CONCORDIA UNIVERSITY

Real-Time Stereo Visual Servoing Of a 6-DOF Robot for Tracking and Grasping Moving Objects

Abolfazl Mohebbi

A Thesis
In The Department of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

February, 2013

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared,

By: **Abolfazl Mohebbi**

Entitled: Real-Time Stereo Visual Servoing of a 6-DOF Robot for Tracking and Grasping Moving Objects

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

Master of Applied Science

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

Signed by the final examining committee:

Rolf Wuthrich	Chair
Weiping Zhu (ECE)	External Examiner
Brandon Gordon	Examiner
Wen-Fang Xie	Thesis Supervisor

Approved by: _____

Chair of Department or Graduate Program Director

_____ _____

Date Dean, Faculty of Engineering and Computer Science

Abstract

Real-Time Stereo Visual Servoing of a 6-DOF Robot for Tracking and Grasping Moving Objects

Robotic systems have been increasingly employed in various industrial, urban, military and exploratory applications during last decades. To enhance the robot control performance, vision data are integrated into the robot control systems. Using visual feedback has a great potential for increasing the flexibility of conventional robotic and mechatronic systems to deal with changing and less-structured environments. How to use visual information in control systems has always been a major research area in robotics and mechatronics. Visual servoing methods which utilize direct feedback from image features to motion control have been proposed to handle many stability and reliability issues in vision-based control systems.

This thesis introduces a stereo Image-based Visual Servoing (IBVS) (to the contrary Position-based Visual Servoing (PBVS)) with eye-in-hand configuration that is able to track and grasp a moving object in real time. The robustness of the control system is increased by the means of accurate 3-D information extracted from binocular images. At first, an image-based visual servoing (IBVS) approach based on stereo vision is proposed for 6 DOF robots. A classical proportional control strategy has been designed and the stereo image interaction matrix which relates the image feature velocity to the cameras' velocity screw has been developed for two cases of parallel and non-parallel cameras installed on the end-effector of the robot. Then, the properties of tracking a moving target and corresponding variant feature points on visual servoing system has been investigated.

Second, a method for position prediction and trajectory estimation of the moving target in order to use in the proposed image-based stereo visual servoing for a real-time grasping task has been proposed and developed through the linear and nonlinear modeling of the system dynamics. Three trajectory estimation algorithms, "Kalman Filter", "Recursive Least Square (RLS)" and "Extended Kalman Filter (EKF)" have been applied to predict the position of moving object in image planes.

Finally, computer simulations and real implementation have been carried out to verify the effectiveness of the proposed method for the task of tracking and grasping a moving object using a 6-DOF manipulator.

To my lovely family,

My parents, ***Abdollah*** *and* ***Fathieh***

My sister, ***Fatemeh***

And my brother, ***Sajjad***

Acknowledgements

By completing this dissertation, my Master's degree takes the final step towards its conclusion. While conceiving these few words, I realized how many people I owe for their support, so that acknowledging all of them within this limited space is not possible. Therefore, I wish to express my sincere gratitude to all those who have helped me in attaining such a goal.

Besides that, it is my pleasure to express my deepest gratitude to my supervisor, Prof. Wen-Fang Xie, for all that I could learn from her talent and experience, and for all the possibilities she gave me during this time. I particularly appreciated the freedom she accorded to my research, and her trust in my capabilities that always encouraged my steps from the very beginning. I am aware that spending these years under her supervision was an invaluable opportunity in my professional and scientific career.

Hearty thanks also go to my colleagues and friends who accompanied me along these years: Mr. Mohammad Keshmiri, Mr. Sadegh Bolouki, Ms. Sara Moghtadernejad, Ms. Saviz Moghtadernejad, Dr. Amir Hajiloo, Mr. Ali Fellah Jahromi, Dr. Yimin Zhao and Mr. Jiannan Zheng. I am also grateful to all those people who have given me support and I didn't have the chance to thank them.

Finally, a special mention for Dr. Mehdi Keshmiri from Isfahan University of Technology who dedicated considerable amounts of his time to teach me the knowledge and expertise gained from his wide experience. He followed me step-by-step and constantly supported me whenever I ran in trouble during my academic life.

Last, but not least, my gratitude and sincere thanks go to my parents for their support, kindness and 'being always there' regardless of my choices, and for having allowed, with their efforts, to get to where I am now.

Table of Contents

List of Figures

List of Tables

Chapter 1 : Introduction

This chapter gives an introduction of Stereo Visual Servoing in grasping moving objects. Some related topics and previous works are given. In Section 1.1 the background and problems of dynamic visual servoing control of robots are outlined. Section 1.2 includes a summary of the stereo vision and its application to robotic servoing tasks. In Section 1.3 an introduction to the application of tracking and grasping moving objects using visual servoing is given. Section 1.4 addresses and discusses the previous works in this field. The motivations, objectives and contributions of this study and organization of the materials of current thesis are also summarized in Section 1.5. And finally Section 1.6 gives a summary of this chapter.

1.1. Visual Servoing: Literature Review

One of the most challenging technological endeavors of human kind has been giving the capabilities of gathering complex information on the environment to machines in order to interact with it in an autonomous manner. The two most important senses which provide sufficient environmental information for human to perform interaction tasks are the tactile sense and the visual sense. The devices that are able to partially imitate these human senses are the force sensors and the visual sensors, respectively. The visual sense is often lacking in many human-made machines. In fact, without visual information, manipulating devices can operate only in "Structured" environments, where every object and its relative position and orientation is known a priori. With the increase of real-time capabilities of visual systems, vision is beginning to be utilized in the automatic control as a powerful and versatile sensor to measure the geometric characteristics of the workpiece.

A "*Visual Servoing System*" is a feedback control system based on visual information. A visual servoing system is essential for autonomous robots working in unknown or unstructured environments. In general, this system is composed of one or more cameras, a processing or computing unit, and specific image processing algorithms to control the position of the robot's end-effector relative to the object or work piece as required by the task. Visual servoing systems have been increasingly used in control of robot manipulators that is based on visual perception of robot and object location. It is a multi-disciplinary research area spanning computer vision, robotics, kinematics, dynamics, control and real-time systems.

The history of visual servoing goes back to the seventies. In the early 1970s, Shirai and Inoue [1] described how *visual feedback*, as the use of vision in the feedback loop, can increase the accuracy in tasks. The term "*visual servoing*" was first introduced by Hill and Park [2] in 1979. Prior to the introduction of this term, the less specific term *visual feedback* was generally used. Afterwards, considerable researches [3, 4] has been performed on the development of visual servoing control systems. The analytical complexity of robot control systems and also processing vision data have made the vision-based control problem challenging. Recently, both computers and video cameras are fast and advanced and consequently are increasingly used as robotic sensors in feedback control systems such that the control of robots employing visual feedback is practically feasible.

1.1.1. Applications of Visual Servoing Systems

The main application of visual servoing in industrial robotics concerns with the control of the end-effector pose (position and orientation) with respect to the pose of objects or obstacles which can be fixed or moving in the workspace of the robot. Positioning or moving some objects, assembling and disassembling mechanical parts, paintings, welding, etc. are the examples of the tasks which these robotic systems can perform [5].

In robot visual servoing system the control of the pose is determined using synthetic "*Image Features*" extracted from a sequence of images captured with imaging devices [6, 7]. These image features are provided by an imaging device e.g. one or more cameras,

mounted on the end effector of the robot or in a fixed position with respect to the robot workspace.

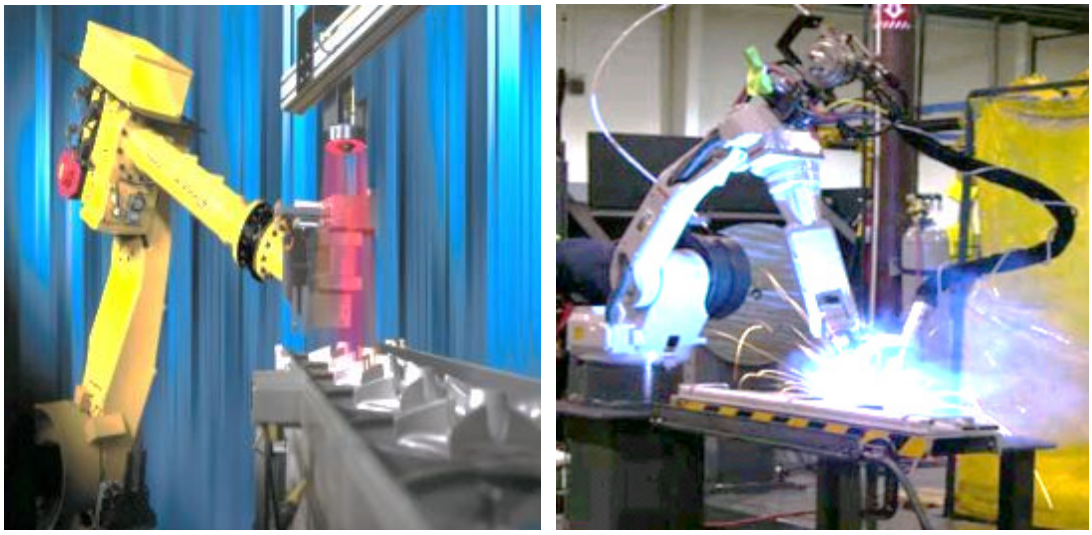


Figure 1.1 Robots using visual feedbacks to perform various tasks, [8], [9]

Visual Servoing tends to be widely used in medical and surgical applications to position instruments or perform the medical operations. For instance "*Laparoscopic Surgery"* is minimally invasive, which needs only several small incisions in the abdominal wall to introduce instruments such as scalpels, scissors, etc., and a laparoscopic camera, such that the surgeon can operate by just looking at the camera images. To avoid the need for another assistant and to free the surgeon from the control task, an independent system that automatically aims the laparoscope is highly desirable. Several researchers have tried to use visual servoing techniques to guide the instrument during the operation (see Figure 1.2) [10].

Figure 1.2 Application of Visual Servoing in Medical Robotic Systems [11]

Control and guidance of unmanned vehicle systems is another example of using visual servoing technique for the exploration or reconnaissance operations. The pose of an unmanned ground vehicle (UGV) is typically required for autonomous navigation and control [12]. Often the pose of an UGV is determined by a global positioning system (GPS) or an inertial measurement unit (IMU). However, both GPS and IMU have many limitations such as signal availability and in many cases high costs. Given recent advances in image processing technology, an interesting approach to overcome the pose measurement problem is to use a visual servoing system (Figure 1.3).

Figure 1.3 Using vision system in a mobile unmanned vehicle [13]

In an another application of visual servoing systems, space robots are used to perform autonomous on-orbit servicing which includes approaching and docking to a target satellite and grasping some complex parts for the purpose of refueling and servicing [14].

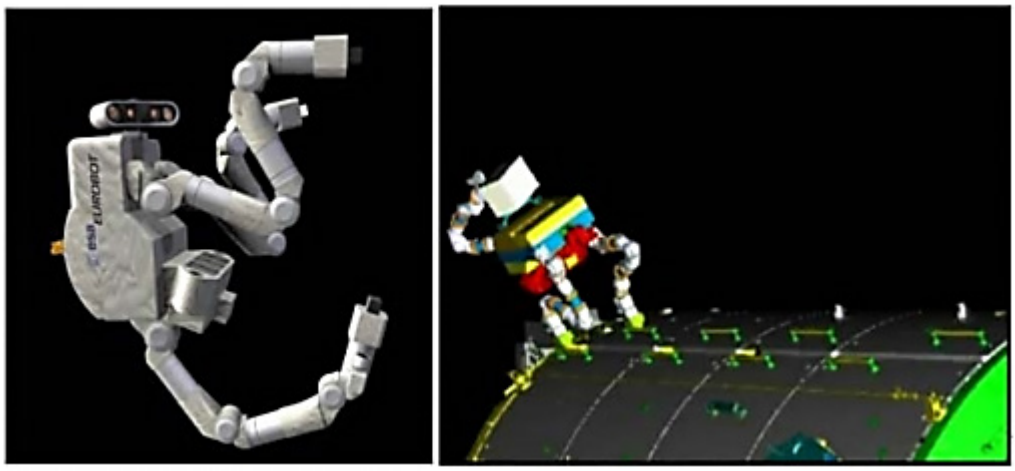


Figure 1.4 A space robotic system in a mission of on-orbit servicing [14]

1.1.2. Classifications of Visual Servoing Systems

Based on the robot-camera configuration, the visual servoing systems are classified as two major classes called *Eye-in-Hand* and *Eye-to-Hand* [15]. In Eye-in-Hand systems the camera is rigidly mounted on the robot end-effector and in Eye-to-Hand configuration the imaging device is fixed in the workspace to observe both the robot and the object. One or more cameras can be used either as these two configurations or a combination of both. Each of these configurations is used in various servoing tasks based on the limitations of the experiments and applications [16]. Figure 1.5 illustrates various camera-robot configurations.

Monocular systems use a camera either as a global sensor ("Eye-to-Hand" stand–alone configuration) or as an Eye–in–Hand configuration. These systems usually adopt some form of model based visual techniques to facilitate the estimation of the depth between the camera and the object [17]. If the Eye-to-Hand configuration is used, a geometric model of the object is commonly used to retrieve the full pose of the object. On the other hand, in the Eye–in–Hand configuration, feature-based tracking techniques are vastly used [18]. A single camera minimizes the processing time needed to extract visual information. However, in the case where the object model is unknown, the loss of depth information limits the servoing operations and complicates the control design.

Two cameras in a stereo configuration can be used to provide complete 3D information about the scene and the object [19]. One of the common approaches is to estimate

the disparity which is then used for depth estimation [20-22]. The fundamental problem of disparity estimation is to match the corresponding features between two or more images [23]. Since we use two different imaging devices, the stereo vision systems require twice as much computational effort in image processing as the monocular systems. Since we use the stereo configuration in the current research, more details about the methods employing stereo vision systems will be thoroughly discussed in subsequent chapters. The multiple cameras provide additional information compared to a single or stereo camera configurations. However, matching across multiple views is usually hard, time consuming and expensive. Therefore, the research reports on employing more than two cameras for controlling a robot are rare.

The second major classification of visual servoing is based on control input signal and involves "*Direct Visual Servo Control*" systems versus *"Look-and-Move"* architectures [24]. In general, direct visual servo controller directly computes the torque inputs of robot joints and uses only the visual feedback to stabilize the system. It is also important to consider the robot dynamics.

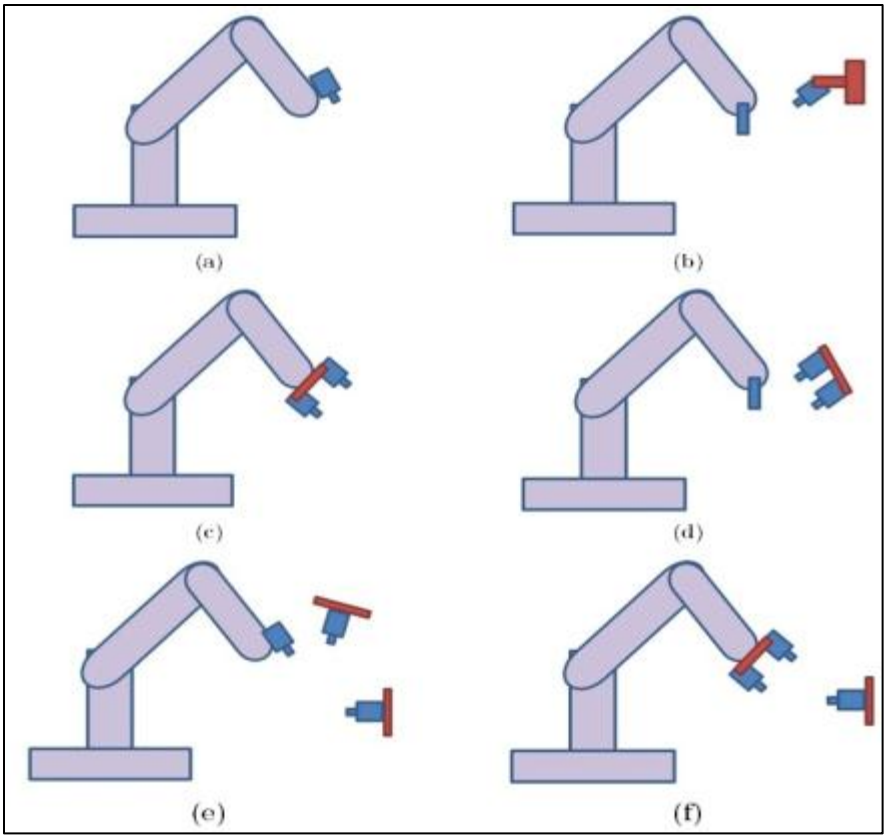


Figure 1.5 Camera–robot configurations used in visual servoing control
(a) Monocular eye in hand (b) Monocular Eye to hand
(c) Stereo eye in hand (d) Stereo eye to hand
(e) and (f) combination of eye in hand and eye to hand configuration

In the look-and-move structure a vision based controller generates the inputs of the joint controller of robot. Therefore the stability of the system relies on both robot joint controller and the vision-based controller. Since the vision system can hardly provide visual information at a high sampling rate due to the limitation of camera, it might affect the stability of the robot system with the vision feedback.

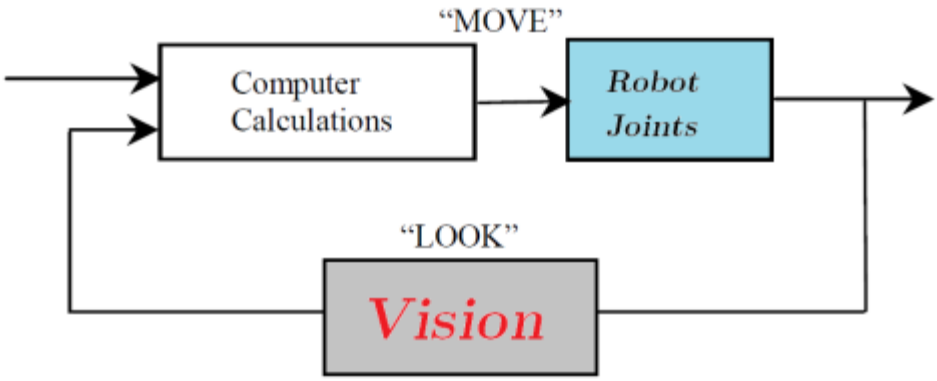
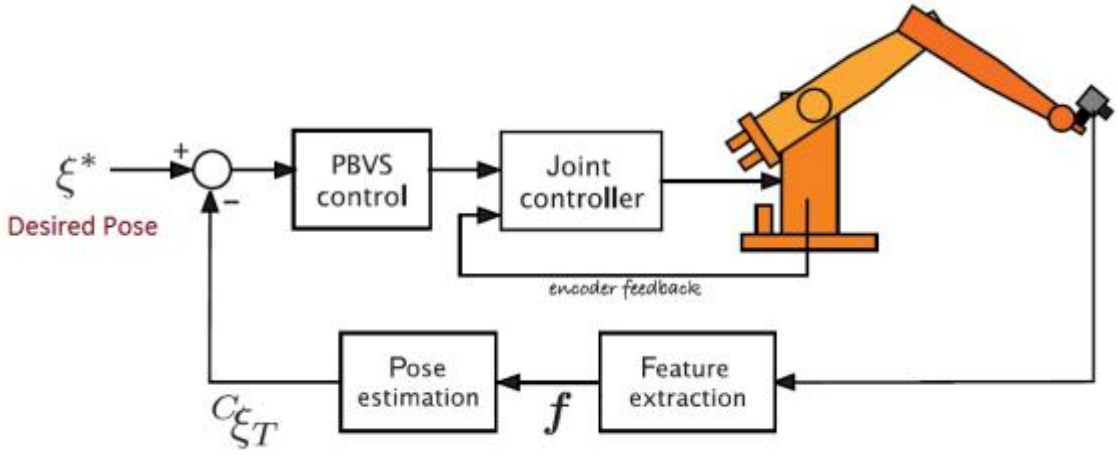


Figure 1.6 Look and Move control Architecture

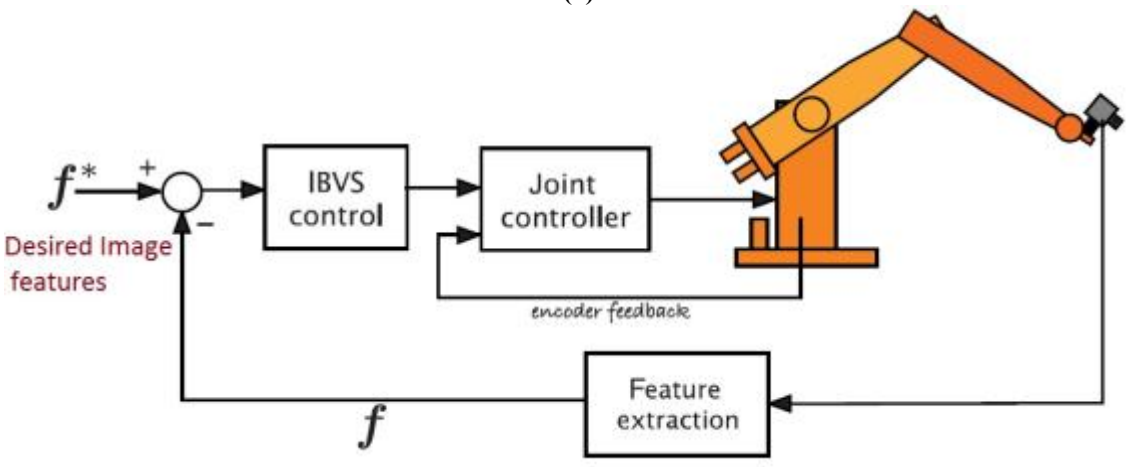
Based on the system error signal, there are two fundamentally different approaches to visual servoing control: Position-Based Visual Servo (PBVS) and Image-Based Visual Servo (IBVS) [25]. Position-based visual servoing (PBVS), shown in Figure 1.7a, uses the observed image features from a calibrated camera and a known geometric model of the target to determine the pose of the target with respect to the camera [26]. In a PBVS system the pose of the target with respect to the camera is estimated [27]. Knowing the target's geometry is essential for the pose estimation problem. The camera intrinsic parameters and the observed image plane features are needed as well. The robot then moves toward that pose and the control is performed in task space. Various algorithms have been proposed for pose estimation [28, 29], it is computationally expensive and relies critically on the accuracy of the camera calibration and the model of the object's geometry.

Image-based visual servoing, shown in Figure 1.7b, does not include the pose estimation step. It uses the image features for control signal calculation directly. The control

is performed in image coordinate space [30, 31]. The desired camera pose with respect to the target is defined implicitly by the image feature values at the desired pose. The image features are highly non-linear functions of camera pose which make IBVS to be a challenging control problem. IBVS differs fundamentally from PBVS in omitting the estimation of the relative pose of the target. The relative pose is implicit in the values of the image features. The control problem can be formulated in terms of image coordinates. The task is to move the feature points to the desired position. Moving the feature points in the image space implicitly changes the pose in the Cartesian space.



(a)



(b)

Figure 1.7 Two Different Approaches of Visual Servoing Control [32]

There are also other visual servoing control schemes which have been proposed to improve the performance of classic IBVS and PBVS approaches. In a "*Hybrid Visual Servo*" the respective advantages of these schemes are combined while trying to avoid their shortcomings [33, 34]. The main idea is to use a hybrid of Cartesian and image space sensory feedback signals to control trajectories in both the Cartesian and image spaces simultaneously. The method known as two and one-half-dimensional (2 1/2-D) Visual Servo [34] was the first to exploit a partitioning approach in combining IBVS and PBVS. The Cartesian and image trajectories are controlled simultaneously so that they are approximate to straight line in 3D Cartesian space and image plane respectively.

More precisely this approach is a "halfway" between the classical position-based and image-based approaches. It avoids their respective disadvantages as follows. To the contrary the PBVS, it does not need any geometric 3D model of the object. In comparison to the image-based visual servoing, it ensures the convergence of the control law in the whole task space and it avoids the image singularities and local minima problem. The method is based on the estimation of the camera displacement (the rotation and the translation of the camera) between the current and the desired views of the object. In each iteration, the rotation between these two views is estimated and thus allows for the translational and the rotational loop to be decoupled. One of the drawbacks of this method is that it is more sensitive to image noise than IBVS, since this scheme directly uses visual features as input of the control law, without any supplementary estimation step [35]. Figure 1.8 shows a block diagram of 2-1/2 D visual servoing.

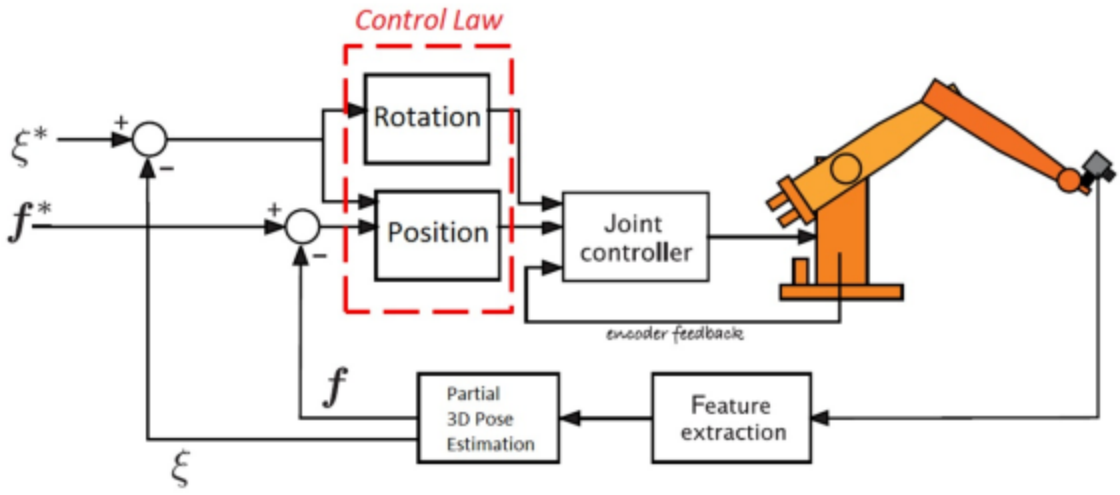


Figure 1.8 A block diagram of 2-1/2 D visual servoing

In the hybrid visual servoing schemes, the goal is to decouple the rotational motions from the translational ones by selecting adequate visual features defined partially in 2-D, and partially in 3-D. Based on this method some researches are performed to find the features that exhibit similar decoupling properties but expressed directly in the image space[36, 37]. More precisely, the goal is to find six features so that each of them is related to only one degree of freedom and makes the interaction matrix relating the image features velocity to the camera velocity screw to be diagonal.

The first work in this area (so called "*Partitioned Visual Servo"*) partitioned the interaction matrix so as to isolate motion related to the optic axis [37]. As it was discussed before we encountered the problem of camera retreat in an IBVS system. More precisely the partitioned methods solve the problem of "camera retreat" by using IBVS to control

some degrees of freedom while using a PBVS controller to control the remaining degrees of freedom.

The "*XY/Z schemes"* consider the *x*- and *y*-axes as one group, and the *z*-axes as another group [36]. The approach is based on a couple of insights. Firstly, the camera retreat problem is a *z*-axis phenomenon: *z*-axis rotation leads to unwanted *z*-axis translation. Secondly, the image plane motion due to *x*- and *y*-axis translational and rotation motion are quite similar. At the end, Figure 1.9 summarizes the different classifications for Visual Servoing Systems.

To illustrate the procedure of visual servoing and to compare different visual servoing approaches, the authors of [30] have introduced a good example. In this example different behaviors for each method are shown. The task is to move the camera from its initial position to a desired position which leads to a desired image. Figure 1.10 shows the example of camera positioning task which has been introduced in [30].

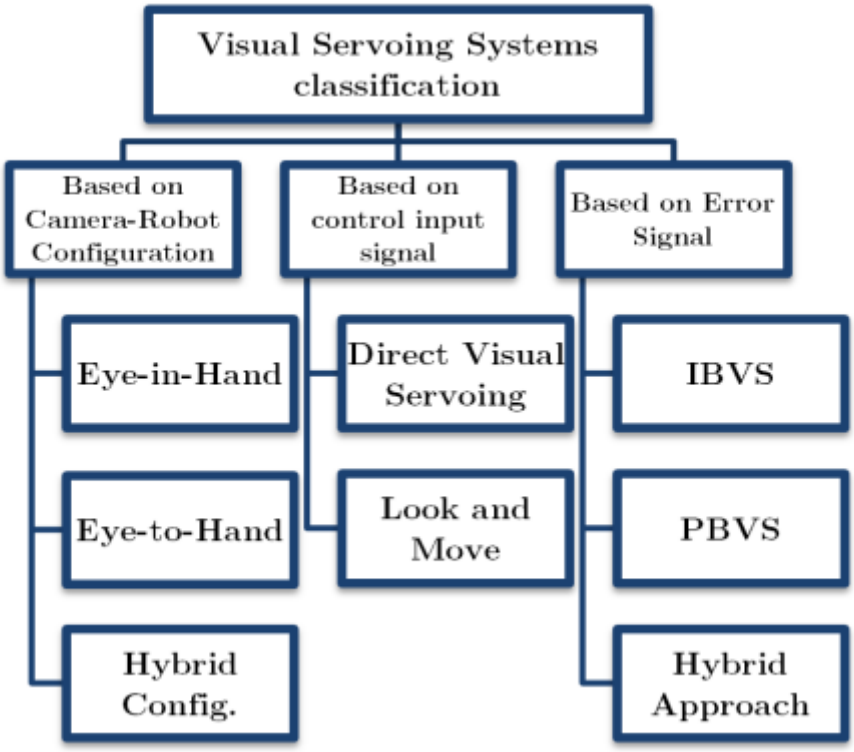


Figure 1.9 Different classifications for visual servoing systems

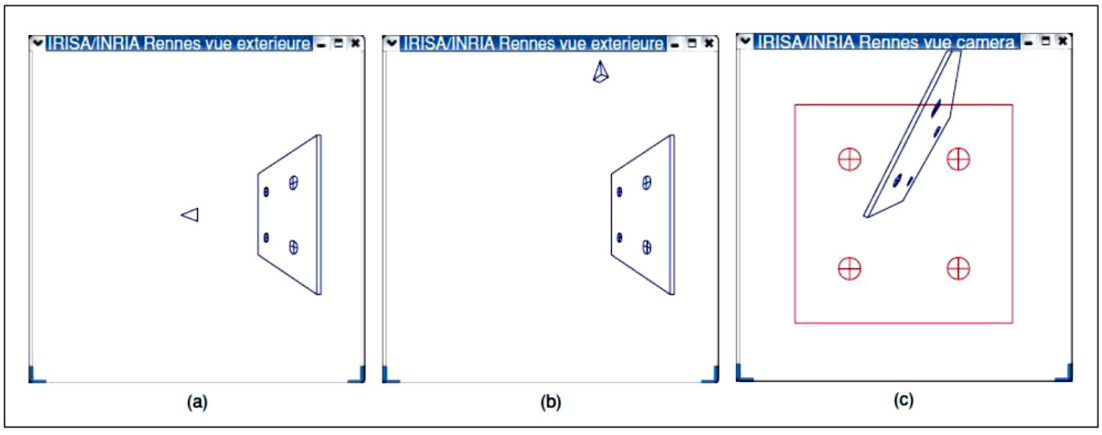


Figure 1.10 a) Desired camera position, b) Initial camera position, c) Camera image in its initial position and desired position (red) [30]

Also the results of different position-based, image-based, hybrid and partitioned visual servoing approaches are given in Figure 1.11.

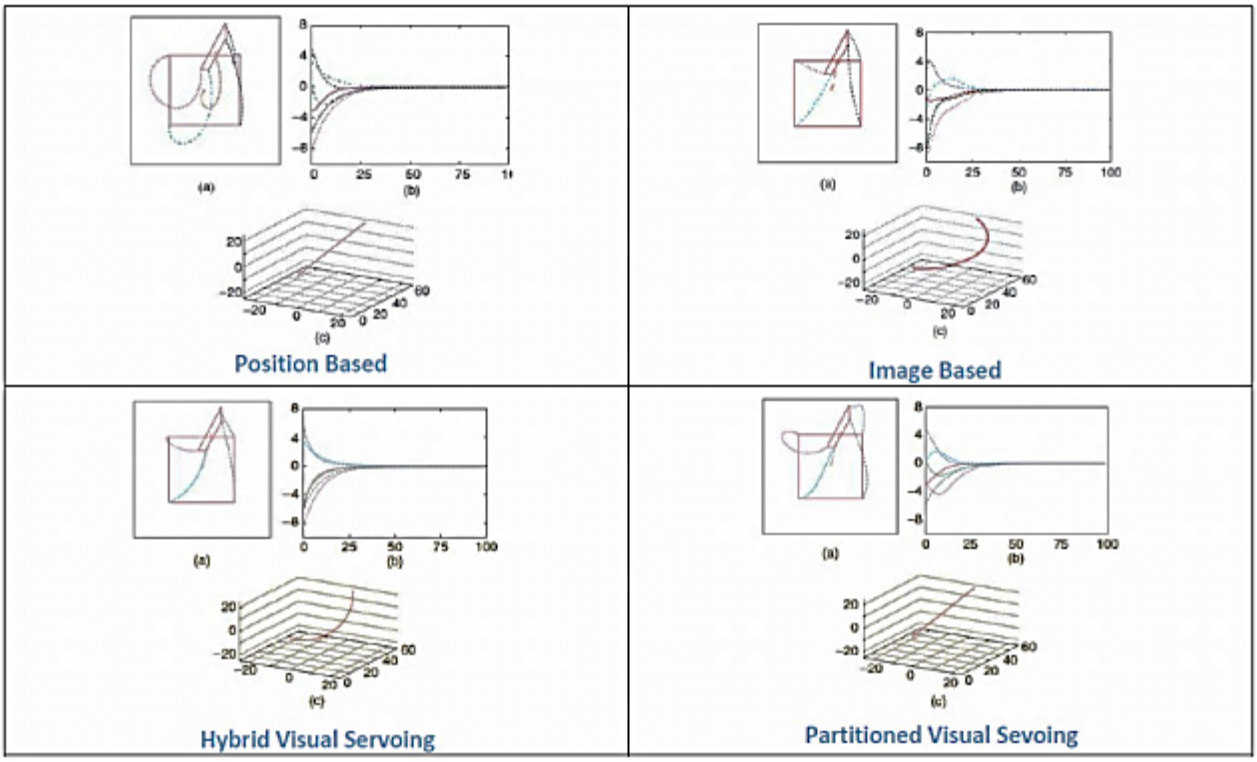


Figure 1.11 Results of visual servoing approaches: a) Trajectory in the image plane, b) Controlling Velocity signals during the motion c) Trajectory in the Cartesian space [30].

1.2. Stereo Vision

It is quite straightforward to apply the image based visual servoing approach to a multi-camera system [38, 39]. If a stereo vision system is used and a 3-D point is visible in both left and right images, it is possible to use it as visual features. In the case of using a monocular vision system where only a single camera is concerned, a certain number of assumptions, such as camera calibration are needed to be made. In that case, it is important to keep in mind that a number of some singularities exist which make visual servoing control of the robot impossible near those configurations. The use of a binocular

vision system avoids these singularities, and requires less strict camera calibration. Hence, in the stereo vision configuration it is also possible to estimate 3-D parameters by using the epipolar geometry that relates the images of the same scene observed from different viewpoints [40].

Stereo visual servoing has many advantages over the classical monocular visual servoing approaches. The depth information can be recovered without any geometrical model of the observed object. It should be noted that even in image based visual servoing, this information is needed for the computation of the image interaction matrix. A comprehensive study and notations are fully described in the next chapters to illustrate these advantages. Figure 1.12 shows a manipulator robot in a stereo visual servoing procedure.

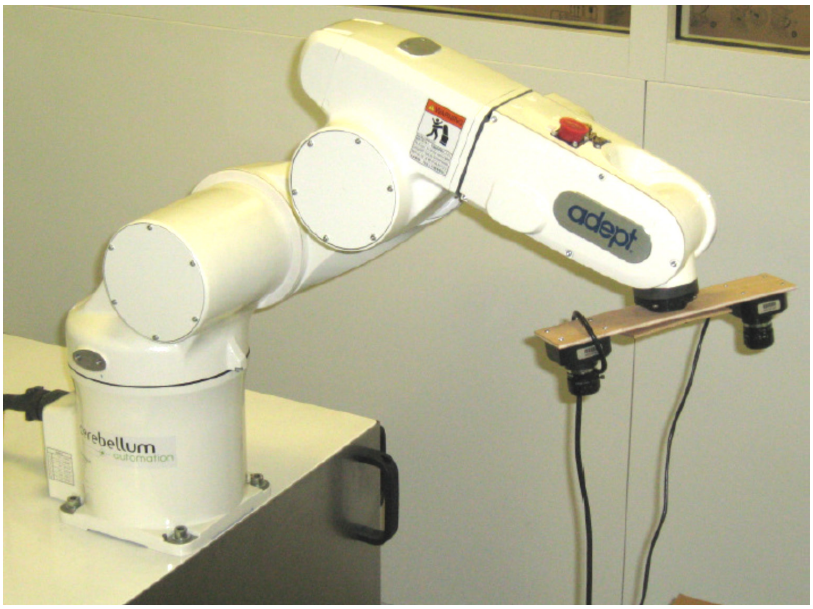


Figure 1.12 A six degrees of freedom robot using a stereo vision to perform servoing tasks [41]

1.3. Catching 3D Moving Objects

Real-time tracking, grasping or intercepting moving objects using robot manipulators has been a part of applications in industrial fields [42]. Under various situations the manipulator is used to grasp the stationary objects. For instance a component on a conveyor arrives at a specific position and stops then the robot moves and grasps the object. This scenario is time consuming and has a low efficiency. Therefore some researchers developed controller to visually guide the robot to catch the object during its movement. These researches have recognized that the main problems in the visual servoing for catching moving objects are to solve the delay of the systems due to image processing or the response of the robot system and to perform the target grasping [43]. These challenges are the major reason for a limited performance in the tracking and grasping process which can be solved via use of predictive algorithms.

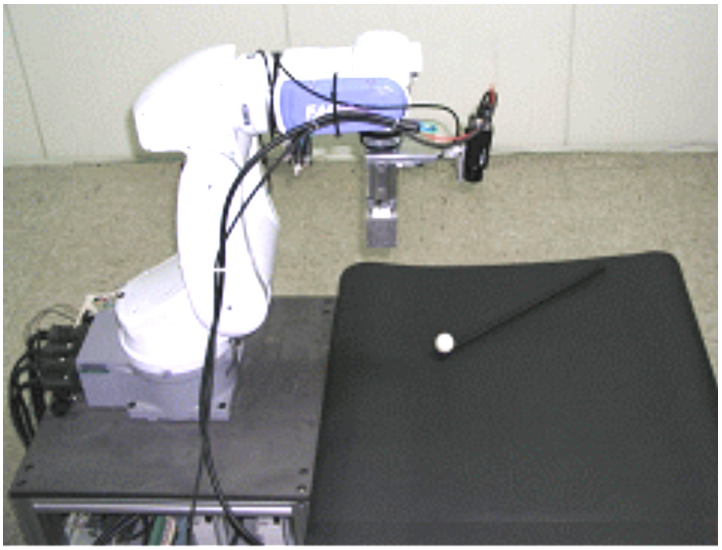


Figure 1.13 A manipulator robot trying to track and catch a moving object [44]

1.4. Previous Works

The research on grasping a target object in motion with a robotic manipulator has been reported in different works. Allen et al. [45] presented a system to track and grasp an electric toy train moving in an oval path using calibrated static stereo cameras. Nomura et al. [46] proposed a method to grasp efficiently the objects and developed a system able to grasp industrial parts moving on a conveyor belt by controlling a 6DOF robot arm with a camera mounted on its gripper. Sen et al. [47] implemented a real time vision system with a single camera for identifying and intercepting several objects. Li et al. [48] proposed a visual servo system for real-time tracking and grasping of a moving object and a parallel method was adopted to increase the matching speed.

Yeoh and Abu-Bakar [49] presented a tracking algorithm based on a linear prediction of second order solved by the maximum entropy method. It attempts to predict the centroid of the moving object in the next frame, based on several past centroid measurements. Matas and Zimmermann [50] represented the tracked object as a constellation of spatially localized linear predictors which were trained on a single image sequence. In a learning stage, sets of pixels whose intensities allow for optimal prediction of the transformations are selected as a support for the linear predictor.

F. Pacheco [51] presented a binocular eye-to-hand visual servoing system that is able to track and grasp a moving object in real time. In the tracking module, they use three linear predictors to predict and generate the trajectory that describes the 3D object posi-

tion in the near future. The training of the predictors is done offline and they adapt online to new movements of the object over time. Their visual servoing system was implemented with a CRS T475 manipulator robot with six degrees of freedom and two fixed cameras in a stereo pair configuration.

1.5. Objectives, Contribution and Thesis Organization

This research work aims at developing a stereo eye-in-hand image based visual servoing system that is able to track and grasp a moving object in real time. The procedure of controlling robot joints to grasp a moving object is illustrated in Figure 1.14. In the current research a robot visual servo control algorithm is proposed by using an image based eye-in-hand stereo visual servoing algorithm combined with a moving object model applied to a motion predictor e.g. an extended Kalman filter or a least square method.

