Enhanced Image-Based Visual Servoing Dealing with Uncertainties

Ahmad Ghasemi

A Thesis
In the Department
of
Mechanical, Industrial and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Mechanical Engineering) at
Concordia University
Montréal, Québec, Canada

January 2020

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Ahmad Ghasemi

Entitled: Enhanced Image-Based Visual Servoing Dealing with Uncertainties

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

Chair
Dr. Olga Ormandjieva

External Examiner
Dr. Farrokh Janabi-Sharifi

External to Program
Dr. Wei-Ping Zhu

Examiner
Dr. Youmin Zhang

Examiner
Dr. Brandon Gordon

Thesis Supervisor
Dr. Wen-Fang Xie

Approved by
Dr. Ivan Contreras, Graduate Program Director

February 10, 2020

Dr. Amir Asif, Dean
Gina Cody School of Engineering & Computer Science

Abstract

Enhanced Image-Based Visual Servoing Dealing with Uncertainties

Ahmad Ghasemi, Ph.D.
Concordia University, 2020

Nowadays, the applications of robots in industrial automation have been considerably increased. There is increasing demand for the dexterous and intelligent robots that can work in unstructured environment. Visual servoing has been developed to meet this need by integration of vision sensors into robotic systems. Although there has been significant development in visual servoing, there still exist some challenges in making it fully functional in the industry environment. The nonlinear nature of visual servoing and also system uncertainties are part of the problems affecting the control performance of visual servoing. The projection of 3D image to 2D image which occurs in the camera creates a source of uncertainty in the system. Another source of uncertainty lies in the camera and robot manipulator's parameters. Moreover, limited field of view (FOV) of the camera is another issues influencing the control performance. There are two main types of visual servoing: position-based and image-based. This project aims to develop a series of new methods of image-based visual servoing (IBVS) which can address the nonlinearity and uncertainty issues and improve the visual servoing performance of industrial robots.

The first method is an adaptive switch IBVS controller for industrial robots in which the adaptive law deals with the uncertainties of the monocular camera in eye-in-hand configuration. The proposed switch control algorithm decouples the rotational and translational camera motions and decomposes the IBVS control into three separate stages with different gains. This method can increase the system response speed and improve the tracking performance of IBVS while dealing with camera uncertainties. The second method is an image feature reconstruction algorithm based on the Kalman filter which is proposed to handle the situation where the image features go outside the camera's FOV. The combination of the switch controller and the feature reconstruction algorithm can not only improve the system response speed and tracking performance of IBVS, but also can ensure the success of servoing in the case of the feature loss. Next, in order to deal with the external disturbance and uncertainties due to the depth of the features, the third new control method is designed to combine proportional derivative (PD) control with sliding mode control (SMC) on a 6-DOF manipulator. The properly tuned PD controller can ensure the fast tracking performance and SMC can deal with the external disturbance and depth uncertainties. In the last stage of the thesis, the fourth new semi off-line trajectory planning method is developed to perform IBVS tasks for a 6-DOF robotic manipulator system. In this method, the camera's velocity screw is parametrized using time-based profiles. The parameters of the velocity profile are then determined such that the velocity profile takes the robot to its desired position. This is done by minimizing the error between the initial and desired features. The algorithm for planning the orientation of the robot is decoupled from the position planning of the robot. This allows a convex optimization problem which lead to a faster and more efficient algorithm. The merit of the proposed method is that it respects all of the system constraints. This method also considers the limitation caused by camera's FOV.

All the developed algorithms in the thesis are validated via tests on a 6-DOF Denso robot in an eye-in-hand configuration.

Acknowledgments

I would like to express my deepest appreciation and gratitude to my supervisor Dr. Wen-Fang Xie. Without her guidance, continuous support, and encouragement throughout my studies and research, this dissertation would have been impossible. For this, I am utmost grateful.

Also, I cannot be thankful enough to my family, friends and those who contributed to the successful completion of my studies at Concordia University for self-devotion, encouragement, and support in the past years. I appreciate the support of Concordia University that covered my financial needs during my studies.

I dedicate this thesis to my lovely mother, father and sisters for their unconditional support in my life.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

$AIBVS$	Augmented Image-Based Visual Servoing
$ESIBVS$	Enhanced Switch Image-Based Visual Servoing
FOV	Field of View
$IBVS$	Image-Based Visual Servoing
ISE	Integrate Square Error
MPC	Model Predictive Control
$PBVS$	Position-Based Visual Servoing
SMC	Sliding Mode Control
UAV	Unmanned Aerial Vehicle
DOF	Degree of Freedom
ITAE	Integral Time-weighted Absolute Error index
NFE	Norm of Feature Errors

Symbols

(x, y)	Projected object coordinates in camera image plane
(X, Y, Z)	Object coordinates with respect to camera
α	Angle between desired and actual features

β	Camera scaling factor
$\mathbf{L_s}$	Image Jacobian matrix
$\mathbf{Q}$	Weight matrix
$\omega_x, \omega_y \& \omega_z$	Rotational elements of camera velocity
τ	Joint torque
θ	Uncertaion parameters vector
a_{ij}	Element of robot Jacobian matrix
c_u, c_v	Camera principal point coordinates
$D(k)$	Kalman filter gain
f	Focal length of the camera
$I_x, I_y \& I_z$	Moments of inertia of the robot links
J_R	Robot Jacobian
J_r	Rotational part of image Jacobian matrix
J_t	Translational part of image Jacobian matrix
J_{img}	Image Jacobian matrix
J_{Rr}	Rotational part of robot Jacobian
J_{Rt}	Translational part of robot Jacobian
K	Proportional gain
L_{v4}	Jacobian matrix for four features
$M(k)$	Measurement vector
OF	Objective function
$P(k)$	Prediction of error covariance matrix

$R(k)$	measurement covariance matrix
S	Sliding surface
s_i	Feature coordinates for $i^{\text{ th}}$ feature
s_{id}	Desired feature coordinates for $i^{\text{ th}}$ feature
T_s	Sampling time
u_i, v_i	Pixel coordinates in image plane for $i^{\text{ th}}$ feature
V	Lyapunov function candidate
V_c	Velocity of the camera
$V_x, V_y \& V_z$	Translational elements of camera velocity
V_{cr}	Camera rotational velocity
V_{ct}	Camera translational velocity
$X(k)$	Feature state vector
Y_i	Regression matrix at i^{th} stage
Z	Depth of the feature
C	Coriolis force
G	Gravitational force
M	Inertia matrix
q	Robot joint position

Chapter 1

Introduction

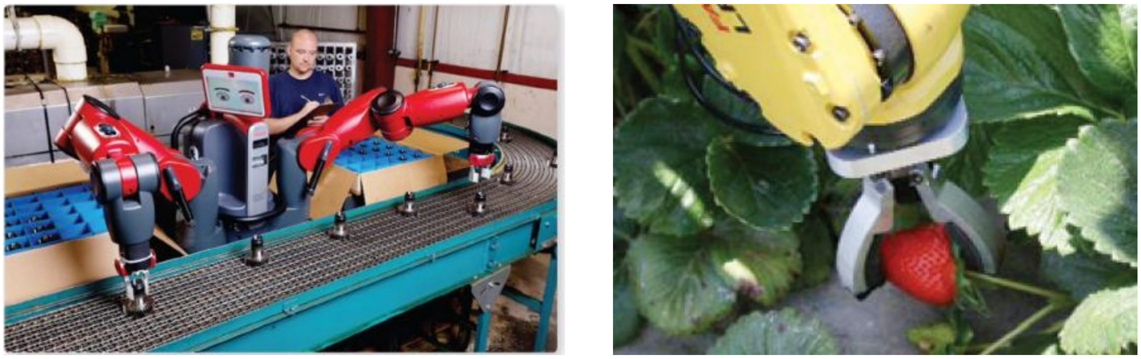
Nowadays, robots are an essential part of the automation industry. However, they are generally limited to operate in structured environments. Conventional robots use open-loop kinematic calculations to determine the end-effector position with respect to a known reference frame. The target object position must also be known with respect to the same coordinate frame. The issue is that any uncertainty of the relevant parameters would cause the task to fail. Vision sensors can deal with these uncertainties by providing non contact and real time measurements of the environment to determine position of the end-effector and target object with respect to each other and the reference frame. Closed-loop position control of the end-effector by exploiting the vision signal as a feedback is referred to as "visual servoing"- the term that appeared to be first used in 1979. In visual servoing, the robot uses the image captured by the camera to determine the position of the end-effector and the target object and uses it as a feedback to control the position of the robotic system [8, 9].

Since early 80's, this topic has been the subject of study among academic researchers aiming to increase the intelligence and dexterity of robotic systems. This research topic is the fusion of many areas such as high speed image processing, kinematics, dynamics, control theroy and real-time computations [10]. Visual servoing has been adopted in a wide range of applications such as robotic welding, teleoperation, missile tracking cameras, fruit picking, robotic ping-pong, juggling, car steering and even aircraft landing [9].

Putting the visual servoing of industrial robots into operation in real unstructured environment is challenging. Since the vision system projects the 3D space into 2D space, one dimension data of the environment which is the depth of the object is lost. This loss of data along with the nonlinearities in the projection leads to difficulties in the integration of vision and robotic systems. Thus, there are some inherent deficiencies that prevent it from being fully applicable. Moreover, industrial application of visual servoing in many cases requires high speed of the task and adequate robustness to uncertainties and camera's limitations. The uncertainties include camera parameters and depth of the features and the main camera limitation is its field of view (FOV). This research work aims to develop a group of new methods to improve the performance and overcome the limitations and deficiencies of visual servoing to make it more feasible to be used in industrial applications.

1.1 Visual Servoing Applications

One main application of visual servoing is in industrial robots. Vision signal is employed as feedback such that the pose (position and orientation) of the robot end-effector is controlled with respect to that of the object until the desired end-effector pose is reached. Some examples of this application are painting, positioning, object grasping (Figure 1.1), assembly and disassembly of mechanical parts [11–13]. In another effort, visual servoing is used in micro and nano positioning as well [14–16].



(a)

(b)

Figure 1.1: Visual servoing applications in robotics: (a) Grasping objects on conveyor belts [1], (b) Robotic fruit picking [2]

Visual servoing techniques is also used in mobile robot navigation (Figure 1.2) where the vision signal from a camera mounted on the robot provides feedback from the environment enabling the controller to guide the robot through the desired path [17, 18].

Figure 1.2: Visual servoing used in unmanned ground vehicle [3]

Medical surgery is another area where visual servoing is used (Figure 1.3). One example is the usage of ultrasonic images as feedback in order to guide a robot carrying the ultrasonic probe to a desired position [19, 20]. Another example is in the laparoscopic surgery where visual servoing can assist the surgeon by automatically guiding the robot carrying the surgical tools to the desired position with respect to the concerned organ [21, 22]

Figure 1.3: Visual servoing used in medical applications [4]

Visual servoing is also employed in the control of unmanned aerial vehicles (UAV) (Figure

1.4). Various applications are in automatic take-off and landing [23, 24], monitoring the structures and bridge inspection [25], cooperative exploration [26], etc.

Figure 1.4: Visual servoing used in UAV applications [5]

The fusion of vision and force measurements has been used in some applications ranging from medical application to human-robot interaction. Examples of the medical applications are ear surgical device [27] and robotic cell injection [28] where the force feedback alongside the vision based motion compensation are used to guide the robot to accomplish the task. Vision and force fusion is used for accurate localization of the grasped object which is used in applications such as part mating or component insertion [29, 30]. Another example is in the area of human-robot interaction in which a guided robot moves securely while cooperating with a human [31, 32].

1.2 Fundamentals of Visual Servoing

Visual servoing has been the subject of research for more than four decades [33] and the phrase ”Visual Servoing” first was used in 1979 [34]. However, it is only in recent years that the significant progress in processing speed has made it functional in real-time industrial applications. Considering the literature, visual servoing may be classified into different categories. In this section, a general classification of visual servoing will be presented, then a more detailed review on the control strategy and control algorithm will be given.

1.2.1 General classification of visual servoing

Visual servoing can be classified based on various features such as the visual servoing strategy, the type of controller, the number of cameras, the camera configuration, the type of image features and the target situation. This classification is summarized in Table 1.1.

Based on the way the vision feedback signal is employed to control the robot, visual servoing can be classified into main two types Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS). Also the combination of these two is known as Hybrid Visual Servoing. Visual servoing strategy will be reviewed in more detail in the next section.

Regarding the control algorithm, various types are studied in the literature. However, proportional control, adaptive control and model predictive control are the three main types.

In another classification, visual servoing is categorized into eye-in-hand and eye-to-hand based on the camera configuration (Figure 1.5). If the camera is mounted on the end-effector it is called “eye-in-hand” and if it is installed in a fixed position looking toward the the robot, it is called “eye-to-hand”.

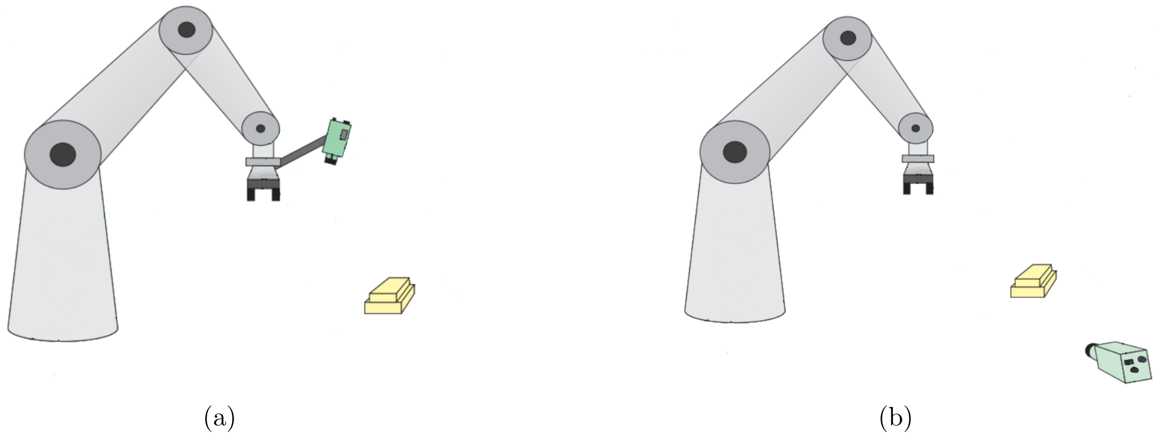


Figure 1.5: Camera configurations in visual servoing: (a) Eye-in-hand (b) Eye-to-hand [6]

Even though eye-to-hand gives wider field of view, some image information may be irrelevant. On the other hand, camera in eye-in-hand focuses on the object with less field of view. The chance that the image taken by the camera is occluded is higher in eye-to-hand than that in eye-in-hand configuration because the robot or other obstacles may block the object image. In both configurations, camera calibration must be performed prior to the task to acquire the intrinsic parameters of the camera such as focal length, resolution and the principle point [9]. The calibration process is normally time consuming. Therefore it has motivated some researchers to investigate some techniques that are robust to camera calibration [35–37]. It has been shown that eye-to-hand configuration is more sensitive to camera calibration. In this thesis, the eye-in-hand configuration is used which allows the camera to focus on the object in the work space.

Visual servoing tasks can be performed using a single camera (monocular), two cameras (stereo vision) or more than two cameras (multiple cameras). A comprehensive survey on these categories can be found in [38]. Single camera needs the least processing time to extract the visual information among these categories. Since each point in the 2D image is related to a line in 3D space, the single camera cannot provide a proper estimation of the distance between the object and the camera (depth of the object). Stereo camera system can resolve this issue by providing more precise depth of the object. The depth computation is done by comparing the differences in different views of the same scene which is considered as a challenge that researchers are actively working on [39–41]. Since the stereo vision system only can use the shared part of the images, its field of view would be less than the monocular vision system. The third and rarest option is using multiple cameras. Although, this system may provide more vision information and better depth estimation compared to the other types, it acquires more time for image processing. The situations that using this type are reasonable are where the object is too large to be fully seen by a mono or stereo vision system, parts of the object are occluded/shadowed or multiple or deforming objects are tracked. Some related work on this system is mentioned in [38]. In this thesis, a monocular vision system is used to obtain faster image processing.

Image feature is another aspect in visual servoing classification. Image feature is defined as a specific piece of information in the image. The control signal is generated by comparing the current image features with the desired ones to reduce the error. The desired features are taken when the robot is in its desired position with respect to the object, then visual servoing task is defined to match the desired and actual image features. Points, lines and segments are the usual image features used in visual servoing. Despite the ease of use and detection, these features are prone to be getting lost in case of occlusion by obstacles which causes the failure of the operation. Recently, several novel features have been used to increase the robustness in visual servoing. While image moments have been widely used in computer vision for pattern recognition, they have recently been adopted in visaul servoing [42]. In this thesis, point features are used as image features.

Visual servoing may be employed to catch moving or stationary objects. In the case of a moving object, trajectory planning is also necessary that poses more challenges. Visual servoing is responsible to keep the object in the field of view and simultaneously reduce the error between the desired and actual position until the object is caught. Various studies have focused on the challenges of catching a moving object [43–47]. This thesis is focused in the stationary object situation.

Table 1.1: Visual servoing classification

Categories	**Main types**		
Visual servoing strategy	Position-based visual servoing (PBVS)	Image-based visual servoing (IBVS)	Hybrid visual servoing
Visual servoing controller	Proportional controller	Adaptive controller	Model predictive controller
Camera configuration	Eye-to-hand	Eye-in-hand	
Number of cameras	Mono vision	Stereo vision	Multiple cameras
Image feature	Point features	Line features	Image moment features
Target situation	Static object	Moving object	

1.2.2 Visual Servoing Strategy

One major classification of visual servoing is done according to visual servoing strategy, namely “ position-based visual servoing (PBVS)” and “ image based visual servoing (IBVS)”. In PBVS, the image features (that could be area of a region, length of a line, position of the center of a region, etc.) are extracted from the camera image. These extarcted information along with information of target geometry and camera model are used to estimate the pose of the camera with respect to the target object. This estimated pose is compared with the desired one to create pose error. In IBVS, error signal is generated directly by comparing the image features and desired ones in the image plane [9]. In the following two sections, the two mentioned methods are explained in detail.

Position-based visual Servoing

Figure 1.6 shows a schematic of position-based servo. Two main parts can be identified in the figure. First part is responsible for taking the image by the camera, extracting of the features from the image and feature interpretation which are in fact reconstruction of 3D information from 2D image information. Second part is PBVS controller which is in charge of generating control signal for the robot using reference position signal and actual position signal.

In this method, the reference point is the relative position and orientation between the object and robot hand. The relative position and orientation are calculated from the images taken by the camera which may be installed on the robot hand (eye-in-hand configuration) or may be installed besides the manipulator (eye to hand configuration).

The robot control problem is well established in robotics community so far. Thus the main topic of research in the field is fast and robust computation of the object pose. In fact, reconstructing 3D information using 2D images information is the main issue for which some algorithms have been developed using various approaches [48].

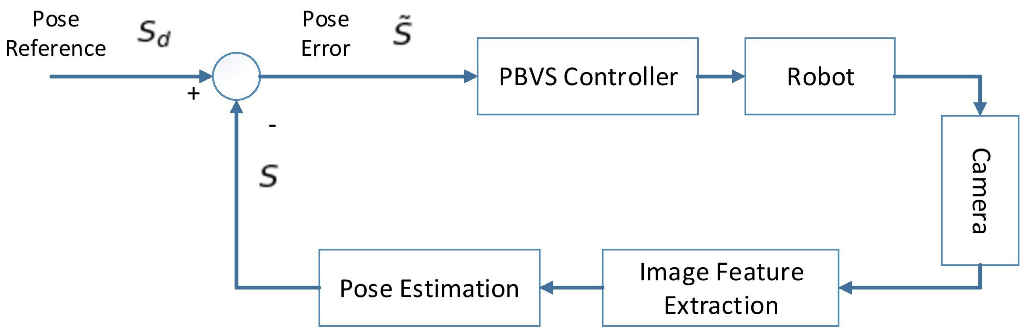


Figure 1.6: PBVS structure

The photographic measurement uses the size and shape of the object to reconstruct the position and orientation of the object and the stereo measurement uses two or more cameras simultaneously. Actually this method is based on comparison of the images taken by cameras. This task is not usually easy in real time applications.

Image-based visual servoing

In the image-based method, the reference signal is defined in the image plane. This scheme is also called feature-based visual servoing. It means that some features are extracted from the image and are compared with the desired reference signal. Figure 1.7 shows the concept of image-based visual servoing. Various quantities could be taken as object feature. The quantities like area of a region, length of a line, position of the center of a region, etc., can be used as features, which could be extracted by applying simple image processing.

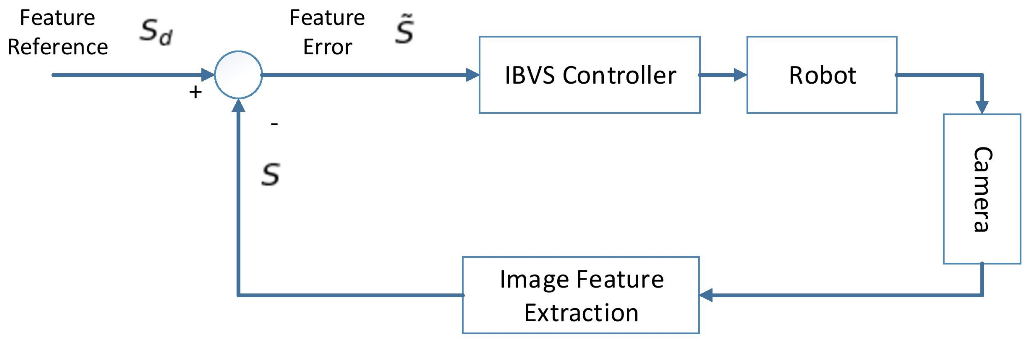


Figure 1.7: IBVS structure

1.2.3 Image-Based Visual Servoing Basics

Since IBVS is referred in all chapters of the thesis, in this section the basics of the IBVS is introduced and related formulations are derived.

The structure of the visual servoing is shown in Figure 1.8. As it is illustrated, it consists of a visual servoing controller block and a robot controller block. A camera mounted on the end-effector captures images and sends the information to an algorithm which extracts image features. These features are compared with desired ones and the difference between them is fed to the visual servoing controller as the feature error. This controller creates a velocity command for the end-effector to compensate this error. Then the velocity command is sent to the robot controller block and compared with the current velocity of the end-effector to create the velocity error. The robot controller could be either a single joint controller or computed torque controller. In the first case, the controller generates separate joint angle commands and in the second one torque command in order to reduce the velocity error.

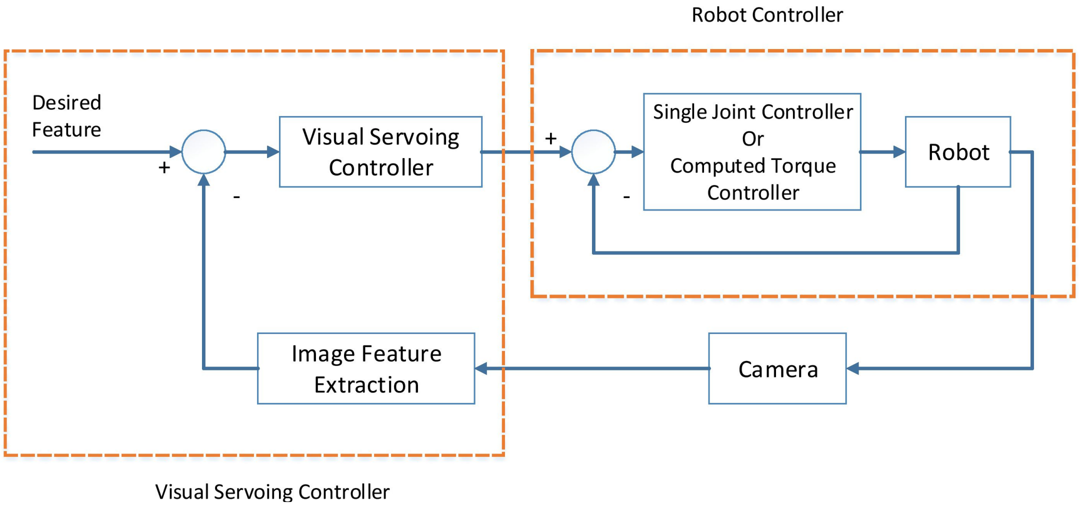
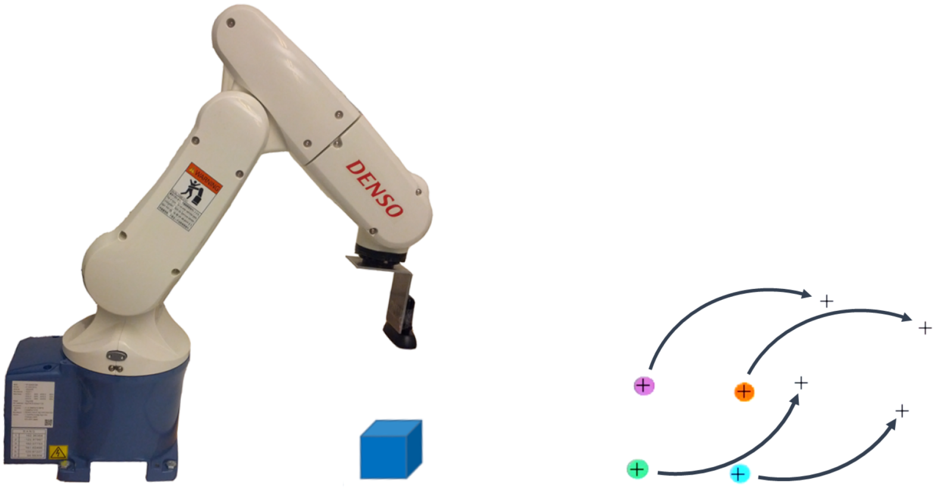

Robot Controller
Desired Feature
+
-
Visual Servoing Controller
+
-
Single Joint Controller Or Computed Torque Controller
Robot
Image Feature Extraction
Camera
Visual Servoing Controller

Figure 1.8: IBVS structure- visual servoing controller vs robot controller

In IBVS, the extracted features are compared with desired ones and feature error is computed. Desired features are extracted from the camera image when it is located in the desired position and orientation with respect to the object. Thus, when the current image is matched with the desired one, it means the end-effector is located in the desired pose with respect to the object and the rest of the task can be catching the object or performing other operation on the object. Therefore the IBVS task is defined as the guidance of the end-effector (camera) in a way that the desired features are matched with the current ones (Figure 1.9).



(a)

(b)

Figure 1.9: Visual servoing task: a) Robot in IBVS operation b) Desired and actual features are matched [7]

In IBVS, the object with (X, Y, Z) coordinates with respect to camera has the projected image coordinates (x, y) in the camera image (Figure 1.10).

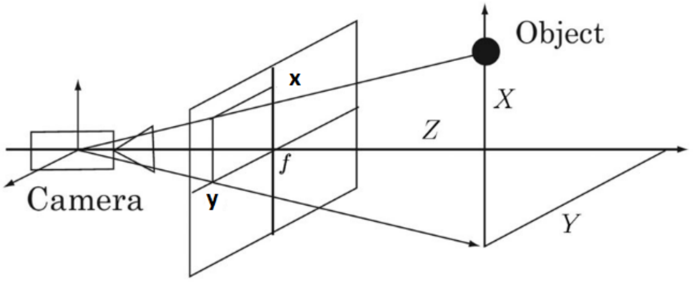


Figure 1.10: Schematic of the camera model [7]

n features are denoted as $s_i(t) = [x_i(t)\ y_i(t)]^T$, and $s_{di} = [x_{di}\ y_{di}]^T$ as the desired image features'i $^{\text{th}}$ coordinates in the image space $(i = 1, .., n)$. The velocity of the camera is defined as $V_c(t)$. The relation between velocity of the camera and image feature velocity can be expressed as:

$$\dot{s}(t) = J_{img}(t)V_c(t), \tag{1.1}$$

where,

$$s(t) = \begin{bmatrix} s_1(t) \\ \vdots \\ s_n(t) \end{bmatrix}, \;\; s_d(t) = \begin{bmatrix} s_{d1}(t) \\ \vdots \\ s_{dn}(t) \end{bmatrix} \tag{1.2}$$

and

$$J_{img}(t) = \begin{bmatrix} J_{img}(s_1(t), Z_1(t)) \\ \vdots \\ J_{img}(s_n(t), Z_n(t)) \end{bmatrix} \tag{1.3}$$

is called the image Jacobian matrix and $Z_1(t), \ldots, Z_n(t)$ are the depth of the features $s_1, ..., s_n$.

Note: In this study, the system is set up in eye-in-hand configuration and the number of features is $n = 4$. It is assumed that the largest distance between features is much less than the depth of the features. Thus, it is inferred that all the features have the same depth Z.

Hence, for the i th feature, the image Jacobian matrix is given as [9]:

$$J_{img}(s_i(t)) = \begin{bmatrix} \frac{f}{Z(t)} & 0 & -\frac{x_i(t)}{Z(t)} & -\frac{x_i(t)y_i(t)}{f} & \frac{f^2+x_i^2}{f} & -y_i(t) \\ 0 & \frac{f}{Z(t)} & -\frac{y_i(t)}{Z(t)} & \frac{-f^2-y_i(t)^2}{f} & \frac{x_i(t)y_i(t)}{f} & x_i(t) \end{bmatrix}, \tag{1.4}$$

where f is the focal length of the camera and $x_i(t)$ and $y_i(t)$ are the projected feature coordinates in the camera frame which are directly related to the camera intrinsic parameters as follows [49].

$$x_i(t) = (u_i(t) - c_u)/f\beta \tag{1.5}$$

and

$$y_i(t) = (v_i(t) - c_v)/f\beta, \tag{1.6}$$

where $u_i(t)$ and $v_i(t)$ are the pixel coordinates in image plane, c_u and c_v are the principal point coordinations and β is the scale factor.

The velocity of the camera can be calculated by manipulating (1.1):

$$V_c(t) = \overset{+}{J}_{img}(t)\dot{s}(t), \tag{1.7}$$

where $\overset{+}{J}_{img}(t)$ is the pseudo-inverse of the image Jacobian matrix. The error signal is defined as $\tilde{s}(t) = s(t) - s_d$ and let $\dot{\tilde{s}}(t) = -K\tilde{s}(t)$. Then the conventional IBVS control law can be designed as:

$$V_c(t) = -K\overset{+}{J}_{img}(t)\tilde{s}(t), \qquad (1.8)$$

where K is the proportional gain.

1.3 Literature Survey

In addition to the type of control strategy corresponding to the feedback type used in visual servoing, various types of controllers have been employed to perform the visual servoing tasks.

Visual feedback in the control loop has aimed to fulfill the need for more dexterous and efficient robots. In many visual servoing applications, a control system with high accuracy and strong robustness is needed, such as cell injection [50], robotic systems [51], automated fault insertion test [52], car steering, aircraft landing and even missile tracking [53].

Between the two major visual servoing control strategies, IBVS has these advantages over PBVS. IBVS directly uses image feature errors to calculate the control signal which reduces computational delay and becomes less sensitive to the camera calibration and target model errors [9, 54]. However, IBVS is still subjected to some parameter uncertainties due to intrinsic and extrinsic parameters [55]. The camera intrinsic parameters include focal length, principal point and camera scaling factors which are estimated through calibration. The extrinsic parameters are due to the estimated depth which introduces the inaccuracy in the image interaction matrix.

Despite the great amount of development of visual servoing technology in the last two decades, it still suffers from a number of problems which prevent it from wide industrial use. High nonlinearity of a visual servoing system is the main cause for its problems. In [56], some potential problems of implementing visual servoing are presented. The most prominent deficiencies of visual servoing, preventing it from practical employment, are listed as follows;

1. Long convergence time
2. Largely dependent on accuracy of camera parameters
3. Interaction matrix singularity
4. Features leaving the FOV or feature loss

5. Lack of robustness to the uncertainties

6. Unknown path of the robot prior to the tasks and lack of 3D space control

7. Poor response to large travels

Various methodologies such as proportional-integral-derivative (PID) control [15, 57, 58], sliding mode control (SMC) [55] and adaptive control [59] have been presented in literature to overcome some of the deficiencies. Image moment features were introduced to deal with the interaction matrix singularity and local minima problem [60–62]. Model predictive visual servoing controller was introduced to deal with the constraints of the system and prevent the features from leaving the FOV [63]. Augmented image-based visual servoing (AIBVS) was presented to make the visual servoing smoother and reduce the risk of features leaving the field of view [64].

The main concept in the above mentioned methods is to generate a velocity or acceleration command as the control input to guide the end-effector smoothly and accurately to the desired position. In the following, a literature survey is given regarding the strategies developed in this research work.

1.3.1 Adaptive Switch Image-Based Visual Servoing

In order to have an effective IBVS which is feasible for practical robotic applications, a fast response system with proven stability is needed. Various studies have been conducted to address and overcome shortcomings of IBVS and enhance its efficiency [56, 65–68]. However, most of the reported IBVS tend to have an overly long converging time that does not meet the demand of industrial applications. Increasing the gain values in the control law is one obvious way to reduce the response time of IBVS. However, there is a limitation on this value because the high gain in IBVS controller tends to make the robotic system shaky and unstable [64]. Moreover, the stability of traditional IBVS system is proved only in a region around the desired position [56]. Also, when the initial feature configuration is far away from the desired one, the converging time is long and possible image singularities may lead to IBVS failure.

On the other hand, one main issue in IBVS is that its performance is dependent on the accuracy of camera calibration. To elaborate, the image Jacobian matrix (J_{img}) relating the image features velocity to the camera velocity, contains the camera intrinsic parameters and depth of the features. The camera parameters can be obtained by calibration process. However, in some cases they may be unreliable especially in industrial applications since they may change during the task. Although many reported IBVS methods have improved the tracking performance by using image moments as features [69] or by optimizing the trajectory planning [68, 70, 71], they have not considered the uncertainties of camera parameters and assume the camera is well calibrated. Some studies have been carried out to deal with the uncertain camera parameters [72–74]. However, in most of these studies, the controller design is kinematic-based, i.e. they consider the robot as an accurate positioning device with negligible dynamics. The kinematic-based controller neglects the dynamics of the robot and render less complex control system. It is easier to implement kinematic-based visual servoing [75] compared with the dynamic-based control method. However, the dynamic-based method considers the dynamic model in the controller design and hence can deliver better control performance in terms of convergence time and robustness with guaranteed stability compared to the kinematic-based one. In recent years, some studies have tried to introduce methods to deal with camera parameter uncertainties by considering dynamics of the robot [76, 77]. However, in these studies, as it is shown in Eq. (1.4), the term associated with depth ($\frac{1}{Z(t)}$) cannot be factorized from the whole image Jacobian matrix. It means Eq. (58) in [76] and Eq. (1) in [77] are not valid formulas and the proposed methods are not applicable to IBVS method. Other similar studies such as [78, 79] are either on stereo eye-to-hand configuration or only verified for 3-DOF robots. Besides the improvement of efficiency and the speed of the proposed controllers over traditional IBVS are not verified. Thus, for eye-in-hand configuration, there is still a lack of research work on a stable fast response method of IBVS to be used in industrial applications with the capability to deal with camera uncertainties.

1.3.2 Visual Servoing dealing with Feature loss

An efficient IBVS feasible for practical robotic operations requires a fast response with strong robustness to feature loss. In fact, feature loss caused by the camera's limited field of view (FOV) prevents the IBVS method from being fully efficient and being applicable to real industrial robots.

The visual features contain much information such as the robots' pose information, the tasks' states, the influence of the environment, the disturbance to the robots, etc. The features are directly related to the motion screw of the end-effector of the robot. The completeness of the feature set during visual servoing is key to fulfilling the IBVS task successfully. Many features have been used in visual servoing such as feature points, image moments, lines, etc. The feature points are known for the ease of image processing and extraction. It is shown that at least three image points are needed for controlling a 6-DOF robot [9]. Hence, four image points are usually used for visual servoing. However, the feature points tend to leave the FOV during the process of visual servoing. A strategy is needed to handle the situation where the features are lost.

There are two main approaches to handle feature loss and/or occlusion caused by the limited FOV of the camera [49]. In the first approach, the controller is designed to avoid occlusion or feature loss, while in the second one, the controller is designed to handle the feature loss.

In the first approach, several techniques have been developed to avoid the feature loss or occlusion. In [80], occlusion avoidance is considered as the second task besides the primary visual servoing task. In [81], a reactive unified convex optimization-based controller was designed to avoid occlusion during tele-operation of a dual-arm robot. Some studies have been carried out in visual trajectory planning considering feature loss avoidance [46, 82, 83]. Model predictive control methods have been adopted in visual servoing to prevent feature loss due to its ability to deal with constraints [84–88]. In [89], predictive control is employed to handle visibility, workspace, and actuator constraints. Despite the success of the studies on preventing feature loss, they suffered from the limited maneuvering workspace of the robot, due to the conservative design required to satisfy many constraints.

In the second approach, the controller tries to handle the feature loss instead of avoiding it. When the loss or occlusion of features occurs, if the remaining visible features are sufficient to generate the non-singular inverse of the image Jacobian matrix, the visual servoing task can still be carried out successfully. In this situation, the rank of the relative Jacobian matrix must be the same as the degrees of freedom [90]. However, this method is no longer effective when the number of remaining visible features become too small to guarantee the full-rankness of the image Jacobian matrix. As studied in [91], another solution is to foresee the position of the lost features and to continue the control process using the predicted features until they become visible again. This method allows partial or complete loss or occlusion of the features. In the second approach [90, 91], the classical IBVS control is employed as the control method, which does not usually provide a fast response. In this research work, an enhanced switch image-based visual servoing (ESIBVS) method is presented to make the switch IBVS control robust in reaction to feature loss.

1.3.3 Enhanced IBVS Controller Using Hybrid PD-SMC Method

PID control has a wide application because of its simplicity and effectiveness. The convergence property of P or PD controller is also satisfactory. However, sometimes sudden variation or small shakiness occur due to image noise or motion vibration. Acceleration input command has been used in some research to reduce these issues [58]. Despite the relative effectiveness of this technique, it is not applicable in all situations because in some robotic systems, only velocity signal can be accepted as control input. Model Predictive Control (MPC) is also used to reduce the error at the end of the prediction horizon [92]. However, the MPC controller is known for its heavy computational burden, which poses challenge for the fast dynamic robot system. In [93], a predictive control method considering visibility constraints and actuator limitations has been proposed and applied in 6-DOF IBVS robotic system. This method provides an efficient solution to the problem of 3D limitations and large displacements. But it does not address the system uncertainty problem due to unknown depth. It is known that SMC has strong robustness for the system model or parameteric uncertainty [94–96]. In recent year, using SMC in IBVS or PBVS to guarantee the robotic system's strong robustness and good tracking performance has been reported in some literature [97–100]. Nonetheless, SMC also has the chattering problem, which could not be neglected completely in some special situations. Meanwhile, some of IBVS control schemes have been only verified on 2-DOF or 3-DOF manipulator in the experiment due to its simple configuration [55, 99, 100]. In this research work, a new hybrid way that combines PD control with SMC in IBVS is proposed to generate the better velocity profile to control the 6-DOF robotic manipulator.

1.3.4 Image-Based Visual Servoing Using Trajectory Planning

Although, lots of researches have been devoted to solve one or two of the above mentioned problems, a reliable and general solution to guarantee the success of visual servoing cannot be found in literature. As the visual servoing problem persist, the researchers aimed to integrate machine vision techniques with trajectory planning techniques to overcome the problems. Trajectory and path planning techniques search and provide the best solution or solutions to accomplish a specific task. These paths are mostly different from the ones that a controllers would produce. Chesi et al. [101] proposed a trajectory planning method for position-based visual servoing. Homogeneous forms were used to parameterize the path and an LMI optimization is carried out to calculate the parameters. Moreover other techniques were also used in PBVS path planning [102]. An adaptive trajectory regeneration method was proposed in [103] for visual servoing in an unstructured environment. Later on navigation guidance technique were integrated with visual servoing to achieve fast visual servoing [104, 105].

Although, the reported trajectory planning techniques demonstrate good performance in executing visual servoing tasks, they were designed for position-based visual servoing. Thus, they suffer from PBVS drawbacks such as sensitivity to model and camera calibration errors. This gap motivated the researchers to develop a trajectory planning technique in an IBVS system [106, 107]. Potential field methods were used to perform IBVS on-line trajectory planning in robotic systems [108, 109]. Potential field techniques are useful in the presence of obstacles and when the system is subjected to constraints. However, both IBVS and PBVS trajectory planning algorithms based on potential field techniques have the following disadvantages. Since, potential field techniques are an on-line planning technique which investigate and plan a path based on the local information, it only considers a constraint when it gets close to it. However when the robot is close to its constraints, it is not possible to go back and choose another path. In addition, potential field techniques suffer from local minima when the attraction force magnitude is equal the repelling force on the robot in the opposite direction.

Generally, in an IBVS trajectory planning, a reference path is produced by the trajectory planner or path planner considering the goals and constraints of the image and the robot. A small controlling error will be defined for each small segment of the trajectory to be followed by the IBVS controller. In such algorithms, the main challenge is to find a path in image space which corresponds to a feasible path in task space. The most basic method developed for solving this problem is using stereo vision and the epipolar geometry constraint between two camera images. Utilizing the privilege of epipolar geometry an image trajectory is generated on both images in a way that corresponds to a feasible or even straight line trajectory in Cartesian space [110, 111]. High load of processing and also decreasing the usable field of view area in cameras are the problems of such solutions. Moreover, using the probabilistic methods for path planning, as reported in some researches [112, 113], is a useful approach, however the computational load of such planning makes it very time consuming.

In this research work, a new image-based trajectory planning algorithm is proposed to overcome the visual servoing deficiencies and develop a reliable algorithm to perform visual servoing tasks.

1.4 Research Objectives and Scopes

Industrial application of visual servoing in many cases requires high speed of the task and adequate robustness to uncertainties and camera limitations. Uncertainties include camera parameters and depth of the features and the main camera limitation is its field of view (FOV). Thus, the main objective of this thesis is to develop a series of new IBVS methods to simultaneously increase the speed of the task and improve its robustness to the mentioned uncertainties and limitations while keeping the stability of the controller. In addition, this thesis aims to find the solution to preventing the feature loss and guaranteeing the success of visual servoing through trajectory planning.

