

VISUAL SERVO TRACKING CONTROL VIA A LYAPUNOV-BASED
APPROACH

By
GUOQIANG HU

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2007

©2007 Guoqiang Hu

To my mother Huanyin Zhang, my father Huade Hu, and my wife Yi Feng for their
endless love and support

ACKNOWLEDGMENTS

Thank you to my advisor, Dr. Warren Dixon, for his guidance and encouragement which will benefit me a life time. As an advisor, he kept polishing my methodology and skills in resolving problems and formulating new problems. As a mentor, he helped me develop professional skills and gave me opportunities to get exposed to professional working environment.

Thanks to my committee members Dr. Thomas Burks, Dr. Carl Crane III, Dr. Seth Hutchinson, and Dr. Rick Lind, for the time and help that they provided.

Thanks to Dr. Nick Gans and Sid Mehta for all the insightful discussions. Thanks to Dr. Joe Kehoe for his suggestions in formatting my defense presentation and dissertation. Finally, thanks to my NCR lab fellows for their friendship during the past three years of joy.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
ABSTRACT	xiii
CHAPTER	
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Problem Statement	2
1.3 Literature Review	8
1.3.1 Basic Visual Servo Control Approaches	8
1.3.2 Visual Servo Control Approaches to Enlarge the FOV	9
1.3.3 Robust and Adaptive Visual Servo Control	11
1.4 Contributions	13
2 BACKGROUND AND PRELIMINARY DEVELOPMENT	16
2.1 Geometric Model	16
2.2 Euclidean Reconstruction	19
2.3 Unit Quaternion Representation of the Rotation Matrix	21
3 LYAPUNOV-BASED VISUAL SERVO TRACKING CONTROL VIA A QUATERNION FORMULATION	25
3.1 Introduction	25
3.2 Control Objective	26
3.3 Control Development	28
3.3.1 Open-Loop Error System	28
3.3.2 Closed-Loop Error System	30
3.3.3 Stability Analysis	32
3.4 Camera-To-Hand Extension	33
3.4.1 Model Development	33
3.4.2 Control Formulation	35
3.5 Simulation Results	38
3.6 Experiment Results	41

	3.6.1 Experiment Configurations	41
	3.6.2 Experiment for Tracking	45
	3.6.3 Experiment for Regulation	47
4	COLLABORATIVE VISUAL SERVO TRACKING CONTROL VIA A DAISY-CHAINING APPROACH	61
	4.1 Introduction	61
	4.2 Problem Scenario	62
	4.3 Geometric Model	64
	4.4 Euclidean Reconstruction	68
	4.5 Control Objective	71
	4.6 Control Development	73
	4.6.1 Open-Loop Error System	73
	4.6.2 Closed-Loop Error System	74
	4.6.3 Stability Analysis	75
	4.7 Simulation Results	76
5	ADAPTIVE VISUAL SERVO TRACKING CONTROL USING A CEN- TRAL CATADIOPTRIC CAMERA	84
	5.1 Introduction	84
	5.2 Geometric Model	85
	5.3 Euclidean Reconstruction	90
	5.4 Control Objective	91
	5.5 Control Development	94
	5.5.1 Open-Loop Error System	94
	5.5.2 Closed-Loop Error System	95
	5.5.3 Stability Analysis	96
6	VISUAL SERVO CONTROL IN THE PRESENCE OF CAMERA CAL- IBRATION ERROR	98
	6.1 Introduction	98
	6.2 Feedback Control Measurements	99
	6.3 Control Objective	101
	6.4 Quaternion Estimation	103
	6.4.1 Estimate Development	103
	6.4.2 Estimate Relationships	104
	6.5 Control Development	106
	6.5.1 Rotation Control	106
	6.5.2 Translation Control	106
	6.6 Stability Analysis	107
	6.7 Simulation Results	110

7	COMBINED ROBUST AND ADAPTIVE HOMOGRAPHY-BASED VISUAL SERVO CONTROL VIA AN UNCALIBRATED CAMERA . . .	117
7.1	Introduction	117
7.2	Camera Geometry and Assumptions	118
7.3	Open-Loop Error System	120
7.3.1	Rotation Error System	120
7.3.2	Translation Error System	121
7.4	Control Development	123
7.4.1	Rotation Control Development and Stability Analysis	123
7.4.2	Translation Control Development and Stability Analysis . . .	125
7.5	Simulation Results	128
8	CONCLUSIONS	135
APPENDIX		
A	UNIT NORM PROPERTY FOR THE QUATERNION ERROR	138
B	ONE PROPERTY OF UNIT QUATERNIONS	140
C	OPEN-LOOP TRANSLATION ERROR SYSTEM	141
D	PROPERTY ON MATRIX NORM	143
E	COMPUTATION OF DEPTH RATIOS	144
F	INEQUALITY DEVELOPMENT	147
G	LINEAR PARAMETERIZATION OF TRANSLATION ERROR SYS- TEM	148
REFERENCES		149
BIOGRAPHICAL SKETCH		157

LIST OF TABLES

<u>Table</u>		<u>page</u>
4-1	Coordinate frames relationships	65

LIST OF FIGURES

Figure	page
2-1 Coordinate frame relationships between a camera viewing a planar patch at different spatiotemporal instances. The coordinate frames \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d are attached to the current, reference and desired locations, respectively.	17
2-2 Coordinate frame relationships between a camera viewing a planar patch at different spatiotemporal instances. The coordinate frames \mathcal{F} and \mathcal{F}^* are attached to the current and reference locations, respectively.	18
3-1 Coordinate frame relationships between a fixed camera and the planes defined by the current, desired, and reference feature points (i.e., π , π_d , and π^*).	33
3-2 Block diagram of the experiment.	42
3-3 The Sony XCD-710CR color firewire camera pointed at the virtual environment.	42
3-4 Virtual reality environment example: a virtual recreation of the US Army's urban warfare training ground at Fort Benning.	44
3-5 Desired image-space coordinates of the four feature points (i.e., $p_d(t)$) in the tracking Matlab simulation shown in a 3D graph. In the figure, "O" denotes the initial image-space positions of the 4 feature points in the desired trajectory, and "*" denotes the corresponding final positions of the feature points.	48
3-6 Current image-space coordinates of the four feature points (i.e., $p_d(t)$) in the tracking Matlab simulation shown in a 3D graph. In the figure, "O" denotes the initial image-space positions of the 4 feature points, and "*" denotes the corresponding final positions of the feature points.	48
3-7 Translation error $e(t)$ in the tracking Matlab simulation.	49
3-8 Rotation quaternion error $\tilde{q}(t)$ in the tracking Matlab simulation.	49
3-9 Pixel coordinate $p_d(t)$ of the four feature points in a sequence of desired images in the tracking Matlab simulation. The upper figure is for the $u_d(t)$ component and the bottom figure is for the $v_d(t)$ component.	50

3-10	Pixel coordinate $p(t)$ of the current pose of the four feature points in the tracking Matlab simulation. The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.	50
3-11	Tracking error $p(t) - p_d(t)$ (in pixels) of the four feature points in the tracking Matlab simulation. The upper figure is for the $u(t) - u_d(t)$ component and the bottom figure is for the $v(t) - v_d(t)$ component.	51
3-12	Linear camera velocity input $v_c(t)$ in the tracking Matlab simulation. . .	51
3-13	Angular camera velocity input $w_c(t)$ in the tracking Matlab simulation. .	52
3-14	Adaptive on-line estimate of z_1^* in the tracking Matlab simulation.	52
3-15	Translation error $e(t)$ in the tracking experiment.	53
3-16	Rotation quaternion error $\tilde{q}(t)$ in the tracking experiment.	53
3-17	Pixel coordinate $p_d(t)$ of the four feature points in a sequence of desired images in the tracking experiment. The upper figure is for the $u_d(t)$ component and the bottom figure is for the $v_d(t)$ component.	54
3-18	Pixel coordinate $p(t)$ of the current pose of the four feature points in the tracking experiment. The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.	54
3-19	Tracking error $p(t) - p_d(t)$ (in pixels) of the four feature points in the tracking experiment. The upper figure is for the $u(t) - u_d(t)$ component and the bottom figure is for the $v(t) - v_d(t)$ component.	55
3-20	Linear camera velocity input $v_c(t)$ in the tracking experiment.	55
3-21	Angular camera velocity input $w_c(t)$ in the tracking experiment.	56
3-22	Adaptive on-line estimate of z_1^* in the tracking experiment.	56
3-23	Translation error $e(t)$ in the regulation experiment.	57
3-24	Rotation quaternion error $q(t)$ in the regulation experiment.	57
3-25	Pixel coordinate $p(t)$ (in pixels) of the current pose of the four feature points in the regulation experiment. The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.	58
3-26	Regulation error $p(t) - p^*$ (in pixels) of the four feature points in the regulation experiment. The upper figure is for the $u(t) - u^*(t)$ component and the bottom figure is for the $v(t) - v^*(t)$ component.	58
3-27	Linear camera velocity input $v_c(t)$ in the regulation experiment.	59
3-28	Angular camera velocity input $w_c(t)$ in the regulation experiment.	59

3-29	Adaptive on-line estimate of z_1^* in the regulation experiment.	60
4-1	Geometric model.	63
4-2	This figure shows the initial positions of the cameras and the feature point planes. The initial positions of the cameras attached to \mathcal{I} and \mathcal{I}_R are denoted by “O”. The feature points on the planes π , π^* and π_d are denoted by “.”. The origins of the coordinate frames \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d are denoted by “*”.	79
4-3	Pixel coordinate $p_{rd}(t)$ of the four feature points on the plane π_d in a sequence of desired images taken by the camera attached to \mathcal{I}_R . The upper figure is for the $u_{rd}(t)$ component and the bottom figure is for the $v_{rd}(t)$ component.	80
4-4	Pixel coordinate $p^*(t)$ of the four feature points on the plane π^* in a sequence of reference images taken by the moving camera attached to \mathcal{I} . The upper figure is for the $u^*(t)$ component and the bottom figure is for the $v^*(t)$ component.	80
4-5	Pixel coordinate $p(t)$ of the four feature points on the plane π in a sequence of images taken by the moving camera attached to \mathcal{I} . The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.	81
4-6	Translation error $e(t)$	81
4-7	Rotation quaternion error $\tilde{q}(t)$	82
4-8	Linear velocity input $v_c(t)$ for the planar patch π	82
4-9	Angular velocity input $w_c(t)$ for the planar patch π	83
5-1	Central catadioptric projection relationship.	85
5-2	Projection model of the central catadioptric camera.	86
5-3	Camera relationships represented in homography.	88
6-1	Unitless translation error between $m_1(t)$ and m_1^*	113
6-2	Quaternion rotation error.	114
6-3	Quaternion rotation error for comparison with different sign.	114
6-4	Image-space error in pixels between $p_i(t)$ and p_i^* . In the figure, “O” denotes the initial positions of the 4 feature points in the image, and “*” denotes the corresponding final positions of the feature points.	115

6-5	Image-space error in pixels between $p_i(t)$ and p_i^* shown in a 3D graph. In the figure, “O” denotes the initial positions of the 4 feature points in the image, and “*” denotes the corresponding final positions of the feature points.	115
6-6	Linear camera velocity control input.	116
6-7	Angular camera velocity control input.	116
7-1	Unitless translation error $e(t)$	131
7-2	Quaternion rotation error $q(t)$	131
7-3	Pixel coordinate $p(t)$ (in pixels) of the current pose of the four feature points in the simulation. The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.	132
7-4	Regulation error $p(t) - p^*$ (in pixels) of the four feature points in the simulation. The upper figure is for the $u(t) - u^*(t)$ component and the bottom figure is for the $v(t) - v^*(t)$ component.	132
7-5	Image-space error in pixels between $p(t)$ and p^* . In the figure, “O” denotes the initial positions of the 4 feature points in the image, and “*” denotes the corresponding final positions of the feature points.	133
7-6	Image-space error in pixels between $p(t)$ and p^* shown in a 3D graph. In the figure, “O” denotes the initial positions of the 4 feature points in the image, and “*” denotes the corresponding final positions of the feature points.	133
7-7	Linear camera velocity control input $v_c(t)$	134
7-8	Angular camera velocity control input $\omega_c(t)$	134

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

VISUAL SERVO TRACKING CONTROL VIA A LYAPUNOV-BASED
APPROACH

By

Guoqiang Hu

December 2007

Chair: Dr. Warren E. Dixon
Major: Mechanical Engineering

Recent advances in image processing, computational technology and control theory are enabling visual servo control to become more prevalent in robotics and autonomous systems applications. In this dissertation, visual servo control algorithms and architectures are developed that exploit the visual feedback from a camera system to achieve a tracking or regulation control objective for a rigid-body object (e.g., the end-effector of a robot manipulator, a satellite, an autonomous vehicle) identified by a patch of feature points.

The first two chapters present the introduction and background information for this dissertation. In the third chapter, a new visual servo tracking control method for a rigid-body object is developed by exploiting a combination of homography techniques, a quaternion parameterization, adaptive control techniques, and nonlinear Lyapunov-based control methods. The desired trajectory to be tracked is represented by a sequence of images (e.g., a video), which can be taken online or offline by a camera. This controller is singularity-free by using the homography techniques and the quaternion parameterization. In the fourth chapter, a new collaborative visual servo control method is developed to enable a rigid-body

object to track a desired trajectory. In contrast to typical camera-to-hand and camera-in-hand visual servo control configurations, the proposed controller is developed using a moving on-board camera viewing a moving object to obtain feedback signals. This collaborative method weakens the field-of-view restriction and enables the control object to perform large area motion. In the fifth chapter, a visual servo controller is developed that yields an asymptotic tracking result for the completely nonlinear camera-in-hand central catadioptric camera system. A panoramic field-of-view is obtained by using the central catadioptric camera. In the sixth chapter, a robust visual servo control method is developed to achieve a regulation control objective in presence of intrinsic camera calibration uncertainties. A quaternion-based estimate for the rotation error signal is developed and used in the controller development. The similarity relationship between the estimated and actual rotation matrices is used to construct the relationship between the estimated and actual quaternions. A Lyapunov-based stability analysis is provided that indicates a unique controller can be developed to achieve the regulation result despite a sign ambiguity in the developed quaternion estimate. In the seventh chapter, a new combined robust and adaptive visual servo control method is developed to asymptotically regulate the feature points in an image to the desired locations while also regulating the pose of the control object without calibrating the camera. These dual objectives are achieved by using a homography-based approach that exploits both image-space and reconstructed Euclidean information in the feedback loop. The robust rotation controller accommodates for the time-varying uncertainties in the rotation error system, and the adaptive translation controller compensates for the unknown calibration parameters in the translation error system. Chapter 8 serves as the conclusions of this dissertation.

CHAPTER 1 INTRODUCTION

1.1 Motivation

Control systems that use information acquired from an imaging source in the feedback loop are defined as visual servo control systems. Visual servo control has developed into a large subset of robotics literature (see [1–4] for a review) because of the enabling capabilities it can provide for autonomy. Recent advances in image processing, computational technology and control theory are enabling visual servo control to become more prevalent in autonomous systems applications (e.g., the autonomous ground vehicles grand challenge and urban challenge sponsored by the U.S. Defense Advanced Research Projects Agency (DARPA)). Instead of relying solely on a global positioning system (GPS) or inertial measurement units (IMU) for navigation and control, image-based methods are a promising approach to provide autonomous vehicles with position and orientation (i.e., pose) information. Specifically, rather than obtain an inertial measurement of an autonomous system, vision systems can be used to recast the navigation and control problem in terms of the image space. In addition to providing feedback relating the local pose of the camera with respect to some target, an image sensor can also be used to relate local sensor information to an inertial reference frame for global control tasks.

Visual servoing requires multidisciplinary expertise to integrate a vision system with the controller for tasks including: selecting the proper imaging hardware; extracting and processing images at rates amenable to closed-loop control; image analysis and feature point extraction/tracking; and recovering/estimating necessary state information from an image, etc. While each of the aforementioned tasks are active topics of research interest in computer vision and image processing

societies, they will not be the focus of this dissertation. The development in this dissertation is based on the assumption that images can be acquired, analyzed, and the resulting data can be provided to the controller without restricting the control rates.

The use of image-based feedback adds complexity and new challenges for the control system design. The scope of this dissertation is focused on issues associated with using reconstructed and estimated state information from a sequence of images to develop a stable closed-loop error system. Particularly, this dissertation focuses on the following problems: 1) how to design a visual servo tracking controller that achieves asymptotic tracking via a quaternion formulation? 2) how to design a collaborative visual servo control scheme when both the camera and the control object are moving? 3) how to design a visual servo controller using a central catadioptric camera? 4) and how to design a visual servo controller that is robust to camera calibration uncertainty?

1.2 Problem Statement

In this dissertation, visual servo control algorithms and architectures are developed that exploit the visual feedback from a camera system to achieve a tracking or regulation control objective for a six degrees of freedom (DOF) rigid-body control object (e.g., the end-effector of a robot manipulator, a satellite, an autonomous vehicle) identified by a patch of feature points. The tracking control objective is for the control object to track a desired trajectory that is encoded by a video obtained from a camera in either the camera-in-hand or camera-to-hand configuration. This video can be taken online or offline by a camera. For example, the motion of a control object can be prerecorded by a camera (for the camera-to-hand configuration) beforehand and used as a desired trajectory, or, a video of the reference object can be prerecorded as a desired trajectory while the camera moves (for the camera-in-hand configuration). The regulation control objective is

for the object to go to a desired pose that is encoded by a prerecorded image. The regulation problem can be considered as a particular case of the tracking problem. For example, when all the images in the sequence of desired images are identical, the tracking problem becomes a regulation problem. The dissertation will address the following problems of interest: 1) visual servo tracking control via a quaternion formulation; 2) collaborative visual servo tracking control using a daisy-chaining approach; 3) visual servo tracking control using a central catadioptric camera; 4) robust visual servo control in presence of camera calibration uncertainty; and 5) combined robust and adaptive visual servo control via an uncalibrated camera. The control development in the dissertation is proven by using nonlinear Lyapunov-based methods and is demonstrated by Matlab simulation and/or experimental results.

1) *Visual servo tracking control via a quaternion formulation.*

Much of the previous visual servo controllers have only been designed to address the regulation problem. Motivated by the need for new advancements to meet visual servo tracking applications, previous research has concentrated on developing different types of path planning techniques [5–9]. Recently, Chen et al. [10] provided a new formulation of the tracking control problem. A homography-based adaptive visual servo controller is developed to enable a robot end-effector to track a prerecorded time-varying reference trajectory determined by a sequence of images. The Euler angle-axis representation is used to represent the rotation error system. Due to the computational singularity limitation of the angle axis extraction algorithm (see Spong and Vidyasagar [11]), rotation angles of $\pm\pi$ were not considered. Motivated by the desire to avoid the rotation singularity completely, an error system and visual servo tracking controller is developed in Chapter 3 based on the quaternion formulation. A homography is constructed from image pairs and decomposed via textbook methods (e.g., Faugeras [12] and

Hartley and Zisserman [13]) to determine the rotation matrix. Once the rotation matrix has been determined, the corresponding unit quaternion can be obtained by numerically robust algorithms (see Hu et al. [14] and Shuster [15]). Then an error system is constructed in terms of the unit quaternion, which is void of singularities. An adaptive controller is then developed and proven to make a camera track a desired trajectory that is determined from a sequence of images. The controller contains an adaptive feedforward term to compensate for the unknown distance from the camera to the observed planar patch. A quaternion-based Lyapunov function is developed to facilitate the control design and the stability analysis.

2) *Collaborative visual servo tracking control using a daisy-chaining approach.*

Unlike typical visual servo controllers in camera-in-hand and camera-to-hand configurations, a unique aspect of the development for this problem is that a moving camera (e.g., a camera mounted on an unmanned air vehicle) is used to provide visual feedback to a moving autonomous vehicle. The control objective is for the autonomous vehicle (identified by a planar patch of feature points) to track the pose of a desired vehicle trajectory that is encoded by a prerecorded video obtained from a fixed camera (e.g., a camera mounted on a satellite, a camera mounted on a building). Several challenges must be resolved to achieve this unexplored control objective. The relative velocity between the moving feature point patch and the moving camera presents a significant challenge. By using a daisy-chaining approach (e.g., [16–19]), Euclidean homography relationships between different camera coordinate frames and feature point patch coordinate frames are developed. These homographies are used to relate coordinate frames attached to the moving camera, the reference object, the control object, and the object used to record the desired trajectory. Another challenge is that for general six DOF motion by both the camera and the planar patch, the normal to planar patch is unknown. By decomposing the homography relationships, the normal to

the moving feature point patch can be obtained. Likewise, the distance between the moving camera, the moving planar patch, and a reference patch are unknown. By using the depth ratios obtained from the homography decomposition, the unknown distance is related to an unknown constant parameter. A Lyapunov-based adaptive estimation law is designed to compensate for the unknown constant parameter. Since the moving camera could be attached to a remotely piloted vehicle with arbitrary rotations, another challenge is to eliminate potential singularities in the rotation parameterization obtained from the homography decomposition. To address this issue, homography-based visual servo control techniques (e.g., [10, 20–22]) are combined with quaternion-based control methods (e.g., [14, 23, 24]), to eliminate singularities associated with the image Jacobian and the rotation error system. By using the quaternion parameterization, the resulting closed-loop rotation error system can be stabilized by a proportional rotation controller combined with a feedforward term that is a function of the desired trajectory.

3) *Visual servo tracking control using a central catadioptric camera.*

Visual servo controllers require the image-space coordinates of some set of Euclidean feature points in the control development; hence, the feature points must remain in the camera’s field-of-view (FOV). Since the FOV of conventional perspective cameras (e.g., pinhole cameras) is restricted, keeping the feature points in the FOV is a fundamental challenge for visual servo control algorithms. The fundamental nature of the FOV problem has resulted in a variety of control and path planning methods (e.g., [6, 7, 25–31]). An alternative solution to the aforementioned algorithmic approaches to resolve the FOV issue is to use advanced optics such as omnidirectional cameras. Catadioptric cameras (one type of omnidirectional camera) are devices which use both mirrors (reflective or catadioptric elements) and lenses (refractive or dioptric elements) to form images [32]. Catadioptric cameras with a single effective viewpoint are classified as central catadioptric cameras,

which are desirable because they yield pure perspective images [33]. In Chapter 5, a visual servo control scheme is presented that yields a tracking result for a camera-in-hand central catadioptric camera system. The tracking controller is developed based on the relative relationships of a central catadioptric camera between the current, reference, and desired camera poses. To find the relative camera pose relationships, homographies are computed based on the projection model of the central catadioptric camera [33–36]. Geyer and Daniilidis [36] proposed a unifying theory to show that all central catadioptric systems are isomorphic to projective mappings from the sphere to a plane with a projection center on the perpendicular axis to the plane. By constructing links between the projected coordinates on the sphere, the homographies up to scalar multiples can be obtained. Various methods can then be applied to decompose the Euclidean homographies to find the corresponding rotation matrices, and depth ratios. The rotation error system is based on the quaternion formulation which has a full-rank interaction matrix. Lyapunov-based methods are utilized to develop the controller and to prove asymptotic tracking.

4) *Robust visual servo control.*

In vision-based control, exact calibration is often required so that the image-space sensor measurements can be related to the Euclidean or joint space for control implementation. Specifically, a camera model (e.g., the pinhole model) is often required to relate pixel coordinates from an image to the (normalized) Euclidean coordinates. The camera model is typically assumed to be exactly known (i.e., the intrinsic calibration parameters are assumed to be known); however, despite the availability of several popular calibration methods (cf. [37–43]), camera calibration can be time consuming, requires some level of expertise, and has inherent inaccuracies. If the calibration parameters are not exactly known, performance degradation and potential unpredictable response from the visual servo controller may occur. The goal of this research is to develop a visual servo

controller which is robust to the intrinsic calibration parameters. As in the previous three problems, the quaternion parameterization will be used to represent the rotation error system. Since the quaternion error cannot be measured precisely due to the uncertain calibration, an estimated quaternion is required to develop the controller. One of the challenges to develop a quaternion estimate is that the estimated rotation matrix is not a true rotation matrix in general. To address this challenge, the similarity relationship between the estimated and actual rotation matrices is used to construct the relationship between the estimated and actual quaternions. A Lyapunov-based stability analysis is provided that indicates a unique controller can be developed to achieve the regulation result.

5) *Combined robust and adaptive visual servo control.*

This research is also motivated by the desire to compensate for uncertain camera calibration. This controller has adaptive updated terms which can compensate for the unknown calibration parameters. The open-loop error system is composed of a rotation error system and a translation error system. One challenge is that the rotation quaternion error is not measurable. To address this problem, an estimated quaternion is obtained based on the image-space information and is used to develop the controller. The transformation between the actual and estimated quaternions is an upper triangular matrix determined by the calibration parameters and the diagonal elements are positive. This fact is exploited to design a robust high-gain controller. Another challenge is that the unknown calibration matrix is coupled in the translation error system. To address this problem, the translation error system is linearly parameterized in terms of the calibration parameters. An adaptive update law is used to estimate the unknown calibration parameters, and a translation controller containing the adaptive compensation terms is used to asymptotically regulate the translation error.

1.3 Literature Review

1.3.1 Basic Visual Servo Control Approaches

Different visual servo control methods can be divided into three main categories including: image-based, position-based, and approaches that make use of a blend of image and position-based approaches. Image-based visual servo control (e.g., [1, 44–47]) consists of a feedback signal that is composed of pure image-space information (i.e., the control objective is defined in terms of an image pixel error). This approach is considered to be more robust to camera calibration and robot kinematic errors and is more likely to keep the relevant image features in the FOV than position-based methods because the feedback is directly obtained from the image without the need to transfer the image-space measurement to another space. A drawback of image-based visual servo control is that since the controller is implemented in the robot joint space, an image-Jacobian is required to relate the derivative of the image-space measurements to the camera’s linear and angular velocities. However, the image-Jacobian typically contains singularities and local minima (see Chaumette [48]), and the controller stability analysis is difficult to obtain in the presence of calibration uncertainty (see Espiau et al. [49]). Another drawback of image-based methods are that since the controller is based on image-feedback, the robot could be commanded along a trajectory that is not physically possible. This issue is described as Chaumette’s conundrum. Further discussion of the Chaumette’s conundrum is provided in Chaumette [48] and Corke and Hutchinson [25].

Position-based visual servo control (e.g., [1, 44, 50–52]) uses reconstructed Euclidean information in the feedback loop. For this approach, the image-Jacobian singularity and local minima problems are avoided, and physically realizable trajectories are generated. However, the approach is susceptible to inaccuracies in the task-space reconstruction if the transformation is corrupted (e.g., uncertain

camera calibration). Also, since the controller does not directly use the image features in the feedback, the commanded robot trajectory may cause the feature points to leave the FOV. A review of these two approaches is provided in [1, 2, 53].

The third class of visual servo controllers use some image-space information combined with some reconstructed information as a means to combine the advantages of these two approaches while avoiding their disadvantages (e.g., [10, 20–22, 24, 25, 54–58]). One particular approach was coined 2.5D visual servo control in [20, 21, 55, 56] because this class of controllers exploits two dimensional image feedback and reconstructed three-dimensional feedback. This class of controllers is also called homography-based visual servo control in [10, 22, 24, 57] because of the underlying reliance of the construction and decomposition of a homography.

1.3.2 Visual Servo Control Approaches to Enlarge the FOV

Visual servo controllers often require the image-space coordinates of some set of Euclidean feature points in the control development; hence, the feature points must remain in the camera’s FOV. Since the FOV of conventional perspective cameras (e.g., pinhole cameras) is restricted, keeping the feature points in the FOV is a fundamental challenge for visual servo control algorithms. The fundamental nature of the FOV problem has resulted in a variety of control and path planning methods (e.g., [6, 7, 25–31]). Corke and Hutchinson [25] and Chesi et al. [26] used partitioned or switching visual servoing methods to keep the object in the FOV. In [6, 7, 27–29], potential fields (or navigation functions) are used to ensure the visibility of all features during the control task. In Benhimane and Malis [30], the focal length of the camera was automatically adjusted (i.e., zoom control) to keep all features in the FOV during the control task by using an intrinsic-free visual servoing approach developed by Malis [59]. In Garcka-Aracil et al. [31], a continuous controller is obtained by using a new smooth task function

with weighted features that allows visibility changes in the image features (i.e., some features can come in and out of the FOV) during the control task. Some researchers have also investigated methods to enlarge the FOV [60–64]. In [60–63], image mosaicing is used to capture multiple images of the scene as a camera moves and the images are stitched together to obtain a larger image. In Swaminathan and Nayar [64], multiple images are fused from multiple cameras mounted in order to have minimally overlapping FOV.

An alternative solution to the aforementioned algorithmic approaches to resolve the FOV issue is to use advanced optics such as omnidirectional cameras. Catadioptric cameras (one type of omnidirectional camera) are devices which use both mirrors (reflective or catadioptric elements) and lenses (refractive or dioptric elements) to form images [32]. Catadioptric systems with a single effective viewpoint are classified as central catadioptric systems, which are desirable because they yield pure perspective images [33]. In Baker and Nayar [34], the complete class of single-lens single-mirror catadioptric systems is derived that satisfy the single viewpoint constraint. Recently, catadioptric systems have been investigated to enlarge the FOV for visual servo control tasks (e.g., [35, 65–72]). Burschka and Hager [65] addressed the visual servoing problem of mobile robots equipped with central catadioptric cameras, in which an estimation of the feature height to the plane of motion is required. Barreto et al. [73] developed a model-based tracking approach of a rigid object using a central catadioptric camera. Mezouar et al. [66] controlled a robotic system using the projection of 3D lines in the image plane of a central catadioptric system. In [35, 65, 66], the inverse of the image Jacobian is required in the controller development which may lead to a singularity problem for certain configurations. Hadj-Abdelkader et al. [67] presented the 2 1/2 D visual servoing approach using omnidirectional cameras, in which the inverse of an estimated image-Jacobian (containing potential singularities) is required in

the controller. Mariottini et al. [68, 69] developed an image-based visual servoing strategy using epipolar geometry for a three DOF mobile robot equipped with a central catadioptric camera. Particularly, the singularity problem in the image Jacobian was addressed by Mariottini et al. [69] using epipolar geometry. In Benhimane and Malis [70], a new approach to visual servo regulation for omnidirectional cameras was developed that uses the feedback of a homography directly (without requiring a decomposition) that also does not require any measure of the 3D information on the observed scene. However, the result in [70] is restricted to be local since the controller is developed based on a linearized open-loop error system at the origin, and the task function is only isomorphic to a restricted region within the omnidirectional view. In Tatsambon and Chaumette [71, 72], a new optimal combination of visual features is proposed for visual servoing from spheres using central catadioptric systems.

1.3.3 Robust and Adaptive Visual Servo Control

Motivated by the desire to incorporate robustness to camera calibration, different control approaches that do not depend on exact camera calibration have been proposed (cf. [9, 21, 74–88]). Efforts such as [74–78] have investigated the development of methods to estimate the image and robot manipulator Jacobians. These methods are composed of some form of recursive Jacobian estimation law and a control law. Specifically, Hosoda and Asada [74] developed a visual servo controller based on a weighted recursive least-squares update law to estimate the image Jacobian. In Jagersand et al. [75], a Broyden Jacobian estimator is applied and a nonlinear least-square optimization method is used for the visual servo control development. Shahamiri and Jagersand [76] used a nullspace-biased Newton-step visual servo strategy with a Broyden Jacobian estimation for online singularity detection and avoidance in an uncalibrated visual servo control problem. In Piepmeier and Lipkin [77] and Piepmeier et al. [78], a recursive least-squares

algorithm is implemented for Jacobian estimation, and a dynamic Gauss-Newton method is used to minimize the squared error in the image plane.

Robust control approaches based on static best-guess estimation of the calibration matrix have been developed to solve the uncalibrated visual servo regulation problem (cf. [21, 82, 87, 88]). Specifically, under a set of assumptions on the rotation and calibration matrix, a kinematic controller was developed by Taylor and Ostrowski [82] that utilizes a constant, best-guess estimate of the calibration parameters to achieve local set-point regulation for the six DOF visual servo control problem. Homography-based visual servoing methods using best-guess estimation are used by Malis and Chaumette [21] and Fang et al. [87] to achieve asymptotic or exponential regulation with respect to both camera and hand-eye calibration errors for the six DOF problem.

The development of traditional adaptive control methods to compensate for uncertainty in the camera calibration matrix is inhibited because of the time-varying uncertainty injected in the transformation from the normalization of the Euclidean coordinates. As a result, initial adaptive control results such as [79–85] were limited to scenarios where the optic axis of the camera was assumed to be perpendicular with the plane formed by the feature points (i.e., the time-varying uncertainty is reduced to a constant uncertainty) or assumed an additional sensor (e.g., ultrasonic sensors, laser-based sensors, additional cameras) could be used to measure the depth information.

More recent approaches exploit geometric relationships between multiple spatiotemporal views of an object to transform the time-varying uncertainty into known time-varying terms multiplied by an unknown constant [9, 21, 86–89]. In Ruf et al. [9], an on-line calibration algorithm was developed for position-based visual servoing. In Liu et al. [86], an adaptive image-based visual servo controller was developed that regulated the feature points in an image to desired locations.

One problem with methods based on the image-Jacobian is that the estimated image-Jacobian may contain singularities. The development in [86] exploits an additional potential force function to drive the estimated parameters away from the values that result in a singular Jacobian matrix. In Chen et al. [89], an adaptive homography-based controller was proposed to address problems of uncertainty in the intrinsic camera calibration parameters and lack of depth measurements. Specifically, an adaptive control strategy was developed from a Lyapunov-based approach that exploits the triangular structure of the calibration matrix. To the best of our knowledge, the result in [89] was the first result that regulates the robot end-effector to a desired position/orientation through visual servoing by actively compensating for the lack of depth measurements and uncertainty in the camera intrinsic calibration matrix with regard to the six DOF regulation problem. However, the relationship between the estimated rotation axis and the actual rotation axis is not correctly developed. A time-varying scaling factor was omitted which is required to relate the estimated rotation matrix and the actual rotation matrix. Specifically, the estimated rotation matrix and the actual rotation matrix were incorrectly related through eigenvectors that are associated with the eigenvalue of 1. An unknown time-varying scalar is required to relate these vectors, and the methods developed in [89] do not appear to be suitable to accommodate for this uncertainty.

1.4 Contributions

The main contribution of this dissertation is the development of visual servo control algorithms and architectures that exploit the visual feedback from a camera system to achieve a tracking or regulation control objective for a rigid-body control object (e.g., the end-effector of a robot manipulator, a satellite, an autonomous vehicle) identified by a patch of feature points. In the process of achieving the main contribution, the following contributions were made:

- A new adaptive homography-based visual servo control method via a quaternion formulation is developed that achieves asymptotic tracking control. This control scheme is singularity-free by exploiting the homography techniques and a quaternion parameterization. The adaptive estimation term in the proposed controller compensates for the unknown depth information dynamically while the controller achieves the asymptotic tracking results.
- A new collaborative visual servo control method is developed to enable a rigid-body object to track a desired trajectory via a daisy-chaining multi-view geometry. In contrast to typical camera-to-hand and camera-in-hand visual servo control configurations, the proposed controller is developed using a moving on-board camera viewing a moving object to obtain feedback signals. This collaborative method weakens the FOV restriction and enables the control object to perform large area motion.
- A visual servo controller is developed that yields an asymptotic tracking result for the complete nonlinear six DOF camera-in-hand central catadioptric camera system. A panoramic FOV is obtained by using the central catadioptric camera.
- A robust visual servo controller is developed to achieve a regulation control objective in presence of intrinsic camera calibration uncertainties. A quaternion-based estimate for the rotation error signal is developed and used in the controller development. The similarity relationship between the estimated and actual rotation matrices is used to construct the relationship between the estimated and actual quaternions. A Lyapunov-based stability analysis is provided that indicates a unique controller can be developed to achieve the regulation result despite a sign ambiguity in the developed quaternion estimate.

