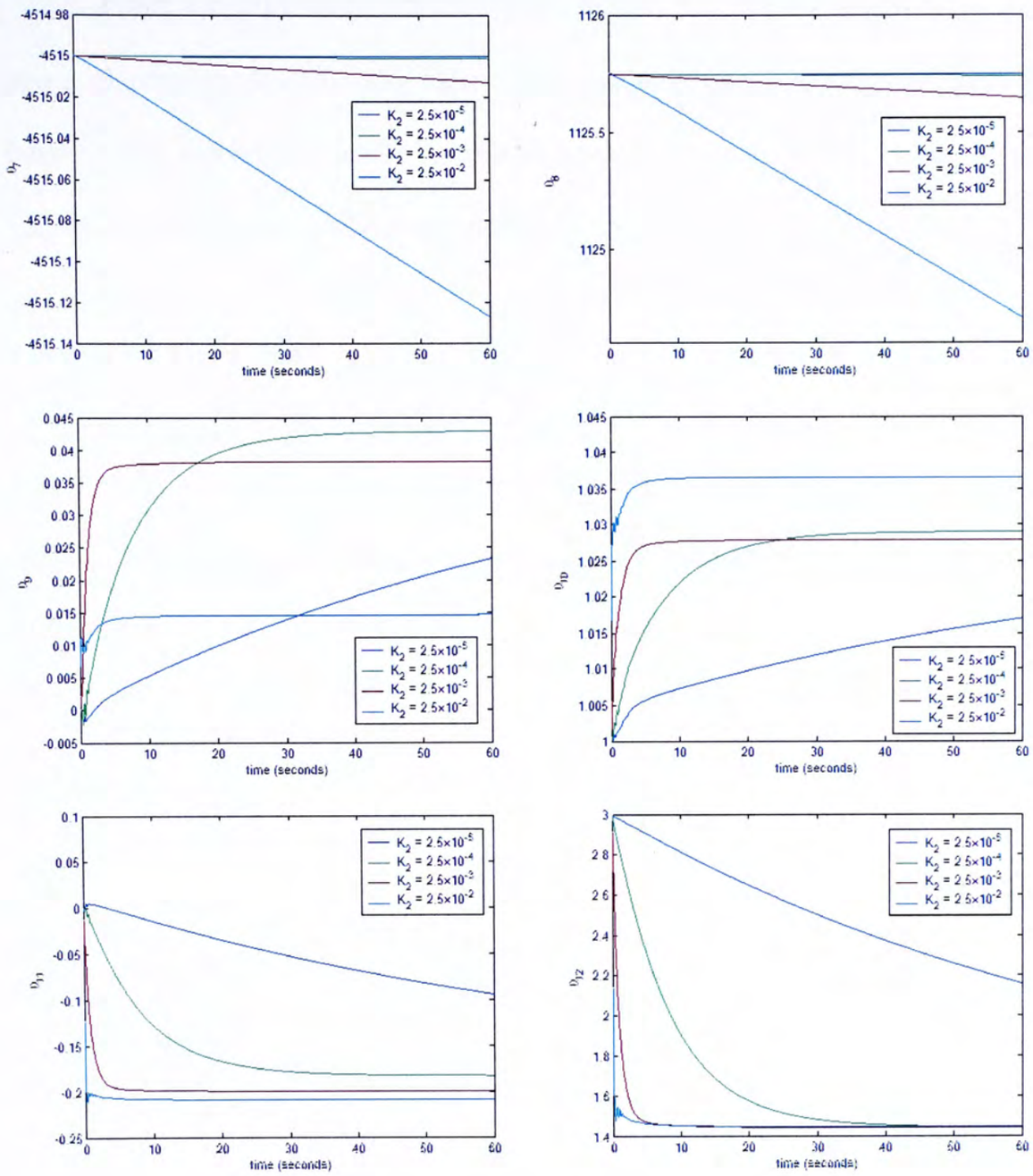


Figure 4.8b: Updating behavoir of last six adaptive parameters of varying K_2

From the above figure, it is shown that for larger K_2, the converging rate of adaptive parameters is faster. Also, oscillation may be generated in the converging of adaptive parameters with larger K_2. Again, with different K_2, the adaptive parameters may have converged to different values.

From the experiment results shown before, for the uniformed K_1 and K_2, some adaptive parameters may have faster converge rate, and some of them may have slower converge rate. To ensure the parameters can be converged to some values, the adaptive parameters are changed to the following:

$K_1 = diag[2.0\times10^{-5},2.0\times10^{-5},2.0\times10^{-5},2.0\times10^{-5},2.0\times10^{-5},2.0\times10^{-5},$
$2.0\times10^{-5},2.0\times10^{-5},2.0,2.0,2.0,2.0]$

$K_2 = diag[12.5,12.5,12.5,12.5,12.5,12.5,12.5,12.5,$
$1.25\times10^{-4},1.25\times10^{-4},1.25\times10^{-4},1.25\times10^{-4}]$

The control performance is shown in the following figure (Fig. 4.9):

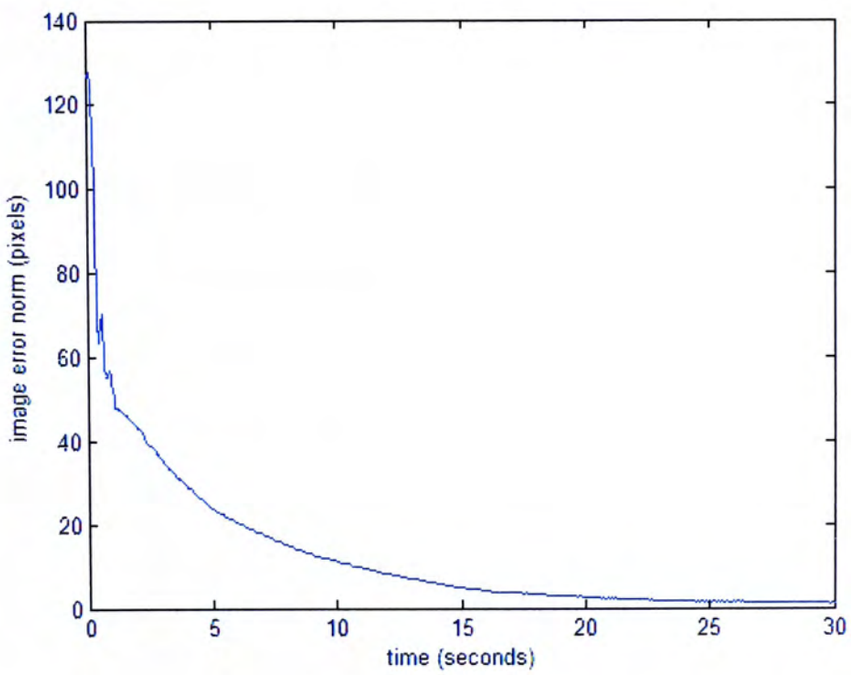


Figure 4.9: Control performance of non-unified K_1 and K_2

The convergences of adaptive parameters are shown as follows (Fig. 4.10 (a) and (b)):