The objectives and scopes of this thesis can be summarized as follows:

1. Develop adaptive switch method of IBVS to deal with camera parameter uncertainties
2. Design switch IBVS that can deal with feature loss
3. Further improve the IBVS robustness by combining PD control with sliding mode control (SMC)
4. Develop semi-off -line trajectory planning method for IBVS method.

1.5 Contributions of the Thesis

This PhD research work is carried out in different stages. The contributions of each stage of the research work are summarized as follows:

First, an IBVS adaptive controller is designed which is capable of dealing with camera parameter uncertainties. In contrast with most of IBVS controllers which are kinematic-based, the proposed controller is designed with considering the dynamics of the robot. The dynamic-based method considers the dynamic model in the controller design and hence can deliver better control performance in terms of convergence time and robustness with guaranteed stability compared to the kinematic-based one. Moreover, a switching scheme is proposed and adopted in the controller design: the controller switches between rotational and translational movements of the end-effector and separate gains are defined for each stage.

This technique provides the chance to overcome the nonlinearity caused by depth parameter and enables us to employ the adaptive methods to estimate the camera parameters in eye-in-hand configuration. Besides, the proposed switching scheme can increase the speed of the visual servoing task by providing different gains for different stages which can be adjusted separately. The stability of the proposed controller is fully investigated by using Lyapunov theory.

Second, a new method is proposed to overcome the shortcomings caused by limited field of view (FOV) of the camera in eye-in-hand IBVS. The designed method in the first stage is effective in increasing the speed of the visual servoing task and dealing with camera uncertainties. However, feature loss caused by the camera's limited FOV still prevents the method from being fully efficient and being applicable to real industrial robots. To resolve this issue, an enhanced switch image-based visual servoing (ESIBVS) method is presented in which a Kalman filter-based feature prediction algorithm is proposed and is combined with the proposed switch method in the first stage to make the switch IBVS control robust in reaction to feature loss. The switch control with the improved tracking performance along with the robustness to feature loss makes it more feasible for industrial robotic applications.

Third, a new hybrid method that combines PD control with sliding mode control (SMC) in IBVS is proposed. The controller deals with the uncertainties in depth of the features and generates better velocity profile to control the 6-DOF robotic manipulator compared to traditional IBVS. The main feature of the proposed hybrid PD-SMC is its less computation burden, compared to the adaptive or predicted control approaches. In addition, it not only can achieve better convergence performance with guaranteed stability, but also owns stronger robustness against uncertainty and disturbance, compared to either IBVS PD or SMC system. Lyapunov direct method is used to prove the stability of the proposed controller.

Last, a new semi off-line trajectory planning method is developed to perform image-based visual servoing (IBVS) tasks for a 6-DOFs robotic manipulator system. The new trajectory planning technique parametrized the camera's velocity screw using time-based profile. The parameters of the velocity profile are then determined such that the velocity profile takes the robot to its desired position. This is done by minimizing the error between the initial and desired features. A depth estimation algorithm is proposed to provide the trajectory planning algorithm with a good estimation of the initial depth. The algorithm for planning the orientation of the robot is decoupled from the position planning of the robot. This allows a convex optimization problem which lead to a faster and more efficient algorithm. The merit of the proposed method is that it respects all of the system constraints. By integrating this technique, the proposed image-based trajectory planning can overcome the above mentioned deficiencies to a great extent.

The performance of all the proposed methods is examined by a 6-DOF robot manipulator with monocular eye-in-hand vision system and their performance is compared with that of traditional IBVS.

1.6 Publications

The results of the presented research work are published in the journals and conferences that are listed below [68, 114–119]:

1. A. Ghasemi, P. Li, and W.-F. Xie, "Adaptive switch image-based visual servoing for industrial robots," International Journal of Control, Automation and Systems, [Online]. Available: https://doi.org/10.1007/s12555-018-0753-y

2. A. Ghasemi, P. Li, W.-F. Xie, and W. Tian, "Enhanced switch image-based visual servoing dealing with featuresloss," Electronics, vol. 8, no. 8, p. 903, 2019.

3. S. Li, A. Ghasemi, W.-F. Xie, and Y. Gao, "An enhanced ibvs controller of a 6dof manipulator using hybrid pd-smc method," International Journal of Control, Automation and Systems, vol. 16, no. 2, pp. 844-855, 2018.

4. M. Keshmiri, W.-F. Xie, and A. Ghasemi, "Visual servoing using an optimized trajectory planning technique for a 4 dofs robotic manipulator," International Journal of Control, Automation and Systems, vol. 15, no. 3, pp. 1362-1373, 2017.

5. S. Li, A. Ghasemi, W.-F. Xie, and Y. Gao, "Sliding mode control (smc) of image-based visual servoing for a 6dof manipulator," in Recent Developments in Sliding Mode Control Theory and Applications. InTech, 2017.

6. A. Ghasemi and W.-F. Xie, "Adaptive image-based visual servoing of 6 dof robots using switch approach," in 2018 IEEE International Conference on Information and Automation (ICIA). IEEE, 2018, pp. 1210-215.

7. A. Ghasemi and W.-F. Xie, "Decoupled image-based visual servoing for robotic manufacturing systems using gain scheduled switch control," in Proceedings of 2017 International Conference on Advanced Mechatronic Systems (ICAMechS). Xiamen, China, 2017, pp. 94-99.

Besides the above mentioned publications, the results of research work in another subject are published in the journal below [120]:

1. P. Li, A. Ghasemi, W. Xie, and W. Tian, "Visual closed-loop dynamic model identification of parallel robots based on optical cmm sensor," Electronics, vol. 8, no. 8, p. 836, 2019.

1.7 Outline

This thesis starts with an introduction and literature review on visual servoing along with the research scope and objectives of the thesis. In Chapter 2, the new adaptive switch IBVS is presented. In Chapter 3, the enhanced switch IBVS dealing with feature loss is introduced. In Chapter 4, the PD-SMC method for IBVS is proposed. The new trajectory planning technique is presented in Chapter 5 and the conclusion and future work are explained in Chapter 6.

Chapter 2

Adaptive Switch Image-Based Visual Servoing

2.1 Introduction

This chapter is focused on presenting a new technique of IBVS using a mono camera in eye-in-hand configuration. As it was discussed in 1.3.1, for eye-in-hand configuration, there is still a lack of research work on a stable fast response method of IBVS to be used in industrial applications with the capability to deal with camera uncertainties. To address this issue, a switching scheme for IBVS is proposed in this chapter: the controller switches between rotational and translational movements of the end-effector and separate gains are set for each stage. On the other hand, the proposed adaptive law deals with the uncertainties in the camera parameters.

2.2 Problem Statement

As it is seen in (1.4), the image Jacobian matrix value is dependent on camera intrinsic parameters and hence the uncertainties in these parameters affect the performance of the controller. Considering the fact that camera calibration process could be time-consuming, the goal of this work is to propose a dynamic-based adaptive controller to deal with camera parameters uncertainties and to reduce the response time of the system to make it feasible for industrial applications.

The task aims to move the robot end-effector with an eye-in-hand camera towards the target in order to match the desired image features with the actual image features. When the goal is reached, many further operations can be done by the robot on the target object such as assembling, welding, etc.

2.3 Adaptive Switch Method

It is noticed that the first 3 columns of the image Jacobian matrix (1.4), is related to translational movement of the end-effector and the next 3 columns to rotational part. If the movement is decoupled to translational and rotational phases, the term $(\frac{1}{Z(t)})$ appearing in the translational part can be factorized from image Jacobian matrix in translational phase. This technique provides the chance to overcome the nonlinearity caused by depth parameter and enables us to employ the adaptive methods to estimate the camera parameters in eye-in-hand configuration. Thus, in this chapter, the switching idea is employed to divide the motion of the robot end-effector into three stages: pure rotation, pure translation and the fine tuning stage which consists of both rotation and translation movements. Another merit of switch method is the chance of setting different gain values in control stages, which improves the tracking performance and speed of the controller while maintaining the overall system stability. Also this method overcomes some inherent drawbacks of IBVS, such as its inability to make pure 180° rotation of the camera around its center, etc. This three-stage gain scheduled switch control also considers the nonlinear robot dynamics. An intuitive feature (the angle between desired and actual features) is proposed for determining the switch condition in control laws. The switch method is combined with an adaptive scheme for estimating the camera parameters and adaptive switch method is introduced. Thus, adaptive switch control method breaks the movement of the end-effector into three separate movements and applies different control gains K to each of them while estimating the camera parameters c_u, c_v, f and β (1.5, 1.6).

2.3.1 Adaptive Switch Controller Design

A 6-DOF robot manipulator with the camera installed at the end effector is considered. The dynamic equation of the robot manipulator is shown as:

$$M(q(t))\ddot{q}(t) + C(q(t), \dot{q}(t))\dot{q}(t) + G(q(t)) = \tau, \tag{2.1}$$

where $q(t)$ and $\dot{q}(t))$ are the robot joint positions and velocities, $M(q(t))$ is the inertia matrix, $C(q(t), \dot{q}(t))$ is the Coriolis force, $G(q(t))$ is the gravitational force and τ is the joint torque.

$V_c(t) = [V_{ct}(t)\ \ V_{cr}(t)]^T \in \mathbb{R}^{(6\times1)}$ is defined as the velocity screw of the camera consisting of the translational velocity $V_{ct}(t) \in \mathbb{R}^{(3\times1)}$ and rotational velocity $V_{cr}(t) \in \mathbb{R}^{(3\times1)}$.

$J_R(t) = [J_{Rt}(t)\ J_{Rr}(t)]^T \in \mathbb{R}^{(6\times6)}$ is also defined as the robot Jacobian which is decomposed to the translational part $J_{Rt}(t) \in \mathbb{R}^{(3\times6)}$ and rotational part $J_{Rr}(t) \in \mathbb{R}^{(3\times6)}$.

Thus, the camera velocity can be expressed as:

$$V_C(t) = J_R(t)\dot{q}(t) = \begin{bmatrix} V_{ct}(t) \\ V_{cr}(t) \end{bmatrix} = \begin{bmatrix} J_{Rt}(t)\dot{q}(t) \\ J_{Rr}(t)\dot{q}(t) \end{bmatrix}, \tag{2.2}$$

where $\dot{q}(t) \in \mathbb{R}^{(6\times1)}$ is the robot joint velocity.

With the assumption that all features have the same depth Z, for the i th feature, the image Jacobian matrix in (1.4), can be decomposed to translational part $J_t(t)$ and rotational part $J_r(t)$:

$$J_{img}(t) = \begin{bmatrix} J_t(t) & J_r(t) \end{bmatrix}, \tag{2.3}$$

where,

$$J_t(t) = \begin{bmatrix} \frac{f}{Z(t)} & 0 & -\frac{x_i(t)}{Z(t)} \\ 0 & \frac{f}{Z(t)} & -\frac{y_i(t)}{Z(t)} \end{bmatrix} \tag{2.4}$$

and

$$J_r(t) = \begin{bmatrix} -\frac{x_i(t)y_i(t)}{f} & \frac{f^2+x_i(t)^2}{f} & -y_i(t) \\ \frac{-f^2-y_i(t)^2}{f} & \frac{x_i(t)y_i(t)}{f} & x_i(t) \end{bmatrix}. \tag{2.5}$$

where, the feature coordinates in the image space $x_i(t)$ and $y_i(t)$ related to the intrinsic camera parameters are expressed in (1.5) and (1.6).

Equation (3.3) can be represented as:

$$J_t(t) = \frac{1}{Z(t)} J_t{}'(t), \tag{2.6}$$

where,

$$J_t{}'(t) = \begin{bmatrix} f & 0 & -x_i(t) \\ 0 & f & -y_i(t) \end{bmatrix}. \tag{2.7}$$

By considering (3.1), (3.3) and (2.5), one can rewrite (1.1) for the i th feature $\dot{s}_i(t) \in \mathbb{R}^{(2\times1)}$

as:

$$\begin{aligned}\dot{s}_i(t) = \begin{bmatrix} J_t(t) & J_r(t) \end{bmatrix} \begin{bmatrix} V_{ct}(t) \\ V_{cr}(t) \end{bmatrix} &= J_t(t)V_{ct}(t) + J_r(t)V_{cr}(t) \\ &= J_t(t)J_{Rt}(q(t))\dot{q}(t) + J_r(t)J_{Rr}(q(t))\dot{q}(t).\end{aligned} \tag{2.8}$$

The adaptive controller is designed based on the switch method, which includes three different stages of camera movement. In the first stage, only rotation command of the camera is turned on. In the second stage, only translational movement is active. In the third stage, the classic IBVS control is adopted where both camera rotation and translation are turned on.

In the first stage, the translation is turned off ($V_{ct} = 0$). Therefore Eq.(2.8) becomes as the following:

$$\dot{s}_i = J_r(t)J_{Rr}(q(t))\dot{q}(t). \tag{2.9}$$

In the second stage, the rotation is turned off ($V_{cr} = 0$), thus Eq. (2.8) becomes:

$$\dot{s}_i(t) = \frac{1}{Z(t)} J_t(t)' J_{Rt}(q(t))\dot{q}(t). \tag{2.10}$$

Finally, in the third stage both translation and rotation movements of the camera are switched on. Thus one has:

$$\dot{s}_i(t) = \frac{1}{Z(t)} J_t(t)' J_{Rt}(q(t))\dot{q}(t) + J_r(t)J_{Rr}(q(t))\dot{q}(t). \tag{2.11}$$

The adaptive controller generates the robot joint torques as the control commands and an adaptive law is developed to estimate the camera parameters. After a transient adaptation process, the feature points reach to the desired ones in the image space.

In the first stage, the camera is in pure rotation. Consider i th feature. Referring to (2.2), the robot Jacobian matrix is represented as:

$$J_R(t) = \begin{bmatrix} a_{11}(t) & \dots & a_{16}(t) \\ \vdots & \ddots & \vdots \\ a_{61}(t) & \dots & a_{66}(t) \end{bmatrix}. \tag{2.12}$$

Using the "ˆ" notation for the uncertain camera parameters and considering (1.5), (1.6) and (2.12), the first element of the i th feature $\dot{s}_i(t) \in \mathbb{R}^{(2\times1)}$ in (2.9) is:

$$\begin{aligned}\dot{s}_{i1}(t) = \sum_{k=1}^{6} \Bigg[& a_{5k}(t)\dot{q_k}(t)+ \\ & (\frac{1}{\hat{f}^2\hat{\beta}^2})[a_{4k}(t)u_i(t)v_i(t)\dot{q_k}(t) - a_{5k}(t)u_i(t)^2\dot{q_k}(t)]+ \\ & (\frac{\hat{c_u}}{\hat{f}^2\hat{\beta}^2})[-a_{4k}(t)v_i(t)\dot{q_k}(t) + 2a_{5k}(t)u_i(t)\dot{q_k}(t)]- \\ & (\frac{\hat{c_v}}{\hat{f}^2\hat{\beta}^2})[-a_{4k}(t)u_i(t)\dot{q_k}(t)] + (\frac{\hat{c_u}\hat{c_v}}{\hat{f}^2\hat{\beta}^2})a_{4k}(t)\dot{q_k}(t)- \\ & (\frac{\hat{c_u}^2}{\hat{f}^2\hat{\beta}^2})a_{5k}(t)\dot{q_k}(t) + (\frac{1}{\hat{f}\hat{\beta}})a_{6k}(t)v_i(t)\dot{q_k}(t)- \\ & (\frac{\hat{c_v}}{\hat{f}\hat{\beta}})a_{6k}(t)\dot{q_k}(t)\Bigg].\end{aligned} \tag{2.13}$$

The second element can be expressed in the following.

$$\begin{aligned}\dot{s}_{i2}(t) = \sum_{k=1}^{6} \Bigg[& a_{4k}(t)\dot{q_k}(t)+ \\ & (\frac{1}{\hat{f}^2\hat{\beta}^2})[a_{4k}(t)v_i(t)^2\dot{q_k}(t) - a_{5k}(t)u_i(t)v_i(t)\dot{q_k}(t)]+ \\ & (\frac{\hat{c_u}}{\hat{f}^2\hat{\beta}^2})[a_{5k}(t)v_i(t)\dot{q_k}(t)]+ \\ & (\frac{\hat{c_v}}{\hat{f}^2\hat{\beta}^2})[-2a_{4k}(t)v_i(t)\dot{q_k}(t) + a_{5k}(t)u_i(t)\dot{q_k}(t)]- \\ & (\frac{\hat{c_u}\hat{c_v}}{\hat{f}^2\hat{\beta}^2})a_{5k}(t)\dot{q_k}(t) - (\frac{1}{\hat{f}\hat{\beta}})a_{6k}(t)u_i(t)\dot{q_k}(t)+ \\ & (\frac{\hat{c_u}}{\hat{f}\hat{\beta}})a_{6k}(t)\dot{q_k}(t) + (\frac{\hat{c_v}^2}{\hat{f}^2\hat{\beta}^2})a_{4k}(t)\dot{q_k}(t)\Bigg].\end{aligned} \tag{2.14}$$

The uncertain parameters can be decoupled from the known values in above equation. Thus, for i th image feature, Eq. (2.9) can be represented as the linear combination of the regression matrix and the estimated parameters as follows:

$$\dot{\hat{s}}_i(t) = \hat{J}_r(t)J_{Rr}(q(t))\dot{q}(t) = Y_1(q(t),\dot{q}(t),s(t))\hat{\theta}(t) = \begin{bmatrix} Y_{1(1,1)} & \dots & Y_{1(1,10)} \\ Y_{1(2,1)} & \dots & Y_{1(2,10)} \end{bmatrix} \begin{bmatrix} \hat{\theta}_1 \\ \vdots \\ \hat{\theta}_{10} \end{bmatrix}, \tag{2.15}$$

where $Y_1(q(t), \dot{q}(t), s(t)) \in \mathbb{R}^{(2\times 10)}$ is the regression matrix which is independent of the camera parameters, $\hat{\theta}(t) \in \mathbb{R}^{(10\times 1)}$ including all the estimated parameters is represented in this form:

$$\hat{\theta}(t) = [1 \; \frac{1}{\hat{f}^2\hat{\beta}^2} \; \frac{\hat{c_u}}{\hat{f}^2\hat{\beta}^2} \; \frac{\hat{c_v}}{\hat{f}^2\hat{\beta}^2} \; \frac{\hat{c_u}\hat{c_v}}{\hat{f}^2\hat{\beta}^2} \; \frac{\hat{c_u}^2}{\hat{f}^2\hat{\beta}^2} \; \frac{1}{\hat{f}\hat{\beta}} \; \frac{\hat{c_v}}{\hat{f}\hat{\beta}} \; \frac{\hat{c_u}}{\hat{f}\hat{\beta}} \; \frac{\hat{c_v}^2}{\hat{f}^2\hat{\beta}^2}]^T \; . \tag{2.16}$$

Three main camera parameters can be extracted from the above equation as $\hat{c_v}$, $\hat{c_u}$ and $\hat{f}\hat{\beta}$.

In this study it is assumed that depth Z is constant and known. Thus, by referring to equations (2.10) and (2.11), it is noted that similar formulation can be expressed for the second and third stages as well. In the second stage, a similar update law continues estimating the camera parameters. For brevity, the derivation of corresponding equations to (2.13) and (2.15) is omitted here. In the third stage, the controller switches back to conventional IBVS and uses the estimated camera parameters in the two previous stages to calculate the image Jacobian matrix.

In order to fulfill the switch between stages, a criteria is needed to facilitate the decoupled movement. One criteria is defined as the norm of feature errors in [121]. In this research work, a more intuitive and effective criteria is proposed as the angle between the desired and actual features. Thus, a new feature α is defined as the angle between the desired features and the actual features as illustrated in Figure 2.1. This feature is used as the criteria of switching between stages. Once the angle α reaches the predefined threshold, the control law is switched to the one in the next stage.

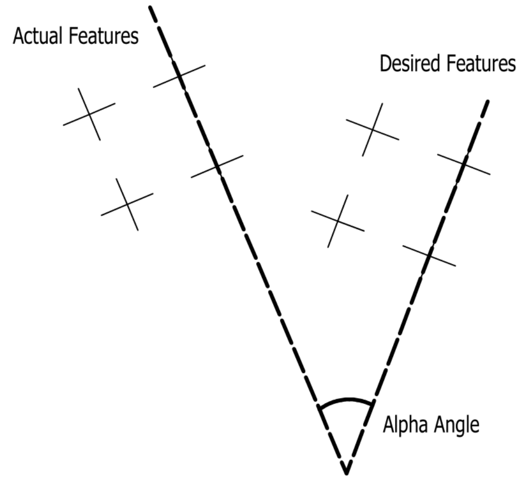


Figure 2.1: New feature–angle α- the angle between the desired and actual features

The overall control law is proposed as the following,

$$\begin{cases} \tau_{s1} = G(q(t)) - K_{v1}\dot{q}(t) - (\hat{J}_r(t)J_{Rr}(q(t))^T K_{p1}\tilde{s}(t), & |\alpha| \geq \alpha_0 \\ \tau_{s2} = G(q(t)) - K_{v2}\dot{q}(t) - (\hat{J}_t(t)J_{Rt}(q(t))^T K_{p2}\tilde{s}(t), & \alpha_1 \leq |\alpha| < \alpha_0 \\ \tau_{s3} = G(q(t)) - K_{v3}\dot{q}(t) - (\hat{J}_{img}(t)J_R(q(t))^T K_{p3}\tilde{s}(t), & otherwise \end{cases}, \tag{2.17}$$

where $\tilde{s}(t) = s(t) - s_d$ is the position error of n feature points and K_{vi} and K_{pi} are symmetric positive definite gain matrices at each stage and τ_{si} $(i = 1, 2, 3)$ is the calculated torque for robot's joints, α_0 and α_1 are two predefined thresholds for the control law to switch to the next stage.

The uncertain camera parameters $\hat{\theta}(t)$ are estimated only in the first and second stage. In the third stage the camera parameters are not updated and the previous values are used in the control law. Thus the update law is proposed as follows:

$$\begin{cases} \dot{\hat{\theta}}(t) = K_a^{-1} Y_1^T K_{p1}\tilde{s}(t), & |\alpha| \geq \alpha_0 \\ \dot{\hat{\theta}}(t) = K_a^{-1} Y_2^T K_{p2}\tilde{s}(t), & \alpha_1 \leq |\alpha| < \alpha_0 \\ \dot{\hat{\theta}}(t) = 0, & otherwise \end{cases}, \tag{2.18}$$

where K_a and K_{Pi} $(i = 1, 2)$ are positive definite diagonal matrices, Y_1 is the regression matrix for the first stage as it is shown in (2.15) and similarly Y_2 is the regression matrix for the second stage. The block diagram of the proposed control method is shown in Figure 2.2.

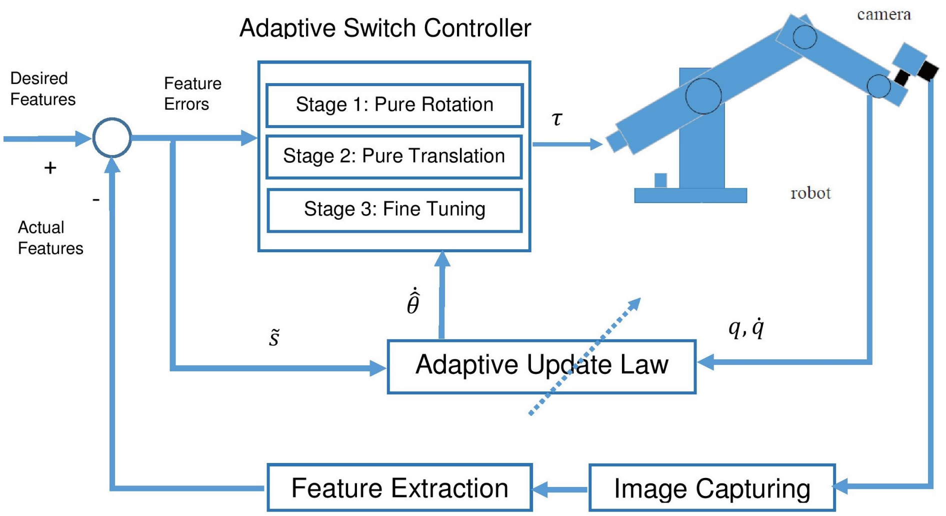


Figure 2.2: Block diagram of the proposed adaptive switch controller

2.4 Stability Proof

Regarding the robot dynamic equation (2.1), the following property is satisfied [122]:

property 1. The matrix $\dot{M}(q(t)) - 2C(q(t), \dot{q}(t))$ is skew-symmetric. It means:

$$\delta^T[\dot{M}(q(t)) - 2C(q(t), \dot{q}(t))]\delta = 0, \tag{2.19}$$

for all $\delta \in \mathbb{R}^n$.

Theorem 1: Consider a 6-DOF robot with an eye-in-hand camera installed at the end-effector (2.1). The proposed adaptive switch controller (2.17) with the update law (2.18) makes the feature error $\tilde{s}(t)$ asymptotically converge to zero, i.e. $\lim_{t\to\infty} \tilde{s}(t) = 0$.

proof:

Considering the first stage of control, a Lyapunov function candidate is defined as:

$$V(t) = \frac{1}{2}[\dot{q}(t)^T M(q(t))\dot{q}(t) + \tilde{s}(t)^T K_{p1}\tilde{s}(t) + \tilde{\theta}(t)^T K_a \tilde{\theta}(t)], \tag{2.20}$$

where $\tilde{\theta}(t) = \theta(t) - \hat{\theta}(t)$ is the difference between real camera parameters $\theta(t)$ and the estimated one $\hat{\theta}(t)$. Differentiating the above equation yields:

$$\dot{V}(t) = \dot{q}(t)^T M(t)\ddot{q}(t) + \frac{1}{2}\dot{q}(t)^T \dot{M}(t)\dot{q}(t) + \tilde{s}(t)^T K_{p1}\dot{s}(t) - \tilde{\theta}(t)^T K_a \dot{\hat{\theta}}(t). \tag{2.21}$$

By substituting dynamic equation (2.1), one can rewrite the above equation as:

$$\begin{aligned} \dot{V}(t) = \dot{q}(t)^T(\tau - G(q(t)) + \frac{1}{2}\dot{q}(t)^T[\dot{M}(t)\dot{q}(t) - 2C(q(t), \dot{q}(t))]\dot{q}(t) \\ + \tilde{s}(t)^T K_{p1}\dot{s}(t) - \tilde{\theta}(t)^T K_a \dot{\hat{\theta}}(t). \end{aligned} \tag{2.22}$$

Now Property 1 is used and the value of τ is substituted from the first stage of adaptive control law (2.17). Therefore, one has:

$$\dot{V}(t) = \dot{q}(t)^T(-K_{v1})\dot{q}(t) - \dot{q}(t)^T[\hat{J}_r(t)J_R(q(t))]^T K_{p1}\tilde{s}(t) + \tilde{s}(t)^T K_{p1}\dot{s}(t) - \tilde{\theta}(t)^T K_a \dot{\hat{\theta}}(t). \tag{2.23}$$

Using (2.9) and first stage of update law(2.18), one obtains:

$$\dot{V}(t) = \dot{q}(t)^T(-K_{v1})\dot{q}(t) - \hat{\theta}(t)^T Y_1^T K_{p1}\tilde{s}(t) + \theta(t)^T Y_1^T K_{p1}\tilde{s}(t) - \tilde{\theta}^T Y_1^T K_{p1}\tilde{s}(t). \tag{2.24}$$

Then one has:

$$\dot{V}(t) = \dot{q}(t)^T(-K_{v1})\dot{q}(t) \leq 0. \tag{2.25}$$

Thus, according to Barbarat's Lemma [123], one can infer that $\lim_{t\to\infty} \tilde{s}(t) = 0$. Therefore the stability of the proposed adaptive controller is proved for the controller in the first stage. For the second stage, by following the similar procedure, one can have:

$$\dot{V}(t) = \dot{q}(t)^T(-K_{v2})\dot{q}(t) - \hat{\theta}^T Y_2^T K_{p2}\tilde{s}(t) + \theta^T Y_2^T K_{p1}\tilde{s}(t) - \tilde{\theta}(t)^T Y_2^T K_{p2}\tilde{s}(t). \tag{2.26}$$

It is assumed that $K_{p1} = K_{p2} + K_\alpha$. Thus the above equation is simplified as:

$$\dot{V}(t) = \dot{q}(t)^T(-K_{v2})\dot{q}(t) + \theta^T Y_2^T K_\alpha \tilde{s}(t) = \dot{q}(t)^T(-K_{v2})\dot{q}(t) + \dot{s}(t)^T K_\alpha \tilde{s}(t). \tag{2.27}$$

The above equation can be also represented as:

$$\dot{V}(t) = -2V(t) + \epsilon_1, \tag{2.28}$$

where ϵ_1 is defined:

$$\begin{aligned}\epsilon_1 &= \dot{q}(t)^T(-K_{v2})\dot{q}(t) + \dot{s}^T K_\alpha \tilde{s}(t) + 2V(t)\\ &= \dot{q}(t)^T(-K_{v2})\dot{q}(t) + \dot{s}(t)^T K_\alpha \tilde{s}(t) + \dot{q}(t)^T M(q(t))\dot{q}(t) + \tilde{s}(t)^T K_{p1}\tilde{s}(t) + \tilde{\theta}(t)^T K_a \tilde{\theta}(t),\end{aligned} \tag{2.29}$$

which can be simplified as:

$$\epsilon_1 = \dot{q}(t)^T(M(q(t)) - K_{v2})\dot{q}(t) + \dot{s}(t)^T K_\alpha \tilde{s}(t) + \tilde{s}(t)^T K_{p1}\tilde{s}(t) + \tilde{\theta}(t)^T K_a \tilde{\theta}(t). \tag{2.30}$$

K_{v2} is chosen so that $M(q(t)) - K_{v2} \geq 0$. The following inequality can be inferred from (2.30):

$$\epsilon_1 \leq \dot{q}(t)^T(M(q(t)) - K_{v2})\dot{q}(t) + \left|\dot{s}(t)^T K_\alpha \tilde{s}(t)\right| + \tilde{s}(t)^T K_{p1}\tilde{s}(t) + \tilde{\theta}(t)^T K_a \tilde{\theta}(t) = \bar{\epsilon}_1, \tag{2.31}$$

in which $\bar{\epsilon}_1$ is positive. Thus the following inequality holds:

$$\dot{V}(t) \leq -2V(t) + \bar{\epsilon}_1. \tag{2.32}$$

Therefore from the boundedness theorem, the stability of the system in the second stage is proved.

In the third stage, the similar procedure is followed and one can have:

$$\dot{V}(t) = \dot{q}(t)^T(-K_{v3})\dot{q}(t) - \hat{\theta}(t)^T Y_3^T K_{p3}\tilde{s}(t) + \theta^T Y_3^T K_{p1}\tilde{s}(t). \tag{2.33}$$

It can be assumed: $K_{p1} = K_{p3} + K_\beta$. Thus the above equation is simplified:

$$\begin{aligned}\dot{V}(t) &= \dot{q}(t)^T(-K_{v3})\dot{q}(t) + \tilde{\theta}(t)^T Y_3^T K_{p3}\tilde{s}(t) + \theta^T Y_3^T K_\beta \tilde{s}(t)\\ &= \dot{q}(t)^T(-K_{v3})\dot{q}(t) + \tilde{\theta}(t)^T Y_3^T K_{p3}\tilde{s}(t) + \dot{s}(t)^T K_\beta \tilde{s}(t).\end{aligned} \tag{2.34}$$

The above equation can be represented as:

$$\dot{V}(t) = -2V(t) + \epsilon_2, \tag{2.35}$$

where ϵ_2 is defined:

$$\begin{aligned}\epsilon_2 &= \dot{q}(t)^T(-K_{v3})\dot{q}(t) + \tilde{\theta}(t)Y_3^T K_{p3}\tilde{s}(t) + \dot{s}(t)^T K_\beta \tilde{s}(t) \\ +2V(t) &= \dot{q}(t)^T(-K_{v3})\dot{q}(t) + \tilde{\theta}(t)Y_3^T K_{p3}\tilde{s}(t) + \dot{s}(t)^T K_\beta \tilde{s}(t) \\ &+\dot{q}(t)^T M(q(t))\dot{q}(t) + \tilde{s}(t)^T K_{p1}\tilde{s}(t) + \tilde{\theta}(t)^T K_a \tilde{\theta}(t),\end{aligned} \tag{2.36}$$

which can be simplified as:

$$\epsilon_2 = \dot{q}(t)^T(M(q(t)) - K_{v3})\dot{q}(t) + \tilde{\theta}(t)Y_3^T K_{p3}\tilde{s}(t) + \dot{s}(t)^T K_\beta \tilde{s}(t) + \tilde{s}(t)^T K_{p1}\tilde{s}(t) + \tilde{\theta}(t)^T K_a \tilde{\theta}(t). \tag{2.37}$$

K_{v3} is chosen so that $M(q(t)) - K_{v3} \geq 0$. The following inequality can be inferred from (2.37):

$$\begin{aligned}\epsilon_2 \leq \dot{q}(t)^T(M(q(t)) - K_{v3})\dot{q}(t) + \left|\dot{s}(t)^T K_\beta \tilde{s}(t)\right| \\ + \left|\tilde{\theta}(t)Y_3^T K_{p3}\tilde{s}(t)\right| + \tilde{s}(t)^T K_{p1}\tilde{s}(t) + \tilde{\theta}(t)^T K_a \tilde{\theta}(t) = \bar{\epsilon}_2,\end{aligned} \tag{2.38}$$

in which $\bar{\epsilon}_2$ is positive. Thus the following inequality holds:

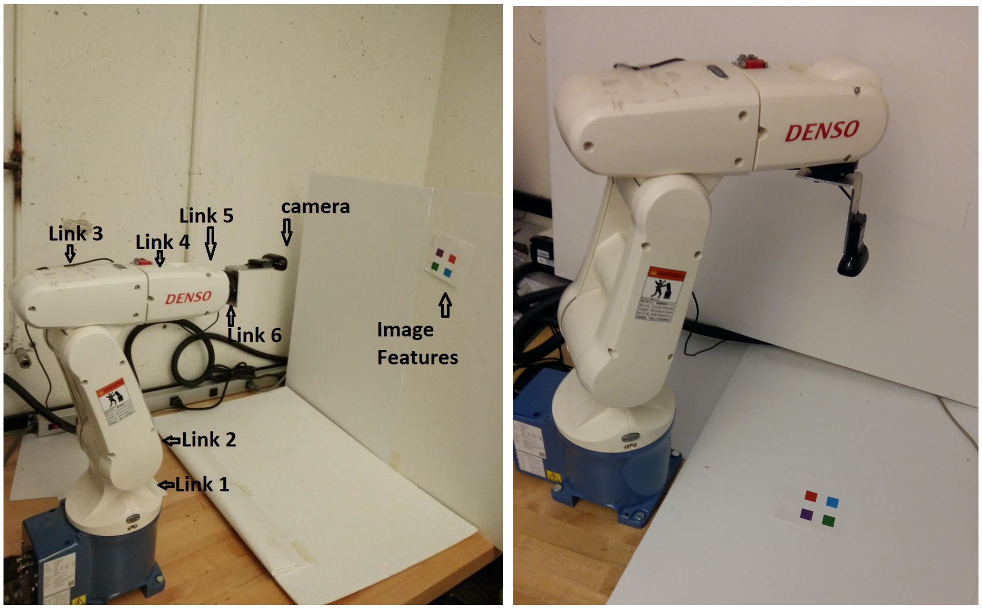
$$\dot{V}(t) \leq -2V(t) + \bar{\epsilon}_2. \tag{2.39}$$

From the boundedness theorem, the stability of the system in the third stage is proved.

2.5 Experimental set-up

In this research work, the proposed methods in all chapters are tested and validated experimentally using the following test-bed.

The experimental set-up includes a VS-6556G Denso robot and a camera mounted on the end-effectors (Figure 2.3). The robot has 6-DOFs and its repeatability is ±0.02 mm in each of x, y and z directions. A camera is used as the vision system, mounted on the robot's end-effector. The camera model is Logitech Webcam HD 720p, which captures the video with a resolution of 1280 × 720 pixels and the capture rate of 30 frames per second. Other camera characteristics are given in Table 2.1.



(a)

(b)

Figure 2.3: Experimental set-up: 6-DOF Denso robot

The mechanical parameters of the robot links (Figure 2.3) are given in Table 2.2. I_x, I_y & I_z are the moments of inertia of the links around x, y & z axis of the frame attached to the center of gravity of the links respectively.

Table 2.1: Camera parameters

Parameter	**Value**
Focal length (f)	0.004 (m)
X axis scaling factor (β)	110000 (pixel/m)
Y axis scaling factor (β)	110000 (pixel/m)
Principal point of x axis (c_u)	120 (pixel)
Principal point of y axis (c_v)	187 (pixel)

Table 2.2: Mechanical parameters of Denso robot links

	Mass(kg)	**Moments of Inertia** ($kg.m^2$)		
		I_x	I_y	I_z
Link 1	3.1	0.013	0.0051	0.0125
Link 2	2.2	0.0054	0.0168	0.0155
Link 3	2.0	0.0053	0.0079	0.0044
Link 4	1.3	0.0054	0.0017	0.0054
Link 5	0.8	0.001	0.001	0.00037
Link 6	0.2	7×10^-5	7×10^-5	0.0001

The experimental setup also consists of a controller and two computers through a double PC bilateral teleoperation (Figure 2.4). PC No.1 (Master PC) communicates with the master robot and transmits the commands to the Slave PC (PC No.2) over the communication network. The slave PC also communicates with the slave robot (Denso robot) and obtains the camera data and sends it back to the master PC over the communication network [124]. The image data taken by the camera is sent to an image processing program written by using Computer Vision Toolbox of MATLAB. This program extracts the center coordinates of the features with the rate of 0.001 seconds, which serve as the feedback signal to the visual servoing controller.

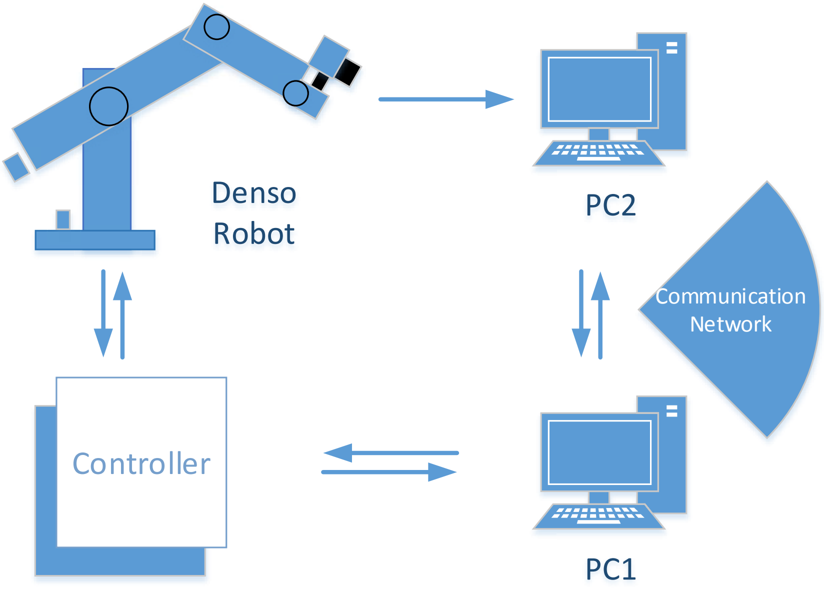


Figure 2.4: Experimental set-up structure

2.6 Experimental Results

In this section the efficiency of the adaptive switch method is evaluated with the experimental set-up described in 2.5. Performance of the adaptive switch method is compared with that of switch IBVS [118, 121] and conventional IBVS, three experiments are conducted with three different initial angles between desired and actual features (α in Figure 2.1). The threshold angles of switching between control stages (α_0 and α_1 in (2.17)) are set as 10.3° and 8.5°. The objective of all these tests is to lead the end-effector in a way that the actual extracted image features match the desired ones. In all these tests, the number of image features is chosen as four. The initial and desired feature point positions of the four features are given in Table 3.3. The controller gains ($K_{p1}, K_{p2}, K_{p3}, K_1, K_2$ and K_3) are tuned by trial and error to obtain the acceptable performance. However, the optimal values can be obtained by an optimization process, which is beyond the scope of this research.

The norm of feature errors (NFE) is defined as below,

$$NFE = \sum_{i=1}^{4} \sqrt{(x_i - x_{di})^2 + (y_i - y_{di})^2}, \tag{2.40}$$

where x_i and y_i are the $i^{\text{ th}}$ feature coordinates and x_{di} and y_{di} are $i^{\text{ th}}$ desired feature coordinates in image plane. In all the tests, the threshold value of NFE is set as 0.005, and when NFE reaches this value, the robot stops. Also to evaluate the tracking performance, Integral Time-weighted Absolute Error index (ITAE) is defined as below:

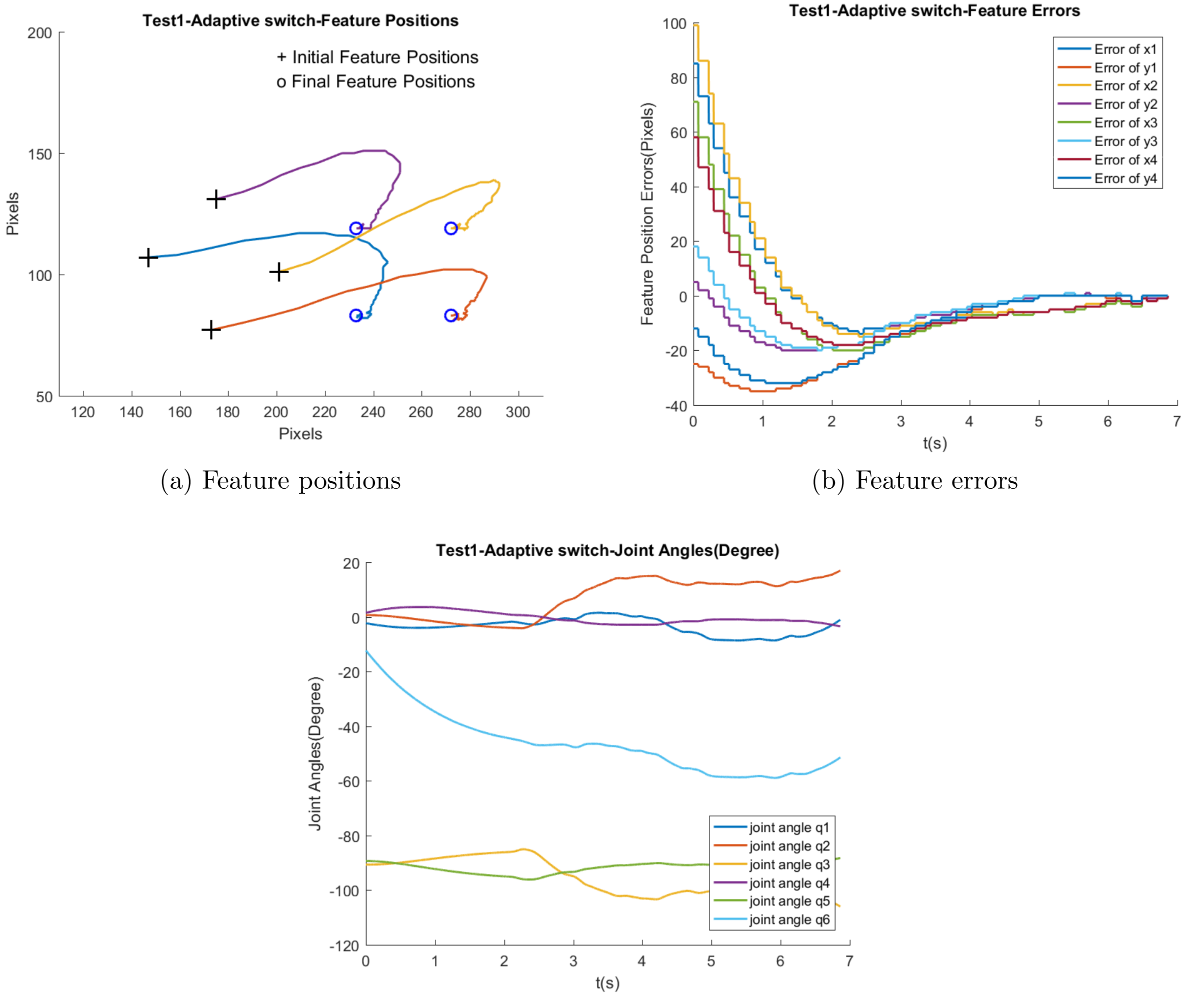
$$ITAE = \int_0^\infty t\,|NFE|\,dt, \tag{2.41}$$

in witch t is the execution time of the test.