- A new combined robust and adaptive visual servo control method is developed to asymptotically regulate the feature points in an image to the desired locations while also regulating the six DOF pose of the control object without calibrating the camera. These dual objectives are achieved by using a homography-based approach that exploits both image-space and reconstructed Euclidean information in the feedback loop. The robust rotation controller that accommodates for the time-varying uncertain scaling factor is developed by exploiting the upper triangular form of the rotation error system and the fact that the diagonal elements of the camera calibration matrix are positive. The adaptive translation controller that compensates for the constant unknown parameters in the translation error system is developed by a certainty-equivalence-based adaptive control method and a nonlinear Lyapunov-based design approach.

CHAPTER 2 BACKGROUND AND PRELIMINARY DEVELOPMENT

The purpose of this chapter is to provide some background information pertaining to the camera geometric model, Euclidean reconstruction, and unit quaternion parameterization approach. Sections 2.1 and 2.2 develop the notation and framework for the camera geometric model and Euclidean reconstruction used in Chapters 3, 6 and 7. Their extensions are also used in Chapters 4 and 5. Section 2.3 reviews the unit quaternion, a rotation representation approach that is used throughout this dissertation.

2.1 Geometric Model

Image processing techniques can often be used to select coplanar and non-collinear feature points within an image. However, if four coplanar feature points are not available then the subsequent development can also exploit the classic eight-points algorithm with no four of the eight feature points being coplanar (see Hartley and Zisserman [13]) or the virtual parallax method (see Boufama and Mohr [90] and Malis [55]) where the non-coplanar points are projected onto a virtual plane. Without loss of generality, the subsequent development is based on the assumption that an object (e.g., the end-effector of a robot manipulator, an aircraft, a tumbling satellite, an autonomous vehicle, etc.) has four coplanar and non-collinear feature points denoted by $O_i \forall i = 1, 2, 3, 4$, and the feature points can be determined from a feature point tracking algorithm (e.g., Kanade-Lucas-Tomasi (KLT) algorithm discussed by Shi and Tomasi [91] and Tomasi and Kanade [92]). The plane defined by the four feature points is denoted by π as depicted in Figure 2–1. The coordinate frame \mathcal{F} in Figure 2–1 is affixed to a camera viewing the object, the stationary coordinate frame \mathcal{F}^* denotes a

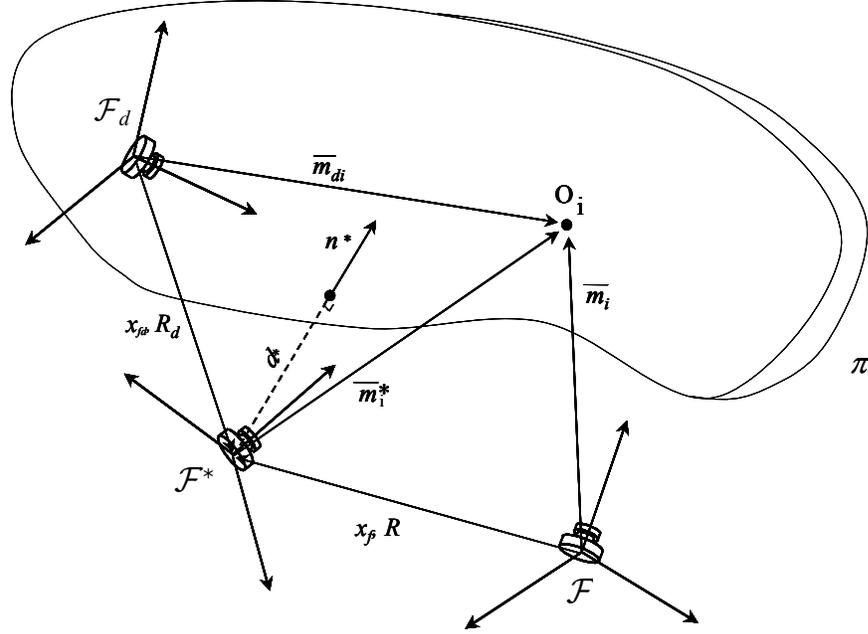


Figure 2–1: Coordinate frame relationships between a camera viewing a planar patch at different spatiotemporal instances. The coordinate frames \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d are attached to the current, reference and desired locations, respectively.

reference location for the camera, and the coordinate frame \mathcal{F}_d that is attached to the desired location of the camera. When the desired location of the camera is a constant, \mathcal{F}_d can be chosen the same as \mathcal{F}^* as shown in Figure 2–2. That is, the tracking problem becomes a more particular regulation problem for the configuration in Figure 2–2.

The vectors $\bar{m}_i(t)$, \bar{m}_i^* , $\bar{m}_{di}(t) \in \mathbb{R}^3$ in Figure 2–1 are defined as

$$\begin{aligned} \bar{m}_i &\triangleq \begin{bmatrix} x_i(t) & y_i(t) & z_i(t) \end{bmatrix}^T \\ \bar{m}_i^* &\triangleq \begin{bmatrix} x_i^* & y_i^* & z_i^* \end{bmatrix}^T \\ \bar{m}_{di} &\triangleq \begin{bmatrix} x_{di}(t) & y_{di}(t) & z_{di}(t) \end{bmatrix}^T, \end{aligned} \quad (2-1)$$

where $x_i(t)$, $y_i(t)$, $z_i(t) \in \mathbb{R}$, x_i^* , y_i^* , $z_i^* \in \mathbb{R}$ and $x_{di}(t)$, $y_{di}(t)$, $z_{di}(t) \in \mathbb{R}$ denote the Euclidean coordinates of the feature points O_i expressed in the frames \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d , respectively. From standard Euclidean geometry, relationships between $\bar{m}_i(t)$,

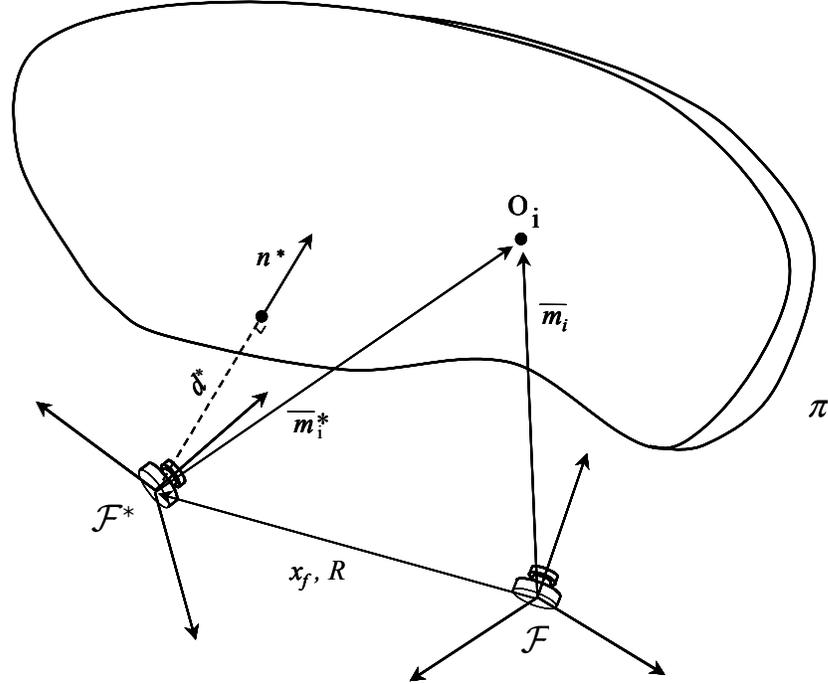


Figure 2–2: Coordinate frame relationships between a camera viewing a planar patch at different spatiotemporal instances. The coordinate frames \mathcal{F} and \mathcal{F}^* are attached to the current and reference locations, respectively.

\bar{m}_i^* and $\bar{m}_{di}(t)$ can be determined as

$$\bar{m}_i = x_f + R\bar{m}_i^* \quad \bar{m}_{di} = x_{fd} + R_d\bar{m}_i^*, \quad (2-2)$$

where $R(t), R_d(t) \in SO(3)$ denote the orientations of \mathcal{F}^* with respect to \mathcal{F} and \mathcal{F}_d , respectively, and $x_f(t), x_{fd}(t) \in \mathbb{R}^3$ denote translation vectors from \mathcal{F} to \mathcal{F}^* and \mathcal{F}_d to \mathcal{F}^* expressed in the coordinates of \mathcal{F} and \mathcal{F}_d , respectively. As also illustrated in Figure 2–1, $n^* \in \mathbb{R}^3$ denotes the constant unit normal to the plane π , and the constant distance from the origin of \mathcal{F}^* to π along the unit normal is denoted by $d^* \triangleq n^{*T}\bar{m}_i^* \in \mathbb{R}$. The normalized Euclidean coordinates, denoted by

$m_i(t), m_i^*, m_{di}(t) \in \mathbb{R}^3$ are defined as

$$\begin{aligned} m_i &\triangleq \frac{\bar{m}_i}{z_i} = \begin{bmatrix} \frac{x_i}{z_i} & \frac{y_i}{z_i} & 1 \end{bmatrix}^T \\ m_i^* &\triangleq \frac{\bar{m}_i^*}{z_i^*} = \begin{bmatrix} \frac{x_i^*}{z_i^*} & \frac{y_i^*}{z_i^*} & 1 \end{bmatrix}^T \\ m_{di} &\triangleq \frac{\bar{m}_{di}}{z_{di}} = \begin{bmatrix} \frac{x_{di}}{z_{di}} & \frac{y_{di}}{z_{di}} & 1 \end{bmatrix}^T \end{aligned} \quad (2-3)$$

with the standard assumption that $z_i(t), z_i^*, z_{di}(t) > \varepsilon$ where ε is an arbitrarily small positive constant. From (2-3), the relationships in (2-2) can be expressed as

$$\begin{aligned} m_i &= \underbrace{\frac{z_i^*}{z_i}}_{\alpha_i} \underbrace{\left(R + \frac{x_f}{d^*} n^{*T} \right)}_H m_i^* \\ m_{di} &= \underbrace{\frac{z_i^*}{z_{di}}}_{\alpha_{di}} \underbrace{\left(R_d + \frac{x_{fd}}{d^*} n^{*T} \right)}_{H_d} m_i^* \end{aligned}, \quad (2-4)$$

where $\alpha_i(t), \alpha_{di}(t) \in \mathbb{R}$ are scaling terms, and $H(t), H_d(t) \in \mathbb{R}^{3 \times 3}$ denote the Euclidean homographies.

2.2 Euclidean Reconstruction

Each feature point on π has a projected pixel coordinate $p_i(t) \in \mathbb{R}^3, p_i^* \in \mathbb{R}^3$ and $p_{di}(t) \in \mathbb{R}^3$ in $\mathcal{F}, \mathcal{F}^*$ and \mathcal{F}_d respectively, denoted by

$$\begin{aligned} p_i &\triangleq \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T \\ p_i^* &\triangleq \begin{bmatrix} u_i^* & v_i^* & 1 \end{bmatrix}^T \\ p_{di} &\triangleq \begin{bmatrix} u_{di} & v_{di} & 1 \end{bmatrix}^T, \end{aligned} \quad (2-5)$$

where $u_i(t), v_i(t), u_i^*, v_i^*, u_{di}(t), v_{di}(t) \in \mathbb{R}$. The projected pixel coordinates $p_i(t), p_i^*$ and $p_{di}(t)$ are related to the normalized task-space coordinates $m_i(t), m_i^*$ and $m_{di}(t)$ by the following global invertible transformation (i.e., the pinhole camera

model)

$$p_i = Am_i \quad p_i^* = Am_i^* \quad p_{di} = Am_{di}, \quad (2-6)$$

where $A \in \mathbb{R}^{3 \times 3}$ is a constant, upper triangular, and invertible intrinsic camera calibration matrix that is explicitly defined as [13]

$$A \triangleq \begin{bmatrix} \alpha & -\alpha \cot \phi & u_0 \\ 0 & \frac{\beta}{\sin \phi} & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2-7)$$

In (2-7), $u_0, v_0 \in \mathbb{R}$ denote the pixel coordinates of the principal point (i.e., the image center that is defined as the frame buffer coordinates of the intersection of the optical axis with the image plane), $\alpha, \beta \in \mathbb{R}$ represent the product of the camera scaling factors and the focal length, and $\phi \in \mathbb{R}$ is the skew angle between the camera axes.

Based on (2-6), the Euclidean relationship in (2-4) can be expressed in terms of the image coordinates as

$$\begin{aligned} p_i &= \alpha_i \underbrace{(AHA^{-1})}_{G} p_i^* \\ p_{di} &= \alpha_{di} \underbrace{(AH_d A^{-1})}_{G_d} p_i^*. \end{aligned} \quad (2-8)$$

By using the feature point pairs $(p_i^*, p_i(t))$ and $(p_i^*, p_{di}(t))$, the projective homography up to a scalar multiple (i.e., G and G_d) can be determined (see Chen et al. [10]). Various methods can then be applied (e.g., see Faugeras and Lustman [93] and Zhang and Hanson [94]) to decompose the Euclidean homographies to obtain the rotation matrices $R(t)$, $R_d(t)$ and the depth ratios $\alpha_i(t)$, $\alpha_{di}(t)$.

2.3 Unit Quaternion Representation of the Rotation Matrix

For a given rotation matrix, several different representations (e.g., Euler angle-axis, direction cosines matrix, Euler angles, unit quaternion (or Euler parameters), etc.) can be utilized to develop the error system. In previous homography-based visual servo control literature, the Euler angle-axis representation has been used to describe the rotation matrix. In the angle-axis parameters (φ, k) , $\varphi(t) \in \mathbb{R}$ represents a rotation angle about a suitable unit vector $k(t) \in \mathbb{R}^3$. The parameters (φ, k) can be easily calculated (e.g., using the algorithm shown in Spong and Vidyasagar [11]).

Given unit vector $k(t)$ and angle $\varphi(t)$, the rotation matrix $R(t) = e^{k^\times \varphi}$ can be calculated using the Rodrigues formula

$$R = e^{k^\times \varphi} = I_3 + k^\times \sin(\varphi) + (k^\times)^2 (1 - \cos(\varphi)), \quad (2-9)$$

where I_3 is the 3×3 identity matrix, and the notation $k^\times(t)$ denotes the following skew-symmetric form of the vector $k(t)$:

$$k^\times = \begin{bmatrix} 0 & -k_3 & k_2 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix} \quad \forall k = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}^T. \quad (2-10)$$

The unit quaternion is a four dimensional vector which can be defined as [23]

$$q \triangleq \begin{bmatrix} q_0 & q_v^T \end{bmatrix}^T. \quad (2-11)$$

In (2-11), $q_v(t) \triangleq \begin{bmatrix} q_{v1}(t) & q_{v2}(t) & q_{v3}(t) \end{bmatrix}^T$, $q_0(t), q_{vi}(t) \in \mathbb{R} \forall i = 1, 2, 3$. The unit quaternion must also satisfy the following nonlinear constraint

$$q^T q = 1. \quad (2-12)$$

This parameterization facilitates the subsequent problem formulation, control development, and stability analysis since the unit quaternion provides a globally nonsingular parameterization of the rotation matrix.

Given (φ, k) , the unit quaternion vector $q(t)$ can be constructed as

$$\begin{bmatrix} q_0(t) \\ q_v(t) \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\varphi(t)}{2}\right) \\ k(t) \sin\left(\frac{\varphi(t)}{2}\right) \end{bmatrix}. \quad (2-13)$$

Based on (2-13), the rotation matrix in (2-9) can be expressed as

$$R(q) = I_3 + 2q_0q_v^\times + 2(q_v^\times)^2 = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T + 2q_0 q_v^\times. \quad (2-14)$$

The rotation matrix in (2-14) is typical in robotics literature where the moving coordinate system is expressed in terms of a fixed coordinate system (typically the coordinate system attached to the base frame). However, the typical representation of the rotation matrix in aerospace literature (e.g., [15]) is

$$R(q) = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T - 2q_0 q_v^\times. \quad (2-15)$$

The difference is due to the fact that the rotation matrix in (2-15) (which is used in the current dissertation) relates the moving coordinate frame \mathcal{F} to the fixed coordinate frame \mathcal{F}^* with the corresponding states expressed in \mathcal{F} . The rotation matrix in (2-15) can be expanded as

$$R(q) = \begin{bmatrix} q_0^2 + q_{v1}^2 - q_{v2}^2 - q_{v3}^2 & 2(q_{v1}q_{v2} + q_{v3}q_0) & 2(q_{v1}q_{v3} - q_{v2}q_0) \\ 2(q_{v1}q_{v2} - q_{v3}q_0) & q_0^2 - q_{v1}^2 + q_{v2}^2 - q_{v3}^2 & 2(q_{v2}q_{v3} + q_{v1}q_0) \\ 2(q_{v1}q_{v3} + q_{v2}q_0) & 2(q_{v2}q_{v3} - q_{v1}q_0) & q_0^2 - q_{v1}^2 - q_{v2}^2 + q_{v3}^2 \end{bmatrix}. \quad (2-16)$$

From (2-16) various approaches could be used to determine $q_0(t)$ and $q_v(t)$; however, numerical significance of the resulting computations can be lost if $q_0(t)$ is close to zero [15]. Shuster [15] developed a method to determine $q_0(t)$ and $q_v(t)$ that provides robustness against such computational issues. Specifically, the

diagonal terms of $R(q)$ can be obtained from (2–12) and (2–16) as

$$R_{11} = 1 - 2(q_{v2}^2 + q_{v3}^2) \quad (2-17)$$

$$R_{22} = 1 - 2(q_{v1}^2 + q_{v3}^2) \quad (2-18)$$

$$R_{33} = 1 - 2(q_{v1}^2 + q_{v2}^2). \quad (2-19)$$

By utilizing (2–12) and (2–17)-(2–19), the following expressions can be developed:

$$\begin{aligned} q_0^2 &= \frac{R_{11} + R_{22} + R_{33} + 1}{4} & (2-20) \\ q_{v1}^2 &= \frac{R_{11} - R_{22} - R_{33} + 1}{4} \\ q_{v2}^2 &= \frac{R_{22} - R_{11} - R_{33} + 1}{4} \\ q_{v3}^2 &= \frac{R_{33} - R_{11} - R_{22} + 1}{4}, \end{aligned}$$

where $q_0(t)$ is restricted to be non-negative without loss of generality (this restriction enables the minimum rotation to be obtained). As stated in [15], the greatest numerical accuracy for computing $q_0(t)$ and $q_v(t)$ is obtained by using the element in (2–20) with the largest value and then computing the remaining terms respectively. For example, if $q_0^2(t)$ has the maximum value in (2–20) then the greatest numerical accuracy can be obtained by computing $q_0(t)$ and $q_v(t)$ as

$$\begin{aligned} q_0 &= \sqrt{\frac{R_{11} + R_{22} + R_{33} + 1}{4}} \\ q_{v1} &= \frac{R_{23} - R_{32}}{4q_0} \\ q_{v2} &= \frac{R_{31} - R_{13}}{4q_0} \\ q_{v3} &= \frac{R_{12} - R_{21}}{4q_0}. \end{aligned} \quad (2-21)$$

Likewise, if $q_{v1}^2(t)$ has the maximum value in (2-20) then the greatest numerical accuracy can be obtained by computing $q_0(t)$ and $q_v(t)$ as

$$\begin{aligned} q_0 &= \frac{R_{23} - R_{32}}{4q_{v1}} \\ q_{v1} &= \pm \sqrt{\frac{R_{11} - R_{22} - R_{33} + 1}{4}} \\ q_{v2} &= \frac{R_{12} + R_{21}}{4q_{v1}} \\ q_{v3} &= \frac{R_{13} + R_{31}}{4q_{v1}}, \end{aligned} \tag{2-22}$$

where the sign of $q_{v1}(t)$ is selected so that $q_0(t) \geq 0$. If $q_{v2}^2(t)$ is the maximum, then

$$\begin{aligned} q_0 &= \frac{R_{31} - R_{13}}{4q_{v2}} \\ q_{v1} &= \frac{R_{12} + R_{21}}{4q_{v2}} \\ q_{v2} &= \pm \sqrt{\frac{R_{22} - R_{11} - R_{33} + 1}{4}} \\ q_{v3} &= \frac{R_{23} + R_{32}}{4q_{v2}}, \end{aligned} \tag{2-23}$$

or if $q_{v3}^2(t)$ is the maximum, then

$$\begin{aligned} q_0 &= \frac{R_{12} - R_{21}}{4q_{v3}} \\ q_{v1} &= \frac{R_{13} + R_{31}}{4q_{v3}} \\ q_{v2} &= \frac{R_{23} + R_{32}}{4q_{v3}} \\ q_{v3} &= \pm \sqrt{\frac{R_{33} - R_{11} - R_{22} + 1}{4}}, \end{aligned} \tag{2-24}$$

where the sign of $q_{v2}(t)$ or $q_{v3}(t)$ is selected so that $q_0(t) \geq 0$.

The expressions in (2-21)-(2-24) indicate that given the rotation matrix $R(t)$ from the homography decomposition, the unit quaternion vector can be determined that represents the rotation without introducing a singularity. The expressions in (2-21)-(2-24) will be utilized in the subsequent control development and stability analysis.

CHAPTER 3
LYAPUNOV-BASED VISUAL SERVO TRACKING CONTROL VIA A
QUATERNION FORMULATION

3.1 Introduction

Previous visual servo controllers typically only address the regulation problem. Motivated by the need for new advancements to meet visual servo tracking applications, previous research has concentrated on developing different types of path planning techniques [5–9]. Recently, Chen et al. [10] developed a new formulation of the tracking control problem. The homography-based adaptive visual servo controller in [10] is developed to enable an actuated object to track a prerecorded time-varying desired trajectory determined by a sequence of images, where the Euler angle-axis representation is used to represent the rotation error system. Due to the computational singularity limitation of the angle axis extraction algorithm (see Spong and Vidyasagar [11]), rotation angles of $\pm\pi$ were not considered.

This chapter considers the previously unexamined problem of six DOF visual servo tracking control with a nonsingular rotation parameterization. A homography is constructed from image pairs and decomposed via textbook methods (e.g., Faugeras [12] and Hartley and Zisserman [13]) to obtain the rotation matrix. Once the rotation matrix has been determined, the corresponding unit quaternion can be determined from globally nonsingular and numerically robust algorithms (e.g., Hu et al. [14] and Shuster [15]). An error system is constructed in terms of the unit quaternion. An adaptive controller is then developed and proven to enable a camera (attached to a rigid-body object) to track a desired trajectory that is determined from a sequence of images. These images can be taken online or offline by a camera. For example, a sequence of images of the reference object can be

prerecorded as the camera moves (a camera-in-hand configuration), and these images can be used as a desired trajectory in a later real-time tracking control. The camera is attached to a rigid-body object (e.g., the end-effector of a robot manipulator, a satellite, an autonomous vehicle, etc.) that can be identified by a planar patch of feature points. The controller contains an adaptive feedforward term to compensate for the unknown distance from the camera to the observed features. A quaternion-based Lyapunov function is developed to facilitate the control design and the stability analysis.

The remainder of this chapter is organized as follows. In Section 3.2, the control objective is formulated in terms of unit quaternion representation. In Section 3.3, the controller is developed, and closed-loop stability analysis is given based on Lyapunov-based methods. In Section 3.4, the control development is extended to the camera-to-hand configuration. In Sections 3.5 and 3.6, Matlab simulations and tracking experiments that were performed in a virtual-reality test-bed for unmanned systems at the University of Florida are used to show the performance of the proposed visual servo tracking controller.

3.2 Control Objective

The control objective is for a camera to track a desired trajectory that is determined by a sequence of images. This objective is based on the assumption that the linear and angular velocities of the camera are control inputs that can be independently controlled (i.e., unconstrained motion) and that the camera is calibrated (i.e., A is known). The signals in (2–5) are the only required measurements to develop the controller.

One of the outcomes of the homography decomposition is the rotation matrices $R(t)$ and $R_d(t)$. From these rotation matrices, several different representations can be utilized to develop the error system. In previous homography-based visual servo control literature, the Euler angle-axis representation has been used to describe the

rotation matrix. In this chapter, the unit quaternion parameterization will be used to describe the rotation matrix. This parameterization facilitates the subsequent problem formulation, control development, and stability analysis since the unit quaternion provides a global nonsingular parameterization of the corresponding rotation matrices. Section 2.3 provides background, definitions and development related to the unit quaternion.

Given the rotation matrices $R(t)$ and $R_d(t)$, the corresponding unit quaternions $q(t)$ and $q_d(t)$ can be calculated by using the numerically robust method (see [14] and [15]) based on the corresponding relationships

$$R(q) = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T - 2q_0 q_v^\times \quad (3-1)$$

$$R_d(q_d) = (q_{0d}^2 - q_{vd}^T q_{vd}) I_3 + 2q_{vd} q_{vd}^T - 2q_{0d} q_{vd}^\times, \quad (3-2)$$

where I_3 is the 3×3 identity matrix, and the notation $q_v^\times(t)$ denotes the skew-symmetric form of the vector $q_v(t)$ as in (2-12).

To quantify the error between the actual and desired camera orientations, the mismatch between rotation matrices $R(t)$ and $R_d(t)$ is defined as

$$\tilde{R} = R R_d^T. \quad (3-3)$$

Based on (3-1)-(3-3),

$$\tilde{R} = (\tilde{q}_0^2 - \tilde{q}_v^T \tilde{q}_v) I_3 + 2\tilde{q}_v \tilde{q}_v^T - 2\tilde{q}_0 \tilde{q}_v^\times, \quad (3-4)$$

where the error quaternion $(\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ is defined as

$$\tilde{q}_0 = q_0 q_{0d} + q_v^T q_{vd} \quad (3-5)$$

$$\tilde{q}_v = q_{0d} q_v - q_0 q_{vd} + q_v^\times q_{vd}.$$

The definition of $\tilde{q}_0(t)$ and $\tilde{q}_v(t)$ in (3-5) makes $(\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ a unit quaternion based on the fact that $q(t)$ and $q_d(t)$ are two unit quaternions (see Appendix A).

The translation error, denoted by $e(t) \in \mathbb{R}^3$, is defined as

$$e = p_e - p_{ed}, \quad (3-6)$$

where $p_e(t), p_{ed}(t) \in \mathbb{R}^3$ are defined as

$$p_e = \begin{bmatrix} u_i & v_i & -\ln(\alpha_i) \end{bmatrix}^T \quad p_{ed} = \begin{bmatrix} u_{di} & v_{di} & -\ln(\alpha_{di}) \end{bmatrix}^T, \quad (3-7)$$

where $i \in \{1, \dots, 4\}$.

In the Euclidean-space (see Figure 2-1), the tracking objective can be quantified as

$$\tilde{R}(t) \rightarrow I_3 \quad \text{as } t \rightarrow \infty \quad (3-8)$$

and

$$\|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (3-9)$$

Since $\tilde{q}(t)$ is a unit quaternion, (3-4), (3-5) and (3-8) can be used to quantify the rotation tracking objective as the desire to regulate $\tilde{q}_v(t)$ as

$$\|\tilde{q}_v(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (3-10)$$

The subsequent section will target the control development based on the objectives in (3-9) and (3-10).

3.3 Control Development

3.3.1 Open-Loop Error System

The actual angular velocity of the camera expressed in \mathcal{F} is defined as $\omega_c(t) \in \mathbb{R}^3$, the desired angular velocity of the camera expressed in \mathcal{F}_d is defined as $\omega_{cd}(t) \in \mathbb{R}^3$, and the relative angular velocity of the camera with respect to \mathcal{F}_d

expressed in \mathcal{F} is defined as $\tilde{\omega}_c(t) \in \mathbb{R}^3$ where

$$\tilde{\omega}_c = \omega_c - \tilde{R}\omega_{cd}. \quad (3-11)$$

The camera angular velocities can be related to the time derivatives of $q(t)$ and $q_d(t)$ as [23]

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_v^T \\ q_0 I_3 + q_v^\times \end{bmatrix} \omega_c \quad (3-12)$$

and

$$\begin{bmatrix} \dot{q}_{0d} \\ \dot{q}_{vd} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_{vd}^T \\ q_{0d} I_3 + q_{vd}^\times \end{bmatrix} \omega_{cd}, \quad (3-13)$$

respectively.

As stated in Remark 3 in Chen et al. [10], a sufficiently smooth function can be used to fit the sequence of feature points to generate the desired trajectory $p_{di}(t)$; hence, it is assumed that $p_{ed}(t)$ and $\dot{p}_{ed}(t)$ are bounded functions of time. In practice, the a priori developed smooth functions $\alpha_{di}(t)$, $R_d(t)$, and $\frac{x_{fd}(t)}{d^*}$ can be constructed as bounded functions with bounded time derivatives. Based on the assumption that $R_d(t)$ is a bounded first order differentiable function with a bounded derivative, the algorithm for computing quaternions in [14] can be used to conclude that $(q_{0d}(t), q_{vd}^T(t))^T$ are bounded first order differentiable functions with a bounded derivative; hence, $(q_{0d}(t), q_{vd}^T(t))^T$ and $(\dot{q}_{0d}(t), \dot{q}_{vd}^T(t))^T$ are bounded. In the subsequent tracking control development, the desired signals $\dot{p}_{ed}(t)$ and $\dot{q}_{vd}(t)$ will be used as feedforward control terms. To avoid the computational singularity in $\theta_d(t)$, the desired trajectory in [10] was generated by carefully choosing the smooth function such that the workspace is limited to $(-\pi, \pi)$. Unlike [10], the use of the quaternion alleviates the restriction on the desired trajectory $p_d(t)$.

From (3-13), the signal $\omega_{cd}(t)$ can be calculated as

$$\omega_{cd} = 2(q_{0d}\dot{q}_{vd} - q_{vd}\dot{q}_{0d}) - 2q_{vd}^\times\dot{q}_{vd}, \quad (3-14)$$

where $(q_{0d}(t), q_{vd}^T(t))^T$, $(\dot{q}_{0d}(t), \dot{q}_{vd}^T(t))^T$ are bounded, so $\omega_{cd}(t)$ is also bounded.

Based on (3-4), (3-5), (3-12) and (3-13), the open-loop rotation error system can be developed as

$$\dot{\tilde{q}} = \frac{1}{2} \begin{bmatrix} -\tilde{q}_v^T \\ \tilde{q}_0 I_3 + \tilde{q}_v^\times \end{bmatrix} (\omega_c - \tilde{R}\omega_{cd}), \quad (3-15)$$

where $\tilde{q}(t) = (\tilde{q}_0(t), \tilde{q}_v^T(t))^T$.

By using (2-5), (2-6), (3-6), (3-11), and the fact that [95]

$$\dot{\bar{m}}_i = -v_c + \bar{m}_i^\times \omega_c, \quad (3-16)$$

where $v_c(t) \in \mathbb{R}^3$ denotes the actual linear velocity of the camera expressed in \mathcal{F} , the open-loop translation error system can be derived as [10]

$$z_i^* \dot{e} = -\alpha_i L_v v_c + (L_v m_i^\times \omega_c - \dot{p}_{ed}) z_i^*, \quad (3-17)$$

where $L_v(t) \in \mathbb{R}^{3 \times 3}$ are defined as

$$L_v = \left(A - \begin{bmatrix} 0 & 0 & u_0 \\ 0 & 0 & v_0 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} 1 & 0 & -\frac{x_i}{z_i} \\ 0 & 1 & -\frac{y_i}{z_i} \\ 0 & 0 & 1 \end{bmatrix}. \quad (3-18)$$

The auxiliary term $L_v(t)$ is an invertible upper triangular matrix.

3.3.2 Closed-Loop Error System

Based on the open-loop rotation error system in (3-15) and the subsequent Lyapunov-based stability analysis, the angular velocity controller is designed as

$$\omega_c = -K_\omega (I_3 + \tilde{q}_v^\times)^{-1} \tilde{q}_v + \tilde{R}\omega_{cd} = -K_\omega \tilde{q}_v + \tilde{R}\omega_{cd}, \quad (3-19)$$

where $K_\omega \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. See Appendix B for the proof that $(I_3 + \tilde{q}_v^\times)^{-1} \tilde{q}_v = \tilde{q}_v$. Based on (3-11), (3-15) and

(3–19), the rotation closed-loop error system can be determined as

$$\begin{aligned}\dot{\tilde{q}}_0 &= \frac{1}{2}\tilde{q}_v^T K_\omega \tilde{q}_v \\ \dot{\tilde{q}}_v &= -\frac{1}{2}(\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v.\end{aligned}\tag{3–20}$$

The contribution of this chapter is the development of the quaternion-based rotation tracking controller. Several other homography-based translation controllers could be combined with the developed rotation controller. For completeness, the following development illustrates how the translation controller and adaptive update law in [10] can be used to complete the six DOF tracking result.

Based on (3–17), the translation control input $v_c(t)$ is designed as

$$v_c = \frac{1}{\alpha_i} L_v^{-1} (K_v e + \hat{z}_i^* (L_v m_i^\times \omega_c - \dot{p}_{ed})),\tag{3–21}$$

where $K_v \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. In (3–21), the parameter estimate $\hat{z}_i^*(t) \in \mathbb{R}$ for the unknown constant z_i^* is defined as

$$\dot{\hat{z}}_i^* = \gamma e^T (L_v m_i^\times \omega_c - \dot{p}_{ed}),\tag{3–22}$$

where $\gamma \in \mathbb{R}$ denotes a positive constant adaptation gain. The controller in (3–21) does not exhibit a singularity since $L_v(t)$ is invertible and $\alpha_i(t) > 0$. From (3–17) and (3–21), the translation closed-loop error system can be listed as

$$z_i^* \dot{e} = -K_v e + (L_v m_i^\times \omega_c - \dot{p}_{ed}) \tilde{z}_i^*,\tag{3–23}$$

where $\tilde{z}_i^*(t) \in \mathbb{R}$ denotes the following parameter estimation error:

$$\tilde{z}_i^* = z_i^* - \hat{z}_i^*.\tag{3–24}$$

3.3.3 Stability Analysis

Theorem 3.1: The controller given in (3–19) and (3–21), along with the adaptive update law in (3–22) ensures global asymptotic tracking in the sense that

$$\|\tilde{q}_v(t)\| \rightarrow 0, \quad \|e(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty. \quad (3-25)$$

Proof: Let $V(t) \in \mathbb{R}$ denote the following differentiable non-negative function (i.e., a Lyapunov candidate):

$$V = \tilde{q}_v^T \tilde{q}_v + (1 - \tilde{q}_0)^2 + \frac{\tilde{z}_i^*}{2} e^T e + \frac{1}{2\gamma} \tilde{z}_i^{*2}. \quad (3-26)$$

The time-derivative of $V(t)$ can be determined as

$$\begin{aligned} \dot{V} &= 2\tilde{q}_v^T \dot{\tilde{q}}_v + 2(1 - \tilde{q}_0)(-\dot{\tilde{q}}_0) + \tilde{z}_i^* e^T \dot{e} + \frac{1}{\gamma} \tilde{z}_i^* \dot{\tilde{z}}_i^* \\ &= -\tilde{q}_v^T (\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v - (1 - \tilde{q}_0) \tilde{q}_v^T K_\omega \tilde{q}_v \\ &\quad + e^T (-K_v e + (L_v m_i^\times \omega_c - \dot{p}_{ed}) \tilde{z}_i^*) - \tilde{z}_i^* e^T (L_v m_i^\times \omega_c - \dot{p}_{ed}) \\ &= -\tilde{q}_v^T K_\omega (\tilde{q}_0 I_3 + \tilde{q}_v^\times + (1 - \tilde{q}_0) I_3) \tilde{q}_v - e^T K_v e \\ &= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e, \end{aligned} \quad (3-27)$$

where (3–20) and (3–22)-(3–24) were utilized. It can be seen from (3–27) that $\dot{V}(t)$ is negative semi-definite.