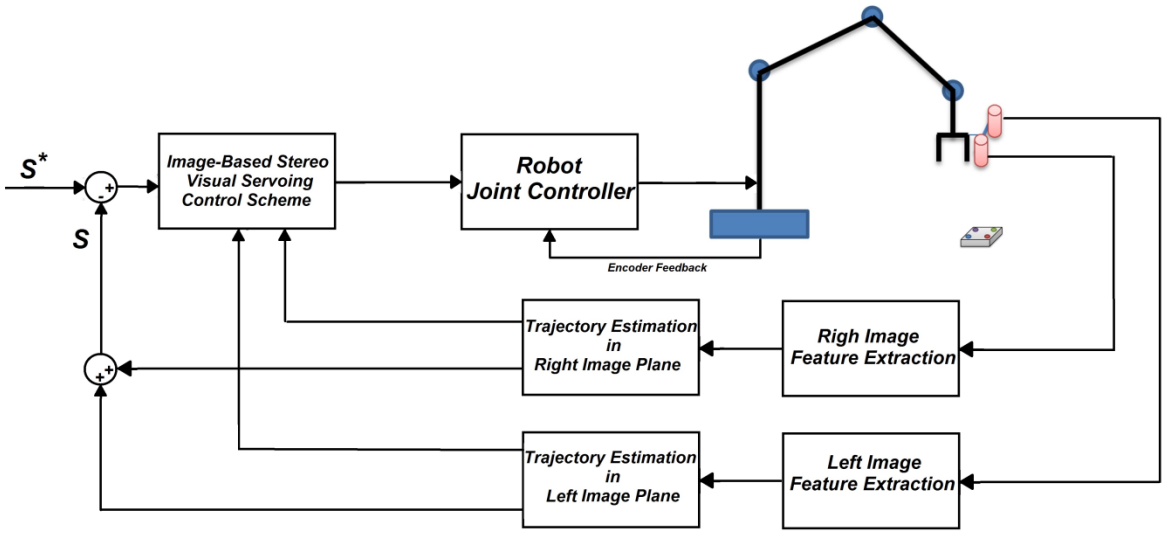


Figure 1.14 The structure of grasping moving object using eye-in-hand stereo Image-based visual servoing

In this research we have addressed the problem of grasping as an eye-in-hand visual servoing problem which can be expressed and defined as “finding the (proper) motion of the manipulator that will grasp a moving object with limited motion velocity”. With regards to the visual servoing problem the procedure of grasping can be also stated as: “finding the motion of the manipulator that will cause the image projections of target’s feature points to move and coincide to the desired image features”. This can be done by pre-defining desired positions for the object’s image features such that the robot moves and aligns the end-effector with the object and reaches towards it. By using a stereo vision system we can extract the exact depth and 3-D position information that helps the procedure of servoing and grasping to be more accurate. Figure 1.15 illustrates the grasping procedure as a visual servoing task with predefined desired image features.

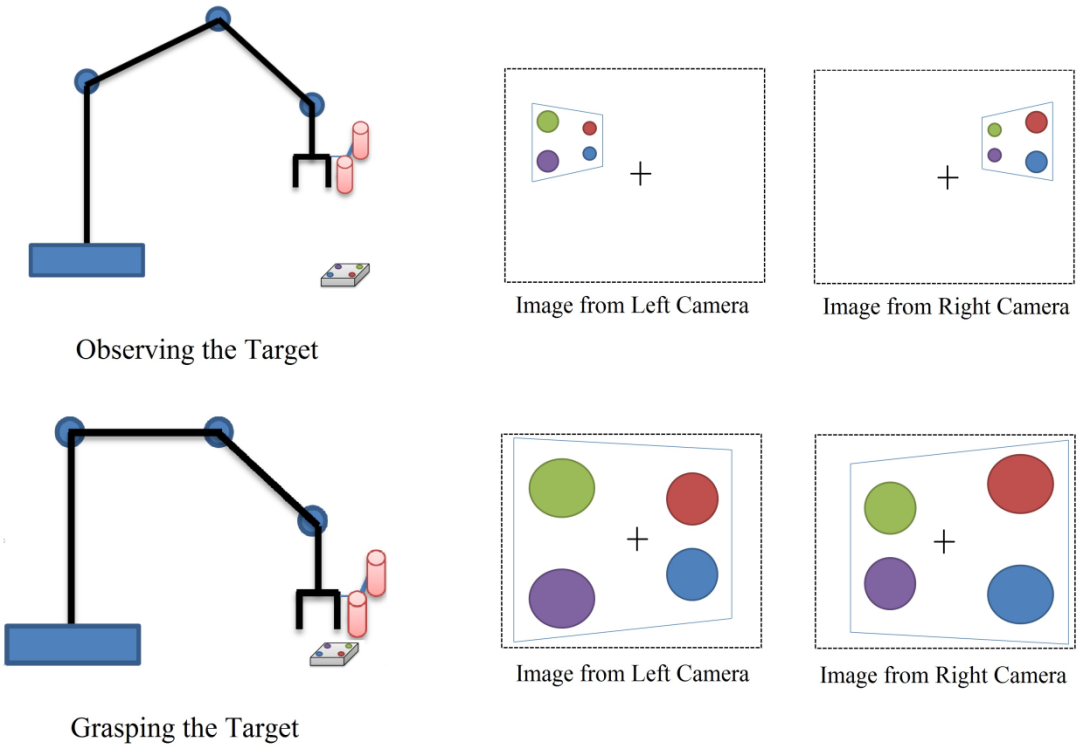


Figure 1.15 Image features in a task of grasping using stereo visual servoing

Accordingly, as shown in the "off-line feature selection" part, the left and right images of an object in the grasping area which are very close to the end-effector are predefined as desired features, S^* . Thus, it is important to mention that the desired image features in this visual servoing system are constant. As the varying image features, the image plane positions of the maneuvering object are grabbed from left and right cameras.

Then using a tracking and trajectory estimation module, a Kalman filter and also a recursive least square prediction method are used to predict and estimate the image position of the object. Using the predicted positions, the interaction matrices for left and right image features are calculated and then a united stereo interaction matrix, J_{st}, is obtained

and used in a *proportional control* scheme. The predicted positions are also used to calculate the feature error which is used in the control signal to produce velocity signals for the robot joint controller.

The extensive computer simulations and real time experiment have been performed with a six-degrees-of-freedom DENSO manipulator robot and two cameras in a stereo pair configuration are mounted on the end effector.

Some related publications from this thesis research results are listed as follows:

- A. Mohebbi, M. Keshmiri, W. F. Xie, "Eye-in-Hand Image Based Stereo Visual Servoing for Tracking and Grasping Moving Objects" submitted to IEEE AIM 2013, 2013
- A. Mohebbi, M. Keshmiri, W. F. Xie, "Tracking and Grasping moving objects through Image-based Visual Servoing systems: Stereo Vs. Mono", to be submitted to the Elsevier- Journal of Robotics and Autonomous Systems, 2013.
- A. Mohebbi, M. Keshmiri, W. F. Xie, "Augmented Image-based Stereo Visual Servoing for Grasping a Moving Object Using Acceleration Commands", to be submitted to IEEE transactions on Industrial Electronics, 2013.

This thesis has six main chapters which are organized as follows; In Chapter 2, the stereo vision system is described and the major elements to design the eye-in-hand stereo visual

servoing system are presented. Chapter 3 is devoted to the detailed description of the trajectory estimation of the moving object and the visual servoing control for the robot to grasp a specific moving target object. In Chapter 4 system model including robot and vision system is discussed and the simulation results on the proposed strategies are illustrated.

Chapter 5 is devoted to describing the experimental setup including Robot, Controller, vision system, stereo rig and also describing camera calibration process. The image processing and feature extraction procedures are briefly described in this chapter. Finally the results on the experiments on different grasping and servoing tasks are given and discussed. Finally the conclusion of the research is drawn and the future work is suggested in Chapter 6.

1.6. Summary

In this chapter, a brief introduction to basic concepts and components of visual servoing system is presented. Different classifications based on camera-robot configuration and control strategy are given. And the advantages and disadvantages of each class are addressed. Brief review on the existing works on the stereo visual servoing for tracking the moving object is given. The information about current research, previous works and thesis organization was also given in this chapter.

Chapter 2 : Image-Based Stereo Visual Servoing

In this chapter an image-based visual servoing approach based on stereo vision is presented and mathematically discussed. As in our experimental test-bed, a pair of cameras is considered to be mounted on the end-effector of the manipulator arm (see Figure 2.1). Most of conventional monocular visual servoing techniques need a priori knowledge of a parameter of depth/distance or the object model to calculate a “fixed” image interaction matrix for the control law which tends to slow down the speed of convergence procedure in servoing task. Using the stereo vision, it can be shown that the epipolar geometry of two images can be used to calculate the image interaction matrix at any position without need of any geometrical model of the observed object, and results in better and faster performance characteristics.

Use of stereo vision significantly increases the quality of execution of the servoing task, especially where the precise, robust and smooth movement is needed. As it is discussed in [52], in monocular visual servoing, a number of singularities may exist which make visual control performance impossible near those configurations. These singularities can be avoided by using a stereo camera system, which requires less strict camera calibrations.

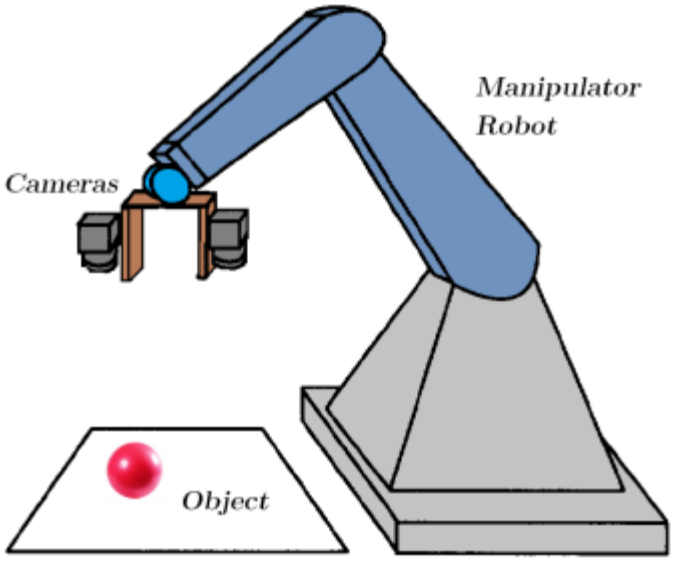


Figure 2.1 A stereo eye-in-hand system mounted on a manipulator robot's end-effector

The rest of this chapter is organized as follows: first, a review of classical monocular image based visual servo control (IBVS) will be thoroughly presented. Then, the models of two stereo images of a set of points are built and two new stereo interaction matrices are derived for both “parallel” can “non-parallel” stereo rig configurations. . The depth information is extracted by the means of epipolar geometry for the interaction matrix calculation.

Based on the interaction matrices, a proportional control scheme has been proposed for the image-based stereo visual servoing system. At the end, the classical monocular IBVS and the proposed stereo IBVS are compared and the advantages and disadvantages of each one with regards to our application will be pointed out.

2.1. Classical Image-Based Visual Servoing Control

In general the main aim of vision-based control schemes is to minimize an error which can be defined by:

$$e(t) = s - s^{*} \tag{2.1}$$

where **s** and $\mathbf{s}^{*}$ are vectors of current and desired visual/image features. It is noted that **s** is a function of image measurements (e.g., the image coordinates of interest points or the image coordinates of the centroid of an object) and a set of parameters that represent the same knowledge about the system (e.g., camera intrinsic parameters or the objects 3-D models)[30].

At this point we consider the case where the target and also the goal pose are fixed, i.e., $\mathbf{s}^{*}$ is constant, and the changes in **s** depend only on camera motion which is controlled with six degrees of freedom (6 DOF), and a camera is attached to the end effector of a six degree-of-freedom manipulator arm.

As it was mentioned before, one of the main classifications of visual servoing schemes is based on system error signal which is mainly the way how **s** (features) is designed. This classification involves two major approaches: 1- image-based visual servoing control (IBVS), in which **s** consists of a set of features that are immediately available in the image data. 2- Position-based visual servoing control (PBVS), in which **s** consists of a set of 3-D parameters, which must be estimated from image measurements. After selecting a proper feature set for **s** we need to design a control scheme which is a velocity controller. To design such controller, we require the relationship between the time variation of **s** and the camera velocity:

$$\dot{s} = J_s u_c, \tag{2.2}$$

where $u_c = (v_c, w_c)$ is the spatial velocity of the camera, v_c is the instantaneous linear velocity of the origin of the camera frame , w_c is the instantaneous angular velocity of the camera frame, $J_s \in R^{k\times 6}$ is called *Jacobian Matrix* or *feature interaction matrix*.

Since $\mathbf{s}^*$ is constant, its time derivatives is equal to zero. Consequently we have:

$$\frac{ds^*}{dt} = 0 \rightarrow \frac{d}{dt}e(t) = \frac{d}{dt}(s - s^*) = \frac{ds}{dt} = \dot{s}.$$

Then we will have:

$$\dot{e} = J_e u_c, \tag{2.3}$$

where $J_e = J_s$. A traditional proportional controller u_c would be the input to the robot controller. By Letting $\dot{e} = -\lambda e$, the control signal can be designed as:

$$u_c = -\lambda J_e^{+} e, \tag{2.4}$$

where $J_e^{+} \in R^{6\times k}$ is the Moore-Penrose pseudo-inverse of J_e that is $J_e^{+} = (J_e^{T} J_e)^{-1} J_e^{T}$ when L_e is of full rank 6. And when $k = 6$ and if $\det(J_e) \neq 0$ we can obtain $u_c = -\lambda J_e^{-1} e$.

In real visual servoing systems, it is impossible to know perfectly either J_e or J_e^{+}. So an approximation or an estimation of one of these two matrices can be determined; e.g. $\hat{J}_e$ and $\hat{J}_e^{+}$. So the control law can be re-written as:

$$u_c = -\lambda \hat{J}_e^{+} e, \tag{2.5}$$

We now can consider the case in which the vision system observes a ***"moving target"*** and desired values for the features are not constant, e.g. $ds^* / dt \neq 0$. The time variation of the error can now be obtained by [9]:

$$\dot{e} = (\frac{\partial e}{\partial r})\, u_c + \frac{\partial e}{\partial t} = J_e u_c + \frac{\partial e}{\partial t} \tag{2.6}$$

where the term $(\partial e / \partial t)$ represents the time variation of *e* caused by the target motion which is considered to have a *constant velocity*. This term must be introduced in the control law to compensate for the target motion. With the same simple proportional control law (by letting $\dot{e} = -\lambda e$), we now can obtain:

$$u_c = \widehat{J_e^+}\ (-\lambda e - \widehat{\frac{\partial e}{\partial t}}), \tag{2.7}$$

where $\widehat{\partial e / \partial t}$ is an estimation or an approximation of $\partial e / \partial t$. In order to find a proper estimation for $\partial e / \partial t$ and cancel the tracking errors caused by the target motion and in the case of a target with constant velocity, a classical integral term can be used in the control law as follows when a discrete time case is considered [8]:

$$\left(\widehat{\frac{\partial e}{\partial t}}\right)_k = \mu_I \sum_{i=0}^{k-1} e_i \tag{2.8}$$

where μ_I is the integral gain, which must be tuned and $\left(\widehat{\frac{\partial e}{\partial t}}\right)_0 = 0$.

Through the available image measurements and the camera velocity calculations, $\partial e / \partial t$ can be directly estimated based on feed-forward control;

$$\left(\widehat{\frac{\partial e}{\partial t}}\right)_k = \frac{e_k - e_{k-1}}{\Delta t} - \left(\widehat{J_e^+}\right)_k \left(u_c\right)_{k-1} \tag{2.9}$$

where Δt is the duration of the control loop, *k* is the current time step.

In order to control the robotic systems using visual feedbacks, it is essential to understand the geometric aspects of the imaging process. Each camera includes a special lens which forms a 2D projection of the objects on the image or sensor plane. There are many projection models to describe these geometric aspects of imaging such as: perspec-

tive projection, scaled orthographic projection, and affine projection. In our research we assume it is perspective projection [53]. For a point, in a three dimensional space ${}^{C}P = [X, Y, Z]^T$ whose coordinates are expressed with respect to the camera coordinate frame, {C}, which projects onto the 2D image plane with coordinates $p = [u, v]^T$, in which *u* and *v* are described in **pixel dimension**. So we have:

$$x = \frac{X}{Z} = \frac{(u - \sigma_u)}{f} \cdot \rho_1 \tag{2.10}$$

$$y = \frac{Y}{Z} = \frac{(v - \sigma_v)}{f} \cdot \rho_2 \tag{2.11}$$

where $[x, y]^T$ are the normalized coordinates from $p = [u, v]^T$, and f is focal length described in **metric** dimensions and we have $s = (x, y)$. Figure 2.2 describes the perspective projection model.

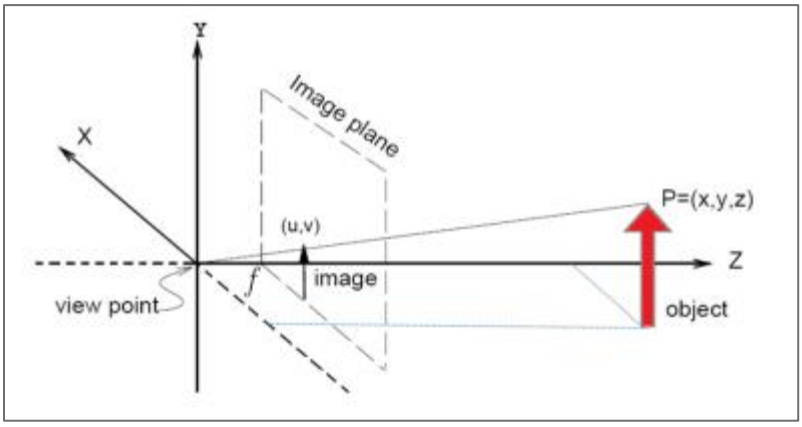


Figure 2.2 Coordinate frame for camera-lens system [26]

where $(\sigma_u, \sigma_v, f, \rho_1, \rho_2)$ is the set of camera intrinsic parameters: σ_u and σ_v are the coordinates of the camera principal point, f is the focal length, and (ρ_1, ρ_2) are the transformation constants from pixel dimensions to metric dimensions where $\rho_2 / \rho_1 = \alpha$.

Taking the time derivative of the perspective projection Equations (2.6), we can obtain:

$$\dot{x} = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{(\dot{X} - x\dot{Z})}{Z} \tag{2.12}$$

$$\dot{y} = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{(\dot{Y} - y\dot{Z})}{Z}. \tag{2.13}$$

The velocity of ${}^{C}P = [X, Y, Z]^T$ can be related to the camera spatial velocity using the following equation:

$${}^{C}\dot{P} = -v_c - w_c \times {}^{C}P \tag{2.14}$$

Then we can obtain:

$$\dot{X} = -v_x - w_y Z + w_x Y \tag{2.15}$$

$$\dot{Y} = -v_y - w_z X + w_x Z \tag{2.16}$$

$$\dot{Z} = -v_z - w_x Y + w_y X. \tag{2.17}$$

And substituting Equations for $\dot{X}, \dot{Y}, \dot{Z}$ into Equation 2.7 we have:

$$\dot{x} = \frac{-v_x}{Z} + \frac{xv_z}{Z} + xyw_x - (1 + x^2)w_y + yw_z \tag{2.18}$$

$$\dot{y} = \frac{-v_y}{Z} + \frac{yv_z}{Z} + (1 + y^2)w_x - xyw_y - xw_z. \tag{2.19}$$

In a matrix format we obtain:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J_p \begin{bmatrix} v_c \\ w_c \end{bmatrix}, \tag{2.20}$$

in which $v_c = [v_x \quad v_y \quad v_z]^T$ and $w_c = [w_x \quad w_y \quad w_z]^T$, J_p is the interaction matrix related to point *p* in the image plane.

$$J_p = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -(1 + x^2) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1 + y^2 & -xy & -x \end{bmatrix}. \tag{2.21}$$

In this interaction matrix, *Z* is the depth of the point with respect to the camera frame. Different control strategies that use this form of the interaction matrix must esti-

mate or approximate the value of depth which is normally time-varying and unknown especially in a monocular camera vision system. To control a 6 DOF system, we need at least three points ($k \geq 6$). In this case, there may exist some configurations of the system that makes J_P to be singular. Furthermore, there exist four various camera poses where $\boldsymbol{e} = \boldsymbol{0}$ and the time derivative of $\boldsymbol{e}$ will cause the singularities. Therefore, we need to consider more than three feature points in the feature vector. If we use the feature vector including four points in image plane, $s = (p_1, p_2, p_3, p_4)$ the Interaction or interaction matrices for three points would be:

$$J_p = \begin{bmatrix} J_{p1} \\ J_{p2} \\ J_{p3} \\ J_{p4} \end{bmatrix}, \tag{2.22}$$

where $J_p \in R^{8\times6}, J_{pi} \in R^{2\times6}, i = 1,2,\dots 4.$

We need to obtain the inverse of J_p which can be approximated by estimating J_p^+. One of the most popular scheme is to choose $\hat{J}_p^+ = J_p^+$ if J_p is known which means that the current depth *Z* of each point is available.

Another approach for estimating J_p^+ is to choose $\hat{J}_p^+ = J_{p^*}^+$, where $J_{p^*}^+$ is the value of J_p^+ for the desired position in which $e = e^* = 0$ [7]. In this case only the desired depth of each point has to be determined so all the 3D parameters are constant and there is no

need to estimate them. Recently, the choice of $\hat{J}_p^{+} = 1/2(J_{p^*} + J_p)^{+}$ has been proposed by Malis [54]. Since the term J_p is involved in this method, the current depth of each point *Z* must be available.

At the end, it is worth mentioning that it is also possible to define the kinematic modeling of the transformation between the image features' velocities and the joints' velocities as follows:

$$\dot{s} = J_T . \dot{q}, \tag{2.23}$$

where J_T is the "Total Jacobian" and defined as:

$$J_T = J_p . {}^{r}M_O . {}^{0}J_R \tag{2.24}$$

where J_p is the image interaction matrix for feature point-*p* , ${}^{r}M_O$ is the velocity transformation between the camera and world coordinate frames and ${}^{0}J_R$ is the robot Jacobian for the robotic manipulator. With a proportional visual servoing control law and assuming κ to be a positive gain, we now can obtain:

$$\dot{q} = -\kappa \hat{J}_T^{+} e \tag{2.25}$$

By tuning the gain κ, we can adjust the convergence speed of visual servoing.

2.2. Interaction Matrix for Image-Based Stereo Visual Servoing

2.2.1. Case I: Parallel Cameras

Having all the necessary information from classical monocular IBVS, we can now extend the approach and control scheme to a stereo image-based visual servoing (hereafter, S-IBVS). In the first case we consider our stereo-vision model composed of two parallel cameras which are perpendicular to the baseline. The focal points of two cameras are apart at distance *b/2* with respect to origin of sensor frame {C} on the baseline which means the origin of the camera frame, is in the center of these points. Focal distance of both cameras is *f* so the image planes and corresponding frames for left and right cameras are located with the distance *f* from the focal points and orthogonal to the optical axis. We assign {L} and {R} as the frames of the left and right images.

Figure 2.3 illustrates the case where both cameras observe a 3D point ${}^{C}P$.

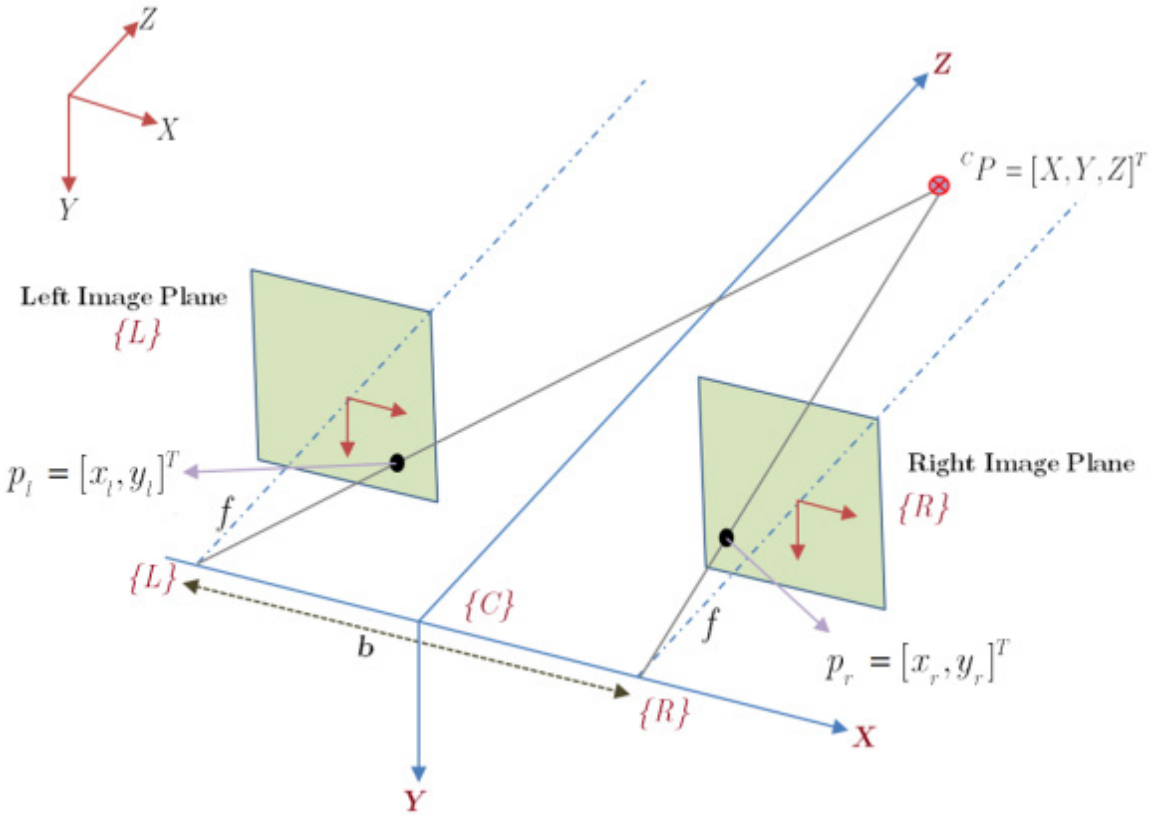


Figure 2.3 Model of the stereo vision system observing a 3D point

The feature vector is defined as $s = [x_l, y_l, x_r, y_r]^T$ where $p_l = [x_l, y_l]^T, p_r = [x_r, y_r]^T$ are the image coordinates of the 3D point, observed by the left and right cameras respectively. Now we can use the following equations to project observed point into left and right image planes;

$$x_l = \frac{X + b/2}{Z} = \frac{(u_l - \sigma_u)}{f^* \alpha} \tag{2.26}$$

$$y_l = \frac{Y}{Z} = \frac{(v_l - \sigma_v)}{f^*}, \tag{2.27}$$

$$x_r = \frac{X - b/2}{Z} = \frac{(u_r - \sigma_u)}{f^* \alpha} \tag{2.28}$$

$$y_r = \frac{Y}{Z} = \frac{(v_r - \sigma_v)}{f^*}, \tag{2.29}$$

where $p_l = [x_l, y_l]^T, p_r = [x_r, y_r]^T$ are the normalized coordinates from $[u_l, v_l]^T, [u_r, v_r]^T$, and f^* is focal length described in ***pixel dimensions.*** $(\sigma_u, \sigma_v, f, f^*, \alpha)$ is the set of camera intrinsic parameters: σ_u and σ_v are the coordinates of the camera principal point, f is the focal length, and α is the ratio of the pixel dimensions where $dy / dx = \alpha$.

Taking the time derivative of the perspective projection Equations (2.15), we can obtain:

$$\dot{x}_l = \frac{\dot{X}}{Z} - \frac{(X + b/2)\dot{Z}}{Z^2} = \frac{(\dot{X} - x_l \dot{Z})}{Z} \tag{2.30}$$

$$\dot{y}_l = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{(\dot{Y} - y_l \dot{Z})}{Z}. \tag{2.31}$$

$$\dot{x}_r = \frac{\dot{X}}{Z} - \frac{(X - b/2)\dot{Z}}{Z^2} = \frac{(\dot{X} - x_r\dot{Z})}{Z} \tag{2.32}$$

$$\dot{y}_r = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{(\dot{Y} - y_r\dot{Z})}{Z}. \tag{2.33}$$

The relation between a velocity of a feature point in an image p_I and a velocity of a feature point in a camera frame ${}^{C}P$ is given as:

$$\dot{p}_I = {}^{I}J_c \, {}^{C}\dot{P} \tag{2.34}$$

where;

$$p_I = [p_l, p_r]^T = [x_l, y_l, x_r, y_r]^T \tag{2.35}$$

and;

$${}^{I}J_c = \begin{bmatrix} \frac{\partial x_l}{\partial X} & \frac{\partial x_l}{\partial Y} & \frac{\partial x_l}{\partial Z} \\ \frac{\partial y_l}{\partial X} & \frac{\partial y_l}{\partial Y} & \frac{\partial y_l}{\partial Z} \\ \frac{\partial x_r}{\partial X} & \frac{\partial x_r}{\partial Y} & \frac{\partial x_r}{\partial Z} \\ \frac{\partial y_r}{\partial X} & \frac{\partial y_r}{\partial Y} & \frac{\partial y_r}{\partial Z} \end{bmatrix} \tag{2.36}$$

Consequently, using Equation (2.18) in (2.21) we obtain;

$$
{}^{I}J_{c} = \begin{bmatrix} \frac{1}{Z} & 0 & -\frac{X + b/2}{Z^2} \\ 0 & \frac{1}{Z} & -\frac{Y}{Z^2} \\ \frac{1}{Z} & 0 & -\frac{X - b/2}{Z^2} \\ 0 & \frac{1}{Z} & -\frac{Y}{Z^2} \end{bmatrix} \tag{2.37}
$$

As it is mentioned before, the velocity of ${}^{C}P = [X, Y, Z]^{T}$ can be related to the camera spatial velocity using the equation:

$$
{}^{C}\dot{P} = -w_{c} \times {}^{C}P - v_{c} \tag{2.38}
$$

So, we can obtain;

$$
\begin{aligned}
{}^{C}\dot{P} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} &= \begin{bmatrix} -w_{y}Z + w_{x}Y - v_{x} \\ -w_{z}X + w_{x}Z - v_{y} \\ -w_{x}Y + w_{y}X - v_{z} \end{bmatrix} \\
&= \begin{bmatrix} -1 & 0 & 0 & 0 & -Z & Y \\ 0 & -1 & 0 & Z & 0 & -X \\ 0 & 0 & -1 & -Y & X & 0 \end{bmatrix} \begin{bmatrix} v_{c} \\ w_{c} \end{bmatrix}
\end{aligned} \tag{2.39}
$$

Substituting Equation (2.17) in (2.21) we have:

$$\dot{p}_I = {}^I J_c \, {}^C\dot{P}$$

$$= {}^I J_c \begin{bmatrix} -1 & 0 & 0 & 0 & -Z & Y \\ 0 & -1 & 0 & Z & 0 & -X \\ 0 & 0 & -1 & -Y & X & 0 \end{bmatrix} u_c = J_{st} u_c \tag{2.40}$$

where $\boldsymbol{J_{st}}$ is the stereo-vision image interaction which expresses the relation between a velocity of a feature point in an image, $\dot{p}_I = [\dot{x}_l, \dot{y}_l, \dot{x}_r, \dot{y}_r]^T$ and a moving velocity u_c of a camera;

$$J_{st} = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{X+b/2}{Z^2} & Y\frac{X+b/2}{Z^2} & -(1+X\frac{X+b/2}{Z^2}) & \frac{Y}{Z} \\ 0 & \frac{-1}{Z} & \frac{Y}{Z^2} & 1+(\frac{Y}{Z})^2 & -\frac{XY}{Z^2} & -\frac{X}{Z} \\ \frac{-1}{Z} & 0 & \frac{X-b/2}{Z^2} & Y\frac{X-b/2}{Z^2} & -(1+X\frac{X-b/2}{Z^2}) & \frac{Y}{Z} \\ 0 & \frac{-1}{Z} & \frac{Y}{Z^2} & 1+(\frac{Y}{Z})^2 & -\frac{XY}{Z^2} & -\frac{X}{Z} \end{bmatrix} \tag{2.41}$$

From the model of the stereo vision projection Equations (2.15)-(2.18) the following equations hold for 3D coordinates of the observed point;

$$X = \frac{b}{2}\frac{x_l + x_r}{x_l - x_r} \tag{2.42}$$

$$Y = y_l \frac{b}{x_l - x_r} = y_r \frac{b}{x_l - x_r} \tag{2.43}$$

$$Z = \frac{b}{x_l - x_r} \tag{2.44}$$

Therefore, we can rewrite the stereo-vision image interaction matrix as:

$$J_{st} = \begin{bmatrix} -\frac{a}{b} & 0 & x_l \frac{a}{b} & x_l y & -(1 + \frac{x_l(x_l + x_r)}{2}) & y \\ 0 & -\frac{a}{b} & y\frac{a}{b} & 1 + y^2 & -y\frac{(x_l + x_r)}{2} & -\frac{(x_l + x_r)}{2} \\ -\frac{a}{b} & 0 & x_r \frac{a}{b} & x_r y & -(1 + \frac{x_r(x_l + x_r)}{2}) & y \\ 0 & -\frac{a}{b} & y\frac{a}{b} & 1 + y^2 & -y\frac{(x_l + x_r)}{2} & -\frac{(x_l + x_r)}{2} \end{bmatrix} \tag{2.45}$$

where $a = x_l - x_r$ is called ***feature point disparity*** and $y = y_l = y_r$. When we use *n* feature points, image interaction matrices ${}^1J_{st},...,{}^nJ_{st}$ in (2.31) are given from the coordinates of each feature points in an image. Therefore in case of *n*-feature points we can express the image interaction matrix as:

$$J_{st} = \begin{bmatrix} {}^1J_{st} \\ \vdots \\ {}^nJ_{st} \end{bmatrix} \tag{2.46}$$

One approach to stack stereo-vision image interaction matrix $\boldsymbol{J_{st}}$ is proposed in [38] in which the image interaction matrices of left and right cameras are calculated using the

spatial motion transform matrix. The velocities expressed in each camera frames are transformed to the sensor frame {C} and then stereo-vision image interaction matrix can be obtained;

$$\dot{s} = \dot{p}_I = \begin{pmatrix} J_l \, {}^{l}M_C \\ J_r \, {}^{r}M_C \end{pmatrix} u_c = J_{st} \, u_c \tag{2.47}$$

where u_c is the kinematic screw applied to the robot end-effector, J_l and J_r are the image interaction matrices related to the left and right cameras respectively, defined by Equation (2.12);

$$J_I = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x_I}{Z} & x_I y_I & -(1+x_I^{\,2}) & y_I \\ 0 & \frac{-1}{Z} & \frac{y_I}{Z} & 1+x_I^{\,2} & -x_I y_I & -x_I \end{bmatrix}, (I = l \ or \ r) \tag{2.48}$$

where ${}^{l}M_C$ and ${}^{r}M_C$ are the transformation matrices of the screw between the left and right camera frames and the sensor frame {C}. Given frames {C} and {$I = l \ or \ r$}, the relationship between the screws is:

$$u_I = {}^{I}M_C \, u_c \tag{2.49}$$

where the ${}^{I}M_C$ is defined as:

$${}^{I}M_C = \begin{pmatrix} {}^{I}R_C & [{}^{I}t_C]_{\times} \, {}^{I}R_C \\ O_{3\times3} & {}^{I}R_C \end{pmatrix} \tag{2.50}$$

In which $[{}^{I}t_{C}]_{\chi}$ is the skew-symmetric matrix associated with the vector *t* and (*R*, *t*) is the rigid body transformation from each camera to sensor frame {C}. Although the resulting interaction matrix J_{st} is the same as that obtained in Equation (2.27) or (2.31), it is easier for this approach to generalize to the other configurations of the cameras. With respect to Equation (2.30) the depth of the observed point can be estimated as:

$$\hat{Z} = \frac{b}{x_l - x_r} = \frac{bf^*}{u_l - u_r} \tag{2.51}$$

and note that $p_l = [x_l, y_l]^T, p_r = [x_r, y_r]^T$ are the normalized coordinates from $[u_l, v_l]^T, [u_r, v_r]^T$, and f^* is focal length described in **pixel** dimensions.

2.2.2. Case II: Non-Parallel Cameras

In the second case we built the stereo-vision model composed of two non-parallel cameras tilted by θ around Y-axis. This helps keeping the object in the cameras field of views during the servoing task especially in the desired position where end-effector and gripper are close to the target object. The focal points of two cameras are apart at distance *b/2* with respect to origin of camera frame or sensor frame {C} on the baseline. The focal distance for both cameras is *f* so the image planes and corresponding frames for left and right cameras are located at the position with the distance *f* from the focal points and or-

thogonal to the optical axis. We assign {L} and {R} as the frames of the left and right images. Figure 2.4 illustrates the case when both cameras observe a 3D point ${}^{C}P$.