Test 1: the initial angle α between the actual features and desired one is 50. K_{p1}, K_{p2} and K_{p3} values in (2.17), are $9 \times 10^{-5} \times I_8$, $9 \times 10^{-4} \times I_8$ and $9 \times 10^{-3} \times I_8$ respectively (I_8 is 8×8 identity matrix). While K_1, K_2 and K_3 in switch method (Eq. (14) of [118]) are 1, 0.1 and 0.05 respectively and constant K in (1.8) for traditional IBVS is set as 0.05. Figure 2.5 indicates the performance of the adaptive switch method. Figure 2.5b shows the eight feature errors' changes with time. The path of features from initial position to final one is illustrated in Figure 2.5a. The feature coordinates are represented in pixel unit. The robot joint angles during the task are shown in Figure 2.5c. Figure 2.6 compares the performance of adaptive switch with that of traditional IBVS and switch method. Figure 2.6a compares NFE (in pixels) of traditional IBVS and adaptive switch IBVS and Figure 2.6b shows the performance comparison between switch IBVS and adaptive switch IBVS. As illustrated in the figure, adaptive switch IBVS demonstrates 74% and 62% shorter response time compared to traditional IBVS and switch IBVS respectively.

Test 2: the initial angle α between the actual features and desired one is 117°. K_{P1}, K_{P2}, K_{P3}, K_1, K_2, K_3 and K values are the same as those in Test 1. Figure 2.7 shows the performance of adaptive switch method and Figure 2.8 represents the comparison of adaptive switch IBVS with traditional IBVS and switch IBVS. It is noted that adaptive switch has a 28% faster response compared to the switch method, while traditional IBVS is not able to complete the task.

Test 3: the initial angle α between the actual features and desired one is 180°. All the designed parameters are kept the same as those in the previous tests. Similar to previous tests, Figure 2.9 shows the performance of adaptive switch method and Figure 2.10 demonstrates the performance comparison of adaptive switch IBVS with switch method and traditional IBVS. The results show that the proposed method is able to overcome one of the inherent drawbacks of IBVS and perform the 180° rotation of the camera around its center

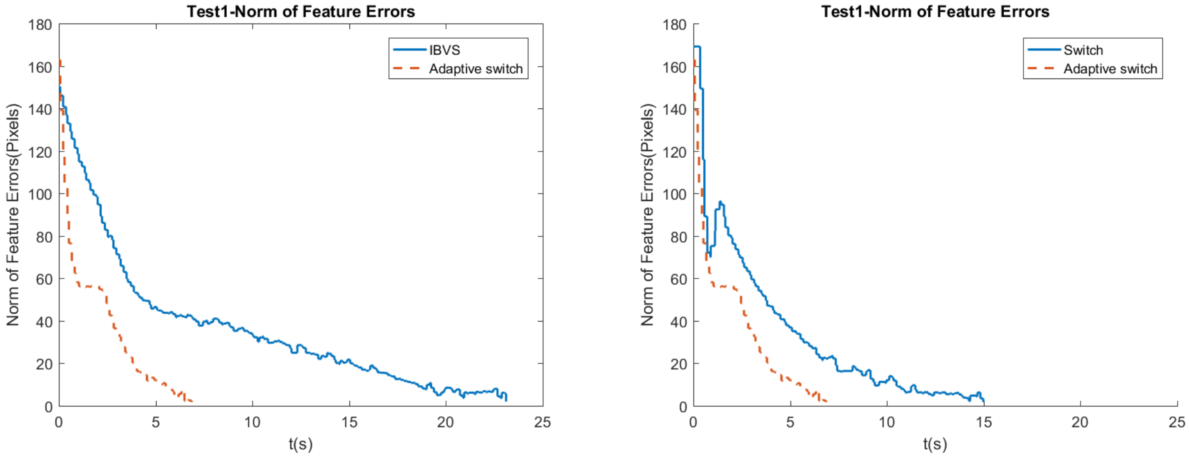


(a) Feature positions

(b) Feature errors

(c) Robot joint angles

Figure 2.5: Test 1 ($\alpha = 50$)- adaptive switch IBVS performance



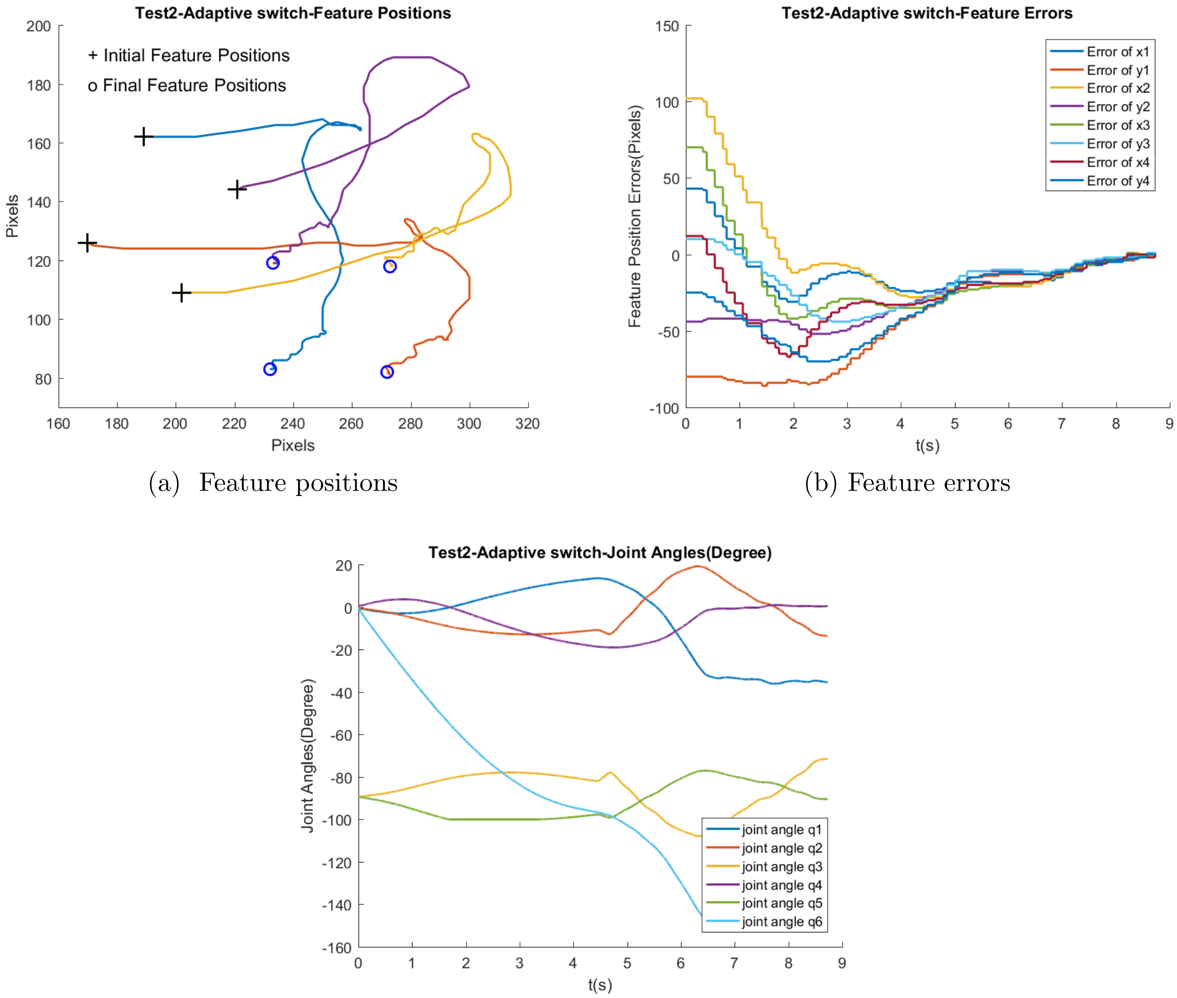
(a) Norm of feature errors-IBVS vs adaptive switch (b) Norm of feature errors-switch vs adaptive switch

Figure 2.6: Test 1 ($\alpha = 50$)- performance comparison of adaptive switch vs traditional IBVS & switch method

successfully.

The results of the experiments are summarized in Tables 2.4 and 2.5. In Table 2.4 the response time of conventional IBVS, switch and adaptive switch IBVS are given for the three tests.

It is worth mentioning that in the performed tests the estimated camera parameters may not necessarily converge to the real values due to the property of the signals. However, the convergence of the parameters to the real values is the indication of controllers' performance. In the experiment, the IBVS task is to move the end-effector so that the image features match the desired ones which cannot guarantee the richness of the signal. However, adaptive control system can deliver good control performance because of its time-varying nature and well-tuned control gains.

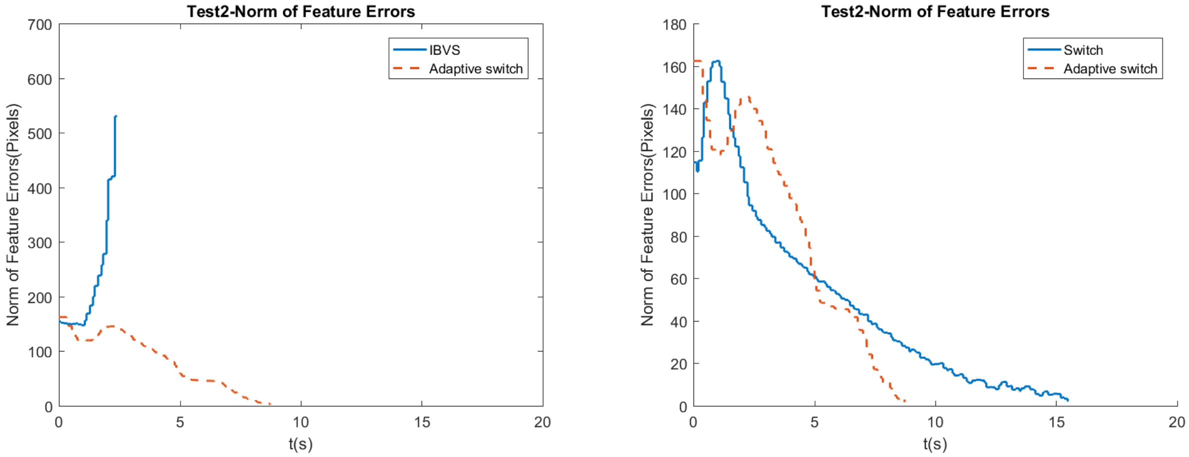


(a) Feature positions

(b) Feature errors

(c) Robot joint angles

Figure 2.7: Test 2 ($\alpha = 117$)- adaptive switch IBVS performance

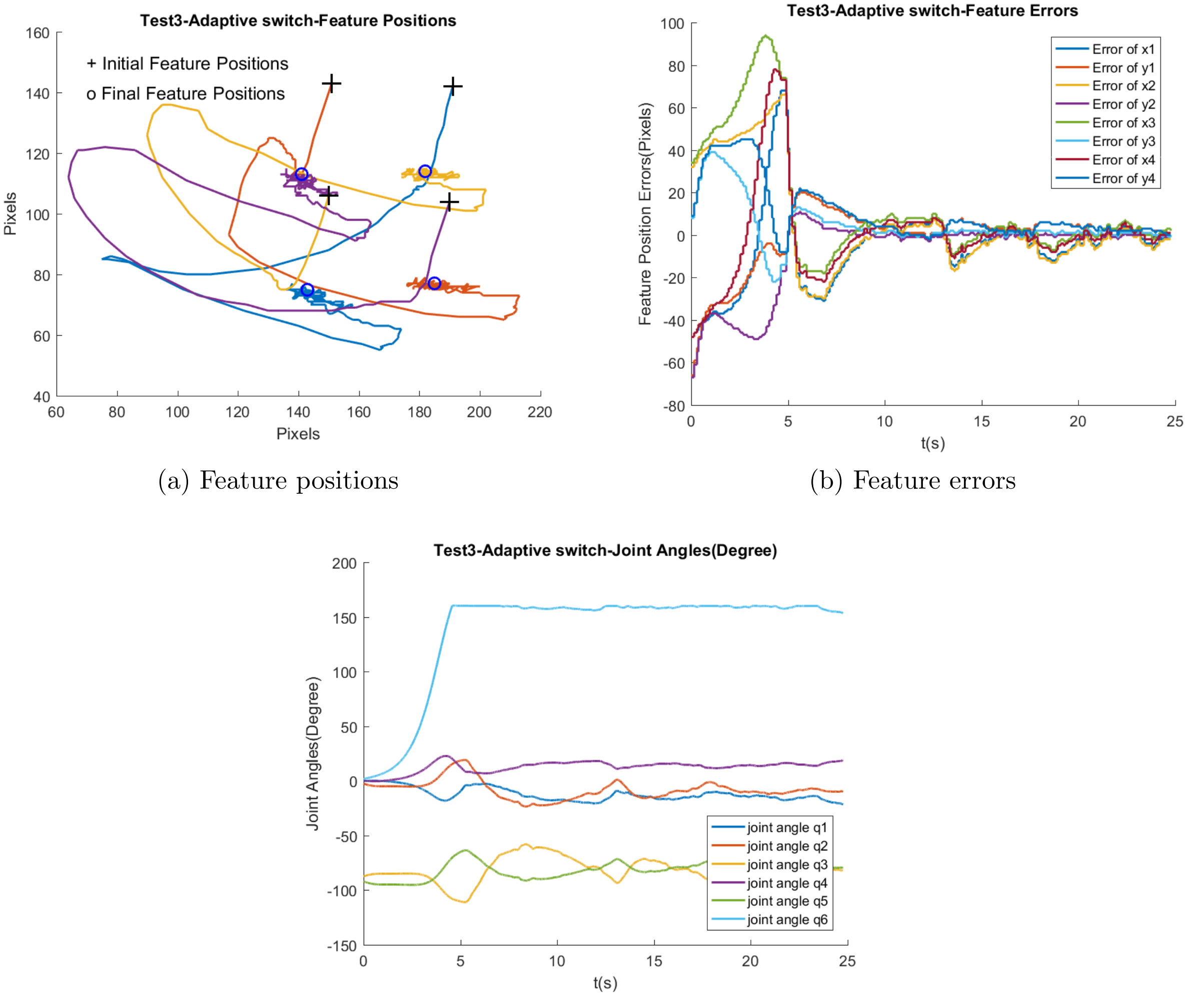


(a) Norm of feature errors-IBVS vs adaptive switch (b) Norm of feature errors-switch vs adaptive switch

Figure 2.8: Test 2 ($\alpha = 117$)- performance comparison of adaptive switch vs traditional IBVS & switch method

Table 2.3: Initial (I) and Desired (D) feature point positions in pixel

		Point 1	**Point 2**	**Point 3**	**Point 4**
		(x, y)	(x, y)	(x, y)	(x, y)
Test 1	I	146 107	172 76	200 99	175 131
	D	232 82	272 82	272 119	233 119
Test 2	I	189 150	170 114	203 96	222 132
	D	232 82	272 82	272 119	233 119
Test 3	I	191 148	151 149	150 111	190 110
	D	143 76	183 76	184 114	142 113

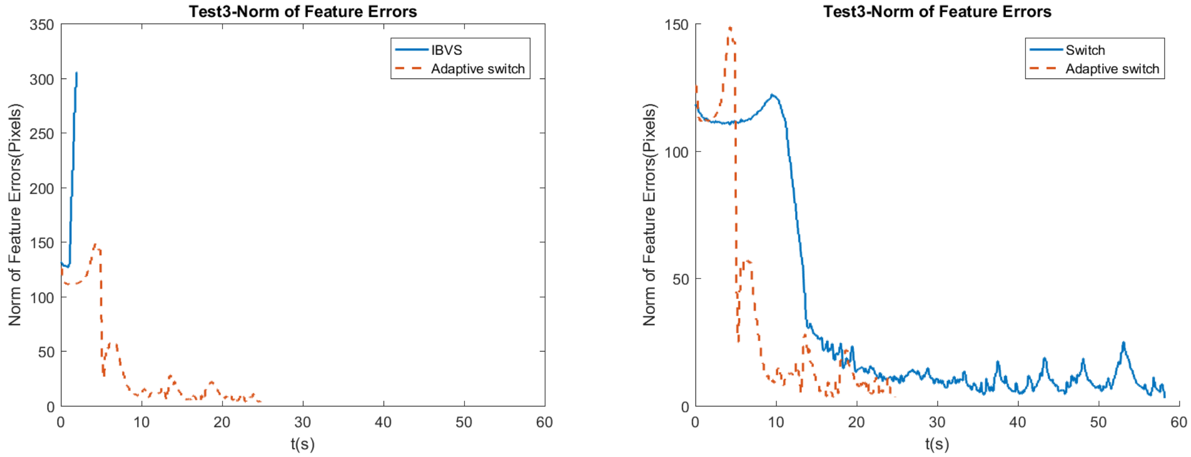


(a) Feature positions

(b) Feature errors

(c) Robot joint angles

Figure 2.9: Test 3 ($\alpha = 180$)- adaptive switch IBVS performance



(a) Norm of feature errors-IBVS vs adaptive switch (b) Norm of feature errors-switch vs adaptive switch

Figure 2.10: Test 3 ($\alpha = 180$)- performance comparison of adaptive switch vs traditional IBVS & switch method

Table 2.4: Comparison of response time for IBVS, switch & proposed adaptive switch method

	Response Time(s)		
	Test 1	**Test 2**	**Test 3**
IBVS	26.58	NA	NA
Switch	17.69	18.5	58
Adaptive Switch	6.86	11.54	24.76

Table 2.5: Comparison of tracking performance (ITAE index) for IBVS, switch & proposed adaptive switch method

	ITAE		
	Test 1	**Test 2**	**Test 3**
IBVS	9.18×10^3	NA	NA
Switch	3.18×10^3	6.16×10^3	2.7×10^4
Adaptive Switch	472.5	4.7×10^3	5.3×10^3

The proposed adaptive switch method is proved to be capable of performing the task for all α angles including 180°. Its response time is 48-78% less than that of switch method. In comparison to conventional IBVS, it is proved to have around 75% faster response for the α angles below 90°. In the case where this angle exceeds 90°, the conventional IBVS fails to complete the task while adaptive switch performs it successfully and better than the switch IBVS does. Table 2.5 compares the tracking performance of the three methods. It is shown that adaptive Switch has the least ITAE which means the best tracking performance. Also, Switch method shows better performance compared to conventional IBVS.

2.7 Summary

In this chapter, an adaptive switch IBVS is proposed for an industrial robot with monocular camera in eye-in-hand configuration. A three stage control scheme is proposed to realize the decoupled rotational and translational movement. The update laws have been developed for estimating the camera intrinsic parameters. The designed controller can overcome some of the inherent drawbacks of traditional IBVS and switch IBVS. The proposed method has been tested in a 6-DOF robotic system with an eye-in-hand camera installed at the end-effector. The experimental results show that response time of this method is almost 75% less than that of traditional IBVS and 48-78% less than that of switch method. Moreover, in the cases where the angle between initial and desired image features is greater than 90°, IBVS normally cannot perform the task while the adaptive switch method performs the task successfully. Especially in the tasks where 180° rotation of the camera around its center is needed, adaptive switch method finishes the task successfully while IBVS fails and the switch IBVS does it with slower response. It is shown that the tracking performance of the proposed method has been improved compared with the switch method and traditional IBVS. The results validate the effectiveness of the proposed adaptive switch method in industrial applications.

Chapter 3

Enhanced Switch Image-Based Visual Servoing Dealing with Features Loss

3.1 Introduction

As it was discussed in previous chapters, many studies have been conducted to overcome the weaknesses of IBVS and improve its efficiency [65, 68, 116, 125]. However, the performance of most reported IBVS is not sufficiently high to meet the requirements of industrial applications [126]. An efficient IBVS feasible for practical robotic operations requires a fast response with strong robustness to feature loss. In the previous chapter, an adaptive switch method was proposed and demonstrated that the controller was able to improve the speed and tracking performance of IBVS and avoid some of its inherent drawbacks. However, feature loss caused by the camera's limited field of view (FOV) still prevents the method from being fully efficient and being applicable to real industrial robots.

In this chapter, an enhanced switch image-based visual servoing (ESIBVS) method is presented in which a Kalman filter-based feature prediction algorithm is proposed and is combined with the proposed switch method in previous chapter to make the switch IBVS control robust in reaction to feature loss. The feature prediction algorithm can predict the lost feature points based on the previously-estimated points. The switch control with the improved tracking performance along with the robustness to feature loss makes it more feasible for industrial robotic applications. To validate the proposed controller, extensive simulations and experiments have been conducted on a 6-DOF Denso robot with a monocular eye-in-hand vision system.

3.2 Problem Statement

While guiding the robot end-effector to make the desired image features match the actual ones, some unexpected situations may occur in IBVS. The first case is feature loss: i.e., some or all of the image features may go beyond the camera's FOV (Figure 3.1). The second case is feature occlusion: i.e., some or all of the image features temporarily become invisible to the camera due to obstacles. The goal of this chapter is to improve the performance of IBVS in terms of response time and tracking performance, while dealing with the feature loss situation. To reach this goal, the performance of the switch method [118, 121] is enhanced when it is combined with the proposed feature reconstruction algorithm.

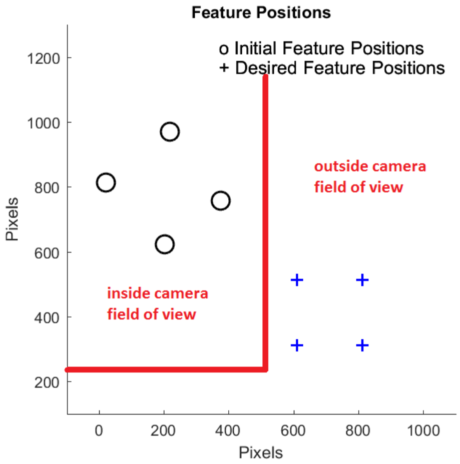


Figure 3.1: Desired and initial feature positions inside and outside the camera's field of view.

3.3 Feature Reconstruction Algorithm

The velocity of the camera $V_c \in \mathbb{R}^{(6\times1)}$ can be divided into the translational velocity $V_{ct} \in \mathbb{R}^{(3\times1)}$ and rotating velocity $V_{cr} \in \mathbb{R}^{(3\times1)}$. Therefore, it can be expressed as:

$$V_c = \begin{bmatrix} V_{ct} \\ V_{cr} \end{bmatrix} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \tag{3.1}$$

Furthermore, for the n^{th} feature ($n = 1, 2, .., 4$), the image Jacobian matrix in (1.4) can be divided into the translational part $J_t(s_n)$ and the rotating part $J_r(s_n)$:

$$J_{img}(s_n) = \begin{bmatrix} J_t(s_n) & J_r(s_n) \end{bmatrix}, \tag{3.2}$$

where,

$$J_t(s_n) = \begin{bmatrix} \frac{f}{Z} & 0 & -\frac{x_n}{Z} \\ 0 & \frac{f}{Z} & -\frac{y_n}{Z} \end{bmatrix} \tag{3.3}$$

and:

$$J_r(s_n) = \begin{bmatrix} -\frac{x_n y_n}{f} & \frac{f^2+x_n^2}{f} & -y_n \\ \frac{-f^2-y_n^2}{f} & \frac{x_n y_n}{f} & x_n \end{bmatrix}, \tag{3.4}$$

where x_n and y_n are the feature coordinates in the image space.

In the design of the switch controller, the movement of the camera during the control task is divided into three different stages [118, 121]. In the first stage, the camera has only pure rotation. In the second stage, the camera has only translational movement. Finally, in the third stage, both camera rotation and translation are used to carry out the fine-tuning.

Considering (1.1), (3.1), (3.3), and (3.4), the feature velocity in the image plane can be expressed as:

In the pure translational stage (first stage):

$$\begin{cases} \dot{x}_n = \frac{f}{Z}V_x - \frac{x_n}{Z}V_z \\ \\ \dot{y}_n = \frac{f}{Z}V_y - \frac{y_n}{Z}V_z \end{cases}. \tag{3.5}$$

In the pure rotating stage (second stage):

$$\begin{cases} \dot{x}_n = -\frac{x_n y_n}{f}\omega_x + \frac{f^2+x_n^2}{f}\omega_y - y_n\omega_z \\ \dot{y}_n = -\frac{f^2+y_n^2}{f}\omega_x + \frac{x_n y_n}{f}\omega_y + x_n\omega_z \end{cases}, \tag{3.6}$$

and in the fine-tuning stage (third stage):

$$\begin{cases} \dot{x}_n = \frac{f}{Z}V_x - \frac{x_n}{Z}V_z - \frac{x_n y_n(t_0)}{f}\omega_x + \frac{f^2+x_n^2}{f}\omega_y - y_n\omega_z \\ \\ \dot{y}_n = \frac{f}{Z}V_y - \frac{y_n}{Z}V_z - \frac{f^2+y_n^2}{f}\omega_x + \frac{x_n y_n}{f}\omega_y + x_n\omega_z \end{cases}. \tag{3.7}$$

To remove the noise in the image processing and feature extraction, a feature state estimator is designed based on the Kalman filter algorithm.

In the formulations below, k denotes the current time instant and $k+1$ the next time instant, while T_s represents the sampling time. The estimated states are denoted by $\hat{}$ notation. Considering four features, the feature state at the current instant (k^{th} sample) is defined as:

$$X(k) = [x_1(k), y_1(k), ...x_4(k), y_4(k), \dot{x}_1(k), \dot{y}_1(k), ..., \dot{x}_4(k), \dot{y}_4(k)]^T, \tag{3.8}$$

or with consideration of (1.3):

$$X(k) = [s(k), \quad \dot{s}(k)], \tag{3.9}$$

where the elements of the vector can be obtained from (3.5), (3.6), or (3.7). Furthermore, the measurement vector represents the vector of the image feature points' coordinates extracted from the images of the camera:

$$M(k) = [x_{m1}(k), y_{m1}(k), ...x_{m4}(k), y_{m4}(k), \dot{x}_{m1}(k), \dot{y}_{m1}(k), ..., \dot{x}_{m4}(k), \dot{y}_{m4}(k)]^T. \tag{3.10}$$

First, the prediction equations are:

$$\begin{aligned} \hat{X}(k|k-1) &= A\hat{X}(k-1|k-1) \\ P(k|k-1) &= AP(k-1|k-1)A^T + Q(k-1), \end{aligned} \tag{3.11}$$

where A is a 16×16 matrix whose diagonal elements equal one, $A_{i,i+8}(i = 1, 2..., 8)$ are equal to sampling time T_s, and the rest of the elements are zero, $P(k|k-1)$ represents the current prediction of the error covariance matrix, which gives a measure of the state estimate accuracy, while $P(k-1|k-1)$ is the previous error covariance matrix, and $Q(k-1)$ represents the process noise covariance computed using the information of the time instant $(k-1)$.

Second, the Kalman filter gain $D(K)$ is:

$$D(k) = P(k|k-1)(P(k|k-1) + R(k-1)^{-1}, \tag{3.12}$$

where $R(k-1)$ is the previous measurement covariance matrix.

Third, the estimation update is given as follows:

$$\begin{aligned} \hat{X}(k|k) &= \hat{X}(k|k-1) + D(k)(M(k) - \hat{X}(k|k-1) \\ P(k|k) &= P(K|k-1) - D(k)P(k|k-1). \end{aligned} \tag{3.13}$$

When the features are out of the FOV of the camera (i.e. $x_{mj}(k) = 0$, $y_{mj}(k) = 0$, $j = 1, 2...4$), the feature reconstruction algorithm is proposed to provide the updated estimation vector under this circumstance. Since the features are out of FOV, the measurement vector will have some elements with zero values. This measurement vector will not lead to a satisfactory performance of switch IBVS. In order to improve the performance, instead of having zero values of the elements of $M(k)$ in (3.10), it is reasonable to assume that the n^{th} feature that goes outside of FOV keeps its velocity at the moment (t_0) of leaving $(\dot{s}_n(t_0))$ during the period of feature loss. Hence, its position (i.e., point coordinates $s_n(t_0) = [x_{mn}(t_0), y_{mn}(t_0)]$) can be generated by integrating the velocity over the time. This means that the elements of $M(k)$ can be represented by this formulation:

$$M(k) = [(K_{ad} \sum_{l=0}^{b} \dot{s}_n(t_0) T_s + s_n(t_0)), \qquad \dot{s}_n(t_0)], \tag{3.14}$$

where $(l = 0, 1, 2, .., b)$ represents the number of time samples during the feature loss period, T_s is the sampling period, and k_{ad} is an adjusting coefficient. Once the feature is visible to the camera again, the actual value of $M(k)$ provided by the camera is used to replace the state estimation (3.14).

3.4 Controller Design

The IBVS controller is designed using the switch scheme. This method can set distinct gain values for the stages of the control law to achieve a fast response system while preserving the system stability.

In order to design the switch controller, the movement of the camera during the control task is divided into three different stages [118, 121]. A criterion is needed for the switch condition between stages. In [121], the norm of feature errors is defined as the switching criterion. In this thesis (as it is described in 2.3), a more intuitive and effective criterion is used. As is shown in Figure 2.1, the switch angle criterion α is introduced as the angle between actual features and the desired ones. As soon as the angle α meets the predefined value, the controller law switches to the next stage.

Based on this criterion, the switching control law is presented as follows:

$$\begin{cases} V_{cs1} = -K_1 J_r^+ e(s), & |\alpha| \geq \alpha_0 \\ V_{cs2} = -K_2 J_t^+ e(s), & \alpha_1 \leq |\alpha| < \alpha_0 \\ V_{cs3} = -K_3 J_{img}^+ e(s), & otherwise \end{cases} , \tag{3.15}$$

where V_{csi} $(i = 1, 2, 3)$ is the velocity of the camera in the i^{th} stage, K_i is the symmetric positive definite gain matrix at each stage, and α_0 and α_1 are two predefined thresholds for the control law to switch to the next stage. The block diagram of the proposed algorithm is shown in Figure 3.2. Furthermore, the flowchart of the whole process of feature reconstruction and control is illustrated in Figure 3.3.

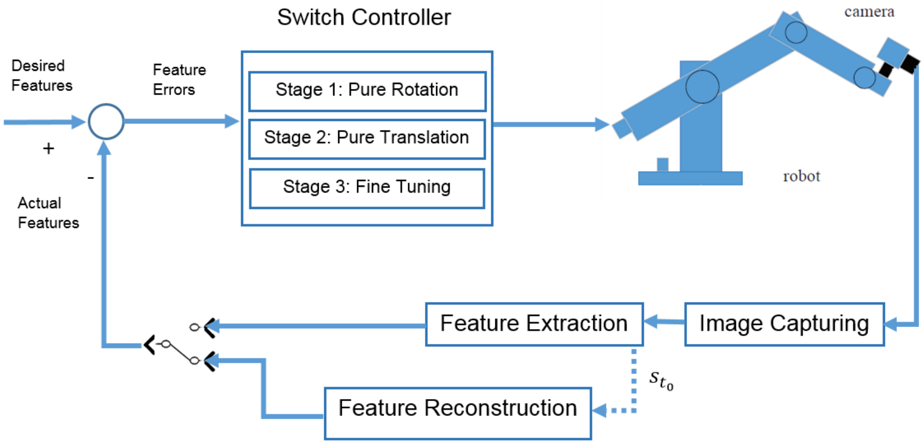


Figure 3.2: Block diagram of the proposed enhanced switch image-based visual servoing (ESIBVS) controller.

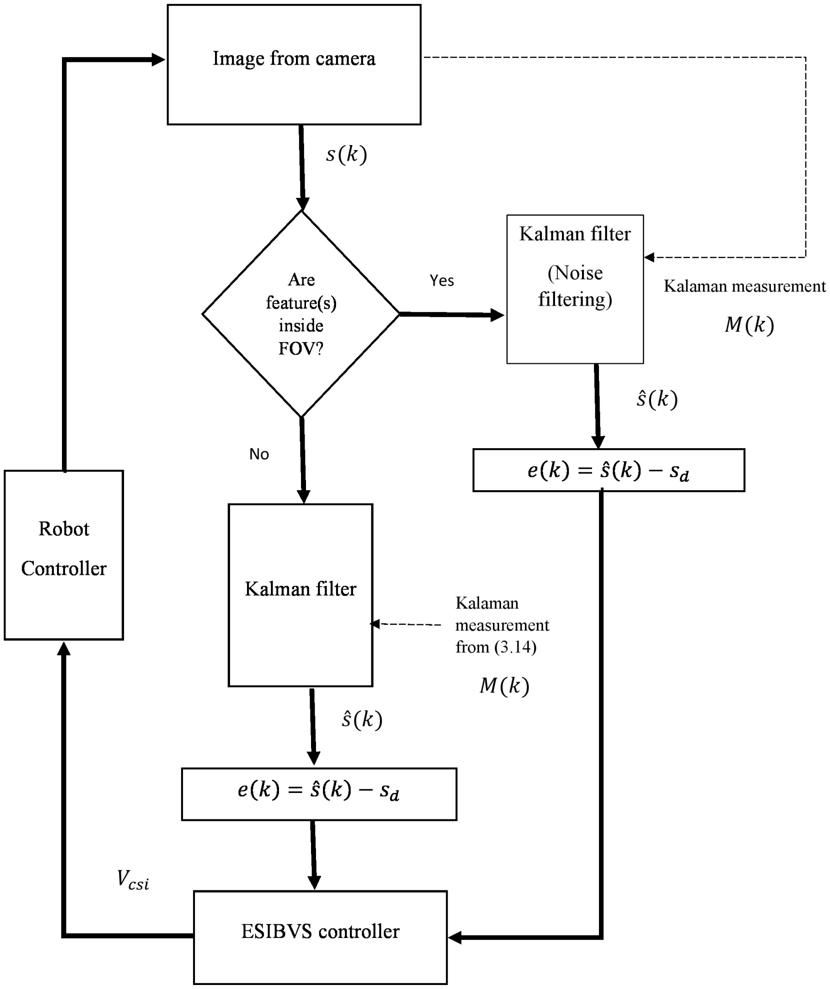


Figure 3.3: Flowchart of the Kalman filter feature reconstruction and control algorithm.

It is expected that in comparison with switch IBVS, the proposed method would ensure the smooth transition of the visual servoing task in the case of the feature loss and provide a better convergence performance.

3.5 Simulation Results

To evaluate the performance of the proposed method, simulation tests are carried out by using MATLAB/SIMULINK software with the Vision and Robotic Toolbox. A 6-DOF DENSO robot with a camera installed in eye-in-hand configuration is simulated. The coordinates of the initial and desired features in the image space are given in Table 3.1. The camera parameters are as shown in Table 2.1.

Table 3.1: Test 1: simulation. Initial (I) and Desired (D) feature point positions in pixels.

		Point 1	**Point 2**	**Point 3**	**Point 4**
		(x, y)	(x, y)	(x, y)	(x, y)
Test 1	I	376 757	202 621	20 814	218 969
	D	612 312	612 512	812 512	812 312

The task is to guide the end-effector to match the actual features with the desired ones in the camera image space. To simulate the condition where the features go outside FOV of the camera in real applications, the FOV of the camera is defined as the limited area shown in Figure 3.4a,b. When the features are in the defined FOV, they had actual position coordinates, and when they went outside FOV, the position coordinates of the features are set to zero. In this case, the proposed feature reconstruction algorithm is activated, and an estimate of the feature positions is generated. The norm of feature errors (NFE) is defined as below,

$$NFE = \sum_{n=1}^{4} \sqrt{(x_n - x_{dn})^2 + (y_n - y_{dn})^2}, \tag{3.16}$$

where x_n and y_n are the n^{th} feature coordinates and x_{dn} and y_{dn} are the n^{th} desired feature coordinates in the image plane.

In the simulation test, we set the initial feature coordinates and the desired ones in a way that the image features are out of FOV. Figures 3.4 and 3.5 demonstrate the performance comparison of the two methods. The paths of image features in the image space are given in Figure 3.4a,b. Figure 3.5a,b shows how the feature errors change with time in the proposed ESIBVS and switch method. Figure 3.5c,d demonstrates the norm of the feature errors' change with time in both methods. As shown in the figures, ESIBVS is able to reduce the norm of the errors to the preset threshold, while in the switch method, the norm of the errors did not converge. The summary of the simulation test is shown in Table 3.2. The results demonstrate how the proposed method is able to handle the situation in which the features went outside of the camera's FOV and completed the task successfully, while the switch method is unable to do so.

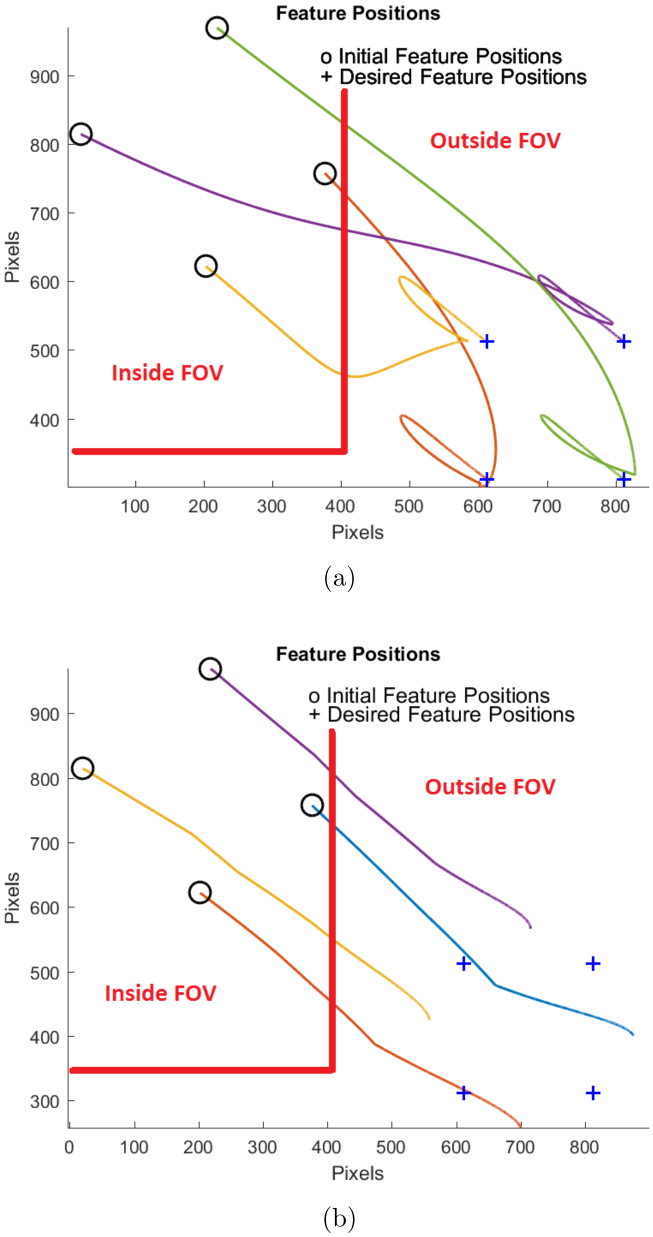
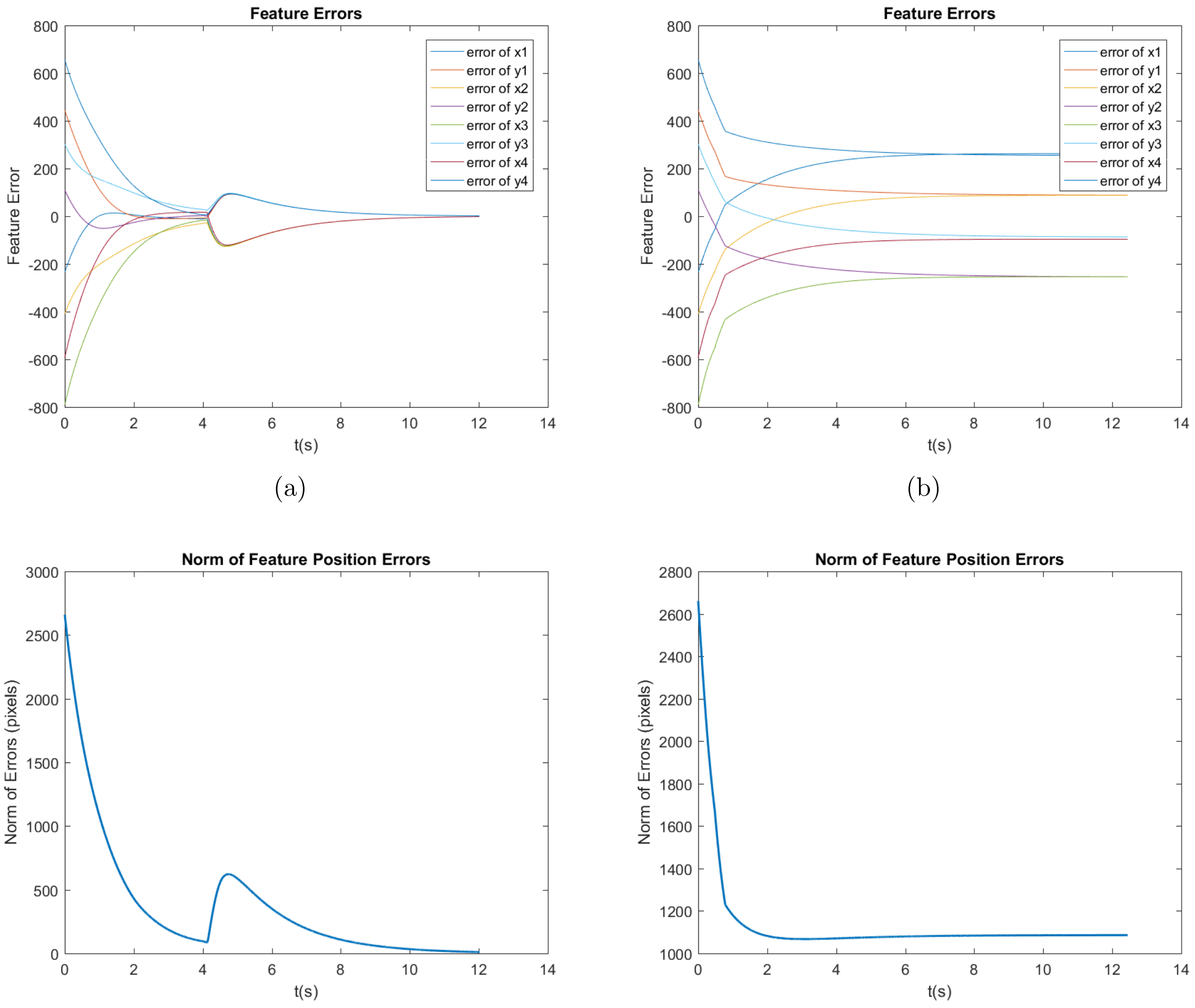


Figure 3.4: Test 1: simulation. Image space feature trajectory comparison of enhanced switch IBVS and switch IBVS. (**a**) Image space feature trajectory in enhanced switch IBVS; (**b**) Image space feature trajectory in switch IBVS.