Based on (3–26) and (3–27), $e(t)$, $\tilde{q}_v(t)$, $\tilde{q}_0(t)$, $\tilde{z}_i^*(t) \in \mathcal{L}_\infty$ and $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_2$. Since $\tilde{z}_i^*(t) \in \mathcal{L}_\infty$, it is clear from (3–24) that $\hat{z}_i^*(t) \in \mathcal{L}_\infty$. Based on the fact that $e(t) \in \mathcal{L}_\infty$, (2–3), (2–6), (3–6) and (3–7) can be used to prove that $m_i(t) \in \mathcal{L}_\infty$. Since $m_i(t) \in \mathcal{L}_\infty$, (3–18) implies that $L_v(t)$, $L_v^{-1}(t) \in \mathcal{L}_\infty$. Based on the fact that $\tilde{q}_v(t)$, $\tilde{q}_0(t) \in \mathcal{L}_\infty$, (3–19) can be used to prove that $\tilde{\omega}_c(t) \in \mathcal{L}_\infty$. Since $\tilde{\omega}_c(t) \in \mathcal{L}_\infty$ and $\omega_{cd}(t)$ is a bounded function, (3–11) can be used to conclude that $\omega_c(t) \in \mathcal{L}_\infty$. Since $\hat{z}_i^*(t)$, $e(t)$, $\tilde{\omega}_c(t)$, $m_i(t)$, $L_v(t)$, $L_v^{-1}(t) \in \mathcal{L}_\infty$ and $\dot{p}_{ed}(t)$ is assumed to be

feature points on π , π^* and π_d can be expressed in terms of \mathcal{I} , respectively, as

$$\begin{aligned} \bar{m}_i(t) &\triangleq \begin{bmatrix} x_i(t) & y_i(t) & z_i(t) \end{bmatrix}^T & \bar{m}_i^* &\triangleq \begin{bmatrix} x_i^* & y_i^* & z_i^* \end{bmatrix}^T \\ \bar{m}_{di}(t) &\triangleq \begin{bmatrix} x_{di}(t) & y_{di}(t) & z_{di}(t) \end{bmatrix}^T \end{aligned} \quad (3-28)$$

under the standard assumption that the distances from the origin of \mathcal{I} to the feature points remains positive (i.e., $z_i(t)$, z_i^* , $z_{di}(t) > \varepsilon$ where ε denotes an arbitrarily small positive constant). Orthogonal coordinate systems \mathcal{F} , \mathcal{F}^* , and \mathcal{F}_d are attached to the planes π , π^* , and π_d , respectively. To relate the coordinate systems, let $R(t)$, R^* , $R_d(t) \in SO(3)$ denote the orientations of \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d with respect to \mathcal{I} , respectively, and let $x_f(t)$, x_f^* , $x_{fd}(t) \in \mathbb{R}^3$ denote the respective translation vectors expressed in \mathcal{I} . As also illustrated in Figure 3-1, $n^* \in \mathbb{R}^3$ denotes the constant unit normal to the plane π^* expressed in \mathcal{I} , and $s_i \in \mathbb{R}^3$ denotes the constant coordinates of the i -th feature point expressed in the corresponding coordinate frames \mathcal{F} , \mathcal{F}^* , and \mathcal{F}_d .

From the geometry between the coordinate frames depicted in Figure 3-1, the following relationships can be developed

$$\bar{m}_i = \bar{x}_f + \bar{R}\bar{m}_i^* \quad \bar{m}_{di} = \bar{x}_{fd} + \bar{R}_d\bar{m}_i^*, \quad (3-29)$$

where $\bar{R}(t)$, $\bar{R}_d(t) \in SO(3)$ and $\bar{x}_f(t)$, $\bar{x}_{fd}(t) \in \mathbb{R}^3$ denote new rotation and translation variables, respectively, defined as

$$\begin{aligned} \bar{R} &= R(R^*)^T & \bar{R}_d &= R_d(R^*)^T \\ \bar{x}_f &= x_f - \bar{R}x_f^* & \bar{x}_{fd} &= x_{fd} - \bar{R}_dx_{fd}^*. \end{aligned} \quad (3-30)$$

Similar to the camera-in-hand configuration, the relationships in (3-29) can be expressed as

$$\bar{m}_i = \left(\bar{R} + \frac{\bar{x}_f}{d^*}n^{*T} \right) \bar{m}_i^* \quad \bar{m}_{di} = \left(\bar{R}_d + \frac{\bar{x}_{fd}}{d^*}n^{*T} \right) \bar{m}_i^*.$$

The rotation matrices $\bar{R}(t)$, $\bar{R}_d(t)$ and the depth ratios $\alpha_i(t)$ and $\alpha_{di}(t)$ can be obtained as described in Section 2.2. The constant rotation matrix R^* can be obtained a priori using various methods (e.g., a second camera, Euclidean measurements) [10]. Based on (3–30), $R(t)$ and $R_d(t)$ can be determined.

The orientations of \mathcal{F}^* with respect to \mathcal{F} and \mathcal{F}_d can be expressed as $R^T(t)R^*$ and $R_d^T(t)R^*$ respectively. To quantify the error between the actual and desired plane orientations, the mismatch between rotation matrices $R^T(t)R^*$ and $R_d^T(t)R^*$ is defined as

$$\tilde{R} = R^T R^* [R_d^T R^*]^T = R^T R_d. \quad (3-31)$$

Similar to the development for the camera-in-hand configuration, the following expression can be obtained:

$$\tilde{R} = (\tilde{q}_0^2 - \tilde{q}_v^T \tilde{q}_v) I_3 + 2\tilde{q}_v \tilde{q}_v^T - 2\tilde{q}_0 \tilde{q}_v^\times, \quad (3-32)$$

where the error quaternion $(\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ is defined as

$$\tilde{q}_0 = q_0 q_{0d} + q_v^T q_{vd} \quad (3-33)$$

$$\tilde{q}_v = q_{0d} q_v - q_0 q_{vd} + q_v^\times q_{vd},$$

where $(q_0(t), q_v^T(t))^T$ and $(q_{0d}(t), q_{vd}^T(t))^T$ are unit quaternions computed from the rotation matrices $R^T(t)R^*$ and $R_d^T(t)R^*$ following the method given in [14].

3.4.2 Control Formulation

By expressing the translation vectors, angular velocity and linear velocity in the body-fixed coordinate frame \mathcal{F} , and by defining rotation matrices $R^T(t)R^*$ and $R_d^T(t)R^*$, the control objective in this section can be formulated in the same manner as done for the camera-in-hand configuration problem. So, the rotation and translation errors can be described the same as those for the camera-in-hand configuration problem. The actual angular velocity of the object expressed in \mathcal{F} is

defined as $\omega_e(t) \in \mathbb{R}^3$, the desired angular velocity of the object expressed in \mathcal{F}_d is defined as $\omega_{ed}(t) \in \mathbb{R}^3$, and the relative angular velocity of the object with respect to \mathcal{F}_d expressed in \mathcal{F} is defined as $\tilde{\omega}_e(t) \in \mathbb{R}^3$ where

$$\tilde{\omega}_e = \omega_e - \tilde{R}\omega_{ed}, \quad (3-34)$$

where $\omega_{ed}(t)$ has the same form as $\omega_{cd}(t)$ in (3-14). The open-loop rotation error system is

$$\dot{\tilde{q}} = \frac{1}{2} \begin{bmatrix} -\tilde{q}_v^T \\ \tilde{q}_0 I_3 + \tilde{q}_v^\times \end{bmatrix} (\omega_e - \tilde{R}\omega_{ed}) \quad (3-35)$$

and the translation error system is [10]

$$z_i^* \dot{e} = \alpha_i L_v R (v_e + \omega_e^\times s_i) - z_i^* \dot{p}_{ed}, \quad (3-36)$$

where $v_e(t) \in \mathbb{R}^3$ denotes the linear velocity of the object expressed in \mathcal{F} .

Based on the open-loop error systems (3-35) and (3-36), and the subsequent stability analysis, the angular and linear camera velocity control inputs for the object are defined as

$$\omega_e = -K_\omega \tilde{q}_v + \tilde{R}\omega_{ed} \quad (3-37)$$

$$v_e = -\frac{1}{\alpha_i} R^T L_v^{-1} (K_v e - \hat{z}_i^* \dot{p}_{ed}) - \omega_e^\times \hat{s}_i. \quad (3-38)$$

In (3-37) and (3-38), $K_\omega, K_v \in \mathbb{R}^{3 \times 3}$ denote diagonal matrices of positive constant control gains, the parameter estimates $\hat{z}_i^*(t) \in \mathbb{R}$, $\hat{s}_i(t) \in \mathbb{R}^3$ for the unknown constants $z_i^*(t)$ and $s_i(t)$ are generated according to the following adaptive update laws

$$\dot{\hat{z}}_i^* = -\gamma e^T \dot{p}_{ed} \quad (3-39)$$

$$\dot{\hat{s}}_i = -\alpha_i \Gamma \omega_e^\times R^T L_v^T e, \quad (3-40)$$

where $\Gamma \in \mathbb{R}^{3 \times 3}$ denotes a positive constant diagonal adaptation gain matrix.

From (3–35) and (3–37), the rotation closed-loop error system can be determined as

$$\begin{aligned}\dot{\tilde{q}}_0 &= \frac{1}{2}\tilde{q}_v^T K_\omega \tilde{q}_v \\ \dot{\tilde{q}}_v &= -\frac{1}{2}(\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v.\end{aligned}\tag{3–41}$$

Based on (3–36) and (3–38), the translation closed-loop error system is given as

$$z_i^* \dot{e} = -K_v e - \tilde{z}_i^* \dot{p}_{ed} + \alpha_i L_v R \omega_e^\times \tilde{s}_i,\tag{3–42}$$

where the parameter estimation error signals $\tilde{z}_i^*(t) \in \mathbb{R}$ and $\tilde{s}_i(t) \in \mathbb{R}^3$ are defined as

$$\tilde{z}_i^* = z_i^* - \hat{z}_i^* \quad \tilde{s}_i = s_i - \hat{s}_i.\tag{3–43}$$

Theorem 3.2: The controller given in (3–37) and (3–38), along with the adaptive update laws in (3–39) and (3–40) ensure global asymptotic tracking in the sense that

$$\|\tilde{q}_v(t)\| \rightarrow 0, \quad \|e(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty.\tag{3–44}$$

Proof: Let $V(t) \in \mathbb{R}$ denote the following differentiable non-negative definite function (i.e., a Lyapunov candidate):

$$V = \tilde{q}_v^T \tilde{q}_v + (1 - \tilde{q}_0)^2 + \frac{z_i^*}{2} e^T e + \frac{1}{2\gamma} \tilde{z}_i^{*2} + \frac{1}{2} \tilde{s}_i^T \Gamma^{-1} \tilde{s}_i.\tag{3–45}$$

The time-derivative of $V(t)$ can be determined as

$$\begin{aligned}
\dot{V} &= 2\tilde{q}_v^T \dot{\tilde{q}}_v + 2(1 - \tilde{q}_0)(-\dot{\tilde{q}}_0) + z_i^* e^T \dot{e} + \frac{1}{\gamma} \tilde{z}_i^* \dot{\tilde{z}}_i^* + \tilde{s}_i^T \Gamma^{-1} \dot{\tilde{s}}_i \\
&= -\tilde{q}_v^T (\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v - (1 - \tilde{q}_0) \tilde{q}_v^T K_\omega \tilde{q}_v \\
&\quad + e^T (-K_v e - \tilde{z}_i^* \dot{p}_{ed} + \alpha_i L_v R \omega_e^\times \tilde{s}_i) + \tilde{z}_i^* e^T \dot{p}_{ed} + \alpha_i \tilde{s}_i^T \omega_e^\times R^T L_v^T e \\
&= -\tilde{q}_v^T K_\omega (\tilde{q}_0 I_3 + \tilde{q}_v^\times + (1 - \tilde{q}_0) I_3) \tilde{q}_v \\
&\quad - e^T K_v e + \alpha_i e^T L_v R \omega_e^\times \tilde{s}_i + \alpha_i \tilde{s}_i^T \omega_e^\times R^T L_v^T e \\
&= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e + \alpha_i (e^T A_e L_v R \omega_e^\times \tilde{s}_i)^T + \alpha_i \tilde{s}_i^T \omega_e^\times R^T L_v^T A_e^T e \\
&= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e + \alpha_i \tilde{s}_i^T (\omega_e^\times)^T R^T L_v^T A_e^T e + \alpha_i \tilde{s}_i^T \omega_e^\times R^T L_v^T A_e^T e \\
&= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e + \alpha_i \tilde{s}_i^T (-\omega_e^\times) R^T L_v^T A_e^T e + \alpha_i \tilde{s}_i^T \omega_e^\times R^T L_v^T A_e^T e \\
&= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e, \tag{3-46}
\end{aligned}$$

where (3-34)-(3-43) were utilized. From (3-45) and (3-46), signal chasing arguments described in the previous section can be used to conclude that the control inputs and all the closed-loop signals are bounded. Barbalat's Lemma [96] can then be used to prove the result given in (3-44).

3.5 Simulation Results

A numerical simulation was performed to illustrate the performance of the tracking controller given in (3-19), (3-21), and the adaptive update law in (3-22). In this simulation, the developed tracking controller aims enable the control object to track the desired trajectory encoded by a sequence of images and rotate more than 360° (see Fig. 3-5).

The camera is assumed to view an object with four coplanar feature points with the following Euclidean coordinates (in [m]):

$$\begin{aligned} O_1 &= \begin{bmatrix} 0.15 & 0.15 & 0 \end{bmatrix}^T & O_2 &= \begin{bmatrix} 0.15 & -0.15 & 0 \end{bmatrix}^T \\ O_3 &= \begin{bmatrix} -0.15 & 0.15 & 0 \end{bmatrix}^T & O_4 &= \begin{bmatrix} -0.15 & -0.15 & 0 \end{bmatrix}^T. \end{aligned} \quad (3-47)$$

The time-varying desired image trajectory was generated by the kinematics of the feature point plane where the desired linear and angular velocities were selected as

$$\begin{aligned} v_{cd} &= \begin{bmatrix} 0.1 \sin(t) & 0.1 \sin(t) & 0 \end{bmatrix} [m/s] \\ \omega_{cd} &= \begin{bmatrix} 0 & 0 & 1.5 \end{bmatrix} [rad/s]. \end{aligned}$$

The initial and desired image-space coordinates were artificially generated. For this example, consider an orthogonal coordinate frame \mathcal{I} with the z -axis opposite to n^* (see Figure 2-1) with the x -axis and y -axis on the plane π . The rotation matrices R_1 between \mathcal{F} and \mathcal{I} , and R_2 between \mathcal{F}^* and \mathcal{I} were set as

$$R_1 = R_x(120^\circ)R_y(-20^\circ)R_z(-80^\circ) \quad (3-48)$$

$$R_2 = R_x(160^\circ)R_y(30^\circ)R_z(30^\circ), \quad (3-49)$$

where $R_x(\cdot)$, $R_y(\cdot)$ and $R_z(\cdot) \in SO(3)$ denote rotation of angle “ \cdot ” (degrees) along the x -axis, y -axis and z -axis, respectively. The translation vectors x_{f1} and x_{f2} between \mathcal{F} and \mathcal{I} (expressed in \mathcal{F}) and between \mathcal{F}^* and \mathcal{I} (expressed in \mathcal{F}^*), respectively, were selected as

$$x_{f1} = \begin{bmatrix} 0.5 & 0.5 & 4.0 \end{bmatrix}^T \quad (3-50)$$

$$x_{f2} = \begin{bmatrix} 1.0 & 1.0 & 4.5 \end{bmatrix}^T. \quad (3-51)$$

The initial rotation matrix R_3 and translation vector x_{f3} between \mathcal{F}_d and \mathcal{I} were set as

$$R_3 = R_x(240^\circ)R_y(-90^\circ)R_z(-30^\circ) \quad x_{f3} = \begin{bmatrix} 0.5 & 1 & 5.0 \end{bmatrix}^T. \quad (3-52)$$

The initial (i.e., $p_i(0)$) and reference (i.e., p_i^*) image-space coordinates of the four feature points in (3-47) were computed as (in pixels)

$$\begin{aligned} p_1(0) &= \begin{bmatrix} 907.91 & 716.04 & 1 \end{bmatrix}^T & p_2(0) &= \begin{bmatrix} 791.93 & 728.95 & 1 \end{bmatrix}^T \\ p_3(0) &= \begin{bmatrix} 762.84 & 694.88 & 1 \end{bmatrix}^T & p_4(0) &= \begin{bmatrix} 871.02 & 683.25 & 1 \end{bmatrix}^T \\ p_1^* &= \begin{bmatrix} 985.70 & 792.70 & 1 \end{bmatrix}^T & p_2^* &= \begin{bmatrix} 1043.4 & 881.20 & 1 \end{bmatrix}^T \\ p_3^* &= \begin{bmatrix} 980.90 & 921.90 & 1 \end{bmatrix}^T & p_4^* &= \begin{bmatrix} 922.00 & 829.00 & 1 \end{bmatrix}^T. \end{aligned}$$

The initial (i.e., $p_{di}(0)$) image-space coordinates of the four feature points in (3-47) for generating the desired trajectory were computed as (in pixels)

$$\begin{aligned} p_{d1}(0) &= \begin{bmatrix} 824.61 & 853.91 & 1 \end{bmatrix}^T & p_{d2}(0) &= \begin{bmatrix} 770.36 & 878.49 & 1 \end{bmatrix}^T \\ p_{d3}(0) &= \begin{bmatrix} 766.59 & 790.50 & 1 \end{bmatrix}^T & p_{d4}(0) &= \begin{bmatrix} 819.03 & 762.69 & 1 \end{bmatrix}^T. \end{aligned}$$

The control gains K_ω in (3-19) and K_v in (3-21) and adaptation gain γ in (3-22) were selected as

$$\begin{aligned} K_\omega &= \text{diag}\{3, 3, 3\} & K_v &= \text{diag}\{15, 15, 15\} \\ \gamma &= 0.0002. \end{aligned}$$

The desired and current image-space trajectories of the feature point plane are shown in Figure 3-5 and Figure 3-6, respectively. The feature point plane rotates more than 360° degrees as shown in these two figures. The resulting translation

and rotation errors are plotted in Figure 3–7 and Figure 3–8, respectively. The errors go to zero asymptotically. The desired image-space trajectory (i.e., $p_{di}(t)$) and the current image-space trajectory (i.e., $p_i(t)$) are shown in Figure 3–9 and Figure 3–10, respectively. The tracking error between the current and desired image-space trajectory is shown in Figure 3–11. The Figures 3–7–3–11 show that the current trajectory tracks the desired trajectory asymptotically. The translation and rotation control inputs are shown in Figure 3–12 and Figure 3–13, respectively. The parameter estimate for z_1^* is shown in Figure 3–14.

3.6 Experiment Results

Simulations verified the performance of the tracking controller given in (3–19), (3–21), and the adaptive update law in (3–22). Experiments were then performed to test robustness and performance in the presence of signal noise, measurement error, calibration error, etc. The experiments were performed in a test-bed at the University of Florida for simulation, design and implementation of vision-based control systems. For the test-bed, a 3D environment can be projected onto large monitors or screens and viewed by a physical camera. Communication between the camera and control processing computers and the environment rendering computers allows closed-loop control of the virtual scene.

3.6.1 Experiment Configurations

A block diagram describing the experimental test-bed is provided in Figure 3–2. The test-bed is based on a virtual environment generated by a virtual reality simulator, composed of five workstations and a database server running virtual reality software. This allows multiple instances of the virtual environment to run at the same time. In this way, camera views can be rigidly connected in a mosaic for a large FOV. Alternately, multiple, independent camera views can pursue their own tasks, such as coordinated control of multiple vehicles. The virtual reality simulator

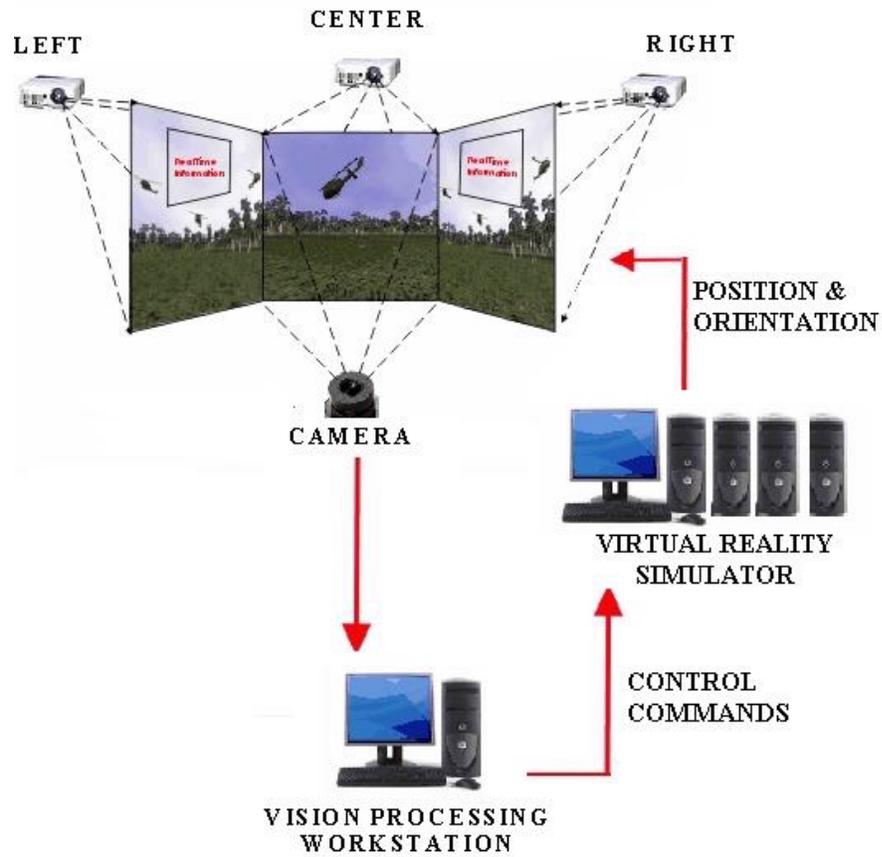


Figure 3-2: Block diagram of the experiment.



Figure 3-3: The Sony XCD-710CR color firewire camera pointed at the virtual environment.

in the experiment is currently capable of displaying three simultaneous displays. A picture of the displays can be seen in Figure 3-3.

The virtual reality simulator utilizes MultiGen-Paradigm's Vega Prime, an OpenGL-based, commercial software package for Microsoft Windows. The virtual environment in the experiment is a recreation of the U.S. Army's urban warfare training ground at Fort Benning, Georgia. The environment has a dense polygon count, detailed textures, high frame rate, and the effects of soft shadows, resulting in very realistic images. A scene from the Fort Benning environment can be seen in Figure 3-4.

The visual sensor in the experiment is a Sony XCD-710CR color firewire camera with a resolution of 1280×768 pixels and fitted with a 12.5mm lens. The camera captures the images on the large screens as shown in Figure 3-3. The images are processed in a vision processing workstation. An application written in C++ acquires images from the camera and process the images to locate and track the feature points (the initial feature points were chosen manually, then the application will identify and track the feature points on its own). The C++ application generates the current and desired pixel coordinates, which can be used to formulate the control command.

In addition to the image processing application, a control command generation application programmed in Matlab also runs in this workstation. The Matlab application communicates data with the image processing application (written in C++) via shared memory buffers. The Matlab application reads the current and desired pixel coordinates from the shared memory buffers, and writes linear and angular camera velocity input into the shared memory buffers. The C++ application writes the current and desired pixel coordinates into the shared memory, and reads camera velocity input from the shared memory buffer. The linear and angular camera velocity control input are sent from the vision processing workstation to the virtual

reality simulator via a TCP socket connection. This development makes extensive use of Intel’s Open Source Computer Vision (OpenCV) Library (see Bradski [97]) and the GNU Scientific Library (GSL) (see Galassi et al. [98]).



Figure 3–4: Virtual reality environment example: a virtual recreation of the US Army’s urban warfare training ground at Fort Benning.

Algorithms, such as the Homography decomposition, are implemented as if the virtual environment is a true 3D scene which the physical camera is viewing. Of course, the camera does not look at the 3D scene directly. The camera views consist of a 3D scene that are projected onto a 2D plane, then projected onto the image plane. That is, the projective homography needed for control exists between the *on-screen* current image and the *on-screen* goal image, but what are given are the camera views of the *on-screen* images. Thus, there exists an additional transform action between the points on the screen and the points in the camera image. A screen-camera calibration matrix G_{cs} can be used to describe this transformation relationship.

Every point on the screen corresponds to only one point in the image. Thus, the constant matrix G_{cs} is a homography and can be determined through a calibration procedure, and effectively replaces the standard calibration of the

physical camera. In the experiment, this matrix is determined to be

$$G_{cs} = \begin{bmatrix} 0.9141 & 0.0039 & -90.9065 \\ -0.0375 & 0.9358 & -50.7003 \\ 0 & 0 & 1 \end{bmatrix}.$$

In addition to G_{cs} , the camera calibration matrix A , corresponding to the virtual camera within Vega Prime, is still required. This matrix can be determined from the settings of the virtual reality program. In this experiment, A was determined to be

$$A = \begin{bmatrix} 1545.1 & 0 & 640 \\ 0 & 1545.1 & 512 \\ 0 & 0 & 1 \end{bmatrix}.$$

3.6.2 Experiment for Tracking

The desired trajectory is in a format of a prerecorded video (a sequence of images). As the view point in the Vega Prime moves (which can be implemented by some chosen velocity functions or manually), the images captured by the camera changes. The pixel coordinates of the features points on the images will be recorded as the desired trajectory. The control objective in this tracking experiment is to send control command to the virtual reality simulator such that the current pose of the feature points tracks the desired pose. The constant reference image was taken as the first image in the sequence.

The control gains K_ω in (3-19) and K_v in (3-21), and adaptation gain γ in (3-22) were selected as

$$K_\omega = \text{diag}\{0.1, 0.1, 1.5\} \quad K_v = \text{diag}\{0.5, 0.5, 0.5\} \quad \gamma = 0.005.$$

During the experiment, the images from the camera are processed with a frame rate of approximately 20 frames/second.

The resulting translation and rotation errors are plotted in Figure 3–15 and Figure 3–16. The desired image-space trajectory (i.e., $p_{di}(t)$) is shown in Figure 3–17, and the current image-space trajectory (i.e., $p_i(t)$) is shown in Figure 3–18. The tracking error between the current and desired image-space trajectories is shown in Figure 3–19. The translation and rotation control outputs are shown in Figure 3–20 and Figure 3–21, respectively. The parameter estimate for z_1^* is shown in Figure 3–22.

In the tracking control, the steady-state tracking error is approximately $\begin{bmatrix} 15[\text{pixel}] & 10[\text{pixel}] & 0.01 \end{bmatrix}$. This steady-state error is caused by the image noise and the camera calibration error in the test-bed. To find the tracking control error, two homographies are computed between the reference image and the current image and desired image, respectively. Due to the image noise and camera calibration error, the error is inserted to the two homographies. Then the tracking error obtained from the mismatch between the two homographies will have larger error. Also, the image noise and calibration error inserts error into the derivative of the desired pixel coordinates, which is used as a feedforward term in the tracking controller. Furthermore, communication between the controller and virtual reality system occurs via a TCP socket, introducing some amount of latency into the system. Note that this pixel error represents less than 1.5% of the image dimensions.

In the following regulation experiment, the derivative of the desired pixel coordinates is equal to zero, and only one homography is computed between the current image and desired set image. The influence of the image noise and calibration error is weakened greatly.

3.6.3 Experiment for Regulation

When the desired pose is a constant, the tracking problem becomes a regulation problem. The control objective in this regulation experiment is to send control command to the virtual reality simulator such that the current pose of the features points is regulated to the desired set pose.

In the experiment, the control gains K_ω in (3-19) and K_v in (3-21), and adaptation gain γ in (3-22) were selected as

$$K_\omega = \text{diag}\{0.4, 0.4, 0.9\} \quad K_v = \text{diag}\{0.5, 0.25, 0.25\} \quad \gamma = 0.005.$$

The resulting translation and rotation errors are plotted in Figure 3-23 and Figure 3-24, respectively. The errors go to zero asymptotically. The current image-space trajectory (i.e., $p_i(t)$) is shown in Figure 3-25. The regulation error between the current and desired set image-space pose is shown in Figure 3-26. The translation and rotation control outputs are shown in Figure 3-27 and Figure 3-28, respectively. The parameter estimate for z_1^* is shown in Figure 3-29.

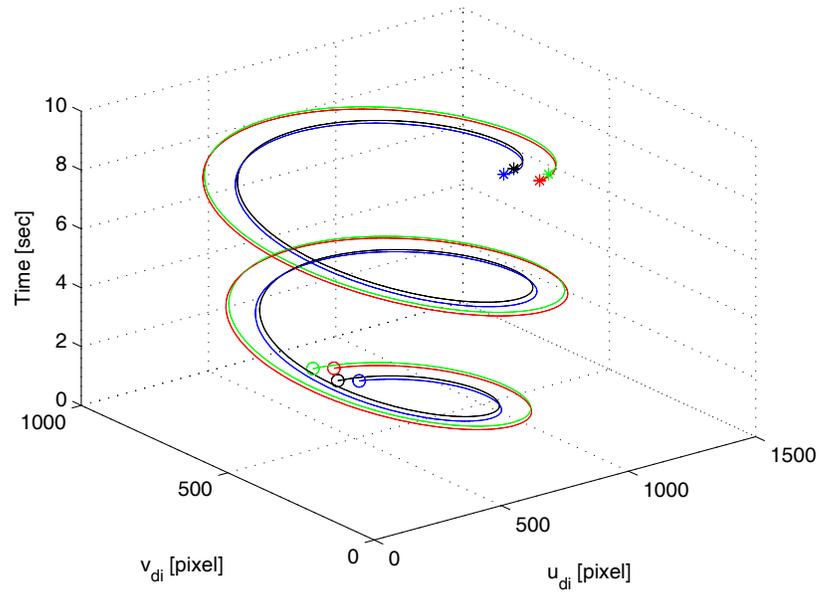


Figure 3–5: Desired image-space coordinates of the four feature points (i.e., $p_d(t)$) in the tracking Matlab simulation shown in a 3D graph. In the figure, “O” denotes the initial image-space positions of the 4 feature points in the desired trajectory, and “*” denotes the corresponding final positions of the feature points.

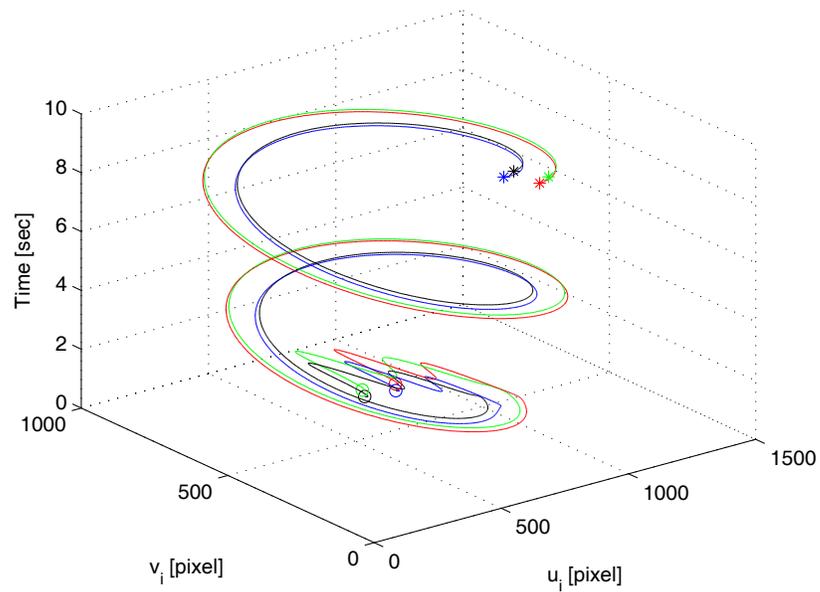


Figure 3–6: Current image-space coordinates of the four feature points (i.e., $p_d(t)$) in the tracking Matlab simulation shown in a 3D graph. In the figure, “O” denotes the initial image-space positions of the 4 feature points, and “*” denotes the corresponding final positions of the feature points.

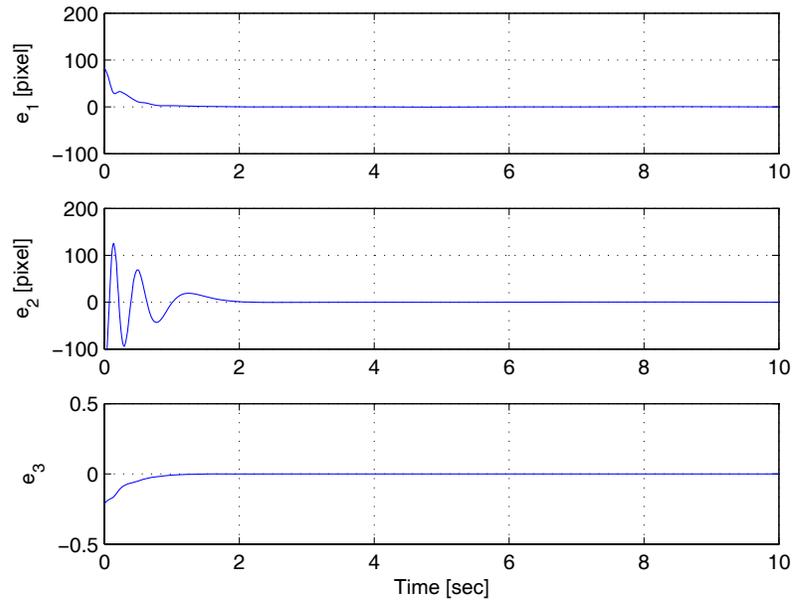


Figure 3–7: Translation error $e(t)$ in the tracking Matlab simulation.