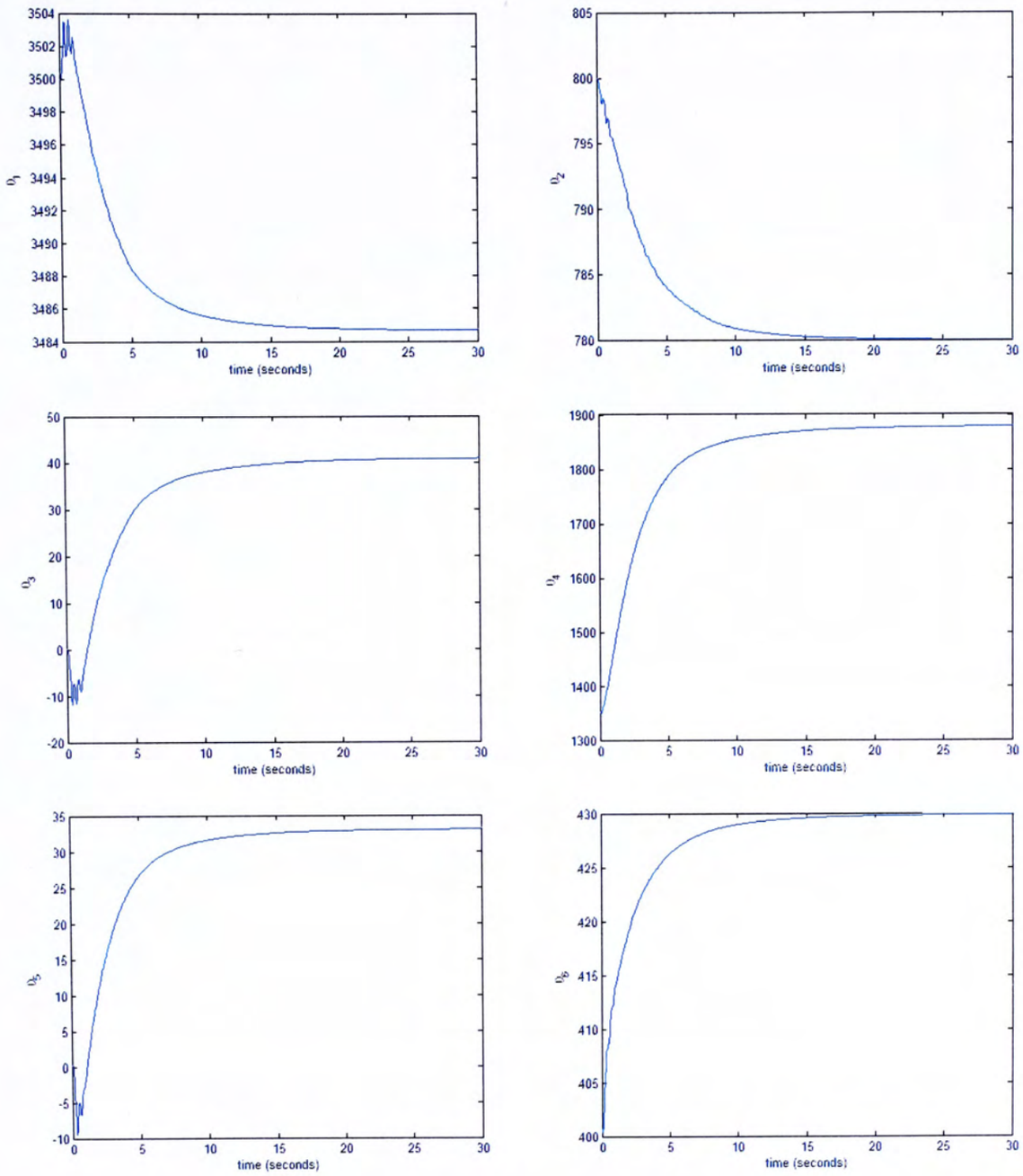


Figure 4.10a: Convergence of first six adaptive parameters

with non-unified K_1 and K_2

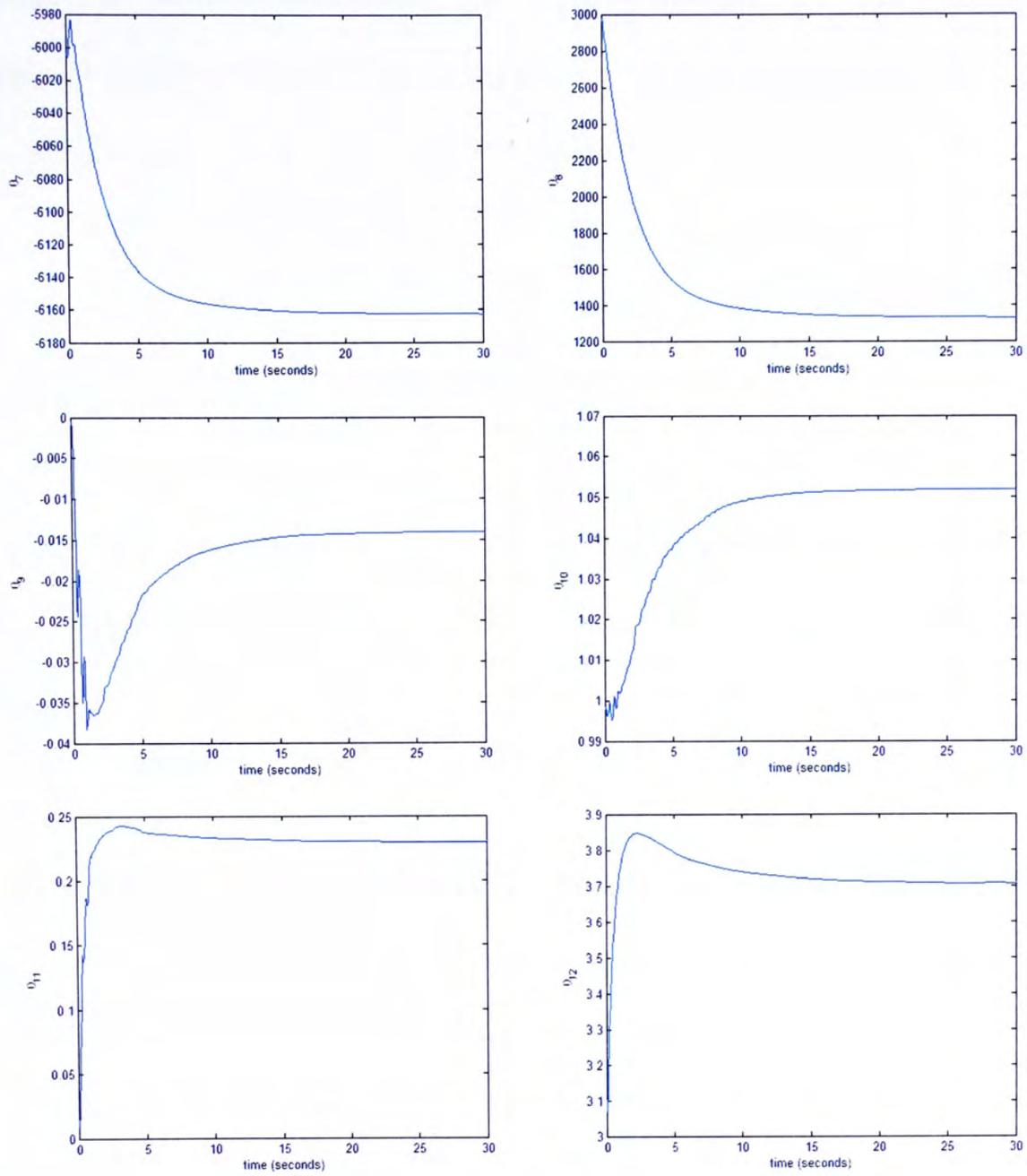


Figure 4.10b: Convergence of last six adaptive parameters

with non-unified K_1 and K_2

From the above figure, it is shown that the adaptive parameters are converged to the following values:

$$\hat{\Theta} = [3484.71, 780.05, 41.06, 1880.00, 33.22, 429.97,$$
$$-6162.60, 1329.45, -0.014, 1.05, 0.23, 3.71]^T$$

Here the adaptive parameters are updated on-line by the algorithm $\Theta(t+\Delta t) = \Theta(t) + \Delta\dot{\Theta}(t)\Delta t$. The parameters may not have converged to the true value up to a scale, as Δt is large (40 milliseconds).

4.4 Gravity Compensator

For the experiment of gravity compensator, the control gains are set as the following: $K_p = 0.00005N \cdot m/pixels$, $K_v = 100N \cdot m \cdot s/rad$, $K_1 = 20$ and $K_2 = 0.0025$. The initial estimated parameters are set as follows: ${}^C\hat{R}_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, ${}^C\hat{p}_B = [0 \quad 0 \quad 3]^T$, $\hat{a}_u = 3500$, $\hat{a}_v = 6000$, $\hat{u}_0 = 800$, and $\hat{v}_0 = 400$. To enhance the effect by gravity compensator, the mass of link 3 has been increased to 0.325 kg. The result of the controller including and excluding gravity compensator is shown in Figure 4.16.