(a)

(b)

(c)

(d)

Figure 3.5: Test 1: simulation. Performance comparison of enhanced switch IBVS vs. switch IBVS. (**a**) Feature errors in enhanced switch IBVS; (**b**) Feature errors in switch IBVS; (**c**) Norm of feature errors in enhanced switch IBVS; (**d**) Norm of feature errors in switch IBVS.

Table 3.2: Test 1: Comparison of simulation results between ESIBVS and switch IBVS.

	Time of Convergence (s)		**Final Norm of Feature Errors (Pixel)**	
	ESIBVS	**Switch IBVS**	**ESIBVS**	**Switch IBVS**
Test 1	12	Does not converge	1.5	Does not converge

3.6 Experimental Results

In this section, to further verify the effectiveness of the proposed method, some experiments are carried out on the set-up described in 2.5 and the results are presented. Four feature points are used in the control task. The goal is to control the end-effector so that the actual features are matched the desired ones.

To evaluate the efficiency of ESIBVS, its performance is compared to that of the switch IBVS method. In all the tests, the threshold value of NFE is set to 0.005 (equivalent to four pixels). When NFE is reached this value, the robot is stopped, and the servoing task is fulfilled. The initial angle α between the actual and desired features (Figure 2.1) is 50°. K_1, K_2, and K_3 in (3.15) are set to 1, 0.4, and 0.3, respectively.

Test 2: In this test, the initial and desired features are set such that they go outside of the FOV of the camera during the test. The initial and desired feature coordinates in the test are given in Table 3.3. Figure 3.6 demonstrates the movement of actual features during the test of ESIBVS. It illustrates how the features go outside of FOV, then are reconstructed, go back to FOV, and finally are matched the desired features.

Table 3.3: Test 2: experiment, Initial (I) and desired (D) feature point positions in pixels.

		Point 1	**Point 2**	**Point 3**	**Point 4**
		(x, y)	(x, y)	(x, y)	(x, y)
Test 2	I	251 132	278 102	306 127	279 157
	D	232 82	272 82	272 119	233 119

Figures 3.7–3.9 show the comparison results between ESIBVS and switch IBVS. Figure 3.7 shows the paths of features in the image space from the initial positions to the desired ones, as well as the camera trajectory in Cartesian space. In the proposed method, the actual and desired features are matched, while in switch IBVS, the actual features do not converge to the desired ones. Figure 3.8 demonstrates the robot joint angles in ESIBVS and switch IBVS. Figure 3.9 shows the comparison regarding the feature errors. The feature errors and the norm of feature errors in the proposed method successfully converge to the desired values (Figure 3.9a,c), while in the switch IBVS, the task could not be completed, and thus, the feature errors do not converge (Figure 3.9b,d).

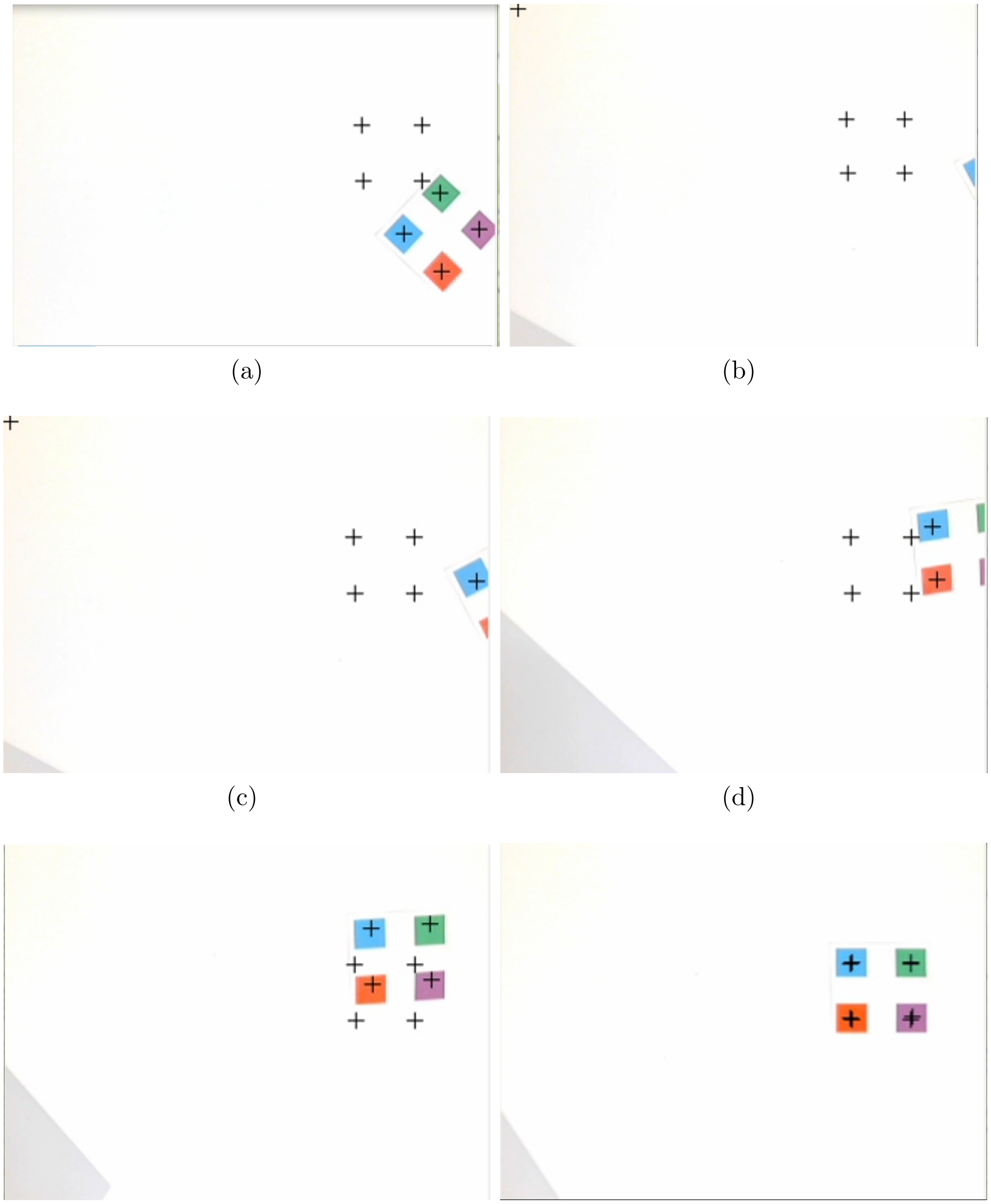


Figure 3.6: Test 2: Snap shots of the camera image during the enhanced switch IBVS test: (**a**) Desired and actual feature positions at the start. (**b**) Actual features are out of FOV. (**c**–**e**) Features are reconstructed and returned to FOV. (**f**) Final match of the desired and actual features.

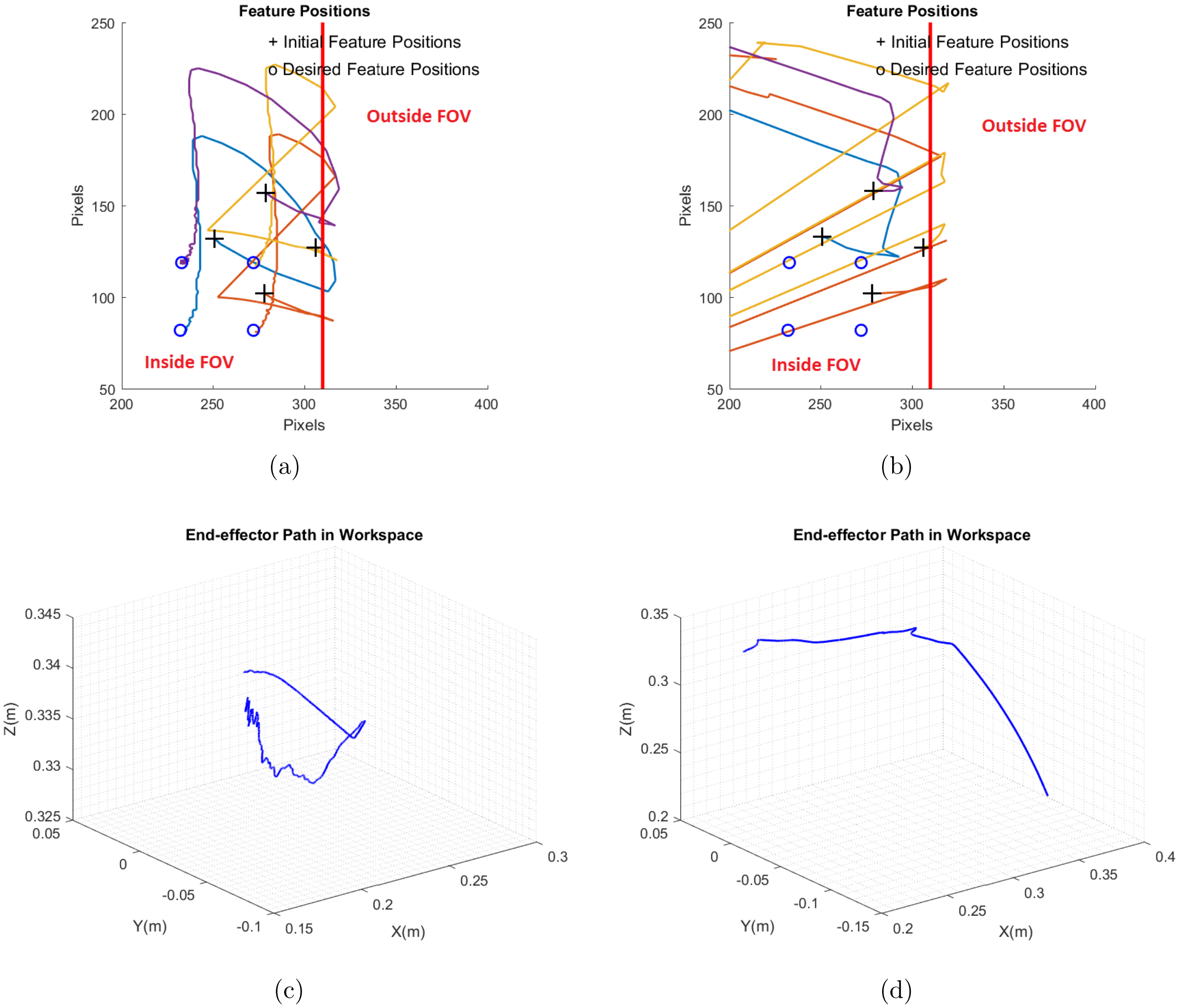
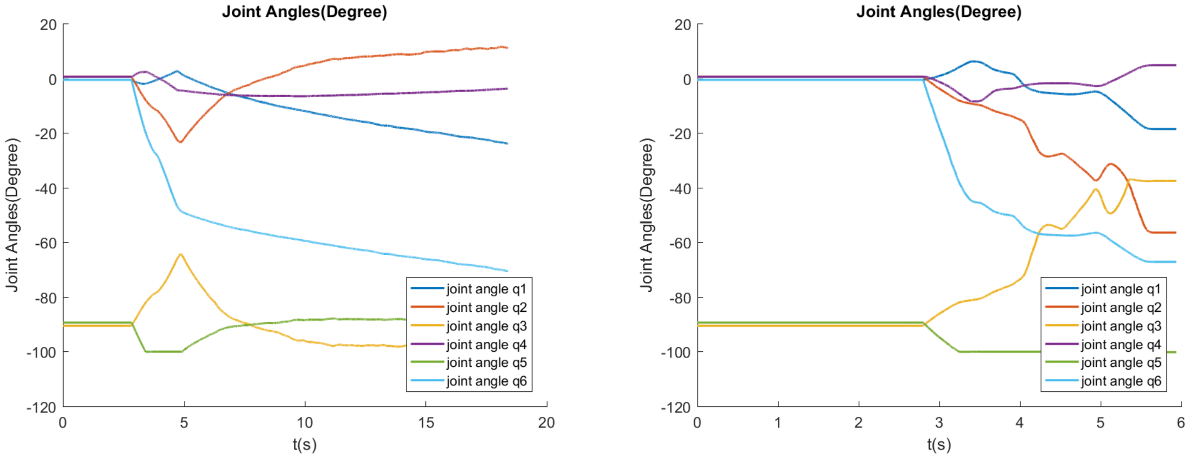


Figure 3.7: Test 2: experiment: Image space feature trajectory and 3D camera trajectory in enhanced switch IBVS and switch IBVS. (**a**) Image space feature trajectory in enhanced switch IBVS; (**b**) Image space feature trajectory in switch IBVS; (**c**) Camera 3D trajectory in enhanced switch IBVS; (**d**) Camera 3D trajectory in switch IBVS.



(a)

(b)

Figure 3.8: Test 2: experiment. Robot joint angles in enhanced switch IBVS and switch IBVS. (**a**) Joint angles (degree) in enhanced switch IBVS; (**b**) Joint angles (degree) in switch IBVS.

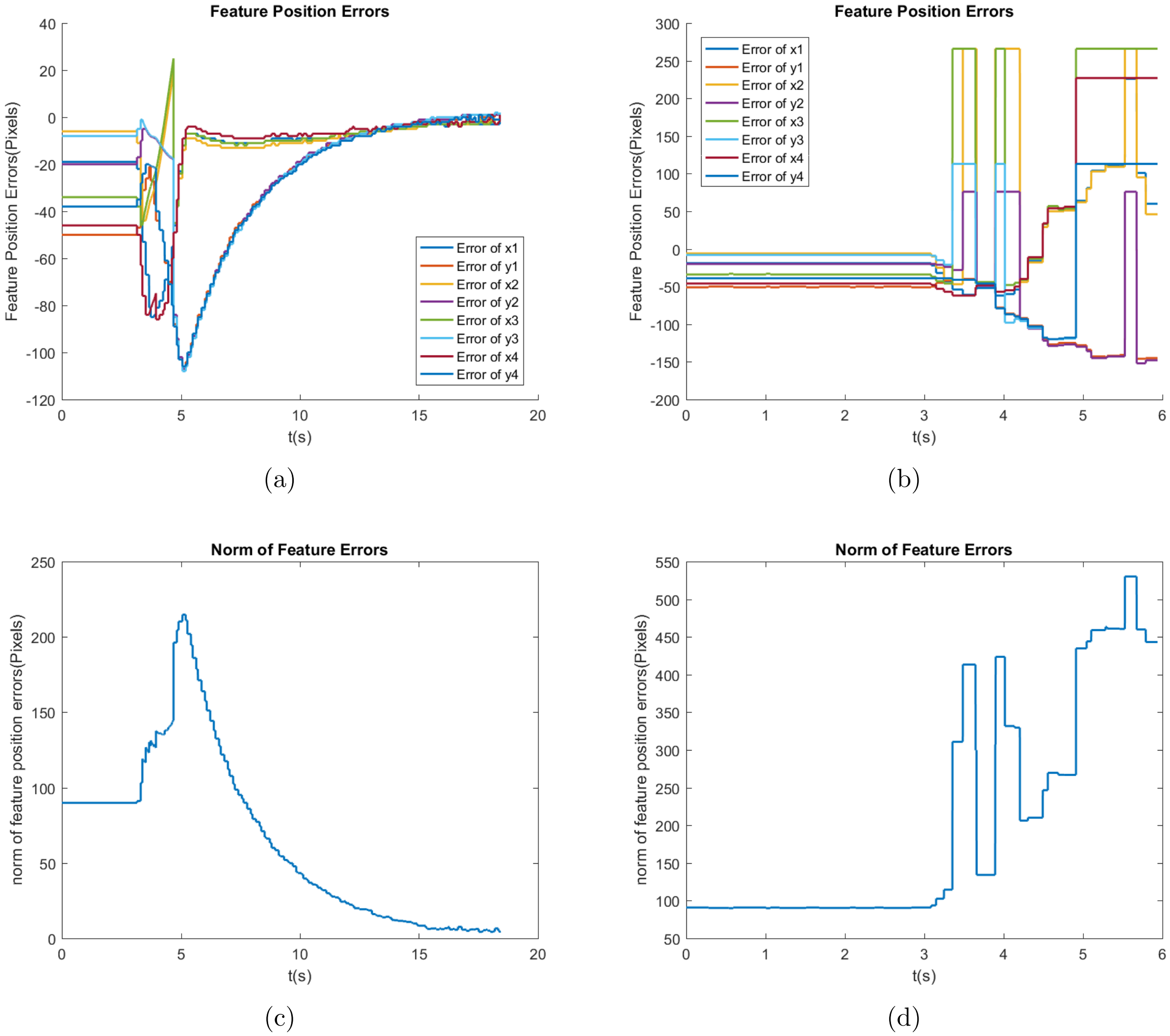


Figure 3.9: Test 2: experiment. Comparison of the feature errors and the norm of feature errors in enhanced switch IBVS and switch IBVS. (**a**) Feature errors in enhanced switch IBVS; (**b**) Feature errors in switch IBVS; (**c**) Norm of feature errors in enhanced switch IBVS; (**d**) Norm of feature errors in switch IBVS.

In order to further validate the performance of ESIBVS regarding the repeatability, the same test is repeated in 10 trials. The time of convergence and the final norms of feature error are shown in Table 3.4. The variations of feature error norms with time in 10 trials of ESIBVS are illustrated in Figure 3.10. As shown in the results, ESIBVS is able to overcome the feature loss and complete the task in each trial, while Switch IBVS is stuck in a point and did not converge.

Table 3.4: Test 2: experiment. Repeatability comparison results.

	Time of Convergence (s)		**Final Norm of Feature Errors (Pixel)**	
	ESIBVS	**Switch IBVS**	**ESIBVS**	**Switch IBVS**
Trial 1	19.95	Does not converge	3.4	Does not converge
Trial 2	18.99	Does not converge	3.1	Does not converge
Trial 3	17.75	Does not converge	2.8	Does not converge
Trial 4	17.94	Does not converge	3.7	Does not converge
Trial 5	19.37	Does not converge	3.6	Does not converge
Trial 6	18.29	Does not converge	2	Does not converge
Trial 7	20.34	Does not converge	3.1	Does not converge
Trial 8	19.03	Does not converge	3.4	Does not converge
Trial 9	18.77	Does not converge	2.4	Does not converge
Trial 10	18.74	Does not converge	3.4	Does not converge

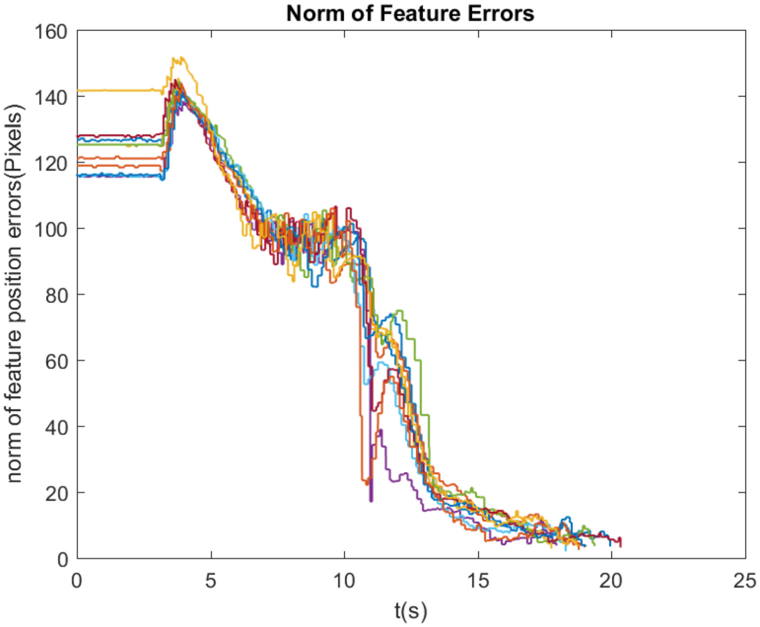


Figure 3.10: Test 2: experiment. The time variations of feature error norms in 10 trials of ESIBVS.

Test 3: In this test, the performance of ESIBVS is compared with that of switch IBVS in the situation where the features do not leave the FOV of the camera. The initial and desired features are set in a way such that the features do not go outside of FOV (Table 3.5).

Table 3.5: Test 3: Experiment. Initial (I) and Desired (D) feature point positions in pixels.

		Point 1	**Point 2**	**Point 3**	**Point 4**
		(x, y)	(x, y)	(x, y)	(x, y)
Test 3	I	108 127	130 97	136 148	158 118
	D	232 82	272 82	272 119	233 119

Similar to the previous tests, ESIBVS and switch IBVS are compared, and the results are shown in Figures 3.11–3.13 and Table 3.6. As shown in the figures, ESIBVS has a 38% shorter convergence time than switch IBVS does, which is owed to the superior noise-filtering ability of the designed Kalman filter.

The experimental results shows the efficiency of ESIBVS in dealing with feature loss while keeping the superior performance of the switch IBVS over traditional IBVS. As it is shown in [11, 118], the switch method is proven to have a better performance in its response time and its tracking performance, making it more feasible for industrial applications in comparison with the conventional IBVS. However, it suffers the drawback of weakness in dealing with feature loss. The proposed ESIBVS solves this problem and make switch IBVS more robust by using the Kalman filter to reconstruct the lost features.

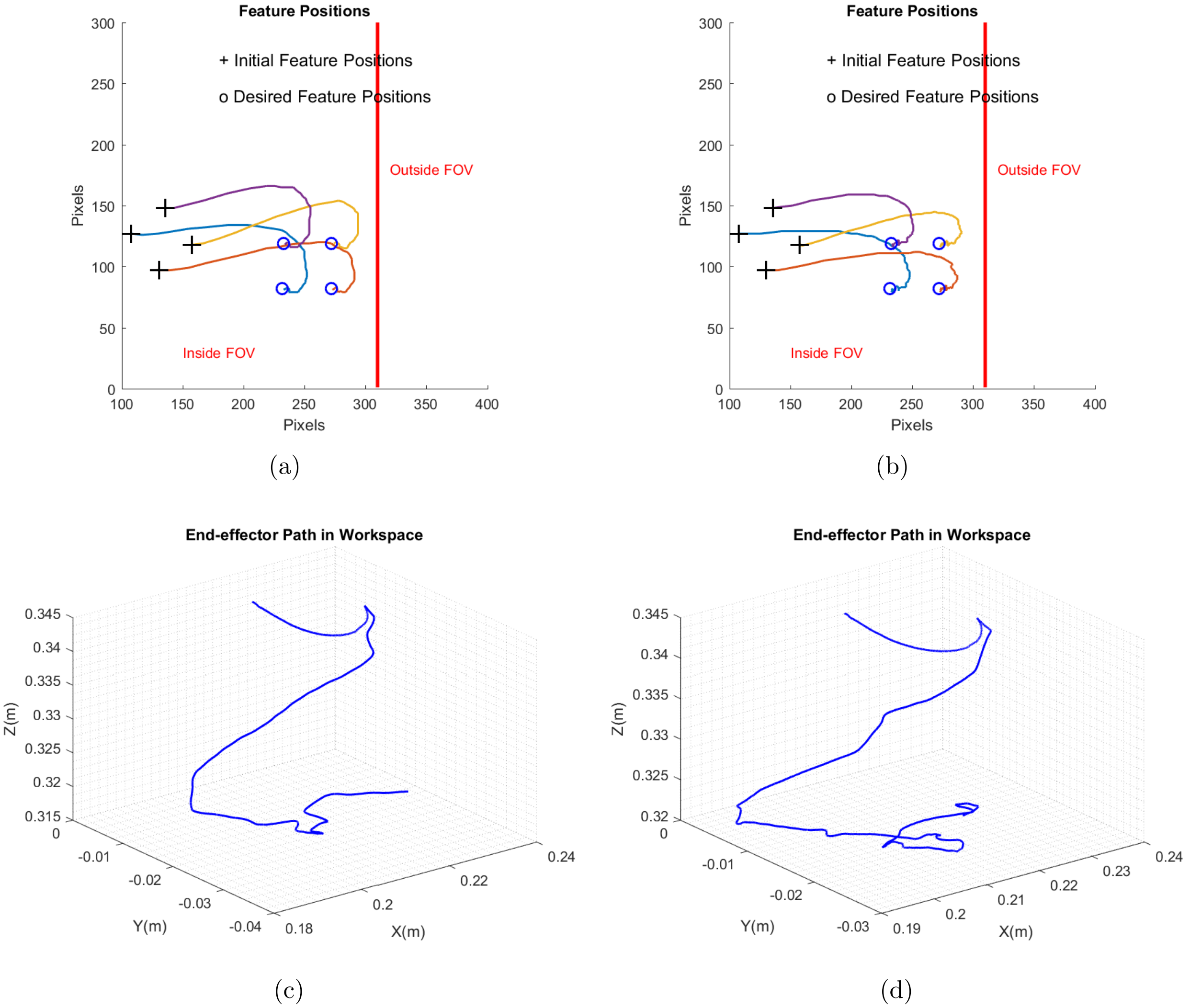
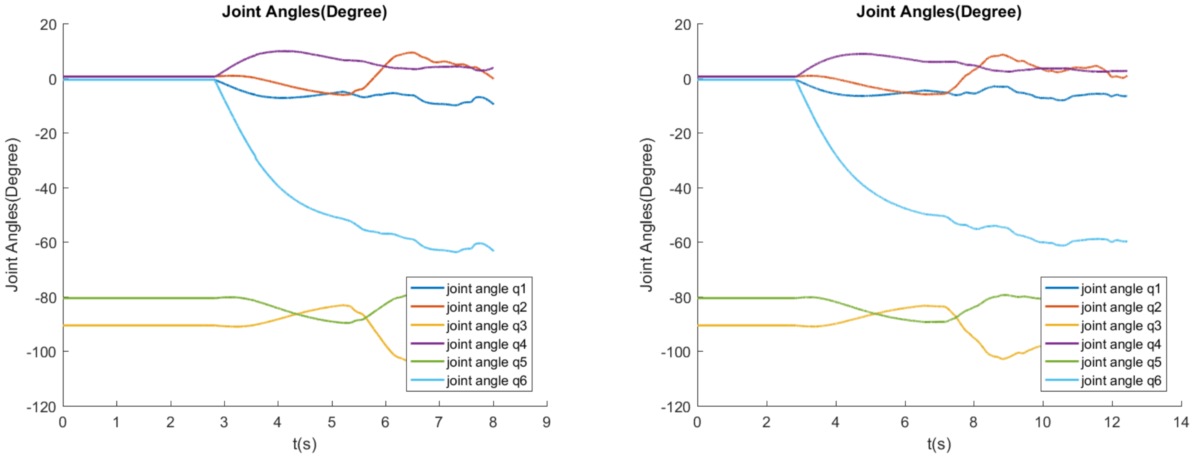


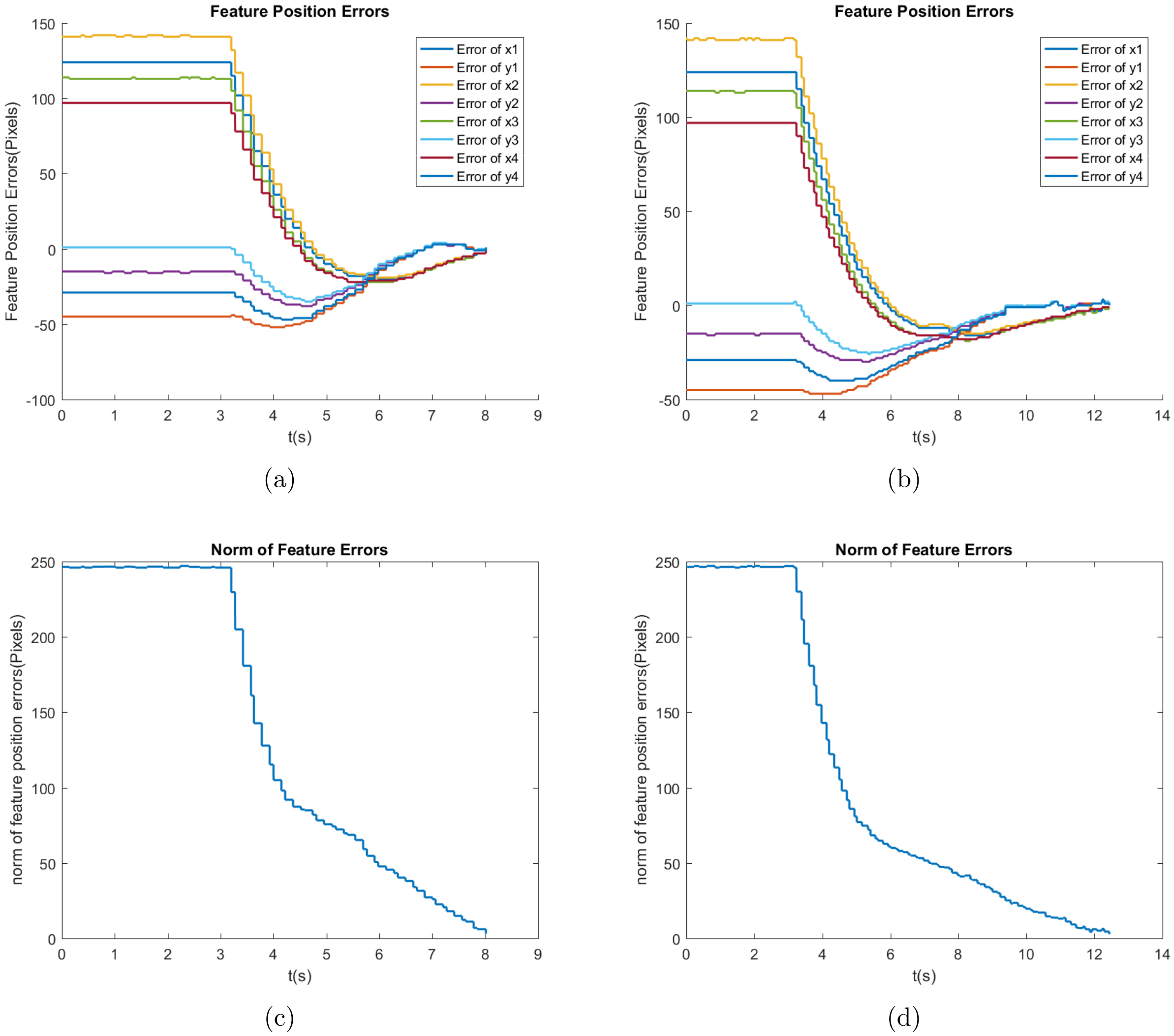
Figure 3.11: Test 3: experiment. Image space feature trajectory and 3D camera trajectory in enhanced switch IBVS and switch IBVS. (**a**) Image space feature trajectory in enhanced switch IBVS; (**b**) Image space feature trajectory in switch IBVS; (**c**) Camera 3D trajectory in enhanced switch IBVS; (**d**) Camera 3D trajectory in switch IBVS.



(a)

(b)

Figure 3.12: Test 3: experiment. Robot joint angles in enhanced switch IBVS and switch IBVS. (**a**) Feature errors in enhanced switch IBVS; (**b**) Feature errors in switch IBVS.



(a)

(b)

(c)

(d)

Figure 3.13: Test 3: experiment. Comparison of feature errors and the norm of feature errors in enhanced switch IBVS and switch IBVS. (**a**) Feature errors in enhanced switch IBVS; (**b**) Feature errors in switch IBVS; (**c**) Norm of feature errors in enhanced switch IBVS; (**d**) Norm of feature errors in switch IBVS.

Table 3.6: Test 3: Comparison of experimental resuts between ESIBVS and Switch IBVS.

	Time of Convergence (s)		**Final Norm of Feature Errors (Pixel)**	
	ESIBVS	**Switch IBVS**	**ESIBVS**	**Switch IBVS**
Test 3	8	12.5	3.4	3.6

3.7 Summary

In this chapter, an enhanced switch IBVS for a 6-DOF industrial robot is proposed. An image feature reconstruction algorithm based on the Kalman filter is proposed to handle feature loss during the process of IBVS. The combination of a three-stage switch controller and feature reconstruction algorithm improve the system response speed and tracking performance of IBVS and simultaneously overcame the problem of feature loss during the task. The proposed method is simulated and then tested on a 6-DOF robotic system with the camera installed in an eye-in-hand configuration. Both simulation and experimental results verify the efficiency of the method.

Chapter 4

Enhanced IBVS Controller Using Hybrid PD-SMC Method

4.1 Introduction

In this chapter, considering the respective advantages of PD and SMC mentioned in 1.3.3, a new hybrid way that combines PD control with SMC in IBVS is proposed to generate the better velocity profile to control the 6-DOF robotic manipulator. The proposed controller can deal with the uncertainties in depth and Lyapunov direct method is used to prove the stability of the proposed controller. The main feature of the proposed hybrid PD-SMC is its less computation burden, compared to the adaptive or predicted control approaches. In addition, it not only can achieve better convergence performance with guaranteed stability, but also owns stronger robustness against uncertainty and disturbance, compared to either IBVS PD or SMC system [58]. The proposed IBVS scheme has been extensively tested on a 6-DOF manipulator.

4.2 System Description

The control problem can be expressed by obtaining the relation between the derivative of the image features and the camera spatial velocity in IBVS [54, 57].

Considering a 6-DOF manipulator, at least three feature points are necessary to avoid the Jacobian matrix singularities and the multiple global minima [57, 58]. For this reason, we use four feature points to control 6-DOF in the image space. Thus, we may rename the

Jacobian matrix J_{img} for four features as L_{v4} and the equation (1.1) in Chapter 1 can be expressed as follows:

$$\dot{e} = \dot{s} = L_{v4} V_c \tag{4.1}$$

where L_{v4} is the Jacobian matrix for four features and e is representing the feature errors.

$$L_{v4} = \begin{bmatrix} L_{v|\ s=s_1} \\ L_{v|\ s=s_2} \\ L_{v|\ s=s_3} \\ L_{v|\ s=s_4} \end{bmatrix}, \tag{4.2}$$

and $s = s_1, \cdots, s_4$ are the image feature points. Since the image Jacobian matrix largely depends on the depth Z and camera intrinsic parameters such as focal length f, there exist some uncertainties in these parameters. In this thesis, we focus on dealing with the uncertainties on the depth. The range of the depth of the visual servoing system is assumed as $Z_{min} \leq Z \leq Z_{max}$. The estimated Jacobian matrix ($\hat{L}_{v4}$) is used in the viusal servoing control design.

4.3 Visual Servoing Controller Design

The most basic design approach of a visual servoing controller is using proportional control to generate the control signal. This controller is also applied to the conventional IBVS, which has the following form,

$$V_c = -K \hat{L}_{v4}^{+} \dot{e}(t), \tag{4.3}$$

where $\hat{L}_{v4}^{+}$ is the pseudo inverse of the estimated Jacobian matrix, K is a positive definite matrix.

The proportional control is an efficient and easily tuned control method. However, this method sometimes cannot achieve the desired dynamic response by only tuning the proportional gain. In this thesis, a properly tuned PD control is used to replace the proportional control, which can improve the control performance with faster control convergence speed and smaller feature errors. Meanwhile, in order to improve the system stability, the sliding mode control is also adopted to compensate the uncertainties of the system. This is an enhanced approach which combines PD control with SMC in IBVS, so-called the hybrid PD-SMC method.

Define the sliding surface S, which will converge to 0 when the image feature errors go forward and stay on it all the time [94, 127].

$$S = e = s(t) - s_d(t), \tag{4.4}$$

where $s = [s_i, i = 1, 2, ...4]$ is the image plane feature, $s_d = [s_{id}, i = 1, 2, ...4]$ is the desired value of the feature, e is the vector of four feature errors between the obtained image features and desired image features. The basic visual servoing controller of IBVS is designed based on the above proportional controller [54], and it is described as the following first order equation,

$$\dot{e} + K_p e = 0. \tag{4.5}$$

To guarantee the tracking performance of the visual servoing system subjected to the uncertain depth, a robust controller is designed to handle such uncertainties. Adding the sliding mode control [55, 128], the above equation is written as

$$V_c = \hat{L}_{v4}^{+}(-K_p e(t) - K_s \mathrm{sgn}(S)), \tag{4.6}$$

where K_s is a positive definite matrix, $\mathrm{sgn}(\cdot)$ is the signum function.

Applying PD control to the visual servoing system [100], the modified control law should be considered as

$$V_c = \hat{L}_{v4}^{+}(-K_d \dot{e}(t) - K_p e(t) - K_s \mathrm{sgn}(S)). \tag{4.7}$$

In general, the above control scheme leads to chattering phenomenon. In order to smooth the chattering, a saturation function is used to replace the sign function. Hence, the control law is described as

$$V_c = \hat{L}_{v4}^{+}(-K_d \dot{e}(t) - K_p e(t) - K_s \mathrm{sat}(S)), \tag{4.8}$$

where sat$(\cdot)$ is the saturation function, which is defined as follows.

$$\text{sat}(S) = \begin{cases} S & \text{if } |S| \leqslant 1 \\ \text{sgn}(S) & \text{otherwise} \end{cases}. \tag{4.9}$$

This control law is a hybrid method which combines PD control and SMC together. The structure of the designed controller is shown in Figure 4.1. Since SMC is well known for its robustness [24, 98–100], by combining the PD and SMC together, it is expected that this hybrid controller will achieve better robustness, faster convergence rate and higher accuracy. This will be demonstrated in both Simulation and Experiment sections.

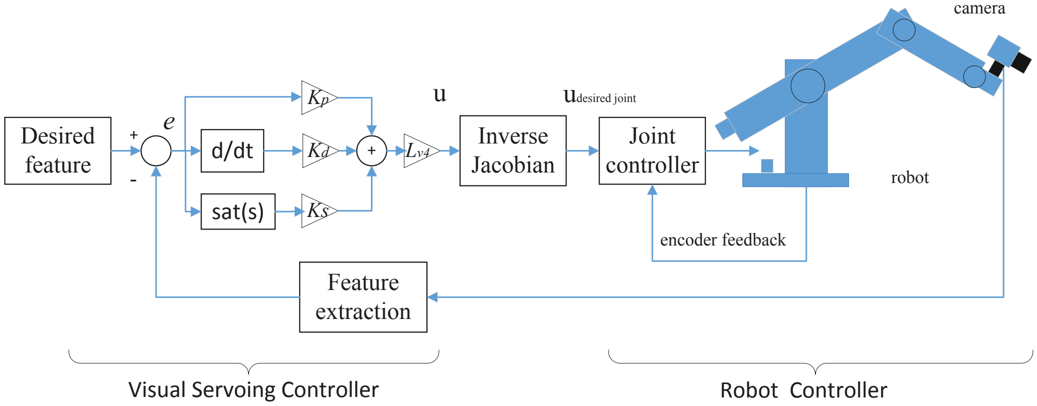


Figure 4.1: Visual servoing system with eye-in-hand configuration

4.4 Stability Analysis

The stability analysis of the proposed controller is based on Lyapunov direct method[94]. Due to the uncertainties in depth, the estimated Jacobian matrix is subjected to the following constraints.

$$(I + \Delta_{min}) \leq L_{v4}\hat{L}_{v4}^{+} \leq (I + \Delta_{max}), \tag{4.10}$$

where Δ_{min} is a matrix of the uncertainties associated with lower bounds of estimated depth Z_{min} and Δ_{max} is a matrix of the uncertainties associated with the upper bounds of the estimated depth Z_{max}. The relationships between Δ_{min} , Δ_{max} with the bounds of depth are illustrated in the Appendix.