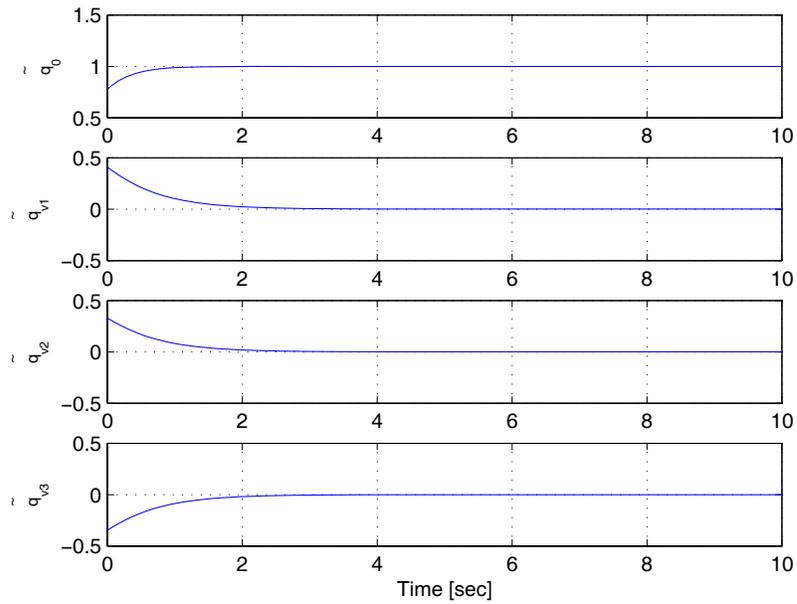


Figure 3–8: Rotation quaternion error $\tilde{q}(t)$ in the tracking Matlab simulation.

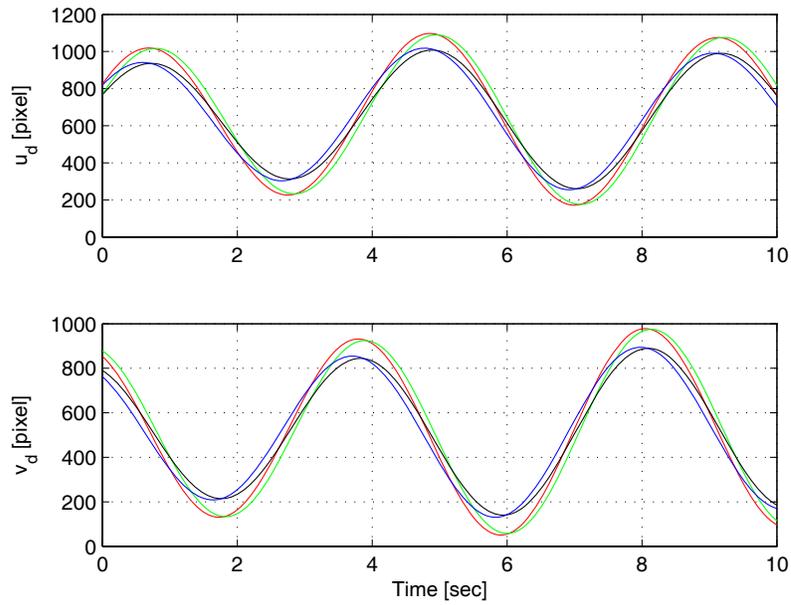


Figure 3–9: Pixel coordinate $p_d(t)$ of the four feature points in a sequence of desired images in the tracking Matlab simulation. The upper figure is for the $u_d(t)$ component and the bottom figure is for the $v_d(t)$ component.

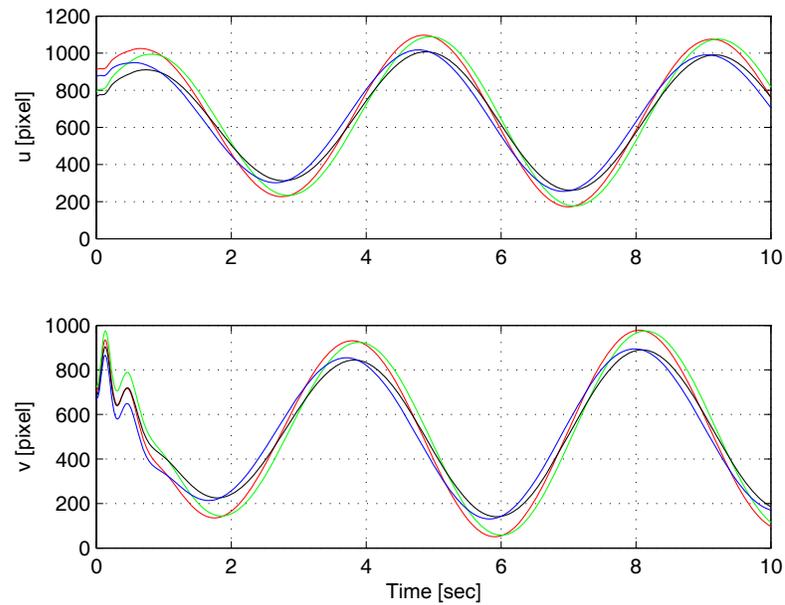


Figure 3–10: Pixel coordinate $p(t)$ of the current pose of the four feature points in the tracking Matlab simulation. The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.

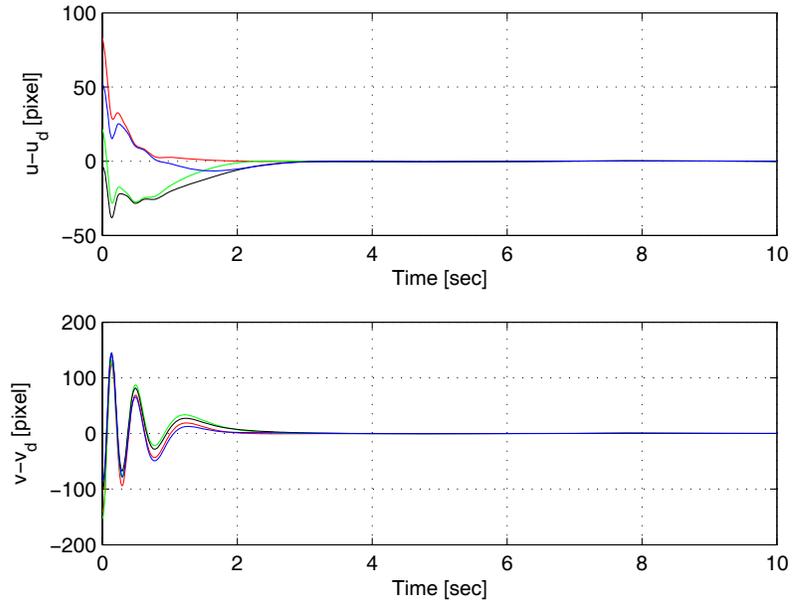


Figure 3–11: Tracking error $p(t) - p_d(t)$ (in pixels) of the four feature points in the tracking Matlab simulation. The upper figure is for the $u(t) - u_d(t)$ component and the bottom figure is for the $v(t) - v_d(t)$ component.

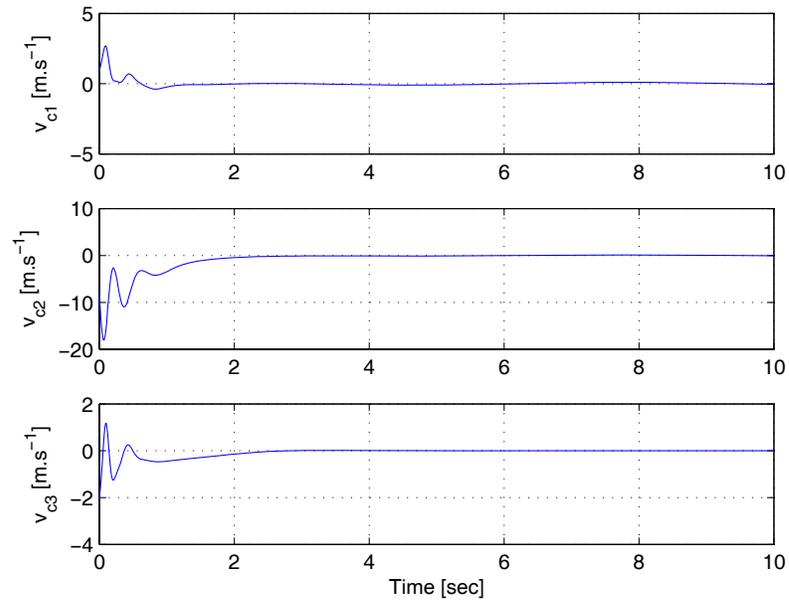


Figure 3–12: Linear camera velocity input $v_c(t)$ in the tracking Matlab simulation.

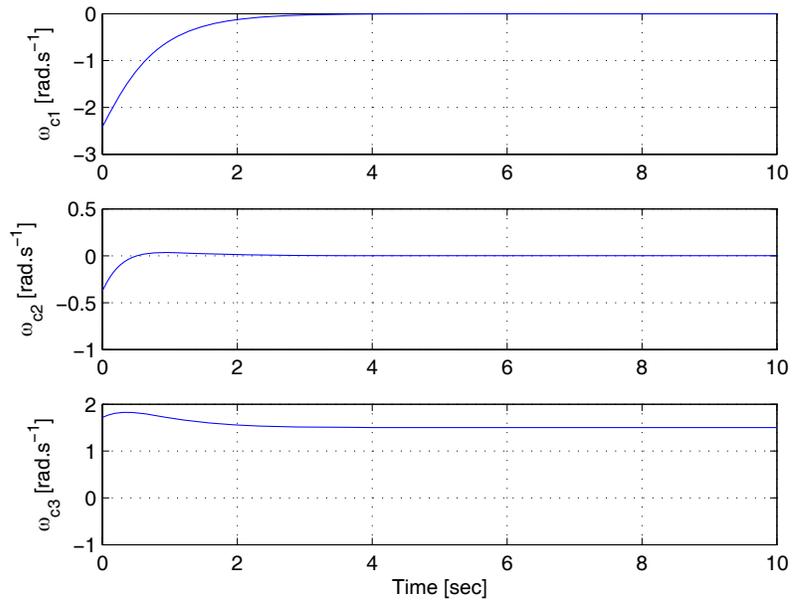


Figure 3–13: Angular camera velocity input $w_c(t)$ in the tracking Matlab simulation.

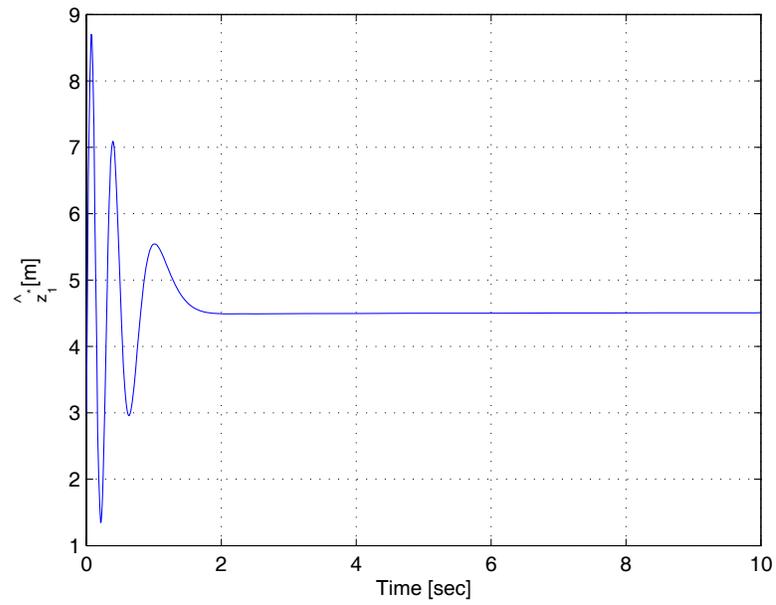


Figure 3–14: Adaptive on-line estimate of z_1^* in the tracking Matlab simulation.

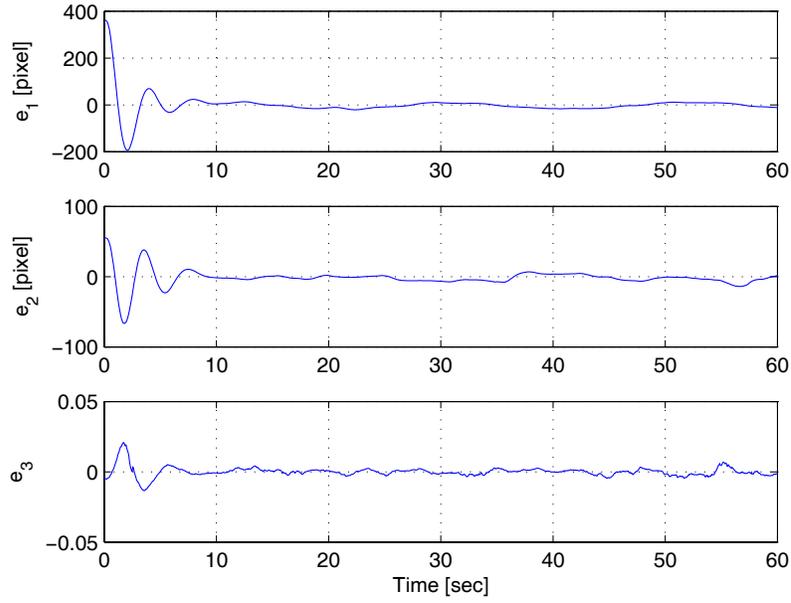


Figure 3–15: Translation error $e(t)$ in the tracking experiment.

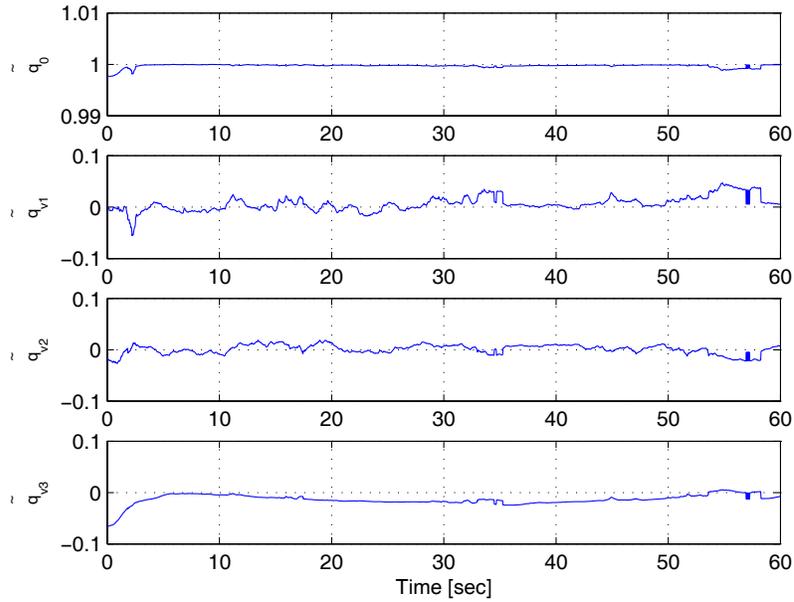


Figure 3–16: Rotation quaternion error $\tilde{q}(t)$ in the tracking experiment.

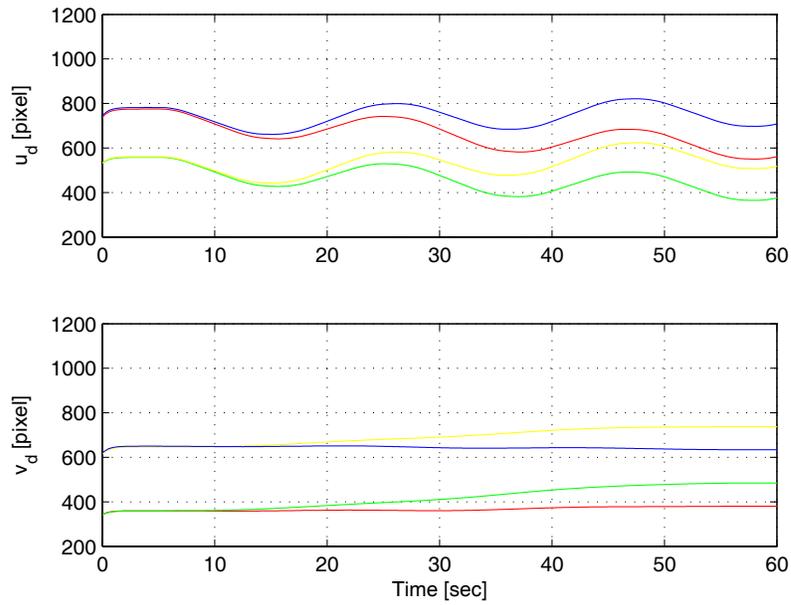


Figure 3–17: Pixel coordinate $p_d(t)$ of the four feature points in a sequence of desired images in the tracking experiment. The upper figure is for the $u_d(t)$ component and the bottom figure is for the $v_d(t)$ component.

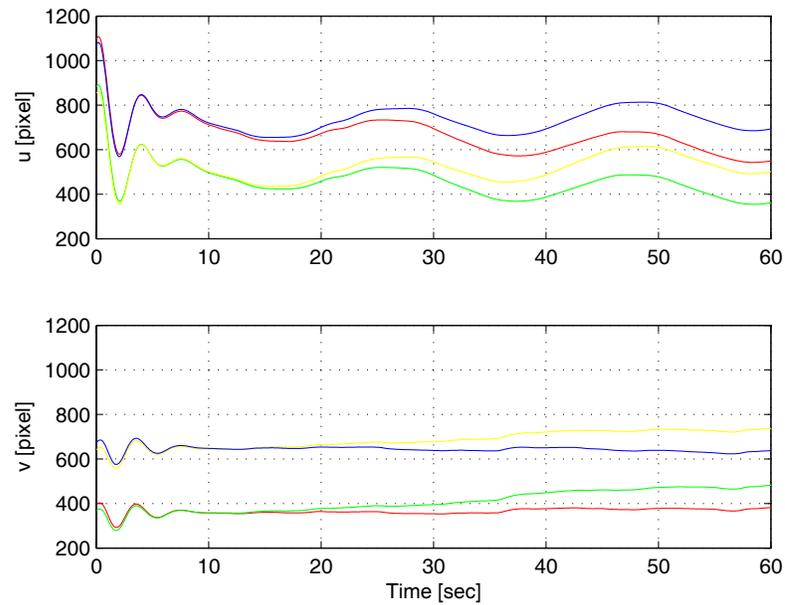


Figure 3–18: Pixel coordinate $p(t)$ of the current pose of the four feature points in the tracking experiment. The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.

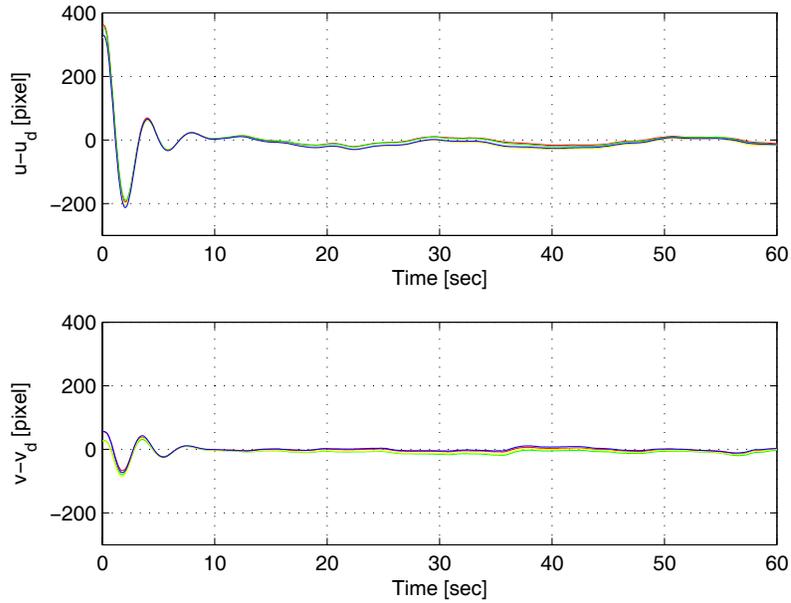


Figure 3–19: Tracking error $p(t) - p_d(t)$ (in pixels) of the four feature points in the tracking experiment. The upper figure is for the $u(t) - u_d(t)$ component and the bottom figure is for the $v(t) - v_d(t)$ component.

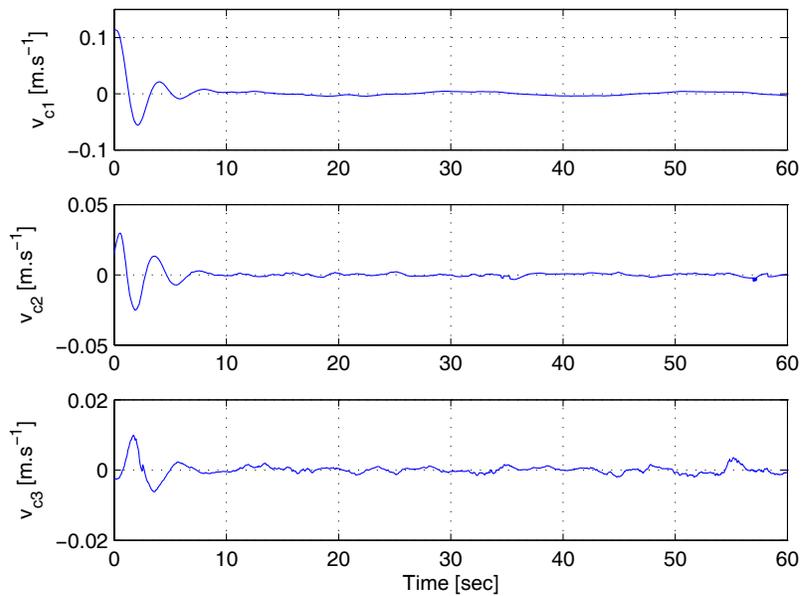


Figure 3–20: Linear camera velocity input $v_c(t)$ in the tracking experiment.

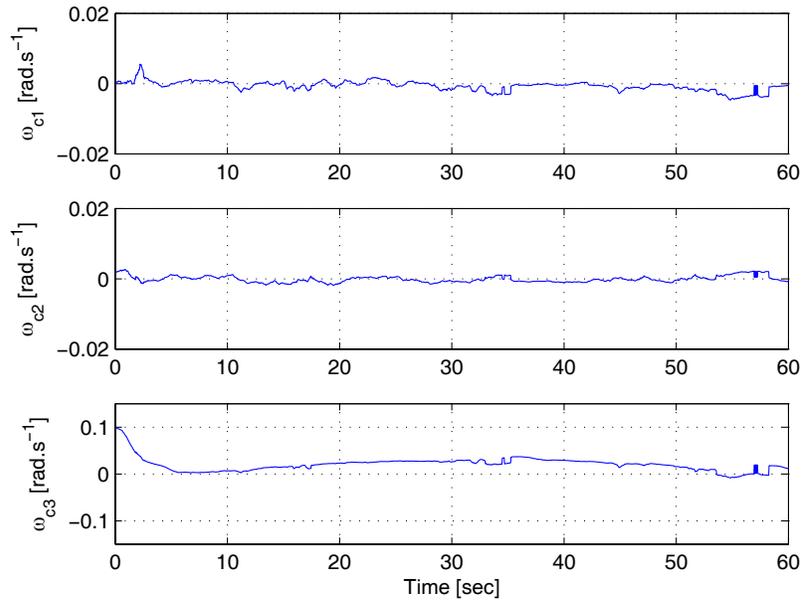


Figure 3–21: Angular camera velocity input $w_c(t)$ in the tracking experiment.

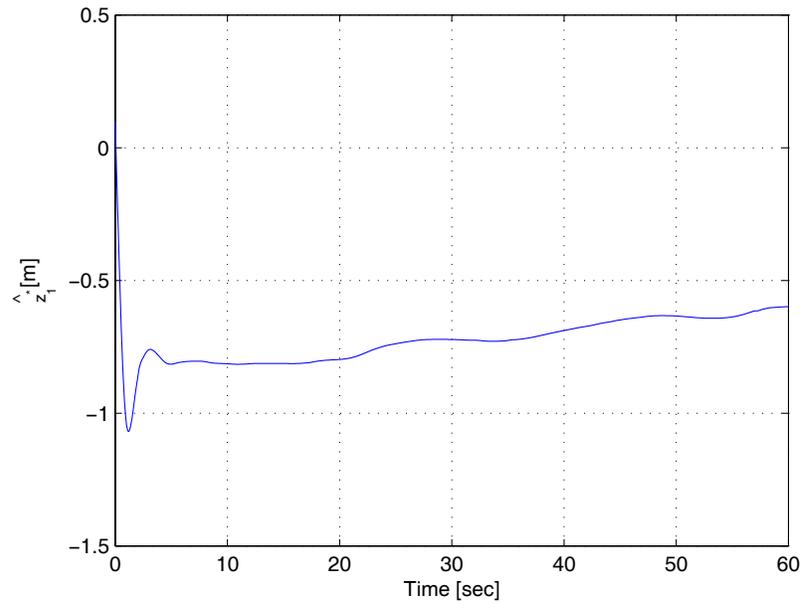


Figure 3–22: Adaptive on-line estimate of z_1^* in the tracking experiment.

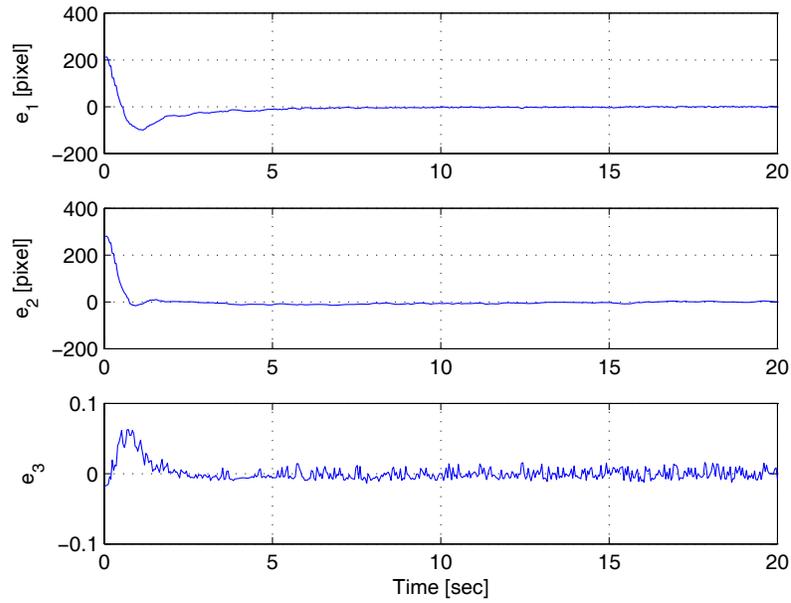


Figure 3–23: Translation error $e(t)$ in the regulation experiment.

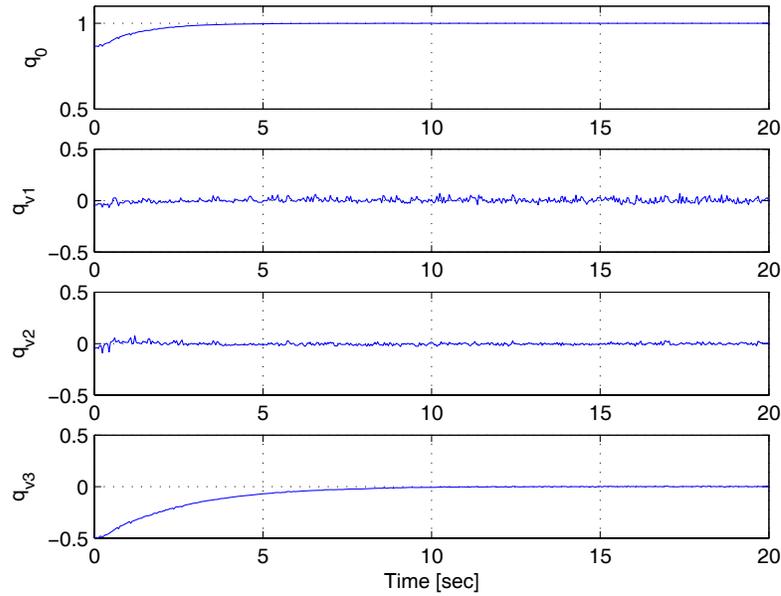


Figure 3–24: Rotation quaternion error $q(t)$ in the regulation experiment.

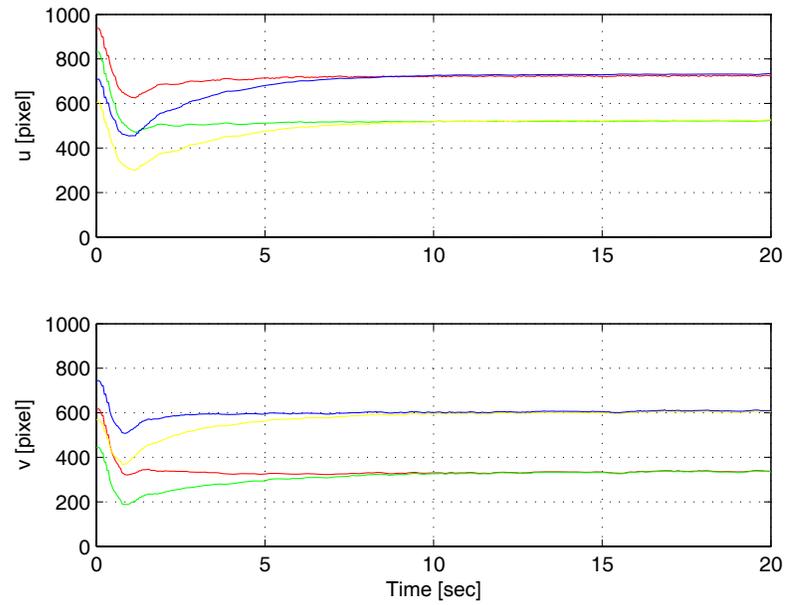


Figure 3–25: Pixel coordinate $p(t)$ (in pixels) of the current pose of the four feature points in the regulation experiment. The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.

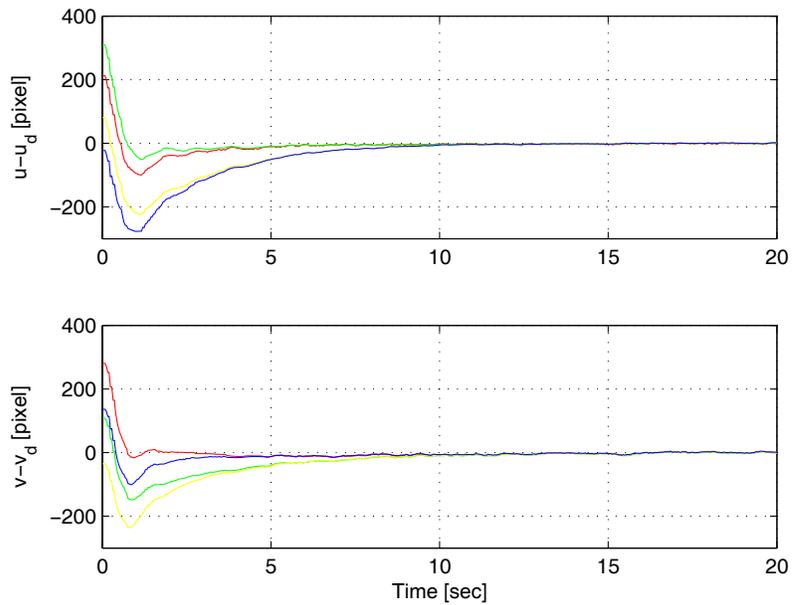


Figure 3–26: Regulation error $p(t) - p^*$ (in pixels) of the four feature points in the regulation experiment. The upper figure is for the $u(t) - u^*(t)$ component and the bottom figure is for the $v(t) - v^*(t)$ component.

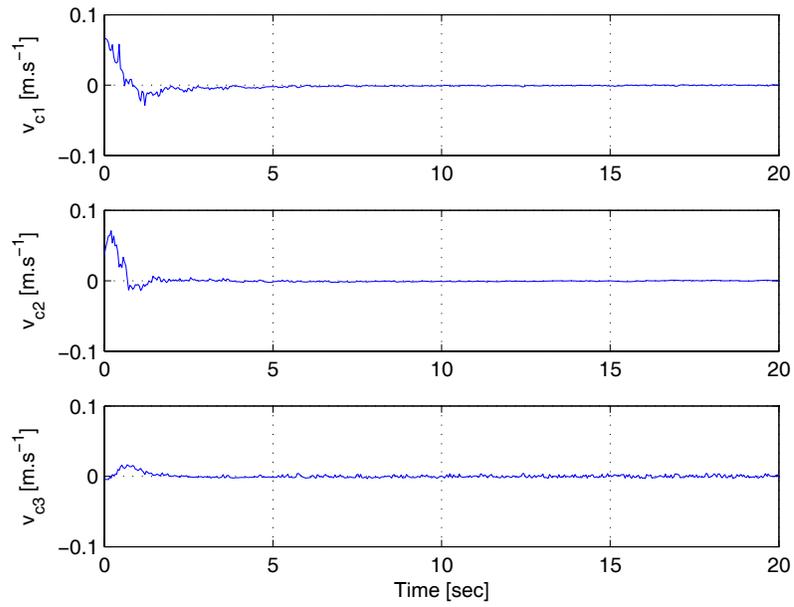


Figure 3–27: Linear camera velocity input $v_c(t)$ in the regulation experiment.

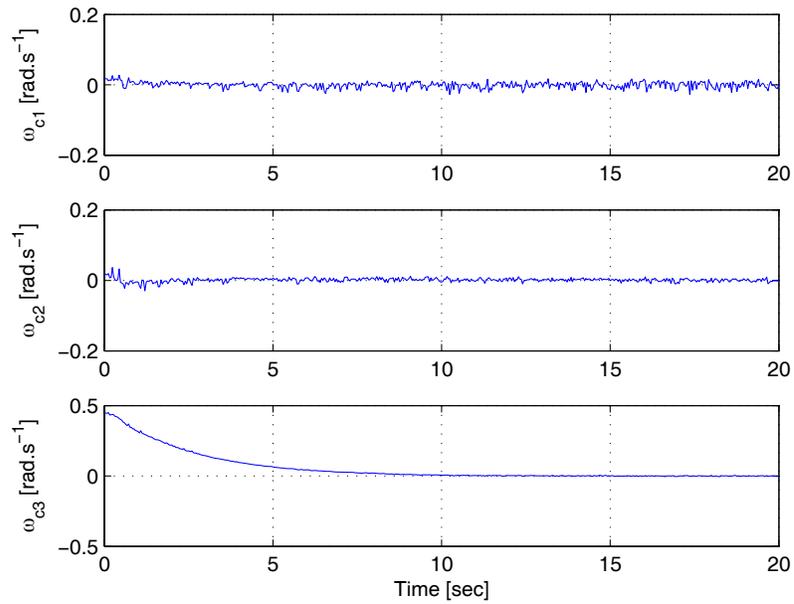


Figure 3–28: Angular camera velocity input $w_c(t)$ in the regulation experiment.