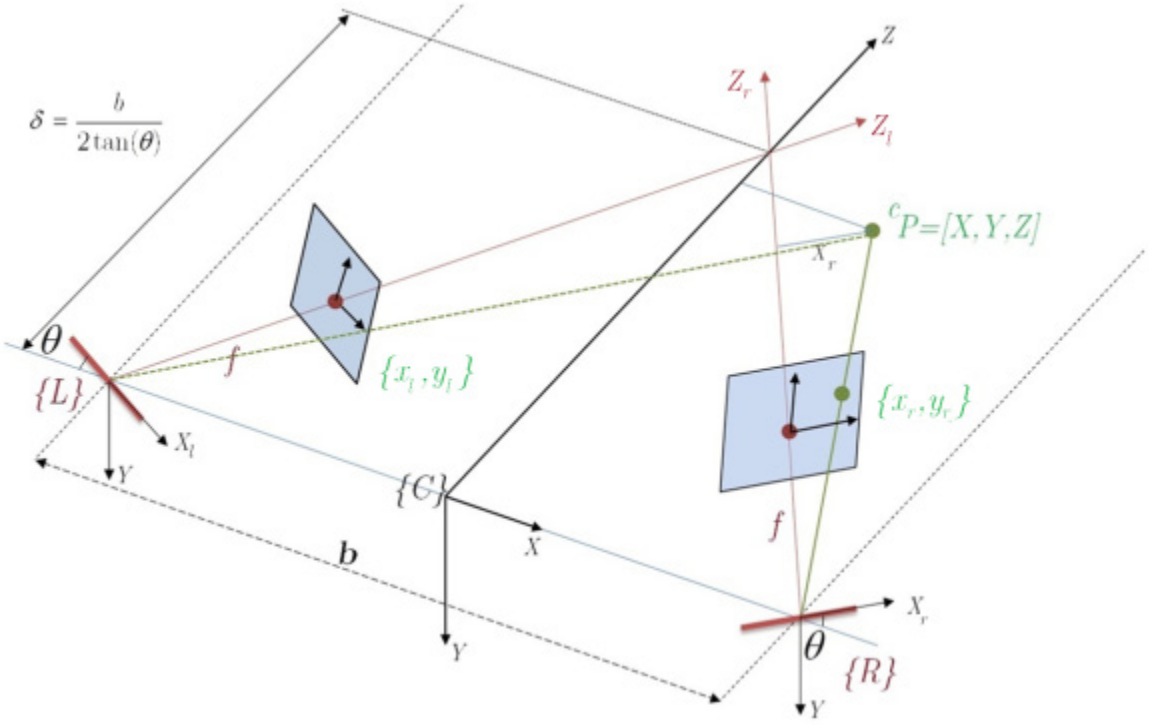


Figure 2.4 Stereo Vision model in a non-parallel camera configuration

As in Case I the feature vector is defined as $s = [x_l, y_l, x_r, y_r]^T$ where $p_l = [x_l, y_l]^T, p_r = [x_r, y_r]^T$ are the image coordinates of the 3D point, observed by the left and right cameras respectively. Now we can use the following equations to project the observed point into left and right image planes;

$$x_l = \frac{X_l}{Z_l} = \frac{(u_l - \sigma_u)}{f^* \alpha} \tag{2.52}$$

$$y_l = \frac{Y}{Z_l} = \frac{(v_l - \sigma_v)}{f^*}, \tag{2.53}$$

$$x_r = \frac{X_r}{Z_r} = \frac{(u_r - \sigma_u)}{f^* \alpha} \tag{2.54}$$

$$y_r = \frac{Y}{Z_r} = \frac{(v_r - \sigma_v)}{f^*}, \tag{2.55}$$

where $p_l = [x_l, y_l]^T, p_r = [x_r, y_r]^T$ are the normalized coordinates from $[u_l, v_l]^T, [u_r, v_r]^T$, and f^* is focal length described in ***pixel dimensions.*** Using the transformation equations in [55] from sensor frame {C} (X-Z plane) to right and left camera frames we have:

$$\begin{pmatrix} X \\ Z \\ 1 \end{pmatrix} = {}^{C}T_r \begin{pmatrix} X_r \\ Z_r \\ 1 \end{pmatrix} \tag{2.56}$$

$${}^{C}T_r = \begin{pmatrix} \cos\theta & -\sin\theta & b/2 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.57}$$

$$\begin{pmatrix} X \\ Z \\ 1 \end{pmatrix} = {}^{C}T_{l} \begin{pmatrix} X_{l} \\ Z_{l} \\ 1 \end{pmatrix} \tag{2.58}$$

$${}^{C}T_{l} = \begin{pmatrix} \cos\theta & \sin\theta & -b/2 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.59}$$

And we obtain the same coordinate (X,Z) in X-Z plane from both right and left camera information :

$$X = X_{r}\cos\theta - Z_{r}\sin\theta + \frac{b}{2} \tag{2.60}$$

$$Z = X_{r}\sin\theta + Z_{r}\cos\theta \tag{2.61}$$

$$X = X_{l}\cos\theta + Z_{l}\sin\theta - \frac{b}{2} \tag{2.62}$$

$$Z = -X_{l}\sin\theta + Z_{l}\cos\theta \tag{2.63}$$

Substituting (2.60) in (2.62) and (2.61) in (2.63) we can also obtain:

$$Z_r + Z_l = \frac{X_r - X_l}{\tan\theta} + \frac{b}{\sin\theta} \tag{2.64}$$

$$Z_r - Z_l = -(X_r + X_l)\tan\theta \tag{2.65}$$

Dividing (2.53) by (2.55) and also (2.52) by (2.54) we get:

$$\frac{Z_r}{Z_l} = \frac{y_l}{y_r} \tag{2.66}$$

$$\frac{X_r}{X_l} = \frac{x_r\, y_l}{x_l\, y_r} \tag{2.67}$$

Now we can solve (2.64-2.66), (2.52) and (2.54) for $X_r, X_l, Z_r, Z_l, \theta$. The results are as follows:

$$\theta = \tan^{-1}\left(\frac{y_r - y_l}{x_r y_l + x_l y_r}\right) \tag{2.68}$$

$$X_l = \frac{b}{\sin\theta}\frac{x_l y_r (y_r - y_l)}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \tag{2.69}$$

$$Z_l = \frac{b}{\sin\theta}\frac{y_r (y_r - y_l)}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \tag{2.70}$$

And consequently for right camera frame we have:

$$X_r = \frac{b}{\sin\theta}\frac{x_r\, y_l(y_r - y_l)}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \tag{2.71}$$

$$Z_r = \frac{b}{\sin\theta}\frac{y_l(y_r - y_l)}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \tag{2.72}$$

Substituting (2.70) in (2.53) we get the *Y* variable which is equal for right and left camera frames as:

$$\begin{aligned} Y &= \frac{b}{\sin\theta}\frac{y_r\, y_l(y_r - y_l)}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \\ &= \frac{b\, y_r\, y_l(x_r y_l + x_l y_r)\left(\dfrac{(y_r - y_l)^2}{(x_r y_l + x_l y_r)^2} + 1\right)^{1/2}}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \end{aligned} \tag{2.73}$$

Now substituting X_l, Z_l, θ to (2.47), (2.56) and (2.57) and simplifying the Equations we will obtain the exact position of the observed point as:

$$P = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \frac{b}{2A}\left[(x_r y_l - x_l y_r)^2 + (y_r - y_l)^2\right] \\ \frac{b}{A\sin\theta} y_r y_l (y_r - y_l) \\ \frac{b}{A} y_r y_l (x_r + x_l) \end{bmatrix} \tag{2.74}$$

where;

$$A = (y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2) \tag{2.75}$$

Expanding (2.68) we obtain:

$$P = \begin{bmatrix} \frac{b}{2}\left(\frac{(x_r y_l - x_l y_r)^2 + (y_r - y_l)^2}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)}\right) \\ \frac{b\, y_r\, y_l (x_r y_l + x_l y_r)\left(\frac{(y_r - y_l)^2}{(x_r y_l + x_l y_r)^2} + 1\right)^{1/2}}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \\ \frac{b y_r y_l (x_r + x_l)}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \end{bmatrix} \tag{2.76}$$

in which the depth value is derived:

$$Z = \frac{b y_r y_l (x_r + x_l)}{(y_r{}^2 - y_l{}^2) - ((x_r y_l)^2 - (x_l y_r)^2)} \tag{2.77}$$

This depth value for the observed point will be used for each feature point to calculate the image interaction matrix using either (2.30) and (2.34) or (2.41) and (2.44). Using (2.41) we can write the transformation matrices from left and right cameras to sensor frame {C} as follows:

$$ {}^{r}M_{C} = \begin{pmatrix} {}^{r}R_{C} & [{}^{r}t_{C}]_{\times}\,{}^{r}R_{C} \\ O_{3\times3} & {}^{r}R_{C} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{b}{2}\sin\theta & 0 & \frac{b}{2}\cos\theta \\ \sin\theta & 0 & \cos\theta & 0 & -\frac{b}{2} & 0 \\ 0 & 0 & 0 & \cos\theta & 0 & -\sin\theta \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.78} $$

$$ {}^{l}M_{C} = \begin{pmatrix} {}^{l}R_{C} & [{}^{l}t_{C}]_{\times}\,{}^{l}R_{C} \\ O_{3\times3} & {}^{l}R_{C} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{b}{2}\sin\theta & 0 & -\frac{b}{2}\cos\theta \\ -\sin\theta & 0 & \cos\theta & 0 & \frac{b}{2} & 0 \\ 0 & 0 & 0 & \cos\theta & 0 & \sin\theta \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.79} $$

Now we can write the image interaction matrices for left and right cameras as:

$$ J_{st} = \begin{pmatrix} J_{l}\,{}^{l}M_{C} \\ J_{r}\,{}^{r}M_{C} \end{pmatrix} \tag{2.80} $$

in which J_{l}, J_{r} are:

$$J_l = \begin{bmatrix} \frac{-1}{Z_l} & 0 & \frac{x_l}{Z_l} & x_l y_l & -(1+x_l^2) & y_l \\ 0 & \frac{-1}{Z_l} & \frac{y_l}{Z_l} & 1+x_l^2 & -x_l y_l & -x_l \end{bmatrix} \tag{2.81}$$

$$J_r = \begin{bmatrix} \frac{-1}{Z_r} & 0 & \frac{x_r}{Z_r} & x_r y_r & -(1+x_r^2) & y_r \\ 0 & \frac{-1}{Z_r} & \frac{y_r}{Z_r} & 1+x_r^2 & -x_r y_r & -x_r \end{bmatrix} \tag{2.82}$$

2.3. Comparison with the Monocular Visual Servoing

As it is discussed and formulated in Section 2.2, conventional monocular visual servoing techniques need a priori knowledge such as a model of the object or a depth parameter to calculate a fixed image interaction matrix or interaction matrix. This property seems to be quite unnatural in practice. Moreover, a fixed image interaction matrix may cause the process of the convergence to be relatively slow. On the contrary, it is possible to calculate the exact image interaction matrix at any position without using a model of the target object in stereo visual servoing. As it is shown in previous section, one of the main advantages of a stereo-vision system over a monocular vision system is the estimation of the depth.

Experiments show that in the monocular IBVS, the convergence speed is slow and overshoot is quite large in tracking. On the other hand, with stereo configuration it is fast in convergence and overshoot is small [56].

When using the monocular vision, at least 4 feature points are required to determine the relative location and orientation uniquely. On the other hand, three feature points seem to be sufficient in the case with the stereo vision where we can easily use more feature points to reduce the influence of noise on an image. The noise effect in monocular vision is more severe than that in stereo vision.

2.4. Summary

In this Chapter, an image-based visual servoing approach based on stereo vision has been introduced. At first, a review of classical monocular image based visual servo control (IBVS) has been presented. Then, an interaction matrix based on stereo vision is developed considering the modeling of two stereo images of a set of points and also extracting depth information.

Secondly, the classical monocular IBVS and the current Stereo visual servoing have been compared. It is concluded that the stereo IBVS method overcomes some problems in visual servoing using the monocular vision. Using the stereo vision makes it possible to calculate the exact image interaction matrix at any position without knowing a model of the target object, and results in better convergence characteristics. The stereo visual

servoing owns advantages over monocular servoing in many aspects such as 3D trajectory, movement smoothness and 3D convergence. [52]. The usage of stereo significantly increases the quality of the servo task execution, especially where the precision, robustness and smoothness of movement is concerned.

Chapter 3 : Trajectory Estimation and Position Prediction of a Moving Object

The processing of the vision system images and control strategy computations are time consuming that affect the information flow and cause delays in robot performance. This time delay in calculating the position of the moving object is one of the main causes of difficulties in the vision-based robotic object grasping. Thus, predicting the position of the moving object can help the system avoid such problems. It is concluded in Chapter 1 that contrary to the “Position-Based” method, “Image-Based” visual servoing control does not need the geometric model of the target or the precise camera calibration process to determine the kinematic relationship between the target and the robot. We also use a binocular vision system to overcome the target model, depth estimation and consequently stacking interaction matrix problems. But in practice we also encounter the problem of

noises that affect the image feature calculations such as camera lens distortion, physical noises or image signal quantization errors. To grasp a moving object firmly without stopping the motion, the robot control system needs to estimate positions at any instance or time interval accurately.

This chapter is dedicated to proposing a method for trajectory estimation of a moving object in order to clearly predict the position of the object in an image-based stereo visual servoing for a real-time grasping procedure. The system dynamics of the object is illustrated in both linear and nonlinear description. Various trajectory estimation algorithms such as “Kalman Filteri”, “Recursive Least Square” and “Extended Kalman Filter (EKF)” are used to predict the position of moving object in image planes.

3.1. Moving Object Trajectory Estimation in Image-Plane: Linear Approach

In order to provide essential position information for an image-based stereo visual servoing approach to grasp a moving object, it is possible to model the motion of the target in image-planes and to predict the trajectory and positions in a near future sequence. According to Equations (2.39) or (2.41) the stereo interaction matrices are computed and used in an image-based control scheme to re-position the image features to the desired positions. This procedure is illustrated in Figure 3.1.

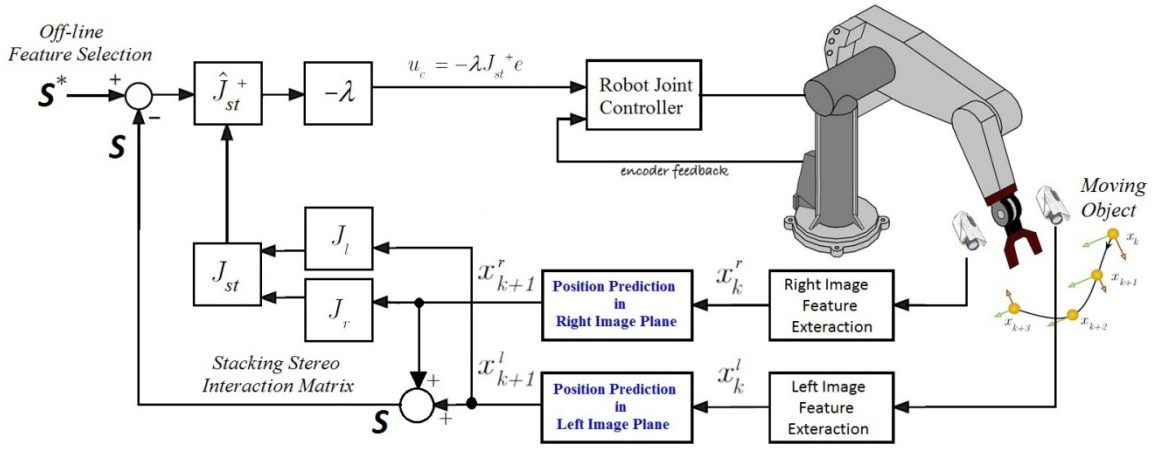


Figure 3.1 Using estimation of object positions in stereo image-plane for an image-based servoing approach to grasp a moving target

3.1.1. Moving Object Modeling

Based on the estimated velocity and acceleration of the moving object in right and left image planes and knowing the current position parameters, the estimated position of the object or the feature points in next instance, $(\hat{x}_k, \hat{y}_k)$ can be predicted as [57]:

$$\hat{x}_k = \hat{x}_{k-1} + \hat{v}_{x|k-1}.\Delta T + \frac{1}{2}\hat{a}_{x|k-1}.\Delta T^2 \tag{3.1}$$

$$dx_k = dx_{k-1} + d^2x_{k-1}.\Delta T \Rightarrow \hat{v}_{x|k} = \hat{v}_{x|k-1} + \hat{a}_{x|k-1}.\Delta T \tag{3.2}$$

$$\hat{y}_k = \hat{y}_{k-1} + \hat{v}_{y|k-1}.\Delta T + \frac{1}{2}\hat{a}_{y|k-1}.\Delta T^2 \tag{3.3}$$

$$dy_k = dy_{k-1} + d^2y_{k-1}.\Delta T \Rightarrow \hat{v}_{y|k} = \hat{v}_{y|k-1} + \hat{a}_{y|k-1}.\Delta T \tag{3.4}$$

where $(\hat{v}_{x|k}, \hat{v}_{y|k}), (\hat{a}_{x|k}, \hat{a}_{y|k}) = (dx_k, dy_k), (d^2x_k, d^2y_k)$ are the Cartesian estimated velocity and acceleration of the moving target and ΔT is the sampling period.

In order to model the moving object in a recursive procedure, Equations (3.1)-(3.4) should be expressed in form of the discrete time state transition and its observation models are as follows:

$$\mathrm{X}_k = \Phi_{k,k-1}\,\mathrm{X}_{k-1} + \mathrm{W}_{k-1}, \tag{3.5}$$

$$\mathrm{Z}_k = \mathrm{H}_k\mathrm{X}_k + \mathrm{V}_k, \tag{3.6}$$

where $\mathrm{X}_k = [x_k,\ y_k,\ dx_k,\ dy_k]^T$ is the state vector, $\mathrm{Z}_k = [x_k,\ y_k]^T$ is the measurement vector, Φ_k is the state transition matrix which represents the transition from one state vector X_{k-1} to the next vector X_k, W_k represents the process noises which is usually considered to be an "Additive Gaussian White Noise" (AWGN) with zero mean and covariance defined by:

$$\mathrm{E}[\mathrm{W}_k\ \mathrm{W}_k^{\,T}] = Q \tag{3.7}$$

and $\mathrm{V}_k = [\gamma_x, \gamma_y]^T$ are the measurement noises in both *x* and *y* direction and is also usually AGWN with zero mean and covariance defined by:

$$\mathrm{E}[\mathrm{V}_k \, \mathrm{V}_k^{\,T}] = R \tag{3.8}$$

H_k is called observation matrix and represents the relationship between the measurement and the state vector. Now we are able to obtain the observation models as follows:

$$\begin{bmatrix} x_k \\ y_k \\ dx_k \\ dy_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ dx_{k-1} \\ dy_{k-1} \end{bmatrix} + \begin{bmatrix} (d^2 x_{k-1} / 2)\Delta T^2 \\ (d^2 y_{k-1} / 2)\Delta T^2 \\ d^2 x_{k-1} \Delta T \\ d^2 y_{k-1} \Delta T \end{bmatrix} \tag{3.9}$$

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ dx_k \\ dy_k \end{bmatrix} + \begin{bmatrix} \gamma_x \\ \gamma_y \end{bmatrix} \tag{3.10}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.11}$$

The measurement vector $\mathrm{Z}_k = [x_k, \, y_k]^T$ is the actual position of an object or a feature point in x-y image planes in right and left cameras which could be obtained using the vision system.

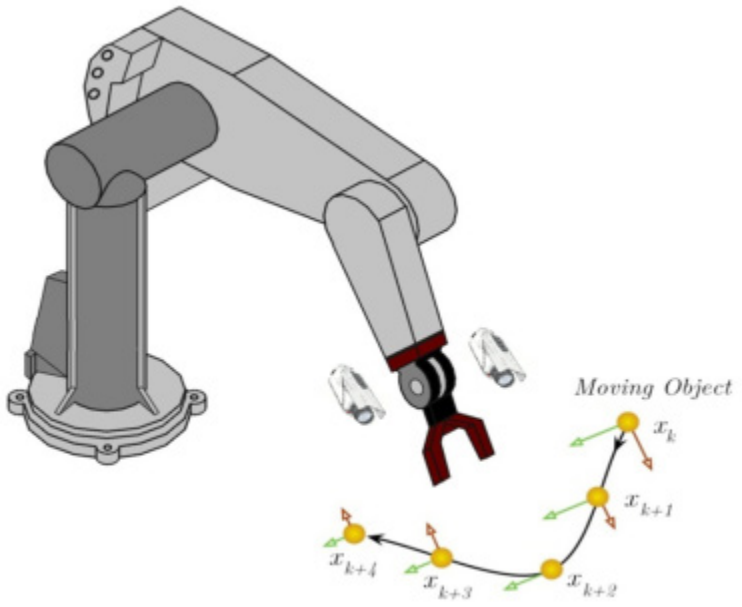


Figure 3.2 Estimation of the trajectory for grasping

3.1.2. Trajectory Estimation using Kalman Filter

The Kalman Filter was originally introduced by Rudolf Kalman at NASA to track the trajectory of spacecraft. Basically, the Kalman filter is a method of combining noisy measurements and predictions of the state of a system to achieve an estimate of its true current state [58]. Kalman filters can be applied to many different types of *linear* dynamical systems with various states such as an object's location, velocity, temperature, voltage, or a combination of these. It can be a helpful tool for tracking different types of moving objects.

The input image information includes uncertainties and noises took place in imaging device and pre-processing steps. The state transition of a moving object also includes irregular components. As a robust state estimator and based on this input image measurements, a Kalman filter can be used to estimate the state variables of the object e.g. $\mathrm{X}_k = [x_k, y_k, dx_k, dy_k]^T$ [59-61]. Using a proper filter gain, the Kalman filter performs a minimization for the estimation error by modifying the state transition model, Φ_k based on the error between the estimated vectors and the measured vectors. In the determination of a proper filter gain, a covariance matrix for estimated errors is calculated as follows:

$$P'_k = \Phi_{k,k-1} P_{k-1} \Phi^T_{k,k-1} + Q_{k-1}, \tag{3.12}$$

where P'_k is the zero-mean covariance matrix for the estimation error and P_{k-1} is the error estimate covariance matrix for the previous step. The error estimate covariance matrix P_k, is a measure of the estimated accuracy of x_k at time iteration *k* and is adjusted over time in the filtering procedure. We only need to consider a reasonable initial value as follows:

- If we know the exact initial state variables: $P_{k=0} = [0]_{4\times4}$
- If the exact initial state variables are unknown: $P_{k=0} = \sigma I_{4\times4}$ where $\sigma \gg 0$

The process noise matrix, *Q* measures the variability of the input signal away from the "ideal" transitions which is defined by the transition matrix. Larger values in *Q* can

make the input signal to have greater variance, thus the filter needs to be more adaptive. On the contrary, the smaller values in Q may cause the output to be smoother, but the filter adaptability to large changes is not quite high. Therefore, defining the process noise may require some fine tuning. Using Equation (3.7) and (3.9) and assuming $d^2x_t = d^2y_t = a$ the process noise matrix can be obtained as:

$$Q = \begin{bmatrix} \frac{1}{4}\Delta T^4 & 0 & 0 & 0 \\ 0 & \frac{1}{4}\Delta T^4 & 0 & 0 \\ 0 & 0 & \Delta T^2 & 0 \\ 0 & 0 & 0 & \Delta T^2 \end{bmatrix} \times a^2 \tag{3.13}$$

The optimal filter gain K_k that minimizes the state estimation errors can be obtained as follows:

$$K_k = P'_k H_k^T [H_k P'_k H_k^T + R_k]^{-1}, \tag{3.14}$$

where H_k is the observation matrix, and R_k is the zero mean covariance matrix of the measurement noise which defines the error of the measuring device. If we can assume that the measurements are enough accurate then small values may be used in R_k. Hence the filter performs less smoothing and the predicted signal will follow the observed signal more closely. Conversely, using great values in R_k means we have less confidence in the accuracy of the system measurements, so more smoothing is performed. For R_k we can obtain from v_k:

$$\mathrm{V}_k = \begin{bmatrix} \gamma_x \\ \gamma_y \end{bmatrix} \Rightarrow \; R_k = \begin{bmatrix} \gamma_x{}^2 & 0 \\ 0 & \gamma_y{}^2 \end{bmatrix} \tag{3.15}$$

The estimation of the state vector $\hat{\mathrm{x}}_k$ from the measurement Z_k is expressed as:

$$\hat{\mathrm{X}}_k = \Phi_{k,k-1}\,\hat{\mathrm{X}}_{k-1} + K_k[\mathrm{Z}_k - H_k\Phi_{k,k-1}\,\hat{\mathrm{X}}_{k-1}]. \tag{3.16}$$

Therefore, $\hat{\mathrm{X}}_k$ is updated based on the new values provided by Z_k. Before re-using Eq. (3.14) for the next step, the covariance matrix of the estimated error P_k, can be modified as follows:

$$P_k = P'_k - K_k H_k P'_k. \tag{3.17}$$

After the current time is updated to *k*+1, a new estimation can be provided performing the same procedure. Figure 3.3 below offers a complete picture of the operation of the Kalman filter.

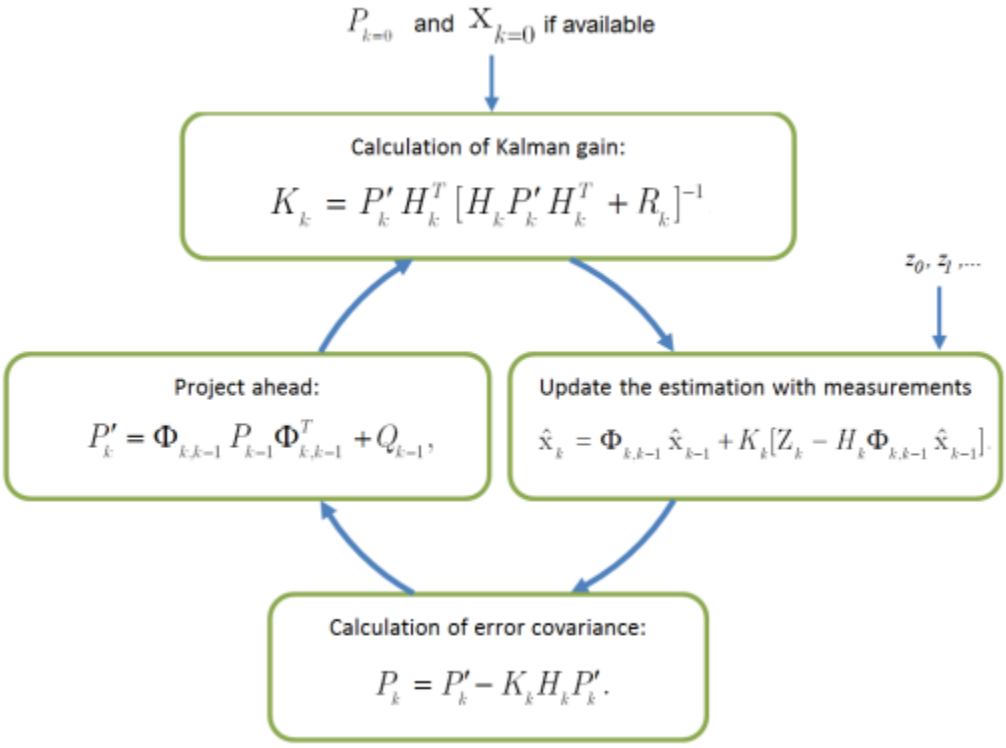


Figure 3.3 Kalman Filter Block Diagram

3.1.3. Trajectory Estimation using Recursive Least Square Method

If enough number of feature points projected in the camera plane are selected, a Recursive Least Squares method can be used to find the best estimation of the state variables of the object e.g. $X_k = [x_k, y_k, dx_k, dy_k]^T$ from the previous states data. The best estimation for time interval *k* can be computed as:

$$\hat{X}_k = \hat{X}_{k-1} + G_k[Z_k - H_k \hat{X}_{k-1}]. \quad (3.18)$$

where $\hat{\mathrm{X}}_k = [\hat{x}_k, \hat{y}_k, d\hat{x}_k, d\hat{y}_k]^T$, G_k is the optimal gain matrix and H_k is the observation matrix. The gain matrix can be computed as follows:

$$G_k = L_k H_k^T \tag{3.19}$$

where L_k is the error covariance matrix for the estimation of the state of time interval *k* and can be expressed as:

$$L_k = (\Phi^{-T}{}_{k,k-1}\ L_{k-1}^{-1}\ \Phi^{-1}{}_{k,k-1} + H_k^T H_k)^{-1}. \tag{3.20}$$

3.2. Moving Object Trajectory Estimation in Image-Plane: Nonlinear Approach

3.2.1. Moving Object Modeling

The motion of an object can be modeled on the *xy* –plane as discrete time-varying model [62]. Thus, we need to decompose the equations into the linear/angular motion elements where w_k and u_k are the angular and linear velocity components and where θ_k represents the direction of the moving object with respect to the *x*-axis on the *xy* -plane. In the next step, the position information can be predicted based on the current motion information such that;

$$x_k = x_{k-1} + u_{k-1} T \cos(\theta_{k-1} + \frac{1}{2} w_{k-1} \Delta T) \tag{3.21}$$

$$y_k = y_{k-1} + u_{k-1} T \sin(\theta_{k-1} + \frac{1}{2} w_{k-1} \Delta T) \tag{3.22}$$

For angular position and also noises, variations and uncertainties in measuring, the angular and linear velocity, we have:

$$\theta_k = \theta_{k-1} + w_{k-1} \Delta T, \tag{3.23}$$

$$u_k = u_{k-1} + \xi_u, \tag{3.24}$$

$$w_k = w_{k-1} + \xi_w, \tag{3.25}$$

where ξ_u and ξ_w can be considered as zero-mean Gaussian random variables. The non-linear state dynamics for the moving object can be expressed as:

$$\mathrm{X}_k = f(\mathrm{X}_{k-1}) + \mathrm{W}_{k-1}, \tag{3.26}$$

$$\mathrm{Z}_k = h(\mathrm{X}_k) + \mathrm{V}_k, \tag{3.27}$$

where $\mathrm{X}_k = [x_k, y_k, \theta_k, u_k, w_k]^T$ is the state vector, Z_k is the measurement vector, W_k represents process measurement noises and V_k is the vector that contains measurement errors in *x* and *y* direction, respectively. Using Equations (3.21-3.27) we can write:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ u_k \\ w_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \\ u_{k-1} \\ w_{k-1} \end{bmatrix} + \begin{bmatrix} u_k \Delta T \cos(\theta_k + \frac{1}{2} w_k \Delta T) \\ u_k \Delta T \sin(\theta_k + \frac{1}{2} w_k \Delta T) \\ w_k \Delta T \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \xi_u \\ \xi_w \end{bmatrix} \tag{3.28}$$

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = h(\mathrm{X}_k) + \begin{bmatrix} \gamma_x \\ \gamma_y \end{bmatrix} \tag{3.29}$$

The measurement vector Z_k is the actual position of an object or a feature point in x-y image planes in right and left cameras which could be obtained using the vision system.

3.2.2. Trajectory Estimation using Extended Kalman Filter (EKF)

The Kalman filter is generally used for estimating the state of a discrete-time controlled process that is governed by a *linear* stochastic differential equation. As an extension to the Kalman filtering, the extended Kalman filter (EKF) is proposed for the systems with *nonlinear* dynamics or output [58]. The Extended Kalman filter (EKF) provides an approximation of the optimal estimate. The non-linear systems dynamics are linearized around the last state estimate. One iteration of the EKF includes the following consecutive steps:

I. Consideration of the last state estimate $\hat{\mathrm{X}}_{k-1}$,

II. Linearizing the system dynamics $\mathrm{X}_k = f(\mathrm{X}_{k-1}) + \mathrm{W}_{k-1}$, around $\hat{\mathrm{X}}_{k-1}$,

III. Applying the predictions in the Kalman filter to the linearized system dynamics.

IV. Linearizing the observation dynamics, $\mathrm{Z}_k = h(\mathrm{X}_k) + \mathrm{V}_k$ around $\hat{\mathrm{X}}_k$,

V. Applying the update cycle of the Kalman filter to the linearized observations.

Let F_k and H_k be the Jacobian matrices of *f(x)* and *h(x)* in Equation (3.26) and (3.27) denoted by:

$$F_k = \nabla f_k \mid_{\hat{X}_{k-1}} \tag{3.30}$$

$$H_k = \nabla h_k \mid_{\hat{X}_{k-1}} \tag{3.31}$$

Thus, the nonlinear state dynamics for the moving object can be approximated with a linear system expressed as:

$$\mathrm{X}_k = F_{k-1}\mathrm{X}_{k-1} + \mathrm{W}_{k-1}, \tag{3.32}$$

$$\mathrm{Z}_k = H_k\mathrm{X}_k + \mathrm{V}_k, \tag{3.33}$$

Using Equations (3.32-3.33), and according to Equation (3.28-3.29) we can obtain:

$$F_k = \begin{bmatrix} 1 & 0 & -u_k \Delta T \sin(\theta_k) & \Delta T \cos(\theta_k) & -\frac{1}{2} u_k \Delta T^2 \sin(\theta_k) \\ 0 & 1 & u_k \Delta T \cos(\theta_k) & \Delta T \sin(\theta_k) & \frac{1}{2} u_k \Delta T^2 \cos(\theta_k) \\ 0 & 0 & 1 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.34}$$

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.35}$$

Thus we can write for the system state dynamics:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \\ u_k \\ w_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & -u_k \Delta T \sin(\theta_k) & \Delta T \cos(\theta_k) & -\frac{1}{2} u_k \Delta T^2 \sin(\theta_k) \\ 0 & 1 & u_k \Delta T \cos(\theta_k) & \Delta T \sin(\theta_k) & \frac{1}{2} u_k \Delta T^2 \cos(\theta_k) \\ 0 & 0 & 1 & 0 & \Delta T \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \\ u_{k-1} \\ w_{k-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \xi_u \\ \xi_w \end{bmatrix} \tag{3.3}$$

$$\begin{bmatrix} x_k \\ y_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \theta_k \\ u_k \\ w_k \end{bmatrix} + \begin{bmatrix} \gamma_x \\ \gamma_y \end{bmatrix} \tag{3.3}$$

The equations for Extended Kalman filter algorithm are stated as follows:

- **Prediction:**

$$\hat{\mathrm{X}}_k^- = f_{k-1}(\hat{\mathrm{X}}_{k-1}) \simeq F_{k-1}\hat{\mathrm{X}}_{k-1} \tag{3.38}$$

$$P_k' = F_{k-1}\,P_{k-1}\,F_{k-1}^T + Q_{k-1}, \tag{3.39}$$

- **Filtering:**

$$\begin{aligned}\hat{\mathrm{X}}_k &= \hat{\mathrm{X}}_k^- + K_k[\mathrm{Z}_k - h_k(\hat{\mathrm{X}}_k^-)] \\ &= \hat{\mathrm{X}}_k^- + K_k[\mathrm{Z}_k - H_k],\end{aligned} \tag{3.40}$$

$$K_k = P_k'\,H_k^T\,[H_k P_k'\,H_k^T + R_k]^{-1}, \tag{3.41}$$

$$P_k = P_k' - K_k H_k P_k'. \tag{3.42}$$

It this quite important to state that the EKF is not an optimal filter, and it is implemented based on a set of approximations. Despite the Kalman filter, the EKF may diverge, if the linearizations of the system and observation dynamics are not good in the entire associated uncertainty domain.

3.3. Summary

In this chapter two different approaches for estimating the trajectory of a moving object have been presented. In the first approach the motion of the target object was described in by the means of a linear model and a Kalman filter and a Recursive Least square method have been utilized to predict the position of the target. In the second approach a non-linear model has been considered and an Extended Kalman Filter has been used to estimate the moving target trajectory. The predicted position information is used to calculate the image interaction matrices in an image-based visual servoing system.

Chapter 4 : System Modeling and Simulations

In this chapter modeling of the robotic stereo visual servoing system and also simulation results for developed methods and algorithms are presented. First, using the specifications of the experimental setup, the modeling and frame definitions of the proposed visual servoing system are presented. Then in the simulation part, the effectiveness of the image-based stereo visual servoing system is validated and compared to a monocular system in a simple servoing task with fixed image features. Then the system performance in a task of tracking and grasping a moving object is examined and the results for utilizing both Kalman filter and recursive least square method for predicting the position and trajectory of the maneuvering target are presented and discussed.

4.1. Stereo Visual Servoing Framework Modeling

In order to design and perform a precise computer simulation for our stereo visual servoing system, it is essential to define a proper and detailed modeling framework. First, based on the parameter definitions and modeling shown in previous chapters, a schematic of Denso-the experiment robot, the attached stereo rig and also axis definitions and transformations will be introduced. Figure 4.1 shows the robotic system with an attached stereo vision system and also the frames on the stereo rig.

It is essential to note that $\{E\} = \{x_e, y_e, z_e\}$ is the end effector frame, $\{C\} = \{X_C, Y_C, Z_C\}$ is the sensor frame which coincides the end effector and $\{X_r, Y_r, Z_r\}$ and $\{X_l, Y_l, Z_l\}$ are the right and left camera frames. In order to derive Equation (2.47) to stack the stereo interaction matrices we need to calculate the transformation matrices from left and right cameras to the sensor frame. In order to control the robot joints for the visual servoing task, we also need the same transformations from sensor frame to the end-effector frame.

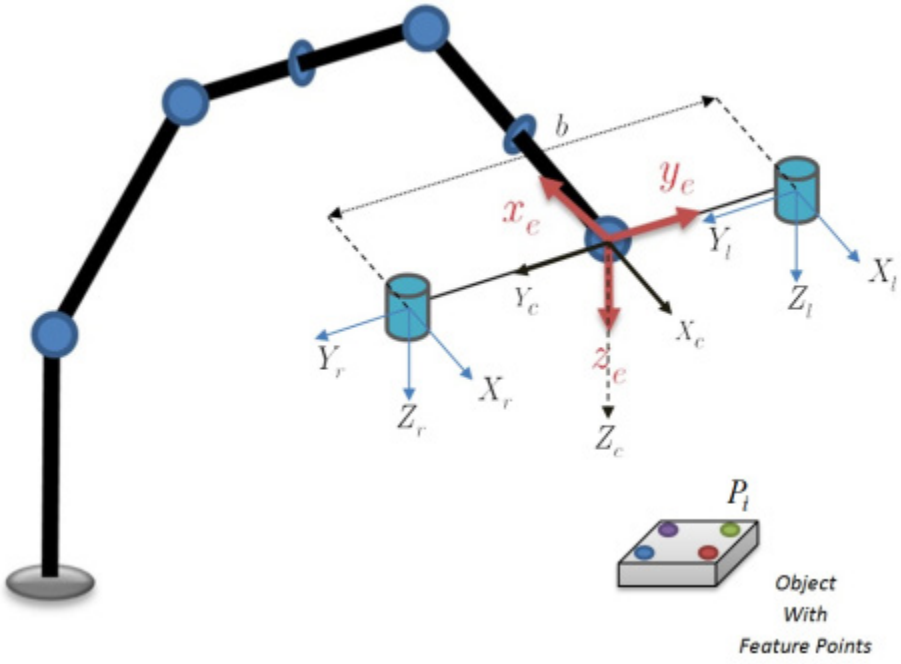


Figure 4.1 Stereo visual servoing framework

Considering ${}^{R}P_i$ and ${}^{L}P_i$ to be the 3D positions of a feature point on the object with regards to the camera frames, we can obtain the 3D position of the same feature point with regards to the sensor frame {C} as follows:

$$ {}^{C}P_i = {}^{C}_{R}R \ {}^{R}P_i + {}^{C}P_{RORG} \tag{4.1} $$

$$ {}^{C}P_i = {}^{C}_{L}R \ {}^{L}P_i + {}^{C}P_{LORG} \tag{4.2} $$

In the case of the parallel cameras and according to the framework geometry we can obtain:

$$ {}^{C}_{R}R = I_{3\times3} \ . \qquad {}^{C}P_{RORG} = [\,0 \quad \frac{b}{2} \quad 0\,]^{T}. $$

$$ {}^{C}_{L}R = I_{3\times3} \ . \qquad {}^{C}P_{LORG} = [\,0 \quad -\frac{b}{2} \quad 0\,]^{T}. $$

Having the transformations from sensor frame to the end-effector frame we can obtain the feature points positions with regards to the end effector frame as follows:

$$ {}^{E}P_{i} = {}^{E}_{C}R \ {}^{C}P_{i} + {}^{E}P_{CORG} \tag{4.3} $$

where according to Figure 4.1 for the rotation and translation we obtain:

$$ {}^{E}_{C}R = {}^{E}_{R}R = {}^{E}_{L}R = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \qquad {}^{E}P_{CORG} = [\,0 \quad 0 \quad 0\,]^{T}. $$

If the Right and Left cameras are tilted about their X axis with θ and $-\theta$ radians, for rotation matrices we obtain:

$$ {}^{C}_{R}R = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}. $$

$$ {}^{C}_{L}R = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix}. $$

4.2. Simulation Results

In this section, the simulation results on the proposed image-based stereo visual servoing system are presented and the performances of the system for two different tasks are tested and compared with a monocular visual servoing system. In the first part all the results are based on a simple image-based task with fixed feature points. Afterwards, the results on a task of grasping a moving object with a sinusoidal trajectory by the 6-DOF Denso robot are presented and discussed.

The modeling for camera/vision system, robotic manipulator and visual servoing control systems are carried out using Robotics and Machine Vision Toolboxes [63, 64] from *Peter I. Corke* [65].

4.2.1. Image-Based Visual Servoing: Stereo VS. Monocular

In this section, the effectiveness of the proposed eye-in-hand image-based stereo visual servoing system in an uncalibrated environment is examined by comparing it to a conventional monocular eye-in-hand system. As in Equation (2.4) in the case of a sta-

tionary object with fixed feature points, the control scheme computes a camera velocity as follows:

$$u_c = -\lambda J_e^{+}(s - s^{*}) \tag{4.4}$$

For the simulation purposes, a model of the experimental robot is designed with the real parameters and frame definitions. Figure 4.2 shows the "DENSO" robot simulation model.

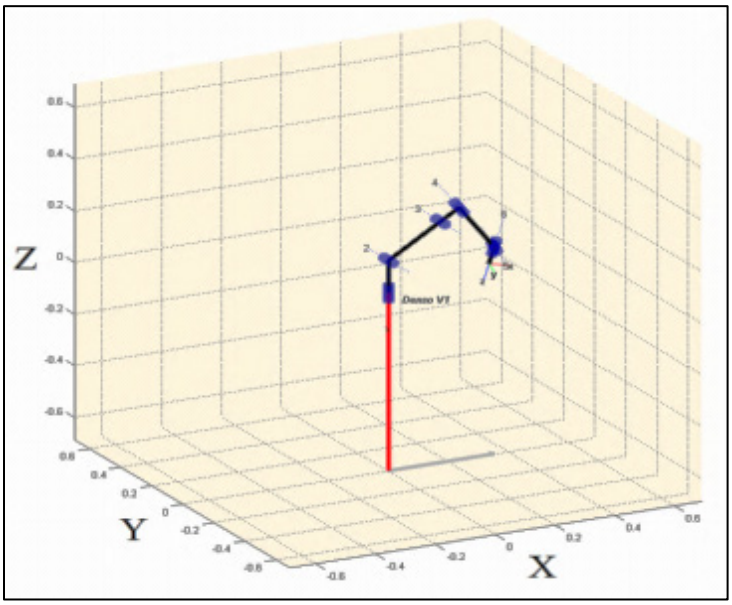


Figure 4.2 Experimental robot simulation 3D-model

Figures 4.3-4.6 show the results obtained from monocular image-based system. The following assumptions have been also taken into account:

Proportional Gain: $\lambda = -0.4$

Initial feature points: $s(t=0) = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix} = \begin{bmatrix} 382 & 641 & 382 & 641 \\ 382 & 382 & 642 & 642 \end{bmatrix}$ Pixels

Desired feature points: $s^* = \begin{bmatrix} p_1^* & p_2^* & p_3^* & p_4^* \end{bmatrix} = \begin{bmatrix} 517 & 848 & 578 & 941 \\ 532 & 463 & 867 & 818 \end{bmatrix}$ Pixels

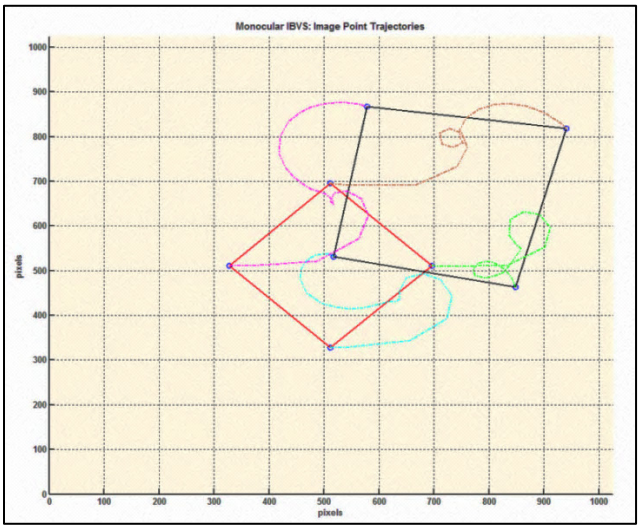


Figure 4.3 Image point trajectories in a monocular IBVS task

It can be inferred from the image feature trajectories that the features of system finally converges to the desired features but it is quite noticeable from Figure 4.3 that there are large displacements for the feature points and consequently the end-effector (see Figure 4.6) to reach the desired positions.

