Visual Servoing of a 5-DOF Mobile Manipulator using
a Panoramic Vision System

A Thesis
submitted to the Faculty of Graduate Studies and Research
In Partial Fulfillment of the Requirements
For the degree of
Master of Applied Science
In Industrial Systems Engineering
University of Regina

By
Yong Zhang
Regina, Saskatchewan, Canada
July, 2007
Copyright 2007: Yong Zhang
NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Yong Zhang, candidate for the degree of Master of Applied Science in Industrial Systems Engineering, has presented a thesis titled, Visual Servoing of a 5-DOF Mobile Manipulator Using a Panoramic Vision System, in an oral examination held on June 22, 2007. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner: Dr. Samira Sadaoui-Mouhoub,  
Department of Computer Science

Supervisor: Dr. Mehran Mehrandezh, Faculty of Engineering

Committee Member: *Dr. Denise Stilling, Faculty of Engineering

Committee Member: Dr. René Mayorga, Faculty of Engineering

Chair of Defense: Professor Arthur Opseth, Faculty of Engineering

*Unable to attend defense
ABSTRACT

In this thesis, a novel visual servoing technique for a 5-degree of freedom (DOF) mobile manipulator with an eye-to-hand camera configuration is introduced. The proposed technique can be categorized as an image based (or 2D) visual servoing using a fixed camera in conjunction with a conic mirror (aka, an omni-directional camera system) providing panoramic vision. Two fictitious landmarks mounted on robot's end-effector along with their mirror reflections, when viewed by the camera, provide enough information for 3D reasoning based on the four points viewed on the image plane. Instead of directly using the image features associated with these four points, five new image features are chosen to make the image Jacobian of full rank. A dual estimation/control strategy based on Extended Kalman Filter (EKF) is utilized to (1) estimate the camera's intrinsic and extrinsic parameters, and (2) track the coordinates of the landmarks and their reflections on the image plane. The relationship between the translational and rotational velocity of a frame attached to the robot's end-effector and rate of change of the proposed image features are fully formulated. The robustness of the proposed visual servoing technique is illustrated through computer simulations and sensitivity tests.
ACKNOWLEDGEMENTS

First and foremost, I would like to thank Dr. Mehran Mehrandezh, who welcomed me to carry out this research under his supervision. His technical insights and especially his ability to provide guidance are greatly acknowledged.

I am thankful to the Faculty of Graduate Studies and Research and the Faculty of Engineering at the University of Regina for providing me with financial support in the form of a Graduate Scholarship and two Joint International Program Awards.
DEDICATION

My great gratitude goes to my family, Z. Zhang, X. Xie and H. Xiao for their love and support.
## TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>ii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iii</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>NOMENCLATURE</td>
<td>xii</td>
</tr>
<tr>
<td>ACRONYMS</td>
<td>xvi</td>
</tr>
<tr>
<td>1. Introduction and Literature Survey</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Problem Definition (overview)</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Background and Related Work</td>
<td>4</td>
</tr>
<tr>
<td>1.2.1 Camera Configurations and Number of Cameras</td>
<td>4</td>
</tr>
<tr>
<td>1.2.2 Visual Servoing Taxonomy</td>
<td>7</td>
</tr>
<tr>
<td>1.2.3 Servoing Control Scheme</td>
<td>12</td>
</tr>
<tr>
<td>1.2.4 Catadioptric Camera Visual Servoing</td>
<td>14</td>
</tr>
<tr>
<td>1.2.5 Kalman Filter in Visual Servoing</td>
<td>15</td>
</tr>
<tr>
<td>1.3 Contributions made by this Thesis</td>
<td>16</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.3.4 Implementation of a Kalman filter using three kinematic models .......................... 58
3.4 Summary .................................................................................................................... 59

4. Singularity in the Image Jacobian ................................................................................. 60
4.1 Singularity in Matrix D ............................................................................................... 60
4.2 Other Possible Singularities in the System ................................................................. 61

5. Simulation Results and Sensitivity Tests ..................................................................... 65
5.1 The Animation Robot ................................................................................................. 65
5.2 Simulation Results ....................................................................................................... 67
5.2.1 Simulation Setup ................................................................................................... 67
5.2.2 Simulation Results Based on the Second-order Kinematic Model ................... 69
5.2.3 Simulation Results Based on the Third-order Kinematic Model ...................... 79
5.2.4 Simulation Results Based on the Gaussian-Markov Kinematic Model ............ 88
5.2.5 Simulation result analysis .................................................................................... 97
5.3 Sensitivity Test Analysis ............................................................................................ 98
5.3.1 Sensitivity Test Results ........................................................................................ 98
5.3.2 Discussion on the Results of the Sensitivity Test ............................................. 105
5.4 Summary ..................................................................................................................... 105

6. Conclusions and Future Work ..................................................................................... 106

REFERENCES .................................................................................................................... 108
LIST OF TABLES

Table 5.1: Denavit-Hartenberg parameters of each joint. 66
Table 5.2: The settling time of each kinematic model (Tolerance 2%). 98
Table 5.3: Test result based on configuration I (d=5000) with the second order kinematic model. 99
Table 5.4: Test result based on configuration I (d=5000) with the third order kinematic model. 99
Table 5.5: Test result based on configuration I (d=5000) with the Gaussian Markov kinematic model. 100
Table 5.6: Test result based on configuration II (d=4000) with the second order kinematic model. 100
Table 5.7: Test result based on configuration II (d=4000) with the third order kinematic model. 101
Table 5.8: Test result based on configuration II (d=4000) with the Gaussian Markov kinematic model. 101
Table 5.9: Test result based on configuration III (d=3000) with the second order kinematic model. 102
Table 5.10: Test result based on configuration III (d=3000) with the third order kinematic model. 102
Table 5.11: Test result based on configuration III (d=3000) with the Gaussian Markov kinematic model. 103
Table 5.12: Statistical results based on configuration I (d=5000).

Table 5.13: Statistical results based on configuration II (d=4000).

Table 5.14: Statistical results based on configuration III (d=3000).
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The proposed system workspace.</td>
</tr>
<tr>
<td>1.2</td>
<td>Camera configurations. (a) Eye-in-hand camera configuration (b) Eye-to-hand camera configuration.</td>
</tr>
<tr>
<td>1.3</td>
<td>Modeling for 3D visual servoing.</td>
</tr>
<tr>
<td>1.4</td>
<td>Modeling for 2D visual servoing.</td>
</tr>
<tr>
<td>1.5</td>
<td>2.5D visual servoing diagram, [3].</td>
</tr>
<tr>
<td>1.6</td>
<td>General scheme of joint space control, [17].</td>
</tr>
<tr>
<td>1.7</td>
<td>General scheme of operational space control, [17].</td>
</tr>
<tr>
<td>2.1</td>
<td>(a) The configuration of the robot, camera and the mirror; (b) the projections of image features on a 2D image plane.</td>
</tr>
<tr>
<td>2.2</td>
<td>Perspective projection model.</td>
</tr>
<tr>
<td>2.3</td>
<td>Parameters to define the location of landmarks in the 3D space.</td>
</tr>
<tr>
<td>2.4</td>
<td>Relationship between landmark #1 and its reflection on the image plane.</td>
</tr>
<tr>
<td>3.1</td>
<td>The ongoing discrete Kalman filter cycle. The time update projects the current state estimate ahead in time. The measurement update adjusts the projected estimate by an actual measurement at that time.</td>
</tr>
<tr>
<td>4.1</td>
<td>Singularity in the Image Jacobian: landmark #1 and its mirror reflection both lie on the camera’s optical axis.</td>
</tr>
</tbody>
</table>
Figure 4.2: Singular configuration: landmark #1 and landmark #2's mirror reflection coincide in the image plane. 62
Figure 4.3: Singular configuration: landmark #2 and landmark #1's mirror reflection coincide in the image plane. 63
Figure 4.4: Singular configuration: landmark #1 and landmark #2 coincide in the image plane. 63
Figure 4.5: Singular configuration: landmark #1's mirror reflection and landmark #2's mirror reflection coincide in the image plane. 64
Figure 5.1: The animation robot. 66
Figure 5.2: Simulation Setup. 67
Figure 5.3: Simulation results based on the second-order kinematic model: (a) Trajectory of the landmarks and their mirror reflections on the 2D image plane. 70
Figure 5.3 (continued): (b) Image error. 71
Figure 5.3 (continued): (c) Trajectory of the landmarks in the 3D space. 72
Figure 5.3 (continued): (d) Position of the robot. 73
Figure 5.3 (continued): (e) Joint trajectory of the robot. 74
Figure 5.3 (continued): (f) Position error of landmark #1. 75
Figure 5.3 (continued): (g) Position error of landmark #2. 76
Figure 5.3 (continued): (h) Estimation error of camera parameters. 77
Figure 5.3 (continued): (i) Estimation of the coordinates corrupted by noise using a KF. 78
Figure 5.4: Simulation results based on the third-order kinematic model: (a) Trajectory of the landmarks on the 2D image plane. 79
Figure 5.4 (continued): (b) Image error ................................................................. 80
Figure 5.4 (continued): (c) Trajectory of the landmarks in the 3D space ............. 81
Figure 5.4 (continued): (d) Position of the robot .................................................. 82
Figure 5.4 (continued): (e) Joint trajectory of the robot ........................................ 83
Figure 5.4 (continued): (f) Position error of landmark #1 ..................................... 84
Figure 5.4 (continued): (g) Position error of landmark #2 ..................................... 85
Figure 5.4 (continued): (h) Estimation error of camera parameters ..................... 86
Figure 5.4 (continued): (i) Estimation of the coordinates corrupted by noise using a KF. .................................................................................................................................. 87

Figure 5.5: Simulation result based on the Gaussian-Markov kinematic model. (a)
Trajectory of the landmarks on the 2D image plane .......................................... 88
Figure 5.5: (b) Image error ...................................................................................... 89
Figure 5.5 (continued): (c) Trajectory of the landmarks in the 3D space ............. 90
Figure 5.5 (continued): (d) Position of the robot .................................................. 91
Figure 5.5 (continued): (e) Joint trajectory of the robot ........................................ 92
Figure 5.5 (continued): (f) Position error of landmark #1 ..................................... 93
Figure 5.5 (continued): (g) Position error of landmark #2 ..................................... 94
Figure 5.5 (continued): (h) Estimation error of camera parameters ..................... 95
Figure 5.5 (continued): (i) Estimation of the coordinates corrupted by noise using a KF. .................................................................................................................................. 96
**NOMENCLATURE**

\( \Phi \): the relative rotation matrix.

\( \Pi \): the interaction matrix between rate of change in the proposed velocity screw of the Mid-Point frame and the rate of change in the location of the landmark in Cartesian space.

\( \Gamma_i \): the interaction matrix related the locations of the object in front of the mirror and its mirror reflection.

\( D \): the interaction matrix between rate of change in the proposed image feature parameters and the rate of change in the location of the landmark on an image.

\( d \): the distance between the center of the camera and the center of the spherical mirror.

\( d_1 \): the distance from the middle point of the two lines connecting the landmark #1 and its mirror reflection to the desired location of the landmark #1 and its mirror reflection.

\( d_2 \): the distance from the middle point of the two lines connecting the landmark #2 and its mirror reflection to the desired location of the landmark #2 and its mirror reflection.

\( d_3 \): the distance from the one-third point of the two lines connecting landmark #1 and its mirror reflection to the desired location of landmark #1 and its mirror reflection.

\( F_i \): the interaction matrix between the rate of change of the velocity screw of the landmarks in the mirror frame and the camera frame.

\( F_k \): the process matrix.
\( f \): the camera's focal lens.

\( f(*) \): the process function.

\( H_k \): the observation matrix.

\( h(*) \): the measurement function.

\( K_k \): the Kalman filter gain.

\( k_x \): the image scaling factors on the x-axis.

\( k_y \): the image scaling factors on the y-axis.

\( l_1 \): the length of the connecting lines between landmark #1 and its mirror reflection.

\( l_2 \): the length of the connecting lines between landmark #2 and its mirror reflection.

\( P^m \): the arbitrarily chosen point represented in the mirror frame.

\( P^c \): the arbitrarily chosen point represented in the camera frame.

\( P^r \): the arbitrarily chosen point represented in the robot frame.

\( P_{rl}^m \): the landmark #1 represented in the mirror frame.

\( P_{m1}^m \): the landmark #1's mirror reflection in the mirror frame.

\( P_k \): the associated error covariance matrix.

\( Q \): the covariance matrix for the process noise.

\( J_p \): the Image Jacobian.

\( R \): the covariance matrix for the measurement noise.

\( ^cT_m \): the homogenous transformation matrix of the mirror's frame with respect to the camera's frame.
$\mathbf{T}_c$: the homogenous transformation matrix of the camera's frame with respect to the robot's frame.

$u$: the image plane's coordinate on the x-axis.

$u_c$: the x-axis coordinate of the principle point on an image.

$u_m$: the x-axis coordinate of the landmark's mirror reflection on an image.

$u_m^*$: the desired x-axis coordinate of the landmark's mirror reflection on an image.

$u_r$: the x-axis coordinate of the landmark on an image.

$u_r^*$: the desired x-axis coordinate of the landmark on an image.

$v$: the image plane's coordinate on the y-axis.

$v_c$: the y-axis coordinate of the principle point on an image.

$v_m$: the y-axis coordinate of the landmark's mirror reflection on an image.

$v_m^*$: the desired y-axis coordinate of the landmark's mirror reflection on an image.

$v_r$: the y-axis coordinate of the landmark on an image.

$v_r^*$: the desired y-axis coordinate of the landmark on an image.

$\mathbf{v}_k$: the measurement noise.

$\mathbf{w}_k$: the process noise.

$x$: the x-axis represented in the Cartesian coordinate system.

$x_k$: the state vector.

$y$: the y-axis represented in the Cartesian coordinate system.

$z$: the z-axis represented in the Cartesian coordinate system.

$z_k$: the measurement vector.
\( \lambda \): the proportional control gain.

\( \alpha_1 \): the x-axis coordinate of the middle point of the line connecting the desired position of the landmark #1 and its reflection on an image.

\( \beta_1 \): the x-axis coordinate of the middle point of the line connecting the desired position of the landmark #1 and its reflection on an image.

\( \alpha_2 \): the x-axis coordinate of the middle point of the line connecting the desired position of the landmark #2 and its reflection on an image.

\( \beta_2 \): the x-axis coordinate of the middle point of the line connecting the desired position of the landmark #2 and its reflection on an image.
ACRONYMS

DOF: Degree of Freedom
EKF: Extended Kalman Filter
IBVS: Image-Based Visual Servoing
KF: Kalman Filter
PBVS: Position-Based Visual Servoing
STD: Standard Deviation
2D: Two Dimensional
3D: Three Dimensional
1. Introduction and Literature Survey

Robots have become important over a wide range of applications – from manufacturing automation, to tele-surgery. Nevertheless, robots are not able to perform well in tasks requiring high accuracy, due to the inherent lack of sensory capability and fast and reliable communication links between the robot’s central control and the sensory units. In this context, visual servoing is proposed to resolve these problems [1]. Closed-loop visual positioning may allow objects to be presented in a non-structured manner to eliminate, or simplify, part feeders and the need for accurate location teaching. A further benefit is that the accuracy requirement of the robot can be relaxed while the high closed-loop performance is maintained.

A novel visual servoing structure that can position and orient a robot in 3D space is proposed. In this system, only one fixed camera is used in an eye-to-hand configuration to obtain a 2D image of the robot and its workspace. In general, extracting depth information from a 2D image is not practical, however this problem can be resolved by using a mirror in conjunction with the fixed camera. In our proposed system, a spherical mirror is used to provide a large field of view. A 2D image captured by a camera looking at the robot, and its reflected image in a nearby mirror can be used to extract depth information. An overview of the proposed approach is given in the following section.

1. The camera is fixed in the workspace.
1.1 Problem Definition (overview)

A new visual servoing structure, using a fixed camera and a spherical mirror, is presented for translational and rotational servoing of a robotic arm. The proposed simulation system is comprised of a 5-DOF anthropomorphic robotic manipulator whose base is mounted on a linear slide, a fixed camera, and a stationary spherical mirror (Figure 1.1).