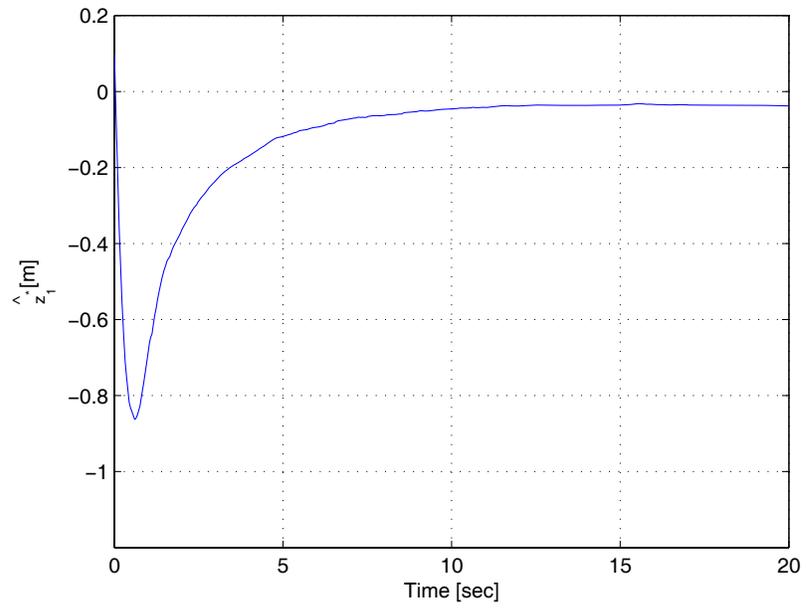


Figure 3–29: Adaptive on-line estimate of z_1^* in the regulation experiment.

CHAPTER 4
COLLABORATIVE VISUAL SERVO TRACKING CONTROL VIA A
DAISY-CHAINING APPROACH

4.1 Introduction

In this chapter, a collaborative trajectory tracking problem is considered for a six DOF rigid-body object (e.g., an autonomous vehicle) identified by a planar patch of feature points. Unlike typical visual servo controllers that require either the camera or the target to remain stationary, a unique aspect of the development in this chapter is that a moving monocular camera (e.g., a camera mounted on an unmanned air vehicle (UAV)) is used to provide feedback to a moving control object. The control objective is for the object to track a desired trajectory that is encoded by a prerecorded video obtained from a fixed camera (e.g., a camera mounted on a satellite, a camera mounted on a building).

Several challenges must be resolved to achieve this unexplored control objective. The relative velocity between the moving planar patch of feature points and the moving camera presents a significant challenge. By using a daisy-chaining approach (e.g., [16–19]), Euclidean homography relationships between different camera coordinate frames and feature point patch coordinate frames are developed. These homographies are used to relate coordinate frames attached to the moving camera, the reference object, the control object, and the object used to record the desired trajectory. Another challenge is that for general six DOF motion by both the camera and the control object, the normal to the planar patch associated with the object is unknown. By decomposing the homography relationships, the normal to the planar patch can be obtained. Likewise, the distance between the moving camera, the moving control object, and the reference object are unknown.

By using the depth ratios obtained from the homography decomposition, the unknown time-varying distance is related to an unknown constant parameter. A Lyapunov-based adaptive estimation law is designed to compensate for the unknown constant parameter. The moving camera could be attached to a remotely piloted vehicle with arbitrary rotations, this requires a parameterization that is valid over a large (possibly unbounded) domain. Additionally, since this work is motivated by problems in the aerospace community, homography-based visual servo control techniques (e.g., [10, 20, 22]) are combined with quaternion-based control methods (e.g., [14, 23, 24]) to facilitate large rotations. By using the quaternion parameterization, the resulting closed-loop rotation error system can be stabilized by a proportional rotation controller combined with a feedforward term that is a function of the desired trajectory.

4.2 Problem Scenario

Over the past decade, a variety of visual servo controllers have been addressed for both camera-to-hand and camera-in-hand configurations (e.g., see [1, 99–101]). For visual servo control applications that exploit either of these camera configurations, either the object or the camera is required to remain stationary. In contrast to typical camera-to-hand or camera-in-hand visual servo control configurations, a moving airborne monocular camera (e.g., a camera attached to a remote controlled aircraft, a camera mounted on a satellite) is used by Mehta et al. [18, 19] to provide pose measurements of a moving sensorless unmanned ground vehicle (UGV) relative to a goal configuration. The results in [18, 19] are restricted to three DOF, and the rotation error system is encoded by Euler angle-axis parameterization.

Consider a stationary coordinate frame \mathcal{I}_R that is attached to a camera and a time-varying coordinate frame \mathcal{F}_d that is attached to some object (e.g., an autonomous vehicle) as depicted in Figure 4–1. The object is identified in an image

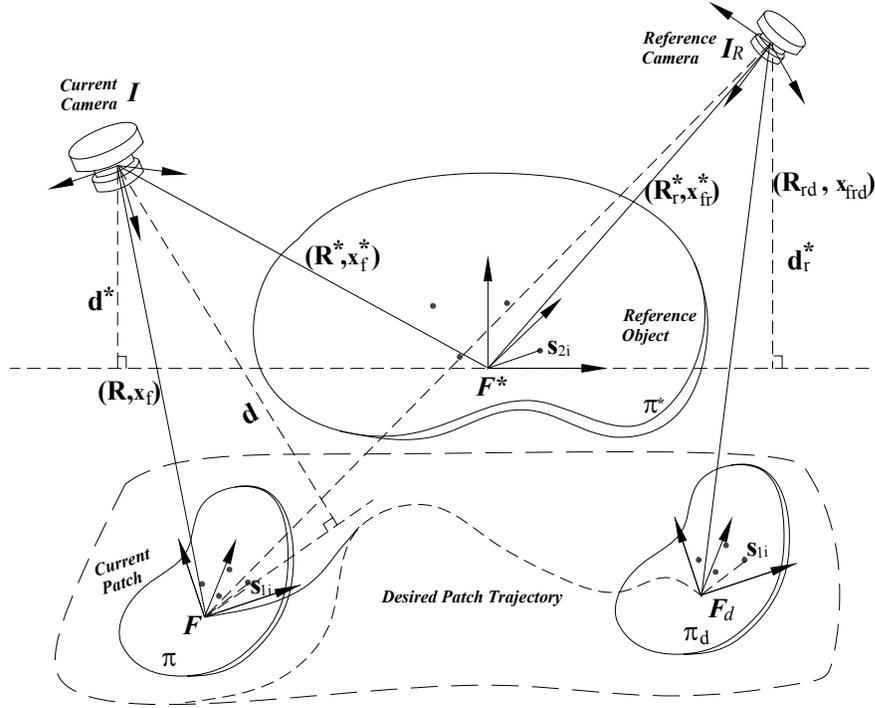


Figure 4-1: Geometric model.

by a collection of feature points that are assumed (without loss of generality) to be coplanar and non-collinear (i.e., a planar patch of feature points). The camera attached to \mathcal{I}_R a priori records a series of snapshots (i.e., a video) of the motion of the object attached to \mathcal{F}_d until it comes to rest (or the video stops recording). A stationary coordinate frame \mathcal{F}^* is attached to a reference object identified by another planar patch of feature points that are assumed to be visible in every frame of the video recorded by the camera. For example, the camera attached to \mathcal{I}_R is on-board a “stationary” satellite that takes a series of snapshots of the relative motion of \mathcal{F}_d with respect to \mathcal{F}^* . Therefore, the desired motion of \mathcal{F}_d can be encoded as a series of relative translations and rotations with respect to the stationary frame \mathcal{F}^* a priori. Spline functions or filter algorithms can be used to generate a smooth desired feature point trajectory [10].

Consider a time-varying coordinate frame \mathcal{I} that is attached to a camera (e.g., a camera attached to a remote controlled aircraft) and a time-varying coordinate

frame \mathcal{F} that is attached to the control object as depicted in Figure 4–1. The camera attached to \mathcal{I} captures snapshots of the planar patches associated with \mathcal{F} and \mathcal{F}^* , respectively. The a priori motion of \mathcal{F}_d represents the desired trajectory of the coordinate system \mathcal{F} , where \mathcal{F} and \mathcal{F}_d are attached to the same object but at different points in time. The camera attached to \mathcal{I}_R is a different camera (with different calibration parameters) as the camera attached to \mathcal{I} . The problem considered in this chapter is to develop a kinematic controller for the object attached to \mathcal{F} so that the time-varying rotation and translation of \mathcal{F} converges to the desired time-varying rotation and translation of \mathcal{F}_d , where the motion of \mathcal{F} is determined from the time-varying overhead camera attached to \mathcal{I} .

4.3 Geometric Model

The relationships between the coordinate systems are as follows (also see Table I): $R(t)$, $R^*(t)$, $R_r(t)$, $R'(t)$, $R_{rd}(t)$, $R_r^* \in SO(3)$ denote the rotation from \mathcal{F} to \mathcal{I} , \mathcal{F}^* to \mathcal{I} , \mathcal{I} to \mathcal{I}_R , \mathcal{F} to \mathcal{I}_R , \mathcal{F}_d to \mathcal{I}_R , and \mathcal{F}^* to \mathcal{I}_R , respectively, $x_f(t)$, $x_f^*(t) \in \mathbb{R}^3$ denote the respective time-varying translation from \mathcal{F} to \mathcal{I} and from \mathcal{F}^* to \mathcal{I} with coordinates expressed in \mathcal{I} , and $x_{fr}(t)$, $x'_{fr}(t)$, $x_{frd}(t)$, $x_{fr}^* \in \mathbb{R}^3$ denote the respective constant translation from \mathcal{I} to \mathcal{I}_R , \mathcal{F} to \mathcal{I}_R , \mathcal{F}_d to \mathcal{I}_R , and from \mathcal{F}^* to \mathcal{I}_R with coordinates expressed in \mathcal{I}_R . From Figure 4–1, the translation $x'_{fr}(t)$ and the rotation $R'(t)$ can be expressed as

$$x'_{fr} = x_{fr}^* + R_r^* R^{*T} (x_f - x_f^*) \quad R' = R_r^* R^{*T} R. \quad (4-1)$$

As illustrated in Figure 4–1, π , π_d and π^* denote the planar patches of feature points associated with \mathcal{F} , \mathcal{F}_d , and \mathcal{F}^* , respectively. $s_{1i} \in \mathbb{R}^3 \forall i = 1, 2, \dots, n$ ($n \geq 4$) denotes the constant Euclidean coordinates of the i -th feature point in \mathcal{F} (and also \mathcal{F}_d), and $s_{2i} \in \mathbb{R}^3 \forall i = 1, 2, \dots, n$ denotes the constant Euclidean coordinates of the i -th feature point in \mathcal{F}^* . From the geometry between

Motion	Frames
$R(t), x_f(t)$	\mathcal{F} to \mathcal{I} in \mathcal{I}
$R^*(t), x_f^*(t)$	\mathcal{F}^* to \mathcal{I} in \mathcal{I}
$R_r(t), x_{fr}(t)$	\mathcal{I} to \mathcal{I}_R
$R'(t), x'_{fr}(t)$	\mathcal{F} to \mathcal{I}_R in \mathcal{I}_R
R_r^*, x_{fr}^*	\mathcal{F}^* to \mathcal{I}_R in \mathcal{I}_R
$R_{rd}(t), x_{frd}(t)$	\mathcal{F}_d to \mathcal{I}_R in \mathcal{I}_R

Table 4-1: Coordinate frames relationships

the coordinate frames depicted in Figure 4-1, the following relationships can be developed

$$\bar{m}_i = x_f + R s_{1i} \quad \bar{m}_{rdi} = x_{frd} + R_{rd} s_{1i} \quad (4-2)$$

$$\bar{m}_{ri}^* = x_{fr}^* + R_r^* s_{2i} \quad \bar{m}'_i = x'_{fr} + R' s_{1i} \quad (4-3)$$

$$\bar{m}_i^* = x_f^* + R^* s_{2i}. \quad (4-4)$$

In (4-2)-(4-4), $\bar{m}_i(t), \bar{m}_i^*(t) \in \mathbb{R}^3$ denote the Euclidean coordinates of the feature points on π and π^* , respectively, expressed in \mathcal{I} as

$$\bar{m}_i(t) \triangleq \begin{bmatrix} x_i(t) & y_i(t) & z_i(t) \end{bmatrix}^T \quad (4-5)$$

$$\bar{m}_i^*(t) \triangleq \begin{bmatrix} x_i^*(t) & y_i^*(t) & z_i^*(t) \end{bmatrix}^T, \quad (4-6)$$

$\bar{m}'_i(t), \bar{m}_{rdi}(t) \in \mathbb{R}^3$ denote the actual and desired time-varying Euclidean coordinates, respectively, of the feature points on π expressed in \mathcal{I}_R as

$$\bar{m}'_i(t) \triangleq \begin{bmatrix} x'_i(t) & y'_i(t) & z'_i(t) \end{bmatrix}^T \quad (4-7)$$

$$\bar{m}_{rdi}(t) \triangleq \begin{bmatrix} x_{rdi}(t) & y_{rdi}(t) & z_{rdi}(t) \end{bmatrix}^T, \quad (4-8)$$

and $\bar{m}_{ri}^* \in \mathbb{R}^3$ denotes the constant Euclidean coordinates of the feature points on the planar patch π^* expressed in \mathcal{I}_R as

$$\bar{m}_{ri}^* \triangleq \begin{bmatrix} x_{ri}^* & y_{ri}^* & z_{ri}^* \end{bmatrix}^T. \quad (4-9)$$

After some algebraic manipulation, the expressions in (4-2)-(4-4) can be rewritten as

$$\bar{m}_i^* = \bar{x}_n + R_n \bar{m}_i \quad (4-10)$$

$$\bar{m}_i = \bar{x}_f + \bar{R} \bar{m}_i^* \quad \bar{m}_{rdi} = \bar{x}_{f_{rd}} + \bar{R}_{rd} \bar{m}_{ri}^* \quad (4-11)$$

$$\bar{m}_{ri}^* = x_{fr} + R_r \bar{m}_i \quad \bar{m}_i' = x_{fr} + R_r \bar{m}_i, \quad (4-12)$$

where $R_n(t)$, $\bar{R}(t)$, $\bar{R}_{rd}(t)$, $R_r(t) \in SO(3)$ and $\bar{x}_n(t)$, $\bar{x}_f(t)$, $\bar{x}_{f_{rd}}(t)$, $x_{fr}(t) \in \mathbb{R}^3$ are new rotational and translational variables, respectively, defined as

$$R_n = R^* R^T \quad \bar{R} = R R^{*T} \quad (4-13)$$

$$\bar{R}_{rd} = R_{rd} R_r^{*T} \quad R_r = R_r^* R^{*T}$$

and

$$\bar{x}_n = x_f^* - R_n (x_f - R (s_{2i} - s_{1i})) \quad (4-14)$$

$$\bar{x}_f = x_f - \bar{R} (x_f^* + R^* (s_{2i} - s_{1i})) \quad (4-15)$$

$$\bar{x}_{f_{rd}} = x_{f_{rd}} - \bar{R}_{rd} (x_{fr}^* + R_r^* (s_{2i} - s_{1i})) \quad (4-16)$$

$$x_{fr} = x_{fr}^* - R_r x_f^* = x_{fr}' - R_r x_f. \quad (4-17)$$

Note that $R_n(t)$, $\bar{R}(t)$ and $\bar{R}_{rd}(t)$ in (4-13) are the rotation matrices between \mathcal{F} and \mathcal{F}^* , \mathcal{F}^* and \mathcal{F} , and \mathcal{F}^* and \mathcal{F}_d , respectively, but $\bar{x}_n(t)$, $\bar{x}_f(t)$ and $\bar{x}_{f_{rd}}(t)$ in (4-14)-(4-16) are not the translation vectors between the corresponding coordinate

frames. However, this will not affect the following controller design because only the rotation matrices will be used in the controller development.

To facilitate the development of a relationship between the actual Euclidean translation of \mathcal{F} to the Euclidean translation that is reconstructed from the image information, the following projective relationships are developed:

$$d(t) = n^T \bar{m}_i \quad d^*(t) = n^{*T} \bar{m}_i^* \quad d_r^* = n_r^{*T} \bar{m}_{ri}^*, \quad (4-18)$$

where $d(t) \in \mathbb{R}$ represents the distance from the origin of \mathcal{I} to π along the unit normal (expressed in \mathcal{I}) to π denoted as $n(t) \in \mathbb{R}^3$, $d^*(t) \in \mathbb{R}$ represents the distance from the origin of \mathcal{I} to π^* along the unit normal (expressed in \mathcal{I}) to π^* denoted as $n^*(t) \in \mathbb{R}^3$, and $d_r^* \in \mathbb{R}$ represents the distance from the origin of \mathcal{I}_R to π^* along the unit normal (expressed in \mathcal{I}_R) to π^* denoted as $n_r^* \in \mathbb{R}^3$ where $n^*(t) = R_r^T(t)n_r^*$. In (4-18), $d(t)$, $d^*(t)$, $d_r^* > \varepsilon$ for some positive constant $\varepsilon \in \mathbb{R}$.

Based on (4-18), the relationships in (4-10)-(4-12) can be expressed as

$$\bar{m}_i^* = \left(R_n + \frac{\bar{x}_n}{d} n^T \right) \bar{m}_i \quad (4-19)$$

$$\bar{m}_i = \left(\bar{R} + \frac{\bar{x}_f}{d^*} n^{*T} \right) \bar{m}_i^* \quad (4-20)$$

$$\bar{m}_{r di} = \left(\bar{R}_{rd} + \frac{\bar{x}_{f rd}}{d_r^*} n_r^{*T} \right) \bar{m}_{ri}^* \quad (4-21)$$

$$\bar{m}_{ri}^* = \left(R_r + \frac{x_{fr} n^{*T}}{d^*} \right) \bar{m}_i^* \quad (4-22)$$

$$\bar{m}_i' = \left(R_r + \frac{x_{fr} n^T}{d} \right) \bar{m}_i. \quad (4-23)$$

As in Chen et al. [10], the subsequent development requires that the constant rotation matrix R_r^* be known. The constant rotation matrix R_r^* can be obtained a priori using various methods (e.g., a second camera, Euclidean measurements). The subsequent development also assumes that the difference between the Euclidean distances ($s_{2i} - s_{1i}$) is a constant $\forall i = 1, \dots, n$. While there are many practical applications that satisfy this assumption (e.g., a simple scenario is that the objects

attached to F and F^* are the same object), the assumption is generally restrictive and is the focus of future research. As described by Hu et al. [17], this assumption can be avoided by using the geometric reconstruction approach [102] under an alternative assumption that the distance between two feature points is precisely known.

4.4 Euclidean Reconstruction

The relationships given by (4-19)-(4-23) provide a means to quantify a translation and rotation error between the different coordinate systems. Since the pose of π , π_d , and π^* cannot be directly measured, a Euclidean reconstruction is developed to obtain the pose error by comparing multiple images acquired from the hovering monocular vision system. To facilitate the subsequent development, the normalized Euclidean coordinates of the feature points in π and π^* can be expressed in terms of \mathcal{I} as $m_i(t) \in \mathbb{R}^3$ and $m_i^*(t) \in \mathbb{R}^3$, respectively, as

$$m_i \triangleq \frac{\bar{m}_i}{z_i} \quad m_i^* \triangleq \frac{\bar{m}_i^*}{z_i^*}. \quad (4-24)$$

Similarly, the normalized Euclidean coordinates of the feature points for the current, desired, and reference image can be expressed in terms of \mathcal{I}_R as $m_i'(t)$, $m_{r di}(t)$, $m_{r i}^* \in \mathbb{R}^3$, respectively, as

$$m_i'(t) \triangleq \frac{\bar{m}_i'(t)}{z_i'(t)} \quad m_{r di}(t) \triangleq \frac{\bar{m}_{r di}(t)}{z_{r di}(t)} \quad m_{r i}^* \triangleq \frac{\bar{m}_{r i}^*}{z_{r i}^*}. \quad (4-25)$$

From the expressions given in (4-20) and (4-24), the rotation and translation between the coordinate systems \mathcal{F} and \mathcal{F}^* , between \mathcal{F}^* and \mathcal{F}_d , and between \mathcal{I} and

\mathcal{I}_R can now be related in terms of the normalized Euclidean coordinates as follows:

$$m_i = \alpha_i (\bar{R} + x_h n^{*T}) m_i^* \quad (4-26)$$

$$m_i^* = \frac{1}{\alpha_i} (R_n + x_{nh} n^T) m_i \quad (4-27)$$

$$m_{rdi} = \alpha_{rdi} (\bar{R}_{rd} + x_{hrd} n_r^{*T}) m_{ri}^* \quad (4-28)$$

$$m_{ri}^* = \alpha_{ri} (R_r + x_{hr} n^{*T}) m_{ri}^*, \quad (4-29)$$

where $\alpha_i(t)$, $\alpha_{rdi}(t)$, $\alpha_{ri}(t) \in \mathbb{R}$ denote depth ratios defined as

$$\alpha_i = \frac{z_i^*}{z_i} \quad \alpha_{rdi} = \frac{z_{ri}^*}{z_{rdi}} \quad \alpha_{ri} = \frac{z_i^*}{z_{ri}^*},$$

and $x_h(t)$, $x_{nh}(t)$, $x_{hrd}(t)$, $x_{hr}(t) \in \mathbb{R}^3$ denote scaled translation vectors that are defined as

$$\begin{aligned} x_h &= \frac{\bar{x}_f}{d^*} & x_{nh} &= \frac{\bar{x}_n}{d} \\ x_{hrd} &= \frac{\bar{x}_{frd}}{d_r^*} & x_{hr} &= \frac{x_{fr}}{d^*}. \end{aligned} \quad (4-30)$$

Since the normalized Euclidean coordinates in (4-26)-(4-29) can not be directly measured, the following relationships (i.e., the pin-hole camera model) are used to determine the normalized Euclidean coordinates from pixel information

$$p_i = A_1 m_i \quad p_i^* = A_1 m_i^* \quad (4-31)$$

$$p_{rdi} = A_2 m_{rdi} \quad p_{ri}^* = A_2 m_{ri}^*, \quad (4-32)$$

where $A_1, A_2 \in \mathbb{R}^{3 \times 3}$ are known, constant, and invertible intrinsic camera calibration matrices of the current camera and the reference camera, respectively. In (4-31) and (4-32), $p_i(t)$ and $p_i^*(t) \in \mathbb{R}^3$ represent the image-space coordinates of the Euclidean feature points on π and π^* expressed in terms of \mathcal{I} as

$$p_i \triangleq \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T \quad p_i^* \triangleq \begin{bmatrix} u_i^* & v_i^* & 1 \end{bmatrix}^T, \quad (4-33)$$

respectively, where $u_i(t), v_i(t), u_i^*(t), v_i^*(t) \in \mathbb{R}$. Similarly, $p_{rdi}(t)$ and $p_{ri}^* \in \mathbb{R}^3$ represent the image-space coordinates of the Euclidean features on π_d and π^* expressed in terms of \mathcal{I}_R as

$$p_{rdi} \triangleq \begin{bmatrix} u_{rdi} & v_{rdi} & 1 \end{bmatrix}^T \quad p_{ri}^* \triangleq \begin{bmatrix} u_{ri}^* & v_{ri}^* & 1 \end{bmatrix}^T \quad (4-34)$$

respectively, where $u_{rdi}(t), v_{rdi}(t), u_{ri}^*, v_{ri}^* \in \mathbb{R}$. By using (4-26)-(4-29) and (4-31)-(4-34), the following relationships can be developed:

$$p_i = \alpha_i \underbrace{\left(A_1 (\bar{R} + x_h n^{*T}) A_1^{-1} \right)}_G p_i^* \quad (4-35)$$

$$p_i^* = \frac{1}{\alpha_i} \underbrace{\left(A_1 (R_n + x_{nh} n^T) A_1^{-1} \right)}_{G_n} p_i \quad (4-36)$$

$$p_{rdi} = \alpha_{rdi} \underbrace{\left(A_2 (\bar{R}_{rd} + x_{hrd} n_r^{*T}) A_2^{-1} \right)}_{G_{rd}} p_{ri}^* \quad (4-37)$$

$$p_{ri}^* = \alpha_{ri} \underbrace{\left(A_2 (R_r + x_{hr} n^{*T}) A_1^{-1} \right)}_{G_r} p_i^* , \quad (4-38)$$

where $G(t), G_n(t), G_{rd}(t), G_r(t) \in \mathbb{R}^{3 \times 3}$ denote projective homographies. Sets of linear equations can be developed from (4-35)-(4-38) to determine the projective homographies up to a scalar multiple. Various techniques can be used (e.g., see Faugeras and Lustman [93] and Zhang and Hanson [94]) to decompose the Euclidean homographies, to obtain $\alpha_i(t), \alpha_{rdi}(t), \alpha_{ri}(t), x_h(t), x_{nh}(t), x_{hrd}(t), x_{hr}(t), \bar{R}(t), R_n(t), \bar{R}_{rd}(t), R_r(t), n^*(t), n_r^*, n(t)$. Given that the constant rotation matrix R_r^* is assumed to be known, the expressions for $\bar{R}_{rd}(t)$ and $R_r(t)$ in (4-13) can be used to determine $R_{rd}(t)$ and $R^*(t)$. Once $R^*(t)$ is determined, the expression for $\bar{R}(t)$ in (4-13) can be used to determine $R(t)$. Also, once $R_r^*, R^{*T}(t)$, and $R(t)$ have been determined, (4-1) can be used to determine $R'(t)$. Since $R_r(t)$,

$x_{hr}(t)$, $\alpha_i(t)$, $n^*(t)$, n_r^* , $n(t)$, $m_i^*(t)$, and $m_i(t)$ can be determined, the following relationship can be used to determine $m_i'(t)$:

$$m_i' = \frac{z_i}{z_i'} \left(R_r + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^T m_i} n^T \right) m_i, \quad (4-39)$$

where the inverse of the ratio $\frac{z_i(t)}{z_i'(t)}$ can be determined as

$$\frac{z_i'}{z_i} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \left(R_r + x_{hr} \alpha_i \frac{n^{*T} m_i^*}{n^T m_i} n^T \right) m_i. \quad (4-40)$$

4.5 Control Objective

The control objective is for a six DOF rigid-body object (e.g., an autonomous vehicle) identified by a planar patch of feature points to track a desired trajectory that is determined by a sequence of images taken by a fixed reference camera. This objective is based on the assumption that the linear and angular velocities of the camera are control inputs that can be independently controlled (i.e., unconstrained motion) and that the reference and desired cameras are calibrated (i.e., A_1 and A_2 are known). The control objective can be stated as $\bar{m}_i'(t) \rightarrow \bar{m}_{rdi}(t)$ (i.e., the Euclidean feature points on π track the corresponding feature points on π_d). Equivalently, the control objective can also be stated in terms of the rotation and translation of the object as $x'_{fr}(t) \rightarrow x_{frd}(t)$ and $R'(t) \rightarrow R_{rd}(t)$. As stated previously, $R'(t)$ and $R_{rd}(t)$ can be computed by decomposing the projective homographies in (4-35)-(4-38) and using (4-1). Once these rotation matrices have been determined, the unit quaternion parameterization is used to describe the rotation matrix. This parameterization facilitates the subsequent problem formulation, control development, and stability analysis since the unit quaternion provides a global nonsingular parameterization of the corresponding rotation matrices. See Section 2.3 for some background about the unit quaternion.

Given the rotation matrices $R'(t)$ and $R_{rd}(t)$, the corresponding unit quaternions $q(t)$ and $q_d(t)$ can be calculated by using the numerically robust method (e.g., see [14] and [15]) based on the corresponding relationships

$$R' = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T + 2q_0 q_v^\times \quad (4-41)$$

$$R_{rd} = (q_{0d}^2 - q_{vd}^T q_{vd}) I_3 + 2q_{vd} q_{vd}^T + 2q_{0d} q_{vd}^\times, \quad (4-42)$$

where I_3 is the 3×3 identity matrix, and the notation $q_v^\times(t)$ denotes the following skew-symmetric form of the vector $q_v(t)$ as in (2-10).

To quantify the rotation error between the feature points on π and π_d , the error between rotation matrices $R'(t)$ and $R_{rd}(t)$ is defined as

$$\tilde{R} = R'^T R_{rd} = (\tilde{q}_0^2 - \tilde{q}_v^T \tilde{q}_v) I_3 + 2\tilde{q}_v \tilde{q}_v^T - 2\tilde{q}_0 \tilde{q}_v^\times, \quad (4-43)$$

where the error quaternion $\tilde{q}(t) = (\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ is defined as

$$\tilde{q} = \begin{bmatrix} \tilde{q}_0 \\ \tilde{q}_v \end{bmatrix} = \begin{bmatrix} q_0 q_{0d} + q_v^T q_{vd} \\ q_{0d} q_v - q_0 q_{vd} + q_v^\times q_{vd} \end{bmatrix}. \quad (4-44)$$

Since $\tilde{q}(t)$ is a unit quaternion, (4-43) can be used to quantify the rotation tracking objective as

$$\|\tilde{q}_v(t)\| \rightarrow 0 \implies \tilde{R}(t) \rightarrow I_3 \quad \text{as } t \rightarrow \infty. \quad (4-45)$$

The translation error, denoted by $e(t) \in \mathbb{R}^3$, is defined as [10, 20]

$$e = m_e - m_{ed}, \quad (4-46)$$

where $m_e(t), m_{ed}(t) \in \mathbb{R}^3$ are defined as

$$m_e = \begin{bmatrix} \frac{x'_i}{z'_i} & \frac{y'_i}{z'_i} & \ln\left(\frac{z'_i}{z_{ri}^*}\right) \end{bmatrix}^T \quad m_{ed} = \begin{bmatrix} \frac{x_{rdi}}{z_{rdi}} & \frac{y_{rdi}}{z_{rdi}} & \ln\left(\frac{z_{rdi}}{z_{ri}^*}\right) \end{bmatrix}^T. \quad (4-47)$$

In (3-7), $\frac{z'_i(t)}{z_{ri}^*(t)}$ and $\frac{z_{rdi}(t)}{z_{ri}^*(t)}$ can be computed as below

$$\frac{z'_i}{z_{ri}^*} = \frac{z'_i}{z_i} \frac{z_i}{z_i^*} \frac{z_i^*}{z_{ri}^*} = \frac{z'_i}{z_i} \frac{1}{\alpha_i} \alpha_{ri} \quad \frac{z_{rdi}}{z_{ri}^*} = \frac{1}{\alpha_{rdi}}.$$

Based on (4-45) and (4-46), the subsequent control development targets the following objectives:

$$\|\tilde{q}_v(t)\| \rightarrow 0 \quad \text{and} \quad \|e(t)\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty. \quad (4-48)$$

4.6 Control Development

4.6.1 Open-Loop Error System

From (4-43) and (4-44), the open-loop rotation error system can be developed as

$$\dot{\tilde{q}} = \frac{1}{2} \begin{bmatrix} -\tilde{q}_v^T \\ \tilde{q}_0 I_3 + \tilde{q}_v^\times \end{bmatrix} (\omega_c - \tilde{R} \omega_{cd}), \quad (4-49)$$

where $\omega_{cd}(t)$ denotes the angular velocity of π_d expressed in \mathcal{F}_d that can be calculated as [23]

$$\omega_{cd} = 2(q_{0d}\dot{q}_{vd} - q_{vd}\dot{q}_{0d}) - 2q_{vd}^\times \dot{q}_{vd}, \quad (4-50)$$

where $(q_{0d}(t), q_{vd}^T(t))^T$, $(\dot{q}_{0d}(t), \dot{q}_{vd}^T(t))^T$ are assumed to be bounded; hence, $\omega_{cd}(t)$ is also bounded. The open-loop translation error system can be derived as (see Appendix C)

$$z_{ri}^* \dot{e} = \frac{z_{ri}^*}{z'_i} L'_v R' (v_c + \omega_c^\times s_i) - z_{ri}^* \dot{m}_{ed}, \quad (4-51)$$

where $v_c(t), \omega_c(t) \in \mathbb{R}^3$ denote the linear and angular velocity vectors of π expressed in \mathcal{F} , respectively, and the auxiliary measurable term $L'_v(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$L'_v = \begin{bmatrix} 1 & 0 & -\frac{x'_i}{z'_i} \\ 0 & 1 & -\frac{y'_i}{z'_i} \\ 0 & 0 & 1 \end{bmatrix}.$$

4.6.2 Closed-Loop Error System

Based on the open-loop rotation error system in (4-49) and the subsequent Lyapunov-based stability analysis, the angular velocity controller is designed as

$$\omega_c = -K_\omega \tilde{q}_v + \tilde{R} \omega_{cd}, \quad (4-52)$$

where $K_\omega \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains.

From (4-49) and (4-52), the rotation closed-loop error system can be determined as

$$\begin{aligned} \dot{\tilde{q}}_0 &= \frac{1}{2} \tilde{q}_v^T K_\omega \tilde{q}_v \\ \dot{\tilde{q}}_v &= -\frac{1}{2} (\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v = -\frac{1}{2} K_\omega \tilde{q}_0 \tilde{q}_v. \end{aligned} \quad (4-53)$$

From (4-51), the translation control input $v_c(t)$ is designed as

$$v_c = -\frac{z_i'}{z_{ri}^*} R'^T L_v'^{-1} (K_v e - \hat{z}_{ri}^* \dot{m}_{ed}) - \omega_c^\times s_i, \quad (4-54)$$

where $K_v \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. In

(4-54), the parameter estimate $\hat{z}_{ri}^*(t) \in \mathbb{R}$ for the unknown constant z_{ri}^* is designed

as

$$\dot{\hat{z}}_{ri}^* = -\gamma e^T \dot{m}_{ed}, \quad (4-55)$$

where $\gamma \in \mathbb{R}$ denotes a positive constant adaptation gain. By using (4-51) and

(4-54), the translation closed-loop error system is

$$z_{ri}^* \dot{e} = -K_v e - \hat{z}_{ri}^* \dot{m}_{ed}, \quad (4-56)$$

where $\tilde{z}_{ri}^*(t) \in \mathbb{R}$ denotes the following parameter estimation error:

$$\tilde{z}_{ri}^* = z_{ri}^* - \hat{z}_{ri}^*. \quad (4-57)$$

4.6.3 Stability Analysis

Theorem 4.1: The controller given in (4-52) and (4-54), along with the adaptive update law in (4-55) ensures asymptotic tracking in the sense that

$$\|\tilde{q}_v(t)\| \rightarrow 0, \quad \|e(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty. \quad (4-58)$$

Proof: Let $V(t) \in \mathbb{R}$ denote the following differentiable non-negative function (i.e., a Lyapunov candidate):

$$V = \tilde{q}_v^T \tilde{q}_v + (1 - \tilde{q}_0)^2 + \frac{z_{ri}^*}{2} e^T e + \frac{1}{2\gamma} \tilde{z}_{ri}^{*2}. \quad (4-59)$$

The time-derivative of $V(t)$ can be determined as

$$\begin{aligned} \dot{V} &= -\tilde{q}_v^T K_\omega \tilde{q}_0 \tilde{q}_v - (1 - \tilde{q}_0) \tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e \\ &\quad + e^T (-K_v e - \tilde{z}_{ri}^* \dot{m}_{ed}) + \tilde{z}_{ri}^* e^T \dot{m}_{ed} \\ &= -\tilde{q}_v^T (\tilde{q}_0 I_3 + (1 - \tilde{q}_0) I_3) K_\omega \tilde{q}_v - e^T K_v e \\ &= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e, \end{aligned} \quad (4-60)$$

where (4-53) and (4-55)-(4-57) were utilized. Based on (4-59) and (4-60), $e(t)$, $\tilde{q}_v(t)$, $\tilde{q}_0(t)$, $\tilde{z}_{ri}^*(t) \in \mathcal{L}_\infty$ and $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_2$. Since $\tilde{z}_{ri}^*(t) \in \mathcal{L}_\infty$, it is clear from (4-57) that $\hat{z}_{ri}^*(t) \in \mathcal{L}_\infty$. Based on the fact that $e(t) \in \mathcal{L}_\infty$, (4-46) and (4-47) can be used to prove that $m_i'(t) \in \mathcal{L}_\infty$, and then $L_v'(t)$, $L_v'^{-1}(t) \in \mathcal{L}_\infty$. Based on the fact that $\tilde{q}_v(t) \in \mathcal{L}_\infty$ and $\omega_{cd}(t)$ is a bounded function, (4-52) can be used to conclude that $\omega_c(t) \in \mathcal{L}_\infty$. Since $\hat{z}_{ri}^*(t)$, $e(t)$, $m_i'(t)$, $L_v'(t)$, $L_v'^{-1}(t) \in \mathcal{L}_\infty$ and $\dot{m}_{ed}(t)$ is bounded, (4-54) can be utilized to prove that $v_c(t) \in \mathcal{L}_\infty$. From the previous results, (4-49)-(4-51) can be used to prove that $\dot{e}(t)$, $\dot{\tilde{q}}_v(t) \in \mathcal{L}_\infty$. Since $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_\infty \cap \mathcal{L}_2$, and $\dot{e}(t)$, $\dot{\tilde{q}}_v(t) \in \mathcal{L}_\infty$, Barbalat's Lemma [96] can be used to conclude the result given in (4-58).