From Figure 4.11, it is shown that the controller including gravity compensator will lead the feature image position error converge to zero. By contrast, there is a significant steady state error in feature image position if the gravity compensator is excluded.

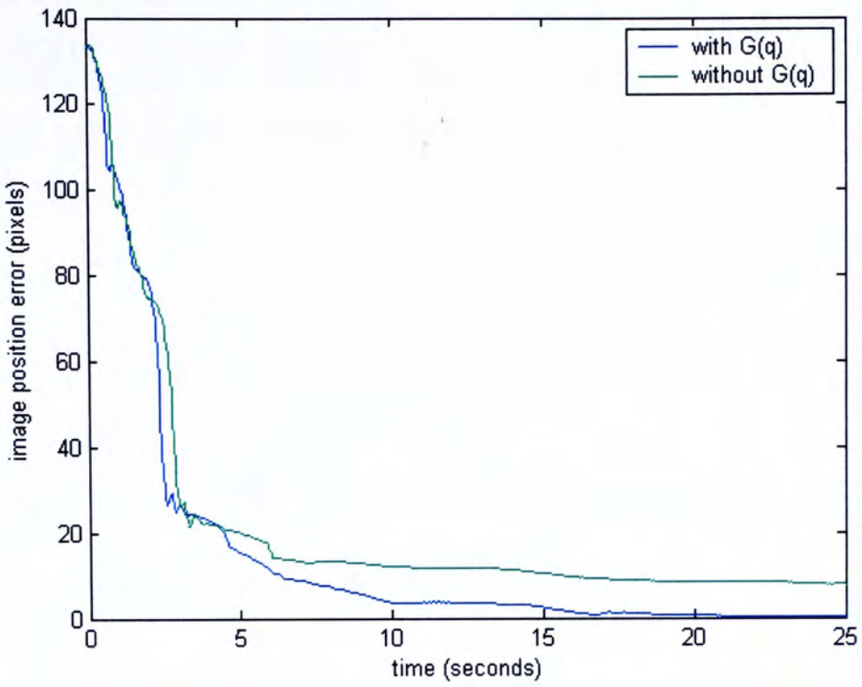


Figure 4.11: Experimental result for the presence and absence of gravity compensator.

4.5 Control Performance with Previous Image Positions

Multiple feature points can also be obtained by a single feature point in motion. During motion, several previous positions of feature points are selected for the feedback control. According to Proposition 2, the adaptive parameters can be estimated to the true value up to a scale. For the following experiment, the control gains are set as follows: $K_p = 0.0003N \cdot m / pixels$, $K_v = 120N \cdot m \cdot s / rad$; the adaptive gains are set as:

$K_1 = diag[2.0\times10^{-5},2.0\times10^{-5},2.0\times10^{-5},2.0\times10^{-5},2.0\times10^{-5},2.0\times10^{-5},$
$2.0\times10^{-5},2.0\times10^{-5},2.0,2.0,2.0,2.0]$

$K_2 = diag[12.5,12.5,12.5,12.5,12.5,12.5,12.5,12.5,$
$1.25\times10^{-4},1.25\times10^{-4},1.25\times10^{-4},1.25\times10^{-4}]$;

and the initial estimated parameters are set as follows: ${}^{C}\hat{R}_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$,

${}^{C}\hat{p}_B = [0 \quad 0 \quad 3]^T$, $\hat{a}_u = 3500$, $\hat{a}_v = 6000$, $\hat{u}_0 = 800$, and $\hat{v}_0 = 400$. The control performance is shown in the following figure (Fig. 4.12):