![Diagram](image)

**Figure 1.1:** The proposed system workspace.

In the proposed system, the robot and its reflection in the mirror are assumed to always be available to the fixed camera and to remain within the camera's field of view (i.e., the
camera will not be occluded). Two landmarks are mounted on the wrist (i.e., the base of the end-effector) of an anthropomorphic robotic arm which is mounted on a linear slide (Figure 1.1). The image information obtained from the landmarks mounted on the robot, and their mirror reflections are used for 3D positioning and orienting of the robot's end-effector. The landmarks mounted on the robot are considered to be geometric points, and therefore, can not constitute a roll. Neglecting the rolling movement of the landmarks, a 5-DOF robot is sufficient to position and orient the robot’s end-effector in 3D.

Image-Based Visual Servoing (IBVS) is utilized in the proposed visual servoing structure, where the computation of the image Jacobian is paramount. The image Jacobian is computed from the image information and the intrinsic1 and extrinsic1 parameters of the camera. These parameters are described in more detail in Chapter 3. A dual estimation/control strategy, based on an Extended Kalman Filter (EKF) is utilized to: (1) estimate the camera’s intrinsic and extrinsic parameters; and (2) servo the robot’s end-effector to its target pose (i.e., pose here refers to a position and orientation captured in the form of a homogenous transformation matrix). The image Jacobian computed at each iteration is then linked with a simple proportional feedback control to achieve the desired position and orientation of a coordinate frame attached to the robot's end-effector. The robotic manipulator is assumed to be controlled by its velocity screw in Cartesian space. Therefore, the dynamics of the robot is not considered in the control loop.

The proposed technique was numerically simulated by deploying a model of a 5-DOF

---

1 Intrinsic parameters of camera include physical property of the camera and its image grabbing mechanism, such as the camera’s lens focus length.

2 Extrinsic parameters of a camera are defined by a homogenous transformation between the camera frame and world frame.
robotic manipulator whose base was mounted on a linear slide that attempted to place its end-effector in a user-defined location/orientation in a 3D workspace. Error analysis and sensitivity tests were carried out to evaluate the robustness of the proposed system.

1.2 Background and Related Work

For the past two decades, a variety of approaches have been developed to increase the overall accuracy/reliability of robot visual servoing systems (e.g., [1, 2]). A number of pertinent articles are reviewed in this section, which include discussions of camera configurations and number of cameras, visual servoing taxonomy, control schemes, catadioptric camera visual servoing, and applications of the Kalman filter in visual servoing. In this section, a consistent terminology is used in discussing the literature, despite the different nomenclature used by various authors.

1.2.1 Camera Configurations and Number of Cameras

Visual servoing systems typically use one of two camera configurations: (1) Eye-in-hand and (2) Eye-to-hand configurations. The details are described below.
Eye-in-hand Camera Configuration

In the Eye-in-hand camera configuration, the camera is mounted on the robot end-effector as shown in Figure 1.2(a). The Eye-in-hand system is common for providing camera motions that increases the working region of a visual sensor. Nevertheless, the system has several deficiencies. Typically, the system fails if manipulators pass through singularities or joint limits in the tracking regions. Objects being tracked can be lost if the objects become defocused, occluded, or if features on the objects lie outside the field of view of the camera. Therefore, the eye-in-hand system is usually only able to deal with these situations by severely constraining object motion [3, 4].
Eye-to-hand Camera Configuration

In the eye-to-hand configuration, the camera is fixed in the workspace and is continuously focusing on the robot's end-effector, as shown in Figure 1.2(b). In this case, the camera image of the target is independent of the robot’s motion. Therefore, the eye-to-hand system has an advantage since a fancy camera with an auto focus feature is not necessary, in contrast to the eye-in-hand system. One can easily control the robot’s end-effector by keeping it in the field of view of the camera during the visual servoing in situations where the robot’s workspace is limited. In our proposed system, we use the eye-to-hand configuration [5, 6].

Hybrid Camera Configurations

More recently, hybrid eye-to-hand/eye-in-hand visual servoing systems have been proposed for 3D translational/rotational control of robotic manipulators (e.g., [7]). Elena, et al. subdivided tasks (tracking an object and knowing the position of the robot arm) and performed each of them with a single camera. The eye-in-hand camera can execute the tracking of the target and the fixed camera can determine the position of the robot arm in 3D space [8].

Mono and Stereo Vision

Monocular vision (i.e., using one single image captured by one camera) is mostly
deployed in 2D visual servoing, where information associated with depth cannot be determined (e.g., [9]). Alternatively, stereo vision [10] is an interpretation of two views of the scene taken from known different viewpoints to resolve depth ambiguity. The location of feature points in one view must be matched with the location of the same feature points in the other view. This matching, or correspondence problem, is not trivial and is subject to error. Another difficulty is in the missing parts problem, where a feature point is visible in only one of the views, which prevents the determination of depth. Matching may be done with a few feature points, such as region centroids, or with corner features, or with fine feature details such as surface texture. In the absence of significant surface texture, a random texture pattern can be projected onto the scene. Another problem with stereo vision is in synchronizing the cameras, since their image acquisition and processing units need to be in real time.

1.2.2 Visual Servoing Taxonomy

Visual servoing systems can be classified into two groups: Position-Based (or 3D) and Image Based (or 2D) visual servoing systems [1]. Both of these visual servoing techniques have their own advantages and disadvantages. A comprehensive comparison between these two systems can be found in [11] and [12]. Recently, a new algorithm (called 2.5D visual servoing) was introduced by Mails and Chaumette [3]. The difference between the systems is a function of the type of controlled parameters in the feedback control loop (i.e., presented either in the image or in the Cartesian domain).
Position-based Visual Servoing (PBVS)

In position-based visual servoing system with the eye-to-hand configuration, let $F^c$ be the coordinate frame attached to the camera; $F^*$ and $F$ be the coordinate frames attached to a pre-specified point mounted on the robot’s end-effector in robot’s desired and current pose, respectively (Figure 1.3). The matrix $^oT_0$ defines the homogeneous transformation matrix from the camera frame to the robot’s end-effector’s current frame. The matrix $^dT_d$ defines the homogeneous transformation matrix from the camera frame to the robot’s end-effector’s desired frame. These two matrices have to be known to calculate the robot’s desired and current pose. The error signal fed to the robot’s position controller is the difference between the desired pose and the current pose of the robot with respect to the camera’s coordinate frame.
Figure 1.3: Modeling for 3D visual servoing.

The main advantage of this approach is that it directly controls the robot's end-effector's trajectory in Cartesian space normally yielding smooth movements. Since no direct control of the image is possible, the image features used in the pose estimation may leave the camera's field of view, which can lead to servoing failure. Also, if the camera is coarse calibrated, or if errors exist in the 3D model of the target, the current and desired target poses will not match perfectly. Finally, since the error in making the pose estimation cannot be computed analytically as a function of the camera calibration errors, the stability of the system cannot be analyzed [10].
Image-based Visual Servoing (IBVS)

In this case, the control error function is expressed directly in the 2D image plane (Figure 1.4). Let \( s \) be a vector representing the current value of visual features observed by the camera and \( s^* \) be the desired value of the vector \( s \) to be reached in the image. The time variation of \( s \) can be related to the manipulator velocity \( v = [u \ o]^T \) according to [1]:

\[
\dot{s} = J_r(v)v
\]  

(1.1)

where \( J_r \) is called the interaction or image Jacobian matrix. The dimension of \( J_r \) is \( k \times m \), where \( k \) is the dimension of the image feature vector, \( s \), and \( m \) is the dimension of the end-effector velocity screw which is normally six. In visual servoing, \( v \), the velocity screw of the end-effector, is manipulated to control the robot and to achieve a desired value in \( s \).

An estimation of the depth can be obtained, as in 3D visual servoing, by using a pose determination algorithm (if a 3D target model is available), or a structure from a known motion algorithm (if the target motion can be measured). In general, this method is known to be robust not only with respect to the camera, but also to the robot calibration errors. Nevertheless, its convergence is theoretically ensured only in a region around the desired state. Furthermore, singularity in the image Jacobian can affect the convergence of the error ([9, 13]). Except in very simple cases, the analysis of the stability, with respect to calibration errors, seems to be impossible, since the system's equations are coupled and non-linear.
2.5 D Visual Servoing

Recently, an algorithm called 2.5D visual servoing was introduced by Mails and Chaumette (e.g., [3, 14]) which is targeted for eye-in-hand configurations. The principle diagram of a 2.5D visual servoing structure is depicted in Figure 1.5. In 2.5D visual servoing, information from the 3D task-space (obtained either through a given 3D model or, more interestingly, through a projective Euclidean reconstruction) is utilized to regulate the rotation error system while information from the 2D image-space is utilized to control the translation error system. Corke and Hutchinson concurrently developed a new hybrid image-based visual servoing method that decouples rotational and translational components of the camera's motion about the z-axis, which is aligned with
the camera’s focus from the remaining degrees of freedom [15]. Initially, 2.5D visual servoing was used in the eye-in-hand configuration, though some attempts have been made to generalize it to the eye-to-hand configuration, as well (e.g., [6]).

![Figure 1.5: 2.5D visual servoing diagram, [3].](image)

1.2.3 Servoing Control Scheme

Two different types of control schemes are generally used to manipulate a robot’s movement: (1) Joint space control scheme and (2) Cartesian space or operational space control scheme. The main objective is to develop a control law that robustly servos the robot, being subjected to unmodeled errors and uncertainties, by reducing the effect of disturbances imposed on the system. The principle diagrams of these two servoing control scheme are depicted in Figure 1.6 and 1.7.

In the joint space control scheme (e.g., [2, 16]), the inverse kinematics of the manipulator is first solved to transform the control set-point input from Cartesian space (i.e., the
desired pose of the robot’s tip in the task space) into that in the robot’s joint space. Then, the joint-space feedback control process tracks the reference inputs. Only joint control variables are in the feedback control loop and do not influence the Cartesian space variables. This control scheme is suitable for motion control in an obstacle-free workspace.

In an operational space control scheme (e.g., [17, 18]), the motion specifications are normally assigned in the operational space and then transformed to joint space by applying inverse kinematics. Besides the inversion of direct kinematics, first-order and second-order differential kinematics are also required to transform the desired robot’s end-effector position, velocity, and acceleration into their corresponding joint level values. The inverse kinematics function is represented in the feedback control loop where the joint trajectory inversion can be replaced with a suitable coordinate transformation. This control scheme is convenient when the manipulator’s end-effector is constrained with obstacles. In our proposal system, we use the operational space control scheme.
1.2.4 Catadioptric Camera Visual Servoing

In the literature, several methods have been proposed for increasing the field of view of the camera. Through so-called omnidirectional visions, one effective way is to combine mirrors with the conventional imaging systems. The obtained sensors are referred to as catadioptric imaging systems. A catadioptric camera system is either a combination of a parabolic mirror with an orthographic camera, or a hyperbolic, elliptical or spherical mirror with a perspective camera. Clearly, visual servoing applications can also benefit from such sensors since they naturally overcome the visibility constraint.

In [5], Kulpate, et al. combined a flat mirror with a perspective camera to position a 3-DOF manipulator in the 3D space and estimated the components of image Jacobian online. Vision-based control of robotic arms, in a single mobile robot or in the formation of mobile robots, has been described in the literature having omidirectional cameras (e.g. [19]). Recently, Barreto, et al. developed a visual servoing control law utilizing a catadioptric vision system, which benefits from an enhanced camera field of view for eye to control the position of a robotic arm [4]. This approach increases the robustness and accuracy of the visual servoing, but, because of complexity of the camera project model,
its success depends highly on the calibration.

More recently, Mezouar, et al. developed a visual servoing control law from 3D straight lines with central catadioptric cameras [21]. Their results showed that the visual servoing schemes can benefit from the wide field of view of catadioptric cameras to overcome the visibility constraints. Abdelkader, et al. also implemented central catadioptric cameras in 2.5D visual servoing [19]. In both approaches the camera only sees the projection of the real objects in the operational space on the mirror. In our proposed system, we have built an omnidirectional camera by utilizing a spherical mirror and a perspective camera. The perspective camera not only continuously looks at landmark reflections in the spherical mirror, but it also continuously looks at landmarks in the operational space. This double gaze facilitates drawing depth information on landmarks using only one single camera.

1.2.5 Kalman Filter in Visual Servoing

The Kalman filter is widely used in visual servoing due to presence of noise in image data (e.g. [18]). In [18], Qian and Su used a Kalman filter to estimate the elements of the image Jacobian. In [21], Wang and Wilson used a Kalman filter for estimating 3D motion parameters for 3D tracking control. In [22], Hideaki, et al. used a Kalman filter to estimate the position and velocity of the feature points in the camera frame. In the proposed system, a dual Kalman filter is employed, with one part estimating the intrinsic and extrinsic parameters of the camera, and the other part tracking the coordinates of the landmarks and their mirror reflections on the image plane.
1.3 Contributions made by this Thesis

This thesis introduces a novel, yet simple approach, for the 3D translational and rotational control of a robotic arm mounted on a linear slide with five degrees of freedom in total, using vision feedback in an eye-to-hand scenario with a fixed camera and a stationary spherical mirror. The image information obtained by viewing the landmarks mounted on the robot and their mirror reflections are utilized for 3D servoing of the pose of the robot’s end-effector. Furthermore, a methodology for estimating the intrinsic and extrinsic parameters of the camera is developed. A tracking filter is implemented to track the coordinates of the landmarks and their mirror reflections on the image plane. Simulation results, from error analysis and sensitivity tests, have been carried out to evaluate the robustness of the system.

1.3.1 Theoretical Framework

- A novel visual servoing structure is presented for 3D positioning and orienting of a robotic arm, using one fixed camera and a spherical mirror, based on image-based visual servoing. The multiple views of the robot on a 2D image, captured by one fixed camera, were utilized to precisely position and orient the robot in a 3D workspace.

- A set of novel image feature parameters has been introduced and deployed in the control scheme, which makes the system less sensitive to noise.

- Visual servoing with on-line camera calibration is introduced to position and
orient the robot manipulator. A Kalman Filter (KF) is utilized to improve the noise sensitivity of the system.

- The singularity in the image Jacobian is also fully discussed in this work.
- Simulation and sensitivity tests based on different configurations of the robot, the mirror and the camera and different initial guesses of the camera’s parameters have been utilized to show the stability and robustness of the proposed visual servoing structures.

1.3.2 Planning Algorithm

The crux of this thesis is to arbitrarily position and orient the end-effector of a 5-DOF robotic manipulator whose base is mounted on a linear slide in a 3D workspace using only one fixed camera and a spherical mirror. The planning steps are briefly discussed here. Firstly, the initial and the desired image parameters are set. Next, the intrinsic and extrinsic parameters of the camera were estimated and used in a resolved-rate servoing law, where the velocity screw of the robot’s end-effector constitutes the control input. The robot is then moved to a next configuration based upon this control input recursively. This procedure continues until the difference between the desired and the current image feature parameters is less than a pre-specified threshold value. Through computer simulations, the proposed technique was shown to precisely position and orient a robotic arm in 3D space.

17

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
1.3.3 Simulation Software

The simulation environment was developed in MATLAB (version 7.0) using the robotics toolbox developed by Corke [23, 24]. Simulation results will be discussed in more detail in Chapter 5.

1.4 Thesis Outline

The Visual servoing structure and problem formulation are discussed in Chapter 2. In Chapter 3, a dual Kalman Filter (1) to identify camera’s parameters, and (2) to estimate the coordinates of the landmarks and their mirror reflections on the image plane is developed for the proposed system. The singularities in the visual servoing system are presented in Chapter 4. Simulations and sensitivity tests are addressed in Chapter 5. Chapter 6 presents the conclusions and future work.
2. System Modeling and Problem Formulation

The simulated system consists of a 5-DOF anthropomorphic robotic manipulator whose base is mounted on a linear slide, with one fixed camera and a stationary sphere mirror, as depicted in Figure 2.1(a). Two fictitious landmarks are assumed to be mounted on the link, which constitute the robot's end-effector. The main objective of this chapter is to find the relationship between the rate of change of a quintuplet image feature and a quintuplet velocity screw of a coordinate frame attached to the robot's end-effector with its origin at the mid-point between two landmarks mounted on the end-effector. We name this coordinate frame Mid-Point frame in the rest of this thesis. The steps to achieve this objective are given as below:

Step 1: Define coordinate transformations of the mirror, camera and robot;

Step 2: Present the camera model used in our proposed visual servoing system;

Step 3: Define a quintuplet image feature;

Step 4: Define a quintuplet velocity screw of the Mid-Point frame;

Step 5: Calculate the image Jacobian which describes the relationship between the rate of change of the quintuplet image feature and the quintuplet velocity screw of the Mid-Point frame.
(u_m^*, v_m^*) denotes the target position of the landmark’s mirror reflection, and (u_r^*, v_r^*) denotes the target position of the landmark.