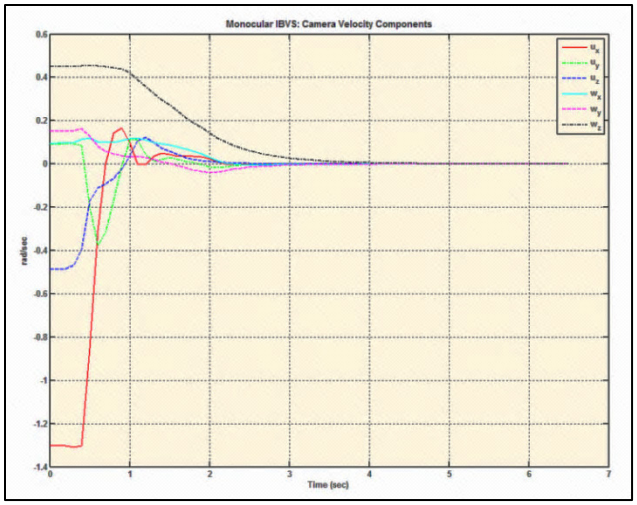


Figure 4.4 Camera velocity components in a monocular IBVS task

The behavior in the image feature errors (Figure 4.5) and also the computed camera velocity components does not present desirable properties for the current monocular IBVS system. It is essential to mention that the point depths in calculating image interaction matrices are assumed to be constant and equal to 1m. Using different depth estimation methods may cause the system to show different performance in servoing tasks.

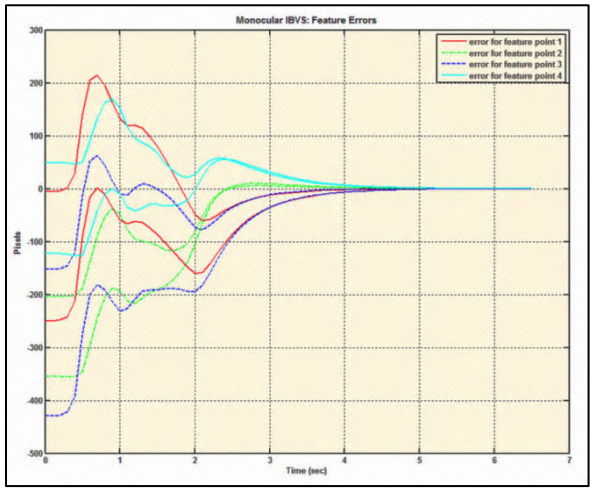


Figure 4.5 Image feature errors in a monocular IBVS task

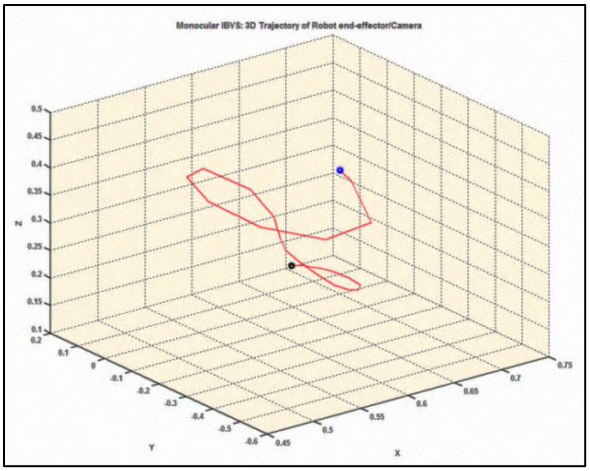
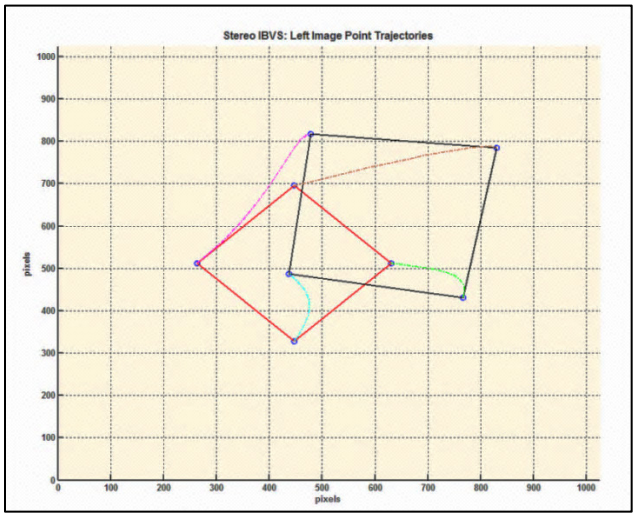
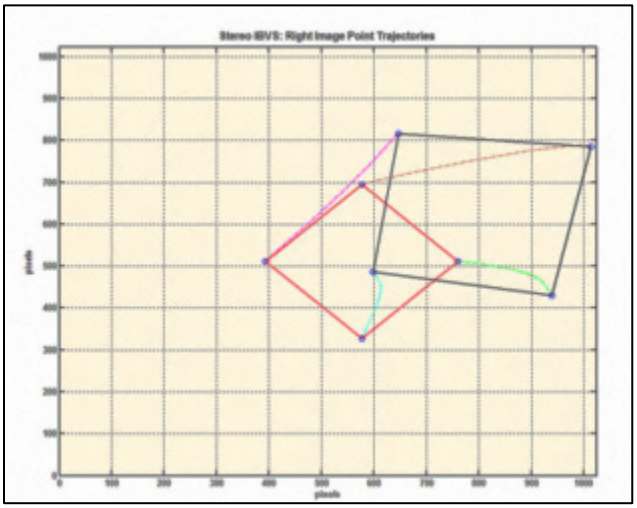


Figure 4.6 3-D Trajectory of the end-effector and camera optical center in a monocular IBVS task

The same task with the same assumptions was assigned to the Stereo IBVS system to achieve to two desired sets of feature points for right and left images. The proportional gain is chosen to be $\lambda = -0.4$ again. Figure 4.7 shows the results for Image feature trajectories for Left and right image planes in a stereo image-based visual servoing task.



(a)



(b)

Figure 4.7 Image feature trajectories for (a) Left image plane and (b) Right image plane in a Stereo IBVS task

Comparing to the complex and meandrous image feature trajectories in the monocular system, the provided trajectories for the image features in the stereo system seem to be quite smoother.

Figure 4.8 illustrates the sensor frame velocity components on the stereo rig. The sensor frame is the frame in the center of the baseline between the camera frames. The cameras are apart at a distance of *b/2* with respect to origin of sensor frame and parameter b is chosen to be 11 cm. According to Figure 4.8 the sensor frame velocity components do not include large oscillations compared to the camera velocity components in monocular system.

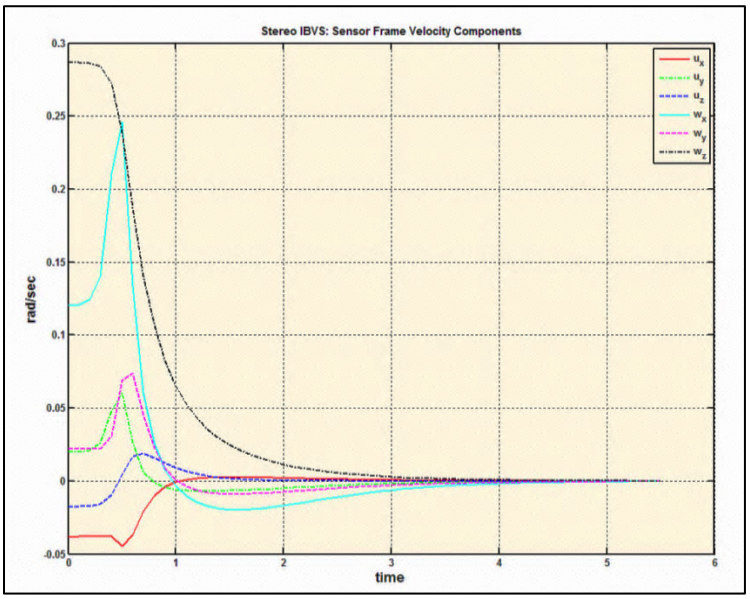
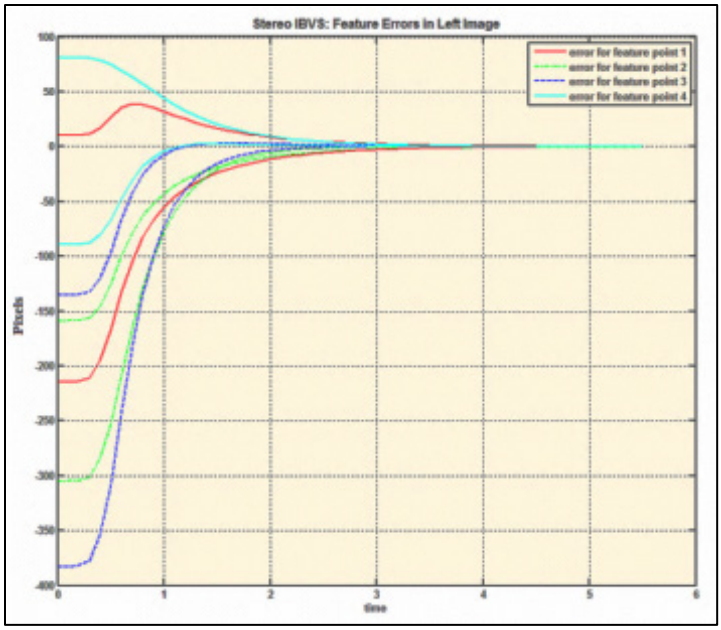
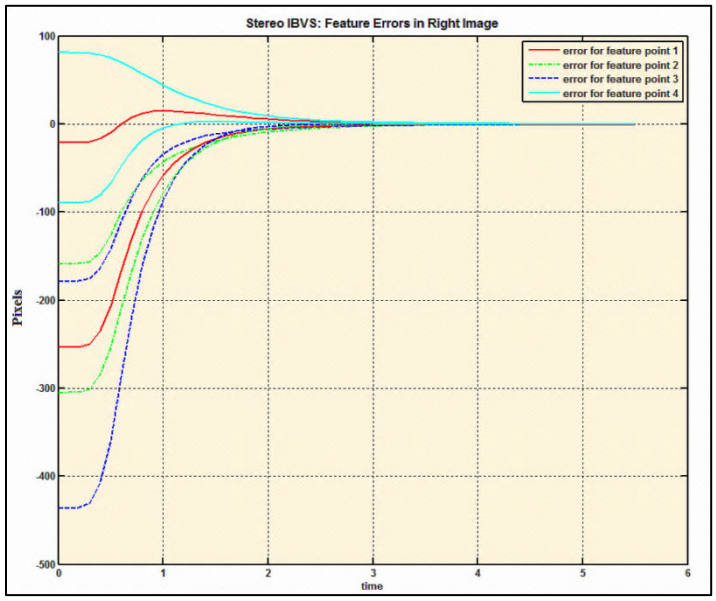


Figure 4.8 Sensor frame velocity components in a Stereo IBVS task

Feature errors in left and right images are illustrated in Figure 4.9. Compared to the monocular case, these errors traverse a smoother trajectory and also include less oscillations and overshoots as well.



(a)



(b)

Figure 4.9 Feature errors in (a) left and (b) right images in the Stereo IBVS task

Figure 4.10 shows the 3D end-effector trajectory during the task.

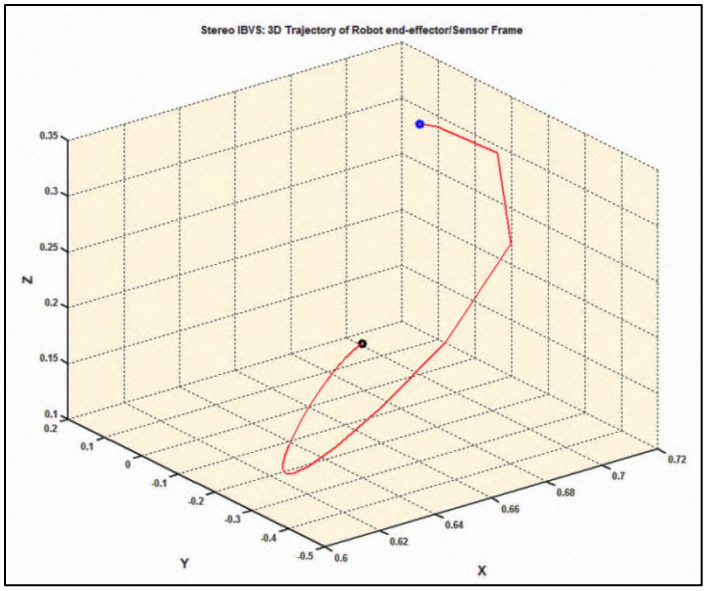


Figure 4.10 3-D Trajectory of the end-effector and sensor frame in a Stereo IBVS task

Finally, we can see from the results that in the case of the monocular vision, the system is slow in convergence and overshoot is large. On the contrary, it is fast in convergence and overshoot is small in the case of the stereo visual servoing. This is because in stereo vision case the image interaction matrices can be updated at every time interval using the calculated depth information. Thus it is possible to generate the correct feedback command which leads the visual servoing system to be more stable.

4.2.2. Tracking and Grasping a Moving Object through Visual Servoing: Stereo VS. Monocular

In this section, the performance of the robotic manipulator system with of the proposed eye-in-hand image-based stereo visual servoing system to track and grasp a mov-

ing object is examined by comparing it to a conventional monocular system in three cases of using Kalman Filter, Extended Kalman Filter (EKF) and a Recursive Least Square method (RLS) to predict object trajectory and position.

In the case of a moving object the control scheme uses an integral term besides the traditional proportional controller and computes the camera velocity as follows:

$$u_c = \widehat{J_e^+} \left(-\lambda e - \mu_I \sum_{i=0}^{k-1} e_i \right) \tag{4.5}$$

For the purpose of simulation the following parameters are chosen for the best results:

Proportional Gain: $\lambda = -3$

Integral Gain: $\mu_I = -0.5$

Object linear velocity $U_{obj} = 2\ cm / \sec$

Desired feature points: $s^* = \begin{bmatrix} p_1^* & p_2^* & p_3^* & p_4^* \end{bmatrix} = \begin{bmatrix} 134 & 936 & 134 & 936 \\ 143 & 58 & 881 & 966 \end{bmatrix}$ Pixels

For the presented simulations we test the grasping algorithm with an object in a sinusoidal motion with linear velocity of 2 cm/sec. As it was mentioned previously, the tracking and grasping task is performed by pre-defining desired positions for the object image features such that the robot moves and aligns the end-effector with the object and

reaches towards it. Figure 4.11 shows the simulation camera view of the desired image feature points in the instant of grasping in the case of a monocular system.

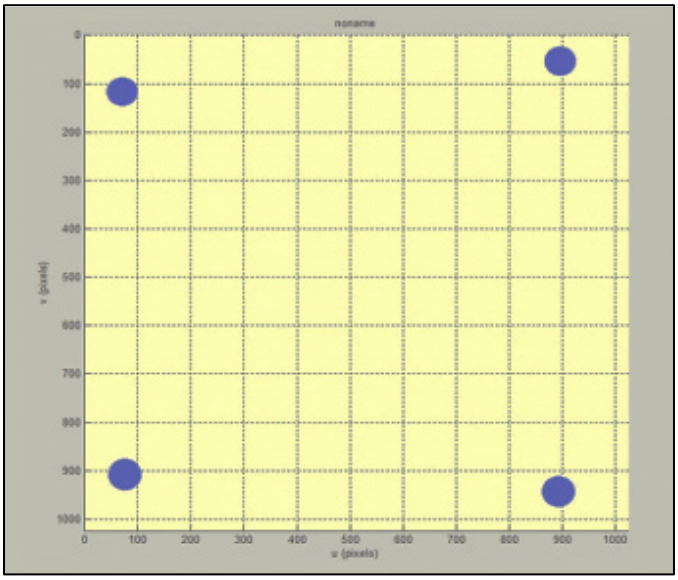


Figure 4.11 Desired image feature points for grasping task in a Monocular case

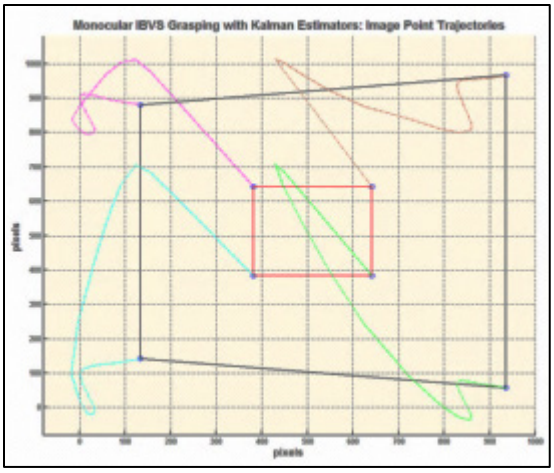
The results of using a monocular visual servoing system with a Kalman estimator for grasping a moving object and the image feature prediction errors are also shown in Figure 4.12. This feature prediction error is the difference between actual positions of projected points in the image plane and predicted ones. According to Equation (3.13) and considering $a = 0.1\ (m / \sec^2)$ and $\Delta T = 1\ (\sec)$ the process noise matrix can be obtained as:

$$Q = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times 10^{-2} \qquad (4.6)$$

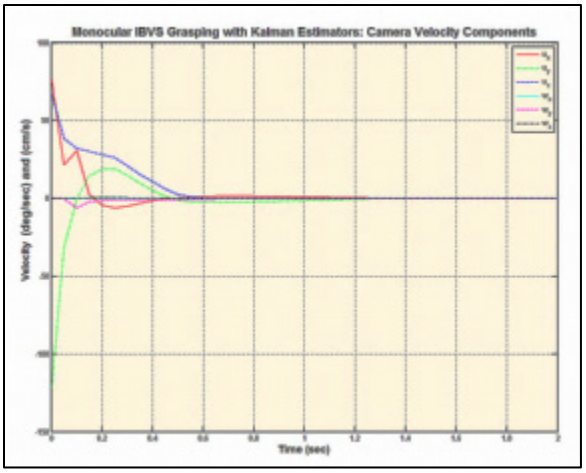
The covariance matrix of the measurement noise and the initial condition for estimated error covariance are assumed to be as follows:

$$R = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \tag{4.7}$$

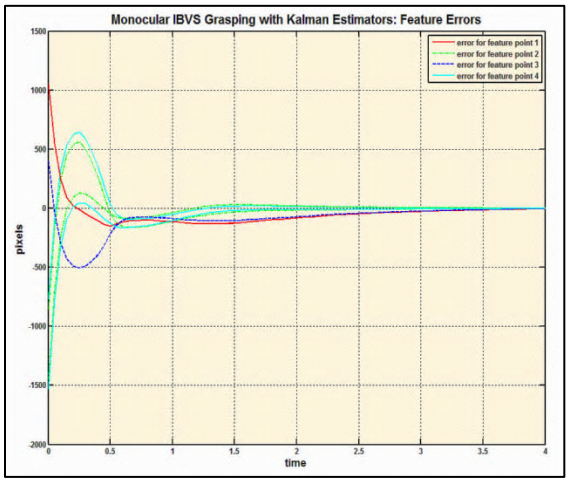
$$P = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \tag{4.8}$$



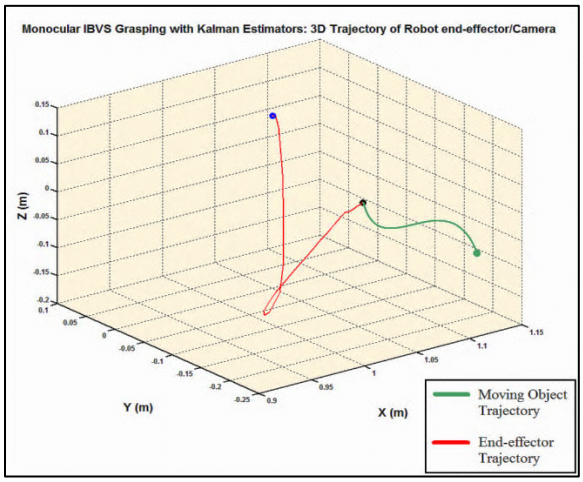
(a)



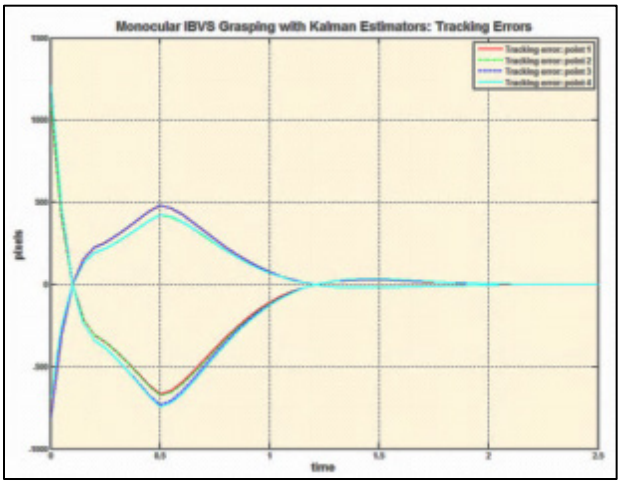
(b)



(c)



(d)

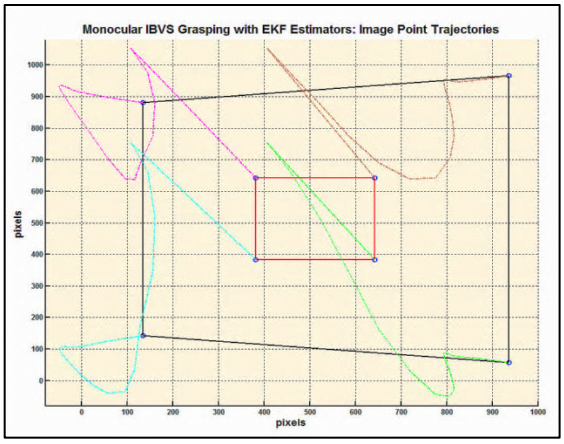


(e)

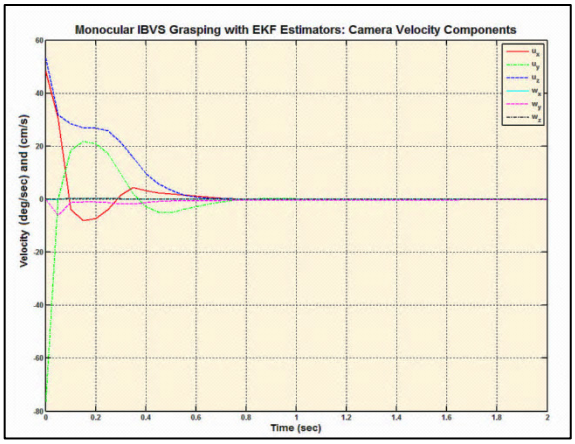
Figure 4.12 Monocular IBVS system behavior in a procedure of grasping a moving object using **Kalman** Estimator: (a) Image feature trajectories (b) Camera frame velocity components (c) Image feature errors (d) Robot end-effector 3D positions (e) Tracking Errors

Figure 4.13 illustrates the results of using a monocular visual servoing system with an Extended Kalman Filter (EKF) estimator for grasping the same moving object. For the filtering parameters a Gaussian random noise covariance of 5 pixel/sec and 10 deg/sec are added to the linear velocity and the angular velocity of the moving object trajectory. Therefore, the state transition noise covariance matrix, Q, and the measurement noise covariance, R , can be expressed as in Equation (4.6)

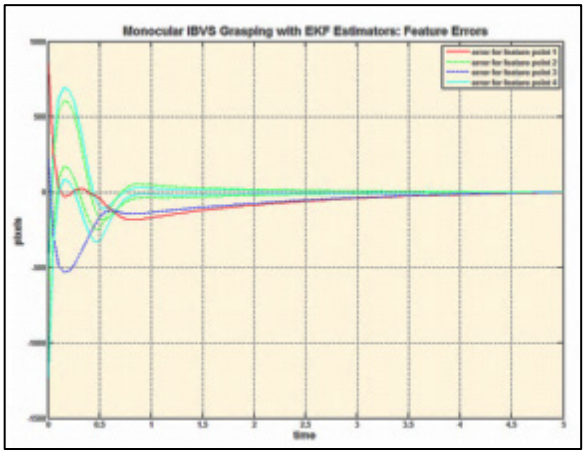
$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 10\dfrac{\pi}{180} \end{pmatrix}, \quad R = I_{2\times 2} \tag{4.9}$$



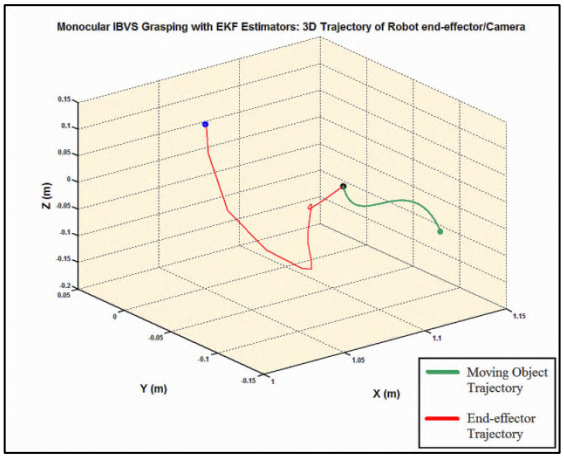
(a)



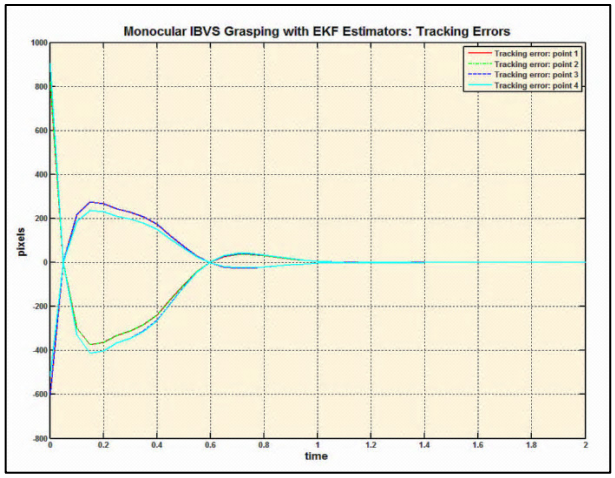
(b)



(c)



(d)



(e)

Figure 4.13 Monocular IBVS system behavior in a procedure of grasping a moving object using **Extended Kalman Filter** (EKF) Estimator: (a) Image feature trajectories (b) Camera frame velocity components (c) Image feature errors (d) Robot end-effector 3D positions (e) Tracking Errors

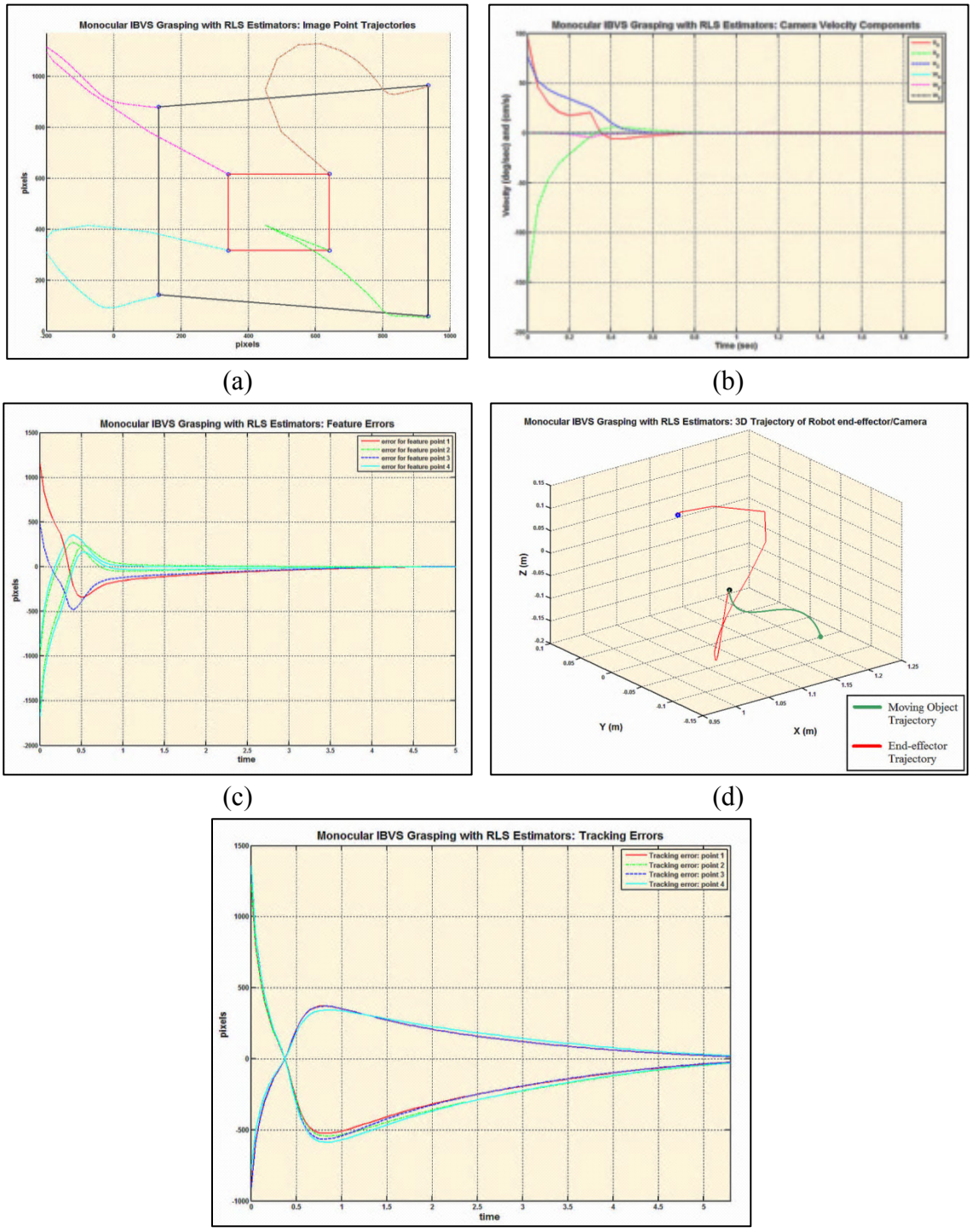
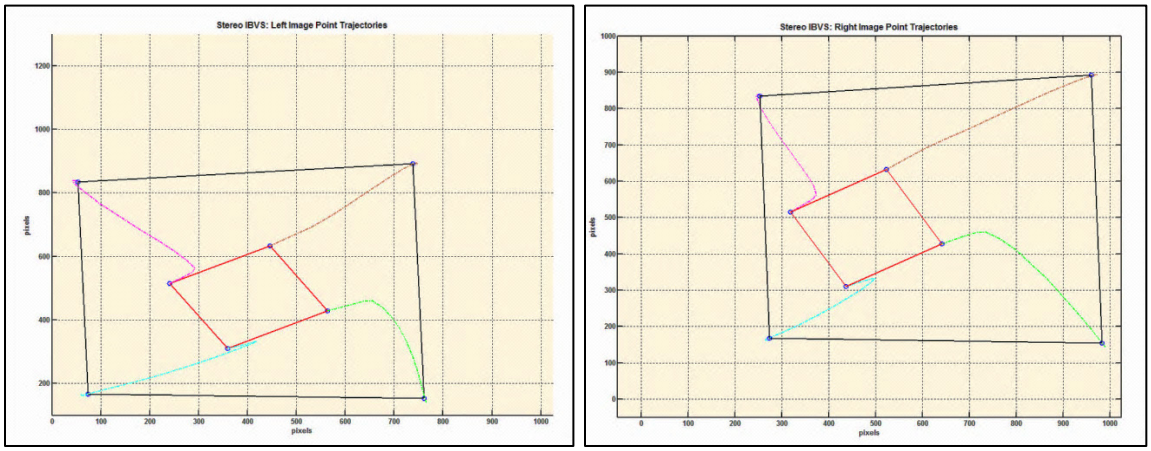


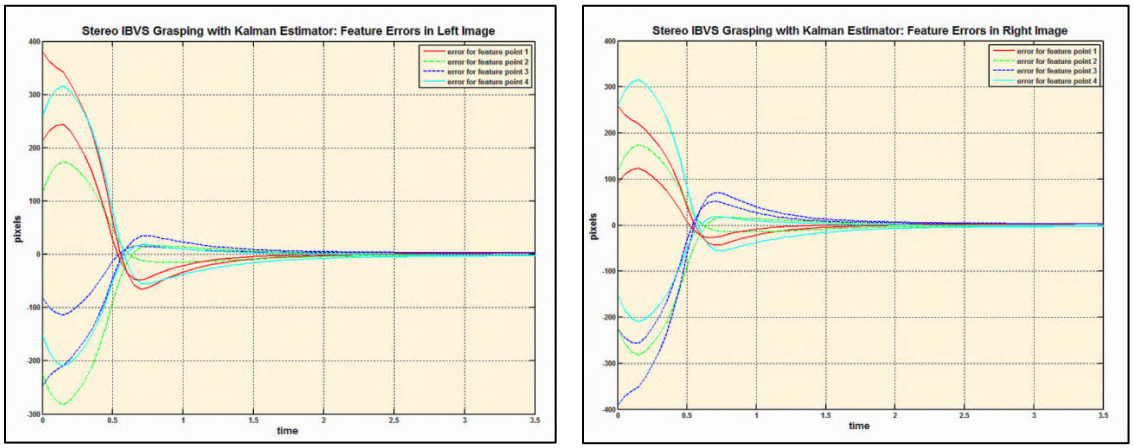
Figure 4.14 Monocular IBVS system behavior in a procedure of grasping a moving object using **RLS** Estimator: (a) Image feature trajectories (b) Camera frame velocity components (c) Image feature errors (d) Robot end-effector 3D positions (e) Tracking Errors

Figure 4.14 also shows the results of using a monocular visual servoing system with a Recursive Least Square estimator for grasping the same moving object. From the simulation results, Figures 4.12-4.14, for a tracking and grasping task, the pixel error due to the measurement noise could be considerably reduced by using EKF, Kalman Filter or Recursive Least Square (RLS) algorithms based on the moving object model. It can be inferred from the simulation results that the Extended Kalman estimator shows the best tracking and convergence performance among all. In addition, the end-effector 3-D position diagrams show that the system with EKF estimator has a better behavior than the other two methods. The camera velocity components in the system with EKF in comparison with the system with RLS estimator started with relatively low speeds. On the other hand, the image feature trajectory behaviors in all of three systems seem to be almost the same.

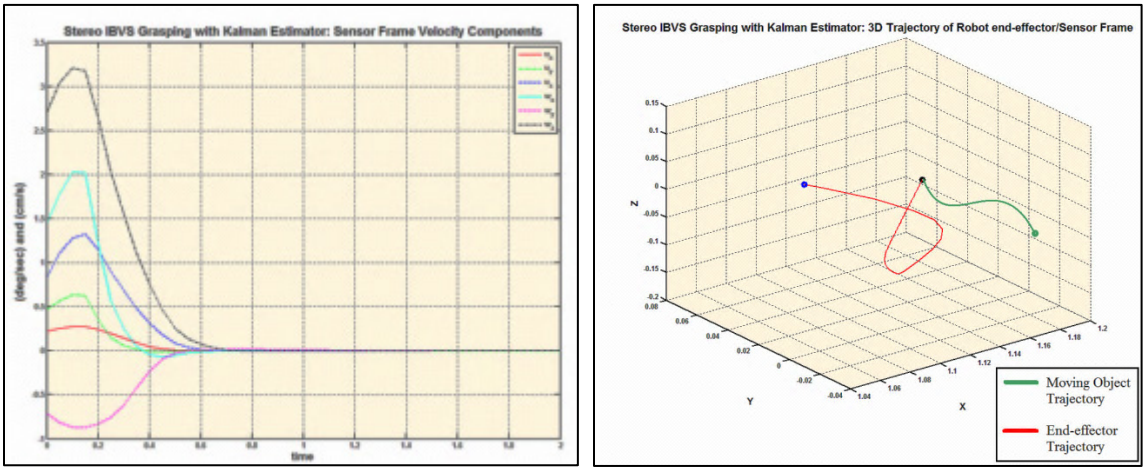
Using the same filtering, the state transition noise covariance and the measurement noise covariance properties which were given in Equations (4.6-4.9), the same grasping procedure is simulated with the proposed stereo visual servoing system and an EKF, a Kalman and a RLS estimator. The stereo system consists of two parallel cameras which are located at a distance of *b/2* with respect to the origin of sensor frame. In order to keep at least 3 selected feature points on the object in both cameras' fields of views during the approaching phase, the distance *b* is selected to be equal to 8 cm. The simulation results for this stereo vision system with parallel cameras and a Kalman estimator are given in Figure 4.15.


Stereo IBVS: Left Image Point Trajectories
pixels
pixels
Stereo IBVS: Right Image Point Trajectories
pixels
pixels

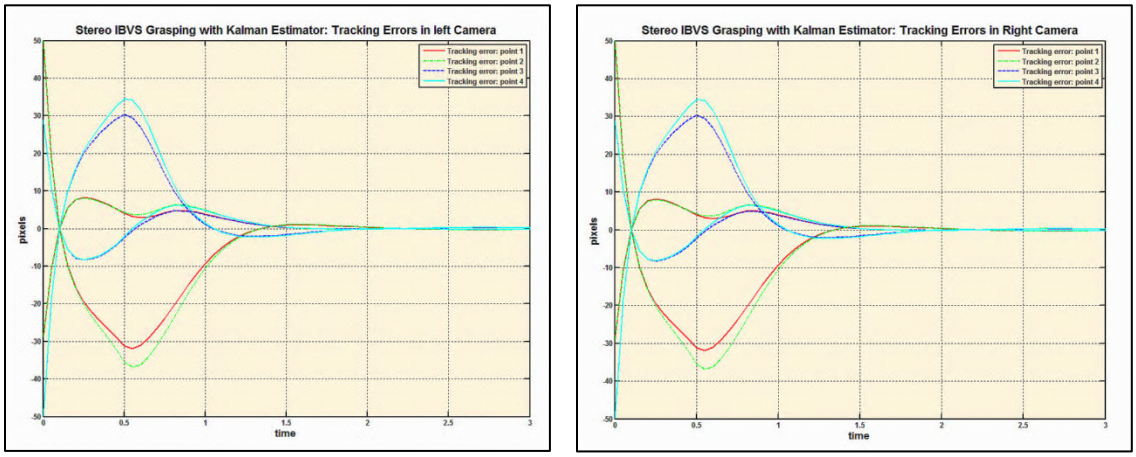
(a)


Stereo IBVS Grasping with Kalman Estimator: Feature Errors in Left Image
error for feature point 1
error for feature point 2
error for feature point 3
error for feature point 4
pixels
time
Stereo IBVS Grasping with Kalman Estimator: Feature Errors in Right Image
error for feature point 1
error for feature point 2
error for feature point 3
error for feature point 4
pixels
time

(b)


Stereo IBVS Grasping with Kalman Estimator: Sensor Frame Velocity Components
time
Stereo IBVS Grasping with Kalman Estimator: 3D Trajectory of Robot end-effector/Sensor Frame
Z
Y
X
Moving Object Trajectory
End-effector Trajectory

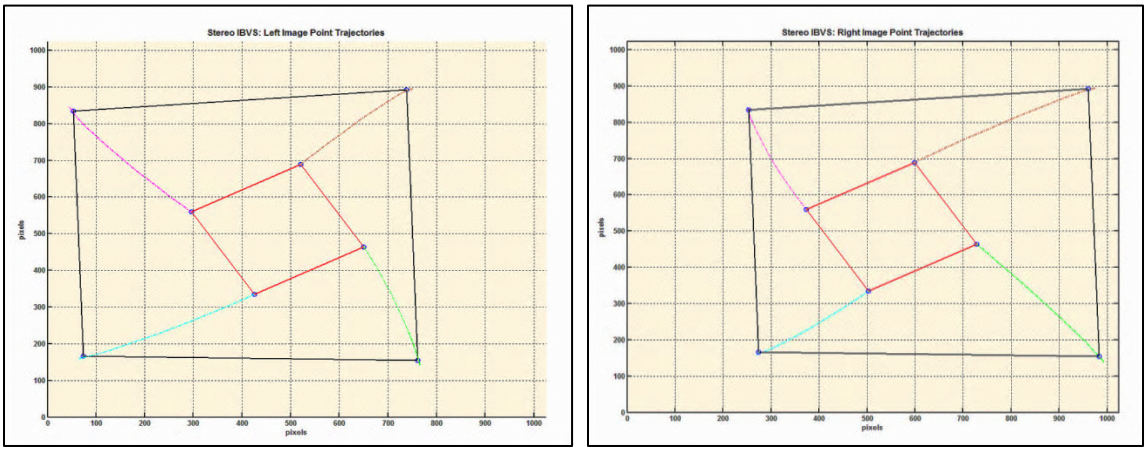
(c) (d)



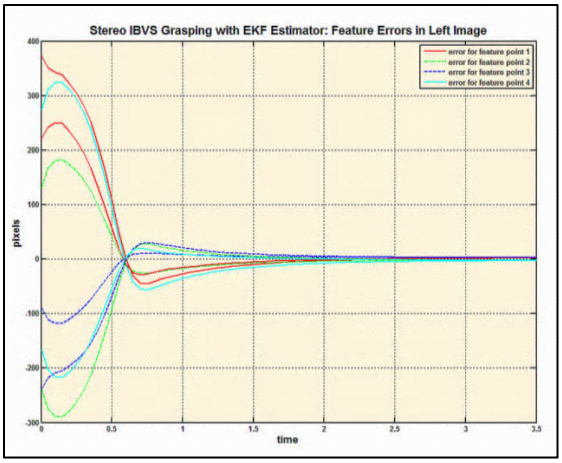
(e)

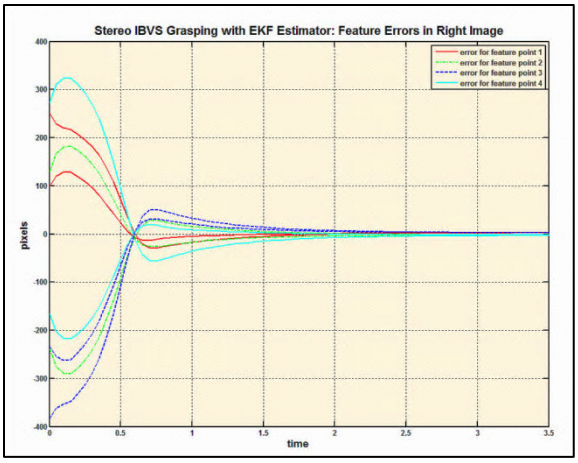
Figure 4.15 Stereo IBVS system with parallel cameras behavior in a procedure of grasping a moving object using **Kalman** Estimator: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

Figure 4.16 illustrates the results of using the same stereo visual servoing system with an EKF estimator for tracking and grasping a moving object.