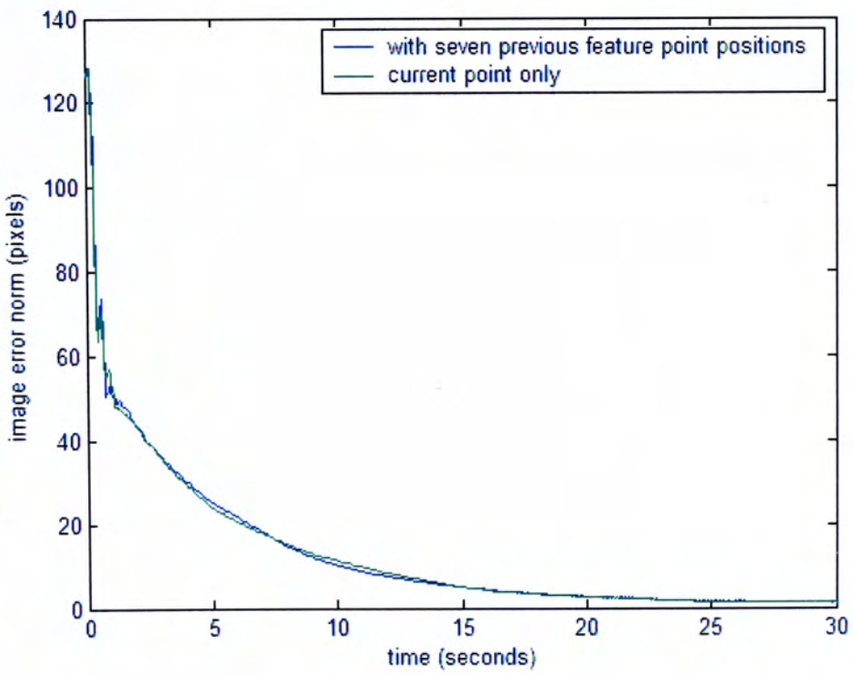


Figure 4.12: Control performance of controller using previous image positions and controller using current image position only.

In terms of control performance, it seems that there is not much difference in single feature point and multiple feature points. The adaptive parameters of system using previous image positions are converged to the following values:

$$\hat{\Theta} = [3486.21, 776.29, 65.19, 2015.28, 14.35, 435.11,$$
$$-6166.90, 1378.02, -0.000720, 1.04, 0.27, 3.99]^T$$

and the adaptive parameters of system using current image point only are converged to the following values:

$$\hat{\Theta} = [3484.71, 780.05, 41.06, 1880.00, 33.22, 429.97,$$
$$-6162.60, 1329.45, -0.014, 1.05, 0.23, 3.71]^T$$

The convergences of adaptive parameters are shown in Fig. 4.13 (a) and (b). Theoretically, for the adaptive algorithm with seven or more image feature points, the adaptive parameters can be estimated to the true values up to a scale. Due to the large sampling period, the parameters may not be adapted to the true values up to a scale. However, the controller still retains asymptotic stability in image position error.

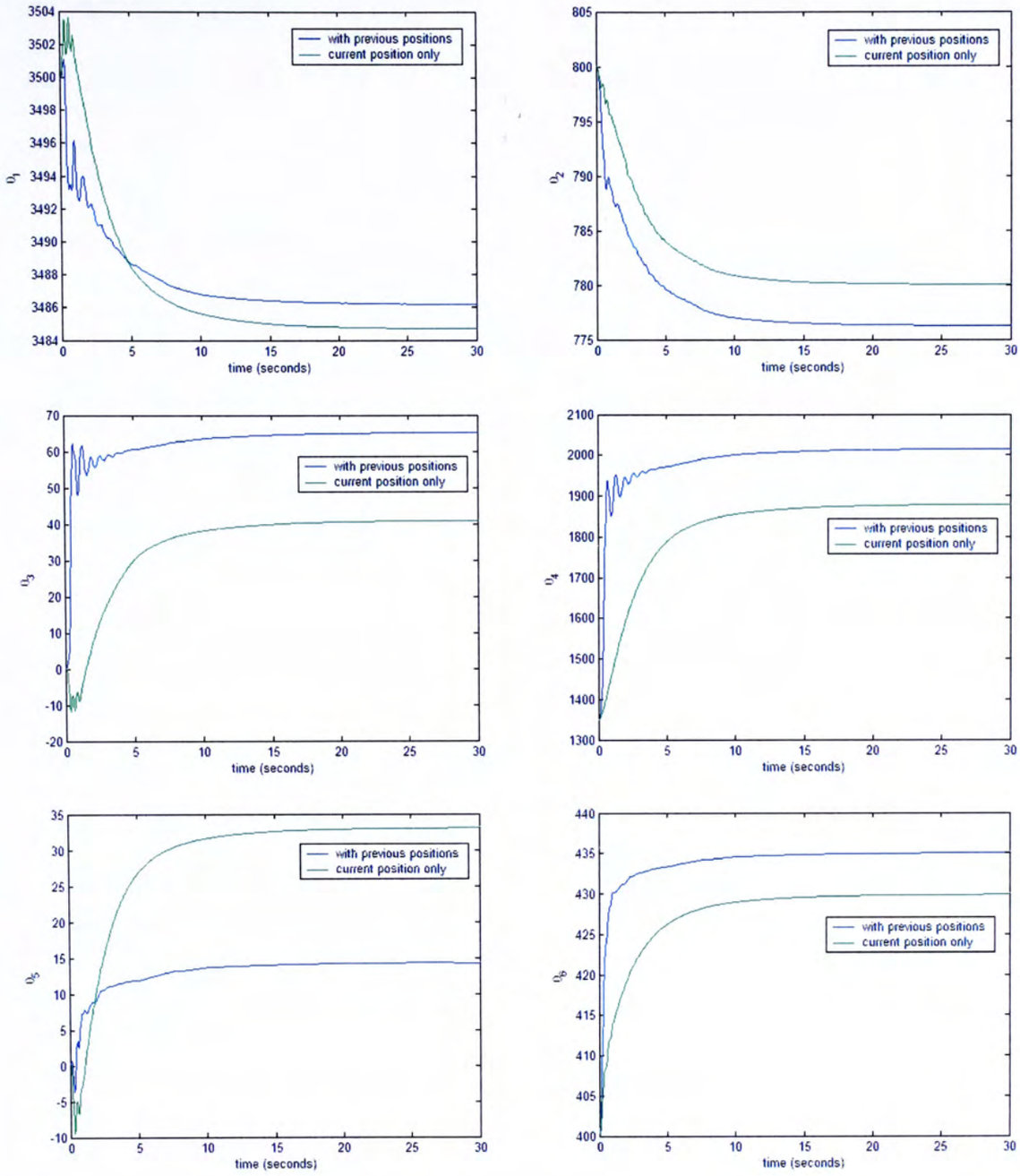


Figure 4.13a: Convergence of first six adaptive parameters with controller using previous image positions and controller using current image position only.