Figure 2.1: (a) The configuration of the robot, camera and the mirror; (b) the projections of image features on a 2D image plane.
2.1 Step 1: Define Coordinate Transformations of the Mirror, Camera, and Robot

Coordinate frames that were assigned to the robot, the camera, and the mirror are shown in Figure 2.1(a). The mirror frame can be mapped to the camera frame by a Homogenous Transformation Matrix $^cT_m$, and the camera frame can also be mapped to the robot frame by $^rT_c$:

\[ \tilde{P}_c = ^cT_m \tilde{P}_m \] (2.1a)

\[ \tilde{P}_r = ^rT_c \tilde{P}_c \] (2.1b)

where $\tilde{p}_m$, $\tilde{p}_c$, and $\tilde{p}_r$ denote the augmented coordinates of an arbitrarily chosen point where in general $\tilde{P} = [P \ 1]^T$ represented in the mirror frame, camera frame, and robot frame, respectively.

The camera is positioned at (0, 0, d), with respect to the mirror frame. Therefore, one can use a constant matrix $^cT_m = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix}$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Based on the spherical convex mirror reflection rule [25], the relationship between the landmarks $P_{ri}^r, P_{r2}^r$ and their reflections $P_{mi}^m, P_{m2}^m$, represented in the sphere mirror frame, is as follows:

$$P_{mi}^m = \Gamma_i P_{ri}^r \quad i = 1,2$$  \hspace{1cm} (2.2)

where

$$\Gamma_i = \frac{r}{2\|P_{ri}^r\| - r} \quad i = 1,2$$  \hspace{1cm} (2.3)

where $r$ is the radius of the sphere mirror.

Using Equation (2.2), one can transform the position of the landmarks and their mirror reflections in the mirror frame, to those in the camera frame by:

$$P_{ci}^c = \mathbf{C}_m P_{mi}^m$$  \hspace{1cm} (2.4a)

$$P_{mci}^c = \mathbf{C}_m P_{mci}^m$$  \hspace{1cm} (2.4b)

From Equations (2.1-2.4) one can relate the coordinates of a landmark to its mirror reflection, represented in the camera frame, as follows:

$$x_{mi}^c = \frac{r}{2\|P_{ri}^r\| - r} x_{ri}^c \quad i = 1,2$$  \hspace{1cm} (2.5a)

\textsuperscript{IV} In this thesis, the superscript “c” refers to the camera frame, the superscript “m” refers to the mirror frame, and the superscript “r” refers to the robot frame. The subscript “r” refers to a real landmark and the subscript “m” refers to the landmark’s mirror reflection; the subscript “1” refers to landmark #1, “2” refers to landmark #2. For example, a point $P_{ri}^m$ represents the position of the landmark #1 represented in the mirror frame.

\textsuperscript{V} $\|\bullet\|$ represents second norm of a vector or a matrix in this thesis.
\[ y_{mi}^c = \frac{r}{2\|p_n\|^2} y_{ri}^c \quad i = 1,2 \quad (2.5b) \]

\[ z_{mi}^c = \frac{r}{2\|p_n\|^2} (z_{ri}^c + d) - d \quad i = 1,2 \quad (2.5c) \]

The time derivative of Equations (2.5a-2.5c) can be written in a matrix form:

\[
\begin{bmatrix}
\dot{x}_{mi}^c \\
\dot{y}_{mi}^c \\
\dot{z}_{mi}^c \\
\end{bmatrix} = F_i \begin{bmatrix}
\dot{x}_{ri}^c \\
\dot{y}_{ri}^c \\
\dot{z}_{ri}^c \\
\end{bmatrix}
\quad i = 1,2 \quad (2.6)
\]

Where

\[
F_i = \frac{r}{2} \begin{bmatrix}
1 & -\frac{y_{ri}^c \cdot x_{ri}^c}{\|p_n\|^3} & -\frac{(z_{ri}^c + d) \cdot x_{ri}^c}{\|p_n\|^3} \\
-\frac{x_{ri}^c \cdot y_{ri}^c}{\|p_n\|^3} & 1 & -\frac{(z_{ri}^c + d) \cdot y_{ri}^c}{\|p_n\|^3} \\
-(z_{ri}^c + d) \cdot x_{ri}^c & -(z_{ri}^c + d) \cdot y_{ri}^c & 1
\end{bmatrix}
\]

\[ , \quad i = 1,2 \]

By using Equation (2.6), one can transform the velocity screw of the landmarks to their mirror reflections, as presented in the camera’s frame.

### 2.2 Step 2: Camera Model

The three possible projection models for the image formation processes are **perspective**, **scaled orthographic**, and **affine projections**. The most commonly used projection model, namely the perspective projection model, was chosen for this work. The perspective projection is schematically shown in Figure 2.2. The relationship between the coordinates
of a point $\mathbf{p}_i = [x_i \ y_i \ z_i]$, in 3D space, and its augmented projection on the 2D image plane, $\mathbf{m}_i = [u_i \ v_i \ 1]$, can be given as:

$$\mathbf{p}_i = \frac{\mathbf{p}_i}{z_i}$$

$$\mathbf{m}_i = A\mathbf{p}_i$$

(2.7a)

(2.7b)

where $\mathbf{p}_i = [x_i \ y_i \ z_i]^T$ and $A = \begin{bmatrix} \alpha_u & 0 & u_c \\ 0 & \alpha_v & v_c \\ 0 & 0 & 1 \end{bmatrix}$, $\alpha_u = f\alpha_u$, $\alpha_v = f\alpha_v$.

The parameter $f$ is the length of the camera’s focal lens; parameters $k_u$ and $k_v$ are scaling factors, corresponding to the effective size of the pixels in the horizontal and vertical directions; and $u_c$ and $v_c$ are offset factors, which are the principle point of the image plane represented in the pixel frame, usually at, or near, the center of the image.

Equation (2.7) is derived with the assumption that the camera’s lens imposes no image twist.
2.3 Step 3: Define a quintuplet image feature

The projection of the landmarks mounted on the robot and their mirror reflections on a 2D image plane is formulated in this section. As shown in Figure 2.1(b), \( u_{rl} \) and \( v_{rl} \) represent the coordinates of the landmark #1 projected onto the image plane, and \( u_{m1} \) and \( v_{m1} \) denote the coordinates of the mirror reflection of landmark #1 on the image plane. The same definition applies to landmark #2. By bringing \([u_{rl}, v_{rl}], [u_{r2}, v_{r2}], [u_{m1}, v_{m1}], \) and \([u_{m2}, v_{m2}]\) simultaneously to their desired values, namely \([u_{rl}^*, v_{rl}^*], [u_{r2}^*, v_{r2}^*], [u_{m1}^*, v_{m1}^*], \) and \([u_{m2}^*, v_{m2}^*]\), one can position and orient the robot in 3D.

The landmarks mounted on the robot are considered to be geometric points, and therefore, cannot constitute a roll. Neglecting the roll of the landmarks, a five DOF robot is

Figure 2.2: Perspective projection model.
sufficient to position and orient the robot’s end-effector in 3D. For this reason, the eight image features, described earlier, can be reduced to five. A number of possible morphologies can achieve this, however, one morphology was identified that would require minimum effort to distinguish real landmarks from their mirror reflections. This particular morphology is now examined in more detail.

The chosen quintuplet image features are: $l_1$, $l_2$, $d_1$, $d_2$, and $d_3$, as shown in Figure (2.1b), where $l_1$ and $l_2$ are the lengths of the connecting lines between a landmark and its mirror reflection. The features, $d_1$ and $d_2$ are the distances from the middle point of the two lines connecting the landmarks and their mirror reflections to the desired location of the landmarks and their mirror reflections. The feature $d_3$ is the distance from the one-third point of the two lines connecting landmark #1 and its mirror reflection to the desired location of landmark #1 and its mirror reflection. These features can be calculated in terms of the image points $[u_{rl}, v_{rl}], [u_{r2}, v_{r2}], [u_{m1}, v_{m1}]$, and $[u_{m2}, v_{m2}]$ as:

\[
l_1(u_{m1}, u_{rl}, v_{m1}, v_{rl}) = \sqrt{(u_{m1} - u_{rl})^2 + (v_{m1} - v_{rl})^2}
\]

\[
l_2(u_{m2}, u_{r2}, v_{m2}, v_{r2}) = \sqrt{(u_{m2} - u_{r2})^2 + (v_{m2} - v_{r2})^2}
\]

\[
d_1(u_{m1}, u_{rl}, v_{m1}, v_{rl}) = \sqrt{\left(\frac{u_{m1} + u_{rl}}{2} - \alpha_1\right)^2 + \left(\frac{v_{m1} + v_{rl}}{2} - \beta_1\right)^2}
\]

\[
d_2(u_{m2}, u_{r2}, v_{m2}, v_{r2}) = \sqrt{\left(\frac{u_{m2} + u_{r2}}{2} - \alpha_2\right)^2 + \left(\frac{v_{m2} + v_{r2}}{2} - \beta_2\right)^2}
\]
where \((\alpha_1, \beta_1)\) and \((\alpha_2, \beta_2)\) represent the 2D image coordinates of the center of the line connecting the desired position of the landmarks and their reflections for landmarks \#1 and \#2, respectively, projected onto the 2D image plane. The coordinates \((\alpha, \beta)\) represent the 2D image coordinates of the one-third point that connects the desired position of landmark \#1 and its reflection, projected onto the 2D image plane. All the values of \(\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha, \beta\) are constant. From Equation (2.8), \(l_1\) and \(d_1\) are seen to be reciprocal in terms of image feature pairs, namely \((u_{r1}, u_{m1})\) and \((v_{r1}, v_{m1})\).

Mathematically, one can easily show that: \(l_1(u_{m1}, u_{r1}, v_{m1}, v_{r1}) = l_1(u_{r1}, u_{m1}, v_{r1}, v_{m1})\), and \(d_1(u_{m1}, u_{r1}, v_{m1}, v_{r1}) = d_1(u_{r1}, u_{m1}, v_{r1}, v_{m1})\). This rule applies to \(l_2(u_{m2}, u_{r2}, v_{m2}, v_{r2})\) and \(d_2(u_{m2}, u_{r2}, v_{m2}, v_{r2})\), as well. Therefore, the value of \(l_1, l_2, d_1, \) and \(d_2\) will be invariant, in terms of image information obtained on the real landmark or its reflection, but the value of \(d_3\) can vary, in terms of image information obtained on the real landmark and its reflection. Nevertheless, the landmarks and their mirror reflections on the image plane can still be distinguished, which will be discussed in the next section. In light of Equation (2.8), the velocity screws can be calculated as follows:

\[
\dot{l}_1 = \frac{1}{l_1} \left[ (u_{m1} - u_{r1}) \dot{u}_{m1} - (u_{m1} - u_{r1}) \dot{u}_{r1} + (v_{m1} - v_{r1}) \dot{v}_{m1} - (v_{m1} - v_{r1}) \dot{v}_{r1} \right] \tag{2.9a}
\]

\[
\dot{l}_2 = \frac{1}{l_2} \left[ (u_{m2} - u_{r2}) \dot{u}_{m2} - (u_{m2} - u_{r2}) \dot{u}_{r2} + (v_{m2} - v_{r2}) \dot{v}_{m2} - (v_{m2} - v_{r2}) \dot{v}_{r2} \right] \tag{2.9b}
\]
\[ \dot{\mathbf{q}} = \frac{1}{2d_i} \begin{bmatrix} \left( \frac{u_{m1}+u_{r1}}{2} \right) - \alpha_i \end{bmatrix} \dot{u}_{m1} + \left[ \left( \frac{u_{m2}+u_{r1}}{2} \right) - \alpha_i \right] \dot{u}_{r1} + \left[ \left( \frac{v_{m1}+v_{r2}}{2} \right) - \beta \right] \dot{v}_{m1} + \left[ \left( \frac{v_{m2}+v_{r1}}{2} \right) - \beta \right] \dot{v}_{r1} \] (2.9c)

\[ \dot{\mathbf{q}} = \frac{1}{2d_2} \begin{bmatrix} \left( \frac{u_{m1}+u_{r2}}{2} \right) - \alpha_2 \end{bmatrix} \dot{u}_{m2} + \left[ \left( \frac{u_{m2}+u_{r2}}{2} \right) - \alpha_2 \right] \dot{u}_{r2} + \left[ \left( \frac{v_{m1}+v_{r2}}{2} \right) - \beta \right] \dot{v}_{m2} + \left[ \left( \frac{v_{m2}+v_{r2}}{2} \right) - \beta \right] \dot{v}_{r2} \] (2.9d)

\[ \dot{\mathbf{q}} = \frac{1}{3d_2} \begin{bmatrix} \left( \frac{u_{m1}+2u_{r1}}{3} \right) - \alpha \end{bmatrix} \dot{u}_{m1} + \left[ \left( \frac{u_{m2}+2u_{r1}}{3} \right) - \alpha \right] \dot{u}_{r1} + \left[ \left( \frac{v_{m1}+2v_{r1}}{3} \right) - \beta \right] \dot{v}_{m1} + \left[ \left( \frac{v_{m2}+2v_{r1}}{3} \right) - \beta \right] \dot{v}_{r1} \] (2.9e)

The above equations can be re-written in matrix form as:

\[ \begin{bmatrix} \dot{u}_{m1} & \dot{v}_{m1} & \dot{u}_{r1} & \dot{v}_{r1} & \dot{u}_{m2} & \dot{v}_{m2} & \dot{u}_{r2} & \dot{v}_{r2} \end{bmatrix}^T = D \begin{bmatrix} \mathbf{u}_m & \mathbf{v}_m & \mathbf{u}_r & \mathbf{v}_r \end{bmatrix}^T \] (2.10)

Where,

\[ D = \begin{bmatrix} D_1_{3x4} & 0_{3x4} \\ 0_{2x4} & D_2_{2x4} \end{bmatrix} \] (2.11)

\[ D_1 = \begin{bmatrix} \begin{bmatrix} \frac{u_{m1}-u_{r1}}{l_1} \end{bmatrix} & \begin{bmatrix} \frac{v_{m1}-v_{r1}}{l_1} \end{bmatrix} & \begin{bmatrix} \frac{u_{m2}-u_{r1}}{l_1} \end{bmatrix} & \begin{bmatrix} \frac{v_{m2}-v_{r1}}{l_1} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \frac{u_{m1}+u_{r1}}{2} - \alpha \end{bmatrix} & \begin{bmatrix} \frac{v_{m1}+v_{r1}}{2} - \beta \end{bmatrix} & \begin{bmatrix} \frac{u_{m2}+u_{r1}}{2} - \alpha \end{bmatrix} & \begin{bmatrix} \frac{v_{m2}+v_{r1}}{2} - \beta \end{bmatrix} \end{bmatrix} \end{bmatrix} \]

\[ D_2 = \begin{bmatrix} \begin{bmatrix} \frac{u_{m1}-u_{r2}}{l_2} \end{bmatrix} & \begin{bmatrix} \frac{v_{m1}-v_{r2}}{l_2} \end{bmatrix} & \begin{bmatrix} \frac{u_{m2}-u_{r2}}{l_2} \end{bmatrix} & \begin{bmatrix} \frac{v_{m2}-v_{r2}}{l_2} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \frac{u_{m1}+u_{r2}}{2} - \alpha_2 \end{bmatrix} & \begin{bmatrix} \frac{v_{m1}+v_{r2}}{2} - \beta_2 \end{bmatrix} & \begin{bmatrix} \frac{u_{m2}+u_{r2}}{2} - \alpha_2 \end{bmatrix} & \begin{bmatrix} \frac{v_{m2}+v_{r2}}{2} - \beta_2 \end{bmatrix} \end{bmatrix} \end{bmatrix} \]
2.4 Step 4: Define a quintuplet velocity screw of the Mid-Point frame

To reduce the size of the velocity vector associated with the two landmarks to a quintuplet velocity screw of the Mid-Point frame which is a coordinate frame attached to the mid-point of a line segment connecting the landmarks, the spatial motion of the landmarks can be uniquely represented using only five parameters, as depicted in Figure (2.3), namely $x_r', y_r', z_r', \phi$, and $\theta$.

Figure 2.3: Parameters to define the location of landmarks in the 3D space.