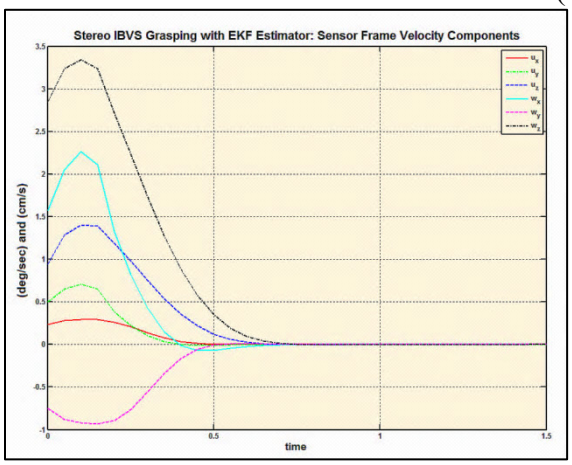


(a)

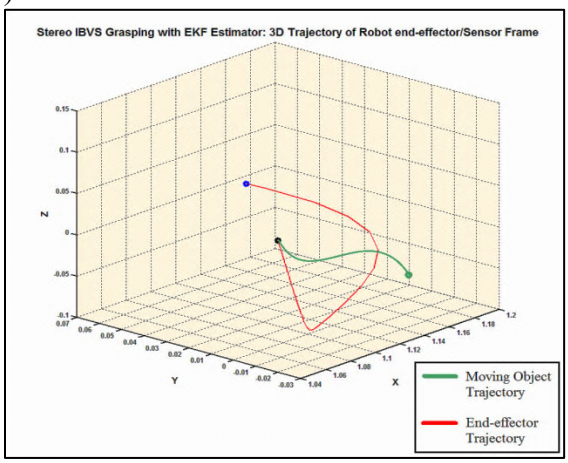




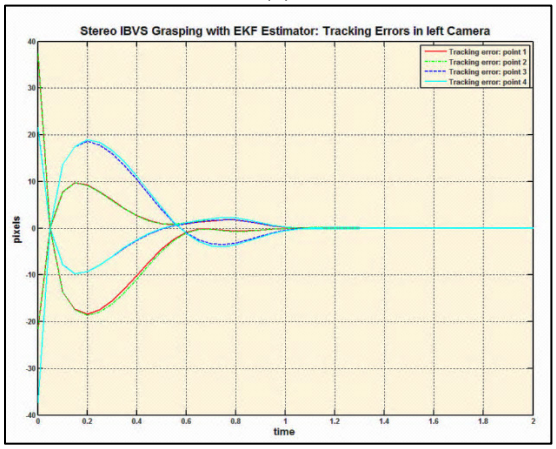
(b)

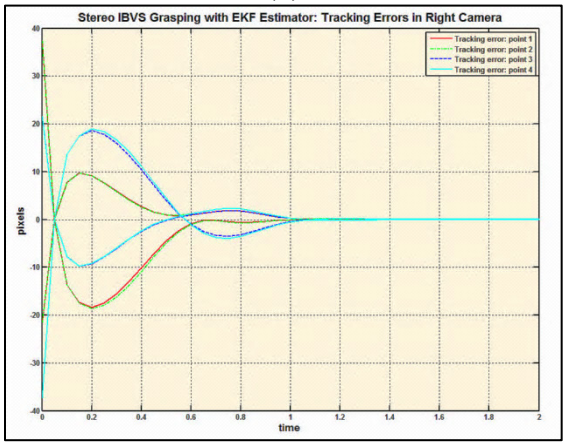


(c)



(d)

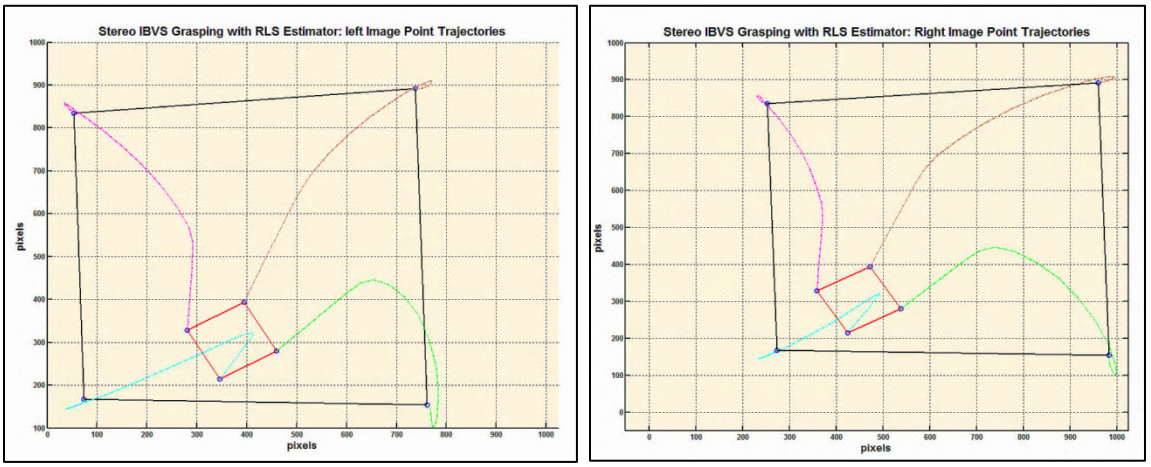




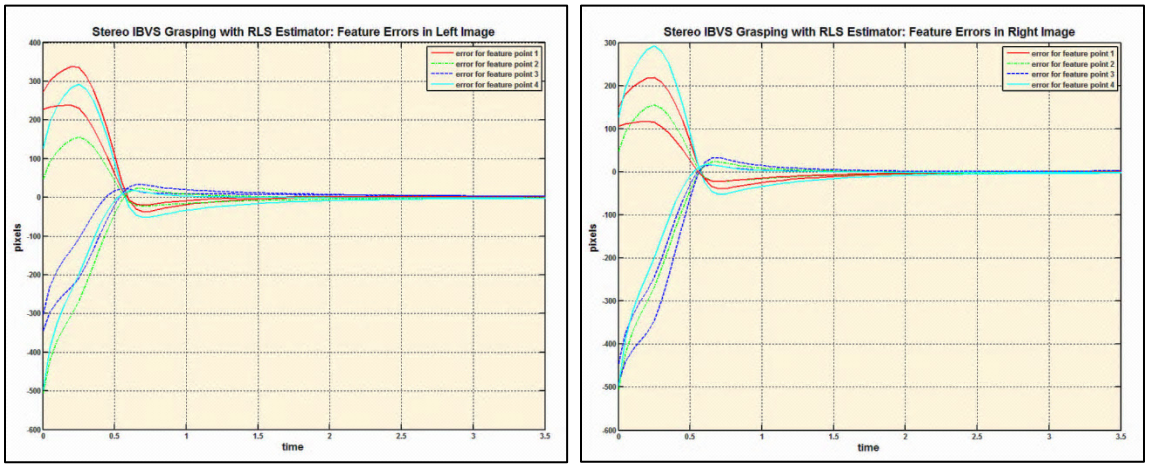
(e)

Figure 4.16 Stereo IBVS system with parallel cameras behavior in a procedure of grasping a moving object using **Extended Kalman** Estimator: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

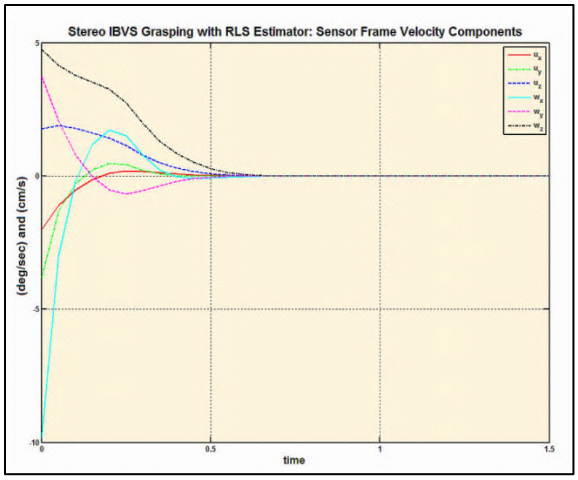
Finally, the results for using a Recursive Least Square (RLS) estimator for tracking and grasping a moving object are shown in Figure 4.17.

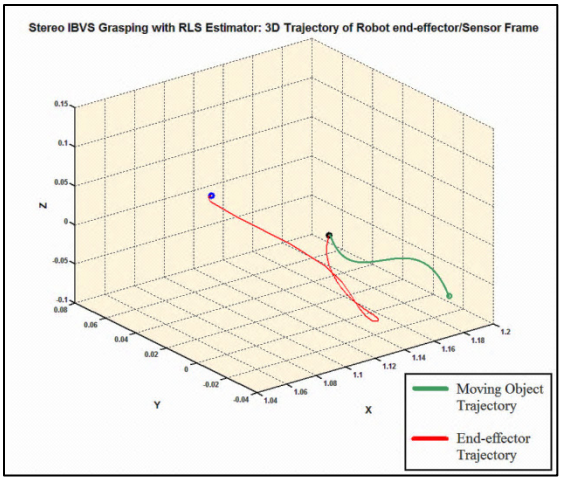


(a)

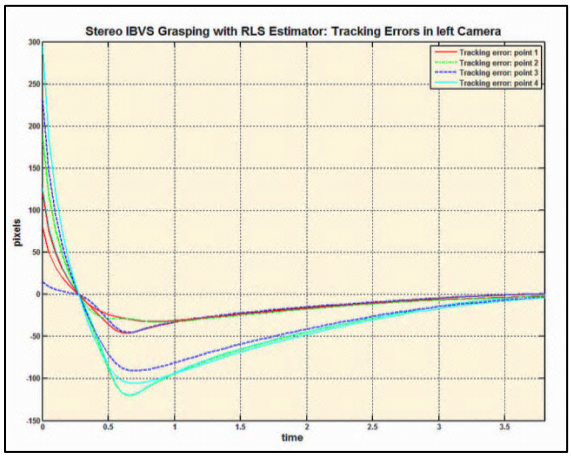


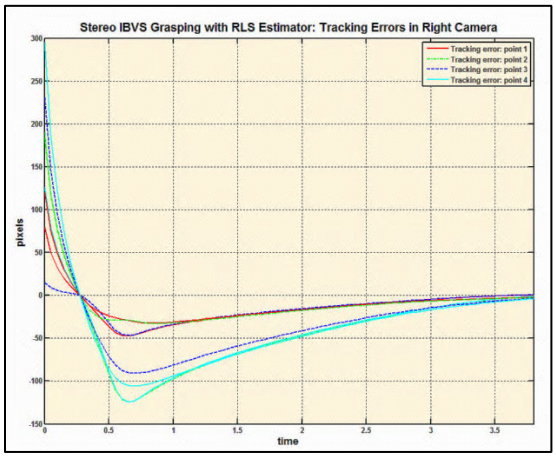
(b)





(c) (d)



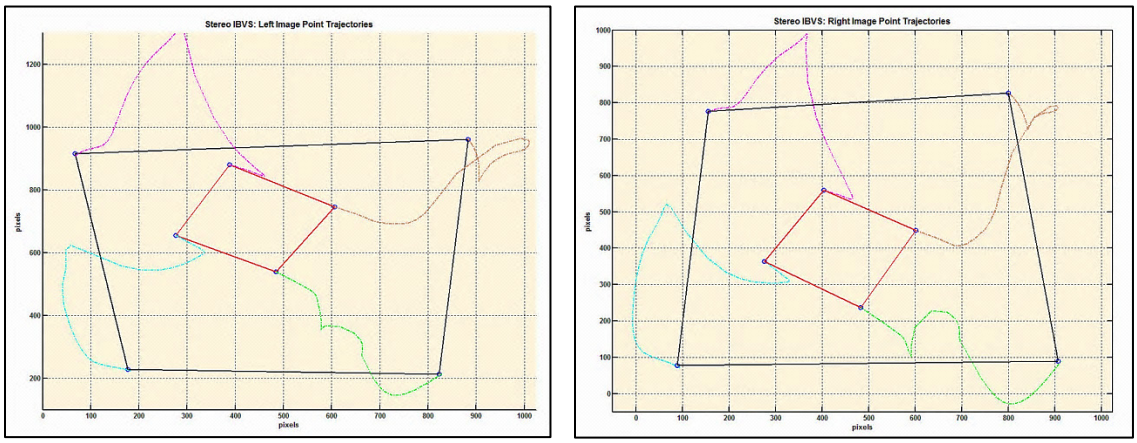


(e)

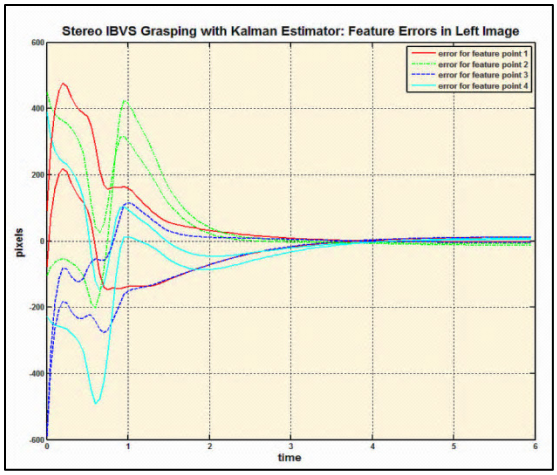
Figure 4.17 Stereo IBVS system with parallel cameras behavior in a procedure of grasping a moving object using **RLS** Estimator: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

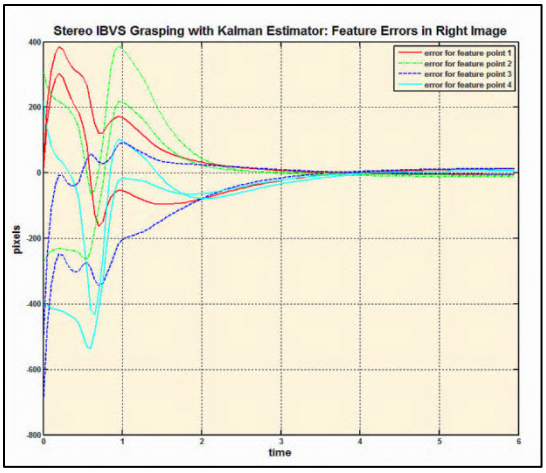
It is quite remarkable from Figures 4.15-4.17 that in comparison with the monocular system, the trajectories of the points in the images are smoother and in the case of using the EKF estimator is almost straight lines and the camera velocity components do not include large oscillations which leads to less energy consumptions. It can be also inferred from the simulation results that the Extended Kalman estimator performs the best in terms of tracking and convergence performance.

In addition, the end-effector 3-D position behavior in the system with EKF estimator is better than in other two methods. The camera velocity components in the system with EKF in comparison with the system with RLS estimator started with relatively lower values. The image feature trajectories of three systems are almost the same. In order to keep the object feature points in the cameras fields of views for both, tracking and approaching phases, the cameras are tilted about their X axis with 30 and -30 degrees. This allows the robot end-effector to get closer to the object for grasping task. The simulation results for stereo visual servoing system with tilted (non-parallel) cameras and a Kalman filter are given in Figure 4.18.

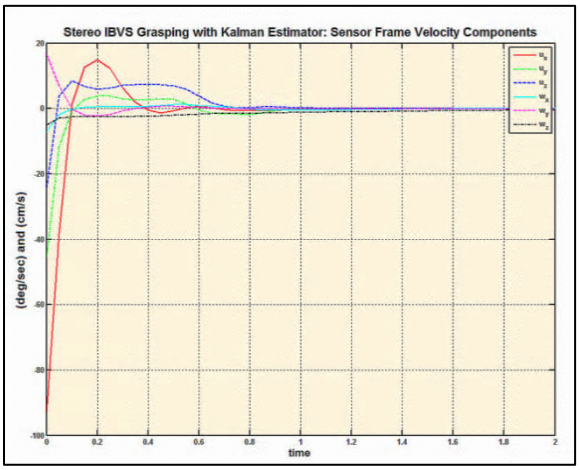


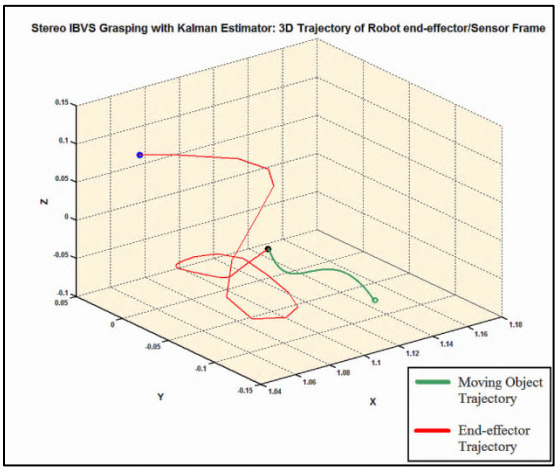
(a)



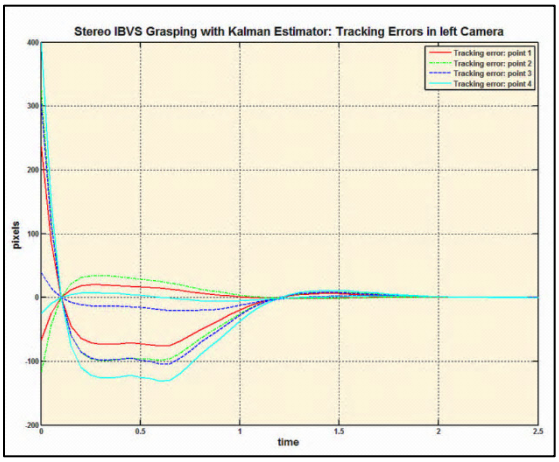


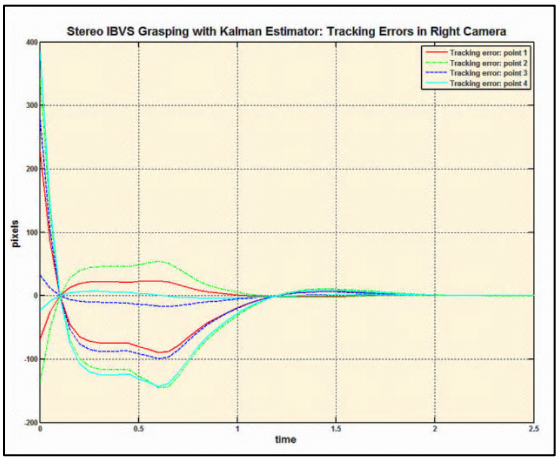
(b)





(c) (d)

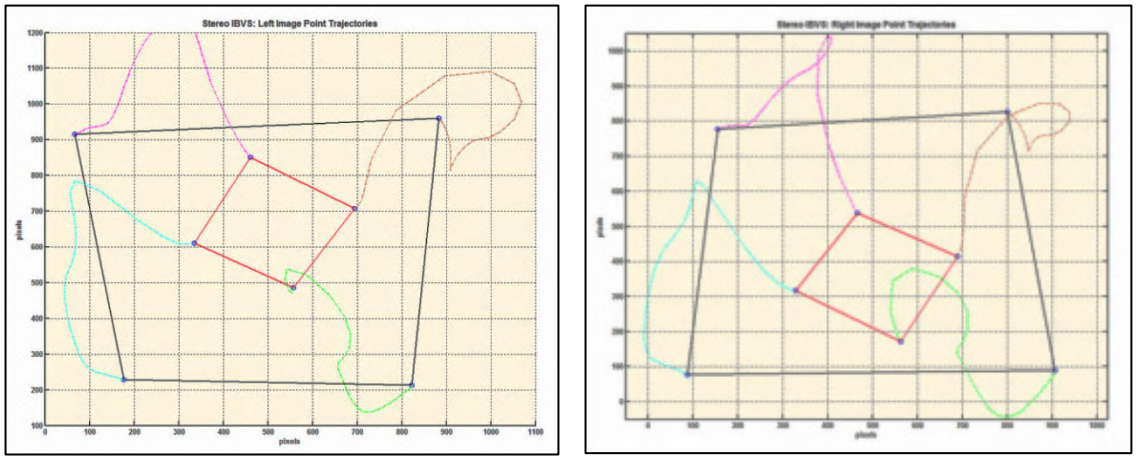




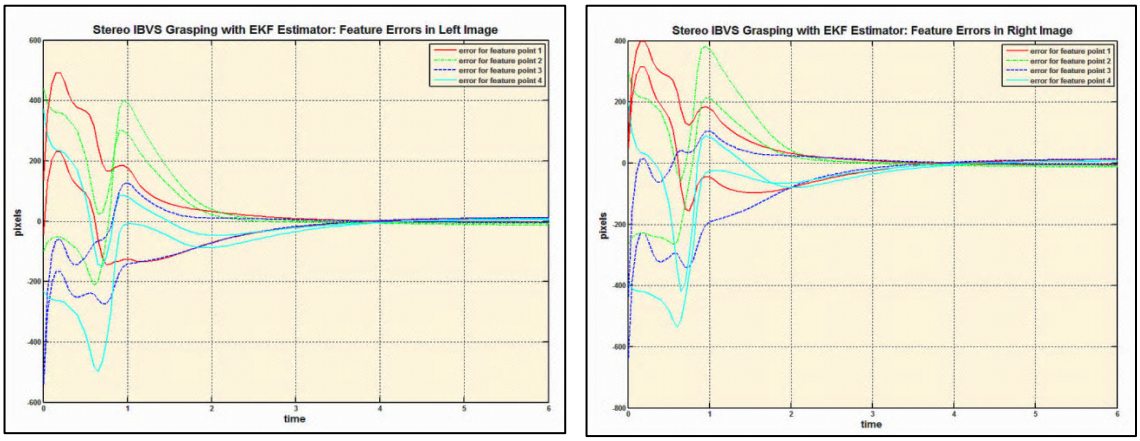
(e)

Figure 4.18 Stereo IBVS system with tilted (non-parallel) cameras behavior in a procedure of grasping a moving object using **Kalman** Estimator: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

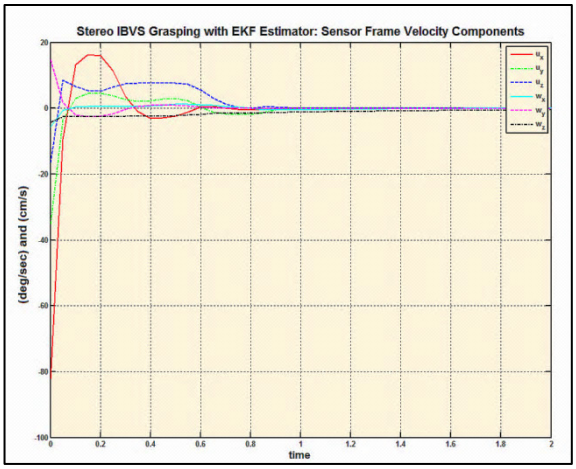
The simulation results of using the indicated stereo visual servoing system with tilted (non-parallel) cameras and an EKF estimator for tracking and grasping a moving object are illustrated in Figure 4.19.



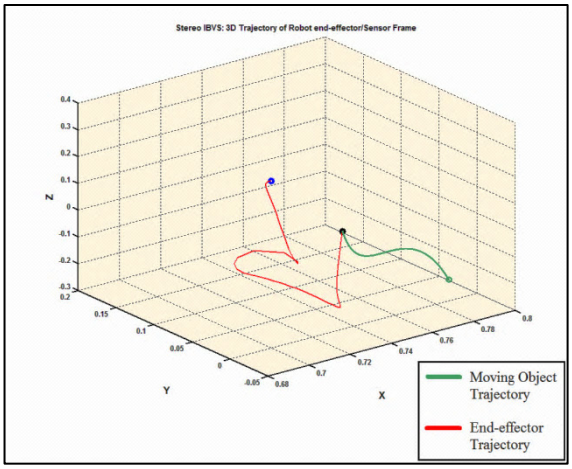
(a)



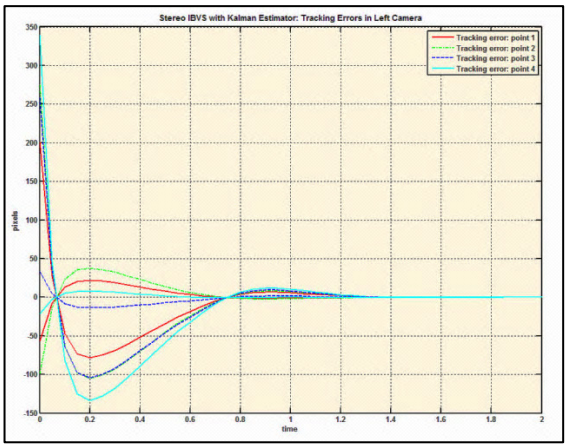
(b)

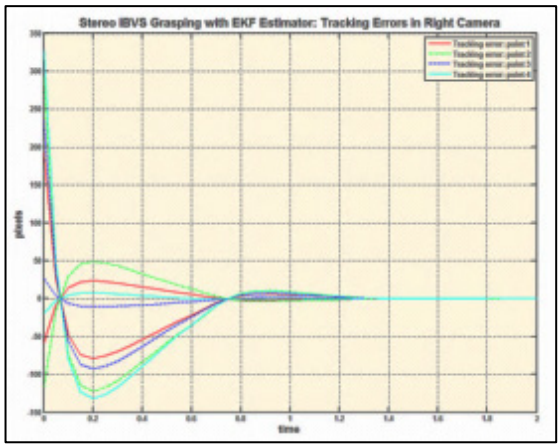


(c)



(d)

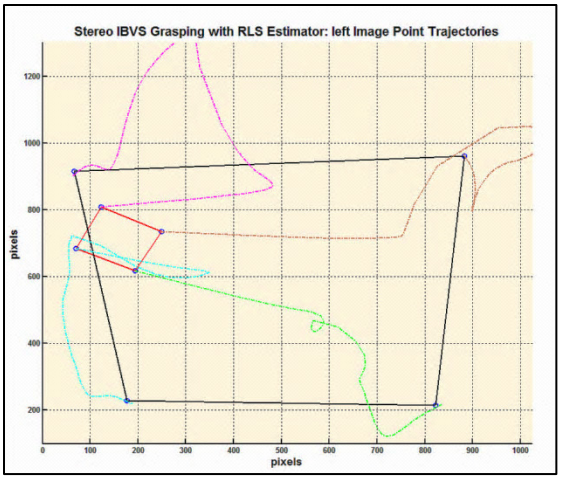


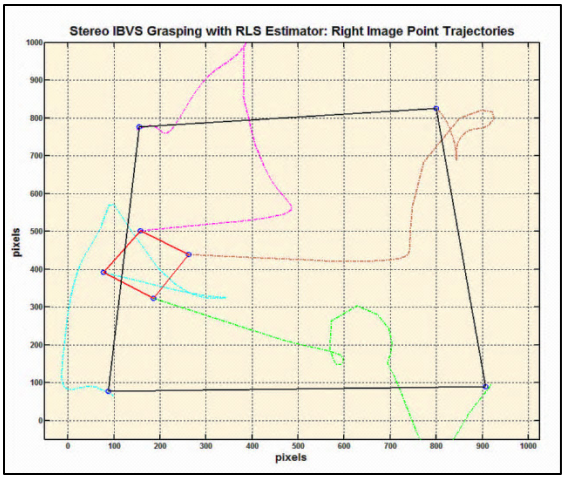


(e)

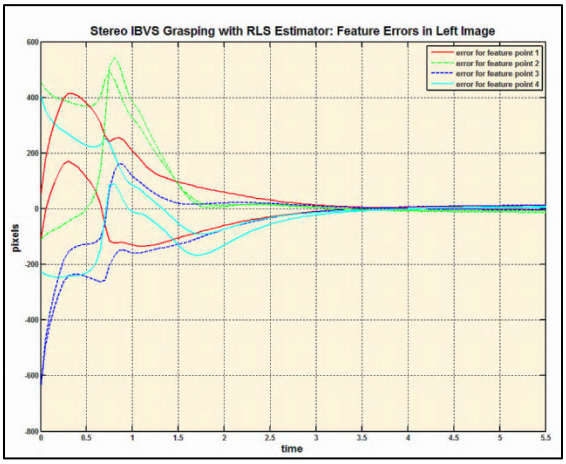
Figure 4.19 Stereo IBVS system with tilted (non-parallel) cameras behavior in a procedure of grasping a moving object using **Extended Kalman** Estimator: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

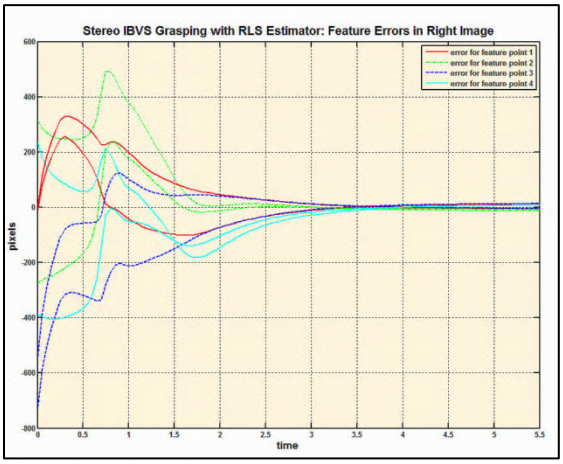
Figure 4.20 illustrates the results of using the same stereo visual servoing system with a RLS estimator for tracking and grasping a moving object.


Stereo IBVS Grasping with RLS Estimator: left Image Point Trajectories
pixels
pixels

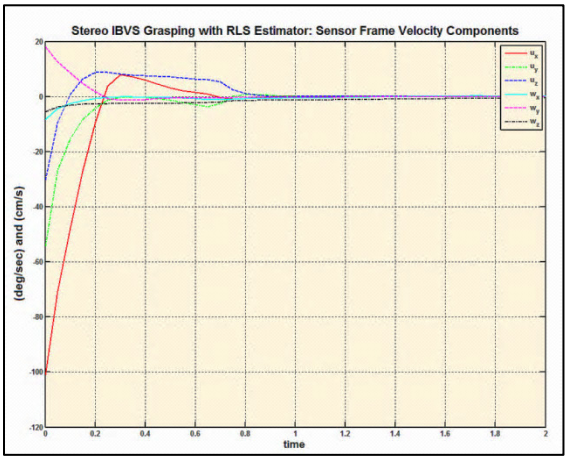

Stereo IBVS Grasping with RLS Estimator: Right Image Point Trajectories
pixels
pixels

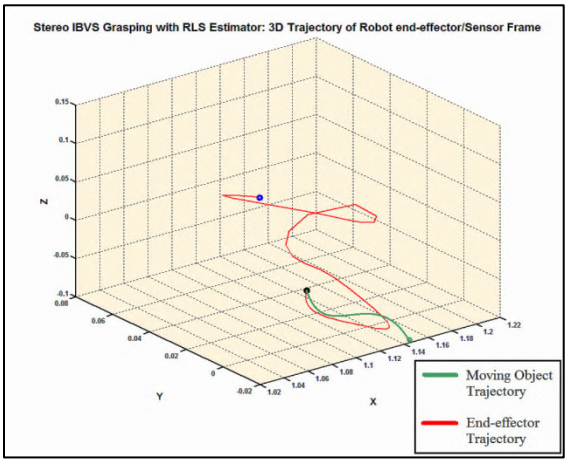
(a)


Stereo IBVS Grasping with RLS Estimator: Feature Errors in Left Image
error for feature point 1
error for feature point 2
error for feature point 3
error for feature point 4
pixels
time

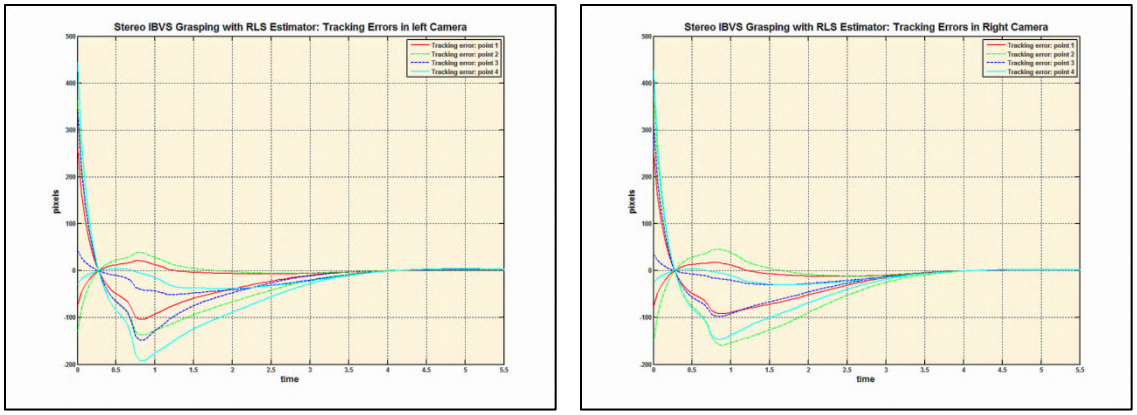

Stereo IBVS Grasping with RLS Estimator: Feature Errors in Right Image
error for feature point 1
error for feature point 2
error for feature point 3
error for feature point 4
pixels
time

(b)


Stereo IBVS Grasping with RLS Estimator: Sensor Frame Velocity Components
(deg/sec) and (cm/s)
time


Stereo IBVS Grasping with RLS Estimator: 3D Trajectory of Robot end-effector/Sensor Frame
Z
Y
X
Moving Object Trajectory
End-effector Trajectory

(c) (d)



(e)

Figure 4.20 Stereo IBVS system with tilted (non-parallel) cameras behavior in a procedure of grasping a moving object using a **RLS** Estimator: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

In the case of using non-parallel cameras, from the results of the simulations, we can draw the conclusion that EKF estimator owns a very good performance in terms of system convergence and tracking.

Comparing the simulation results of both monocular and stereo robot visual servoing systems for a tracking and grasping task, shows that the image feature trajectories in the stereo system are smoother and less complex and the system has a sooner convergence. The smaller sensor frame velocities at the beginning of the motion indicate that the stereo system performs more efficiently compared to the monocular system. Using the stereo vision for visual servoing results in better convergence characteristics because it makes possible to calculate the exact image interaction matrix at any position without knowing a

model of the target object. Table 4.1 shows a summarization of some of the comparison results for all the tracking and grasping cases.

Table 4.1 Comparison results for all the IBVS cases for tracking and grasping of a moving object

IBVS Case	Convergence Time (sec)	Sensor Frame Angular Vel. at t=0 s (deg/sec)	Sensor Frame Linear Vel. at t=0 s (cm/s)	Maximum Tracking Error (pixels)
Monocular+Kalman	4.3	18	160	612
Monocular+EKF	5	12	106	402
Monocular+RLS	5.1	20	193	651
Par. Stereo+Kalman	3.5	4	28	36
Par. Stereo+EKF	2.5	5	35	18
Par. Stereo+RLS	3.6	11	127	124
Non-Par. Stereo +Kalman	6.3	9	105	130
Non-Par. Stereo +EKF	5.8	9	90	121
Non-Par. Stereo+RLS	7	12	122	188

4.3. Summary

The model of the robotic stereo visual servoing system and also simulation results for developed methods and algorithms are presented in this chapter. Then using the computer simulations, the effectiveness of the proposed image-based stereo visual servoing system has been validated and compared to a monocular system in an "image feature positioning" task. Finally the system performance in a task of tracking and grasping a moving object is examined and the results for utilizing a Kalman filter, an EKF and a Recursive Least Square algorithm for predicting the position and trajectory of the maneuvering target have been presented and discussed.

Chapter 5 : Experimental Setup and Results

In this chapter the proposed visual servoing control system and the tracking algorithms are implemented and tested for various tasks in a real-world application. The simulation results on tracking and grasping a moving object are further verified in experiment. This chapter is also devoted to describing the experimental setup including the 6-DOF Robot, controller, vision system, stereo rig and camera calibration process. The image processing and feature extraction procedures are discussed and illustrated in this chapter.

5.1. Experimental framework

Figure 5.1 shows the block diagram for the experiment design. The total system shown in this Figure consists of a 6-DOF DENSO® 6242G robot, a Quanser® open-

architecture control module, two Logitech® C270 digital cameras with 1280 by 720 pixels resolution and two PCs for the robot control and image acquisition.

Figure 5.1 Block diagram for the experimental setup

The image processing, feature extraction and tracking algorithms require a considerable computation. And this will cause the whole implementation process to slow down and affect the system real-time performance. For this matter, the vision system is connected to another PC (PC 2) and the image processing and feature extraction algorithms are processed on this PC. The processing data is transmitted to the host PC connected to the robot using a UDP network connection protocol.

UDP or "User Datagram Protocol" is a simple transport layer protocol for client/server network applications based on Internet Protocol (IP) [66]. UDP is the main alternative to TCP and one of the oldest network protocols in existence, introduced in 1980. UDP is often used in video and image transferring applications or computer games specially tuned for "real-time" performance.

The IBVS algorithm running on the host computer uses the transferred image processing data to guide the robot and control the joint actuators. The robot controlling process will also be carried out on the host computer (PC 1). More information about the implementation system components is presented in the following sections.

5.2. DENSO Robot and Controller

In order to evaluate the proposed stereo visual servoing algorithm, a 6-DOF DENSO [67] robot has been used for real-world implementations. DENSO VP-6242G is a high precision manipulator robot with 6 rotating joints which are powered by AC servo motors. The position feedback is provided by absolute encoders mounted on each joints. The Denso robot end-effector has the ability to host various devices for different applications. The Denso robot, is supplied with Quanser® open-architecture control module which has all the capabilities of an industrial system and is interfaced to QUARC [68].

This module includes six independent amplifiers and built-in feed-forward with PID controllers. The controllers are operating about each motor at a rate of 4kHz. The DENSO 6-Axis Robot and the control module are shown in Figure 5.2.

Figure 5.2 The DENSO VP-6242G 6-Axis robot and the Quarc control module

A brief and general specification of the robot and controller unit is given in Appendix A. For the experiment purposes regarding the research work for current thesis and also for the research group, another 6-DOF manipulator robot—PUMA 260 was initially considered and an up-to-date a controller capable of interfacing with LabView® and Visual C++, has been retrofitted.

Appendix B briefly describes the design and implementation of the above-mentioned controller for 6-DOF PUMA-260® robot while Figure 5.3 shows the implemented Simulink model with Quarc® for the purpose of controlling the robot based on stereo visual servoing.