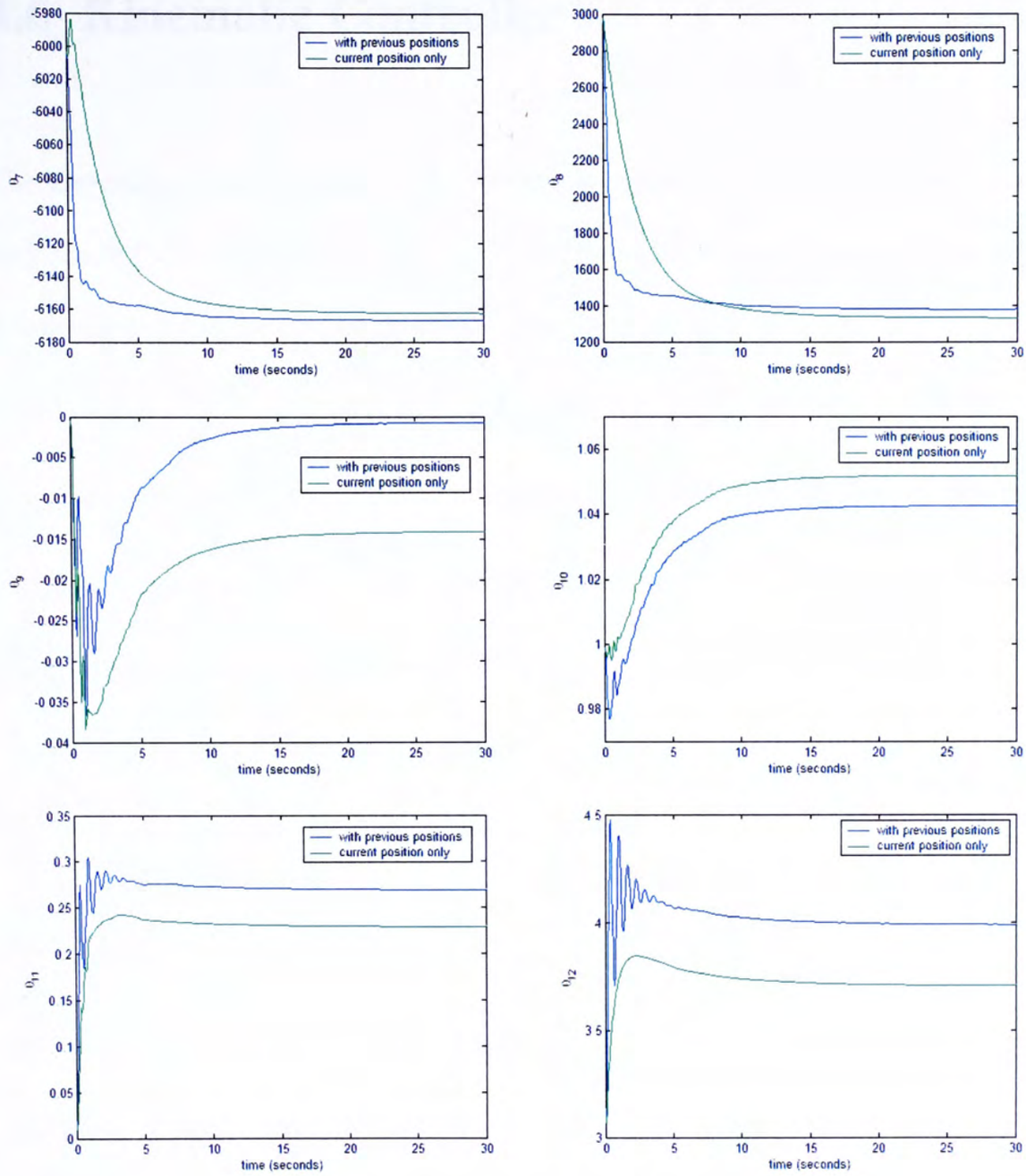


Figure 4.13b: Convergence of last six adaptive parameters with controller using previous image positions and controller using current image position only.

4.6 Kinematic Controller

The kinematic controller used in the following experiments is proposed by Hosoda and Asada [10]. For a system with m image feature points and n degree-of-freedom robot manipulator, the control equation is shown as follows:

$$\dot{q} = \hat{J}_c^{+}\dot{\xi}_d + (I_n - \hat{J}_c^{+}\hat{J}_c)k - K\hat{J}_c^{+}\Delta\xi \qquad (4.1)$$

where $\dot{q}$ is the desired manipulator joint velocity vector; $\hat{J}_c$ is the $2m\times n$ estimated composite Jacobian matrix, and $\hat{J}_c^{+}$ is the pseudo-inverse of $\hat{J}_c$; $\dot{\xi}_d$ is the desired image velocity (for set-point control, $\dot{\xi}_d = 0_{2m\times 1}$); I_n is an $n\times n$ identity matrix; k is an arbitrary vector; K is a positive gain matrix; and $\Delta\xi$ is the image position error. The updating algorithm for $\hat{J}_c$ is shown as follows:

$$\hat{J}_c(t) - \hat{J}_c(t-\Delta t) = \frac{\{[\xi(t) - \xi(t-\Delta t)] - \hat{J}_c(t-\Delta t)\Delta q(t)\}\Delta q^T(t)Q(t)}{\rho + \Delta q^T(t)Q(t)\Delta q(t)}, \text{ if } \|\Delta q\| \neq 0$$

$$\text{or} \qquad \hat{J}_c(t) - \hat{J}_c(t-\Delta t) = 0_{2m\times n}, \text{ if } \|\Delta q\| = 0 \qquad (4.2)$$

where $\Delta q(t) = q(t) - q(t-\Delta t)$; $Q(t)$ is a full rank weighting matrix; and ρ is the forgetting factor, which is $0 \le \rho \le 1$.

The control scheme mentioned above only concerns the desired trajectory of the robot manipulator joints. It does not control the joint motor torques directly.

The following experiment is to compare the performances of both the kinematic controller and the proposed dynamic adaptive controller. For the kinematic controller, the controller parameters are set as follows: $K = diag\lfloor 1.0\times10^{-5} \quad 1.0\times10^{-5} \quad 1.0\times10^{-5} \rfloor$, $k = 0_{3\times 1}$; and for the estimator, the

parameters are set as follows: $\hat{J}_c(0) = \begin{bmatrix} 194 & -160 & 97 \\ 0 & -507 & -254 \end{bmatrix}$, $Q(t) = I_{3\times3}$, and $\rho = 0$.