The point $O_1(x'_r, y'_r, z'_r)$ is the middle point of the line segment connecting $(x'_{r1}, y'_{r1}, z'_{r1})$ and $(x'_{r2}, y'_{r2}, z'_{r2})$. The angle $\phi$ is the angle between the x axis and the projection of the line segment, defined by $(x'_{r1}, y'_{r1}, z'_{r1})$ and $(x'_{r2}, y'_{r2}, z'_{r2})$ on the x-y plane. The angle $\theta$ is
the angle between the line segment defined by \((x_{r1}, y_{r1}, z_{r1})\) and \((x_{r2}, y_{r2}, z_{r2})\) and the x-y plane. The distance between points \((x_{r1}, y_{r1}, z_{r1})\) and \((x_{r2}, y_{r2}, z_{r2})\) is a constant value, denoted by \(L\). With this assumption one can write:

\[
x'_{r1} = x_r - \frac{L}{2} \cos \theta \cos \phi, \quad x'_{r2} = x_r + \frac{L}{2} \cos \theta \cos \phi
\]  
(2.12a)

\[
y'_{r1} = y_r - \frac{L}{2} \cos \theta \sin \phi, \quad y'_{r2} = y_r + \frac{L}{2} \cos \theta \sin \phi
\]  
(2.12b)

\[
z'_{r1} = z_r - \frac{L}{2} \sin \theta, \quad z'_{r2} = z_r + \frac{L}{2} \sin \theta
\]  
(2.12c)

Therefore, one can write:

\[
\begin{bmatrix}
\dot{x}'_{r1} & \dot{y}'_{r1} & \dot{z}'_{r1} & \dot{x}'_{r2} & \dot{y}'_{r2} & \dot{z}'_{r2} & \phi & \theta
\end{bmatrix}^T = \Pi \begin{bmatrix}
\dot{x}_r & \dot{y}_r & \dot{z}_r & \phi & \theta
\end{bmatrix}^T
\]  
(2.13)

Where \(\Pi =\)

\[
\begin{bmatrix}
1 & 0 & 0 & \frac{L}{2} \cos \theta \sin \phi & \frac{L}{2} \sin \theta \cos \phi \\
0 & 1 & 0 & -\frac{L}{2} \cos \theta \cos \phi & \frac{L}{2} \sin \theta \sin \phi \\
0 & 0 & 1 & 0 & -\frac{L}{2} \cos \theta \\
1 & 0 & 0 & -\frac{L}{2} \cos \theta \sin \phi & -\frac{L}{2} \sin \theta \cos \phi \\
0 & 1 & 0 & \frac{L}{2} \cos \theta \cos \phi & -\frac{L}{2} \sin \theta \sin \phi \\
0 & 0 & 1 & 0 & \frac{L}{2} \cos \theta
\end{bmatrix}
\]  
(2.14)
2.5 Step 5: Calculate the image Jacobian

Image Jacobian $J_v$ can be defined to determine the relationship between the rate of change of the quintuplet image feature $[\dot{l}_1 \quad \dot{l}_2 \quad \dot{l}_3 \quad \dot{l}_4 \quad \dot{l}_5]^T$ and the velocity screw of the Mid-Point frame $[\dot{x}_r \quad \dot{y}_r \quad \dot{z}_r \quad \phi \quad \dot{\theta}]^T$ as:

$$[\dot{l}_1 \quad \dot{l}_2 \quad \dot{l}_3 \quad \dot{l}_4 \quad \dot{l}_5]^T = J_v [\dot{x}_r \quad \dot{y}_r \quad \dot{z}_r \quad \phi \quad \dot{\theta}]^T$$  \hspace{1cm} (2.15)

The components of $J_v$ are now described in detail.

Based on the perspective projection model, one can write:

$$\begin{align*}
\dot{u}_1 &= \frac{f_x \dot{x}_r}{z_{r1}} - \frac{u_{r1} \dot{z}_r}{z_{r1}}, \\
\dot{v}_1 &= \frac{f_y \dot{y}_r}{z_{r1}} - \frac{v_{r1} \dot{z}_r}{z_{r1}}, \\
\dot{u}_2 &= \frac{f_x \dot{x}_r}{z_{r2}} - \frac{u_{r2} \dot{z}_r}{z_{r2}}, \\
\dot{v}_2 &= \frac{f_y \dot{y}_r}{z_{r2}} - \frac{v_{r2} \dot{z}_r}{z_{r2}}
\end{align*}$$

where $f_x = f_k_u$, $f_y = f_k_v$.

The above equations can be rewritten in matrix form as:

$$\begin{bmatrix}
\dot{u}_{r1} \\
\dot{v}_{r1} \\
\dot{u}_{r2} \\
\dot{v}_{r2}
\end{bmatrix} = K_i \begin{bmatrix}
\dot{x}_r^c \\
\dot{y}_r^c \\
\dot{z}_r^c
\end{bmatrix} \hspace{1cm} i = 1, 2 \hspace{1cm} (2.16)$$

Where $K_i = 
\begin{bmatrix}
f_x & 0 & -u_{r1}^c & z_{r1}^c \\
0 & f_y & -v_{r1}^c & z_{r1}^c \\
d_{ri}^c & 0 & -u_{r2}^c & z_{r2}^c \\
0 & d_{ri}^c & -v_{r2}^c & z_{r2}^c
\end{bmatrix}$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Here, \( z_{r1}^c \) and \( z_{r2}^c \) represent the depth between the two landmarks and the camera frame, respectively. Equation (2.16) displays the relationship between the landmark's velocity in the camera frame and the rate of change of the image data associated with the landmarks projected into the image plane.

The following equation shows the relationship between the velocity of the landmarks' mirror reflections and rate of change of the image data associated with them in the image plane.

\[
\begin{bmatrix}
\dot{u}_{mi} \\
\dot{v}_{mi}
\end{bmatrix}
= G_i
\begin{bmatrix}
\dot{x}_{mi}^c \\
\dot{y}_{mi}^c \\
\dot{z}_{mi}^c
\end{bmatrix}
\quad i = 1, 2
\tag{2.17}
\]

Where \( G_i = \begin{bmatrix}
f_x & 0 & -U_{mi}^c \\
\frac{f_y}{z_{mi}^c} & -V_{mi}^c \\
0 & \frac{f_y}{z_{mi}^c} & -V_{mi}^c
\end{bmatrix} \)

From Section 2.1, the position of the mirror reflection points is seen to depend on the position of the real point when the mirror is stationary.

From Equations (2.6), (2.16), and (2.17), the following can be written:

\[
\begin{bmatrix}
\dot{u}_{mi} \\
\dot{v}_{mi} \\
\dot{u}_{r1} \\
\dot{v}_{r1} \\
\dot{u}_{m2} \\
\dot{v}_{m2} \\
\dot{u}_{r2} \\
\dot{v}_{r2}
\end{bmatrix}
= C
\begin{bmatrix}
\dot{x}_{r1}^c \\
\dot{y}_{r1}^c \\
\dot{z}_{r1}^c \\
\dot{x}_{r2}^c \\
\dot{y}_{r2}^c \\
\dot{z}_{r2}^c
\end{bmatrix}
\tag{2.18}
\]

where \( C = \begin{bmatrix}
C_1 & 0 \\
0 & C_2
\end{bmatrix} \) and \( C_i = \begin{bmatrix}
G_i F_i \\
K_i
\end{bmatrix} \quad i = 1, 2 \)

Finally, based on Equations (9), (12), and (16), the image Jacobian can be written as:
\[ J_v = DC \begin{bmatrix} \Phi & 0 \\ 0 & \Phi \end{bmatrix} \Pi \]  \hspace{1cm} (2.19)

Where \( \Phi \) denotes the relative rotation between the camera's frame and the robot's frame.

2.6 Distinguishing Landmarks and their Reflections

In this section, the landmark #1 is taken for example and conclusions in this section will also apply to landmark #2.

With reference to Figure 2.4, and based on Equation (2.5), the following can be deduced:

\[ x_{m1}^c = \frac{r}{2\|P_r\| - r} x_r \]  \hspace{1cm} (2.20a)

\[ y_{m1}^c = \frac{r}{2\|P_r\| - r} y_r \]  \hspace{1cm} (2.20b)

\[ z_{m1}^c = \frac{r}{2\|P_r\| - r} (z_r + d) - d \]  \hspace{1cm} (2.20c)

Nevertheless, \( \frac{r}{2} < \|P_r\| < r \) \(^{\text{vi}}\), thus \( 0 < \frac{r}{2\|P_r\| - r} < 1 \)

\(^{\text{vi}}\) The image of the object will always be located between the focus of the spherical mirror and the surface of the spherical mirror \([18]\). One should note that a spherical mirror has multiple focus points.
There are two different conclusions based on three different cases. All the cases will be discussed in the following:

**Case 1:** \( |x_{r1}| \neq 0 \) and \( |y_{r1}| \neq 0 \)

Utilizing Equation (2.20a-2.20c), and the relationship \( 0 < \frac{r}{2\|p_{r1}\|} - r \), one can see:

\[
|x_{m1}| < |x_{r1}| \text{ and } |y_{m1}| < |y_{r1}|. 
\]

Since \( z^r_{r1}, z^m_{m1} \) and \( d \) have the same sign, we can conclude that

\[
|z^m_{m1}| > |d| > |z^r_{r1}|, \text{ considering Equation (2.20c) and the relationship } 0 < \frac{r}{2\|p_{r1}\|} - r < 1.
\]
Also, based on Equation (2.7), the coordinates of the landmark \((u_{rl}, v_{rl})\) and its mirror reflection \((u_{m1}, v_{m1})\) on the image plane can be given as:

\[
\begin{align*}
 u_{rl} &= u_c + \frac{x^c_{rl}}{z^c_{rl}} \alpha_u \\
 v_{rl} &= v_c + \frac{y^c_{rl}}{z^c_{rl}} \alpha_v \\
 u_{m1} &= u_c + \frac{x^c_{m1}}{z^c_{m1}} \alpha_u \\
 v_{m1} &= v_c + \frac{y^c_{m1}}{z^c_{m1}} \alpha_v
\end{align*}
\]

From the above equations, we can conclude:

\[
\begin{align*}
|u_{rl} - u_c| - |u_{m1} - u_c| &= |\alpha_u| \left| \left( \frac{x^c_{rl}}{z^c_{rl}} - \frac{x^c_{m1}}{z^c_{m1}} \right) \right| > 0 \\
|v_{rl} - v_c| - |v_{m1} - v_c| &= |\alpha_v| \left| \left( \frac{y^c_{rl}}{z^c_{rl}} - \frac{y^c_{m1}}{z^c_{m1}} \right) \right| > 0
\end{align*}
\]

Thus, the mirror reflection of landmark #1 projected into the image plane will always remain closer to the center of the image than landmark #1 on the image plane as long as \(|x^c_{rl}| \neq 0\) and \(|y^c_{rl}| \neq 0\).
Case 2: Either $|x_{1i}^c|$ or $|y_{1i}^c|$ is equal to zero

In this case, either Equation (2.22a) or Equation (2.22b) applies. Therefore the mirror reflection of landmark #1 projected into the image plane will still remain closer to the center of the image than landmark #1 on the image plane.

Case 3: Both of $|x_{1i}^c|$ and $|y_{1i}^c|$ are equal to zero

In this case, the camera optical axis passes through the center of the mirror and the system is in a singularity. This particular state will be discussed in detail in Chapter 4.

2.7 Control Law

The image-based visual servoing technique based upon resolved-rate motion control is adopted [1]. In the proposed servoing structure the dynamics of the robot are not taken into account, assuming the robot can be controlled by its velocity screw $T$. The image feature set-point $s^*$ can be derived from the image feature parameters, which place the landmarks and their mirror reflections at their desired locations in the 2D image plane, as follows (according to Figure 1b):

$$s^* = \begin{bmatrix} l_1^* \\ d_1^* \\ l_2^* \\ d_2^* \end{bmatrix}$$ (2.23)
A resolved-rate control law is presented as:

$$T = -\lambda \hat{J}_v^{-1}(s - s^*)$$ (2.24)

Where $T$ denotes the velocity screw of the Mid-Point frame given as:

$$T = \begin{bmatrix} x & y & z & \phi & \theta \end{bmatrix}$$

and $\hat{J}_v^{-1}$ denotes the inverse of the estimated image Jacobian, which ensures that the exponential decrease in the error in image features $(s - s^*)$, and $\lambda$ is a control gain matrix of the appropriate dimension. When $\lambda$ is a positive semi definite matrix, the image error $(s - s^*)$ will exponentially decrease [1]. In order to maintain the velocity of the robot within the given limits, $\lambda$ can be computed as follow, [18]:

$$\lambda = diag(K_T, K_T, K_T, K_R, K_R)$$ (2.25)

Define $[V, \Omega]^T = \hat{J}_v^{-1}(s - s^*)$, $V = [v_x, v_y, v_z]^T$, $\Omega = [\Omega_{\phi}, \Omega_{\theta}]^T$, then $k_T$ and $k_R$ can be calculated as:

$$k_T = \min \left( \frac{\|\Delta V_{\text{max}}\|}{\|V\|}, 1 \right)$$ (2.26a)

$$k_R = \min \left( \frac{\|\Delta \Omega_{\text{max}}\|}{\|\Omega\|}, 0.05 \right)$$ (2.26b)

where $\Delta V_{\text{max}}$ is maximum allowable increment in the translation velocity of the robotic end-effector and $\Delta \Omega_{\text{max}}$ is maximum allowable increment in the rotational velocity of the end-effector.

---

$^\text{VII}$ $\lambda$ is called a positive semi definite matrix if and only if $x^T \lambda x \geq 0$ (for vector $x \neq 0$)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
robotic end-effector. The control gain, calculated in equation (2.25), will prevent the robot from operating over the maximum velocity, which could damage the robot. This control law attempts to move the robot until the error function on the image feature decreases to zero.

2.8 Summary

In this chapter, the coordinate transformations of the mirror, camera and robot have been described. The image Jacobian which is the relationship between the rate of change of a quintuplet image feature and a quintuplet velocity screw of the Mid-point frame has been discussed in detail. The proportional control law for servoing purpose has been also described.
3. A Dual Kalman Filter for Parameters Identification and Tracking

In the proposed system, a dual Kalman Filter (KF) is employed. One part is for estimating the intrinsic and extrinsic parameters of the camera, and the other part is for tracking the coordinates of the landmarks and their mirror reflections on the image plane. In this chapter, the dual KF is discussed more extensively.

3.1 Introduction to Kalman Filter and Extended Kalman Filter

In 1960, R.E. Kalman published his famous paper describing a recursive solution to the discrete-data linear filtering problem [26]. Since that time, due in large part to advances in digital computing, the Kalman filter has been the subject of extensive research and application, particularly in the area of autonomous or assisted navigation. More extensive references include [27, 28].

3.1.1 The Discrete Kalman Filter

This section describes the filter in its original formulation [26] where the measurement occurs, and the system's states are estimated, at discrete points in time.
The Process to be estimated

The Kalman filter addresses the general problem of trying to estimate the state \( x \in \mathbb{R}^n \) of a discrete-time controlled process that is governed by the linear stochastic difference equation:

\[
x_k = F_k x_{k-1} + B_k u_k + w_k
\]

with a measurement \( z \in \mathbb{R}^m \) that is

\[
z_k = H_k x_k + v_k
\]

The random variables \( w_k \) and \( v_k \) represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions:

\[
p(w_k) \sim N(0,Q_k)
\]

\[
p(v_k) \sim N(0, R_k)
\]

The \( n \times n \) matrix \( F_k \) in the difference equation (3.1) relates the state at time step \( k-1 \) to the state at step \( k \). The \( n \times l \) matrix \( B_k \) relates the control input \( u_k \in \mathbb{R}^l \) to the state \( x_k \).

The \( m \times n \) matrix \( H_k \) in the measurement equation (3.2) relates the state to the measurement \( z_k \).
The Discrete Kalman Filter Algorithm

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for projecting forward (in time) the current state and error covariance estimates to obtain the a priori estimates for the next time step. The measurement update equations are responsible for the feedback—i.e. for incorporating a new measurement into the a priori estimate $\hat{x}_{k|k-1}$ to obtain an improved a posteriori estimate $\hat{x}_{k|k}$.

The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Indeed the final estimation algorithm resembles that of a predictor-corrector algorithm for solving numerical problems as shown below in Figure 3.1.