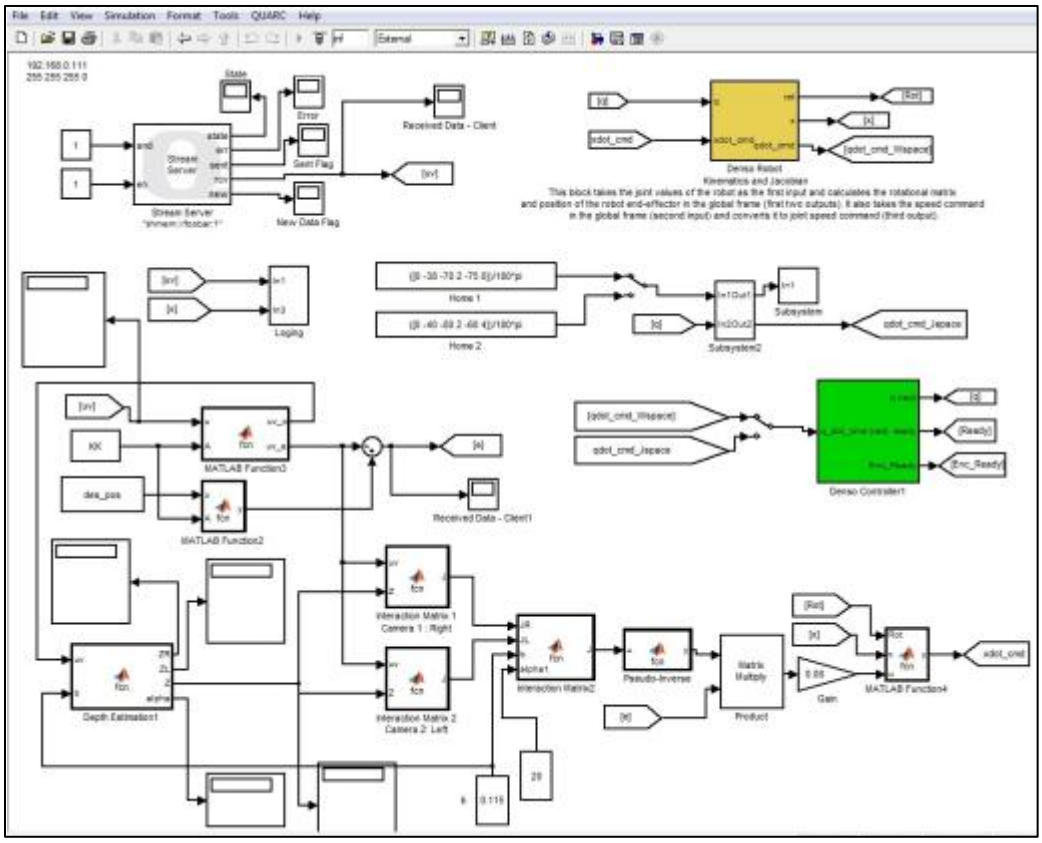


Figure 5.3 The Simulink model for robot stereo visual servoing based on Quarc

5.3. Stereo Vision System

The stereo vision system consists of two Logitech® C270 digital cameras with 1280 by 720 pixels resolution, mounted approximately 12 cm from each other in two different configurations: in the first case the cameras are located on the rig with an angle of about

30 degrees from the line that unites the centers of the cameras. And in the second case the cameras are perpendicular to the baseline. Figure 5.4 shows the Logitech® C270 digital camera and the corresponding imaging axis. Figure 5.5 also shows the DENSO robot with the mounted stereo vision system.

Figure 5.4 Digital camera used in the stereo vision system

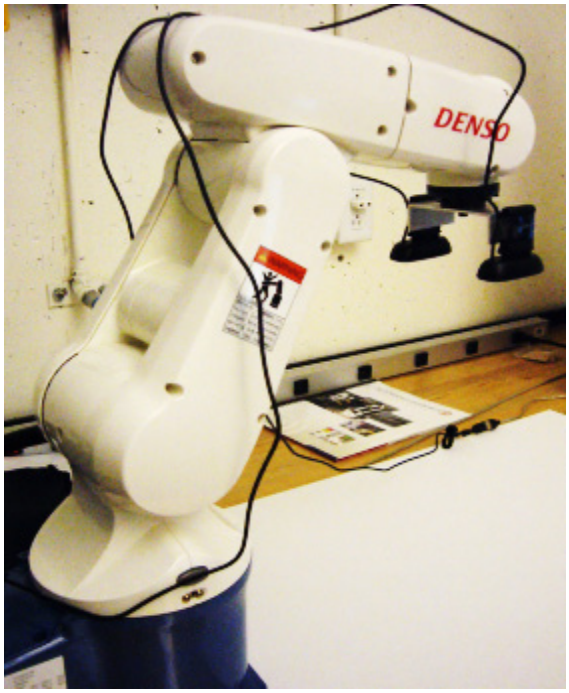


Figure 5.5 DENSO robot with the mounted stereo vision system

5.4. Camera Calibration

Camera calibration is the process of determining the camera's intrinsic parameters and the extrinsic parameters with respect to the world coordinate system. Calibrations techniques rely on sets of world points whose relative coordinates are known and whose corresponding image-plane coordinates are also known [69].

The Camera Calibration Toolbox for MATLAB implements the calibration method to find the camera's intrinsic and extrinsic parameters. The inputs of this toolbox are several images of a model chessboard plane containing the calibration points. Corners on the calibration plane are used as calibration points. Figure 5.6 illustrates this procedure.

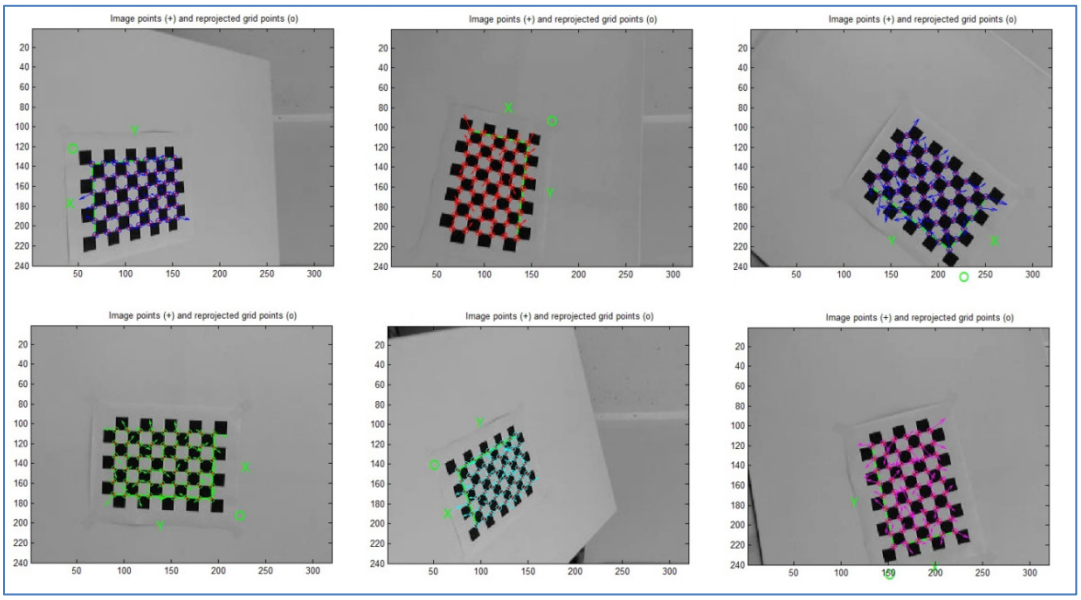


Figure 5.6 Camera calibration procedure using "camera calibration toolbox for MATLAB" with a model chessboard plane

The results for the intrinsic and extrinsic parameters are shown in Figures 5.7-5.8 and Table 5.1.

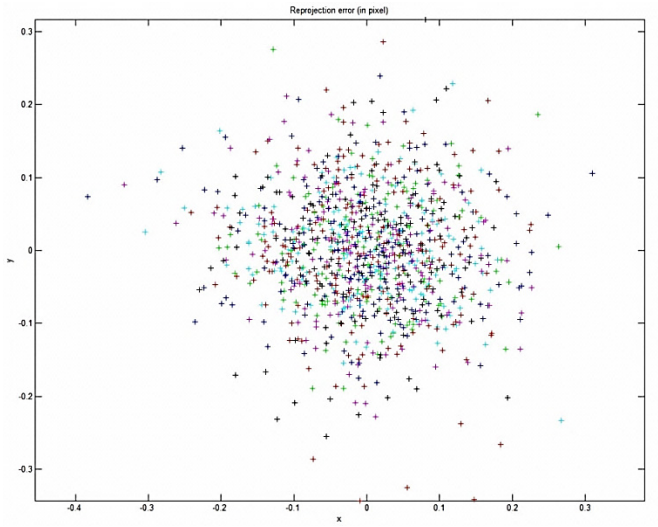


Figure 5.7 Stereo camera calibration - Projection error

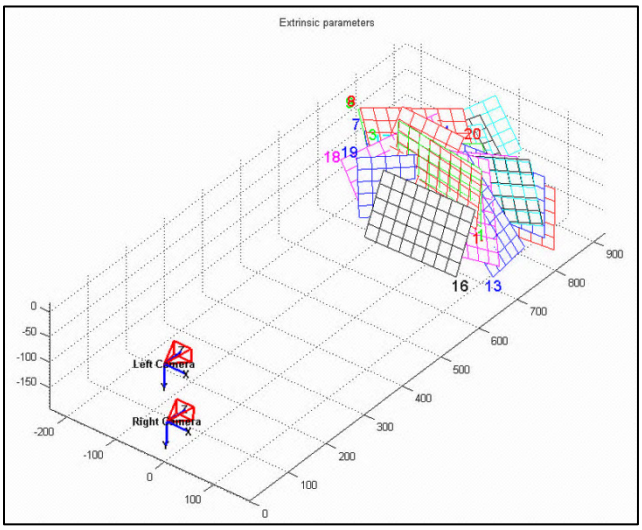


Figure 5.8 Camera calibration results: camera extrinsic parameters

Table 5.1 Camera calibration results: camera intrinsic parameters

Parameter	*Value*
Focal Length	fc = [408.93022 408.45085] ± [5.20372 5.21677]
Principal point	cc = [162.62803 118.94100] ± [5.17699 4.39332]
Skew	alpha_c = [0.00000]: angle of pixel = 90.00000 degrees
Distortion	kc = [-0.01386 0.17520 -0.00125 -0.00132 0.00000] ± [0.03060 0.19913 0.00365 0.00377 0.00000]
Pixel error	err = [0.09454 0.08218]

5.5. Image Processing and Feature Extraction

In order to come up with a reliable image processing and feature extraction algorithm, the first thing to decide is which color space to found the detection algorithms on. We implemented our methods using two different color spaces: RGB and HSV.

5.5.1. RGB-based Feature Extraction

The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors (Figure 5.9). The name of the model comes from the initials of the three additive primary colors, red, green, and blue [43]. The main purpose of using the RGB color model is for the sensing,

representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. RGB is a convenient color model for computer graphics because the human visual system works in a way that is similar to an RGB color space.

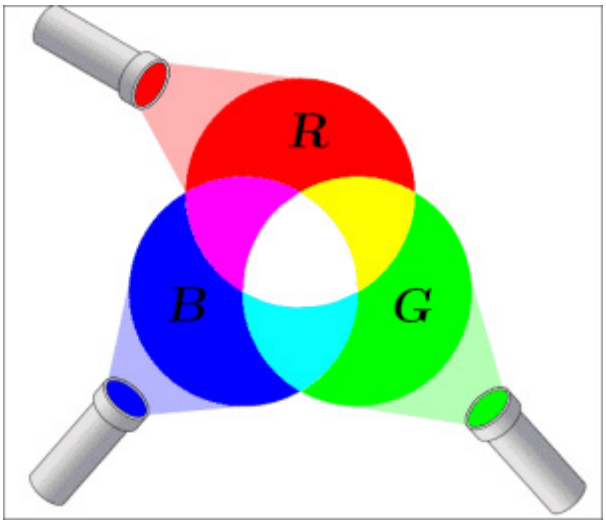


Figure 5.9 RGB color space [61]

Image object detections based on color is one of the quickest and easiest methods for tracking an object from one image frame to the next. The speed of this technique makes it very popular for real-time applications.

Since changing the brightness and light in workspace causes a lot of differences in the captured colors, we considered tolerances for Red, Green and Blue values to detect the objects based on RGB color space. Predefined RGB values for each feature color are saved then for each frame captured by the vision system all of the pixels in the image are examined and the pixels detected to be in the range of the saved RGB values with consid-

ered tolerances are marked. According to Equations (5.1-5.2) and having the total number of detected pixels (N) and also the position of each in the Cartesian frame of the image, it is possible to calculate the centroid of the detected object $(x_{centroid}, y_{centroid})$:

$$x_{centroid} = \frac{1}{N}\sum_{n=0}^{N} x_n \tag{5.1}$$

$$y_{centroid} = \frac{1}{N}\sum_{n=0}^{N} y_n \tag{5.2}$$

5.5.2. HSV-based Feature Extraction

HSV is one of the most common cylindrical-coordinate representations of points in an RGB color model, which rearrange the geometry of RGB in an attempt to be more perceptually relevant than the Cartesian representation [69] . They were developed in the 1970s for computer graphics applications, and are used for color pickers, in color-modification tools in image editing software, and less commonly for image analysis and computer vision. HSV stands for hue, saturation, and value, and is also often called HSB (B for brightness). In each cylinder, the angle around the central vertical axis corresponds to "hue", the distance from the axis corresponds to "saturation", and the distance along the axis corresponds to "lightness", "value" or "brightness". Figure 5.10 shows an illustration of HSV color space.

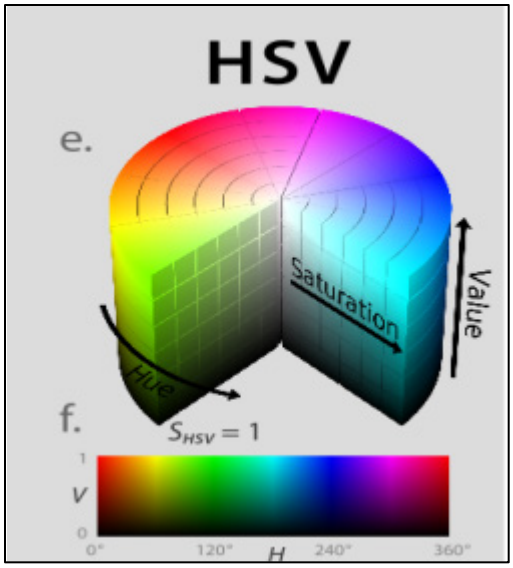


Figure 5.10 HSV color space [68]

The most important flaw about RGB-Based detection methods is that changing the light and brightness of the framework and work-space affects detection so badly that sometimes the program cannot identify the color objects at all. Knowing the fact that the brightness is an independence value in HSV (or HSB) color space, we can come up with a proper tolerance to be considered in program which covers all the values of a predefined color in the workspace. This will certainly make the implementation of the color detection more stable. Figure 5.11 shows the implemented image processing and feature extraction Simulink model for the purpose of visual servoing.

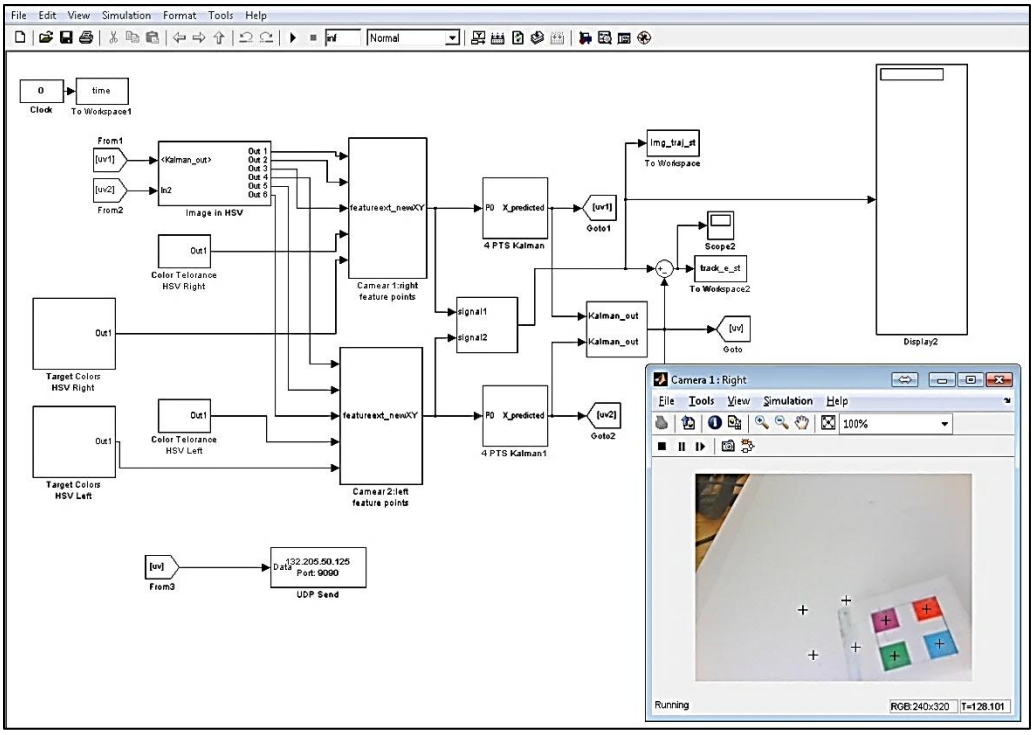


Figure 5.11 The Simulink model for the image processing and feature extraction

5.6. Experimental Results

This section presents various sets of experiments to evaluate the system performance in two cases of using binocular and monocular vision for the proposed Image-based stereo visual servoing system. Firstly, all the results for a simple image-based task with fixed feature points on the object are presented. Afterwards, the experimental results for the task of tracking grasping a moving object with random trajectories by a 6-DOF robot are presented and discussed. Figure 5.12 shows the DENSO robot and the object with feature points in a visual servoing task. The experiment implementations are designed and performed using Mathworks MATLAB® & Simulink® and based on the Quanser QUARC® integrations.

Figure 5.12 Denso Robot and the object in an IBVS task

QUARC® is a multi-functional software suite that easily connects with Mathworks Simulink for rapid controls prototyping and hardware based experiments [43]. QUARC provides Windows-based procedures to make Simulink-designed controllers to be converted into real-time "Microsoft Visual Studio-based" code that can run on many target processor and operating systems combinations. Figure 5.13 shows implemented Simulink models for the experimental IBVS system. Image measurements are noisy, since the experiments are carried out in a standard office environment, without any special illumination.

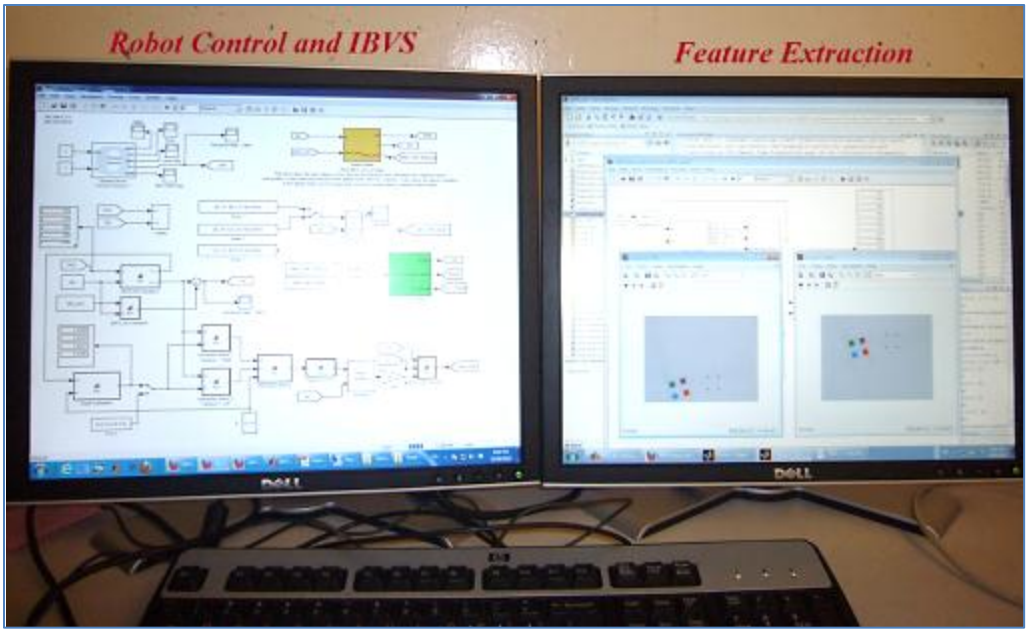


Figure 5.13 implemented Simulink models for the IBVS control system and feature extraction

5.6.1. Image-based Visual Servoing: Stereo VS. Monocular

In this section, an image-based visual servoing system in two cases of binocular and monocular vision is implemented and the effectiveness of the proposed eye-in-hand stereo visual servoing system is examined and the experimental results are discussed. Figures 5.14-5.17 are the presented results obtained from monocular image-based system in a simple servoing task.

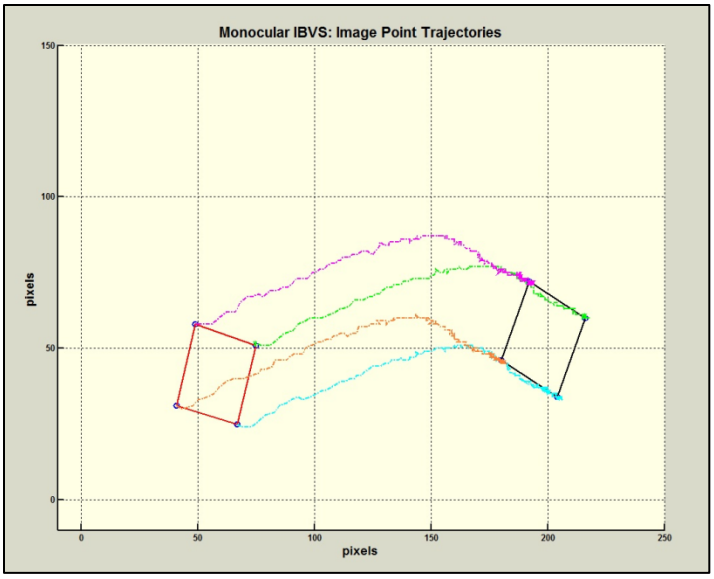


Figure 5.14 Image point trajectories in a monocular IBVS task

Figure 5.14 shows that the system finally converges to the desired feature positions but it is noticeable that the feature point trajectories are not quite smooth and the displacements are relatively large.

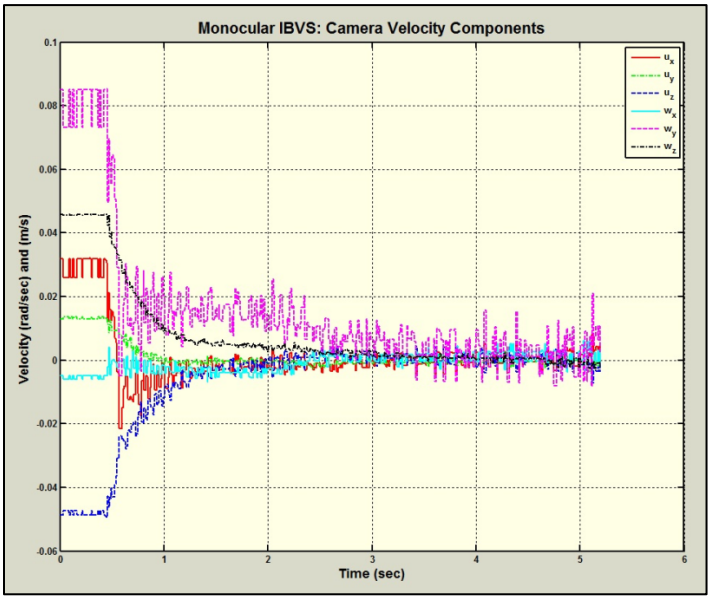


Figure 5.15 Camera velocity components in a monocular IBVS task

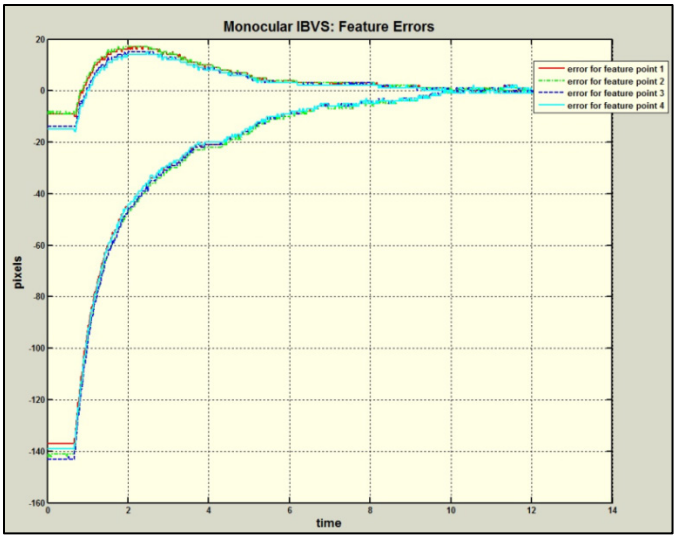


Figure 5.16 Image feature errors in a monocular IBVS task

From the results shown in Figure 5.15 it can be inferred that the behavior in computed camera velocity components does not present desirable and stable properties for the current monocular IBVS system. As assumed in the simulations the feature point depths in calculating image interaction matrices are assumed to be constant and equal to 1m.

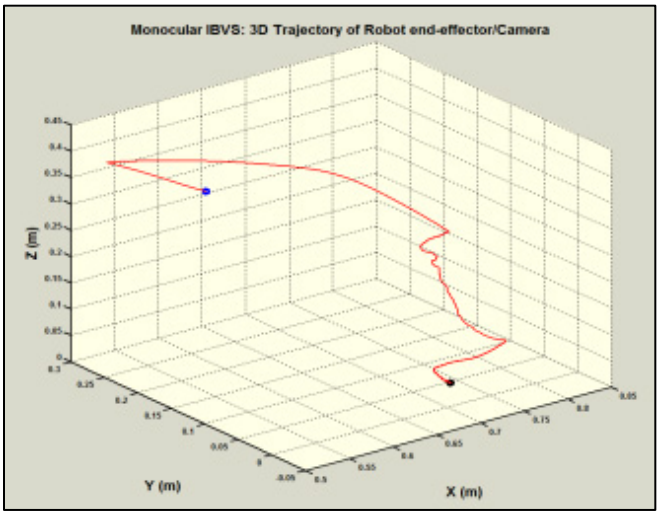


Figure 5.17 3-D Trajectory of the end-effector and camera optical center in a monocular IBVS task

The stereo IBVS system was set for the same servoing task with the same assumptions to achieve two desired sets of feature points for right and left images. The proportional gain for both cases is chosen to be $\lambda = -0.4$. Figure 5.18 shows the results for Image feature trajectories for left and right image planes in a stereo image-based visual servoing task. The provided trajectories for the image features in the stereo system seem to be quite smoother than the trajectories from the monocular IBVS.

Furthermore, according to Figure 5.19 the sensor frame velocity components in stereo visual servoing system do not include large oscillations compared to the camera velocity components in monocular system.

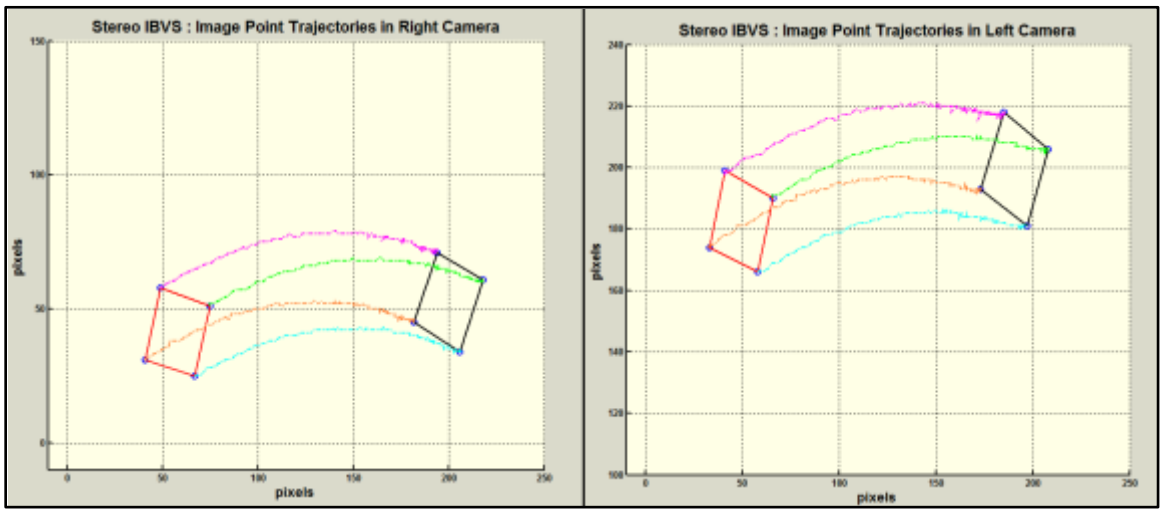


Figure 5.18 Image feature trajectories for left and right image plane in a Stereo IBVS task

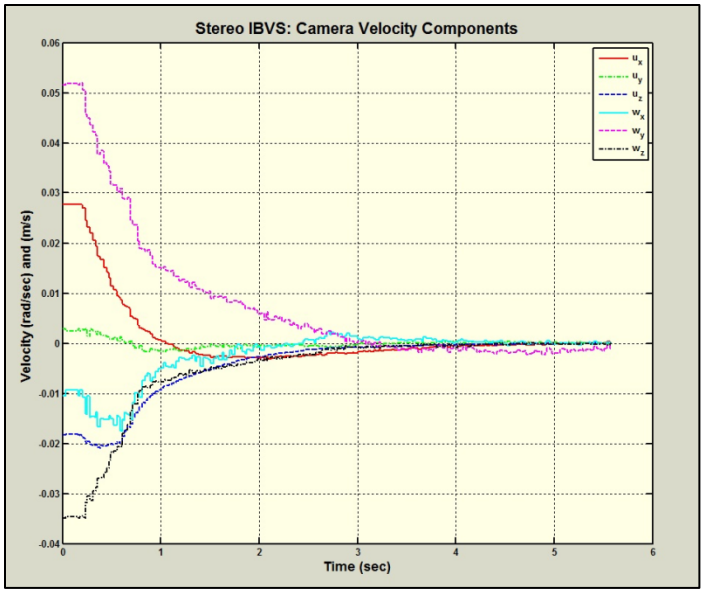


Figure 5.19 Sensor frame velocity components in a Stereo IBVS task

Figure 5.20 illustrates feature errors in left and right images which are approximately identical and comparing to the monocular case, these errors traverse a smoother trajectory and less oscillations are appeared. 3D end-effector trajectory during the task of image-based visual servoing for the stereo vision system is also shown in Figure 5.21.

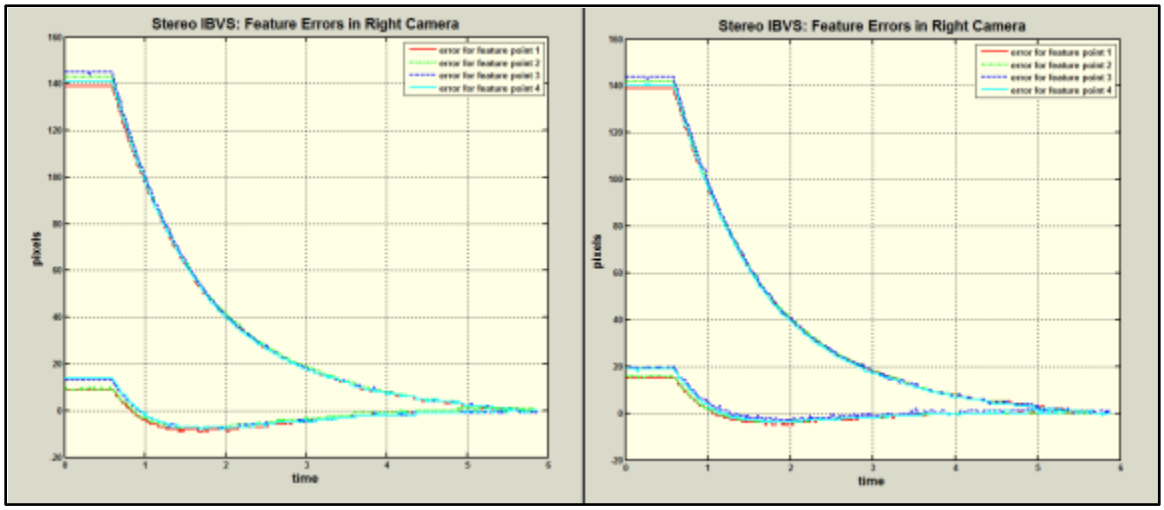


Figure 5.20 Feature errors in left and right images in a Stereo IBVS task

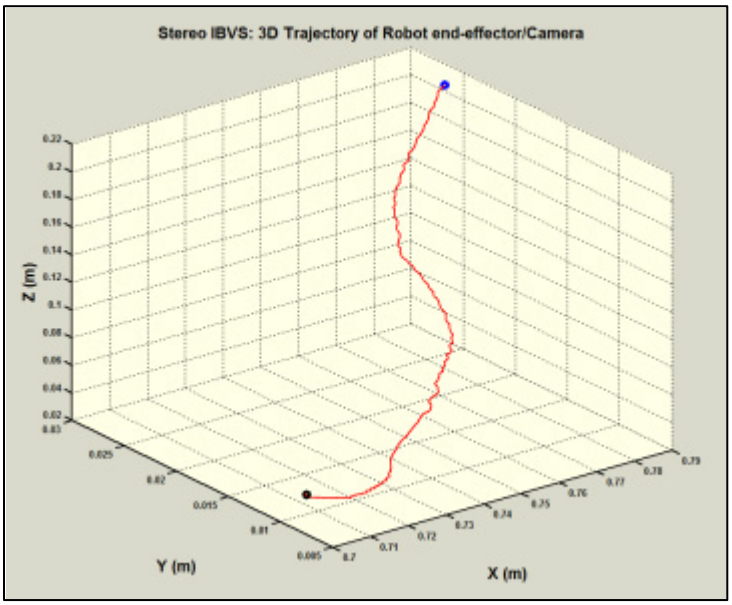


Figure 5.21 3-D Trajectory of the end-effector and sensor frame in a Stereo IBVS task

The difference in behavior, convergence and stability performance in the cases of stereo and monocular vision systems is due to the on-line procedure of depth calculation and updating the exact values of interaction matrices. Consequently, it is possible to generate the correct feedback command which leads to a more stable visual servoing system.

5.6.2. Tracking and Grasping a Moving Object through Visual Servoing: Stereo VS. Monocular

This section is devoted to presenting and discussing the results obtained from the experiments with the 6-DOF DENSO robot and comparing the performance of the proposed eye-in-hand image-based stereo visual servoing system to a conventional monocular system for the task of tracking and grasping a moving object. As performed in computer simulations, the predictions are carried out on three cases of Kalman filter, Extended Kalman Filter (EKF) and a Recursive Least Square method (RLS).

Since generating a motion with sinusoidal trajectory for the object requires some certain instruments, the object is manually and randomly moved within the cameras field of view with a relatively small speed. As it was mentioned previously, the tracking and grasping task is performed by pre-defining desired positions for the object image features such that the robot moves and aligns the end-effector with the object and reaches towards it. Figures 5.22-5.23 show the cameras view of the desired image features in the instant of grasping in both cases of monocular and stereo vision system.

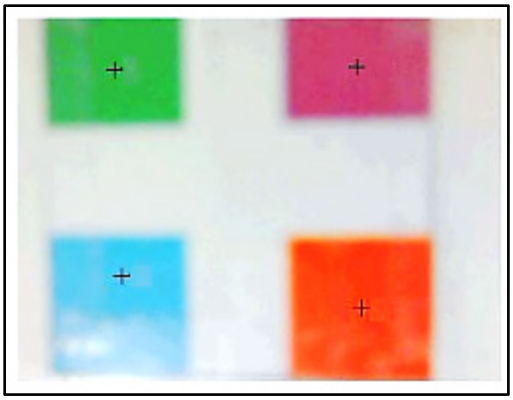


Figure 5.22 Desired image feature points for grasping task in the Monocular case

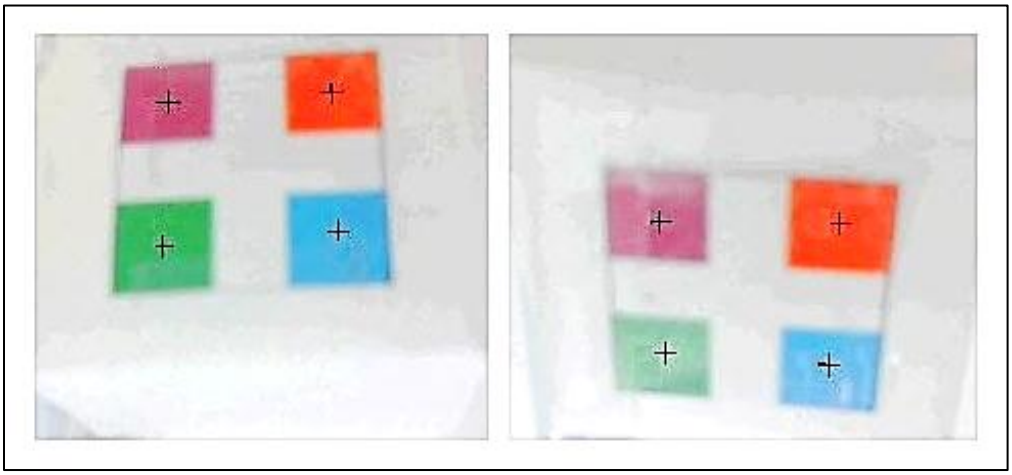
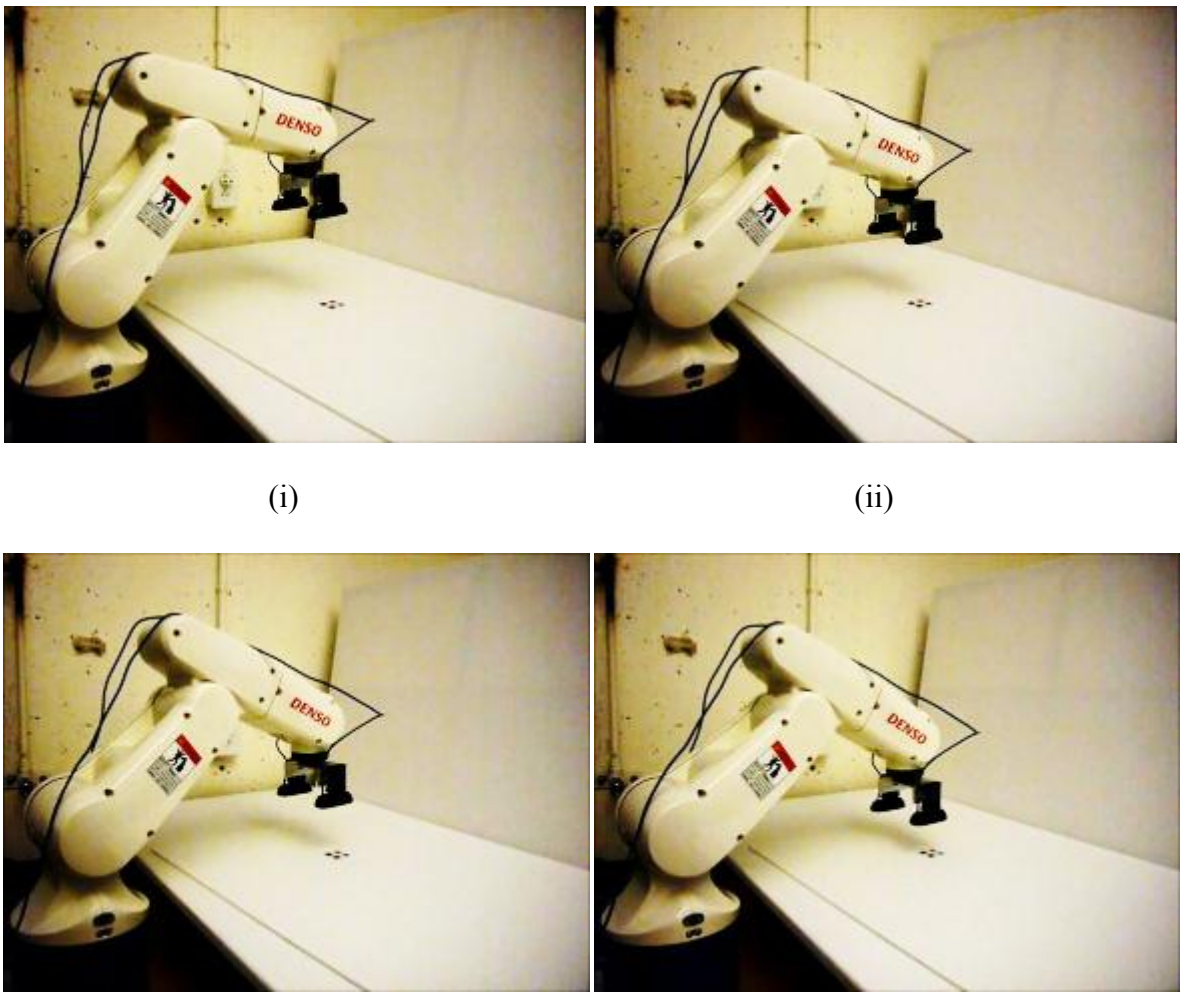


Figure 5.23 Desired image feature points for grasping task in the Stereo case

The results of using a monocular visual servoing system with various estimators to track and grasp a moving object are shown in Figures 5.25-5.27. Figure 5.24 illustrates the procedure of tracking and grasping a moving object through monocular image based visual servoing using 6-DOF DENSO robot.