For the proposed dynamic adaptive controller, the control gains are set as follows: $K_p = 0.0003N \cdot m / pixels$, $K_v = diag[20 \quad 150 \quad 150]N \cdot m \cdot s / rad$; and the adaptive gains are set as follows: $K_1 = 20$ and $K_2 = 0.025$. The initial estimated parameters are set as follows: ${}^C\hat{R}_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$, ${}^C\hat{p}_B = [0 \quad 0 \quad 3]^T$, $\hat{a}_u = 3500$, $\hat{a}_v = 6000$, $\hat{u}_0 = 800$, and $\hat{v}_0 = 400$. The experiment result for comparing both controllers is shown as the following (Fig. 4.14 and 4.15):

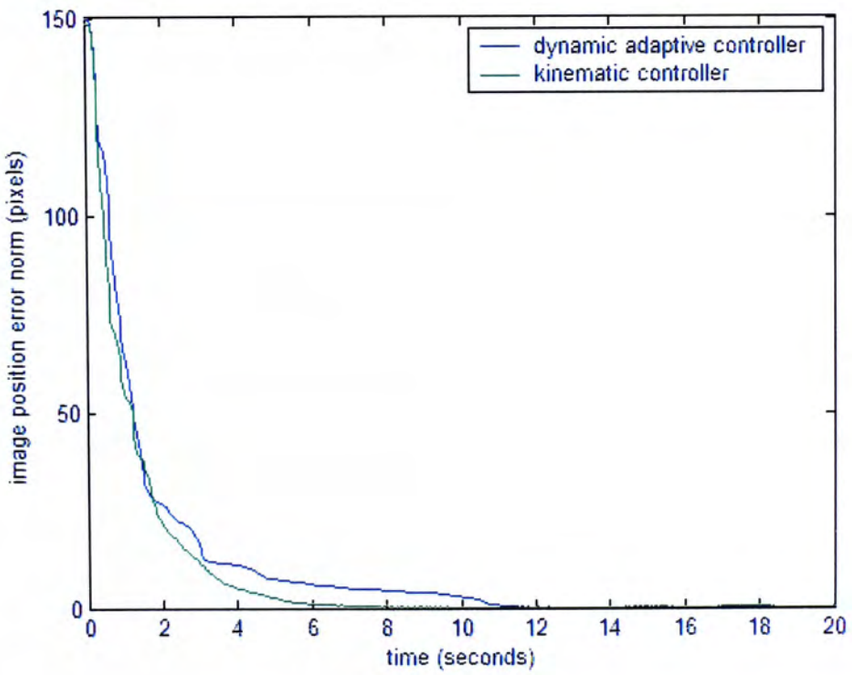


Figure 4.14: Control performances of the kinematic controller and the dynamic adaptive controller

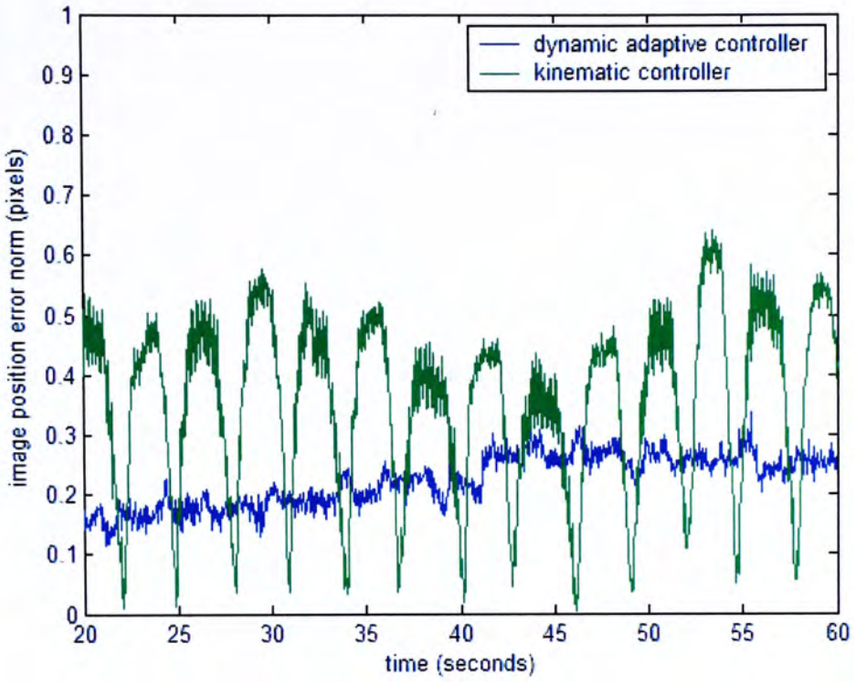


Figure 4.15: Steady state performances of the kinematic controller and the dynamic adaptive controller

To enhance the difference in control performance of the proposed dynamic adaptive controller and the kinematic controller, the mass of link 3 of manipulator is been increased up to 0.325 kg. The experimental result is shown in Figure 4.16.

The control performance of the proposed dynamic adaptive controller may not be so smooth, due to the frictional forces in robot joints. Also, as the kinematic controller is implemented by using embedded function of the motion controller card which has a much shorter sampling period (0.162 milliseconds), the performance of kinematic controller appears to be smooth. However, for steady state, there are ripples in the kinematic controller, while the ripples in the dynamic controller are significantly smaller.

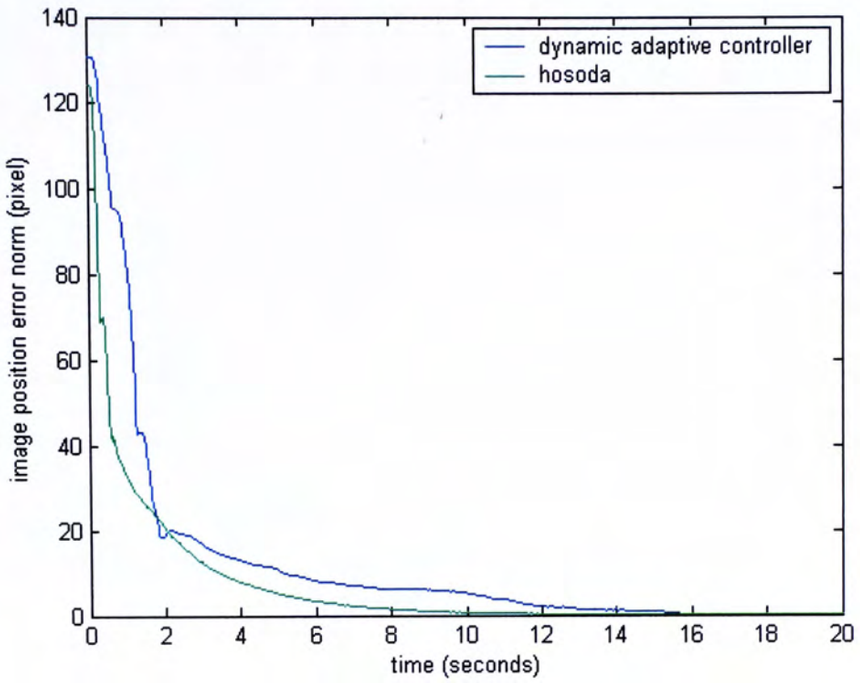


Figure 4.16: Control performances of the kinematic controller and the dynamic adaptive controller with increased manipulator weight

Another experiment is preformed on the motion control board which enables velocity regulation. Without changing the gains and parameters in the controller algorithm, only some parameters are changed in the motion control board to simulate two different motor velocity regulators. Parameters for the kinematic controller and estimator are the same as the above experiment. The result is shown in Figure 4.17.