![Figure 3.1: The ongoing discrete Kalman filter cycle. The time update projects the current state estimate ahead in time. The measurement update adjusts the projected estimate by an actual measurement at that time.](image)
The specific equations for the time and measurement updates are presented below:

Discrete Kalman filter time update equations:

\[ \hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \quad \text{a priori estimate state} \quad (3.5a) \]

\[ P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad \text{a priori error covariance} \quad (3.5b) \]

Discrete Kalman filter measurement update equations:

\[ \tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \quad \text{innovation} \quad (3.6a) \]

\[ S_k = H_k P_{k|k-1} H_k^T + R_k \quad \text{innovation covariance} \quad (3.6b) \]

\[ K_k = P_{k|k-1} H_k^T S_k^{-1} \quad \text{optimal Kalman gain} \quad (3.6c) \]

\[ \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \quad \text{a posteriori estimate state} \quad (3.6d) \]

\[ P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad \text{a posteriori error covariance} \quad (3.6e) \]

The first task during the measurement update is to compute the Kalman filter gain, \( K_k \).

The next step is to actually measure the process to obtain \( z_k \), and then to generate an a posteriori state estimate by incorporating the measurement as in (3.6d). The final step is to obtain an a posteriori error covariance estimate via (3.5e).

After each time and measurement update pair, the process is repeated with the previous posteriori estimates used to project or predict the new priori estimates. This recursive
nature is one of the very appealing features of the Kalman filter. One should note that \( \hat{x}_{t|0}, P_{t|0} \) are the initial values which should be given before the recursive process.

### 3.1.2 The Extended Kalman Filter (EKF)

As described in above section, the Kalman filter addresses the general problem of trying to estimate the state \( \mathbf{x} \in \mathbb{R}^n \) of a discrete-time controlled process that is governed by a linear stochastic difference equation. A Kalman filter that linearizes about the current mean and covariance is referred to as an Extended Kalman Filter or EKF.

**The Process to be estimated**

In something akin to a Taylor series, one can linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates even in the face of non-linear relationships. To do so, one must begin by modifying some of the material presented in above section. Let us assume that our process again has a state vector \( \mathbf{x} \in \mathbb{R}^n \), but that the process is now governed by the non-linear stochastic difference equation:

\[
\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k) \tag{3.7}
\]

with a measurement \( \mathbf{z} \in \mathbb{R}^m \) that is:
\[ z_k = h(x_k, v_k) \quad (3.8) \]

where the random variables \( w_k \) and \( v_k \) again represent the process and measurement noise as in (3.3) and (3.4). The non-linear function \( f(\bullet) \) can be used to compute the predicted state from the previous estimate and similarly the non-linear function \( h(\bullet) \) can be used to compute the predicted measurement from the predicted state.

In practice of course one does not know the individual values of the noise \( w_k \) and \( v_k \) at each time step. However, one can approximate the state and measurement vector without them as:

\[ \tilde{x}_k = f(\hat{x}_{k|k-1}, u_k, 0) \quad (3.9a) \]
\[ \tilde{z}_k = h(\tilde{x}_k, 0) \quad (3.9b) \]

where \( \hat{x}_{k|k-1} \) is some a posteriori estimate of the state (from a previous time step \( k - 1 \)).

**The Computational Origins of the Filter**

To estimate a process with non-linear difference and measurement relationships, new governing equations that linearize an estimate about equation (3.9a) and (3.9b) can be written as:

\[ x_k = \tilde{x}_k + F(x_{k-1} - \hat{x}_{k|k-1}) + Ww_{k-1} \quad (3.10a) \]
\[ z_k = \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_{k-1} \quad (3.10b) \]
where

- $x_k$ and $z_k$ are the actual state and measurement vectors

- $\tilde{x}_k$ and $\tilde{z}_k$ are the approximate state and measurement vectors from equation (3.9a) and equation (3.9b)

- $\hat{x}_{k|k-1}$ a posteriori estimate of the state at step $k-1$

- random variables $w_k$ and $v_k$ again represent the process and measurement noise as in equation (3.3) and (3.4)

- $F$ is the Jacobian matrix of partial derivatives of $f(*)$ with respect to $x$, that is $F_k = \frac{\partial f}{\partial x}

- $W$ is the Jacobian matrix of partial derivatives of $f(*)$ with respect to $w$, that is $W_k = \frac{\partial f}{\partial w}

- $H$ is the Jacobian matrix of partial derivatives of $h(*)$ with respect to $x$, that is $H_k = \frac{\partial h}{\partial x}

- $V$ is the Jacobian matrix of partial derivatives of $h(*)$ with respect to $v$, that is $V_k = \frac{\partial h}{\partial v}$
The Extended Kalman Filter Algorithm

Time update equations for the discretized EKF:

\[ \hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k, 0) \quad a \text{ priori estimate state} \quad (3.11a) \]

\[ P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad a \text{ priori error covariance} \quad (3.11b) \]

As with the basic discrete Kalman filter, the time update equations project the state and covariance estimates from time step \( k - 1 \) to step \( k \). \( F_k \) and \( W_k \) are the process Jacobians at step \( k \), and \( Q_k \) is the process noise covariance (3.3) at step \( k \).

Measurement update equations for the discretized EKF:

\[ \tilde{y}_k = z_k - h(\hat{x}_{k|k-1}, 0) \quad \text{innovation} \quad (3.12a) \]

\[ S_k = H_k P_{k|k-1} H_k^T + R_k \quad \text{innovation covariance} \quad (3.12b) \]

\[ K_k = P_{k|k-1} H_k^T S_k^{-1} \quad \text{optimal Kalman gain} \quad (3.12c) \]

\[ \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \quad a \text{ posteriori estimate state} \quad (3.12d) \]

\[ P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad a \text{ posteriori error covariance} \quad (3.12e) \]

As with the basic discrete Kalman filter, the measurement update equations correct the state and covariance estimates with the measurement \( z_k \). Again \( h(\bullet) \) in (3.12a) comes from (3.10), \( H_k \) and \( V_k \) are the measurement Jacobian at step \( k \), and \( R_k \) is the
measurement noise covariance (3.4) at step \( k \). The basic operation of the EKF is the same as the linear discrete Kalman filter as shown in Figure 3.1.

An important feature of the EKF is that the Jacobian \( H_k \) in the equation for the Kalman gain \( K_k \) serves to correctly propagate or "magnify" only the relevant component of the measurement information. For example, if there is not a one-to-one mapping between the measurement \( z_k \) and the state via \( h(\bullet) \), the Jacobian \( H_k \) affects the Kalman gain so that it only magnifies the portion of the residual \( z_k - h(\hat{x}_{k|k-1},0) \) that does affect the state. Of course if over all measurements there is not a one-to-one mapping between the measurement \( z_k \) and the state via \( h(\bullet) \), then as you might expect the filter will quickly diverge. The control theory term to describe this situation is unobservable.

### 3.2 Parameter Estimations using the Kalman Filter

To determine the image Jacobian, the intrinsic and the extrinsic parameters need to be known. The intrinsic parameters in the system are \( \alpha_u \), \( \alpha_v \), \( u_c \) and \( v_c \) which were described in section 2.2. The extrinsic parameter is only the distance \( d \) between the origin of the camera's frame and the origin of the spherical mirror's frame. The Kalman filter is utilized to estimate these five parameters to calculate image Jacobian through image observations. The five intrinsic and extrinsic parameters of the camera are defined as the state vector \( x_k = [u_c \ v_c \ \alpha_u \ \alpha_v \ d]^T \). Since these five parameters do not vary, the state transition model can be represented as:
\[ x_k = x_{k-1} + w_k \]  

(3.13)

where the \( w_k \) is the process noise, which is assumed to be drawn from a zero mean Gaussian white noise with covariance \( Q_k \).

The coordinates of the landmarks and their mirror reflections on the image plane are defined as the measurement vector \( z_k = [u_{m1} \quad v_{m1} \quad u_{r1} \quad v_{r1} \quad u_{m2} \quad v_{m2} \quad u_{r2} \quad v_{r2}]^T \).

Based on equation (2.7) and the transformation matrix between the mirror frame and the camera frame \( cT_m \), the measurement function \( h(\cdot) \) can be represented as:

\[ z_k = h(x_k, v_k) \]  

(3.14)

where

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & -d \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
h = \begin{bmatrix}
u_c + \frac{x_{m1}}{z_{m1} - d} \alpha_u \\
v_c + \frac{y_{m1}}{z_{m1} - d} \alpha_v \\
u_c + \frac{x_{r1}}{z_{r1}} \alpha_u \\
v_c + \frac{y_{r1}}{z_{r1}} \alpha_v \\
u_c + \frac{x_{m2}}{z_{m2} - d} \alpha_u \\
v_c + \frac{y_{m2}}{z_{m2} - d} \alpha_v \\
u_c + \frac{x_{r2}}{z_{r2} - d} \alpha_u \\
v_c + \frac{y_{r2}}{z_{r2} - d} \alpha_v
\end{bmatrix}
\]  

(3.15)
$v_k$ is the observation noise which is assumed to be zero mean Gaussian white noise with covariance $R_k$.

As the transformation between the mirror frame and the robot frame is known, the parameters $x_m^1, y_m^1, z_m^1, x_r^1, y_r^1, z_r^1, x_m^2, y_m^2, z_m^2, x_r^2, y_r^2$ and $z_r^2$ are all known.

Then the observation matrix $H_k$ can be defined as:

$$
H_k = \frac{\partial h}{\partial \hat{x}} \bigg|_{\hat{x}_{k-1}} = 
\begin{bmatrix}
1 & 0 & \frac{x_m^1}{z_m^1 - d_k} & 0 & \frac{x_m^1}{(z_m^1 - d_k)^2} \\
0 & 1 & 0 & \frac{y_m^1}{z_m^1 - d_k} & \frac{y_m^1}{(z_m^1 - d_k)^2} \\
1 & 0 & \frac{x_r^1}{z_r^1 - d_k} & 0 & \frac{x_r^1}{(z_r^1 - d_k)^2} \\
0 & 1 & 0 & \frac{y_r^1}{z_r^1 - d_k} & \frac{y_r^1}{(z_r^1 - d_k)^2} \\
1 & 0 & \frac{x_m^2}{z_m^2 - d_k} & 0 & \frac{x_m^2}{(z_m^2 - d_k)^2} \\
0 & 1 & 0 & \frac{y_m^2}{z_m^2 - d_k} & \frac{y_m^2}{(z_m^2 - d_k)^2} \\
1 & 0 & \frac{x_r^2}{z_r^2 - d_k} & 0 & \frac{x_r^2}{(z_r^2 - d_k)^2} \\
0 & 1 & 0 & \frac{y_r^2}{z_r^2 - d_k} & \frac{y_r^2}{(z_r^2 - d_k)^2}
\end{bmatrix}
$$

(3.16)

The extended Kalman filter for parameters estimation can be presented as:

**Time update equations:**

\[
\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1} \quad \text{a priori estimate state} \quad (3.17a)
\]

\[
P_{k|k-1} = P_{k-1|k-1} + Q_k \quad \text{a priori error covariance} \quad (3.17b)
\]
Measurement update equations:

\[
\bar{y}_k = z_k - h(\hat{x}_{k|k-1}, 0)
\]

innovation (3.18a)

\[
S_k = H_k P_{k|k-1} H_k^T + R_k
\]

innovation covariance (3.18b)

\[
K_k = P_{k|k-1} H_k^T S_k^{-1}
\]

optimal Kalman gain (3.18c)

\[
\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \bar{y}_k
\]

\emph{a posteriori} estimate state (3.18d)

\[
P_{k|k} = (I - K_k H_k) P_{k|k-1}
\]

\emph{a posteriori} error covariance (3.18e)

3.3 Coordinates Tracking

In this section, three KF filters are developed to improve the noise sensitivity of the system. They are based on three different kinematic models, namely second-order, third-order, and Gaussian-Markov kinematic models.

3.3.1 Second-Order Kinematic Model

In this model, we assume that the rate of change of the variables of the interest, namely \( (u_{r1}, v_{r1}, u_{r2}, v_{r2}, u_{m1}, v_{m1}, u_{m2}, v_{m2}) \) is constant corrupted by a white Gaussian noise, so that the first derivatives of the coordinates of the landmarks and mirror reflections projected onto the image plane, namely
\( \begin{pmatrix} \dot{u}_1 & \dot{v}_1 & \dot{u}_2 & \dot{v}_2 & \dot{u}_m & \dot{v}_m & \dot{u}_n & \dot{v}_n \end{pmatrix} \) can be considered constant over a sampling period. The effect of their higher-order derivatives may then be considered as disturbance noise added to the kinematic model. Mathematically speaking the kinematic model of changes in the image data is:

\[
\frac{d}{dt}(\mathbf{e}(t)) = \mathbf{v}(t) \tag{3.19}
\]

where \( \mathbf{e}(t) = \begin{bmatrix} u_1 & v_1 & u_2 & v_2 & u_m & v_m & u_n & v_n \end{bmatrix} \) and \( \mathbf{v}(t) \) denote a zero-mean white noise with covariance \( q \). The state vector corresponding to (3.19) is

\[
\mathbf{c} = \begin{bmatrix} \mathbf{e} & \dot{\mathbf{e}} \end{bmatrix}^T \tag{3.20}
\]

hence the name second order kinematic model, [28].

The continuous-time state equation is:

\[
\dot{\mathbf{e}}(t) = A\mathbf{e}(t) + D\mathbf{v}(t) \tag{3.21}
\]

where

\[
A = \begin{bmatrix} 0_{8x8} & I_{8x8} \\ 0_{8x8} & 0_{8x8} \end{bmatrix} \tag{3.22a}
\]

\[
D = \begin{bmatrix} 0_{8x1} \\ 1_{8x1} \end{bmatrix} \tag{3.22b}
\]

The discrete time state equation with sampling period \( \Delta T \) will be:

\[
\mathbf{c}_{k+1} = F\mathbf{c}_k + \mathbf{v}_k \tag{3.23}
\]
where

\[ F = e^{Ax\Delta T} = \begin{bmatrix} I_{8x8} & \Delta T \times I_{8x8} \\ 0 & I_{8x8} \end{bmatrix} \tag{3.24} \]

and the discrete time process noise relates to the continuous time as follows:

\[ v_k = \int_0^{\Delta T} e^{A(t\Delta T - t)} Dv(k\Delta T + t) dt \tag{3.25} \]

From the above, the covariance of the discrete time process noise \( v_k \), assuming \( q \) to be constant, is

\[ Q = E(v_k v_k^T) = \begin{bmatrix} \frac{1}{3} \Delta T^3 \times I_{8x8} & \frac{1}{2} \Delta T^2 \times I_{8x8} \\ \frac{1}{2} \Delta T^2 \times I_{8x8} & \Delta T \times I_{8x8} \end{bmatrix} q \tag{3.26} \]

Since the coordinates of the landmarks \((u_{r1}, v_{r1}, u_{r2}, v_{r2}, u_{m1}, v_{m1}, u_{m2}, v_{m2})\) are measurable, the measurement model can be defined as:

\[ z_{k+1} = H z_k + \eta_k \tag{3.27} \]

Where \( z_k = [u_{r1}, v_{r1}, u_{r2}, v_{r2}, u_{m1}, v_{m1}, u_{m2}, v_{m2}]^T \) and \( H = [I_{8x8} \mid 0_{8x8}]^T \)

In equation (3.27), \( \eta_k \) denotes measurement noise, which is assumed to be described by a zero mean white noise with covariance \( R \).
3.3.2 Third-Order Kinematic Model (or Weiner model)

In this model, we assume that the rate of change of the variables of interest namely 
\((\dot{u}_{r_1} \quad \dot{v}_{r_1} \quad \dot{u}_{r_2} \quad \dot{v}_{r_2} \quad \dot{u}_{m_1} \quad \dot{v}_{m_1} \quad \dot{u}_{m_2} \quad \dot{v}_{m_2})\) is assumed to be constant with a white Gaussian noise, so that the second derivatives of the coordinates of the landmarks and mirror reflections projected onto the image plane, namely 
\((\ddot{u}_{r_1} \quad \ddot{v}_{r_1} \quad \ddot{u}_{r_2} \quad \ddot{v}_{r_2} \quad \ddot{u}_{m_1} \quad \ddot{v}_{m_1} \quad \ddot{u}_{m_2} \quad \ddot{v}_{m_2})\) can be considered constant over a sampling period. The effect of their higher-order derivatives may then be considered as disturbance noise added to the kinematic model. Mathematically speaking the kinematic model of changes in the image data is:

\[
\frac{d}{dt}(\ddot{e}(t)) = \nu(t) \tag{3.28}
\]

where \(e(t) = [u_{r_1} \quad v_{r_1} \quad u_{r_2} \quad v_{r_2} \quad u_{m_1} \quad v_{m_1} \quad u_{m_2} \quad v_{m_2}]\) and \(\nu(t)\) denote a zero-mean white noise with covariance \(\sigma\). The state vector corresponding to (3.28) is

\[
\mathbf{e} = [e \quad \dot{e} \quad \ddot{e}]^T \tag{3.29}
\]

hence the name third order kinematic model, [28].