(i) (ii)

(iii) (iv)

Figure 5.24 The Procedure of tracking and grasping a moving object through monocular image based visual servoing using 6-DOF DENSO robot

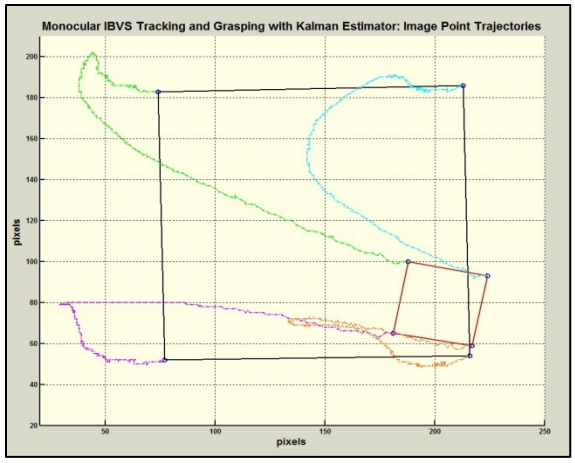
In the case of using Kalman estimators and according to Equation (3.13) and considering $a = 0.1\ (m / \sec^2)$ and $\Delta T = 1\ (\sec)$ the following assumptions for the process noise, covariance matrix of the measurement noise and the initial condition for estimated error covariance are taken into account as:

$$Q = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times 10^{-2} \tag{5.3}$$

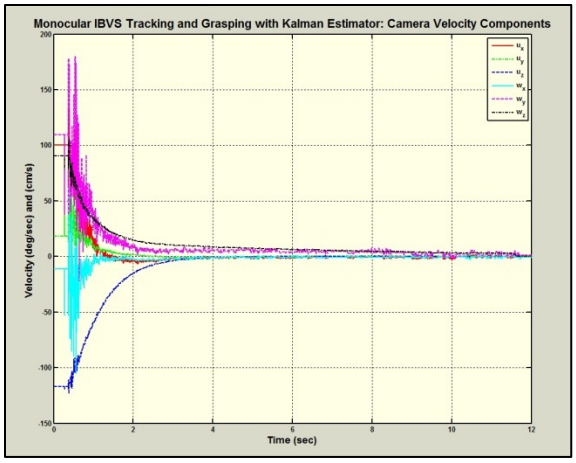
$$R = \begin{bmatrix} 25 & 0 \\ 0 & 25 \end{bmatrix} \tag{5.4}$$

$$P = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix} \tag{5.5}$$

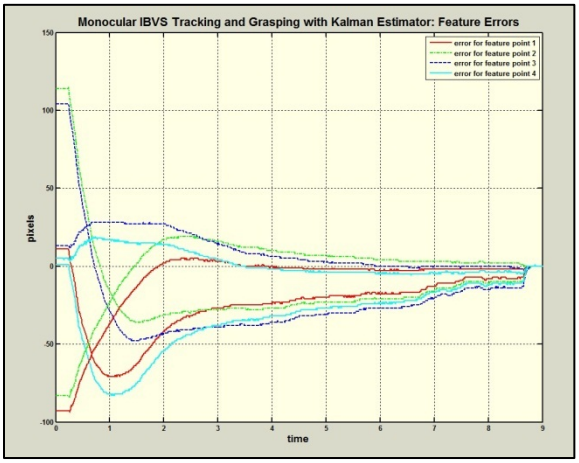
The results of using the monocular visual servoing system with a Kalman estimator for grasping a moving object and the image feature prediction errors are shown in Figure 5.25. As mentioned in previous chapter, this feature prediction error is the difference between actual positions of projected points in the image plane and predicted ones.



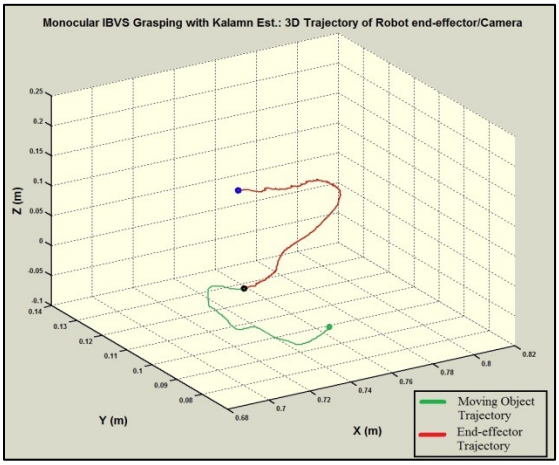
(a)



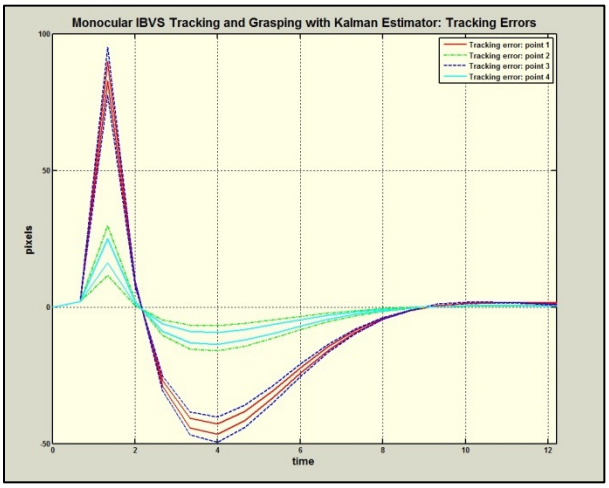
(b)



(c)



(d)

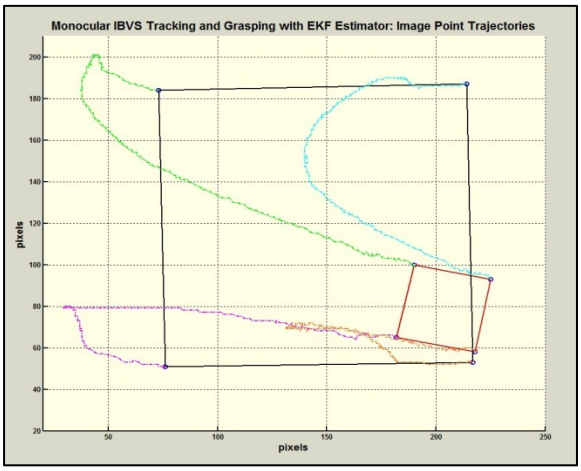


(e)

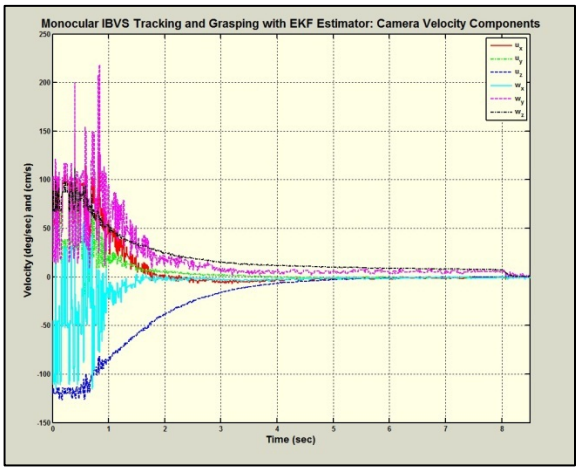
Figure 5.25 Experimental results for the Monocular IBVS system in a procedure of grasping a moving object using **Kalman** Estimator: (a) Image feature trajectories (b) Camera frame velocity components (c) Image feature errors (d) Robot end-effector 3D positions (e) Tracking Errors

Figure 5.26 illustrates the experimental results of using a monocular visual servoing system with an Extended Kalman Filter (EKF) estimator for grasping the same moving object. The state transition noise covariance matrix, Q, and the measurement noise covariance, R, are as follows:

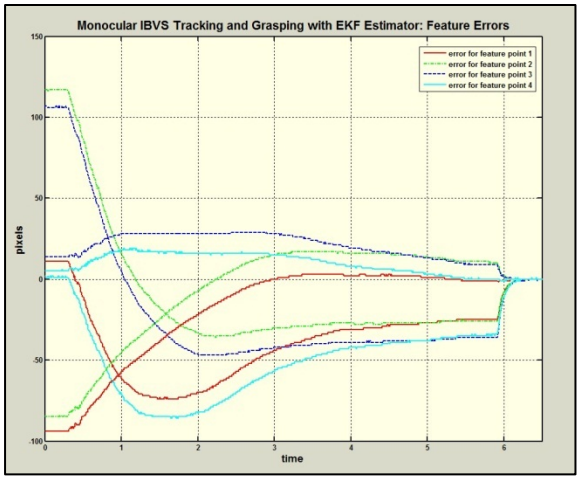
$$Q = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 10\dfrac{\pi}{180} \end{pmatrix}, \quad R = I_{2\times 2} \tag{5.6}$$



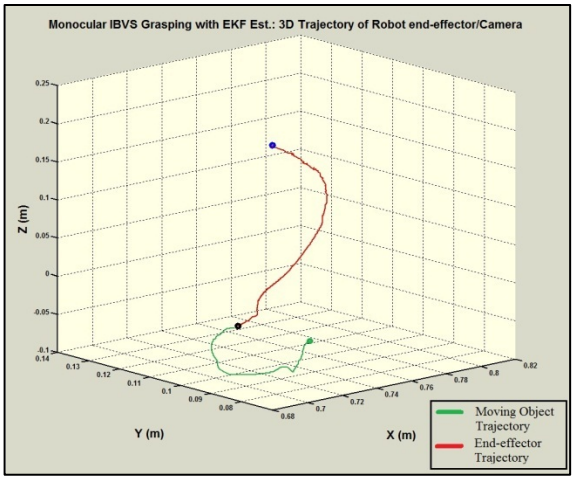
(a)



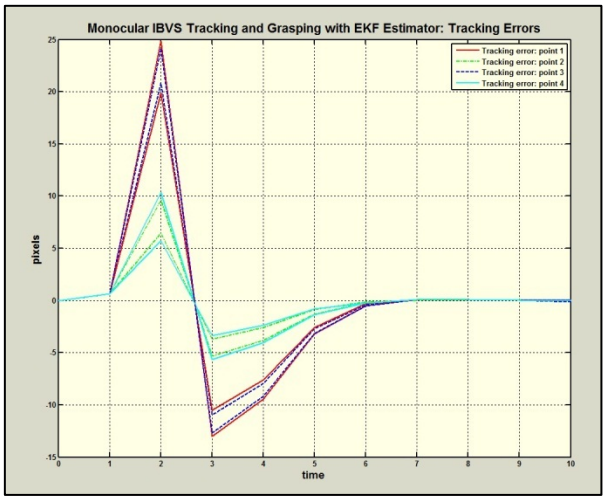
(b)



(c)



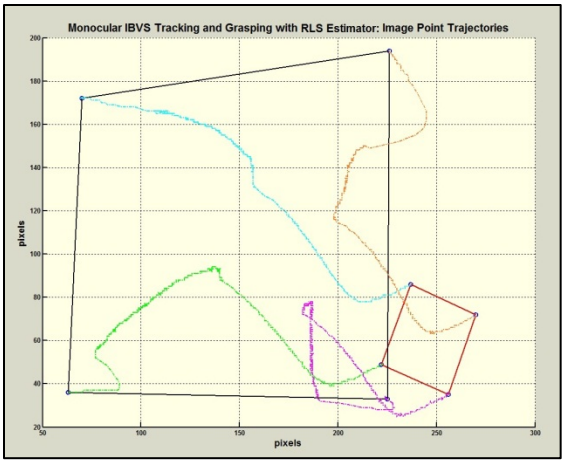
(d)



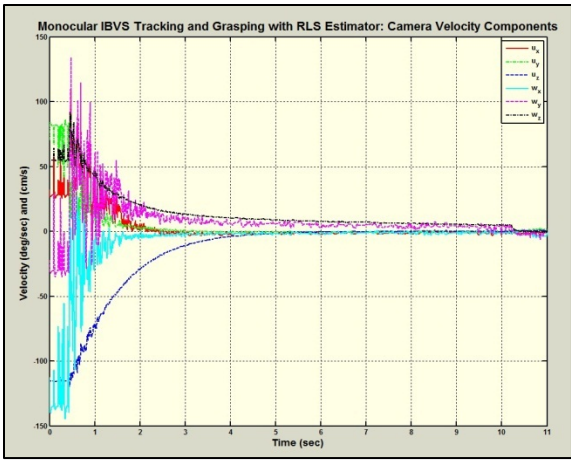
(e)

Figure 5.26 Experimental results for the Monocular IBVS system in a procedure of grasping a moving object using **Extended Kalman Filter** (EKF) Estimator: (a) Image feature trajectories (b) Camera frame velocity components (c) Image feature errors (d) Robot end-effector 3D positions (e) Tracking Errors

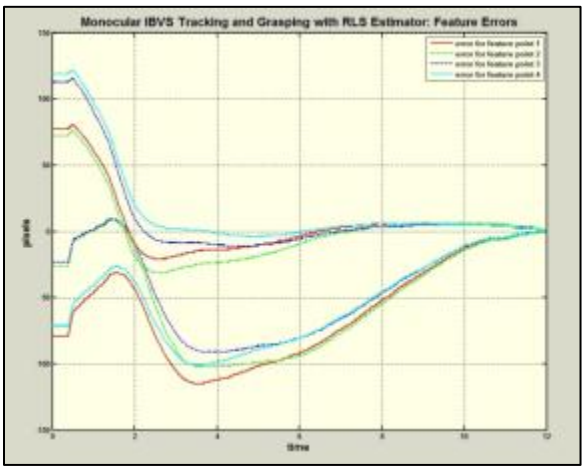
Figure 5.27 presents the experimental results of using a monocular visual servoing system with a Recursive Least Square (RLS) estimator for grasping a moving object.



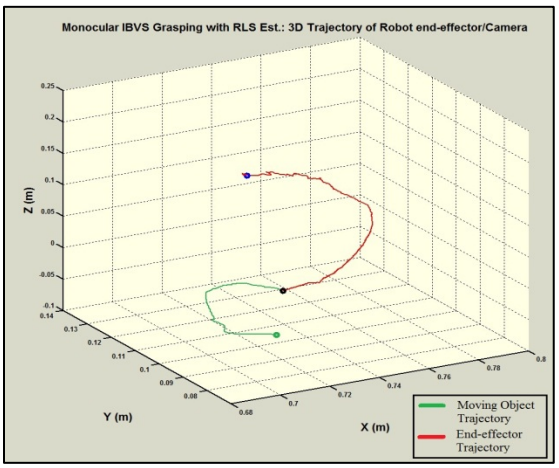
(a)



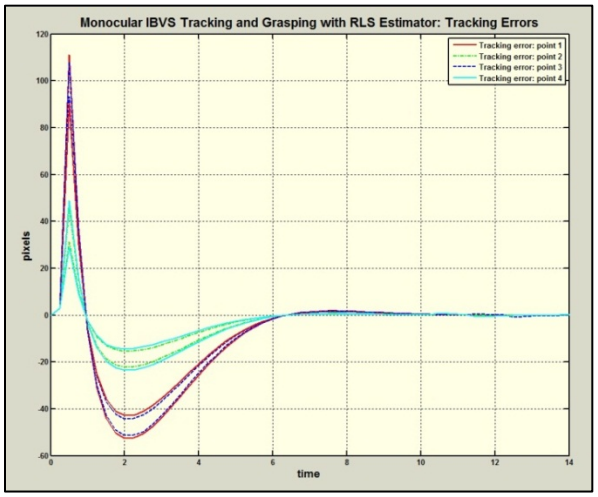
(b)



(c)



(d)

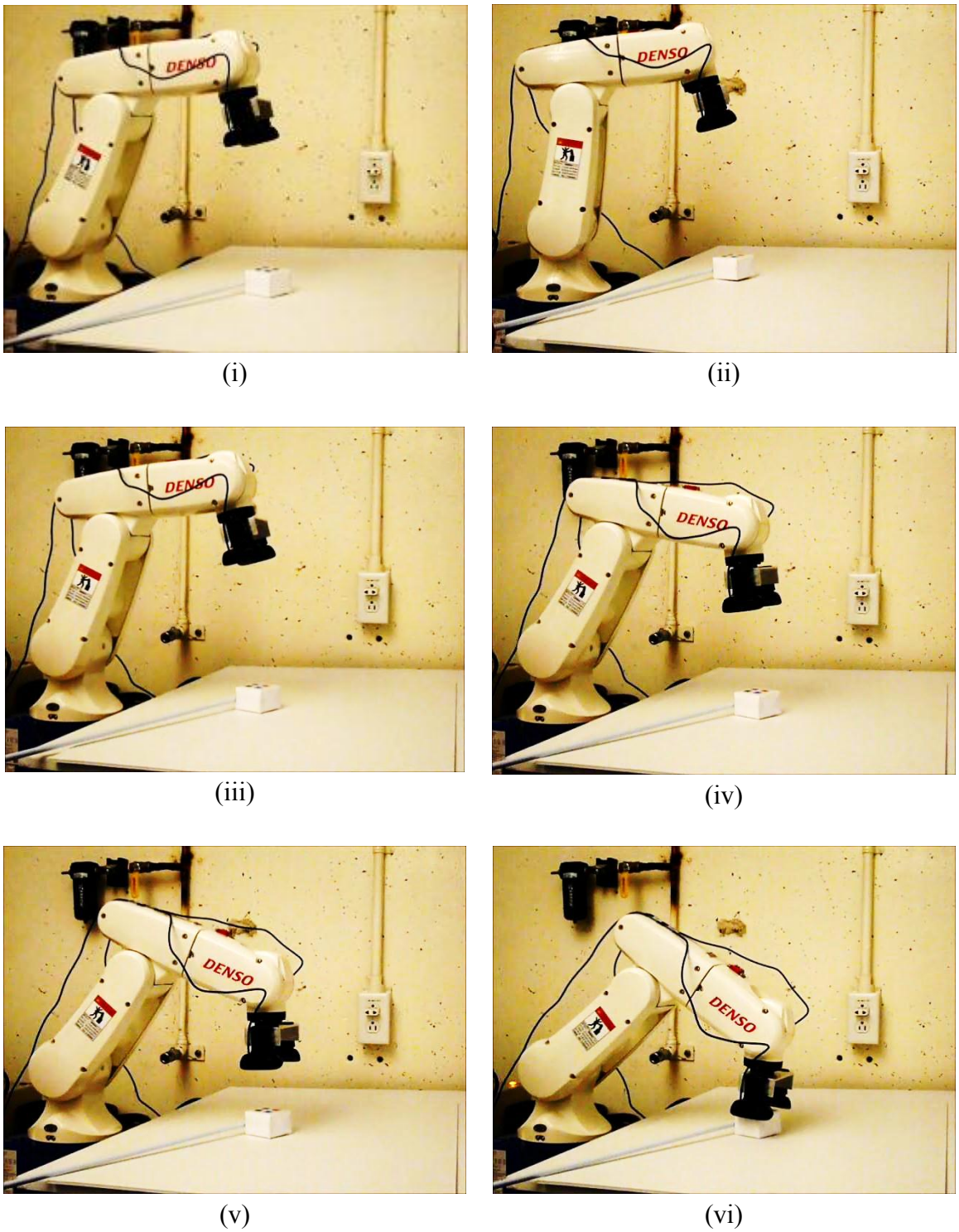


(e)

Figure 5.27 Experimental results for the Monocular IBVS system in a procedure of grasping a moving object using **RLS** Estimator: (a) Image feature trajectories (b) Camera frame velocity components (c) Image feature errors (d) Robot end-effector 3D positions (e) Tracking Errors

It can be inferred from the experimental results that the image feature trajectories in the systems with Kalman filter and EKF are almost the same and they are both quite smoother than the system with RLS estimator. Tracking and convergence performance in the system with Extended Kalman estimator is the highest among all. In addition, the 3-D end-effector trajectory in the system with EKF estimator from the starting point to the catching position has less unnecessary motion and is smoother and totally shows a better behavior than other two methods. The camera velocity components in the system with EKF in comparison with the system with RLS estimator started with relatively low speeds.

Using the same filtering, the state transition noise covariance and the measurement noise covariance properties, an image-based stereo visual servoing system is used for the same grasping procedure with an EKF, a Kalman and a RLS estimator. The stereo system consists of two cameras which are located at a distance of 11 *cm* with respect to the origin of sensor frame. In order to keep sufficient feature points on the object in both cameras fields of view during the grasping procedure, the cameras are tilted about their X axis with 30 and -30 degrees. Figure 5.28 shows the video sequences of the procedure of tracking and grasping a moving object through Stereo image based visual servoing using 6-DOF DENSO robot.



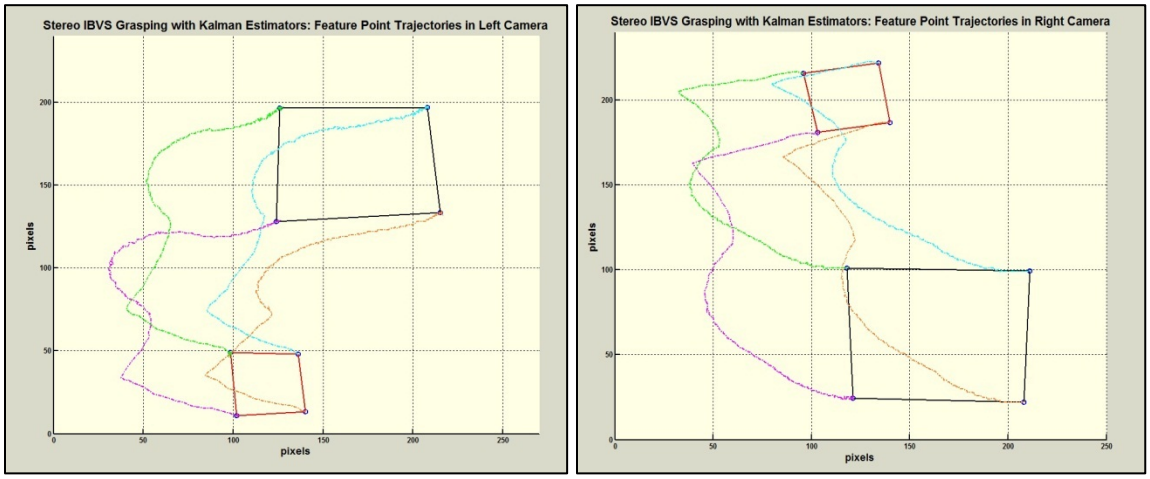
(i) (ii)

(iii) (iv)

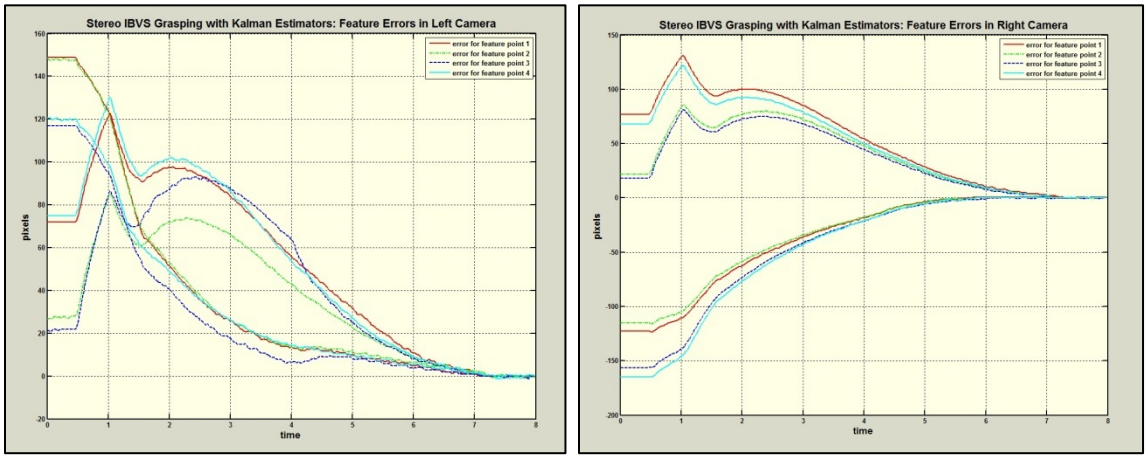
(v) (vi)

Figure 5.28 The procedure of tracking and grasping a moving object through Stereo image based visual servoing using 6-DOF DENSO robot

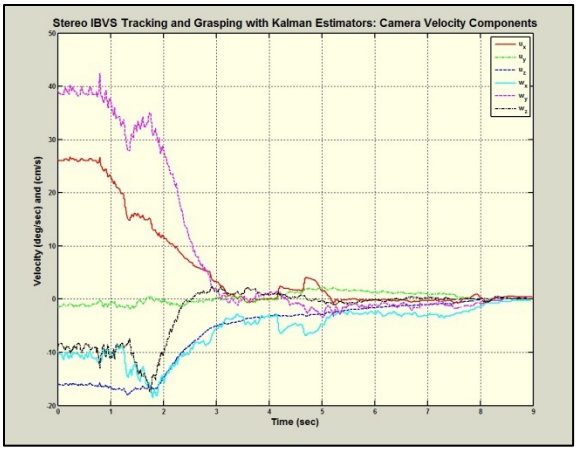
The simulation results for this stereo vision system are given in Figures 5.29-5.31.



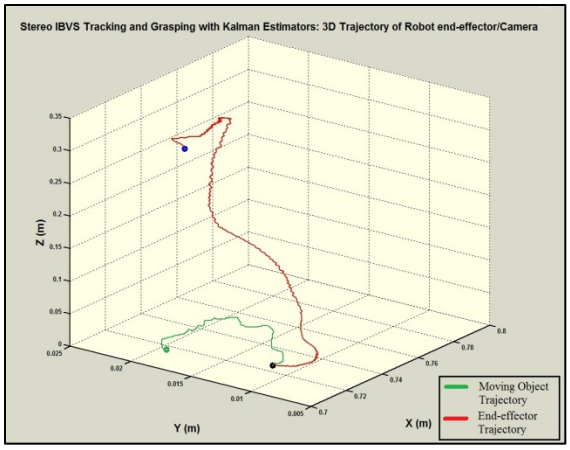
(a)



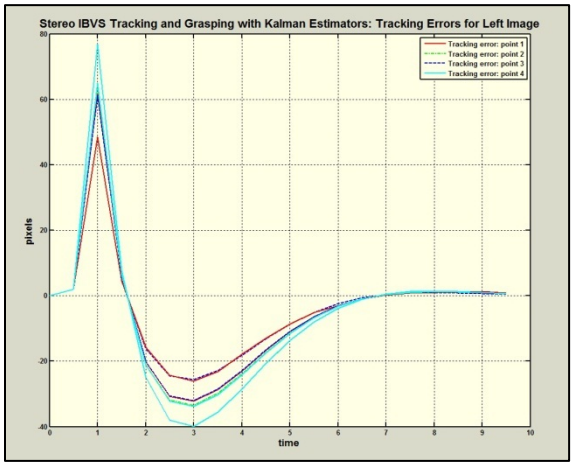
(b)

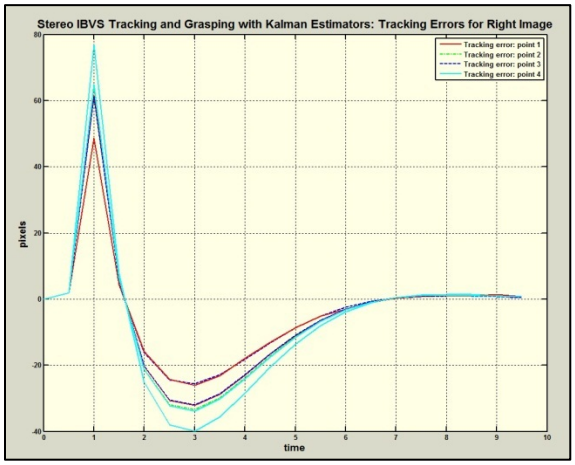


(c)



(d)

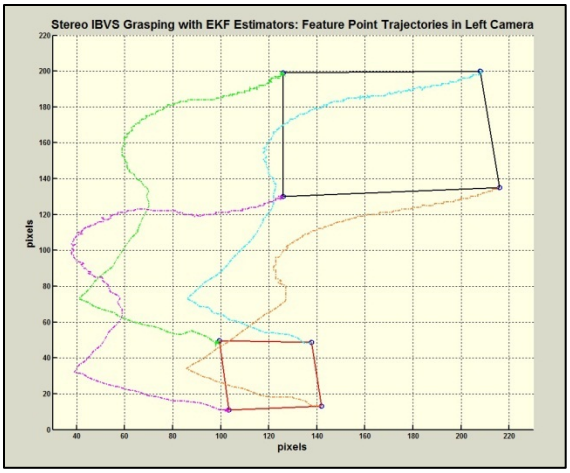


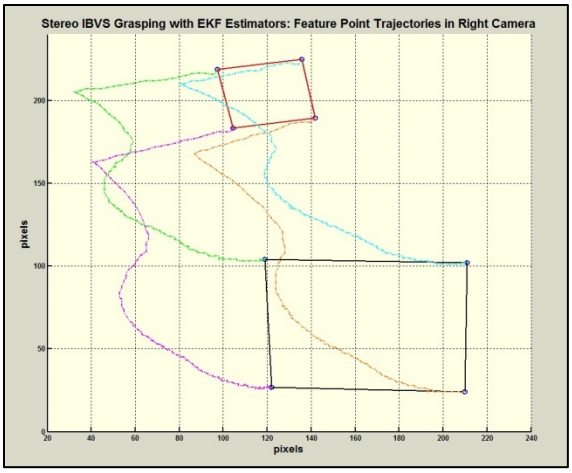


(e)

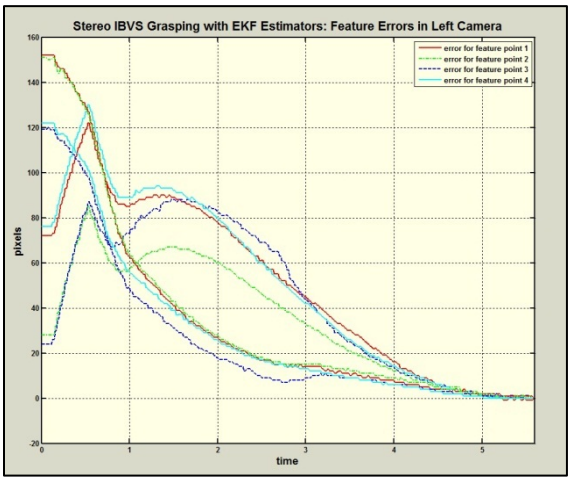
Figure 5.29 Experimental results for the Stereo IBVS system in a procedure of grasping a moving object using **Kalman** Estimator: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

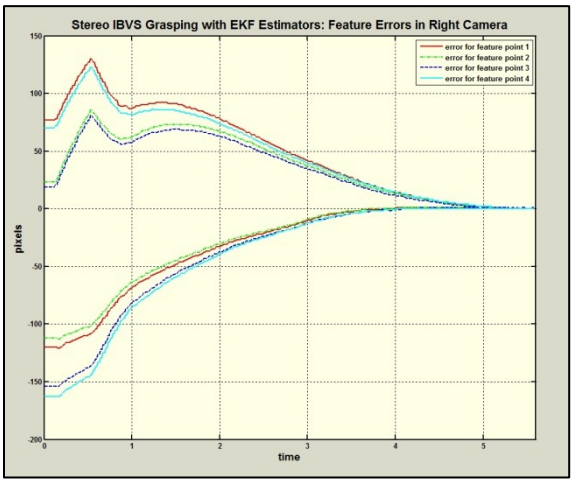
The experimental results of using the indicated stereo visual servoing system with an EKF estimator for tracking and grasping a moving object are illustrated in Figure 5.30.


Stereo IBVS Grasping with EKF Estimators: Feature Point Trajectories in Left Camera
pixels
pixels

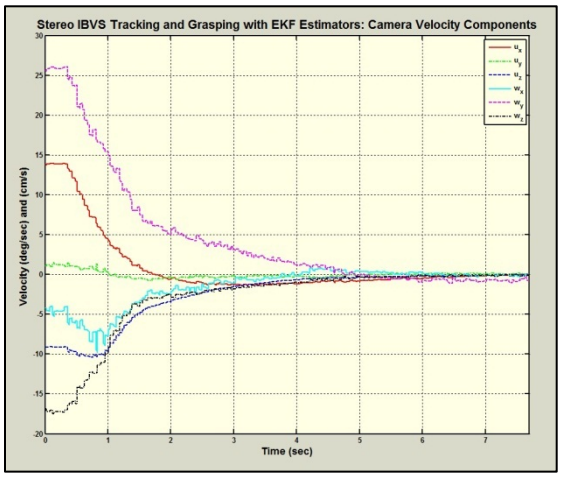

Stereo IBVS Grasping with EKF Estimators: Feature Point Trajectories in Right Camera
pixels
pixels

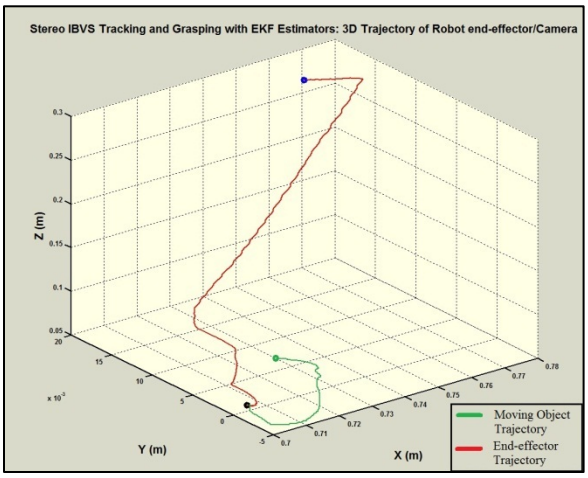
(a)


Stereo IBVS Grasping with EKF Estimators: Feature Errors in Left Camera
error for feature point 1
error for feature point 2
error for feature point 3
error for feature point 4
pixels
time

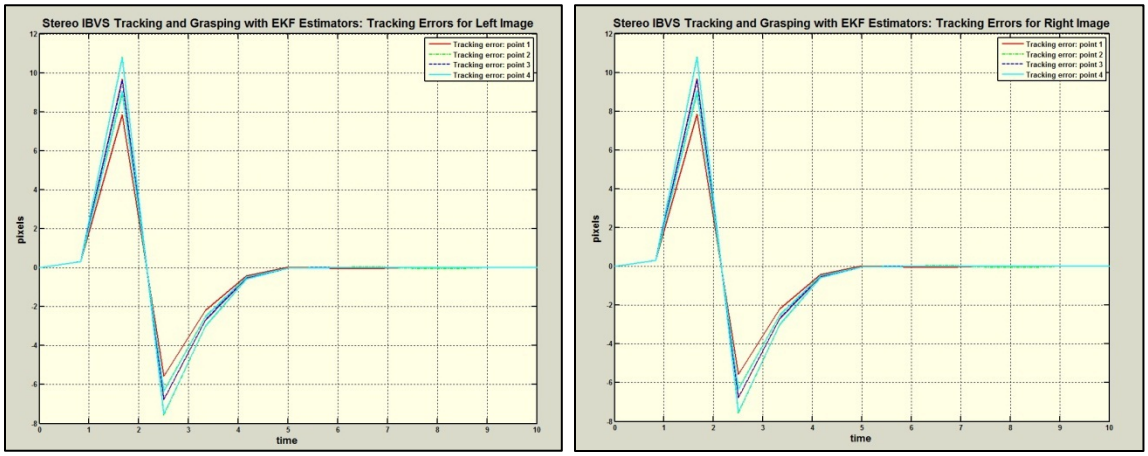

Stereo IBVS Grasping with EKF Estimators: Feature Errors in Right Camera
error for feature point 1
error for feature point 2
error for feature point 3
error for feature point 4
pixels
time

(b)


Stereo IBVS Tracking and Grasping with EKF Estimators: Camera Velocity Components
Velocity (deg/sec) and (cm/s)
Time (sec)


Stereo IBVS Tracking and Grasping with EKF Estimators: 3D Trajectory of Robot end-effector/Camera
Z (m)
Y (m)
X (m)
Moving Object Trajectory
End-effector Trajectory

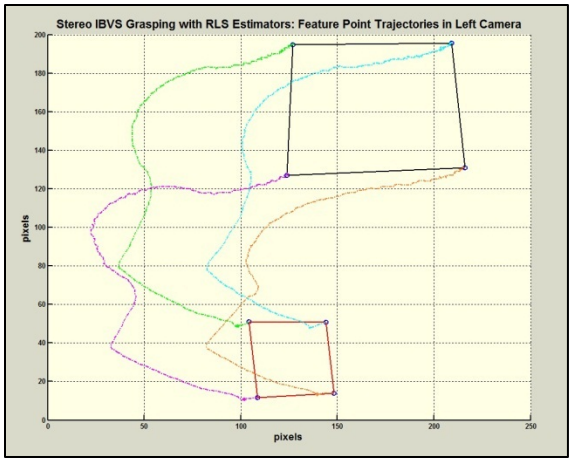
(c) (d)

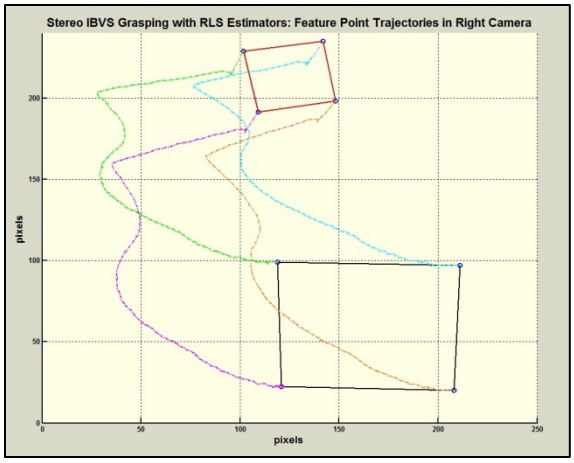


(e)

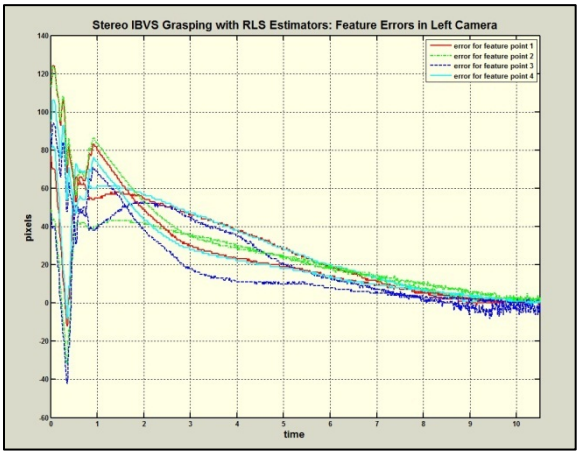
Figure 5.30 Experimental results for the Stereo IBVS system in a procedure of grasping a moving object using **Extended Kalman Filter**: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

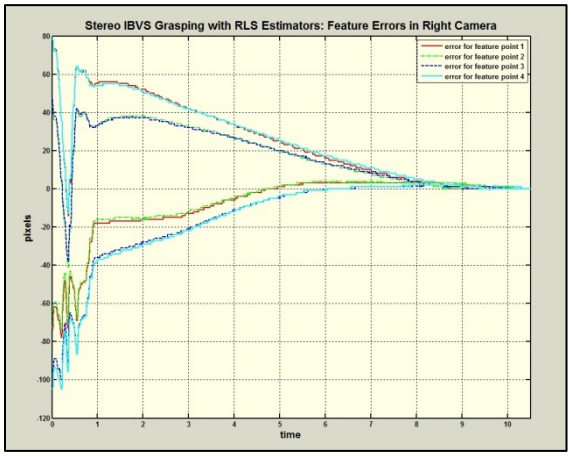
Figure 5.31 illustrates the results for the task of tracking and grasping a moving object with the stereo IBVS system using a RLS estimator.