4.7 Simulation Results

A numerical simulation was performed to illustrate the performance of the tracking controller given in (4-52), (4-54), and the adaptive update law in (4-55).

The camera calibration parameters were chosen as

$$A_1 = A_2 = \begin{bmatrix} 1545.1 & 0 & 640 \\ 0 & 1545.1 & 512 \\ 0 & 0 & 1 \end{bmatrix}.$$

The origins of coordinate frames \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d , and the four coplanar feature points on the planes π , π^* and π_d are chosen such that the feature points have the same Euclidean coordinates (in [m]) in \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d , as

$$\begin{aligned} s_{11} = s_{12} &= \begin{bmatrix} 0 & 0.15 & 0 \end{bmatrix}^T \\ s_{21} = s_{22} &= \begin{bmatrix} 0.15 & 0.15 & 0 \end{bmatrix}^T \\ s_{31} = s_{32} &= \begin{bmatrix} 0.15 & 0 & 0 \end{bmatrix}^T \\ s_{41} = s_{42} &= \begin{bmatrix} -0.15 & 0 & 0 \end{bmatrix}^T. \end{aligned}$$

The time-varying desired image trajectory was generated by the kinematics of the feature point plane where the desired linear and angular velocities were selected as

$$\begin{aligned} v_{cd} &= \begin{bmatrix} 0.1 \sin(t) & 0.1 \sin(t) & 0.1 \sin(t) \end{bmatrix}^T [m/s] \\ \omega_{cd} &= \begin{bmatrix} 0.1 \sin(t) & 0.1 \sin(t) & 0.5 \end{bmatrix}^T [rad/s]. \end{aligned}$$

The moving camera attached to \mathcal{I} is assumed to have a linear velocity vector of

$$\begin{bmatrix} 0.1 \sin(3t) & 0.1 \sin(2t) & 0.1 \sin(t) \end{bmatrix}^T [m/s].$$

The initial rotation matrices $R(0)$ between \mathcal{F} and \mathcal{I} , $R^*(0)$ between \mathcal{F}^* and \mathcal{I} , and $R_{rd}(0)$ between \mathcal{F}_d and \mathcal{I}_R , and the constant rotation matrix R_r^* between \mathcal{F}^*

and \mathcal{I}_R , were set as

$$\begin{aligned} R(0) &= R_x(180^\circ)R_y(0^\circ)R_z(40^\circ) \\ R^*(0) &= R_x(180^\circ)R_y(0^\circ)R_z(-20^\circ) \\ R_{rd}(0) &= R_x(180^\circ)R_y(0^\circ)R_z(20^\circ) \\ R_r^* &= R_x(180^\circ)R_y(0^\circ)R_z(80^\circ). \end{aligned}$$

The initial translation vectors $x_f(0)$ between \mathcal{F} and \mathcal{I} (expressed in \mathcal{I}), $x_f^*(0)$ between \mathcal{F}^* and \mathcal{I} (expressed in \mathcal{I}), and $x_{f_{rd}}(0)$ between \mathcal{F}_d and \mathcal{I}_R (expressed in \mathcal{I}_R), and the constant translation vector $x_{f_r}^*(0)$ between \mathcal{F}^* and \mathcal{I}_R (expressed in \mathcal{I}_R), were selected as

$$\begin{aligned} x_f(0) &= \begin{bmatrix} -0.5 & 0.5 & 4.0 \end{bmatrix}^T \\ x_f^*(0) &= \begin{bmatrix} 1.0 & 1.5 & 3.5 \end{bmatrix}^T \\ x_{f_{rd}}(0) &= \begin{bmatrix} 0.5 & 1.5 & 6.0 \end{bmatrix}^T \\ x_{f_r}^*(0) &= \begin{bmatrix} -1.0 & 1.5 & 4.0 \end{bmatrix}^T. \end{aligned}$$

The initial Euclidean relationship between the cameras, the reference object, the control object, and the object that was used to generate the desired trajectory is shown in Figure 4-2.

The initial image-space coordinates (i.e., $p_i(0)$) of the four feature points attached to the plane π , expressed in \mathcal{I} , were computed as (in pixels)

$$\begin{aligned} p_1(0) &= \begin{bmatrix} 411.9 & 662.6 & 1 \end{bmatrix}^T & p_2(0) &= \begin{bmatrix} 453.4 & 623.3 & 1 \end{bmatrix}^T \\ p_3(0) &= \begin{bmatrix} 487.5 & 665.7 & 1 \end{bmatrix}^T & p_4(0) &= \begin{bmatrix} 405.1 & 745.7 & 1 \end{bmatrix}^T. \end{aligned}$$

The initial reference image-space coordinates (i.e., $p_i^*(0)$ and $p_{r_i}^*$) of the four feature points attached to the plane π^* , expressed in \mathcal{I} and \mathcal{I}_R , respectively, were computed as (in pixels)

$$\begin{aligned} p_1^*(0) &= \begin{bmatrix} 1104.1 & 1112.0 & 1 \end{bmatrix}^T & p_2^*(0) &= \begin{bmatrix} 1166.3 & 1134.6 & 1 \end{bmatrix}^T \\ p_3^*(0) &= \begin{bmatrix} 1143.7 & 1196.8 & 1 \end{bmatrix}^T & p_4^*(0) &= \begin{bmatrix} 1019.2 & 1151.5 & 1 \end{bmatrix}^T \\ p_{r1}^* &= \begin{bmatrix} 196.7 & 1081.4 & 1 \end{bmatrix}^T & p_{r2}^* &= \begin{bmatrix} 206.7 & 1024.3 & 1 \end{bmatrix}^T \\ p_{r3}^* &= \begin{bmatrix} 263.8 & 1034.4 & 1 \end{bmatrix}^T & p_{r4}^* &= \begin{bmatrix} 243.7 & 1148.5 & 1 \end{bmatrix}^T. \end{aligned}$$

The initial image-space coordinates (i.e., $p_{rdi}(0)$) of the four feature points attached to the plane π_d , expressed in \mathcal{I}_R , were computed as (in pixels)

$$\begin{aligned} p_{rd1}(0) &= \begin{bmatrix} 755.5 & 862.0 & 1 \end{bmatrix}^T & p_{rd2}(0) &= \begin{bmatrix} 791.8 & 848.8 & 1 \end{bmatrix}^T \\ p_{rd3}(0) &= \begin{bmatrix} 805.1 & 885.1 & 1 \end{bmatrix}^T & p_{rd4}(0) &= \begin{bmatrix} 732.5 & 911.5 & 1 \end{bmatrix}^T. \end{aligned}$$

The control gains K_ω in (4-57) and K_v in (4-54) and adaptation gain γ in (4-55) were selected as

$$\begin{aligned} K_\omega &= \text{diag}\{1, 1, 1\} & K_v &= \text{diag}\{5, 5, 5\} \\ \gamma &= 20. \end{aligned}$$

The desired image-space trajectory of the feature point plane π_d , taken by the camera attached to \mathcal{I}_R , is shown in Figure 4-3. The current image-space trajectory of the feature point plane π , taken by the camera attached to \mathcal{I} , is shown in Figure 4-5. The reference image-space trajectory of the reference plane π^* , taken by the camera attached to \mathcal{I} , is shown in Figure 4-4. The resulting translation and

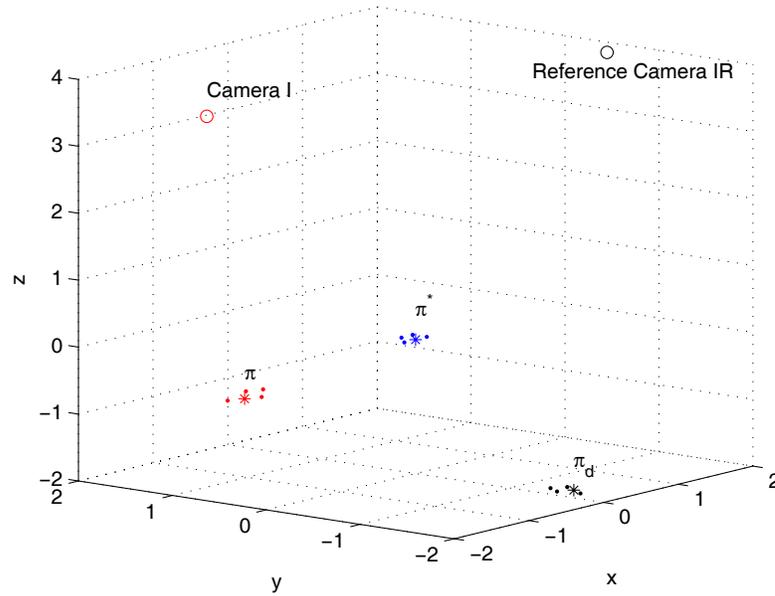


Figure 4–2: This figure shows the initial positions of the cameras and the feature point planes. The initial positions of the cameras attached to \mathcal{I} and \mathcal{I}_R are denoted by “O”. The feature points on the planes π , π^* and π_d are denoted by “.”. The origins of the coordinate frames \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d are denoted by “*”.

rotation tracking errors are plotted in Figure 4–6 and Figure 4–7, respectively. The errors go to zero asymptotically. The translation and rotation control inputs are shown in Figure 4–8 and Figure 4–9, respectively.

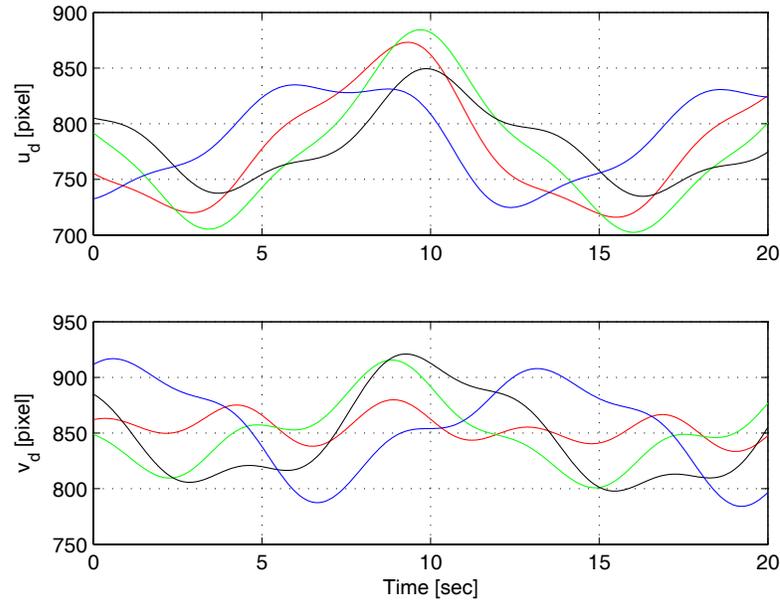


Figure 4–3: Pixel coordinate $p_{rd}(t)$ of the four feature points on the plane π_d in a sequence of desired images taken by the camera attached to \mathcal{I}_R . The upper figure is for the $u_{rd}(t)$ component and the bottom figure is for the $v_{rd}(t)$ component.

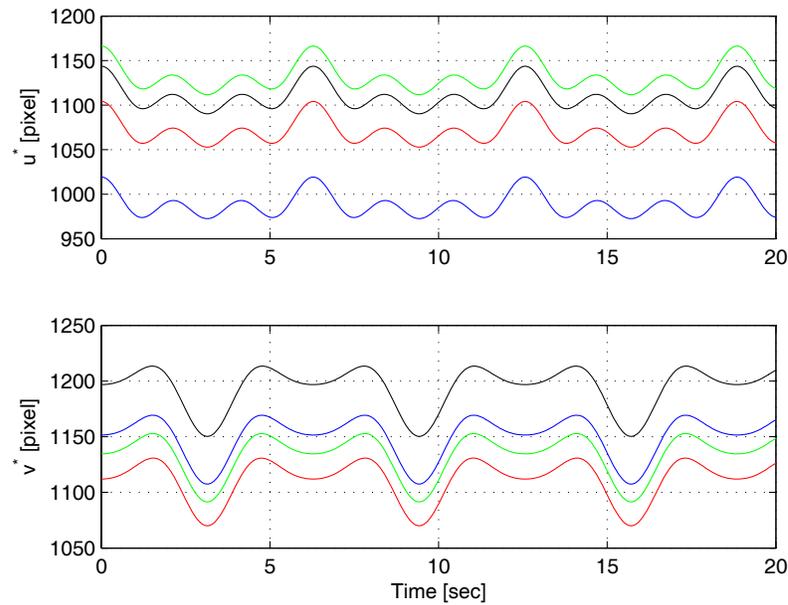


Figure 4–4: Pixel coordinate $p^*(t)$ of the four feature points on the plane π^* in a sequence of reference images taken by the moving camera attached to \mathcal{I} . The upper figure is for the $u^*(t)$ component and the bottom figure is for the $v^*(t)$ component.

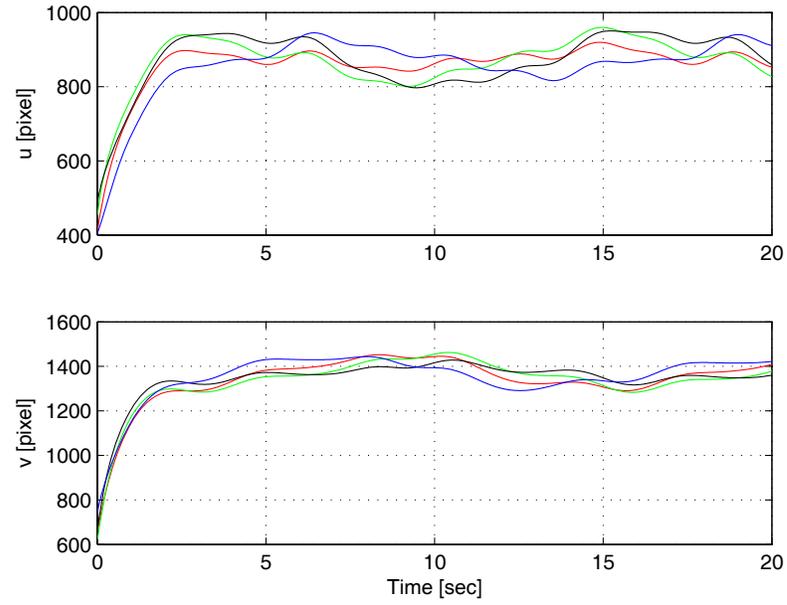


Figure 4–5: Pixel coordinate $p(t)$ of the four feature points on the plane π in a sequence of images taken by the moving camera attached to \mathcal{I} . The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.

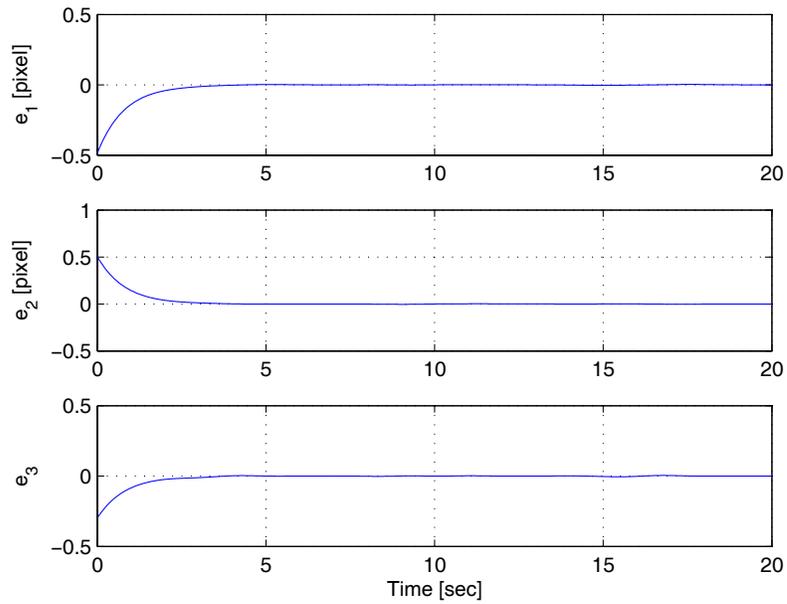


Figure 4–6: Translation error $e(t)$.

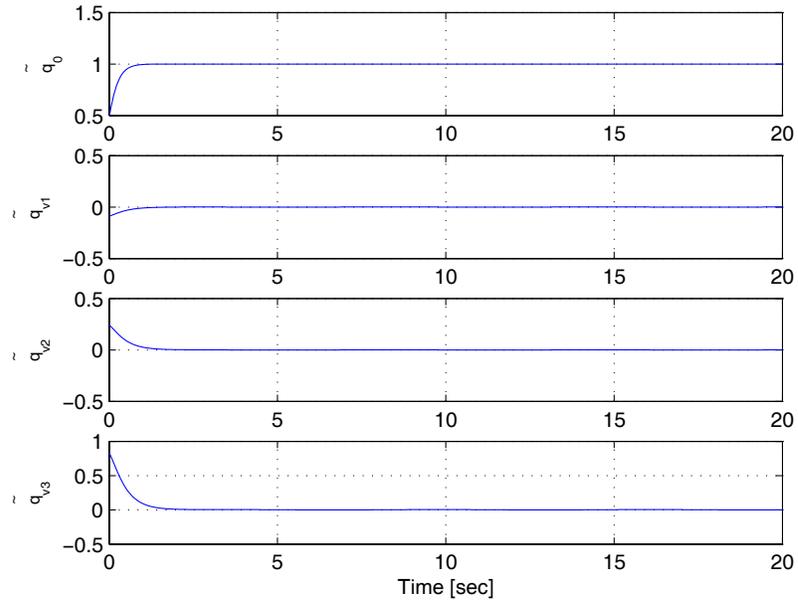


Figure 4-7: Rotation quaternion error $\tilde{q}(t)$.

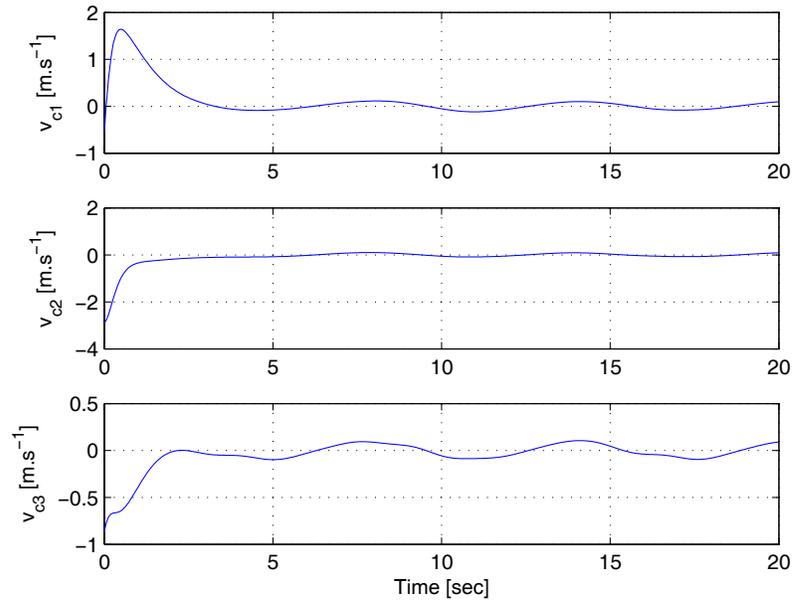


Figure 4-8: Linear velocity input $v_c(t)$ for the planar patch π .

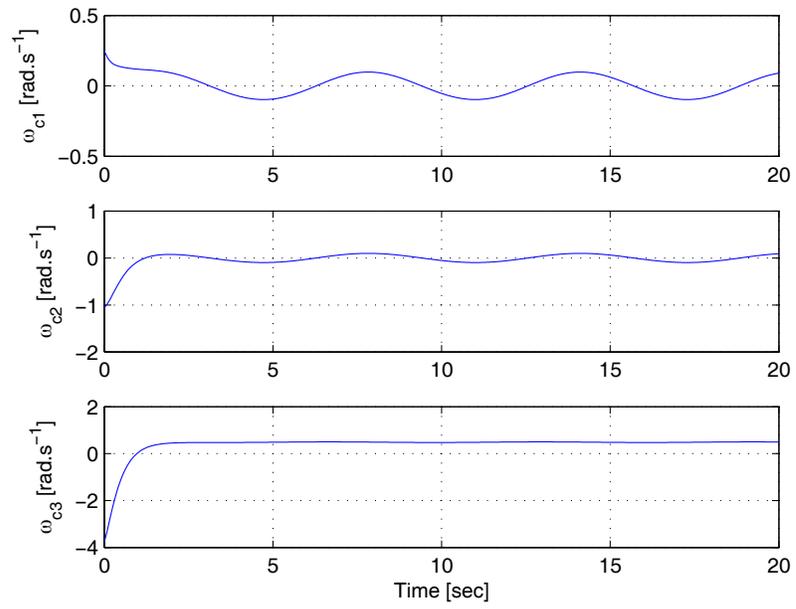


Figure 4-9: Angular velocity input $w_c(t)$ for the planar patch π .

CHAPTER 5
ADAPTIVE VISUAL SERVO TRACKING CONTROL USING A CENTRAL
CATADIOPTRIC CAMERA

5.1 Introduction

In this chapter, an adaptive homography based visual servo tracking control scheme is presented for a camera-in-hand central catadioptric camera system. By using the central catadioptric camera, a full panoramic FOV is obtained. The literature review for visual servo control using central catadioptric cameras are presented in Section 1.3.2. In this chapter, the tracking controller is developed based on the relative relationships of a central catadioptric camera between the current, reference, and desired camera poses. To find the relative camera pose relationships, homographies are computed based on the projection model of the central catadioptric camera [33–36]. As stated by Geyer and Daniilidis [36], a unifying theory was proposed to show that all central catadioptric systems are isomorphic to projective mappings from the sphere to a plane with a projection center on the perpendicular to the plane. By constructing links between the projected coordinates on the sphere, the homographies up to scalar multiples can be obtained. Various methods can then be applied to decompose the Euclidean homographies to find the corresponding rotation matrices, and depth ratios. The rotation error system in this chapter is based on the quaternion formulation which has a full-rank 4×3 interaction matrix. Lyapunov-based methods are utilized to develop the controller and to prove asymptotic tracking.

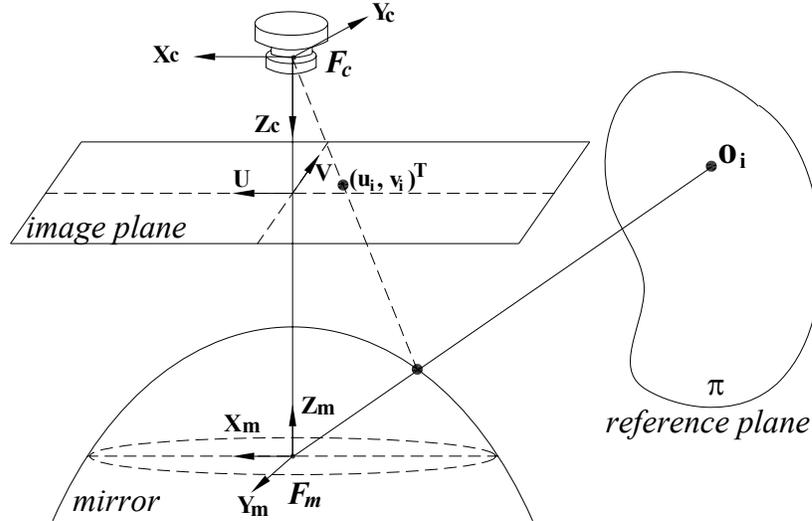


Figure 5–1: Central catadioptric projection relationship.

5.2 Geometric Model

A central catadioptric camera is composed of two elements: a camera and a mirror which are calibrated to yield a single effective viewpoint. Geyer and Daniilidis [36] developed a unifying theory that explains how all central catadioptric systems are isomorphic to projective mappings from the sphere to a plane with a projection center on the optical axis perpendicular to the plane. For the central catadioptric camera depicted in Figure 5–1, the coordinate frames \mathcal{F}_c and \mathcal{F}_m are attached to the foci of the camera and mirror, respectively. Light rays incident to the focal point of the mirror (i.e., the origin of \mathcal{F}_m) are reflected into rays incident with the focal point of the camera (i.e., the origin of \mathcal{F}_c).

Without loss of generality, the subsequent development is based on the assumption that the reflection of four coplanar and non-collinear Euclidean feature points denoted by O_i of some stationary object is represented in the camera image plane by image space coordinates $u_i(t), v_i(t) \in \mathbb{R} \forall i = 1, 2, 3, 4$. The plane defined by the four feature points is denoted by π as depicted in Figure 5–1. The vector

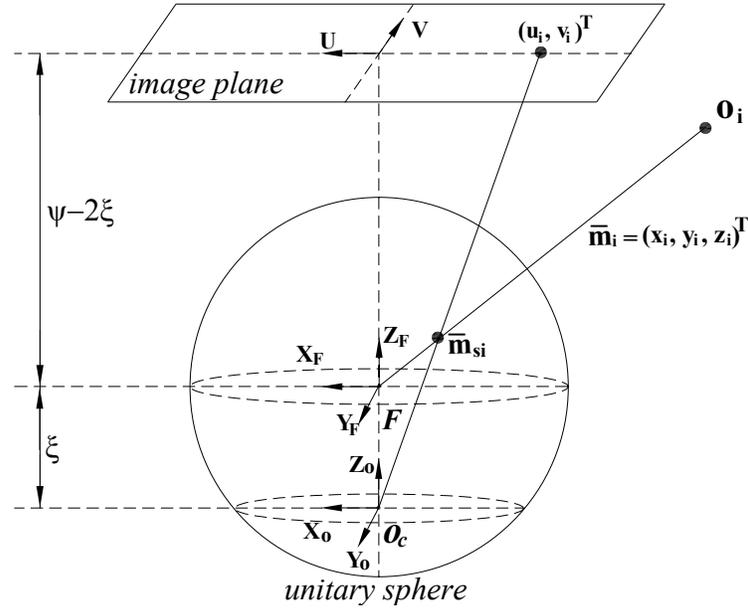


Figure 5-2: Projection model of the central catadioptric camera.

$\bar{m}_i(t) \in \mathbb{R}^3$ in Figure 5-2 is defined as

$$\bar{m}_i \triangleq \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T,$$

where $x_i(t), y_i(t), z_i(t) \in \mathbb{R}$ denote the Euclidean coordinate of the feature points O_i expressed in the frame \mathcal{F} which is affixed to the single effective viewpoint. The projected coordinate of $\bar{m}_i(t)$ can be expressed as

$$\bar{m}_{si} = \frac{\bar{m}_i}{L_i} = \begin{bmatrix} \frac{x_i}{L_i} & \frac{y_i}{L_i} & \frac{z_i}{L_i} \end{bmatrix}^T, \quad (5-1)$$

where $\bar{m}_{si}(t) \in \mathbb{R}^3$ denotes the Euclidean coordinates of O_i projected onto a unit spherical surface expressed in \mathcal{F} , and $L_i(t) \in \mathbb{R}$ is defined as

$$L_i = \sqrt{x_i^2 + y_i^2 + z_i^2}. \quad (5-2)$$

Based on the development in [36], $\bar{m}_{si}(t)$ can be expressed in the coordinate frame O_c , which is attached to the reprojection center, as

$$\bar{m}_{pi} = \left[\frac{x_i}{L_i} \quad \frac{y_i}{L_i} \quad \frac{z_i}{L_i} + \xi \right]^T, \quad (5-3)$$

where $\xi \in \mathbb{R}$ is a known intrinsic parameter of the central catadioptric camera that represents the distance between the single effective viewpoint and the reprojection center (see Figure 5-2). The normalized coordinates of $\bar{m}_{pi}(t)$ in (5-3), denoted by $m_{pi}(t) \in \mathbb{R}^3$, can be expressed as

$$m_{pi} = \left[\frac{\frac{x_i}{L_i}}{\frac{z_i}{L_i} + \xi} \quad \frac{\frac{y_i}{L_i}}{\frac{z_i}{L_i} + \xi} \quad 1 \right]^T = \left[\frac{x_i}{z_i + L_i\xi} \quad \frac{y_i}{z_i + L_i\xi} \quad 1 \right]^T. \quad (5-4)$$

By using (5-4) and the following relationship:

$$1 = \left(\frac{x_i}{L_i} \right)^2 + \left(\frac{y_i}{L_i} \right)^2 + \left(\frac{z_i}{L_i} \right)^2 \quad (5-5)$$

the coordinates of the feature points on the unit spherical surface $\bar{m}_{si}(t)$ can be determined as

$$\bar{m}_{si} = \left[\begin{array}{c} \xi + \frac{\sqrt{(m_{pix}^2 + m_{piy}^2)(1 - \xi^2) + 1}}{m_{pix}^2 + m_{piy}^2 + 1} m_{pix} \\ \xi + \frac{\sqrt{(m_{pix}^2 + m_{piy}^2)(1 - \xi^2) + 1}}{m_{pix}^2 + m_{piy}^2 + 1} m_{piy} \\ -\xi + \frac{\sqrt{(m_{pix}^2 + m_{piy}^2)(1 - \xi^2) + 1}}{m_{pix}^2 + m_{piy}^2 + 1} \end{array} \right]^T, \quad (5-6)$$

where $m_{pix}(t)$ and $m_{piy}(t) \in \mathbb{R}$ are the first two elements of $m_{pi}(t)$. The bijective mapping in (5-6) is unique (i.e., there is no sign ambiguity) because $0 < \xi \leq 1$ [36], and the geometry of the central catadioptric camera given in Figure 5-2 guarantees that the third element of $\bar{m}_{pi}(t)$ is positive; otherwise, the feature point O_i can not be projected onto the image plane.

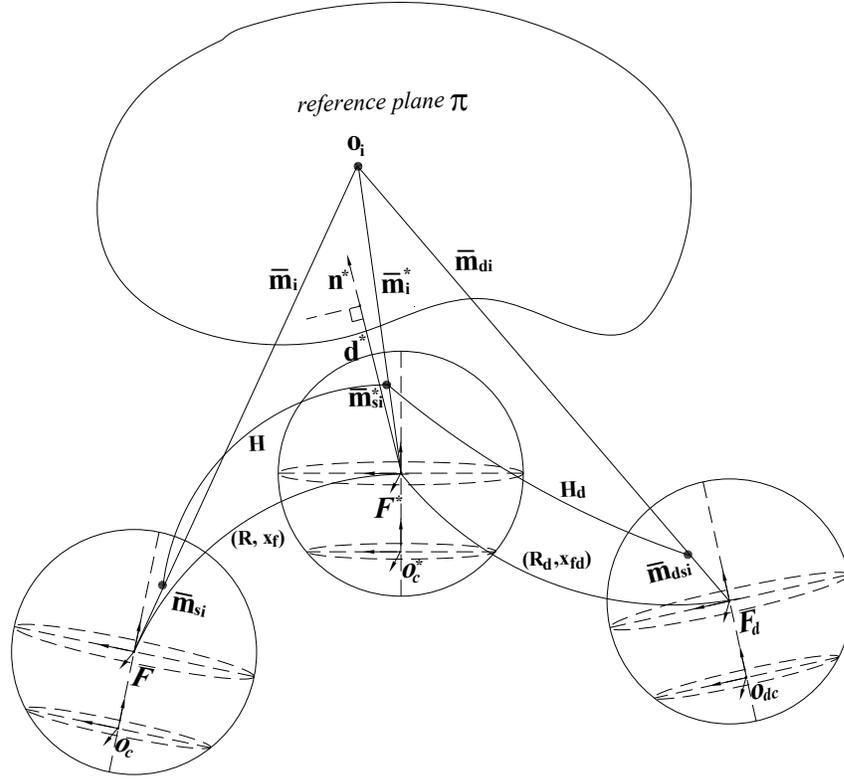


Figure 5-3: Camera relationships represented in homography.

As shown in Figure 5-3, the stationary coordinate frame \mathcal{F}^* denotes a constant reference camera pose that is defined by a reference image, and the coordinate frame \mathcal{F}_d represents the desired time-varying camera pose trajectory defined by a series of images (e.g., a video). The vectors $\bar{m}_i^*, \bar{m}_{di}(t) \in \mathbb{R}^3$ in Figure 5-3 are defined as

$$\bar{m}_i^* \triangleq \begin{bmatrix} x_i^* & y_i^* & z_i^* \end{bmatrix}^T \quad \bar{m}_{di} \triangleq \begin{bmatrix} x_{di}(t) & y_{di}(t) & z_{di}(t) \end{bmatrix}^T, \quad (5-7)$$

where $x_i^*, y_i^*, z_i^* \in \mathbb{R}$ and $x_{di}(t), y_{di}(t), z_{di}(t) \in \mathbb{R}$ denote the Euclidean coordinates of the feature points O_i expressed in the frames \mathcal{F}^* and \mathcal{F}_d , respectively.