The continuous-time state equation is:

\[
\dot{e}(t) = Ae(t) + Du(t) \tag{3.30}
\]

where
The discrete time state equation with sampling period $\Delta T$ will be:

$$c_{k+1} = Fc_k + v_k$$ \hspace{1cm} (3.32)

where

$$F = e^{A\Delta T} = \begin{bmatrix} I_{8\times8} & \Delta T \times I_{8\times8} & \frac{1}{2} \times \Delta T^2 \times I_{8\times8} \\ 0 & I_{8\times8} & \Delta T \times I_{8\times8} \\ 0 & 0 & I_{8\times8} \end{bmatrix}$$ \hspace{1cm} (3.33)

and the discrete time process noise relates to the continuous time as follows:

$$v_k = \int_0^{\Delta T} e^{A(\Delta T-t)} Dv(k\Delta T + t) dt$$ \hspace{1cm} (3.34)

From the above, the covariance of the discrete time process noise $v_k$, assuming $q$ to be constant, is
\[ Q = E(v_k v_k^T) \]

\[
\begin{bmatrix}
\frac{1}{20} \Delta T^5 \times I_{8 \times 8} & \frac{1}{8} \Delta T^4 \times I_{8 \times 8} & \frac{1}{6} \Delta T^3 \times I_{8 \times 8} \\
\frac{1}{8} \Delta T^4 \times I_{8 \times 8} & \frac{1}{6} \Delta T^3 \times I_{8 \times 8} & \frac{1}{2} \Delta T^2 \times I_{8 \times 8} \\
\frac{1}{6} \Delta T^3 \times I_{8 \times 8} & \frac{1}{2} \Delta T^2 \times I_{8 \times 8} & \Delta T \times I_{8 \times 8}
\end{bmatrix}
\] (3.35)

Since the coordinates of the landmarks \((u_r, v_r, u_{r2}, v_{r2}, u_m, v_m, u_{m2}, v_{m2})\) are measurable, the measurement model can be defined as:

\[ z_{k+1} = H z_k + \eta_k \] (3.36)

Where \( z_{k+1} = [u_r, v_r, u_{r2}, v_{r2}, u_m, v_m, u_{m2}, v_{m2}]^T \), \( H = [I_{8 \times 8} \ 0_{8 \times 8} \ 0_{8 \times 8}]^T \)

In equation (3.36), \( \eta_k \) denotes measurement noise, which is assumed to be described by a zero mean white noise with covariance \( R \).

3.3.3 The Gaussian-Markov Kinematic Model

In this model, the landmarks’ equation of motion represented on the image plane is given as:

\[
\frac{d}{dt}(\ddot{z}(t)) = -\frac{1}{\tau} \ddot{z}(t) + \nu(t)
\] (3.37)

where \( \ddot{z}(t) = [u_r, v_r, u_{r2}, v_{r2}, u_m, v_m, u_{m2}, v_{m2}] \) and \( \nu(t) \) denote a zero-mean white noise with covariance \( q \). In this model, it is assumed that the acceleration is changing within a sampling period as opposed to that in the two previous models.
The state vector corresponding to (3.37) is
\[ c = [\xi \dot{\xi} \ddot{\xi}]^T \] (3.38)
this model was suggested by Gaussian, hence its name is Gaussian-Markov kinematic model, [28].

The continuous time state equation is
\[ \dot{c}(t) = Ac(t) + Dv(t) \] (3.39)
where
\[ A = \begin{bmatrix} 0 & I_{8x8} & 0_{8x8} \\ 0_{8x8} & 0_{8x8} & I_{8x8} \\ 0_{8x8} & 0_{8x8} & -\frac{1}{\tau} I_{8x8} \end{bmatrix} \] (3.40a)
\[ D = \begin{bmatrix} 0_{8x1} \\ 0_{8x1} \\ 1_{8x1} \end{bmatrix} \] (3.40b)

The discrete time state equation with sampling period \( \Delta T \) will be:
\[ c_{k+1} = Fc_k + v_k \] (3.41)
where
$$F = e^{(AxAT)} = \begin{bmatrix}
I_{8\times 8} & \Delta T \times I_{8\times 8} & \tau^2 \left( -1 + \frac{\Delta T}{\tau} + e^{\frac{\Delta T}{\tau}} \right) \\
0 & I_{8\times 8} & \tau^2 \left( -1 + e^{\frac{\Delta T}{\tau}} \right) \\
0 & 0 & I_{8\times 8}
\end{bmatrix}$$ \quad (3.42)

and the discrete time process noise relates to the continuous time as follows:

$$v_k = \int_0^{\Delta T} e^{(A(k\Delta T+t))}Dv(k\Delta T+t)dt$$ \quad (3.43)

From the above, the covariance of the discrete time process noise $v_k$, assuming $q$ to be constant and $\frac{\Delta T}{\tau} << 1$, is

$$Q = E(v_kv_k^T) = \frac{2q}{\tau} \begin{bmatrix}
\frac{1}{20} \Delta T^5 \times I_{8\times 8} & \frac{1}{8} \Delta T^4 \times I_{8\times 8} & \frac{1}{6} \Delta T^3 \times I_{8\times 8} \\
\frac{1}{8} \Delta T^4 \times I_{8\times 8} & \frac{1}{6} \Delta T^3 \times I_{8\times 8} & \frac{1}{2} \Delta T^2 \times I_{8\times 8} \\
\frac{1}{6} \Delta T^3 \times I_{8\times 8} & \frac{1}{2} \Delta T^2 \times I_{8\times 8} & \Delta T \times I_{8\times 8}
\end{bmatrix}$$ \quad (3.44)

Since the coordinates of the landmarks $(u_{r1}, v_{r1}, u_{r2}, v_{r2}, u_{m1}, v_{m1}, u_{m2}, v_{m2})$ are measurable, the measurement model can be defined as:

$$z_{k+1} = Hz_k + n_k$$ \quad (3.45)

Where $z_k = [u_{r1}, v_{r1}, u_{r2}, v_{r2}, u_{m1}, v_{m1}, u_{m2}, v_{m2}]^T$, $H = [I_{8\times 8} \mid 0_{8\times 8} \mid 0_{8\times 8}]^T$
In equation (3.45) $\eta_k$ denotes measurement noise, which is assumed to be described by a zero mean white noise with covariance $R$.

One should note that, for small $\frac{\Delta T}{\tau}$, equation (3.42) reduces to equation (3.33), which means for small value of $\frac{\Delta T}{\tau}$, the Gaussian-Markov model will reduce to the third-order model (Weiner model).

3.3.4 Implementation of a Kalman filter using three kinematic models

The three models described above can be used in conjunction with a KF to track the coordinates of the landmarks on the image plane. The recursive equations defining the KF algorithm are given by the following:

Time update equations:

\[
\hat{\mathbf{c}}_{k|k-1} = F\hat{\mathbf{c}}_{k-1|k-1} \quad \text{a priori estimate state} \quad (3.46a)
\]

\[
P_{k|k-1} = FP_{k-1|k-1}F^T + Q \quad \text{a priori error covariance} \quad (3.46b)
\]

Measurement update equations:

\[
\bar{y}_k = z_k - H\hat{\mathbf{c}}_{k|k-1} \quad \text{innovation} \quad (3.47a)
\]

\[
S_k = H_k P_{k|k-1} H_k^T + R \quad \text{innovation covariance} \quad (3.47b)
\]
\[ K_k = P_{k|k-1} H_k^T S_k^{-1} \] optimal Kalman gain (3.47c)

\[ \hat{e}_{k|k} = \hat{e}_{k|k-1} + K_k \tilde{y}_k \] a posteriori estimate state (3.47d)

\[ P_{k|k} = (I - K_k H_k) P_{k|k-1} \] a posteriori error covariance (3.47e)

3.4 Summary

In this chapter, the Kalman filter and the Extended Kalman filter have been fully reviewed. Based on those theories, a dual Kalman filter is developed for system identification and coordinate tracking. In the system identification part, the KF was used to estimate the camera’s intrinsic and extrinsic parameters. In coordinate tracking part, the KF was developed to improve the noise sensitivity of the system and the coordinates of the landmarks and their mirror reflections on the image plane were considered as three different kinematic models, namely second-order, third-order, and Gaussian-Markov kinematic model.
4. Singularity in the Image Jacobian

Singularity of the image Jacobian can occur under two conditions as: (1) $u_{m1} - u_{r1} = 0$ and $v_{m1} - v_{r1} = 0$, or (2) $u_{m2} - u_{r2} = 0$ and $v_{m2} - v_{r2} = 0$. In the following section, the relative configuration in the camera and mirror that would cause the singularity will be investigate in more detail.

4.1 Singularity in Matrix D

Two conditions can make the matrix D (give in equation (2.19)) singular; one is $u_{m1} - u_{r1} = 0$ and $v_{m1} - v_{r1} = 0$, and the other is $u_{m2} - u_{r2} = 0$ and $v_{m2} - v_{r2} = 0$. These two conditions only exist when either of the two landmarks is on the optical axis of the camera. The proof is given below:

Considering landmark #1, and the based projection model of the camera, the following equations can be deduced:

\[
\begin{align*}
    u_{ml} &= \frac{f_x}{z_{ml}} x^c_{ml} + c_x, \\
    u_{rl} &= \frac{f_x}{z_{rl}} x^c_{rl} + c_x, \\
    v_{ml} &= \frac{f_y}{z_{ml}} y^c_{ml} + c_y, \\
    v_{rl} &= \frac{f_y}{z_{rl}} y^c_{rl} + c_y
\end{align*}
\]

Therefore, if $u_{ml} = u_{rl}$, $v_{ml} = v_{rl}$, the one must have: $x^c_{ml} = x^c_{rl}$ and $y^c_{ml} = y^c_{rl}$. From the chapter 2, $z^c_{ml} < z^c_{rl} < 0$, therefore the equations for $x^c_{ml} = x^c_{rl}$ and $y^c_{ml} = y^c_{rl}$ can be
correct only if $x_{m1}^c = x_{r1}^c = 0$ and $y_{m1}^c = y_{r1}^c = 0$. This boils down to a situation where the landmark #1 and its mirror reflection are both on the optical axis (Figure 4.1).

Figure 4.1: Singularity in the Image Jacobian: landmark #1 and its mirror reflection both lie on the camera’s optical axis.

4.2 Other Possible Singularities in the System

There are four possible singular configurations which are discussed below:

Case 1: Landmark #1 and landmark #2’s mirror reflection coincide on the image plane (Figure 4.2).

Case 2: Landmark #2 and landmark #1’s mirror reflection coincide on the image plane (Figure 4.3).

Case 3: Landmark #1 and the landmark #2 coincide on the image plane (Figure 4.4).
Case 4: Landmark #1's reflection and landmark #2's reflection point coincide on the image plane (Figure 4.5).

Figure 4.2: Singular configuration: landmark #1 and landmark #2's mirror reflection coincide in the image plane.
**Figure 4.3:** Singular configuration: landmark #2 and landmark #1's mirror reflection coincide in the image plane.

**Figure 4.4:** Singular configuration: landmark #1 and landmark #2 coincide in the image plane.
Figure 4.5: Singular configuration: landmark #1’s mirror reflection and landmark #2’s mirror reflection coincide in the image plane.

Cases 1, 2, 3 and 4 have a common characteristic in that each case has two points (landmarks or their reflections) that are in the same projection line. Therefore, on the image plane, only one projection point can be seen.

These singular situations could lead the system to a lack of information for calculating the five image features and then to a failure of the visual servoing system.

These singularities can be avoided (or it’s likelihood to occur being minimized) by (1) properly setting up the camera/mirror system with respect to the robot frame in advance or (2) to deviate from the serving control law momentarily where the image Jacobian gets ill-conditioned. In this research work, the former strategy was adopted.
5. Simulation Results and Sensitivity Tests

The simulation environment was developed with MATLAB (version 7.0) using the robotics toolbox developed by Corke [22]. In this chapter, the simulation results and sensitivity tests on the system are presented.

5.1 The Animation Robot

In the proposed visual servoing simulation system, a 5-DOF mobile manipulator is built to make this simulation system more realistic. This 5-DOF robot is built by the Robotic Toolbox. The robot's base moves on a slide defining a prismatic joint as the first joint of the robot and the rest joints of the robot are revolute joints. The kinematic structure of this robot can be succinctly described in terms of Denavit-Hartenberg parameters. The Denavit-Hartenberg [17] parameters of the each joint in the proposed visual servoing simulation system are given in the following table:
Table 5.1: Denavit-Hartenberg parameters of each joint.

<table>
<thead>
<tr>
<th>Joint</th>
<th>Denavit-Hartenberg parameters</th>
<th>operational joint type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>link twist angle (rad) $\alpha_i$</td>
<td>Link offset distance (mm) $A_i$</td>
</tr>
<tr>
<td>1</td>
<td>$\pi/2$</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$\pi/2$</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>200</td>
</tr>
</tbody>
</table>

The animated robot in zero position is shown as:

Figure 5.1: The animation robot.
5.2 Simulation Results

Figure 5.2 shows a schematic diagram of the workspace setup used for the simulations. In the simulations, the distance between the center of the camera’s frame and the center of the mirror’s frame can be changed by the user. The relative configuration between the robot and the mirror is fixed. The initial position and orientation of the robot, and the target position and orientation can be changed by the user in the simulations. The simulation results are presented based on three kinematic models, described in Chapter 3.

5.2.1 Simulation Setup

Figure 5.2: Simulation Setup.
The simulation setup (Figure 5.2) data used in the proposed visual servoing system is given as follows:

a) The robot’s base was located at (600, 800, 2100) mm from the mirror frame (located at the sphere’s center). The radius of the spherical mirror was assumed to be 5000 mm. Through the simulations, the landmarks mounted on the robot and their mirror reflections were assumed to remain within the camera’s field of view at all times.

b) The intrinsic parameters of the camera were chosen to be:
\[ \alpha_u = -998.97, \alpha_v = -916.23, u_c = 342.70, v_c = 236.88 \]. The extrinsic parameter \( d \), which consists of the distance between the camera frame and the mirror frame was 4000 mm.

c) The initial positions of two landmarks were at (-312.5, 221.7, 306.7) mm and (-287.5, 178.3, 393.3) mm with respect to the robot’s base frame. Correspondingly, the middle point of the line segment connecting two landmarks was located at (-300, 200, 350) mm and the initial orientation, defined in section 2.3, were \( \varphi = -\frac{\pi}{3}, \theta = -\frac{\pi}{6} \). The desired positions of two landmarks were located at (362.5, -171.7, 425.0) mm and (437.5, -128.3, 475.0) mm with respect to the robot’s base frame. Correspondingly, the middle point of the line segment connecting two desired points was located at (400, -150, 450) mm and the desired orientations defined in section 2.3 were \( \varphi = -\frac{\pi}{6}, \theta = -\frac{\pi}{3} \).
d) The initial guess of the camera’s intrinsic and extrinsic parameters deviated from their real values by +15% on average.

e) A Gaussian noise with a 2-pixel variance was added to the image extraction algorithm.

5.2.2 Simulation Results Based on the Second-order Kinematic Model

The simulation results based on the second-order kinematic model are illustrated in Figure 5.3.
Figure 5.3: Simulation results based on the second-order kinematic model: (a) Trajectory of the landmarks and their mirror reflections on the 2D image plane.
Figure 5.3 (continued): (b) Image error.
Figure 5.3 (continued): (c) Trajectory of the landmarks in the 3D space.
Figure 5.3 (continued): (d) Position of the robot.
Figure 5.3 (continued): (e) Joint trajectory of the robot.
Figure 5.3 (continued): (f) Position error of landmark #1.
Figure 5.3 (continued): (g) Position error of landmark #2.
Figure 5.3 (continued): (h) Estimation error of camera parameters.
Estimate position (Green line) vs. Real position (Red line)

\[ U_{m1} \text{ (pixel)} \]

\[ V_{m1} \text{ (pixel)} \]

\[ U_{r1} \text{ (pixel)} \]

\[ V_{r1} \text{ (pixel)} \]

\[ U_{m2} \text{ (pixel)} \]

\[ V_{m2} \text{ (pixel)} \]

\[ U_{r2} \text{ (pixel)} \]

\[ V_{r2} \text{ (pixel)} \]

Time (Sec.)