Definition 4.4.1 *Assume that the feature points are in the field view, under the control of the controller, the image error of the feature point P converges to zero, i.e.,*

$$\lim_{t\to+\infty} e(t) = 0. \tag{4.11}$$

Proof: A Lyapunov function is constructed as

$$V = \frac{1}{2}S^T S. \tag{4.12}$$

The time derivative of the above Lyapunov function is obtained as

$$\dot{V} = S^T \dot{S}. \tag{4.13}$$

By substituting (4.8) into (4.1), the following system error dynamic equation is obtained

$$\dot{S}(t) = \dot{e}(t) = L_{v4}\hat{L}_{v4}^{+}(-K_d\dot{e}(t) - K_p e(t) - K_s \mathrm{sat}(S)) - s_d(t). \tag{4.14}$$

Moving the term associated with $\dot{e}$ to the left of the equation yields,

$$\dot{e} = (I + K_d\hat{L}_{v4}L_{v4}^{+})^{-1}L_{v4}\hat{L}_{v4}^{+}(-K_p e(t) - K_s\mathrm{sat}(S)) - (I + K_d\hat{L}_{v4}L_{v4}^{+})^{-1}s_d(t). \tag{4.15}$$

Then the time derivative of Lyapunov function becomes:

$$\dot{V} = (I + K_d L_{v4}\hat{L}_{v4}^{+})^{-1}L_{v4}\hat{L}_{v4}^{+}(-e^T K_p e - K_s|S|) - (I + K_d L_{v4}\hat{L}_{v4}^{+})^{-1}S^T s_d(t). \tag{4.16}$$

It is noted that the rank of $L_{v4}L_{v4}^{+}$ is 6, and $L_{v4}L_{v4}^{+}$ has two null vectors that satisfy $L_{v4}L_{v4}^{+}x = 0$. Matrix $L_{v4}L_{v4}^{+}$ has the maximum rank of 6, and $L_{v4}L_{v4}^{+}$ has two null vectors that satisfy $\{L_{v4}L_{v4}^{+}x = 0, x \in \mathbb{R}^8, x \neq 0\}$. Assuming that x does not belong to the null space of $L_{v4}L_{v4}^{+}$ [57, 58], we have

$$L_{v4}L_{v4}^{+} > 0. \tag{4.17}$$

If K_d is chosen as a positive definite matrix, one has

$$K_d > 0. \tag{4.18}$$

In this way, one can ensure

$$(I + K_d L_{v4} \hat{L}_{v4}^{+})^{-1} L_{v4} \hat{L}_{v4}^{+} > 0. \tag{4.19}$$

Also, K_s is chosen as

$$K_s > \lambda_{max}(I + \Delta_{max})/\lambda_{min}(I + K_d \Delta_{min})^{-1} s_d(t) + \eta, \tag{4.20}$$

where η is a diagonal positive definite matrix whose elements determine the decay rate of $V(t)$ to zero.

Then the time derivative of Lyapunov function becomes:

$$\dot{V} < (I + K_d L_{v4} \hat{L}_{v4}^{+})^{-1} L_{v4} \hat{L}_{v4}^{+} (-e^T K_p e - \eta |S|) < \quad 0. \tag{4.21}$$

According to Barbalat's lemma, we have $\dot{V} = 0$ when $t \to \infty$. And the image features error $e(t) \to 0$ when $t \to \infty$. Hence, the stability of visual servoing system using the hybrid PD-SMC controller is ensured.

4.5 Simulation results

In this section, computer simulations have been conducted on a Puma 560 robot model by using MATLAB Robotics Toolbox and Machine Vision Toolbox [54].

4.5.1 Simulation analysis

A 6-DOF Puma 560 arm is chosen as the manipulator and the camera is mounted on the end effector which assumes no transformation between the end effector and the camera [12]. The camera characteristics are shown in Table 4.1. The maximum linear velocity of Puma 560 is 0.5 (m/s) according to the robot user manual [129, 130].

Table 4.1: Camera parameters used in simulations

Parameter	**Value**
Focal length	0.008 (m)
Principal point	(512, 512)
Camera resolution	1024×1024

To analyze and compare the performance of hybrid PD-SMC IBVS with the conventional IBVS, four simulation tests is conducted, including pure translation (Test 1) and pure rotation (Test 2) of features, hybrid translation and rotation test (Test 3), and disturbance rejection test (Test 4). In these tests, the depth range is $1.6 < Z < 2.2$ meters. The diagnol elements of two parameter matrices K_d, K_s are 1, 3 respectively.

Four feature points are used in visual servoing control. The initial and desired positions of the image features are given in Table 4.2.

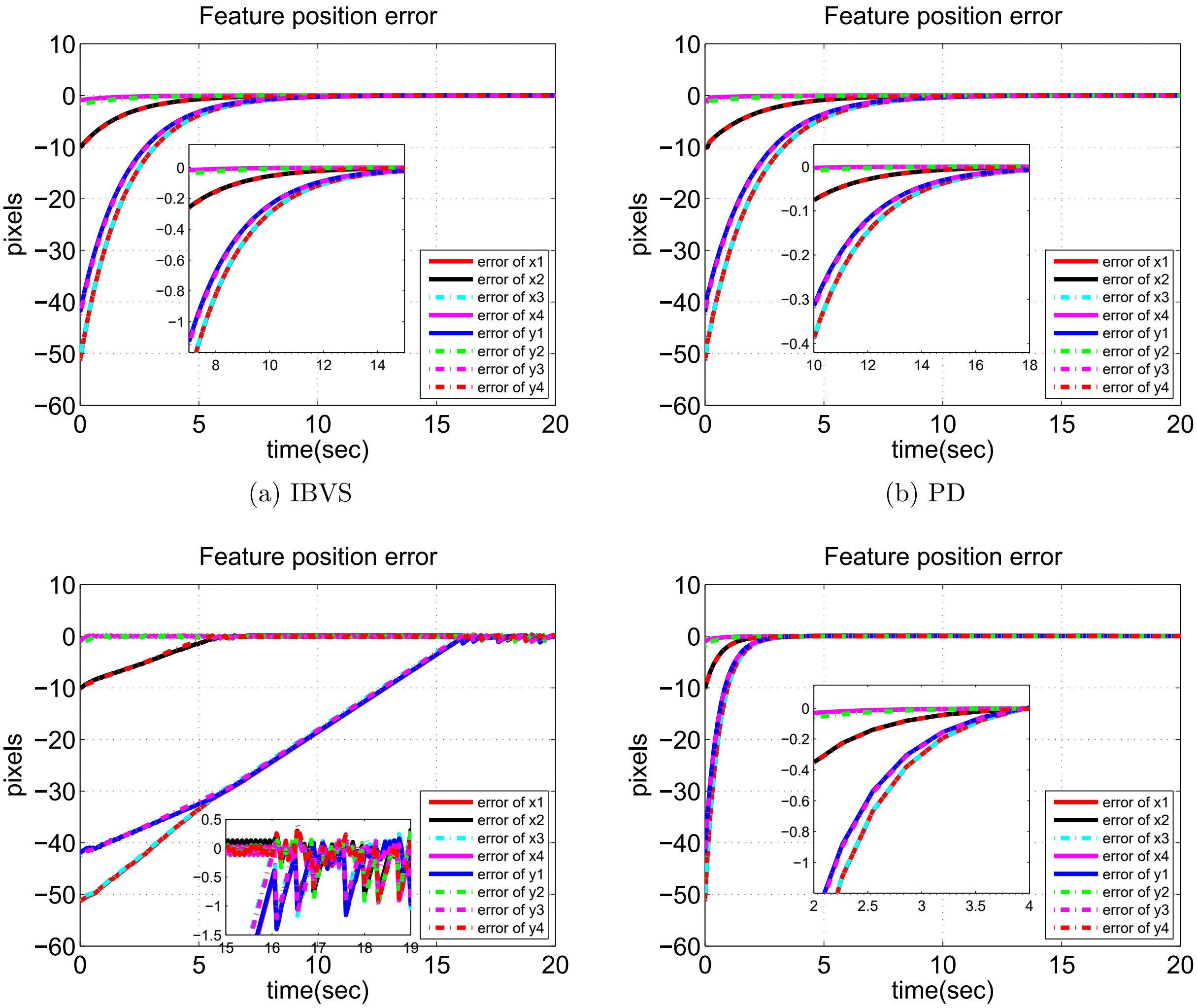
Table 4.2: Initial and desired positions

		Point 1	**Point 2**	**Point 3**	**Point 4**
		$(x_1 \ \ y_1)$	$(x_2 \ \ y_2)$	$(x_3 \ \ y_3)$	$(x_4 \ \ y_4)$
Tests 1&4	Initial	(360 401)	(361 611)	(570 610)	(573 402)
	Desired	(412 412)	(412 612)	(612 612)	(612 412)
Test 2	Initial	(360 401)	(361 611)	(570 610)	(573 402)
	Desired	(362 506)	(466 612)	(572 506)	(466 403)
Test 3	Initial	(389 382)	(350 587)	(556 625)	(594 420)
	Desired	(390 390)	(430 590)	(630 550)	(590 350)

4.5.2 Simulation results by MATLAB

In Test 1, the manipulator moves from initial position to desired position and the trajectory is a pure translational motion. Fig. 4.2 shows the feature position error of proportional (P) control, PD control, SMC control (sign function is used) and hybrid PD-SMC control respectively. Fig. 4.3 shows the joint velocity variations of P control, PD control , SMC control and hybrid PD-SMC control respectively. It can be observed that the system of hybrid PD-SMC owns the fastest convergence rate compared to P control or PD control. The SMC control system utilizing sign function exhibits the chattering phenomenon, so the hybrid PD-SMC system owns better performance. Therefore, only proportional (P) control and hybrid PD-SMC systems are discussed in the following tests. Fig. 4.4 demonstrates the feature trajectories of four feature points in image space of PD-SMC method.

In Test 2, a pure rotational movement of the manipulator is completed. Feature position error variation and Cartesian velocity of the conventional IBVS and hybrid PD-SMC method are compared during a spin, and the results are shown in Fig. 4.5 and Fig. 4.6. It is observed that the results of the hybrid PD-SMC method are better than those of IBVS. Fig. 4.7 shows the initial position, and desired position and rotational trajectories of the feature points.



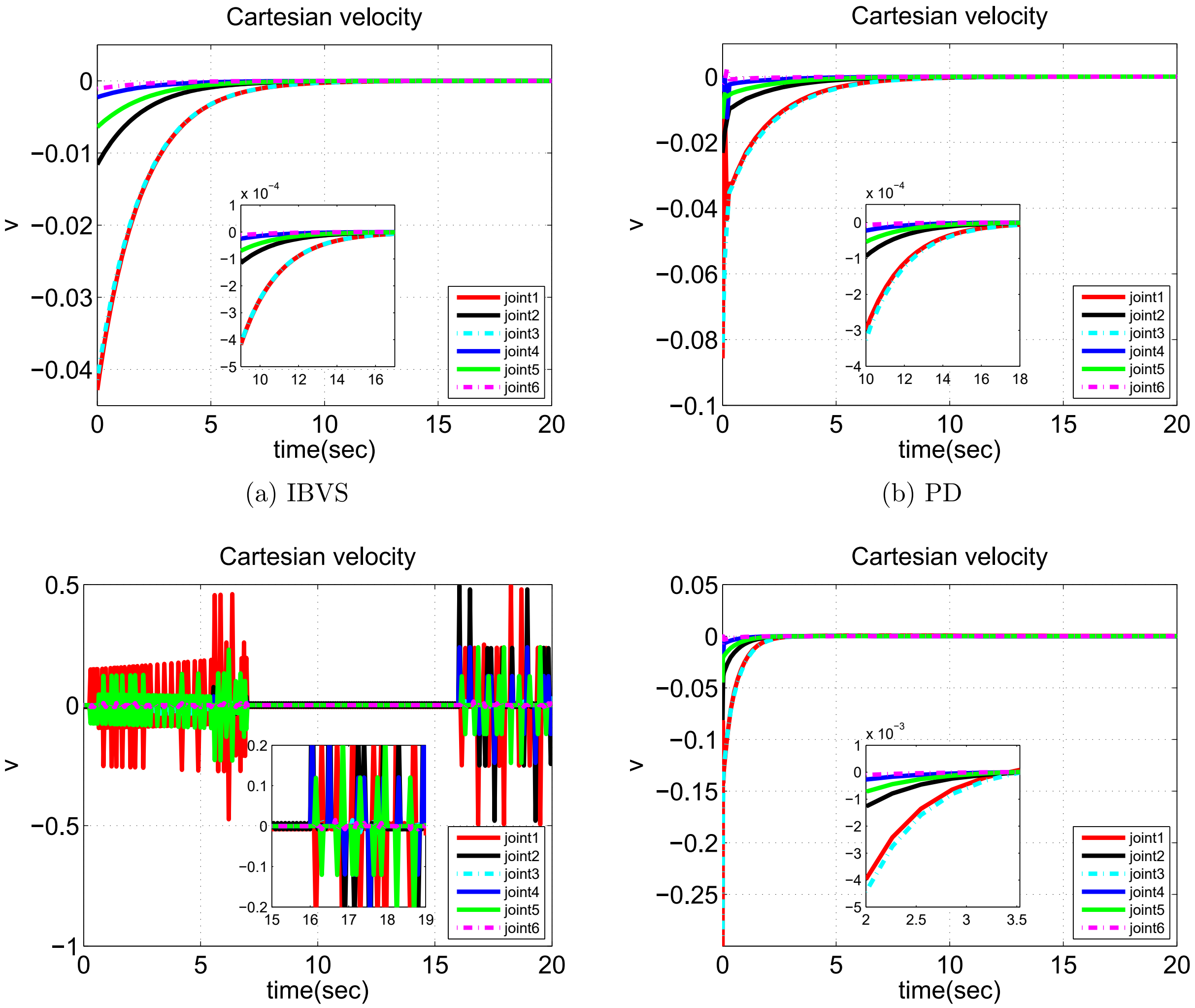
(a) IBVS

(b) PD

(c) SMC

(d) Hybrid PD-SMC

Figure 4.2: Feature error variations in pure translation test (Test 1).



(a) IBVS

(b) PD

(c) SMC

(d) Hybrid PD-SMC

Figure 4.3: Joint velocity variations in pure translation test (Test 1)

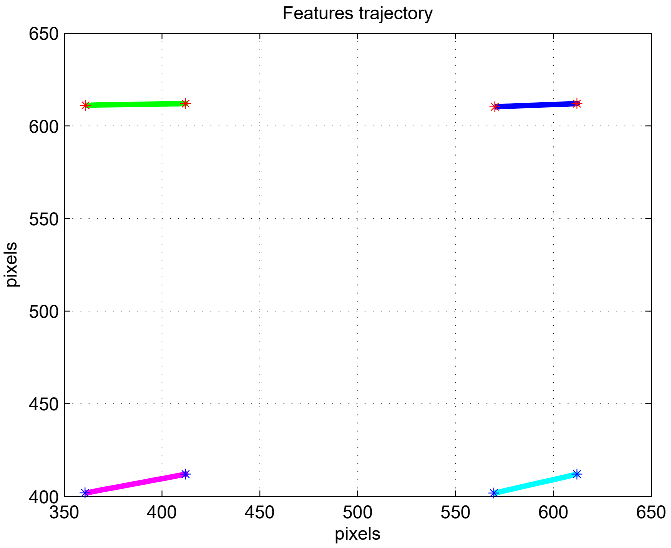
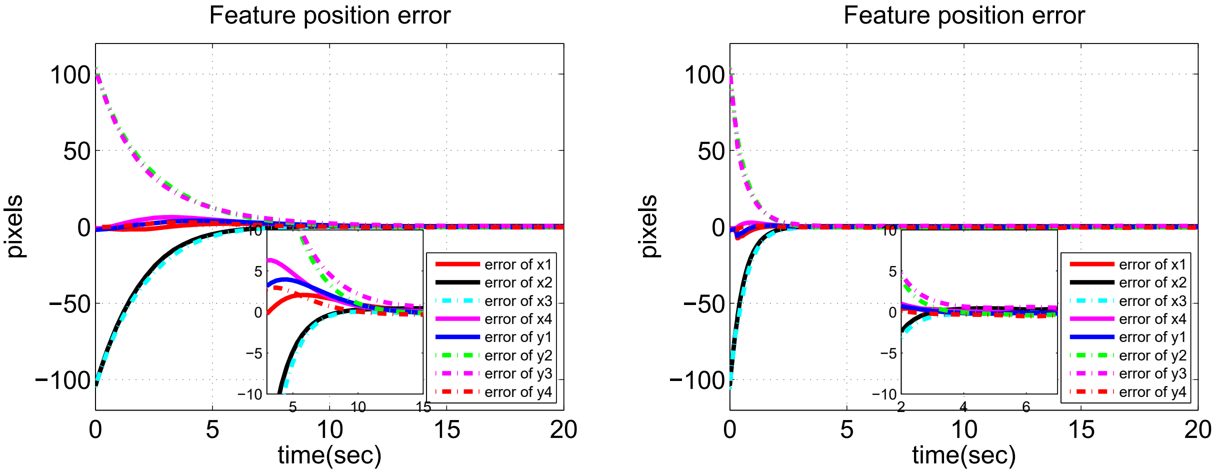


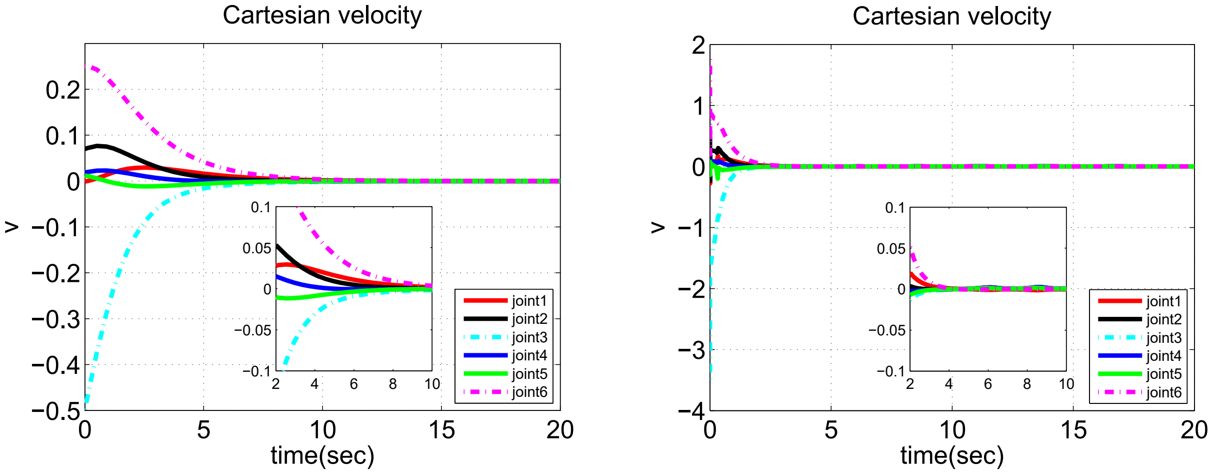
Figure 4.4: Feature trajectories in image space of pure translation test (Test 1)



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.5: Feature error variations in pure rotation test (Test 2)



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.6: Joint velocity variations in pure rotation test (Test 2)

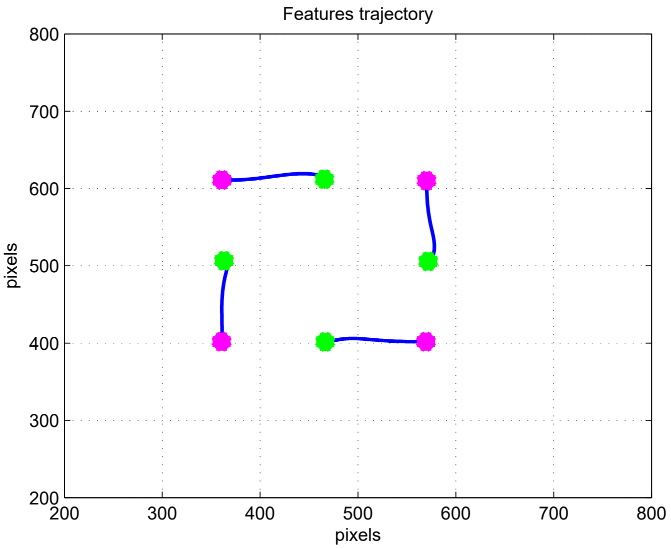
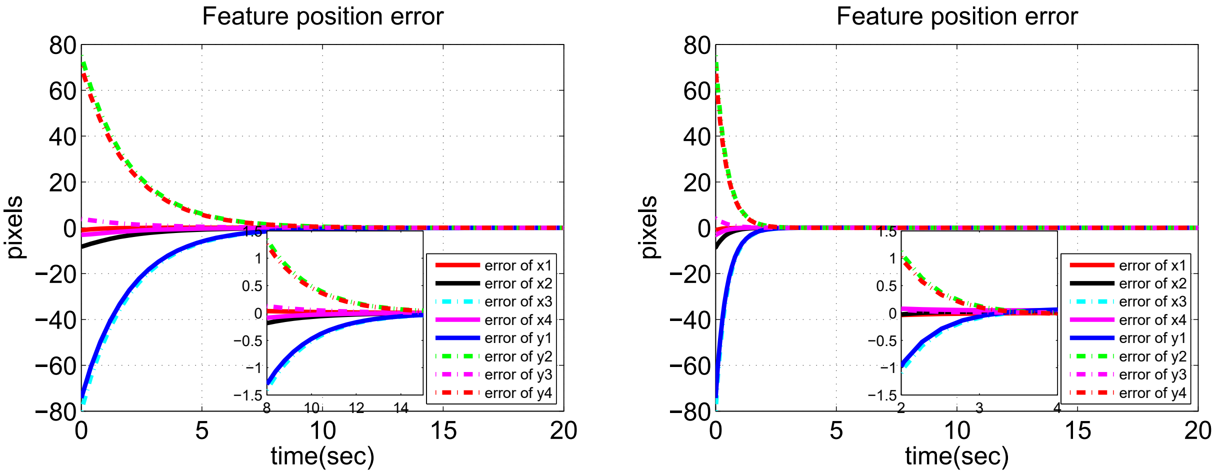


Figure 4.7: Feature trajectories in image space of pure rotation test (Test 2)

To further verify the effectiveness of the proposed method, the hybrid translational and rotational motion around a point is conducted in Test 3. Compared to the pure translational or rotational motion, the hybrid translational and rotational motion is a more complicated process. The results in Fig. 4.8, Fig. 4.9 and Fig. 4.10 demonstrate the hybrid PD-SMC owns higher convergence rate and more accurate trajectory.

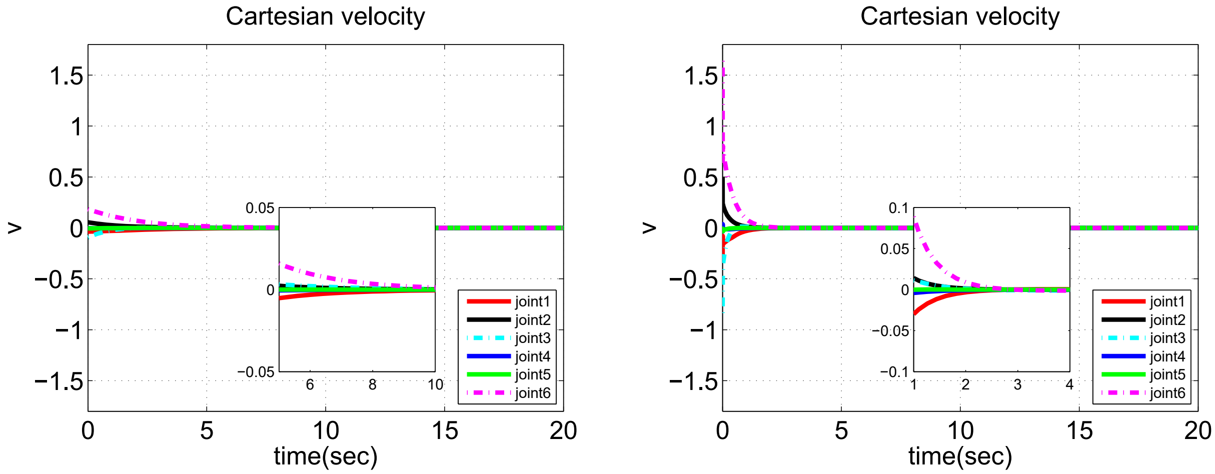
In Test 4, a chirp signal as a disturbance is added near to the desired position and the manipulator is controlled to move from the initial position to the desired position under the added disturbance. This test aims at demonstrating the robustness of the system. Feature position error variation and joint velocity are shown in Fig. 4.11 and Fig. 4.12. The results demonstrate that the system using hybrid PD-SMC method has the better stability and robustness.



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.8: Feature error variations in translation and rotation test (Test 3)



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.9: Joint velocity variations in translation and rotation test (Test 3)

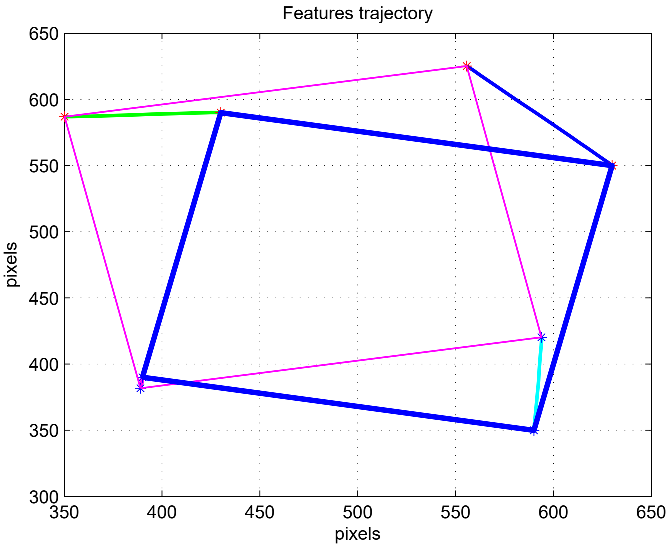
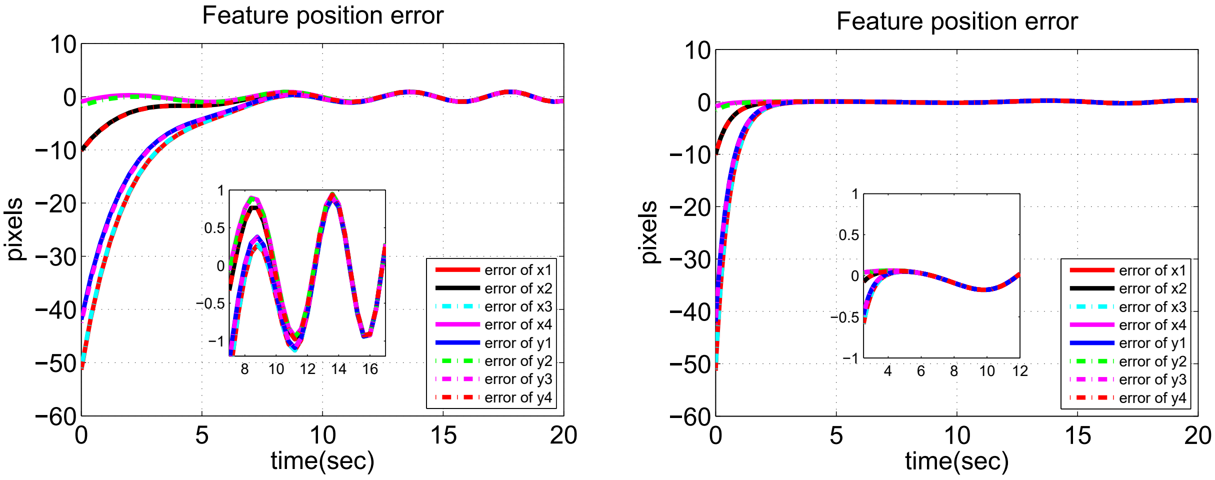


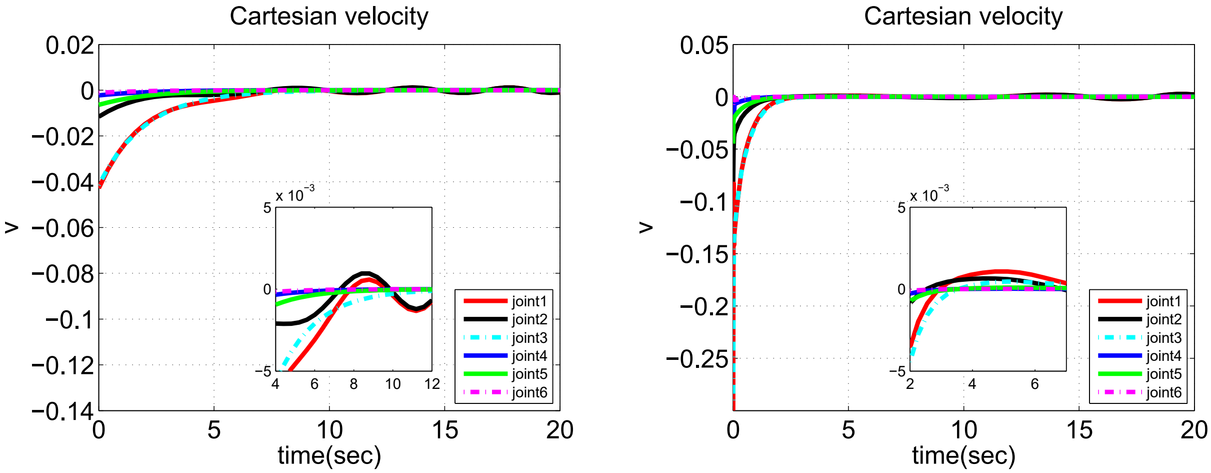
Figure 4.10: Feature trajectories in image space of translation and rotation test (Test 3)



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.11: Feature error variations with disturbance test (Test 4)



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.12: Joint velocity variations with disturbance (Test 4)

From the above simulation results, the accuracy and robustness of the system performance can be summarized in Table 4.3. It is shown that the settling time of Hybrid PD-SMC is shorter than that of IBVS. The added external disturbance has less effect on hybrid PD-SMC controller.

To further compare the performance of IBVS and Hybrid PD-SMC, the performance index ISE (Integrate Square Error) is adopted, which is defined as

$$\text{ISE} = \int_0^T e^2(t)dt. \tag{4.22}$$

The results are summarized in Table 4.4, where the "ISE Total" represents the total Integrate Square Error of feature error x_1, x_2, x_3, x_4 and feature error y_1, y_2, y_3, y_4. Results show that the ISE of the hybrid PD-SMC is smaller than that of the IBVS in the tests.

Table 4.3: Performance comparison of IBVS and Hybrid PD-SMC

		IBVS	**Hybrid PD-SMC**
Test 1	Settling time (seconds)	14	4
	Peak value (pixels)	0	0
Test 2	Settling time (seconds)	11	3.5
	Peak value (pixels)	0	0
Test 3	Settling time (seconds)	10	2.5
	Peak value (pixels)	0	0
Test 4	Settling time (seconds)	10	3
	Peak value (pixels)	1	0.2

Table 4.4: ISE values of IBVS and Hybrid PD-SMC

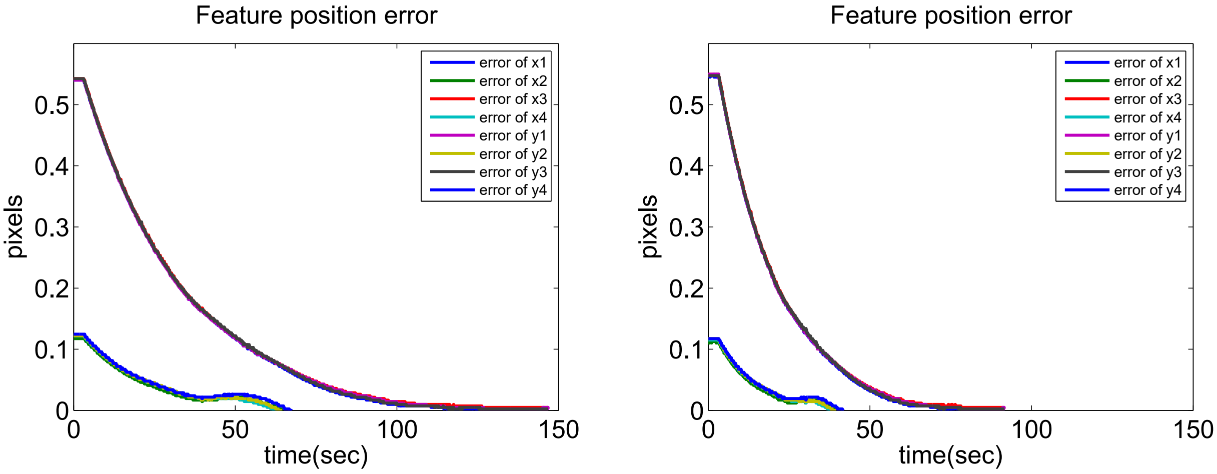
	ISE Total	
	IBVS	**Hybrid PD-SMC**
Test1	1.7875×10^4	5.3609×10^3
Test2	4.5601×10^5	1.6251×10^5
Test3	1.2518×10^5	1.4727×10^4
Test4	1.7639×10^4	5.3348×10^3

4.6 Experimental results

To further validate the performance of the proposed method, three experimental tests are conducted on 6-DOF Denso robot (described in 2.5) including long distance translation and pure rotation of features, and hybrid translation-rotation test. In these tests, the depth range is $0.5 < Z < 0.7$ meters. The diagonal elements of two parameter matrices K_d, K_s are 0.01, 0.26 respectively. Four feature points are used in visual servoing control. The initial and desired positions of the image features are given in Table 4.5.

Table 4.5: Initial and desired positions

		Point 1	**Point 2**	**Point 3**	**Point 4**
		$(x_1 \quad y_1)$	$(x_2 \quad y_2)$	$(x_3 \quad y_3)$	$(x_4 \quad y_4)$
Test 5	Initial	(57 150)	(57 57)	(146 63)	(146 148)
	Desired	(595 270)	(595 175)	(684 177)	(686 275)
Test 6	Initial	(454 213)	(385 146)	(447 81)	(516 148)
	Desired	(602 270)	(600 174)	(688 179)	(691 273)
Test 7	Initial	(103 136)	(196 105)	(225 187)	(134 220)
	Desired	(447 203)	(540 189)	(557 278)	(461 292)

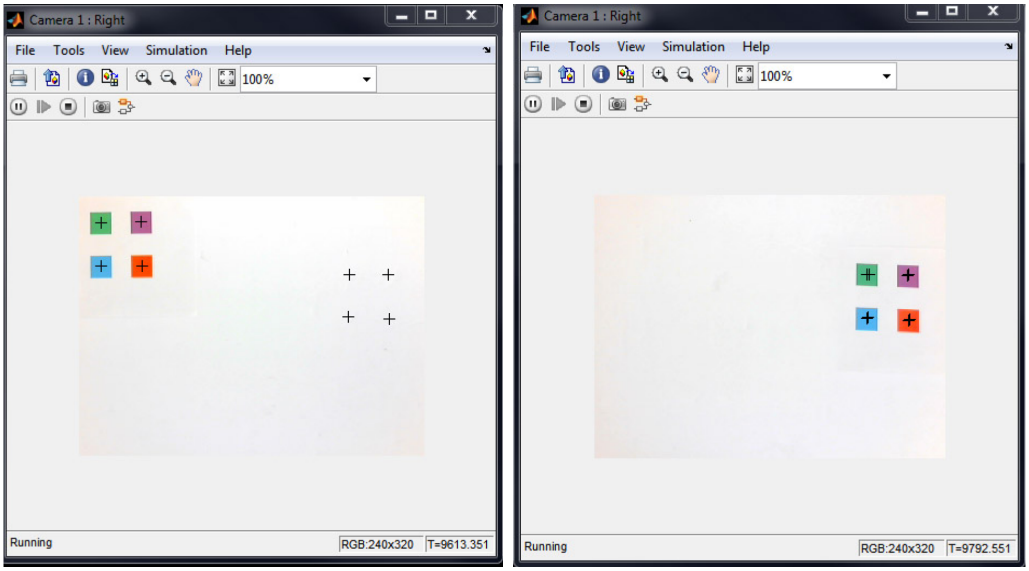


(a) IBVS

(b) Hybrid PD-SMC

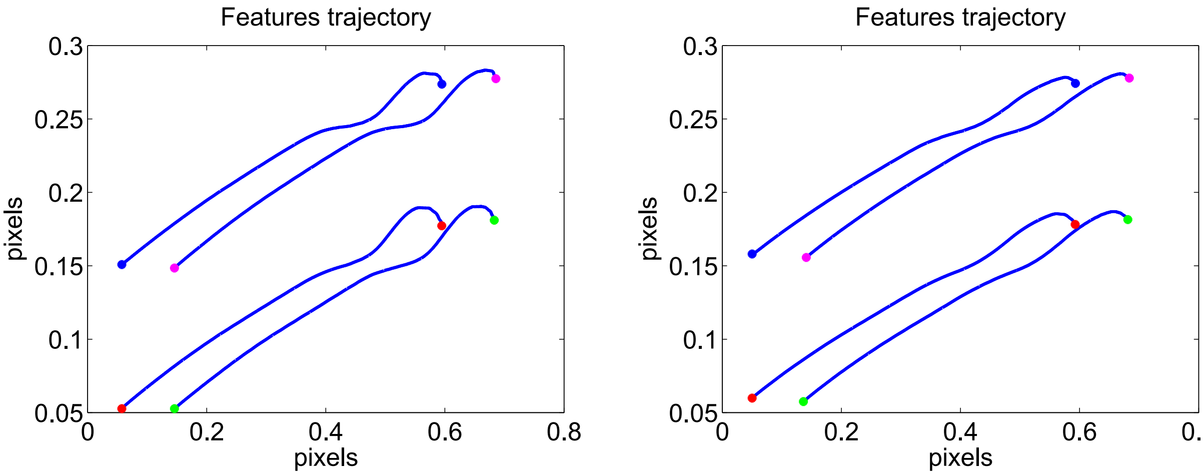
Figure 4.13: Feature error variations in a long distance translational motion (Test 5)

Test 5 is performed to examine the convergence of image feature points when the desired position is far away from the initial one, which needs a long distance translational motion. Fig. 4.13 shows that the feature position errors converge to zero. Fig. 4.14 shows the initial and desired positions captured by the camera. Fig. 4.15 shows the feature trajectories in image space of a long distance translational motion. Fig. 4.16 shows the camera trajectory in Cartesian space.



(a) Initial position (b) Desired position

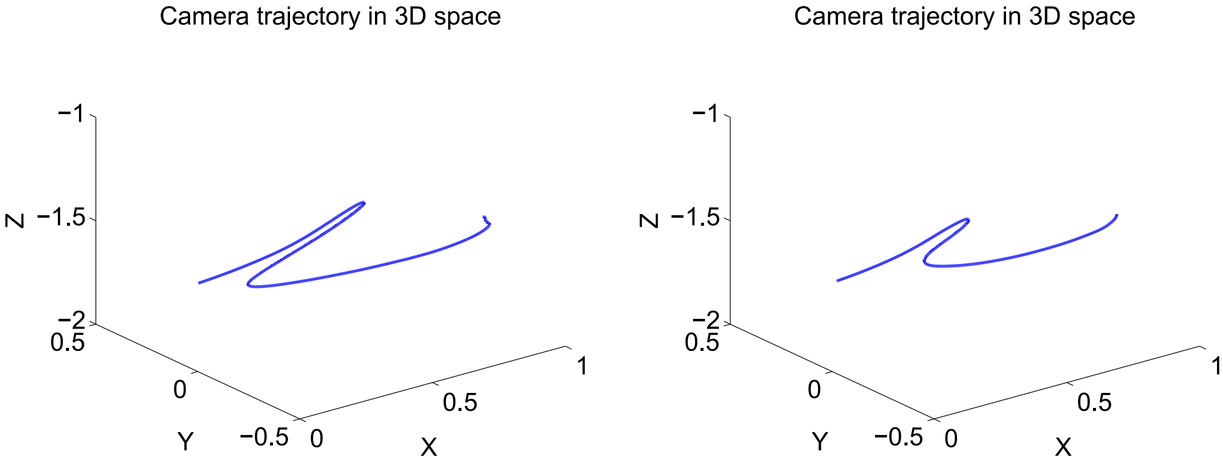
Figure 4.14: Feature position (Test 5)



(a) IBVS (b) Hybrid PD-SMC

Figure 4.15: Feature trajectories in image space of a long distance translational motion (Test 5)

It is shown that the performance of hybrid PD-SMC is better than that of IBVS. The settling time of the hybrid PD-SMC method is shorter than that of conventional method. Furthermore, in hybrid PD-SMC method, the feature trajectory is straighter in image plane and the camera trajectory in Cartesian space is smoother.

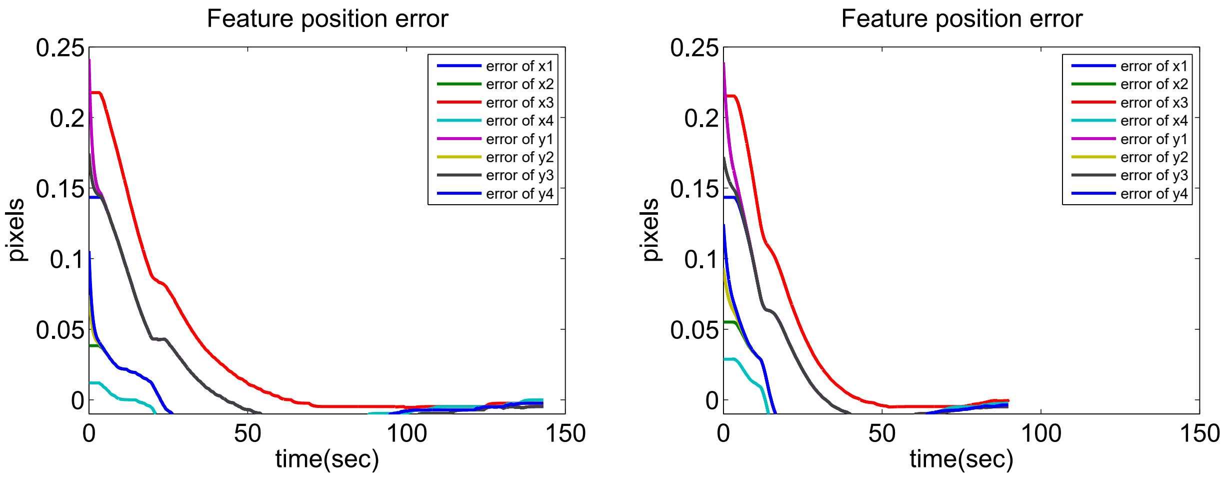


(a) IBVS

(b) Hybrid PD-SMC

Figure 4.16: Camera trajectory in Cartesian space (Test 5)

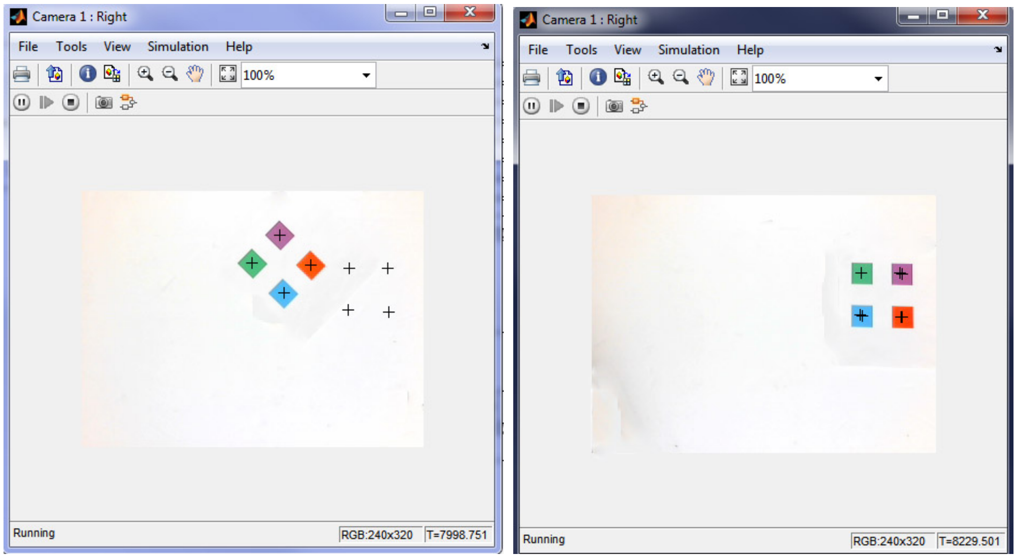
Test 6 is performed to examine the rotation performance of the proposed method, a pure rotation of image feature points has been completed. Fig. 4.17 shows that the feature position errors converge to zero. Fig.4.18 shows the initial and desired positions which are captured by the camera. Fig. 4.19 shows the feature trajectory in image plane. Fig. 4.20 shows the camera trajectory in Cartesian space.