The constant coordinates of \bar{m}_i^* can be projected onto a unit spherical surface expressed in \mathcal{F}^* and \mathcal{O}_c^* , respectively, as

$$\begin{aligned}\bar{m}_{si}^* &= \frac{\bar{m}_i^*}{L_i^*} = \left[\frac{x_i^*}{L_i^*} \quad \frac{y_i^*}{L_i^*} \quad \frac{z_i^*}{L_i^*} \right]^T \\ \bar{m}_{pi}^* &= \frac{\bar{m}_i^*}{L_i^*} = \left[\frac{x_i^*}{L_i^*} \quad \frac{y_i^*}{L_i^*} \quad \frac{z_i^*}{L_i^*} + \xi \right]^T,\end{aligned}\quad (5-8)$$

and the time-varying coordinates $\bar{m}_{di}(t)$ can be projected onto a unit spherical surface expressed in \mathcal{F}_d and \mathcal{O}_{dc} , respectively, as

$$\begin{aligned}\bar{m}_{dsi} &= \frac{\bar{m}_{di}}{L_{di}} = \left[\frac{x_{di}}{L_{di}} \quad \frac{y_{di}}{L_{di}} \quad \frac{z_{di}}{L_{di}} \right]^T \\ \bar{m}_{dpi} &= \left[\frac{x_{di}}{L_{di}} \quad \frac{y_{di}}{L_{di}} \quad \frac{z_{di}}{L_{di}} + \xi \right]^T,\end{aligned}$$

where \bar{m}_{si}^* , \bar{m}_{pi}^* , $\bar{m}_{dsi}(t)$, $\bar{m}_{dpi}(t) \in \mathbb{R}^3$, and L_i^* , $L_{di}(t) \in \mathbb{R}$ are defined as

$$L_i^* = \sqrt{x_i^{*2} + y_i^{*2} + z_i^{*2}} \quad L_{di} = \sqrt{x_{di}^2 + y_{di}^2 + z_{di}^2}.\quad (5-9)$$

The normalized coordinates of $\bar{m}_{dsi}(t)$ denoted as $m_{dsi}(t) \in \mathbb{R}^3$ is defined as

$$m_{dsi} \triangleq \left[\frac{x_{di}(t)}{z_{di}(t)} \quad \frac{y_{di}(t)}{z_{di}(t)} \quad 1 \right]^T,\quad (5-10)$$

which will be used in the controller development. The signal $m_{dsi}(t)$ in (5-10) is measurable because $\bar{m}_{dsi}(t)$ can be computed from the measurable and bounded pixel coordinates $p_{di}(t)$ using similar projective relationships as in (5-6) and (5-17). The normalized coordinates of \bar{m}_{pi}^* , $\bar{m}_{dpi}(t)$ denoted as m_{pi}^* , $m_{dpi}(t) \in \mathbb{R}^3$, respectively, are defined as

$$\begin{aligned}m_{pi}^* &= \left[\frac{x_i^*}{z_i^* + L_i^* \xi} \quad \frac{y_i^*}{z_i^* + L_i^* \xi} \quad 1 \right]^T \\ m_{dpi} &= \left[\frac{x_{di}}{z_{di} + L_{di} \xi} \quad \frac{y_{di}}{z_{di} + L_{di} \xi} \quad 1 \right]^T.\end{aligned}\quad (5-11)$$

From standard Euclidean geometry, the relationships between $\bar{m}_i(t)$, $\bar{m}_{di}(t)$ and \bar{m}_i^* can be determined as

$$\bar{m}_i = x_f + R\bar{m}_i^* \quad \bar{m}_{di} = x_{fd} + R_d\bar{m}_i^*, \quad (5-12)$$

where $x_f(t)$, $x_{fd}(t) \in \mathbb{R}^3$ denote the translation vectors expressed in \mathcal{F} and \mathcal{F}_d , respectively, and $R(t)$, $R_d(t) \in SO(3)$ denote the orientation of \mathcal{F}^* with respect to \mathcal{F} and \mathcal{F}_d , respectively. As also illustrated in Figure 5-3, $n^* \in \mathbb{R}^3$ denotes the constant unit normal to the plane π , and the constant distance from the origin of \mathcal{F}^* to π along the unit normal n^* is denoted by $d^* \in \mathbb{R}$ is defined as

$$d^* \triangleq n^{*T} \bar{m}_i^*. \quad (5-13)$$

By using (5-13), the relationships in (5-12) can be expressed as

$$\bar{m}_i = H\bar{m}_i^* \quad \bar{m}_{di} = H_d\bar{m}_i^*, \quad (5-14)$$

where $H(t)$, $H_d(t) \in \mathbb{R}^{3 \times 3}$ are the Euclidean homographies defined as

$$H = R + \frac{x_f}{d^*} n^{*T} \quad H_d = R_d + \frac{x_{fd}}{d^*} n^{*T}. \quad (5-15)$$

Based on (5-1) and (5-8), the relationship between the Euclidean coordinates in (5-14) can be expressed in terms of the unit spherical surface coordinates as

$$\bar{m}_{si} = \alpha_i H \bar{m}_{si}^* \quad \bar{m}_{dsi} = \alpha_{di} H_d \bar{m}_{si}^*, \quad (5-16)$$

where $\alpha_i(t) \triangleq \frac{L_i^*}{L_i(t)} \in \mathbb{R}$ and $\alpha_{di}(t) \triangleq \frac{L_i^*}{L_{di}(t)}$ are scaling terms.

5.3 Euclidean Reconstruction

The homogenous pixel coordinates of the features points with respect to the camera frames \mathcal{F} , \mathcal{F}^* and \mathcal{F}_d are denoted as $p_i(t)$, p_i^* and $p_{di}(t) \in \mathbb{R}^3$, respectively. They can be related to the normalized coordinates $m_{pi}(t)$, m_{pi}^* and $m_{dpi}(t)$ via the

following linear relationship as

$$p_i = Am_{pi} \quad p_i^* = Am_{pi}^* \quad p_{di} = Am_{dpi}, \quad (5-17)$$

where $A \in \mathbb{R}^{3 \times 3}$ contains the calibrated intrinsic parameters of the camera and the mirror (see Barreto and Araujo [35]). Since the camera and mirror are calibrated (i.e., A is known), $m_{pi}(t)$, m_{pi}^* and $m_{dpi}(t)$ can be computed from the measurable pixel coordinates $p_i(t)$, p_i^* and $p_{di}(t)$ based on (5-17). The expression given in (5-6) can then be used to compute $\bar{m}_{si}(t)$ from $m_{pi}(t)$. Similarly, \bar{m}_{si}^* and $\bar{m}_{dsi}(t)$ can be computed from m_{pi}^* and $m_{dpi}(t)$, respectively. Then based on (5-16), a set of 12 linearly independent equations given by the 4 feature point pairs $(p_i^*, p_i(t))$ with 3 independent equations per feature point can be developed to determine the homography up to a scalar multiple. Various methods can then be applied (e.g., see Faugeras and Lustman [93] and Zhang and Hanson [94]) to decompose the Euclidean homography to obtain $\alpha_i(t)$, $H(t)$, $R(t)$, $\frac{x_f(t)}{d^*}$, and n^* . Similarly, $\alpha_{di}(t)$, $H_d(t)$, $R_d(t)$, $\frac{x_{fd}(t)}{d^*}$ can be obtained from (5-16) using $(p_i^*, p_{di}(t))$. The rotation matrices $R(t)$, $R_d(t)$ and the depth ratios $\alpha_i(t)$, $\alpha_{di}(t)$ will be used in the subsequent control design.

5.4 Control Objective

The control objective is for a camera attaching to an object (i.e., camera-in-hand configuration) which can be identified by a planar patch of feature points to track a desired trajectory that is determined from a sequence of desired images taken during the a priori camera motion. This objective is based on the assumption that the linear and angular velocities of the camera are control inputs that can be independently controlled (i.e., unconstrained motion) and that the camera is calibrated (i.e., A is known). The control objective can be stated as $\bar{m}_i(t) \rightarrow \bar{m}_{di}(t)$ (i.e., Euclidean feature points on π track the corresponding feature points on π_d). Equivalently, the control objective can also be stated in terms of the

rotation and translation of the object as $x_f(t) \rightarrow x_{fd}(t)$ and $R(t) \rightarrow R_d(t)$. As stated previously, $R(t)$ and $R_d(t)$ can be computed by decomposing the projective homographies in (5–16). Once these rotation matrices have been determined, the unit quaternion parameterization is used to describe the rotation matrix. This parameterization facilitates the subsequent problem formulation, control development, and stability analysis since the unit quaternion provides a global nonsingular parameterization of the corresponding rotation matrices.

To quantify the error between the actual and desired camera orientations, the mismatch between the rotation matrices $R(t)$ and $R_d(t)$, denoted by $\tilde{R}(t) \in \mathbb{R}^3$, is defined as

$$\tilde{R} = RR_d^T. \quad (5-18)$$

Given the rotation matrices $R(t)$ and $R_d(t)$ from the homography decomposition, the corresponding unit quaternions $q(t)$ and $q_d(t)$ can be computed by using the numerically robust method (e.g., see [14] and [15]) as

$$R(q) = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T - 2q_0 q_v^\times \quad (5-19)$$

$$R_d(q_d) = (q_{0d}^2 - q_{vd}^T q_{vd}) I_3 + 2q_{vd} q_{vd}^T - 2q_{0d} q_{vd}^\times, \quad (5-20)$$

where I_3 is the 3×3 identity matrix, and the notation $q_v^\times(t)$ denotes the skew-symmetric form of the vector $q_v(t)$ as in (2–10). Based on (5–18)–(5–20), the rotation mismatch can be expressed as

$$\tilde{R} = (\tilde{q}_0^2 - \tilde{q}_v^T \tilde{q}_v) I_3 + 2\tilde{q}_v \tilde{q}_v^T - 2\tilde{q}_0 \tilde{q}_v^\times, \quad (5-21)$$

where the error quaternion $(\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ is defined as [23]

$$\tilde{q}_0 = q_0 q_{0d} + q_v^T q_{vd} \quad (5-22)$$

$$\tilde{q}_v = q_{0d} q_v - q_0 q_{vd} + q_v^\times q_{vd}.$$

Based on (5–18) and (5–21), the rotation tracking control objective $R(t) \rightarrow R_d(t)$ can be formulated as

$$\|\tilde{q}_v(t)\| \rightarrow 0 \implies \tilde{R}(t) \rightarrow I_3 \quad \text{as } t \rightarrow \infty. \quad (5-23)$$

To quantify the position mismatch between the actual and desired camera, the translation tracking error $e(t) \in \mathbb{R}^3$ is defined as

$$e = m_e - m_{ed} = \begin{bmatrix} \frac{x_i}{z_i} - \frac{x_{di}}{z_{di}} & \frac{y_i}{z_i} - \frac{y_{di}}{z_{di}} & \ln \left(\frac{z_i}{z_{di}} \right) \end{bmatrix}^T, \quad (5-24)$$

where $m_e(t), m_{ed}(t) \in \mathbb{R}^3$ are defined as

$$m_e \triangleq \begin{bmatrix} m_{e1} & m_{e2} & m_{e3} \end{bmatrix}^T = \begin{bmatrix} \frac{x_i}{z_i} & \frac{y_i}{z_i} & \ln z_i \end{bmatrix}^T \quad (5-25)$$

$$m_{ed} \triangleq \begin{bmatrix} m_{ed1} & m_{ed2} & m_{ed3} \end{bmatrix}^T = \begin{bmatrix} \frac{x_{di}}{z_{di}} & \frac{y_{di}}{z_{di}} & \ln z_{di} \end{bmatrix}^T. \quad (5-26)$$

Based on (5–23) and (5–24), the subsequent control development targets the following objectives:

$$\|\tilde{q}_v(t)\| \rightarrow 0 \quad \text{and} \quad \|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (5-27)$$

The error signal $\tilde{q}_v(t)$ is measurable since it can be computed from the $R(t)$ and $R_d(t)$ as in [24]. The first two elements of the translation error $e(t)$ are measurable because

$$m_{e1} - m_{ed1} = \frac{x_i/L_i}{z_i/L_i} - \frac{x_{di}/L_{di}}{z_{di}/L_{di}} \quad m_{e2} - m_{ed2} = \frac{y_i/L_i}{z_i/L_i} - \frac{y_{di}/L_{di}}{z_{di}/L_{di}},$$

where $\frac{x_i(t)}{L_i(t)}, \frac{y_i(t)}{L_i(t)}, \frac{z_i(t)}{L_i(t)}$ can be computed from (5–6) and (5–17), and $\frac{x_{di}(t)}{L_{di}(t)}, \frac{y_{di}(t)}{L_{di}(t)}, \frac{z_{di}(t)}{L_{di}(t)}$ can be computed from similar relationships. The third element of the translation error is also measurable since

$$\frac{z_i}{z_{di}} = \frac{(L_i^*/L_{di})}{z_{di}/L_{di}} \frac{z_i/L_i}{(L_i^*/L_i)} = \frac{\alpha_{di}}{\alpha_i} \frac{z_i/L_i}{z_{di}/L_{di}},$$

where $\frac{z_i(t)}{L_i(t)}$ can be computed from (5–6) and (5–17), $\frac{z_{di}(t)}{L_{di}(t)}$ can be computed from similar relationships, and the depth ratios $\alpha_i(t)$, $\alpha_{di}(t)$ can be obtained from the homography decompositions in (5–16).

5.5 Control Development

5.5.1 Open-Loop Error System

The open-loop rotation error system can be developed as [23]

$$\dot{\tilde{q}} = \frac{1}{2} \begin{bmatrix} -\tilde{q}_v^T \\ \tilde{q}_0 I_3 + \tilde{q}_v^\times \end{bmatrix} (\omega_c - \tilde{R}\omega_{cd}), \quad (5-28)$$

where $\tilde{q}(t) = (\tilde{q}_0(t), \tilde{q}_v^T(t))^T$, $\omega_c(t) \in \mathbb{R}^3$ denotes the camera angular velocity control input, and $\omega_{cd}(t) \in \mathbb{R}^3$ denotes the desired angular velocity of the camera that is assumed to be a priori generated as a bounded and continuous function (see [10] for a discussion regarding the development of a smooth desired trajectory from a series of images).

Based on (5–7) and (5–10), the derivative of $\bar{m}_{di}(t)$ is obtained as

$$\dot{\bar{m}}_{di} = z_i^* \frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right), \quad (5-29)$$

where $\beta_{di}(t) \in \mathbb{R}$ is defined as

$$\beta_{di} \triangleq \frac{z_i^*}{z_{di}} = \alpha_{di} \frac{z_i^*/L_i^*}{z_{di}/L_{di}}.$$

Based on (5–29) and the fact that [22]

$$\dot{\bar{m}}_{di} = -v_{cd} + \bar{m}_{di}^\times \omega_{cd}, \quad (5-30)$$

where $v_{cd}(t) \in \mathbb{R}^3$ denotes the desired linear velocity of the camera expressed in \mathcal{F}_d , it can be obtained that

$$v_{cd} = \frac{z_i^*}{\beta_{di}} m_{dsi}^\times \omega_{cd} - z_i^* \frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right). \quad (5-31)$$

After differentiating both sides of (5-24) and using the equations (5-30), (5-31) and

$$\dot{\bar{m}}_i = -v_c + \bar{m}_i^\times \omega_c,$$

the open-loop translation error system can be derived as

$$z_i^* \dot{e} = -\beta_i L_v v_c + z_i^* L_{v\omega} \omega_c - z_i^* \beta_{di} L_{vd} \frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right), \quad (5-32)$$

where $v_c(t) \in \mathbb{R}^3$ denotes the linear velocity input of the camera with respect to \mathcal{F}^* expressed in \mathcal{F} , the Jacobian-like matrices $L_v(t)$, $L_{vd}(t)$, $L_{v\omega}(t) \in \mathbb{R}^{3 \times 3}$ are defined as

$$L_v = \begin{bmatrix} 1 & 0 & -m_{e1} \\ 0 & 1 & -m_{e2} \\ 0 & 0 & 1 \end{bmatrix} \quad L_{vd} = \begin{bmatrix} 1 & 0 & -m_{ed1} \\ 0 & 1 & -m_{ed2} \\ 0 & 0 & 1 \end{bmatrix} \quad (5-33)$$

$$L_{v\omega} = \begin{bmatrix} m_{e1} m_{e2} & -1 - m_{e1}^2 & m_{e2} \\ 1 + m_{e2}^2 & -m_{e1} m_{e2} & -m_{e1} \\ -m_{e2} & m_{e1} & 0 \end{bmatrix}, \quad (5-34)$$

and $\beta_i(t) \in \mathbb{R}$ is defined as

$$\beta_i \triangleq \frac{z_i^*}{z_i} = \alpha_i \frac{z_i^*/L_i^*}{z_i/L_i}.$$

5.5.2 Closed-Loop Error System

Based on the open-loop rotation error system in (5-28) and the subsequent Lyapunov-based stability analysis, the angular velocity controller is designed as

$$\omega_c = -K_\omega \tilde{q}_v + \tilde{R} \omega_{cd}, \quad (5-35)$$

where $K_\omega \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains.

From (5–28) and (5–35), the rotation closed-loop error system can be determined as

$$\begin{aligned}\dot{\tilde{q}}_0 &= \frac{1}{2} \tilde{q}_v^T K_\omega \tilde{q}_v \\ \dot{\tilde{q}}_v &= -\frac{1}{2} (\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v.\end{aligned}\tag{5–36}$$

Based on (5–32), the translation control input $v_c(t)$ is designed as

$$v_c = \beta_i L_v^{-1} \left[K_v e + \hat{z}_i^* \left(L_{v\omega} \omega_c - \beta_{di} L_{vd} \frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right) \right) \right],\tag{5–37}$$

where $K_v \in \mathbb{R}^{3 \times 3}$ denotes a diagonal matrix of positive constant control gains. In

(5–37), the parameter estimate $\hat{z}_i^*(t) \in \mathbb{R}$ for the unknown constant z_i^* is defined as

$$\dot{\hat{z}}_i^* = \gamma e^T \left(L_{v\omega} \omega_c - \beta_{di} L_{vd} \frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right) \right),\tag{5–38}$$

where $\gamma \in \mathbb{R}$ denotes a positive constant adaptation gain. By using (5–32) and

(5–37), the translation closed-loop error system is

$$z_i^* \dot{e} = -K_v e + \left(L_{v\omega} \omega_c - \beta_{di} L_{vd} \frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right) \right) \tilde{z}_i^*,\tag{5–39}$$

where $\tilde{z}_i^*(t) \in \mathbb{R}$ denotes the following parameter estimation error:

$$\tilde{z}_i^* = z_i^* - \hat{z}_i^*.\tag{5–40}$$

5.5.3 Stability Analysis

Theorem 5.1: The controller given in (5–35) and (5–37), along with the adaptive update law in (5–38) ensures asymptotic tracking in the sense that

$$\|\tilde{q}_v(t)\| \rightarrow 0, \quad \|e(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty.\tag{5–41}$$

Proof: Let $V(t) \in \mathbb{R}$ denote the following differentiable non-negative function (i.e., a Lyapunov candidate):

$$V = \tilde{q}_v^T \tilde{q}_v + (1 - \tilde{q}_0)^2 + \frac{\tilde{z}_i^*}{2} e^T e + \frac{1}{2\gamma} \tilde{z}_i^{*2}. \quad (5-42)$$

The time-derivative of $V(t)$ can be determined as

$$\begin{aligned} \dot{V} &= -\tilde{q}_v^T (\tilde{q}_0 I_3 + \tilde{q}_v^\times) K_\omega \tilde{q}_v - (1 - \tilde{q}_0) \tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e \\ &\quad + e^T \left(L_{v\omega} \omega_c - \beta_{di} L_{vd} \frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right) \right) \tilde{z}_i^* \\ &\quad - \tilde{z}_i^* e^T \left(L_{v\omega} \omega_c - \beta_{di} L_{vd} \frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right) \right) \\ &= -\tilde{q}_v^T (\tilde{q}_0 I_3 + \tilde{q}_v^\times + (1 - \tilde{q}_0) I_3) K_\omega \tilde{q}_v - e^T K_v e \\ &= -\tilde{q}_v^T K_\omega \tilde{q}_v - e^T K_v e, \end{aligned} \quad (5-43)$$

where (5-36) and (5-38)-(5-40) were utilized. Based on (5-42) and (5-43), $e(t)$, $\tilde{q}_v(t)$, $\tilde{q}_0(t)$, $\tilde{z}_i^*(t) \in \mathcal{L}_\infty$ and $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_2$. Since $\tilde{z}_i^*(t) \in \mathcal{L}_\infty$, it is clear from (5-40) that $\hat{z}_i^*(t) \in \mathcal{L}_\infty$. Based on the fact that $e(t) \in \mathcal{L}_\infty$, (5-24) and (5-25) can be used to prove that $m_e(t) \in \mathcal{L}_\infty$, and $L_v(t)$, $L_v^{-1}(t)$, $L_{v\omega}(t) \in \mathcal{L}_\infty$. Since $\tilde{q}_v(t) \in \mathcal{L}_\infty$ and $\omega_{cd}(t)$ is a bounded function, (5-35) can be used to conclude that $\omega_c(t) \in \mathcal{L}_\infty$. From $\hat{z}_i^*(t)$, $e(t)$, $m_e(t)$, $L_v(t)$, $L_v^{-1}(t)$, $L_{v\omega}(t) \in \mathcal{L}_\infty$ and $L_{vd}(t)$, $\frac{d}{dt} \left(\frac{1}{\beta_{di}} m_{dsi} \right)$ are bounded, (5-37) can be utilized to prove that $v_c(t) \in \mathcal{L}_\infty$. From the previous results, (5-28)-(5-32) can be used to prove that $\dot{e}(t)$, $\dot{\tilde{q}}_v(t) \in \mathcal{L}_\infty$. Since $e(t)$, $\tilde{q}_v(t) \in \mathcal{L}_\infty \cap \mathcal{L}_2$, and $\dot{e}(t)$, $\dot{\tilde{q}}_v(t) \in \mathcal{L}_\infty$, Barbalat's Lemma [96] can be used to conclude the result given in (5-41).

CHAPTER 6
VISUAL SERVO CONTROL IN THE PRESENCE OF CAMERA CALIBRATION
ERROR

6.1 Introduction

Hu et al. [14] introduced a new quaternion-based visual servo controller for the rotation error system, provided the camera calibration parameters are exactly known. Since the results by Malis and Chaumette [21] and Fang et al. [87] rely heavily on properties of the rotation parameterization to formulate state estimates and a measurable closed-loop error system, the research in this chapter is motivated by the question: *Can state estimates and a measurable closed-loop error system be crafted in terms of the quaternion parameterization when the camera calibration parameters are unknown?* To answer this question, a contribution of this chapter is the development of a quaternion-based estimate for the rotation error system that is related to the actual rotation error, the development of a new closed-loop error system, and a new Lyapunov-based analysis that demonstrates the stability of the quaternion error system. One of the challenges is to develop a quaternion estimate from an estimated rotation matrix that is not a true rotation matrix in general. To address this challenge, the similarity relationship between the estimated and actual rotation matrices is used (as in [21] and [87]) to construct the relationship between the estimated and actual quaternions. A Lyapunov-based stability analysis is provided that indicates a unique controller can be developed to achieve the regulation result despite a sign ambiguity in the developed quaternion estimate. Simulation results are provided in Section 6.7 that illustrate the performance of the developed controller.

6.2 Feedback Control Measurements

The objective in this chapter is to develop a kinematic controller (i.e., the control inputs are considered the linear and angular camera velocities) to ensure the position/orientation of the camera coordinate frame \mathcal{F} is regulated to the desired position/orientation \mathcal{F}^* . The camera geometry is shown in Figure 2-2 and the corresponding Euclidean and image-space relationships are developed in Sections 2.1 and 2.2. The only required sensor measurements for the control development are the image coordinates of the determined feature points (i.e., measurement of the signals in (2-5)), where the static feature point coordinates in the desired image are given a priori. By measuring the current image feature points and given the desired feature points, the relationship in (2-6) can be used to determine the normalized Euclidean coordinates of O_i provided the intrinsic camera calibration matrix is perfectly known. Unfortunately, any uncertainty in A will lead to a corrupted measurement of $m_i(t)$ and m_i^* . The computed normalized coordinates are actually estimates, denoted by $\hat{m}_i(t)$, $\hat{m}_i^* \in \mathbb{R}^3$, of the true values since only a best-guess estimate of A , denoted by $\hat{A} \in \mathbb{R}^{3 \times 3}$, is available in practice. The normalized coordinate estimates can be expressed as [21]

$$\hat{m}_i = \hat{A}^{-1} p_i = \tilde{A} m_i \quad (6-1)$$

$$\hat{m}_i^* = \hat{A}^{-1} p_i^* = \tilde{A} m_i^*, \quad (6-2)$$

where the calibration error matrix $\tilde{A} \in \mathbb{R}^{3 \times 3}$ is defined as

$$\tilde{A} = \hat{A}^{-1} A = \begin{bmatrix} \tilde{A}_{11} & \tilde{A}_{12} & \tilde{A}_{13} \\ 0 & \tilde{A}_{22} & \tilde{A}_{23} \\ 0 & 0 & 1 \end{bmatrix}, \quad (6-3)$$

where \tilde{A}_{11} , \tilde{A}_{12} , \tilde{A}_{13} , \tilde{A}_{22} , $\tilde{A}_{23} \in \mathbb{R}$ denote unknown intrinsic calibration mismatch constants. Since $m_i(t)$ and m_i^* can not be exactly determined, the estimates in

(6-1) and (6-2) can be substituted into (2-4) to obtain the following relationship

$$\hat{m}_i = \alpha_i \hat{H} \hat{m}_i^*, \quad (6-4)$$

where $\hat{H}(t) \in \mathbb{R}^{3 \times 3}$ denotes the estimated Euclidean homography defined as

$$\hat{H} = \tilde{A} H \tilde{A}^{-1}. \quad (6-5)$$

Since $\hat{m}_i(t)$ and \hat{m}_i^* can be determined from (6-1) and (6-2), a set of twelve linear equations can be developed from the four image point pairs, and (6-4) can be used to solve for $\hat{H}(t)$.

As stated in [21], provided additional information is available (e.g., at least 4 vanishing points), various techniques (e.g., see Faugeras and Lustman [93] and Zhang and Hanson [94]) can be used to decompose $\hat{H}(t)$ to obtain the estimated rotation and translation components as

$$\hat{H} = \tilde{A} R \tilde{A}^{-1} + \tilde{A} x_h n^{*T} \tilde{A}^{-1} = \hat{R} + \hat{x}_h \hat{n}^{*T}, \quad (6-6)$$

where $\hat{R}(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$\hat{R} = \tilde{A} R \tilde{A}^{-1}, \quad (6-7)$$

and $\hat{x}_h(t) \in \mathbb{R}^3$, $\hat{n}^* \in \mathbb{R}^3$ denote the estimate of $x_h(t)$ and n^* , respectively, defined as

$$\hat{x}_h = \sigma \tilde{A} x_h \quad \hat{n}^* = \frac{1}{\sigma} \tilde{A}^{-T} n^*,$$

where $\sigma \in \mathbb{R}$ denotes the following positive constant $\sigma = \left\| \tilde{A}^{-T} n^* \right\|$.

For the four vanishing points (see Almansa et al. [103] for a description of how to determine vanishing points in an image), $d^* = \infty$, so that

$$H = R + \frac{x_f}{d^*} n^{*T} = R \quad (6-8)$$

and therefore

$$\hat{H} = \tilde{A}R\tilde{A}^{-1} = \hat{R}. \quad (6-9)$$

Twelve linear equations can be obtained based on (6-4) for the four vanishing points. Assume the 3rd row 3rd column element of $\hat{H}(t)$, denoted as $\hat{H}_{33}(t) \in \mathbb{R}$, is not equal to zero (w.l.o.g.). The normalized matrix $\hat{H}_n(t) \in \mathbb{R}^{3 \times 3}$, defined as

$$\hat{H}_n = \frac{\hat{H}}{\hat{H}_{33}}, \quad (6-10)$$

can be computed based on these twelve linear equations. Based on (6-9)

$$\det(\hat{H}) = \det(\tilde{A}) \det(R) \det(\tilde{A}^{-1}) = 1. \quad (6-11)$$

From (6-10) and (6-11),

$$\hat{H}_{33}^3 \det(\hat{H}_n) = 1, \quad (6-12)$$

and hence,

$$\hat{H} = \frac{\hat{H}_n}{\sqrt[3]{\det(\hat{H}_n)}}, \quad (6-13)$$

which is equal to $\hat{R}(t)$.

6.3 Control Objective

As stated previously, the objective in this chapter is to develop a kinematic controller to ensure the pose of the camera coordinate frame \mathcal{F} is regulated to the desired pose \mathcal{F}^* despite uncertainty in the intrinsic camera calibration matrix. This objective is based on the assumption that the linear and angular velocities of the camera are control inputs that can be independently controlled (i.e., unconstrained motion). For example, the linear and angular camera velocities could be controlled by the end-effector of a robotic manipulator. In addition to uncertainty in the intrinsic camera calibration, uncertainty could also exist in the extrinsic camera calibration (e.g., the uncertainty in the rotation and translation of the camera with respect to the robot end-effector). The development in this chapter could

be directly modified as described in [21] and [87] to compensate for the extrinsic calibration. Therefore, the effects of a mismatch in the extrinsic calibration are not considered in the subsequent development for simplicity.

In the Euclidean space, the rotation control objective can be quantified as

$$R(t) \rightarrow I_3 \quad \text{as } t \rightarrow \infty. \quad (6-14)$$

The subsequent development is formulated in terms of the four dimensional unit quaternion $q(t)$. Given the rotation matrix $R(t)$, the corresponding unit quaternion $q(t)$ can be computed by using the numerically robust method presented in Section 2.3. From (2-12) and (2-15), the rotation regulation objective in (6-14) can also be quantified as the desire to regulate $q_v(t)$ as

$$\|q_v(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (6-15)$$

The focus and contribution of this chapter lies in the ability to develop and prove the stability of a quaternion-based rotation controller in the presence of uncertainty in the camera calibration. The translation controller developed by Fang et al. [87] is also presented and incorporated in the stability analysis to provide an example of how the new class of quaternion-based rotation controllers can be used in conjunction with translation controllers that are robust to camera calibration uncertainty including (for example): the asymptotic translation controllers in [21], and the exponential translation controllers in [87]. The translation error, denoted by $e(t) \in \mathbb{R}^3$, is defined as

$$e = \frac{z_i}{z_i^*} m_i - m_i^*, \quad (6-16)$$

where i can be chosen as any number within $\{1, \dots, 4\}$. The translation objective can be stated as

$$\|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (6-17)$$

The subsequent section will target the control development based on the objectives in (6–15) and (6–17).

6.4 Quaternion Estimation

A method is presented in this section to develop a quaternion-based rotation estimate that can be related to the actual rotation mismatch to facilitate the control development.

6.4.1 Estimate Development

The unit quaternion is related to the angle-axis representation as

$$q_0 = \cos\left(\frac{\theta}{2}\right) \quad q_v = u \sin\left(\frac{\theta}{2}\right), \quad (6-18)$$

where $\theta(t)$ and $u(t)$ are the corresponding rotation angle and unit axis. By using (2–12) and (2–15), the first element of the quaternion can also be expressed in terms of the rotation matrix $R(t)$ as

$$q_0^2 = \frac{\text{tr}(R) + 1}{4},$$

where $q_0(t)$ is restricted to be non-negative as

$$q_0 = \frac{1}{2} \sqrt{1 + \text{tr}(R)} \quad (6-19)$$

without loss of generality (this restriction enables the minimum rotation to be obtained), and $\text{tr}(R)$ denotes the trace of $R(t)$. Based on (6–18) and (6–19), $q_v(t)$ can be determined as

$$q_v = \pm u \sqrt{1 - \cos^2\left(\frac{\theta}{2}\right)} = \pm \frac{1}{2} u \sqrt{3 - \text{tr}(R)}, \quad (6-20)$$

where the rotation axis $u(t)$ is the unit eigenvector with respect to the eigenvalue 1 of $R(t)$. For the quaternion vector in (6–20), the sign ambiguity can be resolved. Specifically, (2–15) can be used to develop the following expression:

$$R^T - R = 4q_0 q_v^\times. \quad (6-21)$$

Since the sign of $q_0(t)$ is restricted (i.e., assumed to be) positive, then a unique solution for $q_v(t)$ can be determined from (6–20) and (6–21).

Based on the similarity between $\hat{R}(t)$ and $R(t)$ as stated in (6–7), the expressions in (6–19) and (6–20) provide motivation to develop the quaternion estimate as

$$\hat{q}_0 = \frac{1}{2} \sqrt{1 + \text{tr}(\hat{R})} \quad (6-22)$$

$$\hat{q}_v = \pm \hat{u} \sin\left(\frac{\hat{\theta}}{2}\right) = \pm \frac{1}{2} \hat{u} \sqrt{3 - \text{tr}(\hat{R})}. \quad (6-23)$$

In (6–22) and (6–23), $\hat{R}(t)$ is the estimated rotation matrix introduced in (6–6) that is computed from the homography decomposition. Since $\hat{R}(t)$ is similar to $R(t)$ (see (6–7)), $\hat{R}(t)$ is guaranteed to have an eigenvalue of 1, where $\hat{u}(t)$ is the unit eigenvector that can be computed from the eigenvalue of 1. Since $\hat{R}(t)$ is not guaranteed to be a true rotation matrix (and it will not be in general), the relationships in (2–15) and (6–21) can not be developed and used to eliminate the sign ambiguity of the eigenvector $\hat{u}(t)$. However, the subsequent stability analysis and simulation results indicate that the same stability result is obtained invariant of the sign of $\hat{u}(t)$. Once the initial sign of $\hat{u}(t)$ is chosen, the same sign can be used for subsequent computations.