(i)

**Figure 5.3 (continued):** (i) Estimation of the coordinates corrupted by noise using a KF.
5.2.3 Simulation Results Based on the Third-order Kinematic Model

The simulation results based on the third-order kinematic model are illustrated in Figure 5.4.

Figure 5.4: Simulation results based on the third-order kinematic model: (a) Trajectory of the landmarks on the 2D image plane.
Figure 5.4(continued): (b) Image error.
Figure 5.4 (continued): (c) Trajectory of the landmarks in the 3D space.
Figure 5.4 (continued): (d) Position of the robot.
Figure 5.4 (continued): (e) Joint trajectory of the robot.
Figure 5.4 (continued): (f) Position error of landmark #1.
Figure 5.4 (continued): (g) Position error of landmark #2.
Figure 5.4 (continued): (h) Estimation error of camera parameters.
Estimate position (Green line) vs. Real position (Red line)

(i)

Figure 5.4 (continued): (i) Estimation of the coordinates corrupted by noise using a KF.
5.2.4 Simulation Results Based on the Gaussian-Markov Kinematic Model

The simulation results based on the Gaussian-Markov kinematic model are illustrated in Figure 5.5.

Figure 5.5: Simulation result based on the Gaussian-Markov kinematic model. (a) Trajectory of the landmarks on the 2D image plane.
Figure 5.5: (b) Image error.
Trajectory of the landmarks in the 3D space

Figure 5.5 (continued): (c) Trajectory of the landmarks in the 3D space.
Position of the robot

\[ X \quad Y \quad Z \quad \phi \quad \theta \]

(d)

Figure 5.5 (continued): (d) Position of the robot.
Figure 5.5 (continued): (e) Joint trajectory of the robot.
Figure 5.5 (continued): (f) Position error of landmark #1.
Figure 5.5 (continued): (g) Position error of landmark #2.
Figure 5.5 (continued): (h) Estimation error of camera parameters.
Estimate position (Green line) vs. Real position (Red line)

\[ U_{m1}, \quad V_{m1} \] (pixel)
\[ U_{r1}, \quad V_{r1} \] (pixel)
\[ U_{m2}, \quad V_{m2} \] (pixel)
\[ U_{r2}, \quad V_{r2} \] (pixel)

Time (Sec.)

Figure 5.5 (continued): (i) Estimation of the coordinates corrupted by noise using a KF.
5.2.5 Simulation result analysis

Figures 5.3(a), 5.4(a), and 5.5(a) show the landmarks’ 2D trajectories on the image plane. Figures 5.3(b), 5.4(b), and 5.5(b) show the decay in the image feature errors over time, depicting that all five image features tend towards their desired values. Figures 5.3(c), 5.4(c), and 5.5(c) show the landmarks’ 3D trajectories in the physical space. In the simulation, the running time is 30 seconds which is fixed and longer than the settling time. Therefore, as can be seen in the figure, when the robot gets in vicinity of its desired position and orientation, there will be still some random movements due to noise in the system. Figures 5.3(d), 5.4(d), and 5.5(d) illustrate the five parameters that define the robot’s position in approaching the desired position. Figures 5.3(e), 5.4(e), and 5.3(e) illustrate the trajectory of each joint of the robot. The position error of landmark #1 approaches zero, as shown in Figures 5.3(f), 5.4(f), and 5.5(f). The position error of landmark #2 approaches zero, as shown in Figures 5.3(g), 5.4(g), and 5.5(g). Figures 5.3(h), 5.4(h), and 5.5(h) illustrate the online estimation of the camera parameters using the KF. Figures 4.2(i), 4.3(i), and 4.4(i) also show estimations of the image points corrupted by Gaussian noise using a KF.

The simulation results given in Figures 5.3-5.5 show that the proposed visual servoing system, based on three different kinematic models, can successfully bring the 5-DOF robot to its desired configuration while estimating camera parameters in all directions. The settling time of each kinematic model is listed in Table 5.2. From Table 5.2, one can find that the settling time of the Gaussian Markov model is the shortest.
Table 5.2: The settling time of each kinematic model (Tolerance 2%).

<table>
<thead>
<tr>
<th>Kinematic Model</th>
<th>Second Order Kinematic model</th>
<th>Third Order Kinematic model</th>
<th>Gaussian Markov Kinematic model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Settling time</td>
<td>15.64s</td>
<td>18.74s</td>
<td>15.07s</td>
</tr>
</tbody>
</table>

5.3 Sensitivity Test Analysis

The proposed visual servoing simulations are tested with three different workspace setups based on three different kinematic models. Among the three workspace setups, only the extrinsic parameter of the camera $d$ is changed. The values of $d$ chosen in the sensitivity tests are 3000mm, 4000mm, 5000mm. One should note that the robot is closer to the camera when the smaller value of $d$ is chosen. Other workspace parameters in the sensitivity test analysis were the same as the workspace setup used in the visual servoing simulation (see section 5.2.1). In each test, seven sets of simulation results were given based on seven different initial guesses of the intrinsic and extrinsic parameters of the camera. The seven initial guesses of the camera’s intrinsic and extrinsic parameters deviated from their real values by +15%, +10%, +5%, 0%, -5%, -10% and -15% respectively on average. The results are presented and discussed in the following section.

5.3.1 Sensitivity Test Results

The test results for both target locations, based on each configuration and each kinematic model, are given in Table 5.3-5.11. In each table, the errors of $X$, $Y$, $Z$, $\phi$ and $\theta$ which define the landmarks’ location in the workspace are reported. The error percentage of the camera’s parameters is also reported in those tables.
Table 5.3: Test result based on configuration I (d=5000) with the second order kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X^f - X^*$</td>
<td>(mm)</td>
<td>10.00</td>
<td>11.83</td>
<td>2.75</td>
<td>16.03</td>
<td>21.71</td>
<td>16.59</td>
</tr>
<tr>
<td>$Y^f - Y^*$</td>
<td>(mm)</td>
<td>2.13</td>
<td>1.29</td>
<td>2.74</td>
<td>2.59</td>
<td>5.20</td>
<td>2.01</td>
</tr>
<tr>
<td>$Z^f - Z^*$</td>
<td>(mm)</td>
<td>6.75</td>
<td>6.46</td>
<td>0.62</td>
<td>6.04</td>
<td>6.02</td>
<td>7.01</td>
</tr>
<tr>
<td>$\varphi^f - \varphi^*$</td>
<td>(rad)</td>
<td>0.15</td>
<td>0.16</td>
<td>0.11</td>
<td>0.09</td>
<td>0.17</td>
<td>0.07</td>
</tr>
<tr>
<td>$\theta^f - \theta^*$</td>
<td>(rad)</td>
<td>0.04</td>
<td>0.03</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td>100% $x_c^f - u_c^*$/</td>
<td>$u^*_c$</td>
<td>0.51%</td>
<td>0.17%</td>
<td>0.41%</td>
<td>0.21%</td>
<td>0.39%</td>
<td>0.26%</td>
</tr>
<tr>
<td>100% $v_c^f - v_c^<em>$/$v_c^</em>$</td>
<td>0.76%</td>
<td>0.62%</td>
<td>0.00%</td>
<td>0.03%</td>
<td>0.10%</td>
<td>0.29%</td>
<td>0.39%</td>
</tr>
<tr>
<td>100% $\alpha_u^f - \alpha_u^<em>$/$\alpha_u^</em>$</td>
<td>1.63%</td>
<td>2.74%</td>
<td>1.88%</td>
<td>1.92%</td>
<td>3.78%</td>
<td>0.92%</td>
<td>1.49%</td>
</tr>
<tr>
<td>100% $\alpha_r^f - \alpha_r^<em>$/$\alpha_r^</em>$</td>
<td>2.98%</td>
<td>3.01%</td>
<td>2.22%</td>
<td>1.72%</td>
<td>3.33%</td>
<td>0.48%</td>
<td>1.48%</td>
</tr>
<tr>
<td>100% $d^f - d^*$/</td>
<td>$d^*$</td>
<td>1.65%</td>
<td>1.84%</td>
<td>1.53%</td>
<td>1.21%</td>
<td>1.98%</td>
<td>0.46%</td>
</tr>
</tbody>
</table>

The superscript "f" defines the final value; the superscript "*" defines the desire or real value.

Table 5.4: Test result based on configuration I (d=5000) with the third order kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X^f - X^*$</td>
<td>(mm)</td>
<td>11.89</td>
<td>15.29</td>
<td>16.97</td>
<td>7.47</td>
<td>32.47</td>
<td>10.52</td>
</tr>
<tr>
<td>$Y^f - Y^*$</td>
<td>(mm)</td>
<td>1.07</td>
<td>2.45</td>
<td>2.37</td>
<td>2.41</td>
<td>8.45</td>
<td>4.60</td>
</tr>
<tr>
<td>$Z^f - Z^*$</td>
<td>(mm)</td>
<td>6.68</td>
<td>4.74</td>
<td>7.47</td>
<td>2.14</td>
<td>15.22</td>
<td>2.90</td>
</tr>
<tr>
<td>$\varphi^f - \varphi^*$</td>
<td>(rad)</td>
<td>0.01</td>
<td>0.05</td>
<td>0.29</td>
<td>0.14</td>
<td>0.11</td>
<td>0.07</td>
</tr>
<tr>
<td>$\theta^f - \theta^*$</td>
<td>(rad)</td>
<td>0.01</td>
<td>0.01</td>
<td>0.06</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
</tr>
<tr>
<td>100% $x_c^f - u_c^*$/</td>
<td>$u^*_c$</td>
<td>0.63%</td>
<td>0.09%</td>
<td>0.19%</td>
<td>0.04%</td>
<td>0.30%</td>
<td>0.38%</td>
</tr>
<tr>
<td>100% $v_c^f - v_c^<em>$/$v_c^</em>$</td>
<td>0.25%</td>
<td>0.35%</td>
<td>0.31%</td>
<td>0.05%</td>
<td>0.56%</td>
<td>0.42%</td>
<td>0.16%</td>
</tr>
<tr>
<td>100% $\alpha_u^f - \alpha_u^<em>$/$\alpha_u^</em>$</td>
<td>0.30%</td>
<td>1.39%</td>
<td>2.24%</td>
<td>1.43%</td>
<td>0.78%</td>
<td>2.64%</td>
<td>0.62%</td>
</tr>
<tr>
<td>100% $\alpha_r^f - \alpha_r^<em>$/$\alpha_r^</em>$</td>
<td>0.05%</td>
<td>1.07%</td>
<td>2.37%</td>
<td>1.55%</td>
<td>1.03%</td>
<td>2.40%</td>
<td>0.68%</td>
</tr>
<tr>
<td>100% $d^f - d^<em>$/$d^</em>$</td>
<td>0.16%</td>
<td>0.90%</td>
<td>1.37%</td>
<td>0.99%</td>
<td>0.57%</td>
<td>1.61%</td>
<td>0.26%</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 5.5: Test result based on configuration I (d=5000) with the Gaussian Markov kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X^f - X^*$ (mm)</td>
<td>10.96</td>
<td>13.37</td>
<td>9.83</td>
<td>8.48</td>
<td>2.65</td>
<td>6.91</td>
<td>5.83</td>
</tr>
<tr>
<td>$Y^f - Y^*$ (mm)</td>
<td>6.57</td>
<td>7.30</td>
<td>5.93</td>
<td>4.68</td>
<td>1.27</td>
<td>1.74</td>
<td>5.37</td>
</tr>
<tr>
<td>$Z^f - Z^*$ (mm)</td>
<td>11.29</td>
<td>13.56</td>
<td>9.09</td>
<td>8.66</td>
<td>3.09</td>
<td>8.49</td>
<td>4.63</td>
</tr>
<tr>
<td>$\phi^f - \phi^*$ (rad)</td>
<td>0.02</td>
<td>0.14</td>
<td>0.28</td>
<td>0.09</td>
<td>0.04</td>
<td>0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>$\theta^f - \theta^*$ (rad)</td>
<td>0.00</td>
<td>0.02</td>
<td>0.06</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>

$100\% \times \frac{|u_e^f - u_e^*|}{u_e^*}$ | 0.80% | 0.17% | 0.16%| 0.01%| 0.41%| 0.29%| 0.19% |
$100\% \times \frac{|v_c^f - v_c^*|}{v_c^*}$ | 0.09% | 0.37% | 0.17%| 0.28%| 0.49%| 0.01%| 0.35% |
$100\% \times \frac{|\alpha_u^f - \alpha_u^*|}{\alpha_u^*}$ | 0.45% | 0.43% | 1.43%| 0.52%| 0.16%| 0.40%| 0.42% |
$100\% \times \frac{|\alpha_v^f - \alpha_v^*|}{\alpha_v^*}$ | 0.09% | 0.22% | 1.52%| 0.46%| 0.08%| 0.23%| 0.48% |
$100\% \times \frac{|d^f - d^*|}{d^*}$ | 0.02% | 0.05% | 0.57%| 0.21%| 0.03%| 0.18%| 0.12% |

Table 5.6: Test result based on configuration II (d=4000) with the second order kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X^f - X^*$ (mm)</td>
<td>20.78</td>
<td>12.39</td>
<td>6.38</td>
<td>4.62</td>
<td>11.04</td>
<td>16.07</td>
<td>3.25</td>
</tr>
<tr>
<td>$Y^f - Y^*$ (mm)</td>
<td>5.85</td>
<td>2.10</td>
<td>1.76</td>
<td>0.31</td>
<td>1.23</td>
<td>5.98</td>
<td>0.62</td>
</tr>
<tr>
<td>$Z^f - Z^*$ (mm)</td>
<td>14.72</td>
<td>8.23</td>
<td>6.84</td>
<td>2.73</td>
<td>8.61</td>
<td>7.12</td>
<td>3.22</td>
</tr>
<tr>
<td>$\phi^f - \phi^*$ (rad)</td>
<td>0.12</td>
<td>0.04</td>
<td>0.09</td>
<td>0.01</td>
<td>0.07</td>
<td>0.03</td>
<td>0.13</td>
</tr>
<tr>
<td>$\theta^f - \theta^*$ (rad)</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.01</td>
<td>0.01</td>
<td>0.03</td>
<td>0.01</td>
</tr>
</tbody>
</table>

$100\% \times \frac{|u_c^f - u_c^*|}{u_c^*}$ | 0.56% | 0.22% | 0.48%| 0.23%| 0.52%| 0.17%| 0.19% |
$100\% \times \frac{|v_c^f - v_c^*|}{v_c^*}$ | 0.71% | 0.40% | 0.24%| 0.14%| 0.21%| 0.43%| 0.32% |
$100\% \times \frac{|\alpha_u^f - \alpha_u^*|}{\alpha_u^*}$ | 0.18% | 1.35% | 0.21%| 0.53%| 1.19%| 0.53%| 1.28% |
$100\% \times \frac{|\alpha_v^f - \alpha_v^*|}{\alpha_v^*}$ | 1.17% | 1.72% | 0.01%| 0.49%| 0.79%| 0.79%| 1.24% |
$100\% \times \frac{|d^f - d^*|}{d^*}$ | 0.49% | 0.88% | 0.11%| 0.29%| 0.26%| 0.44%| 0.78% |