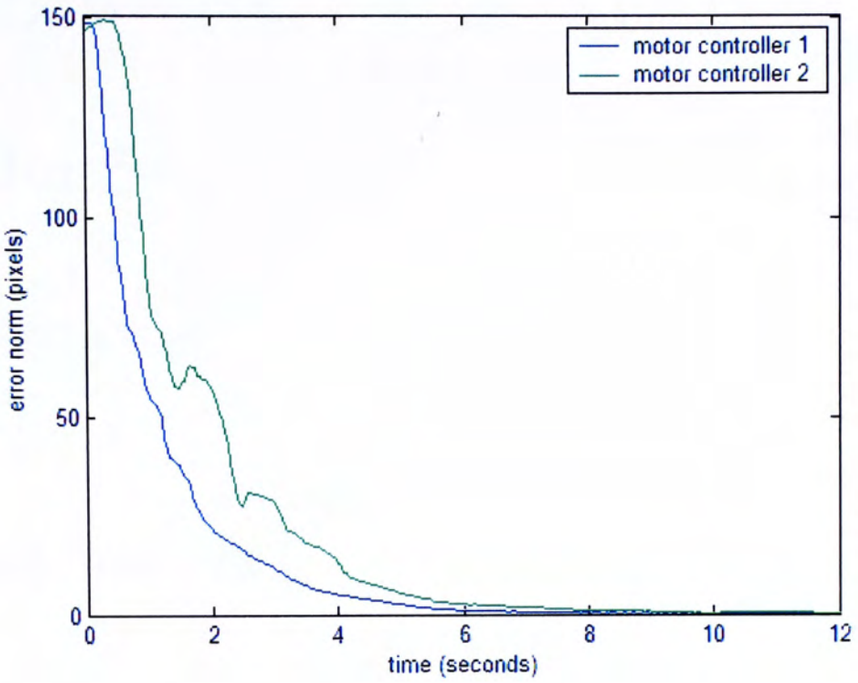


Figure 4.17: Performance of a kinematics visual servo controller with two different motor controllers.

From Figure 4.17, it is shown that the performance of different motor controller could affect visual servoing performance for the same desired trajectory. It is because different motor controllers have different response in reaching the same desired trajectory.

Chapter 5

Conclusions

5.1 Conclusions

In this thesis, the performance of the proposed visual servoing controller has been studied with extensive experimental investigation. The work of this thesis is summarized as follows:

1. Investigate the properties of our new dynamic adaptive visual servo controller through a series of experiments. The properties are studied by experiments comparing performances
   a) among different control gains (proportional gains and derivative gains);
   b) among different adaptive gains;
   c) presence and absence of gravity compensator;
   d) between system using seven previous image positions and current image position only.
2. We compared the performance of our new dynamic adaptive visual servo controller with a kinematic visual servo controller through experiments.

5.2 Future Work

In this thesis, the performance of the proposed image-based visual servoing controller has been experimentally investigated. The future work will be focused on the following issues:

1. The controller is only functioned in point-to-point position control. Trajectory tracking visual servo controller should be developed further.
2. The true robot dynamics in this work is assumed to be known. However, in practice, there are uncertainties in dynamics such as frictions. The controller considering uncertainties in dynamics should be developed.
3. The work in this thesis is only focused on eye-and-hand system. The control algorithm can also be applied on eye-in-hand systems. This should be developed further.
4. The measurement of visual velocity is subjected to big noises due to the low sampling rate of system. This problem should be taken account in future.

Appendix

A. Control Theory

Here, the basics of control theory are briefly reviewed. For details, please refer to the book by Slotine and Li [37].

Definition 1 (Stability)

The equilibrium state $x = 0$ is said to be stable if, for any $R > 0$, there exists $r > 0$, such that if $\|x(0)\| < r$, then $\|x(t)\| < R$ for all $t \geq 0$. Otherwise, the equilibrium point is unstable.

Definition 2 (Asymptotic Stability)

An equilibrium point 0 is asymptotic stable if it is stable, and if in addition there exists some $r > 0$ such that $\|x(0)\| < r$ implies that $x(t) \to 0$ as $t \to \infty$.

Definition 3 (Locally Positive Definite)

A scalar continuous function $V(x)$ is said to be locally positive definite if $V(0)=0$ and, in a ball B_{R_0}

$$x \neq 0 \Rightarrow V(x) > 0$$

if $V(0)=0$ and the above property holds over the whole state space, then $V(x)$ is said to be globally positive definite.

Definition 4 (Lyapunov Function)

If, in a ball B_{R_0}, the function $V(x)$ is positive definite and has continuous partial derivatives, and if its time derivative along any state trajectory of system $\dot{x} = f(x)$ is negative semi-definite, i.e.,

$$\dot{V}(x) \leq 0$$

then $V(x)$ is said to be a Lyapunov function of the system $\dot{x} = f(x)$.

Theorem 1 (Local Stability)

If, in a ball B_{R_0}, there exists a scalar function $V(x)$ with continuous first partial derivatives such that

- $V(x)$ is positive definite (locally in B_{R_0})
- $\dot{V}(x)$ is negative semi-definite (locally in B_{R_0})

then the equilibrium point 0 is stable. If, actually, the derivative $\dot{V}(x)$ is locally negative definite in B_{R_0}, then the stability is asymptotic.

Theorem 2 (Global Stability)

Assume that there exists a scalar function V of the state x, with continuous first order derivatives such that

- $V(x)$ is positive definite
- $\dot{V}(x)$ is negative definite
- $V(x) \to \infty$ as $\|x\| \to \infty$

then the equilibrium at the origin is globally asymptotically stable.

Definition 5 (Invariant Set)

A set G is an invariant set for a dynamic system if every system trajectory which starts from a point in G remains in G for all future time.

Theorem 3 (Global Invariant Set Theorem)

Consider the system $\dot{x} = f(x)$, with f continuous, and let $V(x)$ be a scalar function with continuous first partial derivatives. Assume that

- $V(x) \to \infty$ as $\|x\| \to \infty$
- $\dot{V}(x) \leq 0$ over the whole state space

let R be the set of all points where $\dot{V}(x) = 0$, and M be the largest invariant set in R, then all solutions globally asymptotically converge to M as $t \to \infty$.