Stereo IBVS Grasping with RLS Estimators: Feature Point Trajectories in Left Camera
pixels
pixels

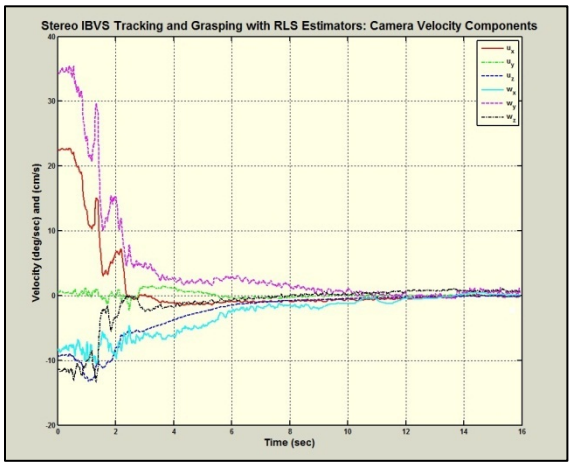

Stereo IBVS Grasping with RLS Estimators: Feature Point Trajectories in Right Camera
pixels
pixels

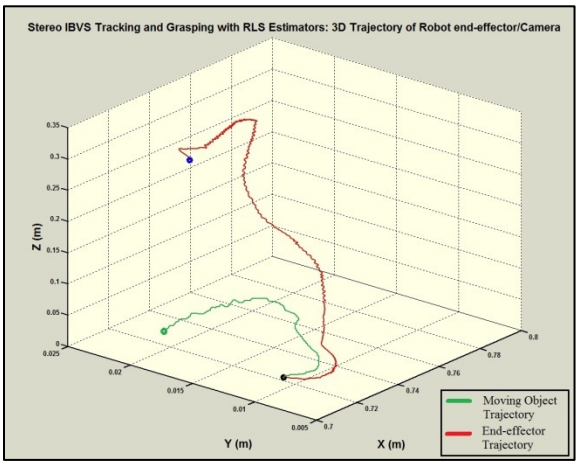
(a)


Stereo IBVS Grasping with RLS Estimators: Feature Errors in Left Camera
error for feature point 1
error for feature point 2
error for feature point 3
error for feature point 4
pixels
time

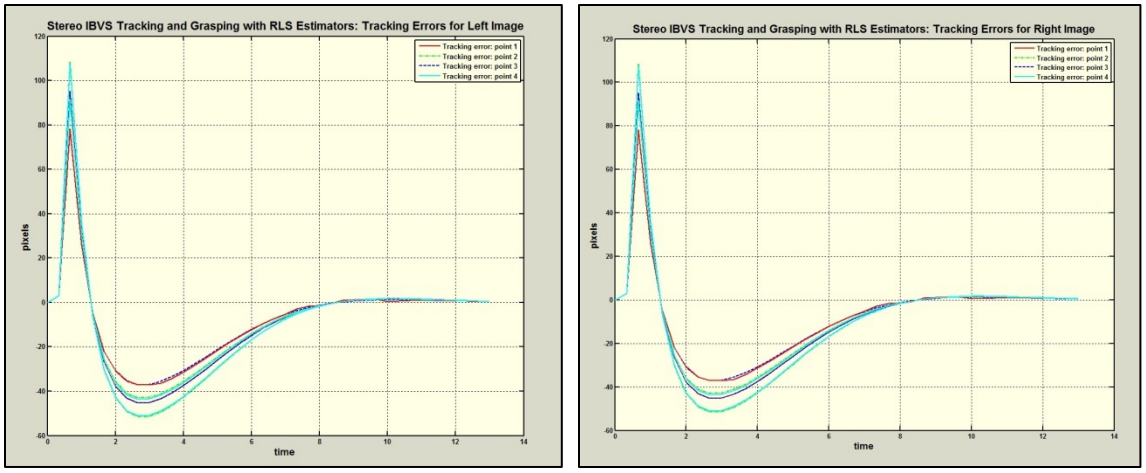

Stereo IBVS Grasping with RLS Estimators: Feature Errors in Right Camera
error for feature point 1
error for feature point 2
error for feature point 3
error for feature point 4
pixels
time

(b)


Stereo IBVS Tracking and Grasping with RLS Estimators: Camera Velocity Components
Velocity (deg/sec) and (cm/s)
Time (sec)


Stereo IBVS Tracking and Grasping with RLS Estimators: 3D Trajectory of Robot end-effector/Camera
Z (m)
Y (m)
X (m)
Moving Object Trajectory
End-effector Trajectory

(c) (d)



(e)

Figure 5.31 Stereo IBVS system with tilted (non-parallel) cameras behavior in a procedure of grasping a moving object using a **RLS** Estimator: (a) Image feature trajectories in left and right images (b) Image feature errors for left and right cameras (c) Camera frame velocity components (d) Robot end-effector 3D positions (e) Tracking errors in left and right images

It is quite remarkable from experimental results that in comparison with the monocular system, the trajectories of the points in the images from the stereo system are smoother and the camera velocity components do not include large oscillations and start from smaller values. It can be also inferred from the results that the case with the EKF estimator shows better tracking and convergence performance and has a better behavior in end-effector 3-D trajectories. The camera velocity components in the system with EKF in comparison with the system with RLS estimator starts with relatively lower values.

Once more, the image feature trajectory behaviors in all of three systems seem to be almost the same but the system has a faster convergence. This better performance is attributed to the fact that it is possible to calculate the exact image interaction matrix at any

position without knowing a model of the target object. The speed of convergence will be faster and the oscillations of the graphs will be smaller if more precise calibration is performed for the cameras and most importantly stereo rig. If the transformation matrices are not properly calculated, there exists a persistent rotation velocity around z direction, w_z, due to calibration error.

Table 5.2 shows a summarization of the comparison results for all the tracking and grasping cases.

Table 5.2 Comparison the experimental results for all the IBVS cases for tracking and grasping of a moving object

IBVS Case	Convergence Time (sec)	Sensor Frame Angular Vel. at t=0 s (deg/sec)	Sensor Frame Linear Vel. at t=0 s (cm/s)	Maximum Tracking Error (pixels)
Monocular+Kalman	8.8	101	42	91
Monocular+EKF	6.5	94	32	26
Monocular+RLS	11.8	109	45	110
Stereo+Kalman	7.2	39	19	77
Stereo+EKF	5.5	31	15	11
Stereo+RLS	10	41	21	108

5.7. Summary

In order to verify the simulation results, this chapter has been devoted to describing the experimental setup including the 6-DOF Robot, controller, vision system, stereo rig and camera calibration process and also results for proposed image-based stereo visual servoing system for various servoing tasks and most importantly for a mission of tracking and grasping a moving object. In this chapter the visual servoing control system and the tracking algorithms are implemented with three different estimators: Kalman Filter, Extended Kalman Filter and Recursive Least Square. All three estimators have been added to both Monocular and Stereo systems and at the end all of the cases have been compared to each other and the results have been presented and discussed.

Chapter 6 : Conclusion and Future Works

In this chapter, the main conclusions and contributions of this thesis are summarized from the obtained results and then some possible extensions and future works are suggested as well.

6.1. Contributions and Research Conclusion

This thesis presents a novel eye-in-hand image-based stereo visual servoing system for a real-time task of tracking and grasping a moving object in an uncalibrated environment. The main contributions and conclusions of this research work can be listed as follows:

- An image-based visual servoing (IBVS) approach based on stereo vision has been presented and mathematically discussed and compared to the case of Monocular IBVS.
- The effect of having a moving target and corresponding feature points on visual servoing system and governing equations has been considered and mathematically discussed.
- The method for stacking the proper image interaction matrices for the case of image based stereo visual servoing has been developed for two cases of parallel and non-parallel cameras and the exact depth information has been extracted from the geometry of the vision system and used in image interaction matrices.
- A method for trajectory estimation of a moving object has been proposed to predict the position of the object which is used in an image-based stereo visual servoing for a real-time grasping procedure. The system dynamics of the object has been modeled in both linear and nonlinear description in image plane instead of 3-D space. Various trajectory estimation algorithms such as "Kalman Filtering", "Recursive Least Square" and "Extended Kalman Filtering (EKF)" have been used to predict the position of moving object in image planes.
- The robustness of the proposed visual servoing system has been examined through computer simulations and experiments. The precise depth estimation of the object and new image interaction matrices have been used as well as various motion predictors such as: an extended Kalman filter, Kalman filter or a recursive least square method.

6.2. Future Works

Although the effectiveness of the proposed image-based method in tracking and grasping a moving object has been examined in the current research work and verified by the computer simulations and experimental results, there exist a number of limitations in practical tasks and real-world applications. Accordingly, the most important problems are identified and addressed as follows:

- Due to speed limitations of the 6-DOF robot joints, the target should not have a high speed. Thus in some cases such as a flying thrown object, this method cannot show a very successful effort.
- The advantage of using an eye-in-hand camera configuration is that there is no need to calibrate the environment and measure the geometry of the framework. Since the cameras are mounted on the robot, the fields of sight are limited and cannot cover the whole environment around the robot. Thus, the space for the target maneuvers is limited.
- The classical proportional IBVS scheme cannot deal with nonlinear constraint such as joint limits, actuator saturations and visibility constraints.

As the related future research works and to overcome the mentioned problems and limitations, the following topics are suggested:

- Using an active (actuated) eye-to-hand stereo vision system to extend the vision system visibility constraints.
- Utilizing Adaptive or Fuzzy time series-based prediction and trajectory estimation for a better tracking and prediction performance.
- Using new image features such as lines and image moments to improve the robustness of the visual servoing system.
- Employing a new IBVS scheme based on model predictive control to deal with nonlinearities.

References

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

[2] S. I. A. I. Center and G. J. Agin, *Real Time Control of a Robot with a Mobile Camera*: SRI International, 1979.

[3] P. Corke, "Visual Control of Robot Manipulators - A Review," in *Visual Servoing*, 1993, pp. 1-31.

[4] A. C. Sanderson and L. E. Wciss, "Image-based visual servo control using relational graph editor signals," *Proc. IEEE,* pp. 1074-1077, 1980.

[5] P. Corke, *Visual control of robots : high-performance visual servoing* vol. 2. Taunton England ; New York, NY: Research Studies Press ; Wiley, 1996.

[6] J. T. Feddema and O. R. Mitchell, "Vision-guided servoing with feature-based trajectory generation [for robots]," *Robotics and Automation, IEEE Transactions on,* vol. 5, pp. 691-700, 1989.

[7] B. Espiau*, et al.*, "A new approach to visual servoing in robotics," *Robotics and Automation, IEEE Transactions on,* vol. 8, pp. 313-326, 1992.

[8] Faunc-Robotics. (2009). *FANUC Robotics Industrial Robots and Robotic Automated Systems*. Available: http://www.fanucrobotics.com

[9] Robot-Worx. (2008). *Robot Use in the Present and Future*. Available: http://www.robots.com/articles/viewing/robot-use-in-the-present-and-future/535

[10] W. Guo-Qing*, et al.*, "Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation," *Engineering in Medicine and Biology Magazine, IEEE,* vol. 16, pp. 40-45, 1997.

[11] D. B. Samadi. (2012). *da Vinci Si robotic system wants to be your surgeon*. Available: http://www.roboticoncology.com

[12] S. S. Mehtatt*, et al.*, "Visual servo control of an unmanned ground vehicle via a moving airborne monocular camera," in *American Control Conference, 2006*, 2006, p. 6 pp.

[13] Macroswiss. (2012). *Macroswiss Unmanned Ground Vehicles*. Available: http://www.armedforces-int.com/article/micro-spy-robot.html

[14] S. Ye*, et al.*, "Study on intelligent visual servoing of space robot for cooperative target capturing," in *Information and Automation (ICIA), 2012 International Conference on*, 2012, pp. 733-738.

[15] G. Flandin*, et al.*, "Eye-in-hand/eye-to-hand cooperation for visual servoing," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 2000, pp. 2741-2746 vol.3.

[16] G. Chesi and K. Hashimoto, "Static-eye against hand-eye visual servoing," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, 2002, pp. 2854-2859 vol.3.

[17] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing The confluence of vision and control." vol. 237, D. Kriegman*, et al.*, Eds., ed: Springer Berlin / Heidelberg, 1998, pp. 66-78.

[18] J. Armstrong Piepmeier*, et al.*, "Uncalibrated eye-in-hand visual servoing," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 2002, pp. 568-573 vol.1.

[19] E. Cervera*, et al.*, "Is 3D useful in stereo visual control?," in *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, 2002, pp. 1630-1635 vol.2.

[20] D.-J. Kim*, et al.*, "Eye-in-hand stereo visual servoing of an assistive robot arm in unstructured environments," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 2009, pp. 2326-2331.

[21] N. Maru*, et al.*, "Manipulator control by using servoing with the stereo vision," in *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, 1993, pp. 1866-1870 vol.3.

[22] R. Mahony*, et al.*, "Choice of image features for depth-axis control in image based visual servo control," presented at the IEEE International Conference on Intelligent Robots and Systems, 2002.

[23] M. A. Arlotti and M. N. Granieri, "A perception technique for a 3-D robotic stereo eye-in-hand vision system," in *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, 1991, pp. 1626-1629 vol.2.

[24] A. C. Sanderson and L. E. Weiss, "Image Based Visual Servo Control Using Relational Graph Error Signal," presented at the IEEE International Conference of Cybernetics and Society, 1980.

[25] K. Hashimoto and T. Noritsugu, "Performance and sensitivity in visual servoing," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, 1998, pp. 2321-2326 vol.3.

[26] S. Hutchinson*, et al.*, "A tutorial on visual servo control," *Robotics and Automation, IEEE Transactions on,* vol. 12, pp. 651-670, 1996.

[27] W. J. Wilson*, et al.*, "Relative end-effector control using Cartesian position based visual servoing," *Robotics and Automation, IEEE Transactions on,* vol. 12, pp. 684-696, 1996.

[28] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artif. Intell.,* vol. 31, pp. 355-395, 1987.

[29] G. Chesi and K. Hashimoto, "A simple technique for improving camera displacement estimation in eye-in-hand visual servoing," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 26, pp. 1239-1242, 2004.

[30] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *Robotics & Automation Magazine, IEEE,* vol. 13, pp. 82-90, 2006.

[31] L. Weiss and A. Sanderson, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal of Robotics and Automation,* vol. 3, pp. 404-417, 1987.

[32] P. I. Corke and SpringerLink. (2011). *Robotics, vision and control fundamental algorithms in MATLAB®.* Available: http://dx.doi.org/10.1007/978-3-642-20144-8

[33] F. Chaumette and S. Hutchinson, "Visual Servo Control, Part II: Advanced Approaches," *IEEE Robotics and Automation Magazine,* vol. 14, pp. 109-118, 2007.

[34] F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 2000, pp. 630-635 vol.1.

[35] E. Malis and F. Chaumette, "2-1/2D visual servoing," *IEEE Trans. Robot. Automat.,* vol. 15, pp. 238–250, 1999.

[36] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *Robotics and Automation, IEEE Transactions on,* vol. 17, pp. 507-515, 2001.

[37] P. Y. Oh and K. Allen, "Visual servoing by partitioning degrees of freedom," *Robotics and Automation, IEEE Transactions on,* vol. 17, pp. 1-17, 2001.

[38] P. Martinet and E. Cervera, "Stacking Jacobians properly in stereo visual servoing," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001, pp. 717-722 vol.1.

[39] G. D. Hager*, et al.*, "Robot hand-eye coordination based on stereo vision," *Control Systems, IEEE,* vol. 15, pp. 30-39, 1995.

[40] O. D. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig," presented at the Proceedings of the Second European Conference on Computer Vision, 1992.

[41] F. Alkhalil*, et al.*, "Visual servoing of an articulated object based on stereovision," in *Robotic and Sensors Environments (ROSE), 2011 IEEE International Symposium on*, 2011, pp. 71-76.

[42] A. Castano and S. Hutchinson, "Visual compliance: task-directed visual servo control," *Robotics and Automation, IEEE Transactions on,* vol. 10, pp. 334-342, 1994.

[43] H. Fassler, *et al.*, "A robot ping pong player: optimized mechanics, high performance 3D vision, and intelligent sensor control," *Robotersysteme,* pp. 161-170, 1990.

[44] M.-S. Kim, *et al.*, "Robot Visual Servo through Trajectory Estimation of a Moving Object Using Kalman Filter

Emerging Intelligent Computing Technology and Applications." vol. 5754, D.-S. Huang, *et al.*, Eds., ed: Springer Berlin / Heidelberg, 2009, pp. 1122-1130.

[45] P. K. Allen, *et al.*, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *Robotics and Automation, IEEE Transactions on,* vol. 9, pp. 152-165, 1993.

[46] H. Nomura and T. Naito, "Integrated visual servoing system to grasp industrial parts moving on conveyer by controlling 6DOF arm," in *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, 2000, pp. 1768-1775 vol.3.

[47] G. Sen Gupta, *et al.*, "Identification and prediction of a moving object using real-time global vision sensing," in *Instrumentation and Measurement Technology Conference, 2003. IMTC '03. Proceedings of the 20th IEEE*, 2003, pp. 1402-1406 vol.2.

[48] G. Li and J. Zhao, "A Real-time Stereo Visual Servoing for Moving Object Grasping Based Parallel Algorithms," in *Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference on*, 2007, pp. 2886-2891.

[49] P. Y. Yeoh and S. A. R. Abu-Bakar, "Accurate real-time object tracking with linear prediction method," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, 2003, pp. III-941-4 vol.2.

[50] Ji, *et al.*, "Learning efficient linear predictors for motion estimation," presented at the Proceedings of the 5th Indian conference on Computer Vision, Graphics and Image Processing, Madurai, India, 2006.

[51] J. F. Pacheco, *et al.*, "Binocular visual tracking and grasping of a moving object with a 3D trajectory predictor," *Journal of Applied Research and Technology,* vol. 7, pp. 259-274, 2009.

[52] B. Lamiroy, *et al.*, "Controlling robots with two cameras: how to do it properly," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, 2000, pp. 2100-2105 vol.3.

[53] B. K. P. Horn, *Robot Vision* Cambridge, MA: MIT Press, 1986.

[54] E. Malis, "Improving vision-based control using efficient second-order minimization techniques," in *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 2004, pp. 1843-1848 Vol.2.

[55] J. J. Craig, *Introduction to Robotics: Mechanics and Control*: Addison-Wesley, 1989.

[56] H. Kase*, et al.*, "Visual servoing of the manipulator using the stereo vision," in *Industrial Electronics, Control, and Instrumentation, 1993. Proceedings of the IECON '93., International Conference on*, 1993, pp. 1791-1796 vol.3.

[57] R. Berg, "Estimation and prediction for maneuvering target trajectories," *Automatic Control, IEEE Transactions on,* vol. 28, pp. 294-304, 1983.

[58] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," University of North Carolina at Chapel Hill1995.

[59] P. K. Allen*, et al.*, "Trajectory filtering and prediction for automated tracking and grasping of a moving object," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, 1992, pp. 1850-1856 vol.2.

[60] M. Yi*, et al.*, "Vision guided navigation for a nonholonomic mobile robot," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, 1997, pp. 3069-3074 vol.3.

[61] J. Park*, et al.*, "Trajectory estimation of a moving object using Kalman filter and Kohonen networks," *Robotica,* vol. 25, pp. 567-574, 2007.

[62] S. M. LaValle and R. Sharma, "On Motion Planning in Changing, Partially Predictable Environments," *The International Journal of Robotics Research,* vol. 16, pp. 775-805, December 1, 1997 1997.

[63] P. I. Corke, "A robotics toolbox for MATLAB," *Robotics & Automation Magazine, IEEE,* vol. 3, pp. 24-32, 1996.

[64] P. I. Corke, "The Machine Vision Toolbox: a MATLAB toolbox for vision and vision-based control," *Robotics & Automation Magazine, IEEE,* vol. 12, pp. 16-25, 2005.

[65] P. I. Corke, "Robotics, Vision and Control," 2012.

[66] J. Postel, "User Datagram Protocol IP protocol," *RFC 760, USC/Information Sciences Institute,,* January 1980.

[67] Denso-Robotics, "5 and 6 Axis Articulated Robots," 2012.

[68] Quanser, "Quanser 6-Axis Articulated Robot- Open Architecture Enabled with QUARC," 2011.

[69] Wikipedia, "HSL and HSV Color space," 2012.

[70] "Quancer Quarc 2.2 - Online User information," 2013.

Appendix A: 6-DOF DENSO® Robot and Quanser® Controller

This chapter is devoted to presenting some general specifications and descriptions of the DENSO robot and QUANSER controller unit and QUARC control software.

A.1. Denso 6-Axis Robot Specifications

The Denso 6-Axis robot system includes a 6-DOF robotic arm and a Quanser control module. The Denso arm consists of six joints and corresponding six encoders that measure the angular position of the six motors. The encoders and motors specifications are summarized in Table (A.1). The encoders resolution, motors gear ratios, motors torque constants, and joints hard stop limits are listed in this Table.

Table A.1 Motor and encoder calibration specifications for Denso 6-Axis VP6242G [68]

Motor / Encoder Starting with Base	*Motor Gear Ratio*	*Encoder Calibration (count/deg)*	*Torque Constant N.m/Amp*	*Joint maximum stop limit (deg)*	*Joint minimum stop limit (deg)*
1	120:1	43690.666670	0.38	160	-160
2	160:1	58254.222220	0.38	120	-120
3	120:1	43690.666670	0.22	160	20
4	100:1	36408.888890	0.21	160	-160
5	100:1	36408.888890	0.21	120	-120
6	100:1	36408.888890	0.21	360	-360

Figure A.1 demonstrate the robot joint coordinate systems including the world frame 0 and the joint frames which are used to define the kinematics/inverse kinematics and Jacobian matrix. Link lengths are also illustrated in Figure A.1 where the robot is in a completely straightened up situation. In this configuration, all the joints encoder values are zero and the axes in frames 1, 3, 4, 5, and 6 are parallel to their counterpart axes in global frame. The joints 2, 3, and 5 are zero when the robot is completely straightened up as depicted in Figure A.1. Figure A.2 demonstrates the Denso robot workable space from right and top view.

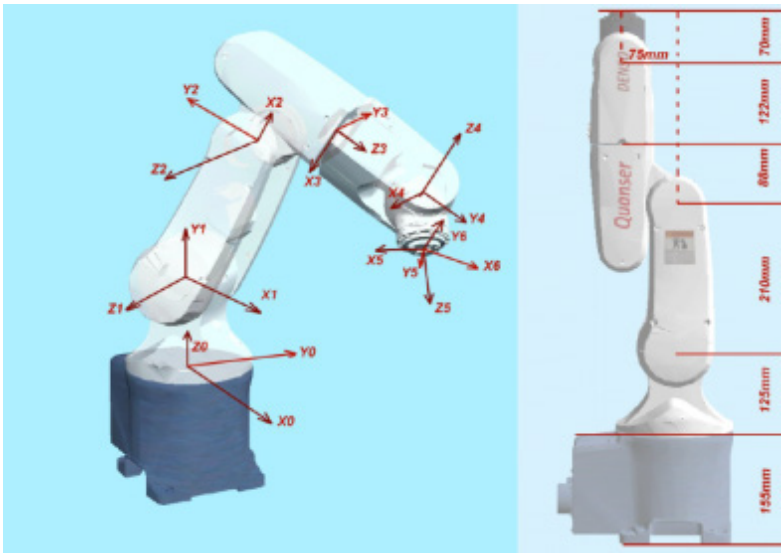


Figure A.1 World frame, joint frames and dimensions used for kinematics calculation [68]

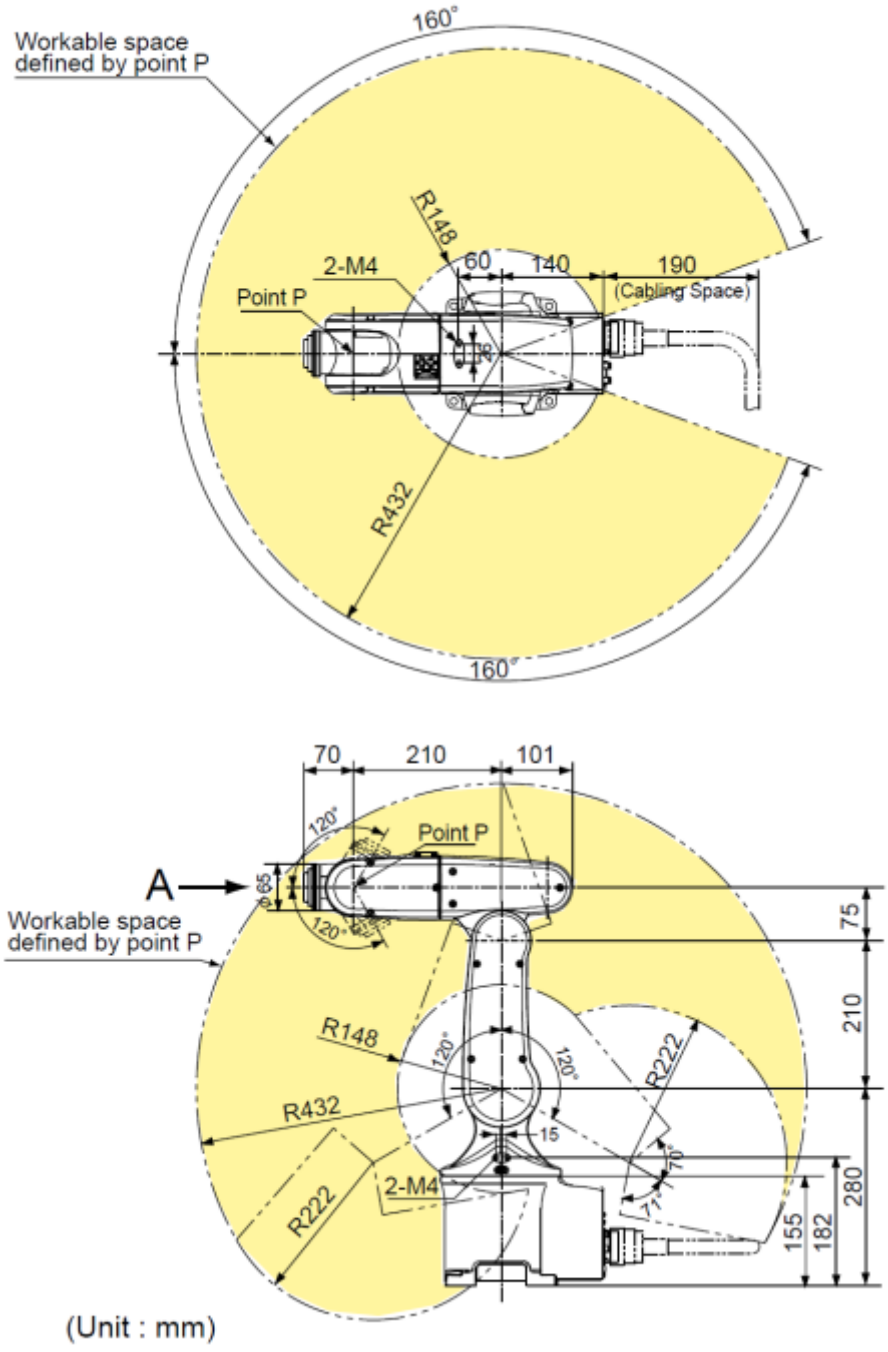


Figure A.2 Denso robot work-space [68]

A.2. QUARC® Software

QUARC® is a multi-functional interface software which is able to connect to Simulink for the purpose of implementations and hardware-in-the-loop experiments [70]. QUARC provides Specific Simulink block sets so that the Simulink can establish a connection with the hardware. These block sets vary depending on the hardware and it capabilities. In order to implement various experiments with the Denso robot, QUARC provides two prepared Simulink blocks for; “Denso Read” and “Denso Write”. Denso Read starts a connection with the Denso robot controller to read the robot joint positions and state and send it to other parts of the Simulink files. Denso write sends the joint commands to the control unit and also gives the ability to set the PID controller gains for position and velocity control mode of the robot. The feed forward current command can also be sent to the robot using this block. User has the option of using potion control or velocity control through the function of this block.

Appendix B: Design and Implementation of a Controller unit for 6-DOF PUMA® 260 Robot

The Department of Mechanical and Industrial Engineering at Concordia University, Canada had, for a number of years, a functional PUMA 260 manipulator robot with its original control hardware and human interfaces. This chapter describes the procedure of designing and implementation of a controller unit consisted of interfacing the robot arm with a PC and developing software for the purpose of real-time implementation of a visual servoing system.

The PUMA 260 is a six-degree-of freedom robotic manipulator that uses six dc servomotors for joint control. Joint positions are measured using encoders and potentiometers. Three large motors provide control of the waist, shoulder, and elbow, while three smaller motors position the orientation of the wrist. A picture of PUMA-260 and original

hardware is shown in Figure B.1 while Figure B.2 shows a schematic of this robot and Table B.1 includes the technical specifications.

Figure B.1 PUMA260 and original hardware [41]

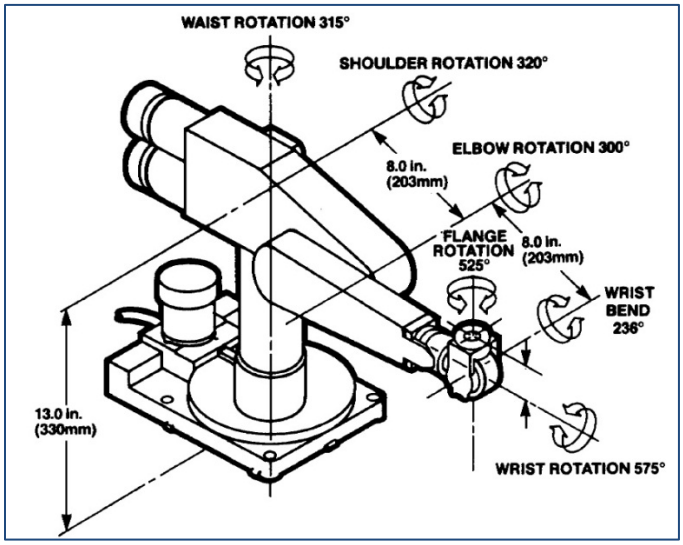


Figure B.2 Schematic of PUMA260 [41]

Table B.1 PUMA260 Technical Specifications

Performance		Physical Characteristics	
REPEATABILITY	±.002 in. (±0.05 mm)	ARM WEIGHT	15 lbs. (6.8 Kg)
LOAD CAPACITY	2.2 lbs. (1.0 Kg)	CONTROLLER SIZE	19" x 12.5" x 23.6" (475 mm x 312.5 mm x 590 mm)
STRAIGHT LINE VELOCITY	49.0 in/s max. (1.25 m/s max.)	CONTROLLER WEIGHT	80 lbs. (36 Kg)
ENVIRONMENTAL REQUIREMENTS	50-120°F (10-50°C) 80% humidity (non-condensing). Shielded against industrial line fluctuations and human electro-static discharge	CONTROLLER CABLE LENGTH	15 ft. (4.5 m) Standard 50 ft. (15 m) Optional

B.1. Robot Arm Kinematics

It is a common practice for the analysis of the robot arm system to define a world coordinate system and a local coordinate system attached to each joint. In order to design a joint control system for PUMA we need to derive a forward kinematic model to obtain position and velocity relations. Defining Denavit-Hartenberg (D-H) parameters and finding homogenous transformations we can obtain following results. Table B.2 contains D-H parameters for every link of the robot and Figure B.3 shows the corresponding model. The workspace of the robot's end effector can also be found in Figure B.4.

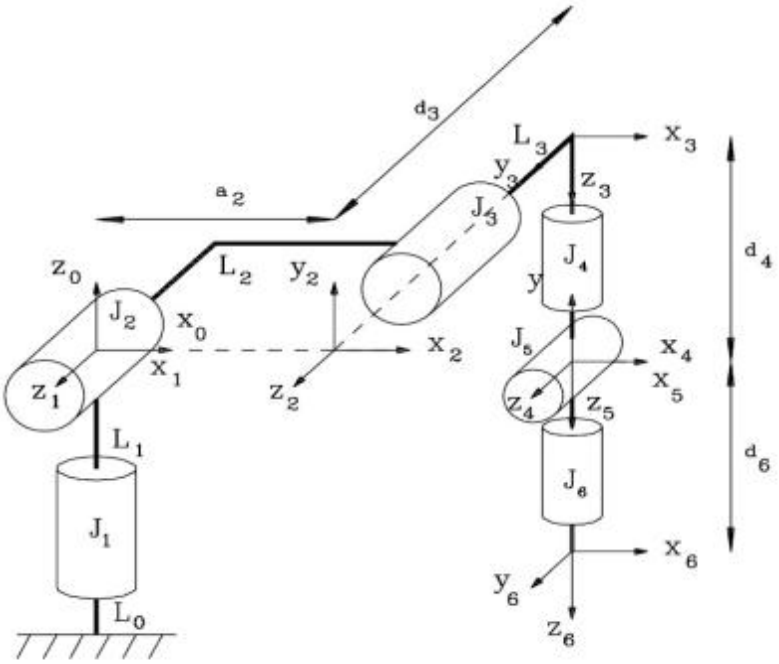


Figure B.3 Puma 260 D-H parameters [14]

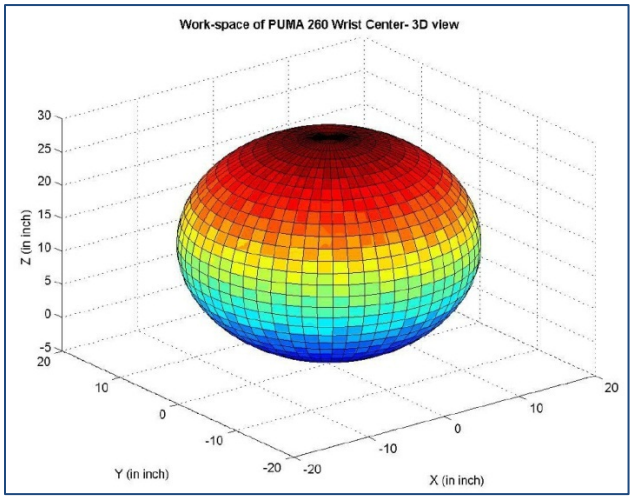


Figure B.4 Puma 260 workspace

Table B.2 PUMA 260 D-H Parameters

	d	θ	a	α
L1	0	θ_1	0	90^o
L2	0	θ_2	a_2	0^o
L3	$-d_3$	θ_3	0	90^o
L4	d_4	θ_4	0	-90^o
L5	0	θ_5	0	90^o
L6	d_6	θ_6	0	0^o

$$ {}^{0}T_{3} = \begin{bmatrix} C_1C_2C_3 - C_1S_2S_3 & S_1 & C_1S_2C_3 + C_1C_2S_3 & a_2C_1C_2 - d_3S_1 \\ C_2C_3S_1 - S_1S_2S_3 & -C_1 & S_1S_2C_3 + S_1C_2S_3 & d_3C_1 + a_2S_1C_2 \\ S_2C_3 - C_2S_3 & 0 & S_2S_3 - C_2S_3 & a_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(B.1)} $$

$$ {}^{3}T_{6} = \begin{bmatrix} C_4C_5C_6 - S_4S_6 & -(S_4C_6) - C_4C_5S_6 & C_4S_5 & d_6C_4S_5 \\ S_4C_5C_6 + C_4S_6 & C_4C_6 - S_4C_5S_6 & S_4S_5 & d_6S_4S_5 \\ -C_6S_5 & S_5S_6 & C_5 & d_4 + d_6C_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(B.2)} $$

$$ {}^{0}T_{6} = {}^{0}T_{3}\ {}^{3}T_{6} \quad \text{(B.3)} $$

B.2. Hardware Architecture

The designed robot controller unit consists of the following hardware parts:

- A computer with PCI slot
- National Instrument® PCI-7344 Motion Controller Board
- Six Advanced Motion® servo amplifiers (30A20AC) for each joint

- National Instrument® UMI 7764 accessory (Universal Motion Interface)
- Cables for motion an digital I-O connectors and related adapters

Figure B.5 shows the hardware components and connections as an architecture overview of the system. The software user interface sends signals through PC to PCI motion controller board and this board sends voltage regulations to servo amplifier and proper velocity and torque are applied to joint motors with the current and voltage signals sent from servo amplifier. Figures B.6-B.7 show the final system and hardware connected to PUMA 260 and a Vision-Based Control test-bed.

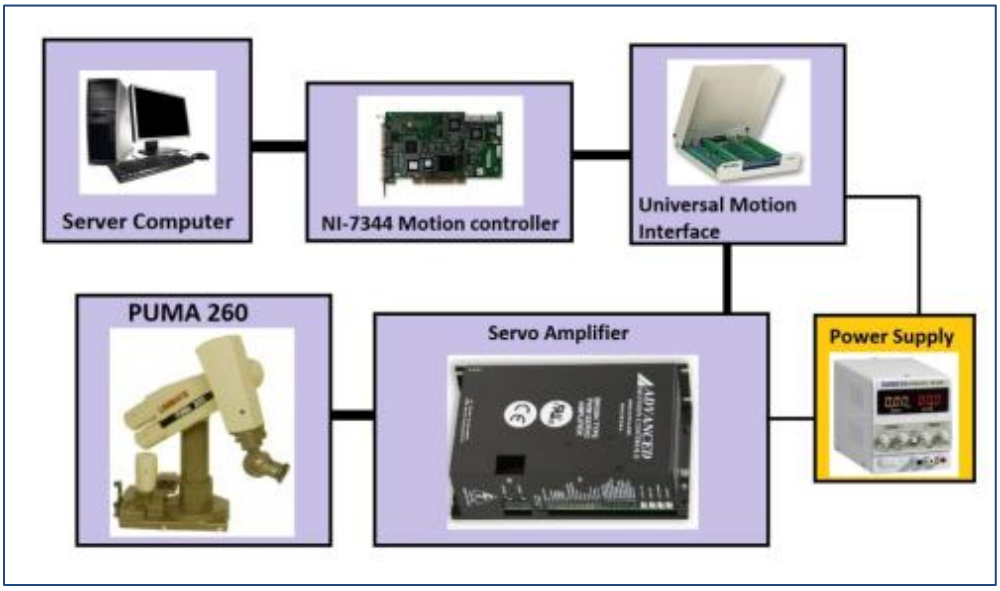


Figure B.5 Hardware components and connections of controller unit

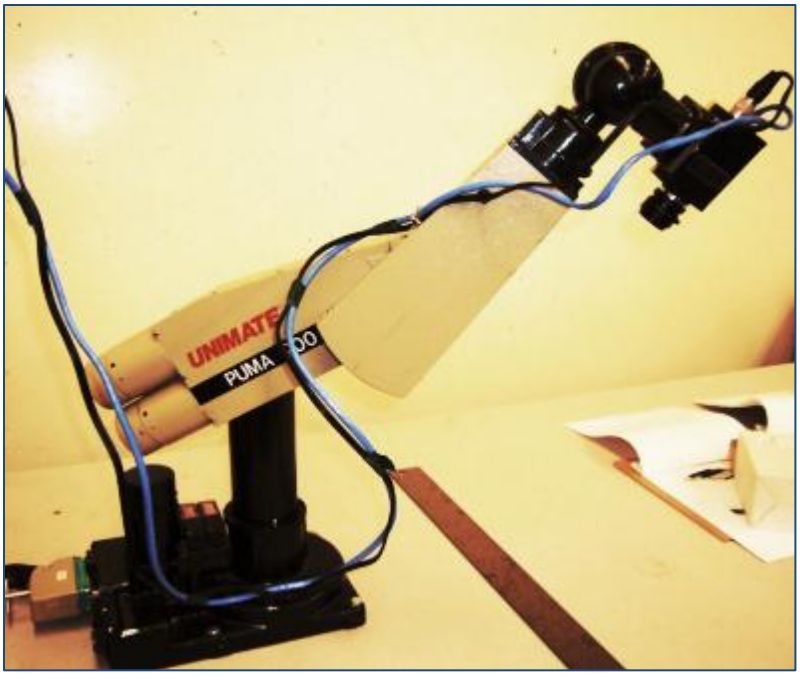


Figure B.6 PUMA 260 with a camera mounted on the End-Effector

Figure B.7 The final control system and hardware connected to PUMA 260

B.3. Software and Graphical User Interface

The implemented software architecture is based on National Interface LabView® and its related block-sets running under Microsoft® Windows 7. LabView enables rapid design of control algorithms and establish a stable hardware interface with motion control boards and also allows specific functions to be implemented in C code as S-functions. Figure B.8 shows Ni-Motion Assistant software and the GUI designed by NI-LabView to control the robot joints.

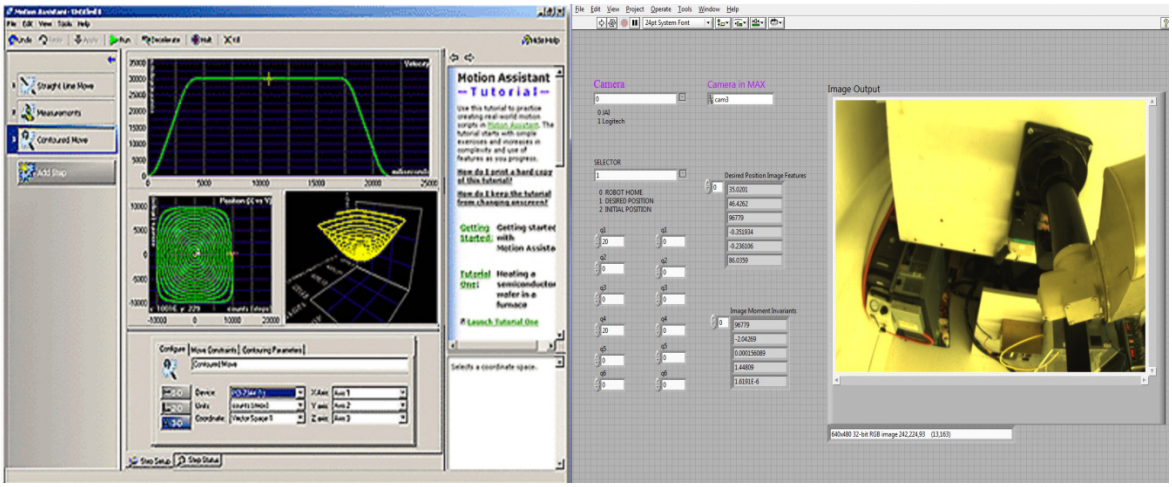


Figure B.8 Ni-Motion Assistant software and the GUI designed by NI-LabView to control the robot joints