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
### Table 5.7: Test result based on configuration II (d=4000) with the third order kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X^f - X^*$ (mm)</td>
<td>10.48</td>
<td>0.38</td>
<td>22.68</td>
<td>20.72</td>
<td>26.35</td>
<td>28.13</td>
<td>7.27</td>
</tr>
<tr>
<td>$Y^f - Y^*$ (mm)</td>
<td>2.72</td>
<td>0.32</td>
<td>6.18</td>
<td>5.29</td>
<td>9.21</td>
<td>11.82</td>
<td>4.52</td>
</tr>
<tr>
<td>$Z^f - Z^*$ (mm)</td>
<td>7.95</td>
<td>0.60</td>
<td>14.45</td>
<td>13.11</td>
<td>15.45</td>
<td>15.40</td>
<td>1.69</td>
</tr>
<tr>
<td>$\phi^f - \phi^*$ (rad)</td>
<td>0.03</td>
<td>0.03</td>
<td>0.08</td>
<td>0.16</td>
<td>0.07</td>
<td>0.01</td>
<td>0.09</td>
</tr>
<tr>
<td>$\theta^f - \theta^*$ (rad)</td>
<td>0.03</td>
<td>0.03</td>
<td>0.01</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>$100% \times \frac{u^f_c - u^<em>}{u^</em>_c}$</td>
<td>0.63%</td>
<td>0.00%</td>
<td>0.15%</td>
<td>0.19%</td>
<td>0.28%</td>
<td>0.25%</td>
<td>0.13%</td>
</tr>
<tr>
<td>$100% \times \frac{v^f_c - v^<em>}{v^</em>_c}$</td>
<td>0.09%</td>
<td>0.36%</td>
<td>0.42%</td>
<td>0.10%</td>
<td>0.48%</td>
<td>0.31%</td>
<td>0.34%</td>
</tr>
<tr>
<td>$100% \times \frac{\alpha_u^f - \alpha_u^<em>}{\alpha_u^</em>}$</td>
<td>0.88%</td>
<td>0.98%</td>
<td>2.58%</td>
<td>1.87%</td>
<td>0.61%</td>
<td>1.39%</td>
<td>0.05%</td>
</tr>
<tr>
<td>$100% \times \frac{d^f - d^<em>}{d^</em>}$</td>
<td>0.23%</td>
<td>0.48%</td>
<td>1.40%</td>
<td>1.01%</td>
<td>0.36%</td>
<td>0.85%</td>
<td>0.10%</td>
</tr>
</tbody>
</table>

### Table 5.8: Test result based on configuration II (d=4000) with the Gaussian Markov kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X^f - X^*$ (mm)</td>
<td>13.92</td>
<td>14.16</td>
<td>1.19</td>
<td>5.10</td>
<td>4.38</td>
<td>7.64</td>
<td>13.15</td>
</tr>
<tr>
<td>$Y^f - Y^*$ (mm)</td>
<td>6.53</td>
<td>4.10</td>
<td>2.99</td>
<td>3.42</td>
<td>1.89</td>
<td>2.92</td>
<td>2.83</td>
</tr>
<tr>
<td>$Z^f - Z^*$ (mm)</td>
<td>6.49</td>
<td>6.92</td>
<td>2.73</td>
<td>2.40</td>
<td>1.62</td>
<td>4.10</td>
<td>8.33</td>
</tr>
<tr>
<td>$\phi^f - \phi^*$ (rad)</td>
<td>0.03</td>
<td>0.04</td>
<td>0.03</td>
<td>0.07</td>
<td>0.18</td>
<td>0.04</td>
<td>0.06</td>
</tr>
<tr>
<td>$\theta^f - \theta^*$ (rad)</td>
<td>0.03</td>
<td>0.02</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>$100% \times \frac{u^f_c - u^<em>}{u^</em>_c}$</td>
<td>0.50%</td>
<td>0.05%</td>
<td>0.13%</td>
<td>0.02%</td>
<td>0.44%</td>
<td>0.39%</td>
<td>0.08%</td>
</tr>
<tr>
<td>$100% \times \frac{v^f_c - v^<em>}{v^</em>_c}$</td>
<td>0.09%</td>
<td>0.26%</td>
<td>0.37%</td>
<td>0.24%</td>
<td>0.37%</td>
<td>0.35%</td>
<td>0.22%</td>
</tr>
<tr>
<td>$100% \times \frac{\alpha_u^f - \alpha_u^<em>}{\alpha_u^</em>}$</td>
<td>0.96%</td>
<td>0.48%</td>
<td>2.90%</td>
<td>0.14%</td>
<td>1.69%</td>
<td>1.40%</td>
<td>0.07%</td>
</tr>
<tr>
<td>$100% \times \frac{d^f - d^<em>}{d^</em>}$</td>
<td>0.50%</td>
<td>0.47%</td>
<td>2.96%</td>
<td>1.46%</td>
<td>0.01%</td>
<td>0.16%</td>
<td>0.06%</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 5.9: Test result based on configuration III (d=3000) with the second order kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X^f - X^*) (mm)</td>
<td>10.89</td>
<td>10.80</td>
<td>15.05</td>
<td>12.35</td>
<td>0.34</td>
<td>10.26</td>
<td></td>
</tr>
<tr>
<td>(Y^f - Y^*) (mm)</td>
<td>5.58</td>
<td>4.42</td>
<td>9.87</td>
<td>7.59</td>
<td>1.04</td>
<td>6.61</td>
<td></td>
</tr>
<tr>
<td>(Z^f - Z^*) (mm)</td>
<td>11.66</td>
<td>11.88</td>
<td>14.00</td>
<td>12.99</td>
<td>1.17</td>
<td>10.09</td>
<td></td>
</tr>
<tr>
<td>(\varphi^f - \varphi^*) (rad)</td>
<td>0.02</td>
<td>0.05</td>
<td>0.00</td>
<td>0.03</td>
<td>0.01</td>
<td>0.08</td>
<td>0.11</td>
</tr>
<tr>
<td>(\theta^f - \theta^*) (rad)</td>
<td>0.04</td>
<td>0.01</td>
<td>0.00</td>
<td>0.02</td>
<td>0.02</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>(100% \times \frac{u_c^f - u_c^<em>}{u_c^</em>})</td>
<td>0.64%</td>
<td>0.29%</td>
<td>0.66%</td>
<td>0.37%</td>
<td>0.52%</td>
<td>0.07%</td>
<td>0.38%</td>
</tr>
<tr>
<td>(100% \times \frac{v_c^f - v_c^<em>}{v_c^</em>})</td>
<td>1.01%</td>
<td>0.57%</td>
<td>0.40%</td>
<td>0.28%</td>
<td>0.27%</td>
<td>0.45%</td>
<td>0.11%</td>
</tr>
<tr>
<td>(100% \times \frac{\alpha_u^f - \alpha_u^<em>}{\alpha_u^</em>})</td>
<td>0.17%</td>
<td>0.42%</td>
<td>0.70%</td>
<td>0.15%</td>
<td>0.21%</td>
<td>0.22%</td>
<td>0.32%</td>
</tr>
<tr>
<td>(100% \times \frac{\alpha_v^f - \alpha_v^<em>}{\alpha_v^</em>})</td>
<td>0.88%</td>
<td>0.68%</td>
<td>0.50%</td>
<td>0.07%</td>
<td>0.01%</td>
<td>0.47%</td>
<td>0.39%</td>
</tr>
<tr>
<td>(100% \times \frac{d^f - d^<em>}{d^</em>})</td>
<td>0.26%</td>
<td>0.25%</td>
<td>0.14%</td>
<td>0.01%</td>
<td>0.09%</td>
<td>0.22%</td>
<td>0.23%</td>
</tr>
</tbody>
</table>

Table 5.10: Test result based on configuration III (d=3000) with the third order kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>(X^f - X^*) (mm)</td>
<td>10.96</td>
<td>13.37</td>
<td>9.83</td>
<td>8.48</td>
<td>2.65</td>
<td>6.91</td>
<td>5.83</td>
</tr>
<tr>
<td>(Y^f - Y^*) (mm)</td>
<td>6.57</td>
<td>7.30</td>
<td>5.93</td>
<td>4.68</td>
<td>1.27</td>
<td>1.74</td>
<td>5.37</td>
</tr>
<tr>
<td>(Z^f - Z^*) (mm)</td>
<td>11.29</td>
<td>13.56</td>
<td>9.09</td>
<td>8.66</td>
<td>3.09</td>
<td>8.49</td>
<td>4.63</td>
</tr>
<tr>
<td>(\varphi^f - \varphi^*) (rad)</td>
<td>0.02</td>
<td>0.14</td>
<td>0.28</td>
<td>0.09</td>
<td>0.04</td>
<td>0.04</td>
<td>0.02</td>
</tr>
<tr>
<td>(\theta^f - \theta^*) (rad)</td>
<td>0.00</td>
<td>0.02</td>
<td>0.06</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>(100% \times \frac{u_c^f - u_c^<em>}{u_c^</em>})</td>
<td>0.80%</td>
<td>0.17%</td>
<td>0.16%</td>
<td>0.01%</td>
<td>0.41%</td>
<td>0.29%</td>
<td>0.19%</td>
</tr>
<tr>
<td>(100% \times \frac{v_c^f - v_c^<em>}{v_c^</em>})</td>
<td>0.09%</td>
<td>0.37%</td>
<td>0.17%</td>
<td>0.28%</td>
<td>0.49%</td>
<td>0.01%</td>
<td>0.35%</td>
</tr>
<tr>
<td>(100% \times \frac{\alpha_u^f - \alpha_u^<em>}{\alpha_u^</em>})</td>
<td>0.45%</td>
<td>0.43%</td>
<td>1.43%</td>
<td>0.52%</td>
<td>0.16%</td>
<td>0.40%</td>
<td>0.42%</td>
</tr>
<tr>
<td>(100% \times \frac{\alpha_v^f - \alpha_v^<em>}{\alpha_v^</em>})</td>
<td>0.09%</td>
<td>0.22%</td>
<td>1.52%</td>
<td>0.46%</td>
<td>0.08%</td>
<td>0.23%</td>
<td>0.48%</td>
</tr>
<tr>
<td>(100% \times \frac{d^f - d^<em>}{d^</em>})</td>
<td>0.02%</td>
<td>0.05%</td>
<td>0.57%</td>
<td>0.21%</td>
<td>0.03%</td>
<td>0.18%</td>
<td>0.12%</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 5.11: Test result based on configuration III (d=3000) with the Gaussian Markov kinematic model.

<table>
<thead>
<tr>
<th></th>
<th>+15%</th>
<th>+10%</th>
<th>+5%</th>
<th>0%</th>
<th>-5%</th>
<th>-10%</th>
<th>-15%</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X^f - X^*$ (mm)</td>
<td>16.74</td>
<td>11.73</td>
<td>10.05</td>
<td>11.09</td>
<td>11.12</td>
<td>6.73</td>
<td>14.15</td>
</tr>
<tr>
<td>$Y^f - Y^*$ (mm)</td>
<td>10.18</td>
<td>6.57</td>
<td>8.34</td>
<td>8.26</td>
<td>5.83</td>
<td>3.90</td>
<td>6.98</td>
</tr>
<tr>
<td>$Z^f - Z^*$ (mm)</td>
<td>17.50</td>
<td>12.16</td>
<td>7.45</td>
<td>9.73</td>
<td>11.80</td>
<td>5.83</td>
<td>14.64</td>
</tr>
<tr>
<td>$\phi^f - \phi^*$ (rad)</td>
<td>0.09</td>
<td>0.07</td>
<td>0.05</td>
<td>0.04</td>
<td>0.01</td>
<td>0.14</td>
<td>0.05</td>
</tr>
<tr>
<td>$\theta^f - \theta^*$ (rad)</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.02</td>
<td>0.00</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>$100% \times \frac{u^f - u^<em>}{u^</em>}$</td>
<td>0.80%</td>
<td>0.01%</td>
<td>0.03%</td>
<td>0.16%</td>
<td>0.35%</td>
<td>0.26%</td>
<td>0.18%</td>
</tr>
<tr>
<td>$100% \times \frac{v^f - v^<em>}{v^</em>}$</td>
<td>0.08%</td>
<td>0.36%</td>
<td>0.21%</td>
<td>0.01%</td>
<td>0.58%</td>
<td>0.06%</td>
<td>0.15%</td>
</tr>
<tr>
<td>$100% \times \frac{\alpha_u^f - \alpha_u^<em>}{\alpha_u^</em>}$</td>
<td>0.37%</td>
<td>0.04%</td>
<td>1.34%</td>
<td>0.31%</td>
<td>0.16%</td>
<td>0.55%</td>
<td>0.56%</td>
</tr>
<tr>
<td>$100% \times \frac{\alpha_v^f - \alpha_v^<em>}{\alpha_v^</em>}$</td>
<td>0.12%</td>
<td>0.11%</td>
<td>1.32%</td>
<td>0.27%</td>
<td>0.03%</td>
<td>0.39%</td>
<td>0.52%</td>
</tr>
<tr>
<td>$100% \times \frac{d^f - d^<em>}{d^</em>}$</td>
<td>0.03%</td>
<td>0.10%</td>
<td>0.49%</td>
<td>0.10%</td>
<td>0.08%</td>
<td>0.24%</td>
<td>0.16%</td>
</tr>
</tbody>
</table>
### Table 5.12: Statistical results based on configuration I (d=5000).

<table>
<thead>
<tr>
<th>Kinematic Model</th>
<th>MEAN</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ΔX (mm)</td>
<td>ΔY (mm)</td>
</tr>
<tr>
<td>Second order</td>
<td>14.22</td>
<td>2.86</td>
</tr>
<tr>
<td>Third order</td>
<td>14.53</td>
<td>3.06</td>
</tr>
<tr>
<td>Gaussian Markov</td>
<td>8.29</td>
<td>4.69</td>
</tr>
</tbody>
</table>

### Table 5.13: Statistical results based on configuration II (d=4000).

<table>
<thead>
<tr>
<th>Kinematic Model</th>
<th>MEAN</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ΔX (mm)</td>
<td>ΔY (mm)</td>
</tr>
<tr>
<td>Second order</td>
<td>10.65</td>
<td>2.55</td>
</tr>
<tr>
<td>Third order</td>
<td>16.57</td>
<td>5.73</td>
</tr>
<tr>
<td>Gaussian Markov</td>
<td>8.51</td>
<td>3.52</td>
</tr>
</tbody>
</table>

### Table 5.14: Statistical results based on configuration III (d=3000).

<table>
<thead>
<tr>
<th>Kinematic Model</th>
<th>MEAN</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ΔX (mm)</td>
<td>ΔY (mm)</td>
</tr>
<tr>
<td>Second order</td>
<td>8.71</td>
<td>5.03</td>
</tr>
<tr>
<td>Third order</td>
<td>8.29</td>
<td>4.69</td>
</tr>
<tr>
<td>Gaussian Markov</td>
<td>11.66</td>
<td>7.15</td>
</tr>
</tbody>
</table>
5.3.2 Discussion on the Results of the Sensitivity Test

Data from the test results (Tables 5.3-5.11) were used to calculate the statistical moments such as mean and Standard Deviation (STD). The statistical parameters based upon three configurations are given in Tables 5.12-5.14. In the first and second configurations, the Gaussian Markov kinematic model yields the lowest mean error and STD values. In the third configuration, the Gaussian-Markov kinematic model yields the highest mean error and STD values. However, the position and orientation error still remain within the acceptable range. The error percentage of all the estimated parameters of the camera is acceptable as well.

5.4 Summary

The sensitivity of the proposed visual servoing simulation was determined by comparing the results associated with three different configurations and seven different initial guesses of the camera’s parameters. The relative distance between the robot and camera did not play a critical role in the overall performance of the system.
6. Conclusions and Future Work

A novel visual servoing has been presented for 3D positioning and orienting a robotic arm fixed on a linear slide using one camera and a spherical mirror. Image-based visual servoing was employed to position and orient two landmarks mounted on the robot's end-effector in 3D space. Pose information, obtained by viewing landmarks and their mirror reflections, was utilized for servoing purpose. It was assumed that the robot and its reflection on the spherical mirror remain in the camera's field of view all the time. A novel set of image feature points that yielded a full-rank image Jacobian was also introduced. This could further simplify the operation of the image extraction unit. By making use of the proposed image features, the minimum information would be required to distinguish between the landmarks and their mirror reflections in the 2D image captured by the camera for tracking purpose.

To implement the proposed approach, a dual Kalman filter was introduced. One filter was used to estimate the intrinsic and extrinsic parameters of the camera (therefore there was no need for offline camera calibration). Another filter was used to track pixel coordinates of the landmarks and their mirror reflections projected on to the image plane. Three different kinematics models, namely a second order, a third order, and a Gaussian-Markov were simulated. The performance of the proposed visual servoing structure based on this dual Kalman filter was presented through numerous computer simulations. It was shown that the proposed method can position and orient the robot in all directions repeatedly with high precision.
A number of simulations based on different configurations of the robot, the mirror and the camera and different initial guesses of the camera's parameters were also carried out to do sensitivity analysis. A comparison among three different kinematic models was presented and discussed.

Implementation of the proposal method in real-world scenarios will be carried out in the future. In the real-world scenarios, robot's dynamics and image processing time need to be carefully considered. Utilization of a centralized Catadioptric vision system is also under investigation. Employing a 2.5D visual servoing strategy to overcome singularities in robot's motion and that in image Jacobian will be investigated in the future.
REFERENCES