Theorem 4 (Barbalat's Lemma)

If a scalar function $V(x,t)$ satisfies the following conditions:

- $V(x,t)$ is lower bounded
- $\dot{V}(x,t)$ is negative semi-definite
- $\dot{V}(x,t)$ is uniformly continuous in time

then $\dot{V}(x,t) \to 0$ as $t \to \infty$.

Bibliography

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control", *IEEE Transaction on Robotics and Automation*, vol. 12, pp. 651-670, Oct. 1996.

[2] Y. Shirai and H. Inoue, "Guiding a robot by visual feedback in assembling tasks", *Pattern Recognition*, vol. 5, pp. 99-108, 1973.

[3] J. Hill and W. T. Park, "Real time control of a robot with a mobile camera", *Proceedings of 9th International Symposium on Industrial Robots*, Washington, D. C., Mar. 1979, pp. 233-246.

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

[5] E. Malis, F. Chaumette, and S. Boudet, "Positioning a coarse-calibrated camera with respect to an unknown object by 2D 1/2 visual servoing", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1352-1359, 1998.

[6] R. C. Luo, R. E. Mullen Jr., and D. E. Wessel, "An adaptive robotic tracking system using optical flow", *Proceedings of the IEEE International Conference on Robotic and Automation*, pp. 568-573, 1988.

[7] R. S. Ahluwalia and L. M. Fogwell, "A modular approach on visual servoing", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 943-950, 1986.

[8] N. P. Papanikolopoulos and P. K. Khosla, "Adaptive robotic visual tracking: theory and experiments", *IEEE Transaction on Automatic Control*, Vol. 38, No. 3, pp. 429-425, 1993.

[9] N. P. Papanikolopoulos, B. Nelson, and P. K. Khosla, "Six degree-of-freedom hand/eye visual tracking with uncertain parameters", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 174-179, 1994.

[10] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true Jacobian", *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 186-191, 1994.

[11] B. H. Yoshimi and P. K. Allen, "Active, uncalibrated visual servoing", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 156-161, 1994.

[12] J. T. Feddema and C. S. G. Lee, "Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera", *IEEE Transaction on System, Man, and Cybernetics*, Vol. 20, No. 5, pp. 1172-1183, 1990.

[13] P. I. Corke and M. C. Good, "Dynamics effects in high-performance visual servoing", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1838-1843, 1992.

[14] P. I. Corke and M. C. Good, "Dynamics effects in visual closed-loop systems", *IEEE Transaction on Robotics and Automation*, Vol. 12, No. 5, pp. 671-683, Oct. 1996.

[15] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems", *IEEE/ASME Transaction on Mechatronics*, Vol. 5, No. 1, pp. 39-48, Mar. 2000.

[16] R. Kelly, "Robust asymptotically stable visual servoing of planar robots", *IEEE Transaction on Robotic and Automation*, Vol. 12, No. 5, pp. 759-766, Oct. 1996.

[17] A. Maruyama and M. Fujita, "Robust visual servo control for planar manipulators with the eye-in-hand configurations", *Proceedings of the 36th IEEE Conference on Decision and Control*, pp. 2551-2552, 1997.

[18] E. Zergeroglu, D. M. Dawson, M. de Queiroz and S. Nagarkatti, "Robust visual-servo control of planar robot manipulators in the presence of uncertainty", *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 4137-4142, 1999.

[19] M. Asada, T. Tanaka, and K. Hosoda, "Adaptive binocular visual servoing for independently moving target tracking", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2076-2081, 2000.

[20] H. Kase, N. Maru, A. Nishikawa, S. Yamada, and F. Miyazaki, "Visual servoing of the manipulator using the stereo vision", *Proceedings of IEEE International Conference on Industrial Electronics, Control, and Instrumentation*, pp. 1791-1796, 1993.

[21] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2267-2272, 1991.

[22] Y. T. Shen, G. L. Xiang, Y. H. Liu and K. J. Li, "Uncalibrated visual servoing of planar robots", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 580-585, 2002.

[23] L. Hsu and P. L. S. Aquino, "Adaptive visual tracking with uncertain manipulator dynamics and uncalibrated camera", *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 1248-1253, 1999.

[24] E. Malis, F. Chaumette, and S. Boudet, "2-1/2-D visual servoing", *IEEE Transaction on Robotics and Automation*, Vol. 15, No. 2, 1999.

[25] F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings", *Procedings of IEEE International Conference on Robotics and Automation*, pp. 630-635, 2000.

[26] C. C. Cheah, M. Hirano, S. Kawamura and S. Arimoto, "Approximate Jacobian control for robots with uncertain kinematics and dynamics", *IEEE Transaction on Robotics and Automation*, Vol. 19, No. 4, pp. 692-702, 2003.

[27] C. C. Cheah, M. Hirano, S. Kawamura and S. Arimoto, "Approximate Jacobian control with task-space damping for robot manipulators", *IEEE Transaction on Automatic Control*, Vol. 49, No. 5, pp. 752-757, 2004.

[28] E. Malis and F. Chaumette, "2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement", *International Journal of Computer Vision*, Vol. 37, No. 1, pp. 79-97, 2000.

[29] C. J. Fang and S. K. Lin, "A performance criterion for the depth estimation with application to robot visual servo control", *Journal of Robotic System*, Vol. 18, pp. 609-622, 2001.

[30] F. Conticelli and B. Allotta, "Nonlinear controllability and stability analysis of adaptive image-based system", *IEEE Transaction on Robotics and Automation*, Vol. 17, pp. 208-241, 2001.

[31] F. Conticelli, B. Allotta, and C. Colombo, "Hybrid visual servoing: a combination of nonlinear control and linear vision", *Robotics and Automation Systems*, Vol. 29, pp. 243-256, 1999.

[32] L. Matthies and T. Kanade, "Kalman filter-based algorithms for estimating depth from image sequences", *International Journal of Computer Vision*, Vol. 3, pp. 209-236, 1989.

[33] C. E. Smith, S. A. Brandt, and N. P. Papanikolopoulos, "Eye-in-hand robotic tasks in uncalibrated environments", *IEEE Transaction on Robotics and Automation*, Vol. 13, No. 6, pp. 903-914, 1997

[34] R. Kelly and A. Coello, "Analysis and experimentation of transpose Jacobian based Cartesian regulators", *Robotica*, Vol. 17, No. 3, pp. 303-312, 1999.

[35] C. C. Cheah, S. Kawamura, and S. Arimoto, "Feedback control for robotic manipulators with an uncertain Jacobian matrix", *Journal of Robotic Systems*, Vol. 16, No. 2, pp. 119-134, 1999.

[36] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2003.

[37] J. J. E. Slotine and W. P. Li, *Applied Nonlinear Control*, Prentice Hall, 1991.