(a) IBVS

(b) Hybrid PD-SMC

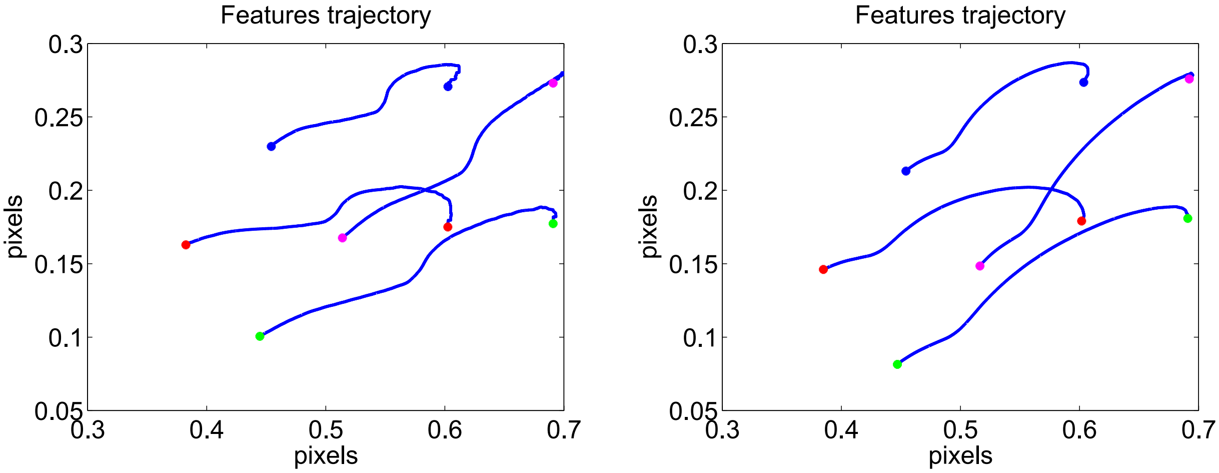
Figure 4.17: Feature error variations in a pure rotational motion (Test 6)



(a) Initial position

(b) Desired position

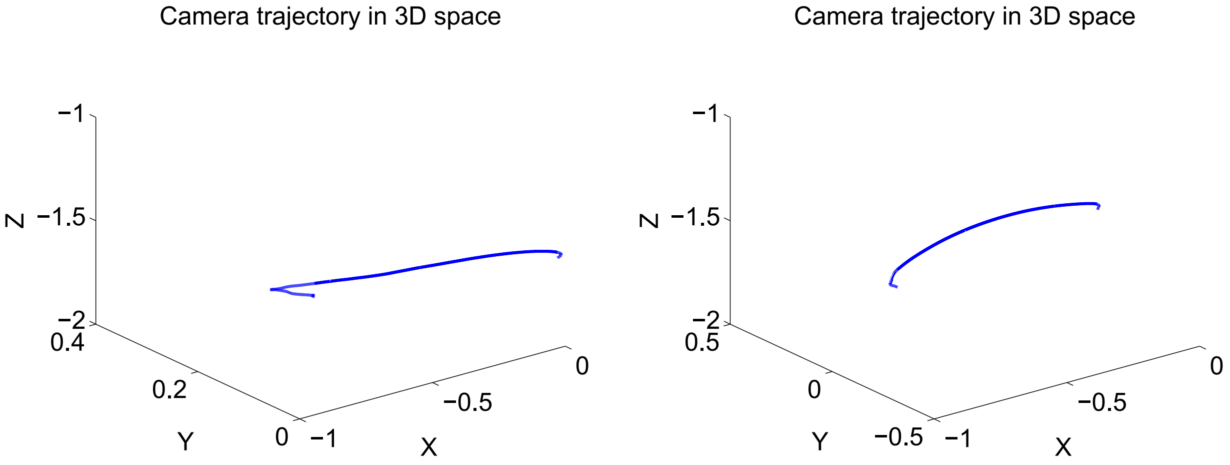
Figure 4.18: Feature position (Test 6)



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.19: Feature trajectories in image space of a pure rotational motion (Test 6)



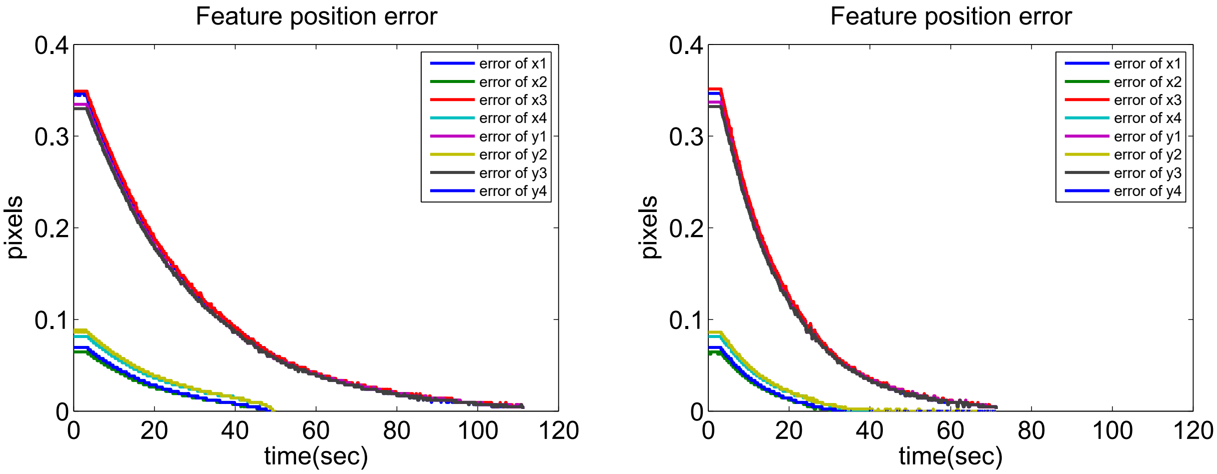
(a) IBVS

(b) Hybrid PD-SMC

Figure 4.20: Camera trajectory in Cartesian space (Test 6)

It is obvious that the test is successfully performed to validate the superior performance of hybrid PD-SMC. Fig. 4.17, Fig. 4.18, Fig. 4.19 and Fig. 4.20 show the comparison of experimental results, which is in agreement with those of Test 5.

Test 7 is a hybrid translational-rotational motion process. In this experimental test, the translational and rotational motions of features are incorporated in one process. In the initial stage of the movement, the translational motion is implemented. In the final stage of the movement, the rotational motion is performed. Fig. 4.21 shows the feature position error variations of IBVS and hybrid PD-SMC. It is observed that the hybrid PD-SMC system owns the higher convergence rate. Fig. 4.22 shows the six joints's velocity of IBVS and hybrid PD-SMC respectively. The joint velocity variations using hybrid PD-SMC method have regularly changing pattern and smaller shaking phenomenon.

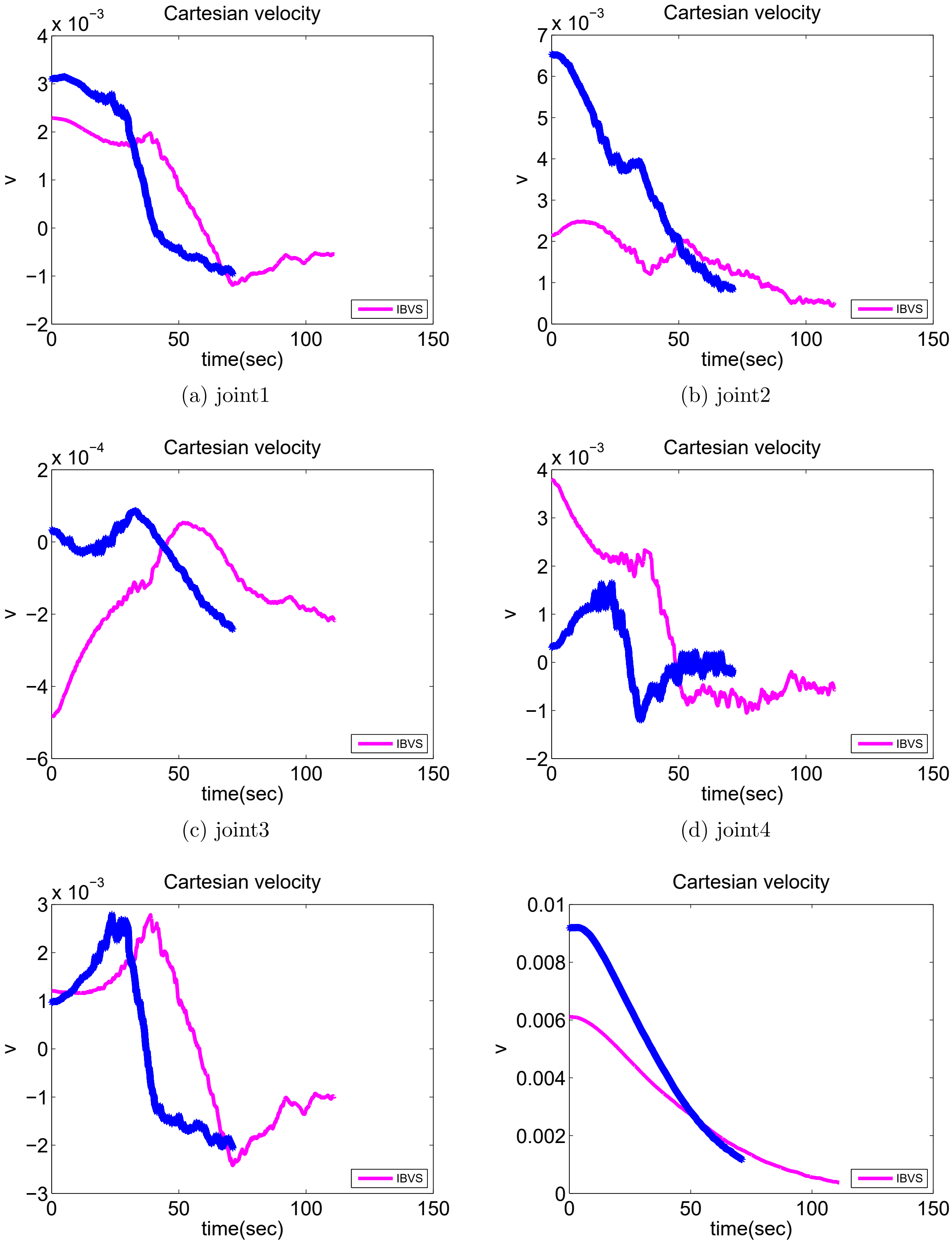


(a) IBVS

(b) Hybrid PD-SMC

Figure 4.21: Feature error variations (Test 7)

Fig. 4.23 (a) and Fig. 4.23 (b) show the image feature points from initial position to final position and the trajectory by using IBVS and hybrid PD-SMC respectively. It is observed that the hybrid PD-SMC performs better in the final stage than IBVS in terms of smoothness and length of its trajectories in image plane.



(a) joint1

(b) joint2

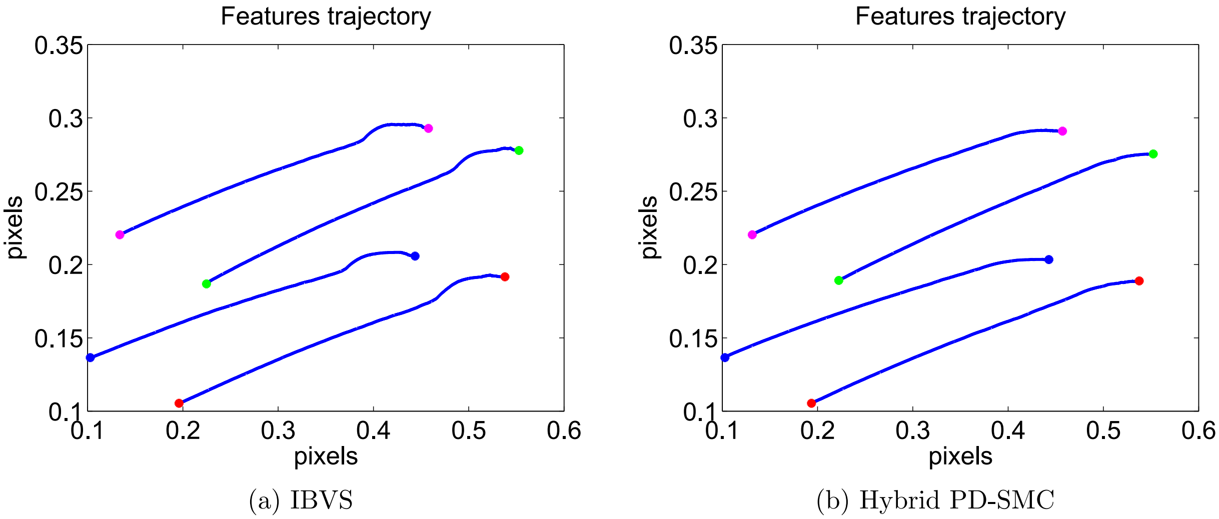
(c) joint3

(d) joint4

(e) joint5

(f) joint6

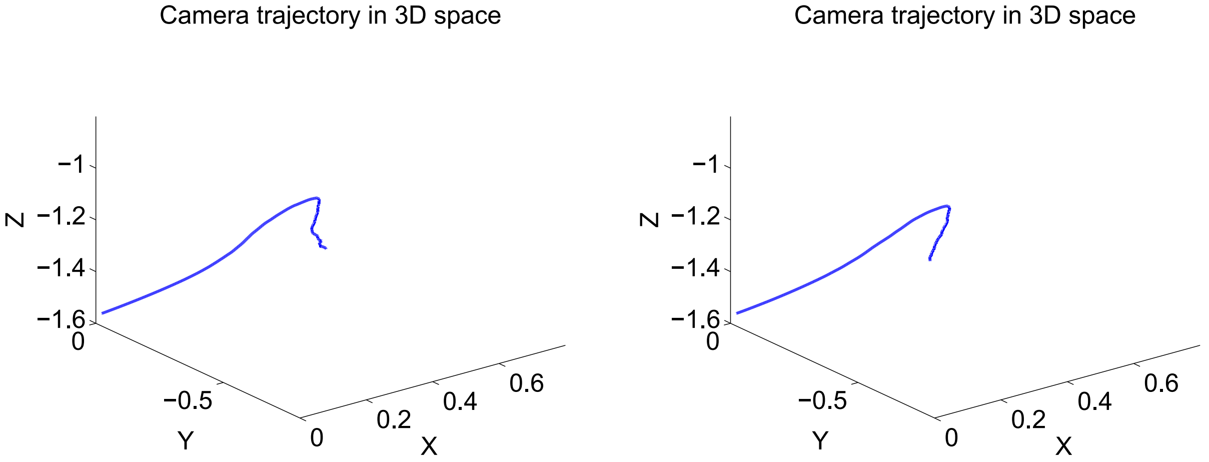
Figure 4.22: Joint velocity of using IBVS and hybrid PD-SMC (Test 7)



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.23: Features trajectories (Test 7)



(a) IBVS

(b) Hybrid PD-SMC

Figure 4.24: Three dimensional trajectory of the camera (Test 7)

Fig. 4.24 (a) and Fig. 4.24 (b) show the camera trajectory in 3D space of IBVS and hybrid PD-SMC respectively. It can be seen that the camera trajectory of hybrid PD-SMC system is smoother and straighter.

The above experimental results on the system performance are summarized in Table 4.6. It is noted that the settling time of Hybrid PD-SMC is less than that of IBVS. Furthermore, external disturbances may exist during the movement in real experiments. The robustness against the random disturbances during the experiment is demonstrated in movement. By comparing the feature trajectories in image space in Fig. 4.15, Fig. 4.19 and Fig. 4.23, one notices that the interference effect was significant on the final stage of the movement. When reaching this stage, the feature trajectories in image space of the hybrid PD-SMC system are smoother and straighter than those of conventional IBVS. By comparing the camera trajectory in 3D space in Fig. 4.16 and Fig. 4.20, it can be observed that the trajectory in hybrid PD-SMC system is more preferred in critical converted position. By comparing the camera trajectory in 3D space in Fig. 4.24, especially on the stage close to the desired position, one can draw the conclusion that the hybrid PD-SMC system has better robustness and efficiency.

The performance index ISE (Integrate Square Error) is also used to further compare the performance of IBVS and Hybrid PD-SMC. The results are described in Table 4.7, where the "ISE Total" represents the total Integrate Square Error of feature error x_1, x_2, x_3, x_4 and feature error y_1, y_2, y_3, y_4. Table 4.7 shows that the ISE of the hybrid PD-SMC is smaller than that of the IBVS in three tests.

Table 4.6: Performance comparison of IBVS and Hybrid PD-SMC

	Settling time (seconds)	
	IBVS	**Hybrid PD-SMC**
Test 5	146	91
Test 6	143	89
Test 7	111	71

Table 4.7: ISE values of IBVS and Hybrid PD-SMC.

	ISE Total	
	IBVS	**Hybrid PD-SMC**
Test5	1160.6	791.5
Test6	112.6	92.7
Test7	438.6	296.8

Therefore, one can see that the accuracy and robustness of the hybrid PD-SMC system are better than those of IBVS system. The effectiveness of the proposed controller is validated in three experimental tests.

4.7 Summary

An enhanced IBVS which combines PD control with SMC is presented in this chapter. The purpose of this approach is to improve the visual servoing performance by taking advantages of PD control and SMC so that the proposed hybrid PD-SMC controller owns good robustness against the disturbance and uncertainties due to the estimated depth. Also it has fast convergence rate. The stability of the enhanced IBVS system is proven by using Lyapunv function method. Simulation and experimental tests demonstrate that the proposed hybrid PD-SMC IBVS excels greatly the classic IBVS controller.

Chapter 5

Image-Based Visual Servoing Using Trajectory Planning

5.1 Introduction

In previous chapters of this research work, several methods are proposed that could deal with system uncertainties and increase the visual servoing task speed. On the other hand, they could successfully overcome some of the mentioned deficiencies. Specifically switch method could overcome the weakness of IBVS about pure rotation around camera's center. Sliding method and switch method increase the stability of the system in long distance tasks and ESIBVS tackles the problem of features leaving the field of view. However, still some of these deficiencies still remained unsolved.

In this chapter, a new image-based trajectory planning algorithm is proposed to overcome the visual servoing deficiencies and develop a reliable algorithm to perform visual servoing tasks. In this approach, a trajectory is generated based on the information received from the image plane. However the trajectory is in the Cartesian space and relates the end effector velocity to the motion of the features in image space. For this matter, the camera's velocity screw is separated into elements. Each velocity element is parameterized using a time based function which is refereed to as the velocity profiles. The velocity profile parameters are determined through an optimization process which minimizes the features errors. In order to facilitate and speed up the optimization technique, some new features are introduced. Due to the highly coupled behavior of the features due to the motion of camera, the optimization problem is a non-convex problem. By decoupling the orientation planning from positioning problem the problem becomes a convex problem. A convexity analysis is performed to show the convexity of the optimization problem. Similar to other IBVS systems, depth estimation plays an important role in the performance of the proposed trajectory planning algorithm. A primary depth estimation technique is introduced. Having the initial depth, the object depth could be integrated during the visual servoing task. By integrating this technique, the proposed image-based trajectory planning can overcome IBVS deficiencies to a great extent. In addition, this method eliminates the field of view constrains exist in conventional IBVS systems. This technique exploits the benefits of global off-line planning in visual servoing. However high speed of the algorithm allows the fast and easy execution of the algorithm.

Calibration error could deviate the robot from its ideal path. However, the robot is taken to a situation close enough to the desired location. The desired location will then be reached using an augmented image-based visual servoing (AIBVS) controller [64]. In other words, the trajectory planning algorithm is switched to a controller at the end of its path to compensate for any inaccuracy of the system performance. In summary, the whole visual servoing procedure consist of 3 stages. The first stage is the depth estimation stage. The second stage is the trajectory planning stage. Finally, in the third stage the trajectory planning block switches to a visual servoing controller block.

Experimental tests are performed on a 6-DOFs Denso robot to validate the proposed method. The results show that in the situations where the visual servoing task fails using traditional method, it performs successfully using the proposed method in this thesis.

5.2 Visual Servoing System

In this chapter, the goal is to develop a trajectory planning algorithm for an imaged-based visual servoing task. Image-based visual servoing is performed based on the difference between the current image features and the desired ones. A picture is taken of the object when the robot's end-effector is in desired position with respect to the object. This picture is used as the desired picture. The IBVS controller generates a velocity or acceleration command to eliminate the existing error. The visual servoing task is complete when the image features match the target features.

All the system modeling is based on a 6-DOFs robotic system with a pinhole CCD camera mounted on its end effector. Let $\mathfrak{F}_b$ be the robot base frame, $\mathfrak{F}_e$ be the end-effector frame and $\mathfrak{F}_c$ be the camera frame (5.1). The object is stationary in the workspace and is characterized

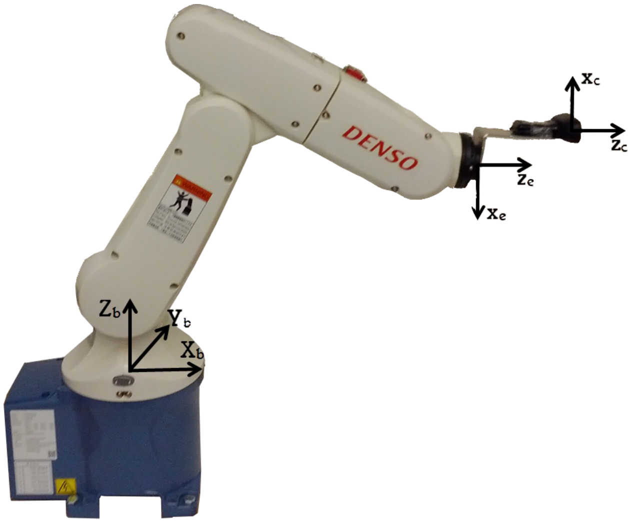


Figure 5.1: Denso robot

by 4 feature points on its four corners. A mentionable merit of IBVS is that it does not require the object frame. Having the projection of a 3D points on the image plane of the camera (Figure 1.10) the relation between the motion of the camera and the motion of the features could be calculated from 5.1.

$$\dot{\mathbf{p}} = \mathbf{L}_s {}^c\mathbf{V}_c, \tag{5.1}$$

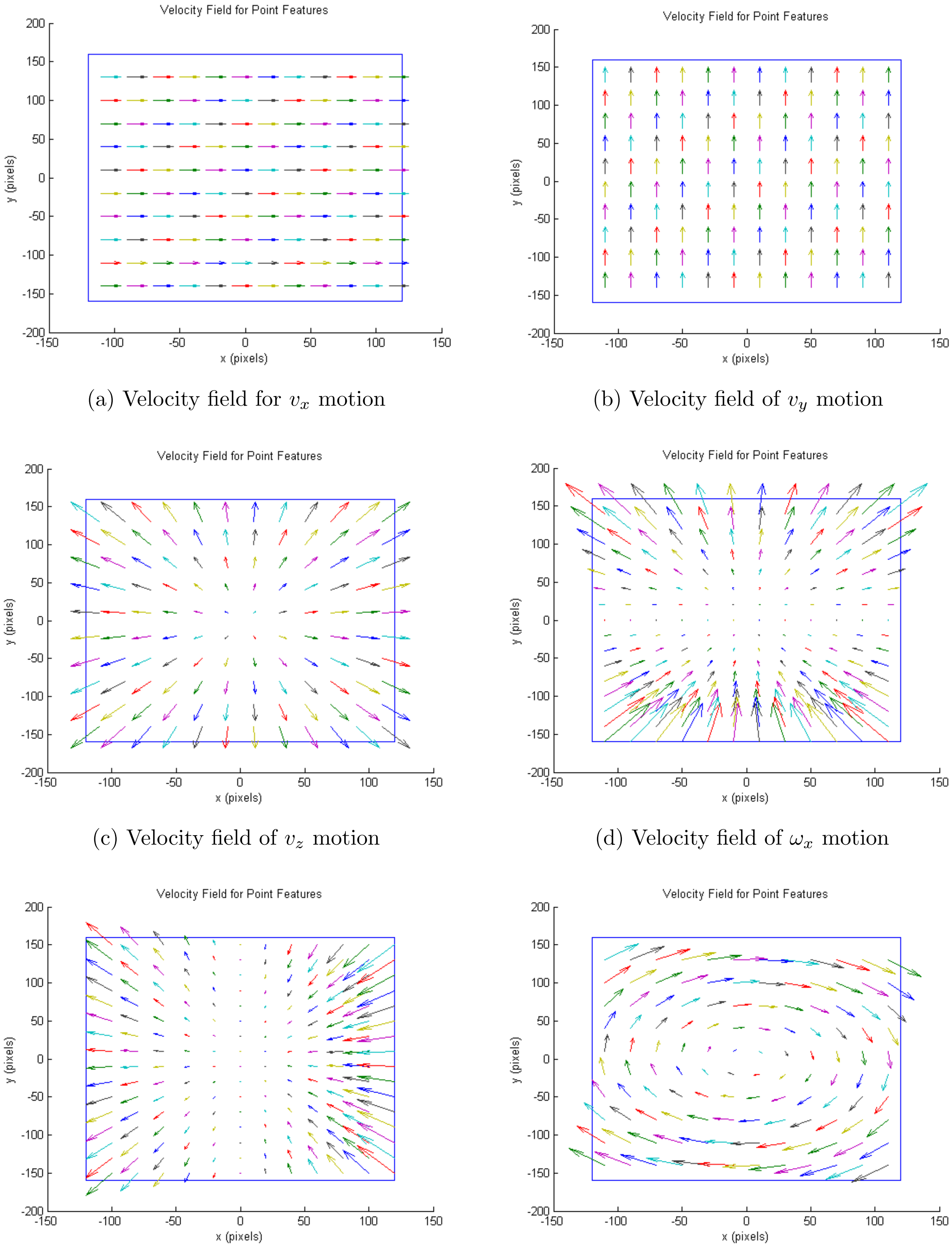
where

$$\mathbf{L}_s = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \tag{5.2}$$

is the Jacobian matrix, x and y represent the point coordinates in image plane in meter represented in camera frame, Z is the depth of the object with respect to the camera and ${}^c\mathbf{V}_c = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^{\mathrm{T}}$ is the camera's velocity screw represented in camera frame.

5.3 Trajectory Planning

The robot could perform 6 degrees of motion to reach any desired pose (including position and orientation). The effect of each motion could be calculated using equation (5.2). Figure 5.2 shows how each motion affects the feature point position.



(a) Velocity field for v_x motion

(b) Velocity field of v_y motion

(c) Velocity field of v_z motion

(d) Velocity field of ω_x motion

(e) Velocity field of ω_y motion

(f) Velocity field of ω_z motion

Figure 5.2: Velocity field of the features subject to camera velocities

The first two elements of the velocity screw create linear motions in the same direction for all features (Figures 5.2a and 5.2b). These two camera motions are used for displacing the features in x and y direction of the image plane. A camera motion in Z_c direction creates an outward motion for the features which is in the direction of line connecting the center of the image to the image feature (Figure 5.2c). A negative motion in Z direction will create an inward motion for the features. This motion could compensate the distances between the features. The fourth and fifth element of the velocity screw create a complicated motion in the features. It creates an inward motion for features in one side of the image and an outward motion for the features on the other side of the image (Figures 5.2d and 5.2e). The last element of the velocity screw rotates the features about the center of image (Figure 5.2f).

The concept behind the trajectory planning is that any target features could be reached by using a combination of shown feature motions. Six basic velocity profiles are generated for each of the camera's velocity screw elements. The effect of the generated velocity screw can be calculated using equation (5.2). In other words, by superposing the velocity fields caused by each element of the velocity screw, the final position of the features could be calculated. The parameters of the camera velocity are then determined by minimizing the error between the image features and the target ones.

The features velocity in image space could be written as a function of velocity screw elements, given by

$$\begin{aligned} \dot{x}_i &= \tfrac{-1}{Z}v_x + \tfrac{x_i}{Z}v_z + x_i y_i \omega_x - (1+x_i^2)\omega_y + y_i\omega_z \\ \dot{y}_i &= \tfrac{-1}{Z}v_y + \tfrac{y_i}{Z}v_z + (1+y_i)^2\omega_x - x_i y_i \omega_y - x_i\omega_z, \end{aligned} \tag{5.3}$$

where $\dot{x}_i$ and $\dot{y}_i$ are the velocities of the ith image feature in x and y direction, respectively. Consequently, the image feature position could be calculated as

$$\begin{aligned} x_{it} &= \int_{t_0}^{t} (\dot{x}_i(t))dt + x_{i0} \\ y_{it} &= \int_{t_0}^{t} (\dot{y}_i(t))dt + y_{i0}, \end{aligned} \tag{5.4}$$

where x_{i_0} and y_{i_0} are the initial coordinates of the image features and x_{it} and y_{it} are the locations of the image features at time t. Thus, by knowing the initial position of the features and the velocity of the camera the position of the features can be calculated at each time.

5.3.1 Image Features

The Jacobian matrix achieved for point features (equation (5.2)) is highly nonlinear and coupled. In order to facilitate the optimization process some new features are presented in this research. The new set of image features is as

$$\mathbf{s_n} = \begin{bmatrix} x_c & y_c & p_z & \theta_x & \theta_y & \theta_z \end{bmatrix}^{\mathrm{T}}, \tag{5.5}$$

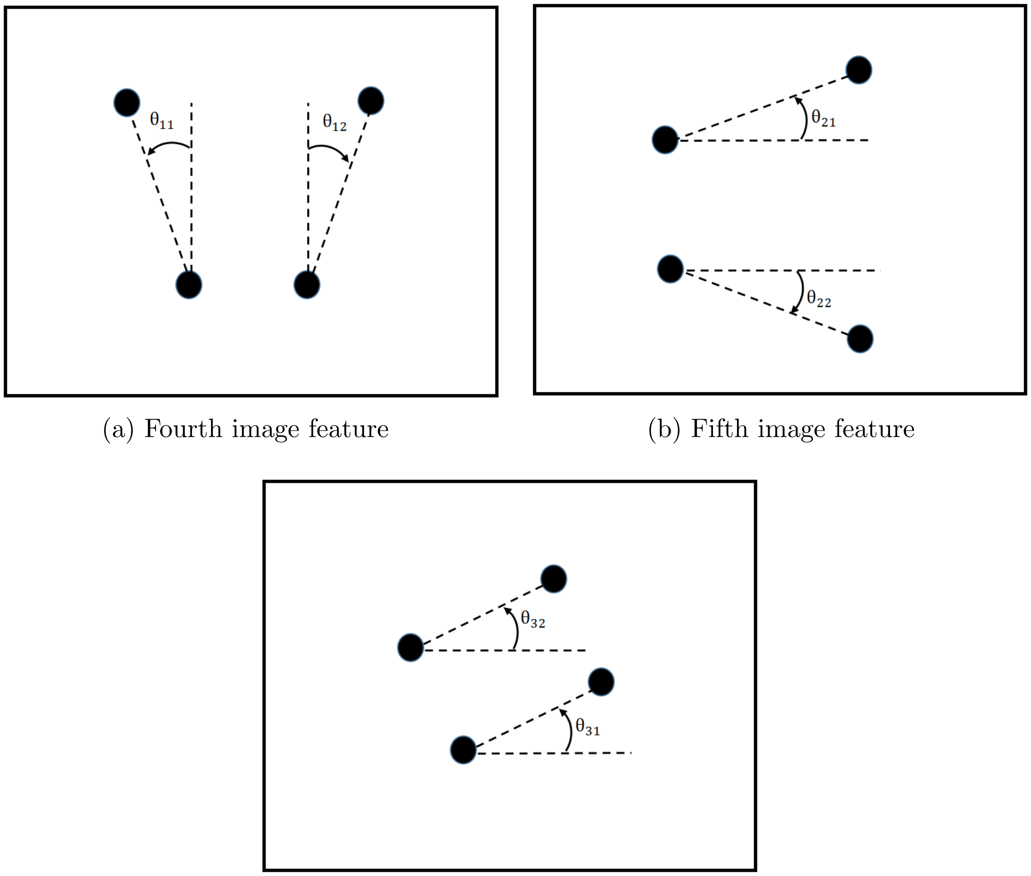
where x_c and y_c are the centers of the feature points and p_z is the perimeter of the lines connecting each consecutive feature point which are given as

$$\begin{array}{l} x_c = \frac{\sum_{i=1}^{4} x_i(t)}{4} \\ y_c = \frac{\sum_{i=1}^{4} y_i(t)}{4} \\ p_z = \sum_{i=1}^{4} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \end{array} . \tag{5.6}$$

$\theta_x(t)$, $\theta_y(t)$ and $\theta_y(t)$ are defined based on the deformation that is made in the features by rotating the camera about ${}^cX_c$, ${}^cY_c$ and ${}^cZ_c$. These features are given by

$$\begin{array}{l} \theta_x = \frac{\theta_{11}+\theta_{12}}{2} \\ \theta_y = \frac{\theta_{21}+\theta_{22}}{2} \\ \theta_z = \frac{\theta_{31}+\theta_{32}}{2} \end{array} , \tag{5.7}$$

where θ_{11}, θ_{12}, θ_{21}, θ_{22}, θ_{31}, θ_{32} are shown in the Figure 5.3.



(a) Fourth image feature

(b) Fifth image feature

(c) Sixth image feature

Figure 5.3: Last three image features definition

5.3.2 Depth Estimation

To accurately calculate the location of the feature points, the distance between the object and the camera is required. The motion of the camera in Z_c direction is known from v_z element of the velocity screw which is given as a parameterized equation of time ($v_{tp_z} = f_z(t)$). Thus the depth Z can be calculated at any time t from

$$Z_t = \int_{t_0}^{t} f_z(t) + Z_0, \tag{5.8}$$

where, Z_0 is the initial depth of the object with respect to the camera coordinates. If the initial depth of the object is estimated, accurately, the depth in the rest of the times could be calculated. Let us recall that using a stereo camera the depth of the object could be calculated [131–133]. This is done by applying the epipolar geometry constraint that exist between the features in images planes of each camera. In a simple case where the two cameras are mounted parallel to each other (Figure 5.4), the depth of the object with respect to the cameras can be calculated using the disparity of the images from equation (5.9).

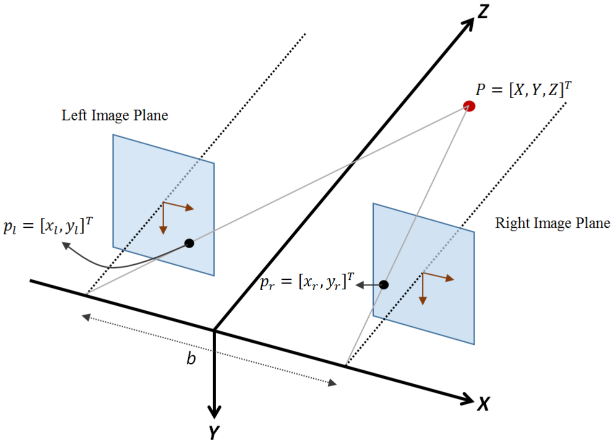


Figure 5.4: Stereo camera model

$$Z_c = \frac{b}{x_l - x_r}, \tag{5.9}$$

where Z_c is the depth of the object in the camera coordinates, x_r and x_l are the features x coordinates in left and right cameras, respectively and b is the distance between the cameras. We can conclude that by having two image of an object from a camera from which the second image is taken at a location with a displacement of b along X_c from the first location of the camera, the same equation could be used to calculate the object depth. Thus, by moving the camera along X_c by a small displacement b and using the initial and the final image feature position and the depth of the object could be calculated from equation (5.9), This procedure takes about 1 second to complete which is feasible in experiment.

5.3.3 Parameterizing the Velocity Profile

A general predefined velocity profile is selected and named $\mathbf{V}_t(t)$. In a visual servoing task which deals with a stationary object, the robot starts from stationary situation and ends in a stationary situation. Thus, the selected profile needs to satisfy the following conditions.

$$\begin{array}{l} \mathbf{V}_t(0) = 0 \\ \mathbf{V}_t(t_f) = 0 \end{array} , \tag{5.10}$$

where t_f is the final time which we planned to have the robot at the target position. Some examples of these functions could be a trapezoid function, a polynomial function or half cycle of a sinusoidal wave. However, more complicated trajectories with more parameters could be used such as higher order polynomial especially for the cases where other objective functions such as energy or path length are used for optimization.

In this research, half cycle of a sinusoidal profile is used to parameterize the velocity profile. The velocity profile could be shown as follows

$$\mathbf{V}_t(t) = \mathbf{v}_m \sin(\tfrac{\pi t}{t_f}) \qquad 0 \leq t \leq t_f \ , \tag{5.11}$$

where $\mathbf{v}_m$ is the vector of maximum speed that the camera reaches within the profile and it is given as;

$$\mathbf{v}_m = \begin{bmatrix} v_{mx} \; v_{my} \; v_{mz} \; v_{m\omega_x} \; v_{m\omega_y} \; v_{m\omega_z} \end{bmatrix}^{\mathrm{T}} , \tag{5.12}$$

where v_{mx},v_{my},v_{mz}, $v_{m\omega_x}$, $v_{m\omega_y}$ and $v_{m\omega_z}$ are the maximum velocity of each element in the velocity screw, respectively. The final time, t_f, is selected by the user depending on the desired speed of the task. Thus, each profile have only one parameter to be designed and the overall number of design parameters of the system is six.

5.3.4 Decoupling Orientation Planning from Position Planning

Testing the trajectory planning as explained above shows that the system is highly nonlinear and the optimization process is not convex. In some cases the process doesn't converges and in other cases their is no guarantee that it converges in a reasonable time. Due to the important role that the convergence time plays in feasibility of the algorithm for an industrial application, it is proposed to decouple the orientation planning from position planning. Decoupled visual servoing controller is presented in [11]. In this thesis decoupled trajectory planning is investigated.

Decoupling is performed as explained at follows. First, the last three velocity screw elements are planned in the optimization process so that they take the last three feature set elements to their desired values. Second, the first three elements of the velocity screw is planned to eliminate the error existing in the first three elements of the feature set. The last three joints of the robot is responsible for the fixing the orientation and the first three joints of the robot is responsible for positioning. As it is investigated in the next section, using the selected features and decoupling the planning process creates a convex optimization process.

5.4 Optimization and Convexity Analysis

Let us define the objective function as the quadratic form of the selected features error, given by

$$OF = (\mathbf{s}_n(t_f) - \mathbf{s}_{nd})^{\mathrm{T}}\mathbf{Q}(\mathbf{s}(t_f) - \mathbf{s}_d), \tag{5.13}$$

where $\mathbf{Q}$ is an orthogonal matrix introducing the desired weight of each error in the optimization process. An important point that needs to be considered, is that the trajectory planning procedure must be completed in a reasonable time. Otherwise, the method would be useless for real word applications because of the delay that is imposed to the system. One important factor that leads to fast convergence of the optimization problem is the convexity of the optimization problem. In this section the convexity of the problem is investigated. To start, let us review the following main theorems regarding convexity of a problem.

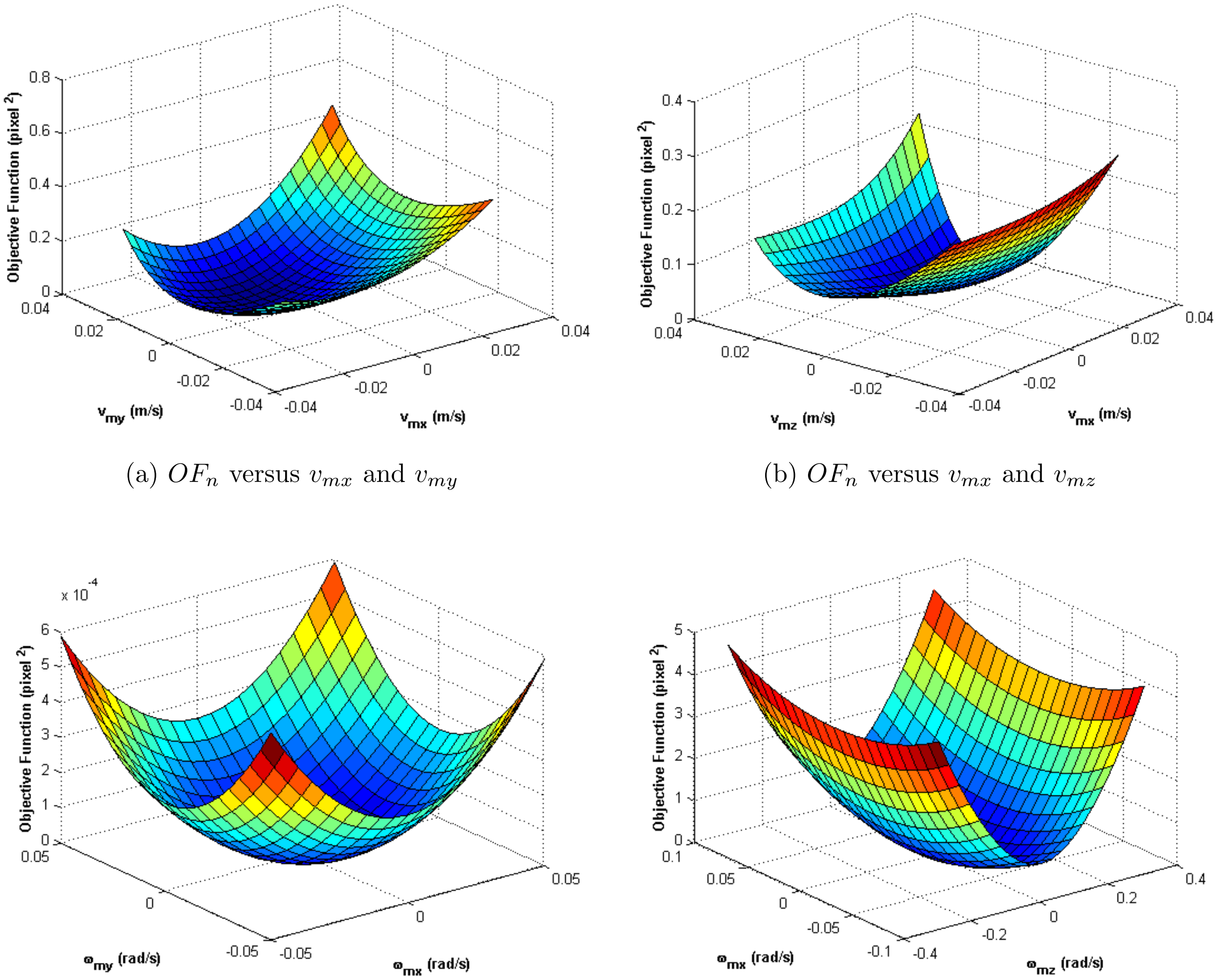
Theorem 1: If $f(x^*)$ is a local minimum for a convex function $f(x)$ defined on a convex feasible set S, then it is also a global minimum [134].

Theorem 2: A function of n variable $f(x_1, x_2, ..., x_n)$ is defined on a convex set S is convex if and only if the Hessian matrix of the function is positive semidefinite or positive definite at all points in the set S [134].

Proving the convexity of the objective function given in equation (5.13) requires the Hessian matrix of OF. Chinneck [135] introduced a method to discover the convexity of a program using numerical method. Accordingly, a code is generated to numerically calculate the Hessian matrix ([136]) of the objective function for a desired span of the desired parameters. The design parameter range depends on the physical limitations of the robot. In this case, the design parameters are the maximum velocity of the end-effector in the associated DOF. Knowing the speed limits of the robotic system, this could be identified. In our test the following ranges have been used,

$$\begin{aligned}
-0.1 \leq v_{mx} \leq 0.1 \quad (m/sec)\\
-0.1 \leq v_{my} \leq 0.1 \quad (m/sec)\\
-0.1 \leq v_{mz} \leq 0.1 \quad (m/sec)\\
-0.1 \leq v_{m\omega_x} \leq 0.1 \quad (rad/sec)\\
-0.1 \leq v_{m\omega_y} \leq 0.1 \quad (rad/sec)\\
-0.3 \leq v_{m\omega_z} \leq 0.3 \quad (rad/sec)
\end{aligned}. \tag{5.14}$$

To demonstrate the results of this investigation, without the loss of generality, we chose the initial and desired locations such that the robot needs a motion in all the 6-DOFs to reach the desired position. The final time t_f is selected as 10 (sec). The changes to the objective function for different values of the design parameters are shown in Figure 5.5. To be able to show these variations in 3D plot format, the variation of the objective function is shown due to the changes in two parameters at each figure. All available combinations are presented. The variation of the objective function due to changes in v_{mx} and v_{my} are shown in Figure 5.5a. The variation of the objective function due to the changes in $v_{mx} - v_{mz}$ is shown in Figure 5.5b. Because of the similarity in behavior of the system due to change in v_{mx} and v_{my} all the diagrams related to changes in v_{my} are omitted here and one can refer to the figures showing the variations due to the changes in v_{mx}. Moreover, due to the fact that the trajectory planning is decoupled, the orientation never interfere with the positioning. Thus, it is not required to check the convexity of the system due to a combined linear and angular motion. To check for the convexity of the system due to the angular motions the changes in the objective function is introduced due to the changes in $\omega_{mx} - \omega_{my}$ and $\omega_{mx} - \omega_{mz}$. These changes are shown in Figures 5.5c and 5.5d. The convexity of the objective function is clearly demonstrated in the Figures 5.5.



(a) OF_n versus v_{mx} and v_{my}

(b) OF_n versus v_{mx} and v_{mz}

(c) OF_n versus ω_{mx} and ω_{my}

(d) OF_n versus ω_{mx} and ω_{mz}

Figure 5.5: Objective function due to different parameters' changes

5.5 Constrains

One of the main issues in conventional visual servoing is that it does not limit the robot within the system constraints. In addition, by just limiting the system within the constraints the convergence of the system to the target point cannot be guaranteed. The highly coupled nature of visual servoing system could cause the controlling law to take the robot toward and beyond its boundaries while IBVS is attempting to fix the camera's orientation. This can be easily observed in a visual servoing task using a conventional controller. Thus, limiting the system motion like a model predictive controller would do, is not sufficient to stabilize the system [137]. On the other hand, in a trajectory planning algorithm, the generated trajectory could be examined beforehand to guarantee that reaches the target while respecting the constraints. Two main constraints are considered in this research. The first constraint is associated with the robots working space. The second constraint is the robot joint limits. These constraints are discussed in details in the following sections.

It is good to note that, limiting the system to keep the features inside the field of view of the camera is vital to the success of the task in an IBVS conventional visual servoing. The proposed method integrates the equation of motion and predicts the features position at different time moments. Thus, it only requires the initial and the final positions of the features. Consequently, limiting the features inside the field of view is not necessary in this method.

5.5.1 Working Space Constraint

The planned trajectory is feasible only if it is inside the robot working space at all times. Every robot has its own working space. The typical working space of a serial manipulator is a part of sphere with the radius equal to the length of the arms when they are aligned in the same direction. This could be formulated in a polar system as follows,

$$\begin{array}{lcl} X_c = R_c \cos(\theta_c) \cos(\alpha_c) & & 0 < R_c \leq R_{c_{max}} \\ Y_c = R_c \cos(\theta_c) \sin(\alpha_c) & \quad and \quad & \theta_{c_{min}} < \theta_c \leq \theta_{c_{max}} \\ Z_c = R_c \sin(\theta_c) & & \alpha_{c_{min}} < \alpha_c \leq \alpha_{c_{max}} \end{array} , \tag{5.15}$$

where, $\mathbf{P}_c = [X_c, Y_c, Z_c]^{\mathrm{T}}$ and $\mathbf{P}_{p_c} = [R_c, \theta_c, \alpha_c]^{\mathrm{T}}$ are the cameras coordinates in Cartesian and polar systems, $R_{c_{max}}$ is the maximum possible length of the robot's arm, $\theta_{c_{min}}$ and $\theta_{c_{max}}$ are the minimum and maximum angles of the robot's arm about its base X axis, $\alpha_{c_{min}}$ and $\alpha_{c_{max}}$ are the minimum and maximum angles of the robot's arm about its base Z axis.

5.5.2 Joints Space Constraint

Keeping the robot inside the working space is not enough to accomplish a visual servoing task. In addition to work space constraint, it is necessary to make sure the robot respects its joint limits and does not collide with itself. These constraints can be formulated as follows.

$$\mathbf{q}_{min} \leq \mathbf{q} \leq \mathbf{q}_{max}, \tag{5.16}$$

where $\mathbf{q}$ is the robot joint vector and $\mathbf{q}_{min}$ and $\mathbf{q}_{max}$ are defined as the robot's joint limits. The end-effector position is known at all time during the servoing. A function is required to transform the robot's end-effector coordinates to robot joints' value. This function is the inverse kinematic of the robot. The constraint could be written as

$$\mathbf{q}_{min} \leq \mathbf{I}(\mathbf{P_c}) \leq \mathbf{q}_{max}, \tag{5.17}$$

where $\mathbf{I}(\mathbf{P}_c)$ is the inverse kinematic function of the robot.

In order to have a completely convex optimization problem the constraints should also be convex functions. The convexity of these functions are investigated in reference [101].

5.6 Visual Servoing Controller

In the cases where there are some uncertainties in the system model, the generated trajectory locates the features with a small error with respect to the target position. To compensate for such errors, a visual servoing controller is required. To design the controller, the relation between the robots end effector acceleration and the features is required. This relationship can be obtained by taking double time derivatives of both sides equation as follows

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \frac{\ddot{X}}{Z} - \frac{\ddot{Z}X}{Z^2} - 2\frac{\dot{Z}\dot{X}}{Z^2} + 2\frac{\dot{Z}^2X}{Z^3} \\ \frac{\ddot{Y}}{Z} - \frac{\ddot{Z}Y}{Z^2} - 2\frac{\dot{Z}\dot{Y}}{Z^2} + 2\frac{\dot{Z}^2Y}{Z^3} \end{bmatrix}. \tag{5.18}$$

We shall use the well-known kinematic equations (5.19) and (5.20), to find the relationship between the camera motion and the features.

$$\dot{\mathbf{P}} = -\mathbf{v} - \omega \times \mathbf{P}, \tag{5.19}$$

$$\ddot{\mathbf{P}} = -\mathbf{a} - \alpha \times \mathbf{P} + 2\omega \times \mathbf{v} + \omega \times (\omega \times \mathbf{P}), \tag{5.20}$$

where $\mathbf{v}$ and $\mathbf{a}$ are camera's velocity and acceleration vectors, ω and α are camera's angular velocity and acceleration vectors, respectively. Applying equation (5.19) and (5.20) to equation (5.18) and repeating it for all four features, we obtain

$$\ddot{\mathbf{s}}_4 = \mathbf{L}_{a4}\mathbf{A} + \mathbf{L}_{v4}, \tag{5.21}$$

where $\mathbf{A}$ is the cameras acceleration screw which is given as $\mathbf{A} = [a_x\ a_y\ a_z\ \alpha_x\ \alpha_y\ \alpha_z]^{\mathrm{T}}$ and

$$\mathbf{L}_{a4} = \Big[\mathbf{L}_a|_{\mathbf{p}=\mathbf{p}_1}\ \ \dots\ \ \mathbf{L}_a|_{\mathbf{p}=\mathbf{p}_4}\Big]^{\mathrm{T}}, \tag{5.22}$$

$$\mathbf{L}_{v4} = \Big[\mathbf{L}_v|_{\mathbf{p}=\mathbf{p}_1}\ \ \dots\ \ \mathbf{L}_v|_{\mathbf{p}=\mathbf{p}_4}\Big]^{\mathrm{T}}, \tag{5.23}$$

and

$$\mathbf{L}_a = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & x \end{bmatrix}. \tag{5.24}$$

$\mathbf{L}_v$ is obtained by imposing the two last terms of equation (5.19) to equation (5.20) and can be written as

$$\mathbf{L}_v = \begin{bmatrix} \mathbf{V}^{\mathrm{T}}\mathbf{O}_x\mathbf{V} \\ \mathbf{V}^{\mathrm{T}}\mathbf{O}_y\mathbf{V} \end{bmatrix}, \tag{5.25}$$

where $\mathbf{V}$ is the camera's velocity screw which is given as $\mathbf{V} = [v_x\ v_y\ v_z\ \omega_x\ \omega_y\ \omega_z]^{\mathrm{T}}$ and $\mathbf{O}_x$ and $\mathbf{O}_y$ can be calculated from

$$\mathbf{O}_x = \begin{bmatrix} 0 & 0 & \frac{f}{Z^2} & -\frac{y}{Z} & \frac{3x}{2Z} & 0 \\ 0 & 0 & 0 & -\frac{x}{2Z} & 0 & -\frac{f}{2Z} \\ \frac{f}{Z^2} & 0 & \frac{2x}{Z^2} & \frac{2xy}{fZ} & -\frac{f}{2Z}-\frac{2x^2}{Zf} & \frac{y}{Z} \\ -\frac{y}{Z} & -\frac{x}{2Z} & \frac{2xy}{fZ} & x+\frac{2xy^2}{f^2} & -\frac{y}{2}-\frac{2x^2y}{f^2} & \frac{f}{2}-\frac{x^2}{2f}+\frac{y^2}{f} \\ \frac{3x}{2Z} & 0 & -\frac{f}{2Z}-\frac{2x^2}{Zf} & -\frac{y}{2}-\frac{2x^2y}{f^2} & 2x+\frac{2x^3}{f^2} & -\frac{3xy}{2f} \\ 0 & -\frac{f}{2Z} & \frac{y}{Z} & \frac{f}{2}-\frac{x^2}{2f}+\frac{y^2}{f} & -\frac{3xy}{2f} & -x \end{bmatrix} \tag{5.26}$$

and

$$\mathbf{O}_y = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{2y}{Z} & \frac{f}{2Z} \\ 0 & 0 & \frac{f}{Z^2} & -\frac{3y}{2Z} & \frac{x}{Z} & 0 \\ 0 & \frac{f}{Z^2} & \frac{2y}{Z^2} & \frac{f}{2Z}+\frac{y^2}{Zf} & -\frac{xy}{Zf} & -\frac{x}{Z} \\ 0 & -\frac{3y}{2Z} & -\frac{f}{2Z}+\frac{y^2}{Zf} & 2y+\frac{2y^3}{f^2} & -\frac{x}{2}-\frac{2xy^2}{f^2} & -\frac{3xy}{2f} \\ \frac{f}{2Z} & \frac{x}{y} & -\frac{xy}{Zf} & -\frac{x}{2}-\frac{2xy^2}{f^2} & y+\frac{2x^2y}{f^2} & \frac{f}{2}+\frac{x^2}{f}-\frac{y^2}{2f} \\ \frac{f}{2Z} & 0 & -\frac{x}{Z} & \frac{3xy}{2f} & \frac{f}{2}+\frac{x^2}{f}-\frac{y^2}{2f} & -y \end{bmatrix}. \tag{5.27}$$

Defining the controller error as

$$\mathbf{e} = \mathbf{s}_4 - \mathbf{s}_{d4}, \tag{5.28}$$

where $\mathbf{s}_{d4}$ is the desired feature points on the image plane, the following controlling law could eliminate the error due to an exponential decrease given in equation (5.30).

$$\mathbf{A}_c = \mathbf{L}_{a4}^{+}(-\lambda_v \dot{\mathbf{e}} - \lambda_p \mathbf{e} - \mathbf{L}_{v4}), \tag{5.29}$$

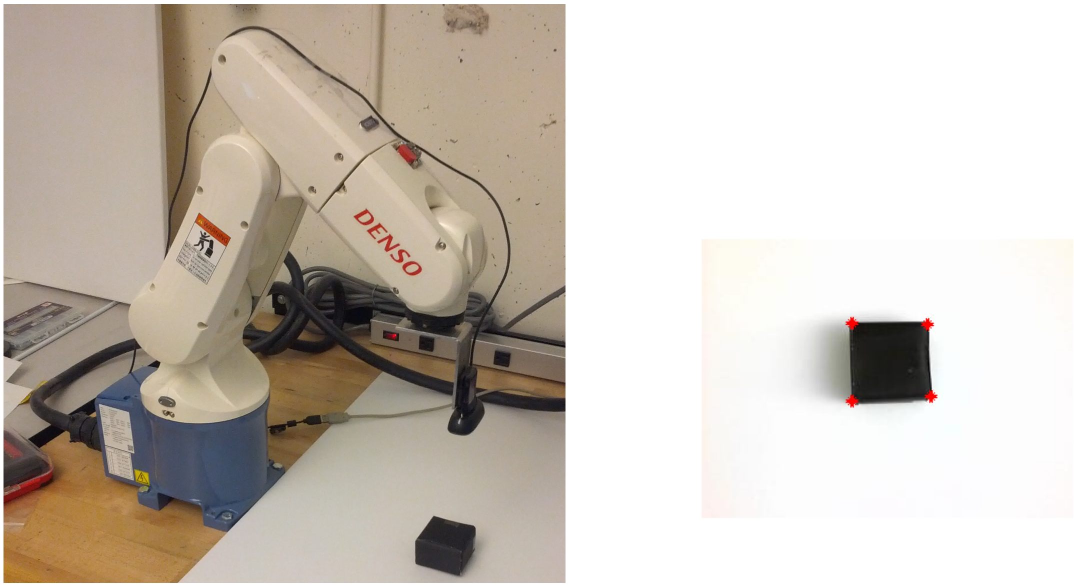
where $\mathbf{A}_c$ is the acceleration command. $\mathbf{L}_{a4}^{+}$ is pseudo inverse of the image Jacobian matrix. λ_v and λ_p are the derivative and proportional gains.

$$\ddot{\mathbf{e}} + \lambda_v \dot{\mathbf{e}} + \lambda_p \mathbf{e} = 0, \tag{5.30}$$

where $\mathbf{e}$, $\dot{\mathbf{e}}$ and $\ddot{\mathbf{e}}$ are the features position and velocity and acceleration errors. The stability of this controller is proven in [138].

5.7 Experimental Results

In this section, the results of the experimental tests of the proposed algorithm on DENSO robot (Figure 5.6) are presented. Full specifications of the experimental setup are described in 2.5. A cubic shape object is used as the target object. Four corners of the top plane of the object is used as the features. Harris algorithm is used to extract the cube corners [139]. Figures 5.6b shows the picture of the cube taken by the camera and the extracted features.



(a) Denso robot in operation

(b) Image of the cubic object taken by the camera on the robot

Figure 5.6: Experimental setup and the object

The object is stationary in the working space. The visual servoing task is finished when the image features match the desired features. Each complete test consists of four stages. First, the depth estimation algorithm moves the end-effector in X_c direction by 5cm to take the stereoscopic image and estimates the depth of the object. Second, using the current image features, desired image features and the initial depth of the object, the trajectory planning algorithm generates the appropriate angular velocity through optimization to reorient the camera to a parallel plane as the object plane. This is done by matching the three last selected features. After that, the positioning trajectory is generated by matching the first three selected features. Due to the nonlinearity of the selected objective function, an interior point algorithm [140] is used to solve the optimization problem. In the third stage, the generated velocity is applied to the robot to take it to the desired position. At the fourth stage, an AIBVS [138] controller is executed to compensate for any difference between the image features and the desired image features caused by the uncertainties in system model. As it is shown in the results, most of the tests may not require the last stage, since the trajectory planning exactly matches the features with the desired ones. The flowchart for the trajectory planning procedure is provided in Figure 5.7. Four different tests with different strategies have been performed to ensure the algorithms validity.

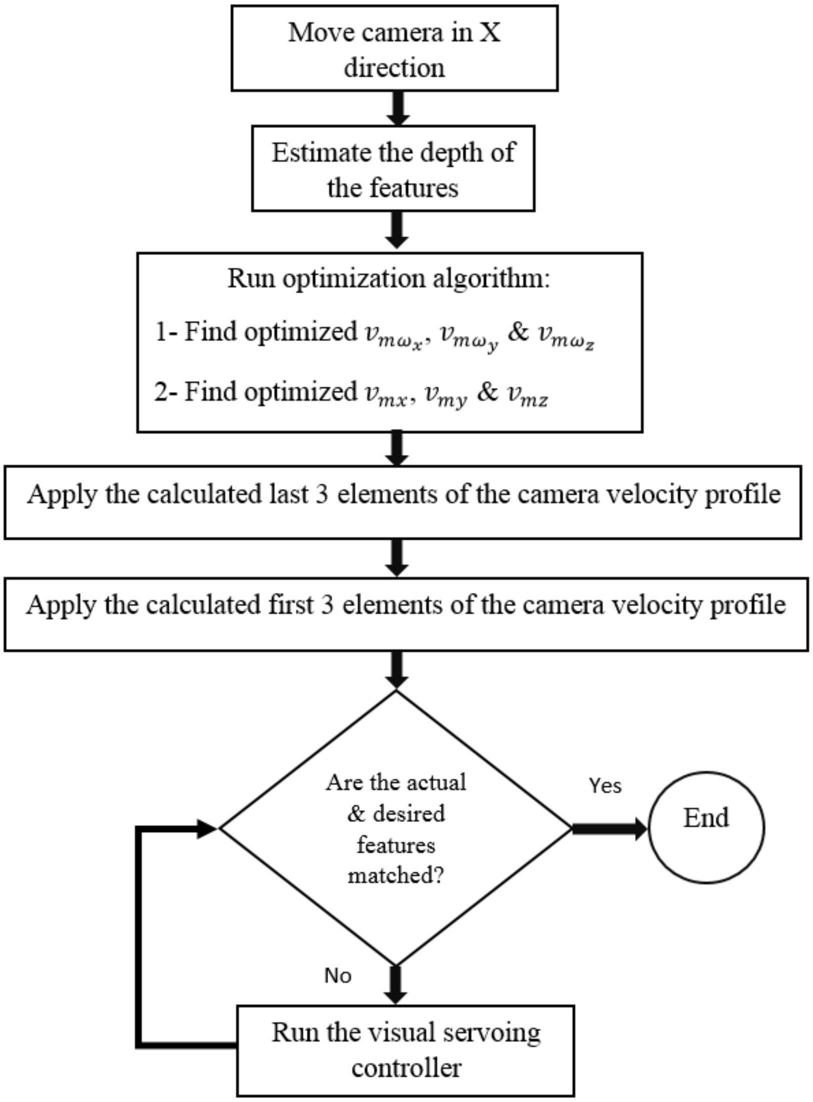

Move camera in X direction
Estimate the depth of the features
Run optimization algorithm:
1- Find optimized $v_{m\omega_x}$, $v_{m\omega_y}$ & $v_{m\omega_z}$
2- Find optimized v_{mx}, v_{my} & v_{mz}
Apply the calculated last 3 elements of the camera velocity profile
Apply the calculated first 3 elements of the camera velocity profile
Are the actual & desired features matched?
Yes
End
No
Run the visual servoing controller

Figure 5.7: Flowchart for the trajectory planning

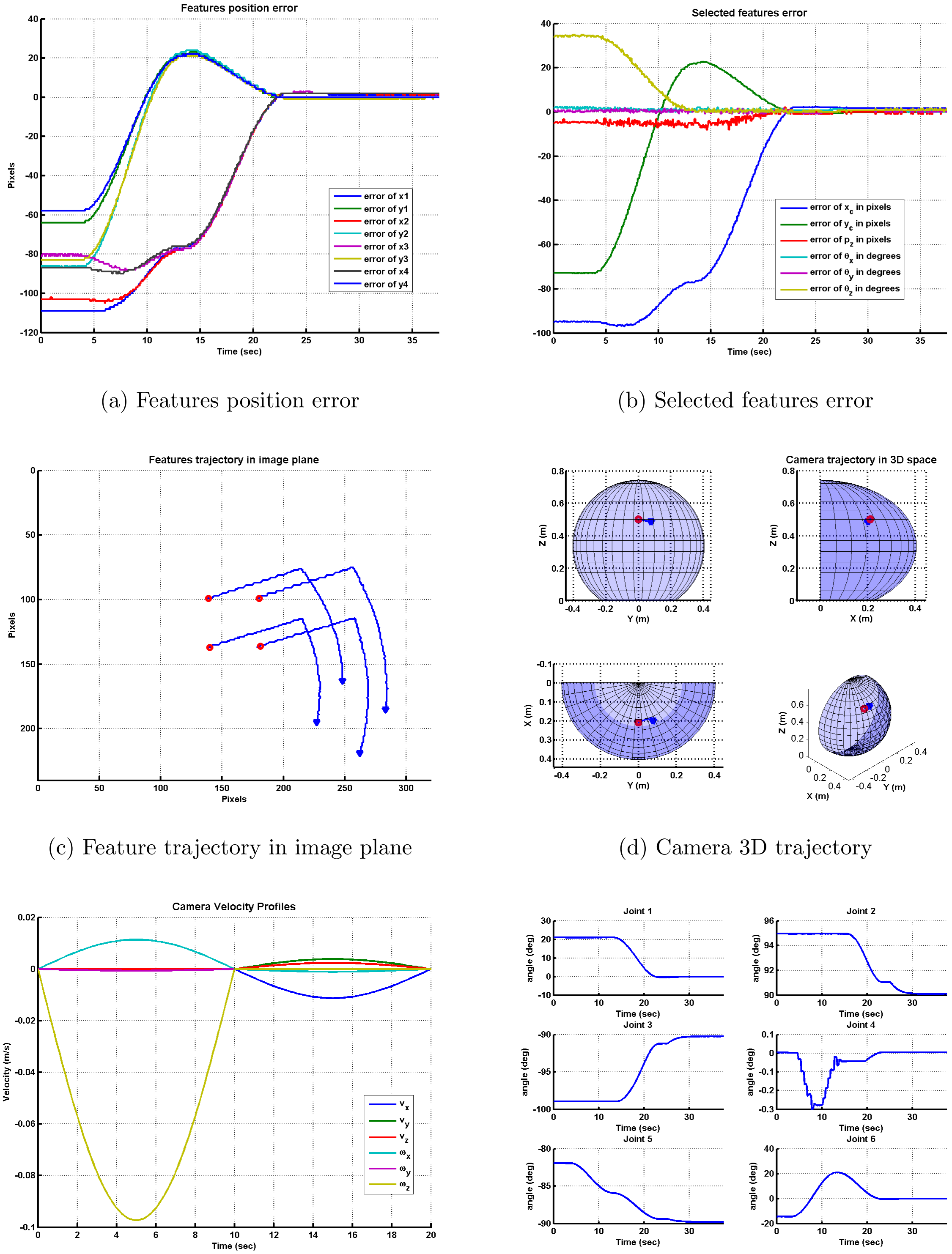
Test 1

In the first test, our aim is to show the performance of the system on performing a relativity simple visual servoing task. The initial and desired locations of the features are given in the Table 5.1.

Table 5.1: Initial(I) and Desired(D) location of feature points in pixel

		Point1		**Point2**		**Point3**		**Point4**	
		(x	y)	(x	y)	(x	y)	(x	y)
Test 1	I	248	163	283	185	262	219	227	195
	D	138	99	179	99	179	136	138	137
Test 2	I	32	106	75	12	242	83	155	181
	D	139	100	179	98	180	135	139	136
Test 3	I	137	99	178	99	179	136	138	137
	D	190	154	129	154	128	98	190	98
Test 4	I	107	210	16	206	26	133	114	137
	D	291	212	203	229	187	154	276	136

The trajectory planning algorithm generates the velocity profiles shown in Figure 5.8e. Applying the velocities to the robot, the robot is taken to the desired position. The first sine cycle is related to the orientation planning and the second part is related to the positioning. The features trajectory in image space and the camera trajectory in 3D space are shown in Figures 5.8c and 5.8d. The half sphere in this figure shows the workspace of the robot. The robot joint angles during the robot motion are shown in Figure 5.8f. Since, the system model is sufficiently accurate, the desired position is reached using the velocity profiles and the fourth stage of the algorithm is not required for this test. In the first stage of the algorithm, the robot moves the camera by $10cm$ in X_c direction and the depth estimation is $0.4m$. The optimization process in this test takes less than a second to complete using a Intel Xeon E31220 3.10GHz CPU.



(a) Features position error

(b) Selected features error

(c) Feature trajectory in image plane

(d) Camera 3D trajectory

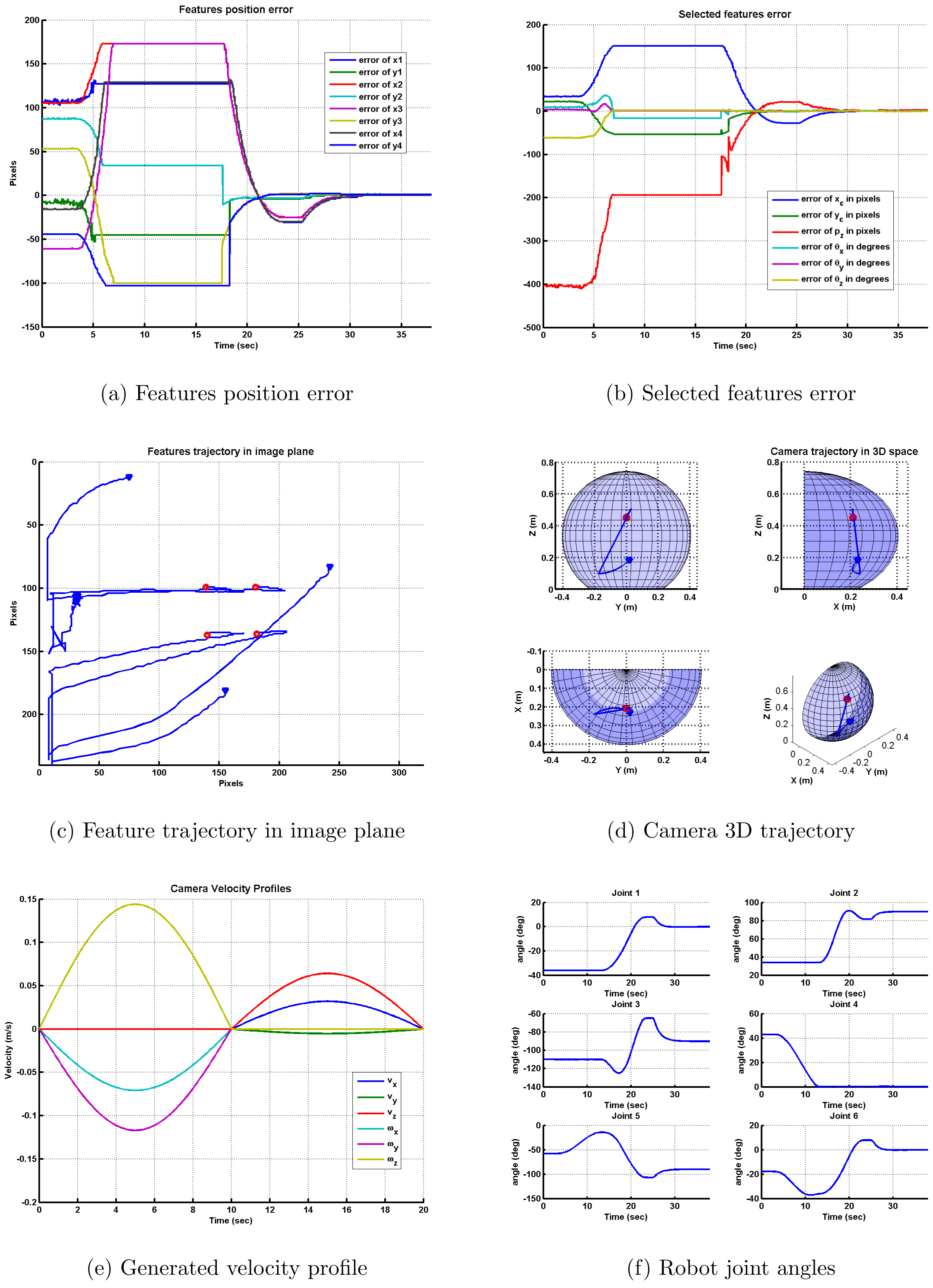
(e) Generated velocity profile

(f) Robot joint angles

Figure 5.8: Results for Test 1

Test 2

In the second test, some of the advantages of the proposed method to IBVS controller are shown. A relatively complicated task is chosen for this matter. The initial and final position of the robot is given in Table 5.1. The results of this test is given in Figures 5.9. The optimization process creates the velocity profile given in Figure 5.9e. The first part of the velocity profile is to orient the camera to be parallel to the object's feature plane. These velocity profiles only moves the three last joints. This cause the features to move out of the FOV. however since this algorithm is an off line planning it only depends on the initial and desired location of the features. Within the algorithm it is assumed that the camera FOV is unlimited. The features eventually return to the real FOV of the camera as the robot completes the created path. The constant lines in the feature error and selected features error in Figures 5.9a and 5.8b are related to the time that the features are out of camera's FOV. It is shown that the task is completed keeping the robot in its workspace. The joint angles are also shown in Figure 5.9f.



(a) Features position error

(b) Selected features error

(c) Feature trajectory in image plane

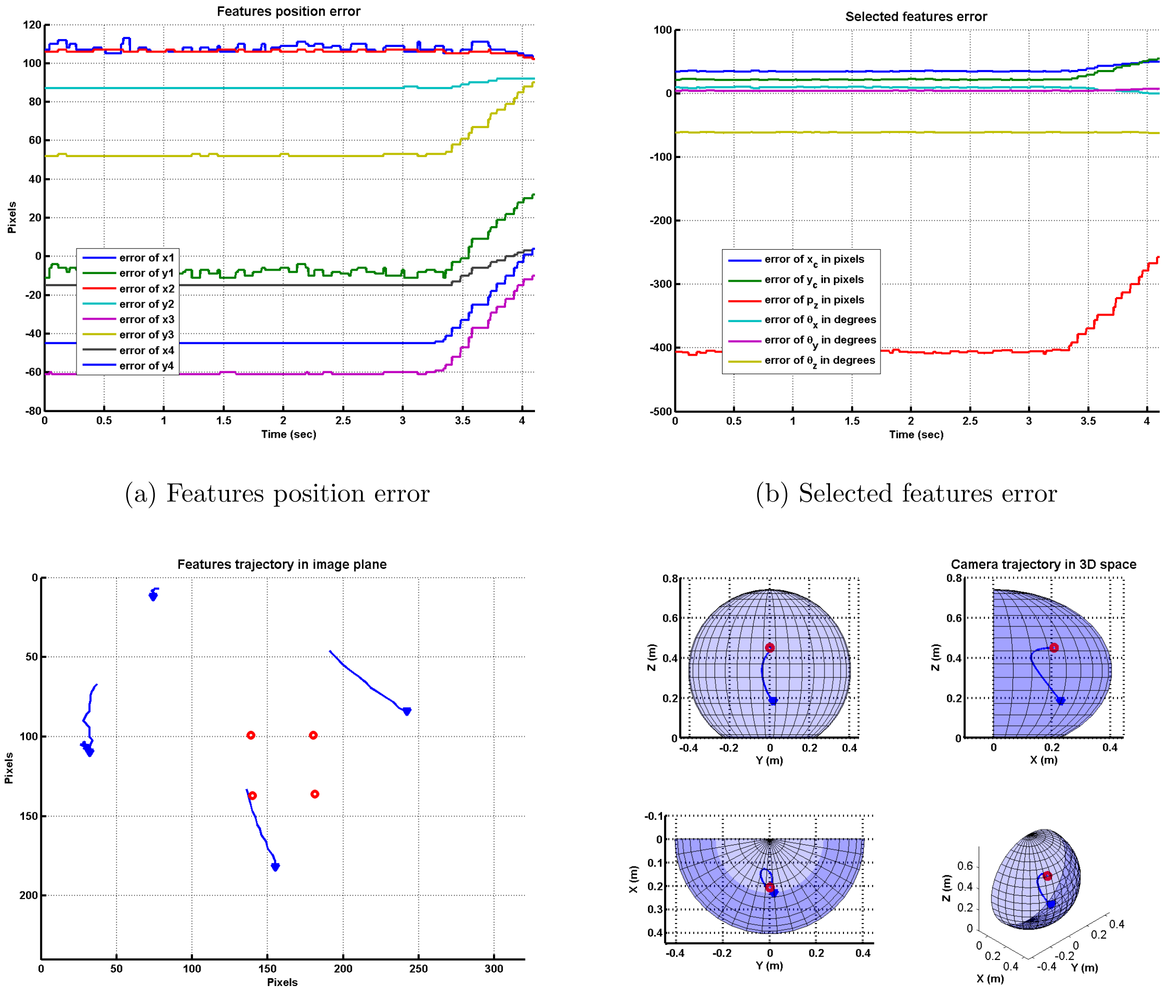
(d) Camera 3D trajectory

(e) Generated velocity profile

(f) Robot joint angles

Figure 5.9: Results for Test 2

The same task is done using an IBVS controller. The results are given in Figure 5.10



(a) Features position error

(b) Selected features error

(c) Feature trajectory in image plane

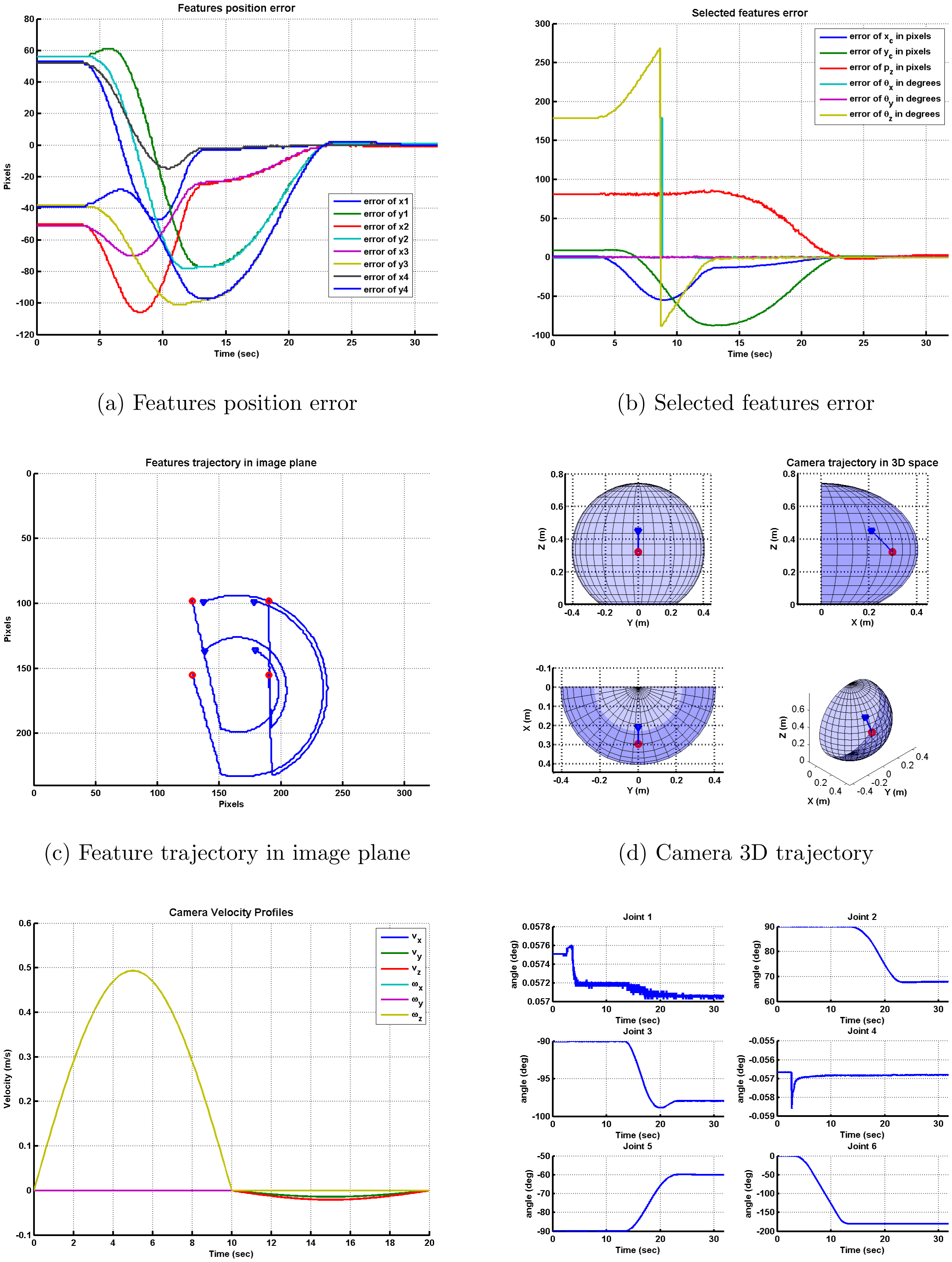
(d) Camera 3D trajectory

Figure 5.10: Results for Test 2 for IBVS

As shown in Figure 5.10c, the rotation required for this task takes the features out of the field of view. The IBVS controller depends on the features position at each instant. As soon as the features run out of the field of view the controller have false data from the features position and it cause the task to fail.

Test 3

For the third test, another common problems of IBVS is investigated using the proposed method. The visual servoing fails when a 180 degrees rotation of the camera is required to reach its desired position [141]. A test is prepared including a 180 degrees rotation in the end effector motion. The initial and desired locations of the feature points are given in Table 5.1. The result of this test is shown in Figure 5.11. The same test is conducted using IBVS controllers. The results are shown in Figures 5.12. The results show that, similar to the previous test, the IBVS controller tries to match the features through the shortest path available which results in a motion of camera in the Z_c direction. This continues until the end-effector reaches its physical limits and the robot stops, as shown in Figure 5.12d.



(a) Features position error

(b) Selected features error

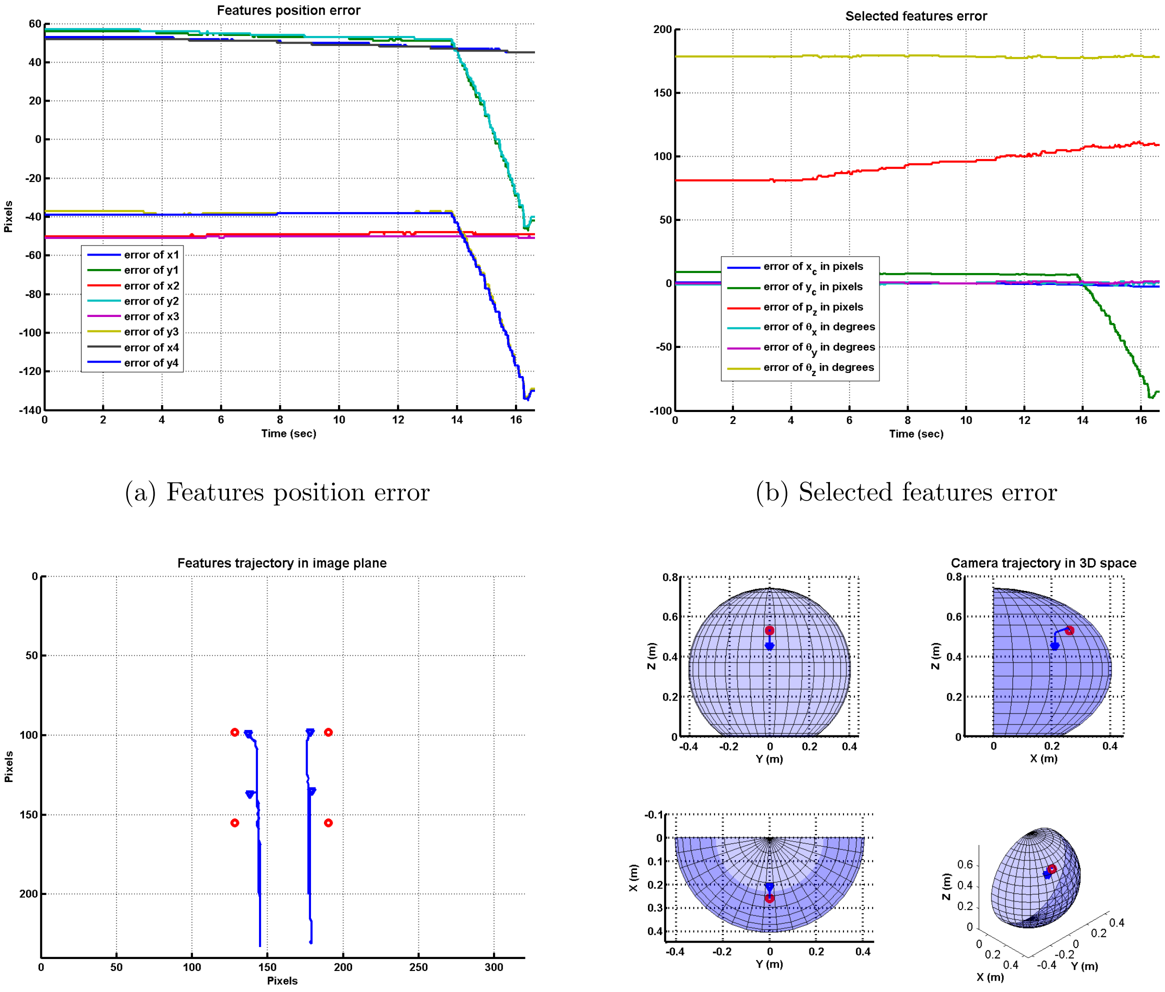
(c) Feature trajectory in image plane

(d) Camera 3D trajectory

(e) Generated velocity profile

(f) Robot joint angles

Figure 5.11: Results for Test 3



(a) Features position error

(b) Selected features error

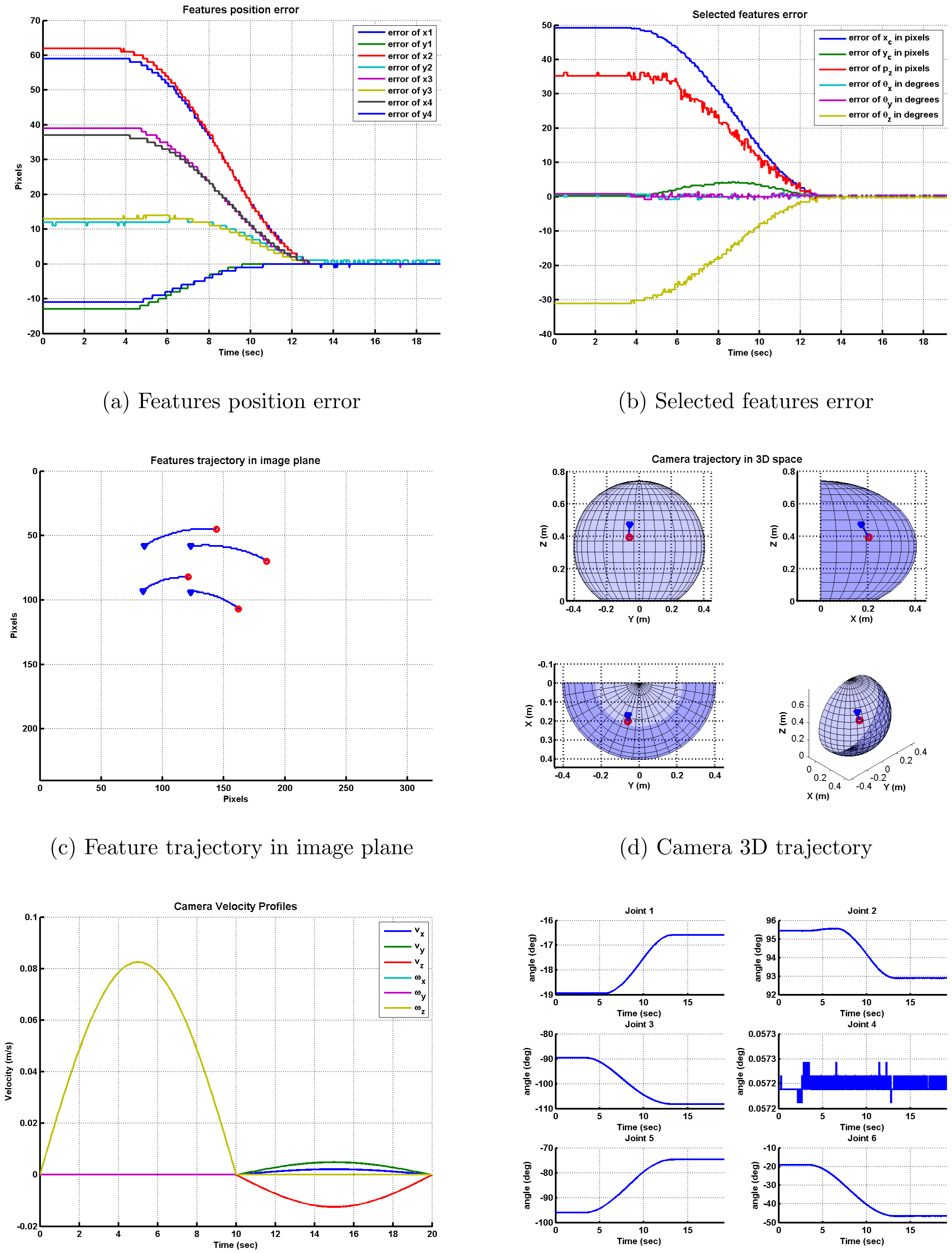
(c) Feature trajectory in image plane

(d) Camera 3D trajectory

Figure 5.12: Results for Test 3 for IBVS

Test4

Another challenges in conventional visual servoing is the local minima problem. In an IBVS controller, the Jacobian matrix is a 8x6 matrix. The inverse of this matrix, which is used to produce the controlling law, is a 8x6 matrix and has two vector of null space. If the features error vector is a factor of these null space vectors the controller generates a zero velocity vector as the controlling command. This cause the system to get stuck in that spot. In the trajectory planning algorithm, the inverse of the Jacobian matrix is not used. consequently, the local minima problem is solved. The next test demonstrates this ability in the proposed algorithm. The initial and desired locations of the feature points are given in Table 5.1. The desired features are chosen so that the vector of feature position error is in the null space of the Jacobian matrix. The results are shown in Figures 5.13. We can see that the proposed algorithm produces a velocity profile to take the robot to the desired position while the IBVS controller produces a zero velocity vector.



(a) Features position error

(b) Selected features error

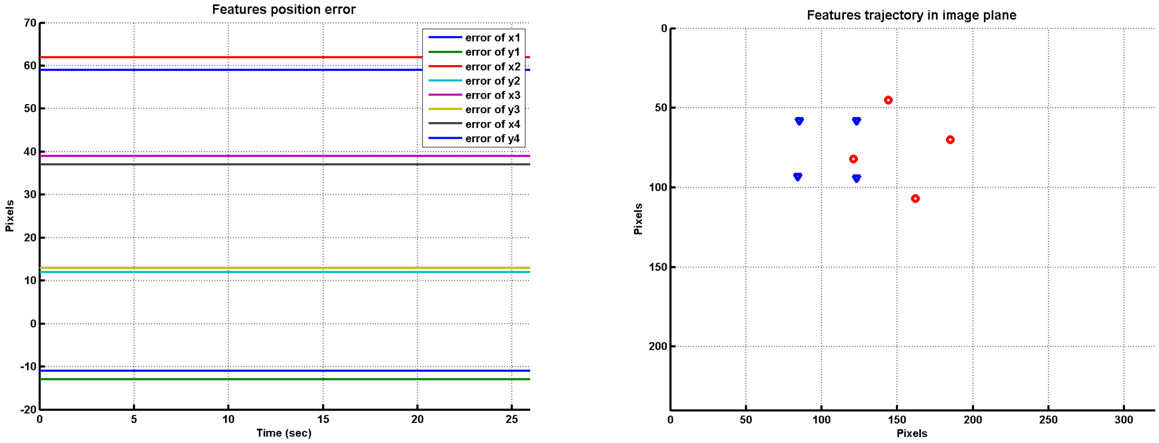
(c) Feature trajectory in image plane

(d) Camera 3D trajectory

(e) Generated velocity profile

(f) Robot joint angles

Figure 5.13: Results for Test 4



(a) Features position error

(b) Selected features error

Figure 5.14: Results for Test 4 using IBVS controller

5.8 Summary

In this chapter, a novel visual servoing technique using optimized trajectory planning is proposed. An optimized trajectory is planned from the initial robot's position to a position where the image features match the desired ones. The trajectory is based on a predefined trajectory which satisfies the system's initial and final conditions. The trajectory parameters are determined through an optimization procedure by minimizing the errors between the image features and the desired ones. In order to speed up the optimization process, four new features are introduced. Using these features, the optimization problem becomes a convex problem. A depth estimation method is proposed to provide the object depth to the trajectory planning algorithm. An AIBVS visual servoing controller is used to compensate for any probable errors appeared in matching the features with the desired ones. Experimental tests validate the proposed method and exhibit its advantages over IBVS controllers. The results show the reliability of the proposed method compared to IBVS techniques. Moreover, the optimization problem is designed so that it could converge to solution in one second. The optimization process time is less than a second in all tests. Therefore, the method could be easily used in an industrial application.

Chapter 6

Conclusion and Future Works

6.1 Summary of the thesis

Integration of vision and robotic systems has increased the dexterity and intelligence of industrial robots. This thesis focuses on image-based visual servoing (IBVS) which uses the features taken by the vision system as a feedback in a robotic system to guide the robot to the desired pose. Researchers have introduced various methods in visual servoing to improve its performance. In this thesis, a series of new methods are proposed to overcome the current shortcomings. The proposed methods aim to increase the robustness of the IBVS to uncertainties and camera limitations and also overcome some of its drawbacks. In the following, the proposed methods are summarized:

1. Adaptive Switch Image-Based Visual Servoing

   An adaptive switch IBVS for an industrial robot with monocular camera in eye-in-hand configuration is proposed. A three stage control scheme is proposed to realize the decoupled rotational and translational movement. The update laws are developed for estimating the camera intrinsic parameters. The designed controller can overcome some of the inherent drawbacks of traditional IBVS and switch IBVS. The proposed method is tested in a 6-DOF robotic system with an eye-in-hand camera installed at the end-effector. The results validate the effectiveness of the proposed adaptive switch method in industrial applications. The experimental results show that response time of this method is much less than that of traditional IBVS and less than that of switch method. Moreover, in the cases where the angle between initial and desired image features is greater than 90°, IBVS normally cannot perform the task while the adaptive switch method performs the task successfully. Especially in the tasks where 180° rotation of the camera around its center is needed, adaptive switch method finishes the task successfully while IBVS fails and the switch IBVS does it with slower response. It is shown that the tracking performance of the proposed method has been improved compared with the switch method and traditional IBVS.

2. Enhanced Switch Image-Based Visual Servoing Dealing with Features Loss

   An enhanced switch IBVS for a 6-DOF industrial robot is proposed. One problem that may occur during IBVS tasks is the features loss or occlusion due to the limited field of view of the camera. An image feature reconstruction algorithm based on the Kalman filter is proposed to handle feature loss during the process of IBVS. The combination of a three-stage switch controller and feature reconstruction algorithm improves the system response speed and tracking performance of IBVS and simultaneously overcomes the problem of feature loss during the task. The proposed method is simulated and then tested on a 6-DOF robotic system with the camera installed in an eye-in-hand configuration. Both simulation and experimental results verifies the efficiency of the method.

3. Enhanced IBVS Controller Using Hybrid PD-SMC Method

   An enhanced IBVS which combines PD control with SMC is presented. The purpose of this approach is to improve the visual servoing performance by taking advantages of PD control and SMC so that the proposed hybrid PD-SMC controller owns good robustness against the disturbance and uncertainties due to the estimated depth. Also it has fast convergence rate. The stability of the enhanced IBVS system is proven by using Lyapunv function method. Simulation and experimental tests demonstrate that the proposed hybrid PD-SMC IBVS excels greatly the classic IBVS controller.

4. Image-Based Visual Servoing Using Trajectory Planning

   A novel visual servoing technique is proposed. This technique is performed by planning a trajectory from the initial robot's position to a position where the image features match the desired ones. The trajectory is based on optimizing a predefined path which satisfies the system's initial and final conditions. The trajectory parameters are identified through an optimization procedure by minimizing the error between the image features and the desired ones. In order to speed up the optimization process, four new features are introduced. Moreover the planning procedure is decoupled to two stages of orientation planning and position planning. This is necessary to have a convex problem. A depth estimation method is proposed to provide the object depth to the trajectory planning algorithm. After performing the velocity profile generated from the trajectory planning algorithm, A visual servoing controller is used to compensate for any probable errors appeared in matching the features with the desired ones. Experimental tests validate the proposed method and exhibits its advantages over IBVS controllers. The results show that in cases where the IBVS controller is unable to complete the visual servoing task, the proposed algorithm is successful.

6.2 Future work

This thesis focuses on introducing new methods of IBVS. The next step can be adopting the proposed techniques in vision and force fusion to be used in applications such as deburring and welding. In this thesis, for simplicity it is assumed that depth of the features are the same. It is useful to develop similar methods to consider the situation the depths of the features are different. Extending the proposed methods to catch moving objects is also another potential work.

In the adaptive switch method, the proposed method can be extended to include simultaneous estimation of the depth parameter Z, robot dynamic parameters along with the camera parameters in the adaptive switch control design. Again all the unknown parameters including depth, camera and dynamic parameters can be collected in a vector similar to $\hat{\theta}$ in (2.16) and the proposed method may be extended to estimate them during the on-line control process. In the case where there are considerable differences in depths of the features, stereo cameras can be used to estimate the depths and a similar control method can be developed for this case.

The enhanced switch IBVS method may be extended to make it more robust to uncertainties such as the depth of features and camera parameters. In addition, the effect of different sampling periods on the performance of the proposed ESIBVS may be investigated.

The developed methods in the thesis are applicable for static objects. In the future work these methods may be modified to be used for moving objects as well.

Bibliography

[1] Grasping objects on conveyor belts. [Online]. Available: https://www.machinedesign.com/robotics/baxter-robot-rest-us

[2] Robotic fruit picking. [Online]. Available: https://strawberryplants.org/strawberry-picking-robot

[3] Guardium ugv 2011. [Online]. Available: https://en.wheelsage.org/unsorted/guardium_ugv%28tm%29/pictures/l22a9w/

[4] da vinci. [Online]. Available: https://www.engadget.com/2009/04/07/video-da-vinci-si-robotic-system-wants-to-be-your-surgeon/

[5] How do drones work and what is drone technology. [Online]. Available: https://www.dronezon.com/learn-about-drones-quadcopters/what-is-drone-technology-or-how-does-drone-technology-work/

[6] V. Lippiello, B. Siciliano, and L. Villani, "Eye-in-hand/eye-to-hand multi-camera visual servoing," in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 5354–5359.

[7] M. Keshmiri, "Image based visual servoing using trajectory planning and augmented visual servoing controller," Ph.D. dissertation, Concordia University, 2014.

[8] K. Hashimoto, "A review on vision-based control of robot manipulators," *Advanced Robotics*, vol. 17, no. 10, pp. 969–991, 2003.

[9] S. Hutchinson, G. D. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[10] P. I. Corke and S. A. Hutchinson, “A new partitioned approach to image-based visual servo control,” *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 507–515, 2001.

[11] W.-F. Xie, Z. Li, X.-W. Tu, and C. Perron, “Switching control of image-based visual servoing with laser pointer in robotic manufacturing systems,” *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 520–529, 2008.

[12] I. Siradjuddin, L. Behera, T. M. McGinnity, and S. Coleman, “Image-based visual servoing of a 7-dof robot manipulator using an adaptive distributed fuzzy pd controller,” *IEEE/ASME Transactions On Mechatronics*, vol. 19, no. 2, pp. 512–523, 2013.

[13] D.-J. Kim, R. Lovelett, and A. Behal, “Eye-in-hand stereo visual servoing of an assistive robot arm in unstructured environments,” in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2326–2331.

[14] B. Tamadazte, N. L.-F. Piat, and E. Marchand, “A direct visual servoing scheme for automatic nanopositioning,” *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 4, pp. 728–736, 2011.

[15] J. Wang and H. Cho, “Micropeg and hole alignment using image moments based visual servoing method,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 3, pp. 1286–1294, 2008.

[16] C. D. Onal and M. Sitti, “Visual servoing-based autonomous 2-d manipulation of microparticles using a nanoprobe,” *IEEE Transactions on control systems technology*, vol. 15, no. 5, pp. 842–852, 2007.

[17] Y. Fang, X. Liu, and X. Zhang, “Adaptive active visual servoing of nonholonomic mobile robots,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 486–497, 2011.

[18] X. Zhang, Y. Fang, and X. Liu, “Motion-estimation-based visual servoing of nonholonomic mobile robots,” *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1167–1175, 2011.

[19] R. Mebarki, A. Krupa, and F. Chaumette, "2-d ultrasound probe complete guidance by visual servoing using image moments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 296–306, 2010.

[20] M.-A. Vitrani, G. Morel, and T. Ortmaier, "Automatic guidance of a surgical instrument with ultrasound based visual servoing," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation.* IEEE, 2005, pp. 508–513.

[21] A. Krupa, C. Doignon, J. Gangloff, and M. De Mathelin, "Combined image-based and depth visual servoing applied to robotized laparoscopic surgery," in *IEEE/RSJ international conference on intelligent robots and systems*, vol. 1. IEEE, 2002, pp. 323–329.

[22] T. Osa, C. Staub, and A. Knoll, "Framework of automatic robot surgery system using visual servoing," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2010, pp. 1837–1842.

[23] L. Coutard, F. Chaumette, and J.-M. Pflimlin, "Automatic landing on aircraft carrier by visual servoing," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2011, pp. 2843–2848.

[24] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a vtol uav on a moving platform using image-based visual servoing," in *2012 IEEE international conference on robotics and automation.* IEEE, 2012, pp. 971–976.

[25] N. Metni and T. Hamel, "A uav for bridge inspection: Visual servoing control law with orientation limits," *Automation in construction*, vol. 17, no. 1, pp. 3–10, 2007.

[26] P. Rudol, M. Wzorek, G. Conte, and P. Doherty, "Micro unmanned aerial vehicle visual servoing for cooperative indoor exploration," in *2008 IEEE Aerospace Conference.* IEEE, 2008, pp. 1–10.

[27] W. Liang, W. Gao, S. Chen, and K. K. Tan, "Stabilization for an ear surgical device using force feedback and vision-based motion compensation," in *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on.* IEEE, 2014, pp. 943–948.

[28] H. Huang, D. Sun, J. K. Mills, and S. H. Cheng, "Integrated vision and force control in suspended cell injection system: Towards automatic batch biomanipulation," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3413–3418.

[29] P. Hebert, N. Hudson, J. Ma, and J. Burdick, "Fusion of stereo vision, force-torque, and joint sensors for estimation of in-hand object location," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5935–5941.

[30] J. Pomares Baeza, F. Torres Medina, and P. Gil Vázquez, "Visual servoing and force control fusion for complex insertion tasks," 2007.

[31] S. Kuhn, T. Gecks, and D. Henrich, "Velocity control for safe robot guidance based on fused vision and force/torque data," in *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*. IEEE, 2006, pp. 485–492.

[32] M. Bdiwi, A. Kolker, J. Suchỳ, and A. Winkler, "Automated assistance robot system for transferring model-free objects from/to human hand using vision/force control," in *Social Robotics*. Springer, 2013, pp. 40–53.

[33] Y. Shirai and H. Inoue, "Guiding a robot by visual feedback in assembling tasks," *Pattern recognition*, vol. 5, no. 2, pp. 99–108, 1973.

[34] J. Hill, "Real time control of a robot with a mobile camera," in *9th Int. Symp. on Industrial Robots, 1979*, 1979, pp. 233–246.

[35] E. Malis, "Vision-based control invariant to camera intrinsic parameters: stability analysis and path tracking," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 217–222.

[36] H. Sutanto, R. Sharma, and V. Varma, "Image based autodocking without calibration," in *Proceedings of International Conference on Robotics and Automation*, vol. 2. IEEE, 1997, pp. 974–979.

[37] J. Qian and J. Su, "Online estimation of image jacobian matrix by kalman-bucy filter

for uncalibrated stereo vision feedback," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 562–567.

[38] D. Kragic, H. I. Christensen *et al.*, "Survey on visual servoing for manipulation," *Computational Vision and Active Perception Laboratory, Fiskartorpsv*, vol. 15, p. 2002, 2002.

[39] O.-E. Ng and V. Ganapathy, "A novel modular framework for stereo vision," in *2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2009, pp. 857–862.

[40] E. Malis, F. Chaumette, and S. Boudet, "Multi-cameras visual servoing," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 4. IEEE, 2000, pp. 3183–3188.

[41] H. Li, M. Jin, and L. Zou, "A new binocular stereo visual servoing model," in *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 1. IEEE, 2008, pp. 461–465.

[42] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Transactions on Robotics*, vol. 20, no. 4, pp. 713–723, 2004.

[43] G. C. Buttazzo, B. Allotta, and F. P. Fanizza, "Mousebuster: A robot for real-time catching," *IEEE Control Systems Magazine*, vol. 14, no. 1, pp. 49–56, 1994.

[44] D. G. Fernandes and P. U. Lima, "A testbed for robotic visual servoing and catching of moving objects," in *1998 IEEE International Conference on Electronics, Circuits and Systems. Surfing the Waves of Science and Technology (Cat. No. 98EX196)*, vol. 2. IEEE, 1998, pp. 475–478.

[45] A. Namiki and M. Ishikawa, "Vision-based online trajectory generation and its application to catching," in *Control Problems in Robotics*. Springer, 2003, pp. 249–264.

[46] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.

[47] M. Keshmiri, M. Keshmiri, and A. Mohebbi, "Augmented online point to point trajectory planning, a new approach in catching a moving object by a manipulator," in *IEEE ICCA 2010*. IEEE, 2010, pp. 1349–1354.

[48] R. A. Jarvis, "A perspective on range finding techniques for computer vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 2, pp. 122–139, 1983.

[49] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine,*, vol. 13, no. 4, pp. 82–90, 2006.

[50] A. Ghanbari, W. Wang, C. E. Hann, J. G. Chase, and X. Chen, "Cell image recognition and visual servo control for automated cell injection," in *2009 4th International Conference on Autonomous Robots and Agents*. IEEE, 2009, pp. 92–96.

[51] Y. Zhao, W.-F. Xie, and S. Liu, "Image-based visual servoing using improved image moments in 6-dof robot systems," *International Journal of Control, Automation, and Systems*, vol. 11, no. 3, pp. 586–596, Feb. 2013.

[52] T. Banlue, P. Sooraksa, and S. Noppanakeepong, "A practical position-based visual servo design and implementation for automated fault insertion test," *International Journal of Control, Automation, and Systems*, vol. 12, no. 5, pp. 1090–1101, May. 2014.

[53] K. Hashimoto, "A review on vision-based control of robot manipulators," *Advanced Robotics*, vol. 17, no. 10, pp. 969–991, 2003.

[54] P. Corke, *Robotics, Vision and Control*. Springer, 2011.

[55] J. k. Kim, D. w. Kim, S. j. Choi, and S. c. Won, "Image-based visual servoing using sliding mode control," in *2006 SICE-ICASE International Joint Conference*, Oct. 2006, pp. 4996–5001.

[56] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*. Springer, 1998, pp. 66–78.

[57] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, Dec. 2006.

[58] M. Keshmiri, W. F. Xie, and A. Mohebbi, “Augmented image-based visual servoing of a manipulator using acceleration command,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5444–5452, Oct. 2014.

[59] Y. Fang, X. Liu, and X. Zhang, “Adaptive active visual servoing of nonholonomic mobile robots,” *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 486–493, Jan. 2012.

[60] P. Corke and S. Hutchinson, “A new partitioned approach to image-based visual servo control,” *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 4, pp. 507–515, 2001.

[61] F. Chaumette, “Image moments: a general and useful set of features for visual servoing,” *Robotics, IEEE Transactions on*, vol. 20, no. 4, pp. 713–723, 2004.

[62] O. Tahri and F. Chaumette, “Point-based and region-based image moments for visual servoing of planar objects,” *Robotics, IEEE Transactions on*, vol. 21, no. 6, pp. 1116–1127, 2005.

[63] G. Allibert, E. Courtial, and F. Chaumette, “Predictive control for constrained image-based visual servoing,” *Robotics, IEEE Transactions on*, vol. 26, no. 5, pp. 933–939, Oct. 2010.

[64] M. Keshmiri, W.-F. Xie, and A. Mohebbi, “Augmented image-based visual servoing of a manipulator using acceleration command,” *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5444–5452, 2014.

[65] N. R. Gans and S. A. Hutchinson, “Stable visual servoing through hybrid switched-system control,” *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 530–540, 2007.

[66] E. Malis, F. Chaumette, and S. Boudet, “21/2d visual servoing,” *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[67] S. Li, A. Ghasemi, W. Xie, and Y. Gao, “An enhanced ibvs controller of a 6 dof manipulator using hybrid pd-smc method,” *International Journal of Control, Automation and Systems*, vol. 16, no. 2, pp. 844–855, Apr 2018. [Online]. Available: https://doi.org/10.1007/s12555-016-0720-4

[68] M. Keshmiri, W.-F. Xie, and A. Ghasemi, “Visual servoing using an optimized trajectory planning technique for a 4 dofs robotic manipulator,” *International Journal of Control, Automation and Systems*, vol. 15, no. 3, pp. 1362–1373, 2017.

[69] Y. Zhao, W. F. Xie, and S. Liu, “Image-based visual servoing using improved image moments in 6-DOF robot systems,” *International Journal of Control, Automation and Systems*, vol. 11, no. 3, pp. 586–596, 2013.

[70] D. Zheng, H. Wang, W. Chen, and Y. Wang, “Planning and Tracking in Image Space for Image-Based Visual Servoing of a Quadrotor,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3376–3385, 2018.

[71] M. Keshmiri and W.-F. Xie, “Image-based visual servoing using an optimized trajectory planning technique,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 359–370, 2017.

[72] R. Kelly, “Robust asymptotically stable visual servoing of planar robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 759–766, 1996.

[73] Y. Shen, G. Xiang, Y.-H. Liu, and K. Li, “Uncalibrated visual servoing of planar robots,” in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'02)*, vol. 1. Washington, USA, 2002, pp. 580–585.

[74] J. M. Sebastián, L. Pari, L. Angel, and A. Traslosheros, “Uncalibrated visual servoing using the fundamental matrix,” *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 1–10, 2009.

[75] G. Dong and Z. H. Zhu, “Kinematics-based incremental visual servo for robotic capture of non-cooperative target,” *Robotics and Autonomous Systems*, vol. 112, pp. 221–228, 2019.

[76] C. C. Cheah, S. P. Hou, Y. Zhao, and J.-J. E. Slotine, “Adaptive vision and force tracking control for robots with constraint uncertainty,” *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 3, pp. 389–399, 2010.

[77] H. Wang, “Adaptive visual tracking for robotic systems without image-space velocity measurement,” *Automatica*, vol. 55, pp. 294–301, 2015.

[78] C. Cai, E. Dean-León, D. Mendoza, N. Somani, and A. Knoll, “Uncalibrated 3d stereo image-based dynamic visual servoing for robot manipulators,” in *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Tokyo, Japan, 2013, pp. 63–70.

[79] X. Liang, H. Wang, Y.-H. Liu, W. Chen, and J. Zhao, “A unified design method for adaptive visual tracking control of robots with eye-in-hand/fixed camera configuration,” *Automatica*, vol. 59, pp. 97–105, 2015.

[80] D. Folio and V. Cadenat, “A controller to avoid both occlusions and obstacles during a vision-based navigation task in a cluttered environment,” in *Proceedings of the 44th IEEE Conference on Decision and Control.* IEEE, 2005, pp. 3898–3903.

[81] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, “Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 796–803, 2018.

[82] T. Shen and G. Chesi, “Visual servoing path planning for cameras obeying the unified model,” *Advanced Robotics*, vol. 26, no. 8-9, pp. 843–860, 2012.

[83] M. Kazemi, K. K. Gupta, and M. Mehrandezh, “Randomized kinodynamic planning for robust visual servoing,” *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1197–1211, 2013.

[84] T. Murao, T. Yamada, and M. Fujita, “Predictive visual feedback control with eye-in-hand system via stabilizing receding horizon approach,” in *Decision and Control, 2006 45th IEEE Conference on.* IEEE, 2006, pp. 1758–1763.

[85] M. Sauvée, P. Poignet, E. Dombre, and E. Courtial, “Image based visual servoing through nonlinear model predictive control,” in *Decision and Control, 2006 45th IEEE Conference on.* IEEE, 2006, pp. 1776–1781.

[86] C. Lazar, A. Burlacu, and C. Copot, “Predictive control architecture for visual servoing of robot manipulators,” in *IFAC World Congress, Milano (Italy)*, 2011, pp. 9464–9469.

[87] A. Hajiloo, M. Keshmiri, W.-F. Xie, and T.-T. Wang, “Robust online model predictive control for a constrained image-based visual servoing,” *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2242–2250, 2016.

[88] A. Assa and F. Janabi-Sharifi, "Robust model predictive control for visual servoing," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2014, pp. 2715–2720.

[89] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, Oct. 2010.

[90] N. García-Aracil, E. Malis, R. Aracil-Santonja, and C. Pérez-Vidal, "Continuous visual servoing despite the changes of visibility in image features," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1214–1220, 2005.

[91] N. Cazy, P.-B. Wieber, P. R. Giordano, and F. Chaumette, "Visual servoing when visual information is missing: Experimental comparison of visual feature prediction schemes," in *ICRA'15-IEEE International Conference on Robotics and Automation.* IEEE, 2015, pp. 6031–6036.

[92] W. Wroblewski, "Implementation of a model predictive control algorithm for a 6dof manipulator - simulation results," in *Proceedings of the Fourth International Workshop on Robot Motion and Control*, Jun. 2004, pp. 209–212.

[93] E. C. G. Allibert and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, Oct. 2010.

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

[95] M. Asad, A. I. Bhatti, S. Iqbal, and Y. Asfia, "A smooth integral sliding mode controller and disturbance estimator design," *International Journal of Control, Automation, and Systems*, vol. 13, no. 6, pp. 1326–1336, Jan. 2015.

[96] Q. H. Ngo, N. P. Nguyen, C. N. Nguyen, T. H. Tran, and K.-S. Hong, "Fuzzy sliding mode control of container cranes," *International Journal of Control, Automation, and Systems*, vol. 13, no. 2, pp. 419–425, Aug. 2015.

[97] T. Yüksel, "Ibvs with fuzzy sliding mode for robot manipulators," in *Recent Advances in Sliding Modes (RASM), 2015 International Workshop on*, Apr. 2015, pp. 1–6.

[98] M. Parsapour, S. RayatDoost, and H. D. Taghirad, "Position based sliding mode control for visual servoing system," in *Robotics and Mechatronics (ICRoM), 2013 First RSI/ISM International Conference on*, Feb. 2013, pp. 337–342.

[99] F. Li and H. Xie, "Sliding mode variable structure control for visual servoing system," *International Journal of Automation and Computing*, vol. 7, no. 3, pp. 317–323, Aug. 2010.

[100] J. M. Acob, V. Pano, and P. R. Ouyang, "Hybrid pd sliding mode control of a two degree-of-freedom parallel robotic manipulator," in *2013 10th IEEE International Conference on Control and Automation (ICCA)*, Jun. 2013, pp. 1760–1765.

[101] G. Chesi, "Visual servoing path planning via homogeneous forms and lmi optimizations," *Robotics, IEEE Transactions on*, vol. 25, no. 2, pp. 281–291, 2009.

[102] G. Chesi and Y. Hung, "Global path-planning for constrained and optimal visual servoing," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 1050–1060, 2007.

[103] M. Keshmiri, M. Keshmiri, and A. Mohebbi, "Augmented online point to point trajectory planning, a new approach in catching a moving object by a manipulator," in *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, 2010, pp. 1349–1354.

[104] M. Keshmiri and M. Keshmiri, "Performance comparison of various navigation guidance methods in interception of a moving object by a serial manipulator considering its kinematic and dynamic limits," in *Methods and Models in Automation and Robotics (MMAR), 2010 15th International Conference on*, 2010, pp. 212–217.

[105] M. Keshmiri and W. F. Xie, "Catching moving objects using a navigation guidance technique in a robotic visual servoing system," in *2013 American Control Conference*. IEEE, 2013, pp. 6302–6307.

[106] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *Robotics and Automation, IEEE Transactions on*, vol. 18, no. 4, pp. 534–549, 2002.

[107] L. Deng, F. Janabi-Sharifi, and W. J. Wilson, "Hybrid motion control and planning strategies for visual servoing," *Industrial Electronics, IEEE Transactions on*, vol. 52, no. 4, pp. 1024–1040, 2005.

[108] S. Masoud and A. Masoud, “Motion planning in the presence of directional and regional avoidance constraints using nonlinear, anisotropic, harmonic potential fields: a physical metaphor,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 32, no. 6, pp. 705–723, 2002.

[109] A. A. Masoud and M. M. Bayoumi, “Intercepting a maneuvering target in a multidimensional stationary environment using a wave equation potential field strategy,” in *Proceedings of 1994 9th IEEE International Symposium on Intelligent Control*. IEEE, 1994, pp. 243–248.

[110] K. Hosoda, K. Sakamoto, and M. Asada, “Trajectory generation for obstacle avoidance of uncalibrated stereo visual servoing without 3d reconstruction,” *Journal of the Robotics Society of Japan*, vol. 15, no. 2, pp. 290–295, 1997.

[111] J. S. Park and M.-J. Chung, “Path planning with uncalibrated stereo rig for image-based visual servoing under large pose discrepancy,” *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 2, pp. 250–258, Apr. 2003.

[112] M. Kazemi, M. Mehrandezh, and K. Gupta, “Kinodynamic planning for visual servoing,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 2478–2484.

[113] M. Kazemi, K. Gupta, and M. Mehrandezh, “Global path planning for robust visual servoing in complex environments,” in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 326–332.

[114] A. Ghasemi, P. Li, and W.-F. Xie, “Adaptive switch image-based visual servoing for industrial robots,” *International Journal of Control, Automation and Systems*. [Online]. Available: https://doi.org/10.1007/s12555-018-0753-y

[115] A. Ghasemi, P. Li, W.-F. Xie, and W. Tian, “Enhanced switch image-based visual servoing dealing with featuresloss,” *Electronics*, vol. 8, no. 8, p. 903, 2019.

[116] S. Li, A. Ghasemi, W.-F. Xie, and Y. Gao, “An enhanced ibvs controller of a 6dof manipulator using hybrid pd-smc method,” *International Journal of Control, Automation and Systems*, vol. 16, no. 2, pp. 844–855, 2018.

[117] S. Li, A. Ghasemi, W.-F. Xie, and Y. Gao, "Sliding mode control (smc) of image-based visual servoing for a 6dof manipulator," in *Recent Developments in Sliding Mode Control Theory and Applications*. InTech, 2017.

[118] A. Ghasemi and W.-F. Xie, "Decoupled image-based visual servoing for robotic manufacturing systems using gain scheduled switch control," in *Proceedings of 2017 International Conference on Advanced Mechatronic Systems (ICAMechS)*. Xiamen, China, 2017, pp. 94–99.

[119] A. Ghasemi and W.-F. Xie, "Adaptive image-based visual servoing of 6 dof robots using switch approach," in *2018 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2018, pp. 1210–1215.

[120] P. Li, A. Ghasemi, W. Xie, and W. Tian, "Visual closed-loop dynamic model identification of parallel robots based on optical cmm sensor," *Electronics*, vol. 8, no. 8, p. 836, 2019.

[121] W.-F. Xie, Z. Li, X.-W. Tu, and C. Perron, "Switching control of image-based visual servoing with laser pointer in robotic manufacturing systems," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 2, pp. 520–529, 2009.

[122] S. Arimoto, *Control Theory of Non-Linear Mechanical Systems: A Passivity-Based and Circuit-Theoretic Approach (Oxford Engineering Science Series)*. Oxford University Press, 1996.

[123] H. K. Khalil and J. Grizzle, "Nonlinear systems, vol. 3," *Prentice hall Upper Saddle River*, 2002.

[124] *Quanser 6-Axis Articulated Robot*, Markham, ON, Canada: Quanser/Denso Specialty Plants, 2011.

[125] 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, 1999.

[126] M. Keshmiri and W.-F. Xie, "Image-based visual servoing using an optimized trajectory planning technique," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 359–370, 2016.

[127] J.-L. Chang, "Sliding mode control design for mismatched uncertain systems using output feedback," *International Journal of Control, Automation, and Systems*, vol. 14, no. 2, pp. 579–586, Apr. 2016.

[128] T. Fujio, H. Shibasaki, R. Tanaka, T. Murakami, and Y. Ishida, "Sliding mode control based on a modified linear control input," *International Journal of Control, Automation, and Systems*, vol. 14, no. 1, pp. 115–127, Mar. 2016.

[129] J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed., Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[130] P. I. Corke and B. Armstrong-Hlouvry, "A meta-study of puma 560 dynamics: A critical appraisal of literature data," *Robotica*, vol. 13, no. 03, pp. 253–258, 1995.

[131] S. Zhang, C. Wang, and S.-C. Chan, "A new high resolution depth map estimation system using stereo vision and kinect depth sensing," *Journal of Signal Processing Systems*, vol. 79, no. 1, pp. 19–31, 2015.

[132] C. Strecha and L. Van Gool, "Motion—stereo integration for depth estimation," in *European Conference on Computer Vision.* Springer, 2002, pp. 170–185.

[133] M. Ehsani, F.-Y. Wang, and G. L. Brosch, *Transportation Technologies for Sustainability.* Springer, 2013.

[134] J. Arora, *Introduction to Optimum Design.* Elsevier Science, 2004.

[135] J. W. Chinneck, "Discovering the characteristics of mathematical programs via sampling," *Optimization Methods and Software*, vol. 17, no. 2, pp. 319–352, 2002.

[136] H. D. Scolnik and M. J. Gambini, "A new method to compute second derivatives," *Journal of Computer Science & Technology*, vol. 1, 2001.

[137] G. Allibert, E. Courtial, and F. Chaumette, "Visual servoing via nonlinear predictive control," in *Visual Servoing via Advanced Numerical Methods.* Springer, 2010, pp. 375–393.

[138] M. Keshmiri, W. XIE, and A. Mohebbi, "Augmented image based visual servoing of a manipulator using acceleration command," vol. 61, no. 9, Sep. 2014.

[139] C. Harris and M. Stephens, “A combined corner and edge detector,” in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[140] R. Byrd, M. Hribar, and J. Nocedal, “An interior point algorithm for large-scale nonlinear programming,” *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 877–900, 1999.

[141] F. Chaumette, “Potential problems of stability and convergence in image-based and position-based visual servoing,” in *The Confluence of Vision and Control.* LNCIS Series, No 237, Springer-Verlag, pp. 66–78.

Appendix

The relationships between Δ_{min} , Δ_{max} with the bounds of depth are illustrated as follows. To simplify the case, we only consider the translational motions of the robot since they are most affected by the uncertainties of the depth. Thus we have:

$$\begin{bmatrix} \dot{x_1} \\ \dot{y_1} \\ \vdots \\ \dot{y_4} \end{bmatrix} = \hat{L}_{v4T} \begin{bmatrix} v_{cx} \\ v_{cy} \\ v_{cz} \end{bmatrix} \tag{A.1}$$

where

$$\hat{L}_{v4T} = \frac{1}{\hat{Z}} \begin{bmatrix} -f & 0 & x_1 \\ 0 & -f & y_1 \\ \vdots & \vdots & \vdots \\ 0 & -f & y_4 \end{bmatrix} = \frac{1}{\hat{Z}} \hat{L}_{v4T1} \tag{A.2}$$

and

$$\hat{L}_{v4T1} = \begin{bmatrix} -f & 0 & x_1 \\ 0 & -f & y_1 \\ \vdots & \vdots & \vdots \\ 0 & -f & y_4 \end{bmatrix} \tag{A.3}$$

Then the pseudo inverse of $\hat{L}_{v4T}$ is,

$$\hat{L}_{v4T}^{+} = \hat{Z}(\hat{L}_{v4T1}^{T}\hat{L}_{v4T1})^{-1}\hat{L}_{v4T1}^{T} \tag{A.4}$$

And one has

$$L_{v4T}\hat{L}^{+}_{v4T1} = \frac{\hat{Z}}{Z}(\hat{L}^{T}_{v4T1}\hat{L}_{v4T1})^{-1}\hat{L}^{T}_{v4T1} \tag{A.5}$$

Thus, the estimated interaction matrix relating to the translation motion is subjected to the following constrains.

$$(I + \Delta_{minT}) \leq L_{v4T}\hat{L}^{+}_{v4T} \leq (I + \Delta_{maxT}) \tag{A.6}$$

where matrices Δ_{minT} , Δ_{maxT} satisfy the following equations.

$$\Delta_{minT} = \frac{\hat{Z}_{min}}{Z}[(\hat{L}^{T}_{v4T1}\hat{L}_{v4T1})^{-1}\hat{L}^{T}_{v4T1} - I] \tag{A.7}$$

$$\Delta_{maxT} = \frac{\hat{Z}_{max}}{Z}[(\hat{L}^{T}_{v4T1}\hat{L}_{v4T1})^{-1}\hat{L}^{T}_{v4T1} - I] \tag{A.8}$$

For the 6DOF motions of the robot, it can be infered that matrix Δ_{min} is associated with the lower bound of depth $\hat{Z}_{min}$ and Δ_{max} is associated with the upper bound of the depth $\hat{Z}_{max}$.