6.4.2 Estimate Relationships

Based on the fact that $\hat{R}(t)$ is similar to $R(t)$ (see (6–7)), the properties that similar matrices have the same trace and eigenvalues can be used to relate the quaternion estimate and the actual quaternion. Since similar matrices have the same trace, (6–19) and (6–22) can be used to conclude that

$$\hat{q}_0 = q_0. \quad (6-24)$$

As stated earlier, since similar matrices have the same eigenvalues, $\hat{R}(t)$ is guaranteed to have an eigenvalue of 1 with the associated eigenvector $\hat{u}(t)$. The following

relationships can be developed based on (6-7)

$$\hat{u} = \hat{R}\hat{u} = \tilde{A}R\tilde{A}^{-1}\hat{u}. \quad (6-25)$$

Premultiplying \tilde{A}^{-1} on both sides of (6-25) yields

$$\tilde{A}^{-1}\hat{u} = R\tilde{A}^{-1}\hat{u}. \quad (6-26)$$

Hence, $\tilde{A}^{-1}\hat{u}(t)$ is an eigenvector with respect to the eigenvalue 1 of $R(t)$ that can be expressed as

$$\tilde{A}^{-1}\hat{u} = \pm\gamma u, \quad (6-27)$$

where $\gamma \in \mathbb{R}$ is defined as

$$\gamma = \frac{1}{\|\tilde{A}u\|}. \quad (6-28)$$

Based on (6-20), (6-23), and (6-27), the estimated quaternion vector can now be related to the actual quaternion vector as

$$\hat{q}_v = \pm\gamma\tilde{A}q_v. \quad (6-29)$$

By using (2-12), (6-24), (6-28) and (6-29),

$$\hat{q}_0^2 + \|\hat{q}_v\|^2 = q_0^2 + \frac{1}{\|\tilde{A}u\|^2} \|\tilde{A}q_v\|^2. \quad (6-30)$$

Based on (6-28) and the fact that $u(t)$ is a unit vector,

$$\gamma = \frac{\|u\| \left\| \sin\left(\frac{\theta}{2}\right) \right\|}{\|\tilde{A}u\| \left\| \sin\left(\frac{\theta}{2}\right) \right\|} = \frac{\|q_v\|}{\|\tilde{A}q_v\|}. \quad (6-31)$$

From (6-28) and (6-31),

$$\hat{q}_0^2 + \|\hat{q}_v\|^2 = q_0^2 + \frac{\|q_v\|^2}{\|\tilde{A}q_v\|^2} \|\tilde{A}q_v\|^2 = 1.$$

6.5 Control Development

6.5.1 Rotation Control

The rotation open-loop error system can be developed by taking the time derivative of $q(t)$ as

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_v^T \\ q_0 I_3 + q_v^\times \end{bmatrix} \omega_c, \quad (6-32)$$

where $\omega_c(t) \in \mathbb{R}^3$ denotes the angular velocity of the camera with respect to \mathcal{F}^* expressed in \mathcal{F} . Based on the open-loop error system in (6-32) and the subsequent stability analysis, the angular velocity controller is designed as

$$\omega_c = -K_\omega \hat{q}_v, \quad (6-33)$$

where $K_\omega \in \mathbb{R}$ denotes a positive control gain. Substituting (6-33) into (6-32), the rotation closed-loop error system can be developed as

$$\dot{q}_0 = \frac{1}{2} K_\omega q_v^T \hat{q}_v \quad (6-34)$$

$$\dot{q}_v = -\frac{1}{2} K_\omega (q_0 I_3 + q_v^\times) \hat{q}_v. \quad (6-35)$$

6.5.2 Translation Control

The contribution of this chapter is the rotation estimate and associated control development. The translation controller developed in this section is provided for completeness. As stated previously, translation controllers such as the class developed by Malis and Chaumette [21] and Fang et al. [87] can be combined with the developed quaternion-based rotation controller. To facilitate the subsequent stability analysis for the six DOF problem, a translation controller proposed in [87] is provided in this section, which is given by

$$v_c = K_v \hat{e}, \quad (6-36)$$

where $K_v \in \mathbb{R}$ denotes a positive control gain, and $\hat{e}(t) \in \mathbb{R}^3$ is defined as

$$\hat{e} = \frac{z_i}{z_i^*} \hat{m}_i - \hat{m}_i^*, \quad (6-37)$$

where $\hat{m}_i(t)$ and \hat{m}_i^* can be computed from (6-1) and (6-2), respectively, and the ratio $\frac{z_i}{z_i^*}$ can be computed from the decomposition of the estimated Euclidean homography in (6-4). The open-loop translation error system can be determined as

$$\dot{e} = -\frac{1}{z_i^*} v_c - \omega_c^\times e + [m_i^*]^\times \omega_c. \quad (6-38)$$

After substituting (6-33) and (6-36) into (6-38), the resulting closed-loop translation error system can be determined as

$$\dot{e} = \left(-K_v \frac{1}{z_i^*} \tilde{A} + [K_\omega \hat{q}_v]^\times \right) e - K_\omega [m_i^*]^\times \hat{q}_v. \quad (6-39)$$

6.6 Stability Analysis

As stated previously, the quaternion estimate $\hat{q}_v(t)$ has a sign ambiguity, but either choice of the sign will yield the same stability result. The following analysis is developed for the case where

$$\hat{q}_v = \gamma \tilde{A} q_v. \quad (6-40)$$

A discussion is provided at the end of the analysis, that describes how the stability can be proven for the case when

$$\hat{q}_v = -\gamma \tilde{A} q_v.$$

Theorem 6.1: The controller given in (6-33) and (6-36) ensures asymptotic regulation in the sense that

$$\|q_v(t)\| \rightarrow 0, \quad \|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad (6-41)$$

provided K_v is selected sufficiently large (see the subsequent proof), and the following inequalities are satisfied

$$\lambda_{\min} \left\{ \frac{1}{2} (\tilde{A} + \tilde{A}^T) \right\} \geq \lambda_0 \quad (6-42)$$

$$\lambda_{\max} \left\{ \frac{1}{2} (\tilde{A} + \tilde{A}^T) \right\} \leq \lambda_1, \quad (6-43)$$

where $\lambda_0, \lambda_1 \in \mathbb{R}$ are positive constants, and $\lambda_{\min} \{\cdot\}$ and $\lambda_{\max} \{\cdot\}$ denote the minimal and maximal eigenvalues of $\frac{1}{2} (\tilde{A} + \tilde{A}^T)$, respectively.

Proof: Let $V(t) \in \mathbb{R}$ denote the following differentiable non-negative function (i.e., a Lyapunov candidate):

$$V = q_v^T q_v + (1 - q_0)^2 + e^T e. \quad (6-44)$$

After cancelling common terms, $\dot{V}(t)$ can be expressed as

$$\begin{aligned} \dot{V}_1 &= 2q_v^T \dot{q}_v - 2(1 - q_0)\dot{q}_0 + e^T \dot{e} \\ &= -\gamma K_\omega q_v^T (q_0 I_3 + q_v^\times) \tilde{A} q_v - \gamma K_\omega (1 - q_0) q_v^T \tilde{A} q_v \\ &\quad + e^T \left(-K_v \frac{1}{z_i^*} \tilde{A} + \gamma \left[K_\omega \tilde{A} q_v \right]^\times \right) e - \gamma K_\omega e^T [m_i^*]^\times \tilde{A} q_v \\ &= \gamma K_\omega q_v^T \left[-(q_0 I_3 + q_v^\times) - (1 - q_0) I_3 \right] \tilde{A} q_v \\ &\quad + e^T \left(-K_v \frac{1}{z_i^*} \tilde{A} + \gamma \left[K_\omega \tilde{A} q_v \right]^\times \right) e - \gamma K_\omega e^T [m_i^*]^\times \tilde{A} q_v \\ &= -\gamma K_\omega q_v^T \tilde{A} q_v - K_v \frac{1}{z_i^*} e^T \tilde{A} e + \gamma e^T \left(\left[K_\omega \tilde{A} q_v \right]^\times \right) e - \gamma K_\omega e^T [m_i^*]^\times \tilde{A} q_v. \end{aligned} \quad (6-45)$$

By using the inequality (6-42), the term $\gamma K_\omega q_v^T \tilde{A} q_v$ satisfies

$$\begin{aligned} \gamma K_\omega q_v^T \tilde{A} q_v &= \frac{1}{2} \gamma K_\omega q_v^T (\tilde{A} + \tilde{A}^T) q_v \\ &\geq \gamma K_\omega \lambda_{\min} \left\{ \frac{1}{2} (\tilde{A} + \tilde{A}^T) \right\} \|q_v\|^2 \geq \gamma K_\omega \lambda_0 \|q_v\|^2. \end{aligned} \quad (6-46)$$

Since

$$e^T \tilde{A} e = \frac{1}{2} e^T (\tilde{A} + \tilde{A}^T) e \geq \lambda_0 \|e\|^2,$$

the term $-K_v \frac{1}{z_i^*} e^T \tilde{A} e$ satisfies

$$-K_v \frac{1}{z_i^*} e^T \tilde{A} e \leq -K_v \frac{1}{z_i^*} \lambda_0 \|e\|^2. \quad (6-47)$$

Based on the property that $\|[\xi]^\times\|_2 = \|\xi\| \forall \xi \in \mathbb{R}^3$ (see Appendix D) and $\|\hat{q}_v\| \leq 1$, the term $\gamma e^T \left([K_\omega \tilde{A} q_v]^\times \right) e$ satisfies

$$\begin{aligned} \gamma e^T \left([K_\omega \tilde{A} q_v]^\times \right) e &= \gamma e^T ([K_\omega \hat{q}_v]^\times) e \\ &\leq \gamma \|K_\omega \hat{q}_v\|_2 \|e\|^2 \leq \gamma K_\omega \|\hat{q}_v\| \|e\|^2 \leq \gamma K_\omega \|e\|^2. \end{aligned} \quad (6-48)$$

From (6-31), the term $-\gamma K_\omega e^T [m_i^*]^\times \tilde{A} q_v$ satisfies

$$\begin{aligned} -\gamma K_\omega e^T [m_i^*]^\times \tilde{A} q_v &= -\frac{\|q_v\|}{\|\tilde{A} q_v\|} K_\omega e^T [m_i^*]^\times \tilde{A} q_v \\ &\leq \frac{\|q_v\|}{\|\tilde{A} q_v\|} K_\omega \| [m_i^*]^\times \|_2 \|e\| \|\tilde{A} q_v\| \\ &= K_\omega \|m_i^*\| \|q_v\| \|e\|. \end{aligned} \quad (6-49)$$

By using (6-46)-(6-49), the expression in (6-45) can be upper bounded as

$$\dot{V} \leq -\gamma K_\omega \lambda_0 \|q_v\|^2 - \frac{1}{z_i^*} (K_{v1} + K_{v2}) \lambda_0 \|e\|^2 + \gamma K_\omega \|e\|^2 + K_\omega \|m_i^*\| \|q_v\| \|e\|, \quad (6-50)$$

where the control gain K_v is separated into two different control gains as $K_v = K_{v1} + K_{v2}$. The following inequality can be obtained after completing the squares:

$$K_\omega \|m_i^*\| \|q_v\| \|e\| \leq \frac{z_i^* K_\omega^2 \|m_i^*\|^2}{4K_{v1} \lambda_0} \|q_v\|^2 + \frac{1}{z_i^*} K_{v1} \lambda_0 \|e\|^2. \quad (6-51)$$

From (6-51), the inequality (6-50) can be rewritten as

$$\dot{V} \leq -\frac{K_\omega \lambda_0 \gamma}{K_{v1}} \left(K_{v1} - \frac{z_i^* K_\omega \|m_i^*\|^2}{4\lambda_0^2 \gamma} \right) \|q_v\|^2 - \frac{\lambda_0}{z_i^*} \left(K_{v2} - \frac{z_i^* \gamma K_\omega}{\lambda_0} \right) \|e\|^2.$$

Based on the definition of $\gamma(t)$ in (6–28), the inequalities (6–42) and (6–43), and the assumption that m_i^* and z_i^* are bounded, there exist two positive bounding constant c_1 and $c_2 \in \mathbb{R}$ satisfying the following inequalities:

$$\frac{z_i^* K_\omega \|m_i^*\|^2}{4\lambda_0^2 \gamma} < c_1 \quad \text{and} \quad \frac{z_i^* \gamma K_\omega}{\lambda_0} < c_2$$

the control parameter K_v can be selected large enough to ensure that $\dot{V}(t)$ is negative semi-definite as

$$\dot{V} \leq -\frac{K_\omega \lambda_0 \gamma}{K_{v1}} \|q_v\|^2 - \frac{\lambda_0}{z_i^*} \|e\|^2. \quad (6-52)$$

Based on (6–44) and (6–52), standard signal chasing arguments can be used to conclude that the control inputs and all the closed-loop signals are bounded. The expression in (6–52) can also be used to conclude that $q_v(t)$ and $e(t) \in \mathcal{L}_2$. Since $q_v(t), \dot{q}_v(t), e(t), \dot{e}(t) \in \mathcal{L}_\infty$ and $q_v(t), e(t) \in \mathcal{L}_2$, Barbalat’s Lemma [96] can be used to prove the result given in (6–41). ■

By modifying the Lyapunov function in (6–44) as

$$V = q_v^T q_v + (1 + q_0)^2 + e^T e,$$

the same stability analysis arguments can be used to prove Theorem 6.1 for the case when

$$\hat{q}_v = -\gamma \tilde{A} q_v.$$

Since the sign ambiguity in (6–29) does not affect the control development and stability analysis, only the positive sign in (6–29) needs to be considered in the future control development for convenience.

6.7 Simulation Results

Numerical simulations were performed to illustrate the performance of the controller given in (6–33) and (6–36). The intrinsic camera calibration matrix is

given by

$$A = \begin{bmatrix} 122.5 & -3.77 & 100 \\ 0 & 122.56 & 100 \\ 0 & 0 & 1 \end{bmatrix}.$$

The best-guess estimation for A was selected as

$$\hat{A} = \begin{bmatrix} 100 & -4 & 80 \\ 0 & 100 & 110 \\ 0 & 0 & 1 \end{bmatrix}.$$

The camera is assumed to view an object with four coplanar feature points with the following Euclidean coordinates (in [m]):

$$\begin{aligned} O_1 &= \begin{bmatrix} 0.05 & 0.05 & 0 \end{bmatrix}^T & O_2 &= \begin{bmatrix} 0.05 & -0.05 & 0 \end{bmatrix}^T \\ O_3 &= \begin{bmatrix} -0.05 & 0.05 & 0 \end{bmatrix}^T & O_4 &= \begin{bmatrix} -0.05 & -0.05 & 0 \end{bmatrix}^T. \end{aligned} \quad (6-53)$$

The normalized coordinates of the vanishing points were selected as

$$\begin{aligned} &\begin{bmatrix} 0.02 & 0.02 & 1 \end{bmatrix}^T & & \begin{bmatrix} 0.02 & -0.02 & 1 \end{bmatrix}^T \\ &\begin{bmatrix} -0.02 & 0.02 & 1 \end{bmatrix}^T & & \begin{bmatrix} -0.02 & -0.02 & 1 \end{bmatrix}^T. \end{aligned}$$

Consider an orthogonal coordinate frame \mathcal{I} with the z -axis opposite to n^* (see Figure 2-2) with the x -axis and y -axis on the plane π . The rotation matrices R_1 between \mathcal{F} and \mathcal{I} , and R_2 between \mathcal{F}^* and \mathcal{I} were set as

$$R_1 = R_x(160^\circ)R_y(30^\circ)R_z(-30^\circ) \quad R_2 = R_x(120^\circ)R_y(-20^\circ)R_z(80^\circ),$$

where $R_x(\cdot)$, $R_y(\cdot)$ and $R_z(\cdot) \in SO(3)$ denote rotation of angle “ \cdot ” (degrees) along the x -axis, y -axis and z -axis, respectively. The translation vectors $x_{f_1}(t)$ and $x_{f_2}(t)$ between \mathcal{F} and \mathcal{I} (expressed in \mathcal{F}) and between \mathcal{F}^* and \mathcal{I} (expressed in \mathcal{F}^*),

respectively, were selected as

$$x_{f1} = \begin{bmatrix} 0.5 & 0.5 & 2.5 \end{bmatrix}^T \quad x_{f2} = \begin{bmatrix} 1.0 & 1.0 & 3.5 \end{bmatrix}^T .$$

The initial (i.e., $p_i(0)$) and desired (i.e., p_i^*) image-space coordinates of the four feature points in (6–53) were computed as (in pixels)

$$\begin{aligned} p_1(0) &= \begin{bmatrix} 126.50 & 123.64 & 1 \end{bmatrix}^T & p_2(0) &= \begin{bmatrix} 124.24 & 127.91 & 1 \end{bmatrix}^T \\ p_3(0) &= \begin{bmatrix} 120.92 & 125.40 & 1 \end{bmatrix}^T & p_4(0) &= \begin{bmatrix} 123.25 & 121.11 & 1 \end{bmatrix}^T \\ p_1^* &= \begin{bmatrix} 132.17 & 133.17 & 1 \end{bmatrix}^T & p_2^* &= \begin{bmatrix} 135.72 & 133.61 & 1 \end{bmatrix}^T \\ p_3^* &= \begin{bmatrix} 135.71 & 136.91 & 1 \end{bmatrix}^T & p_4^* &= \begin{bmatrix} 132.10 & 136.44 & 1 \end{bmatrix}^T . \end{aligned}$$

The initial (i.e., $p_{vi}(0)$) and desired (i.e., p_{vi}^*) image-space coordinates of the four vanishing points in (6–53) were computed as (in pixels)

$$\begin{aligned} p_{v1}(0) &= \begin{bmatrix} 124.02 & 139.34 & 1 \end{bmatrix}^T & p_{v2}(0) &= \begin{bmatrix} 129.02 & 141.61 & 1 \end{bmatrix}^T \\ p_{v3}(0) &= \begin{bmatrix} 131.02 & 136.54 & 1 \end{bmatrix}^T & p_{v4}(0) &= \begin{bmatrix} 126.03 & 134.35 & 1 \end{bmatrix}^T \\ p_{v1}^* &= \begin{bmatrix} 102.37 & 102.45 & 1 \end{bmatrix}^T & p_{v2}^* &= \begin{bmatrix} 102.53 & 97.55 & 1 \end{bmatrix}^T \\ p_{v3}^* &= \begin{bmatrix} 97.63 & 97.55 & 1 \end{bmatrix}^T & p_{v4}^* &= \begin{bmatrix} 97.47 & 102.45 & 1 \end{bmatrix}^T . \end{aligned}$$

The control gains K_ω in (6–33) and K_v in (6–36) were selected as

$$K_\omega = 5 \quad K_v = 5.$$

The resulting translation and rotation errors are plotted in Figure 6–1 and Figure 6–2, respectively. The image-space pixel error (i.e., $p_i(t) - p_i^*$) is shown in Figure 6–4, and is also depicted in Figure 6–5 in a 3D format. The translation and rotation control outputs are shown in Figure 6–6 and Figure 6–7, respectively. For

different choice of sign of the quaternion estimate, the asymptotic result can still be achieved. In contrast to the quaternion estimate in Figure 6–2, a quaternion estimate with different sign is shown in Figure 6–3.

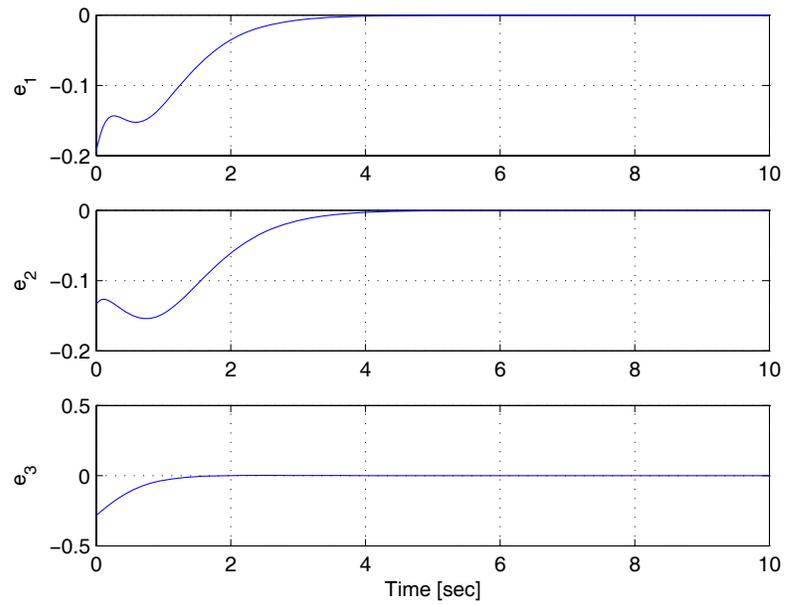


Figure 6–1: Unitless translation error between $m_1(t)$ and m_1^* .

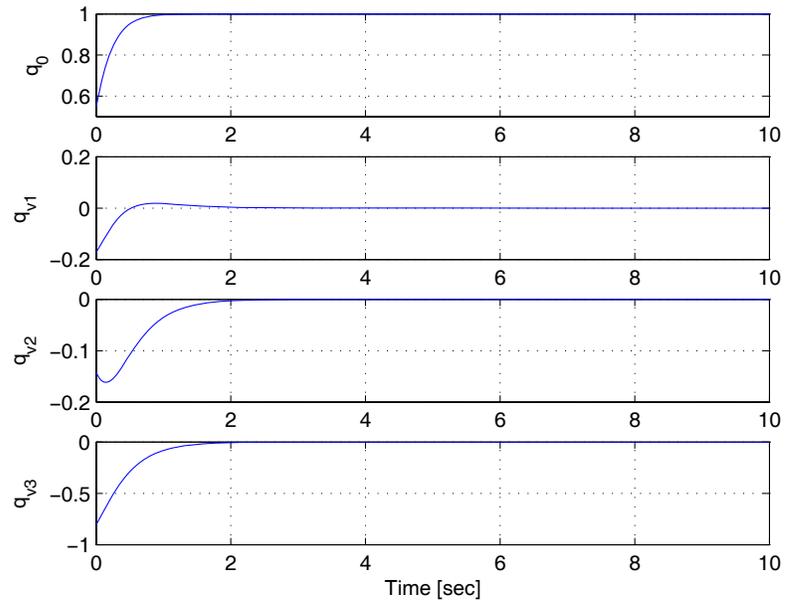


Figure 6-2: Quaternion rotation error.

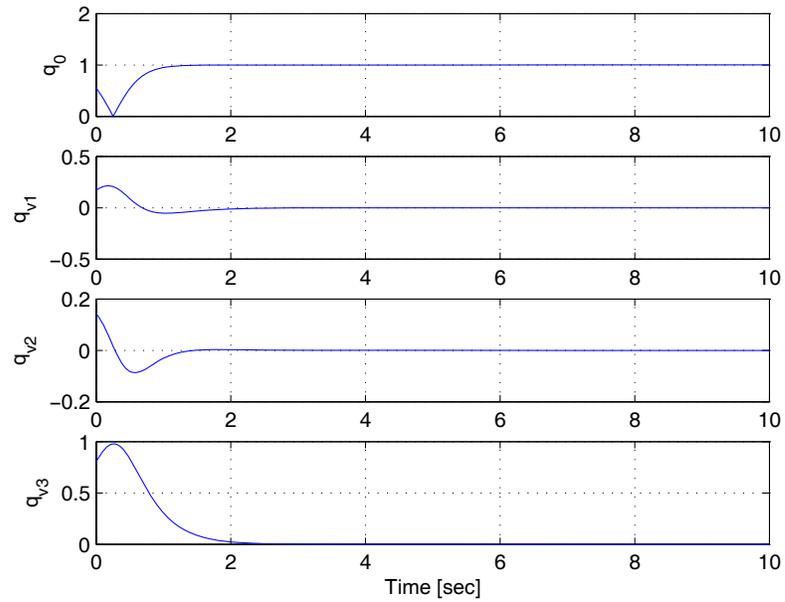


Figure 6-3: Quaternion rotation error for comparison with different sign.

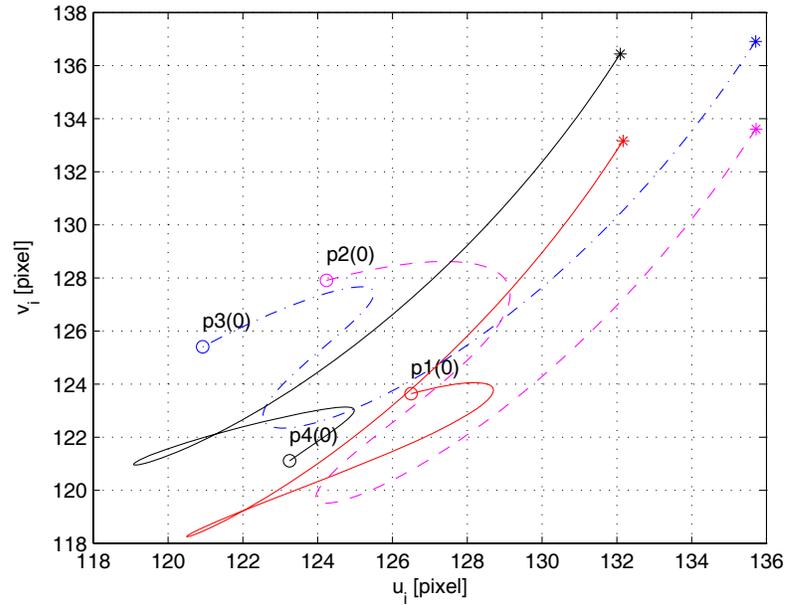


Figure 6–4: Image-space error in pixles between $p_i(t)$ and p_i^* . In the figure, “O” denotes the initial positions of the 4 feature points in the image, and “*” denotes the corresponding final positions of the feature points.

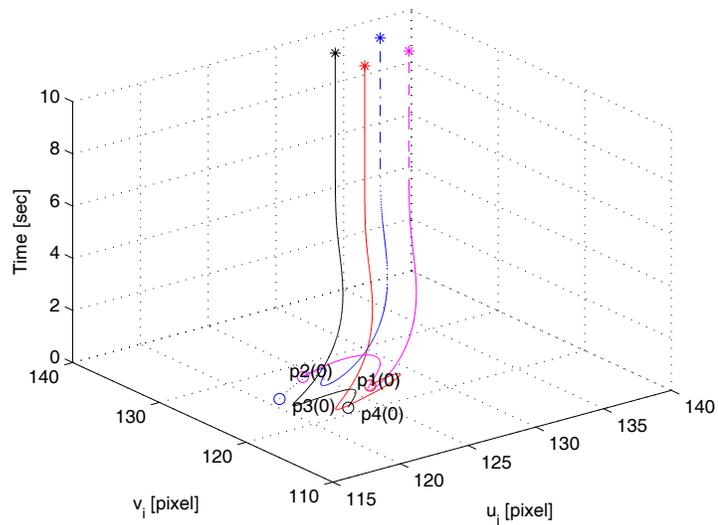


Figure 6–5: Image-space error in pixles between $p_i(t)$ and p_i^* shown in a 3D graph. In the figure, “O” denotes the initial positions of the 4 feature points in the image, and “*” denotes the corresponding final positions of the feature points.

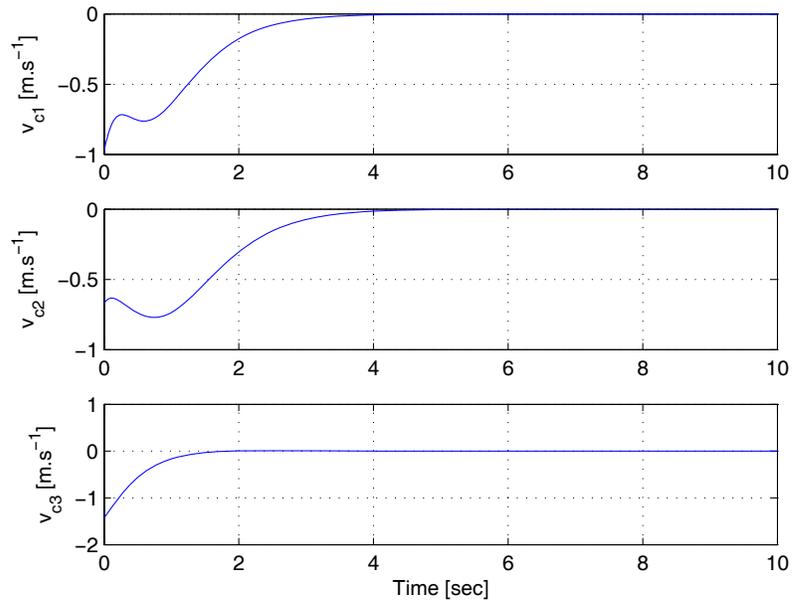


Figure 6-6: Linear camera velocity control input.

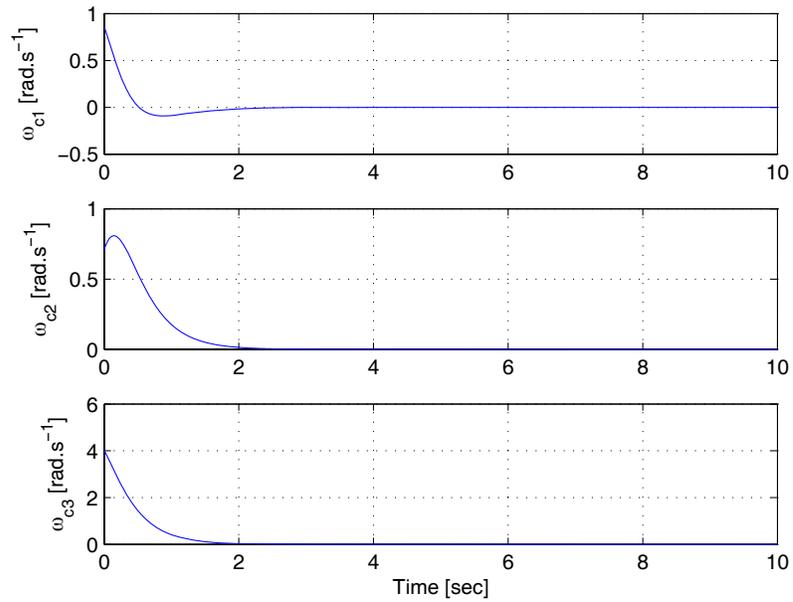


Figure 6-7: Angular camera velocity control input.

CHAPTER 7
COMBINED ROBUST AND ADAPTIVE HOMOGRAPHY-BASED VISUAL
SERVO CONTROL VIA AN UNCALIBRATED CAMERA

7.1 Introduction

In this chapter, a new combined robust and adaptive visual servo controller is developed to asymptotically regulate the feature points of a rigid-body object (identified a planar patch of feature points) in an image to the desired feature point locations while also regulating the six DOF pose of the camera (which is affixed to the object). These dual objectives are achieved by using a homography-based approach that exploits both image-space and reconstructed Euclidean information in the feedback loop. In comparison to pure image-based feedback approaches, some advantages of using a homography-based method include: realizable Euclidean camera trajectories (see Chaumette [48] and Corke and Hutchinson [25] for a discussion of Chaumette's Conundrum); a nonsingular image-Jacobian; and both the camera position and orientation and the feature point coordinates are included in the error system. Since some image-space information is used in the feedback-loop of the developed homography-based controller, the image features are less likely to leave the FOV in comparison with pure position-based approaches. The developed controller is composed of the same adaptive translation controller as in the preliminary results in Chen et al. [89] and a new robust rotation controller. The contribution of the result is the development of the robust angular velocity controller that accommodates for the time-varying uncertain scaling factor by exploiting the upper triangular form of the rotation error system and the fact that the diagonal elements of the camera calibration matrix are positive.

7.2 Camera Geometry and Assumptions

The camera geometry for this chapter is shown in Figure 2–2 and the corresponding Euclidean and image-space relationships are developed in Sections 2.1 and 2.2. For convenience in the following development, the camera calibration matrix A is rewritten as

$$A \triangleq \begin{bmatrix} \alpha & -\alpha \cot \phi & u_0 \\ 0 & \frac{\beta}{\sin \phi} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}. \quad (7-1)$$

Based on the physical meaning of the elements of A , the diagonal calibration elements are positive (i.e., $a_{11}, a_{22} > 0$).

The following two assumptions are made for the convenience of the controller development. They are so reasonable such that they can be considered properties of the considered vision system.

Assumption 1: The bounds of a_{11} and a_{22} are assumed to be known as

$$\underline{\zeta}_{a_{11}} < a_{11} < \bar{\zeta}_{a_{11}} \quad \underline{\zeta}_{a_{22}} < a_{22} < \bar{\zeta}_{a_{22}}. \quad (7-2)$$

The absolute values of a_{12}, a_{13}, a_{23} are upper bounded as

$$|a_{12}| < \bar{\zeta}_{a_{12}} \quad |a_{13}| < \bar{\zeta}_{a_{13}} \quad |a_{23}| < \bar{\zeta}_{a_{23}}. \quad (7-3)$$

In (7–2) and (7–3), $\underline{\zeta}_{a_{11}}, \bar{\zeta}_{a_{11}}, \underline{\zeta}_{a_{22}}, \bar{\zeta}_{a_{22}}, \bar{\zeta}_{a_{12}}, \bar{\zeta}_{a_{13}}$ and $\bar{\zeta}_{a_{23}}$ are known positive constants.

Assumption 2: The reference plane is within the camera's FOV and not at infinity. That is, there exist positive constants $\underline{\zeta}_{z_i}$ and $\bar{\zeta}_{z_i}$ such that

$$\underline{\zeta}_{z_i} < z_i(t) < \bar{\zeta}_{z_i}. \quad (7-4)$$

Based on (2-4)-(2-6), the homography relationship based on measurable pixel coordinates is:

$$p_i = \alpha_i A H A^{-1} p_i^*. \quad (7-5)$$

Since A is unknown, standard homography computation and decomposition algorithms can't be applied to extract the rotation and translation from the homography.

As stated in Malis and Chaumette [21], if some additional information is known, such as four vanishing points, the rotation matrix can be obtained. For the vanishing points (see Almansa et al. [103] for a description of how to determine vanishing points in an image), $d^* = \infty$, so that

$$H = R + \frac{x_f}{d^*} n^{*T} = R. \quad (7-6)$$

Based on (7-6), the relationship in (7-5) can be expressed as

$$p_i = \alpha_i \bar{R} p_i^*, \quad (7-7)$$

where $\bar{R}(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$\bar{R} = A R A^{-1}. \quad (7-8)$$

For the four vanishing points, twelve linear equations can be obtained based on (7-7). After normalizing $\bar{R}(t)$ by one nonzero element (e.g., $\bar{R}_{33}(t) \in \mathbb{R}$ which is assumed to be the third row third column element of $\bar{R}(t)$ without loss of generality) twelve equations can be used to solve for twelve unknowns. The twelve unknowns are given by the eight unknown elements of the normalized $\bar{R}(t)$, denoted by $\bar{R}_n(t) \in \mathbb{R}^{3 \times 3}$ defined as

$$\bar{R}_n \triangleq \frac{\bar{R}}{\bar{R}_{33}} \quad (7-9)$$

and the four unknowns are given by $\bar{R}_{33}(t)\alpha_i(t)$. From the definition of $\bar{R}_n(t)$ in (7-9), the fact that

$$\det(\bar{R}) = \det(A) \det(R) \det(A^{-1}) = 1 \quad (7-10)$$

can be used to conclude that

$$\bar{R}_{33}^3 \det(\bar{R}_n) = 1, \quad (7-11)$$

and hence based on (7-9) and (7-11),

$$\bar{R} = \frac{\bar{R}_n}{\sqrt[3]{\det(\bar{R}_n)}}. \quad (7-12)$$

After $\bar{R}(t)$ is obtained, the original four feature points on the reference plane can be used to determine the depth ratio $\alpha_i(t)$ as shown in Appendix E.

7.3 Open-Loop Error System

7.3.1 Rotation Error System

If the rotation matrix $R(t)$ introduced in (2-4) were known, then the corresponding unit quaternion $q(t) \triangleq \begin{bmatrix} q_0(t) & q_v^T(t) \end{bmatrix}^T$ can be calculated using the numerically robust method presented in Section 2.3. Given $R(t)$, the quaternion $q(t)$ can also be written as

$$q_0 = \frac{1}{2} \sqrt{1 + \text{tr}(R)} \quad (7-13)$$

$$q_v = \frac{1}{2} u \sqrt{3 - \text{tr}(R)}, \quad (7-14)$$

where $u(t) \in \mathbb{R}^3$ is a unit eigenvector of $R(t)$ with respect to the eigenvalue 1. The open-loop rotation error system for $q(t)$ can be obtained as (see Dixon et al. [23])

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_v^T \\ q_0 I_3 + q_v^\times \end{bmatrix} \omega_c, \quad (7-15)$$

where $\omega_c(t) \in \mathbb{R}^3$ defines the angular velocity of the camera expressed in \mathcal{F} .

The quaternion $q(t)$ given in (7-13)-(7-15) is not measurable since $R(t)$ is unknown. However, since $\bar{R}(t)$ can be determined as described in (7-12), the same algorithm as shown in equations (7-13) and (7-14) can be used to determine a corresponding measurable quaternion $(\bar{q}_0(t), \bar{q}_v^T(t))^T$ as

$$\bar{q}_0 = \frac{1}{2}\sqrt{1 + tr(\bar{R})} \quad (7-16)$$

$$\bar{q}_v = \frac{1}{2}\bar{u}\sqrt{3 - tr(\bar{R})}, \quad (7-17)$$

where $\bar{u}(t) \in \mathbb{R}^3$ is a unit eigenvector of $\bar{R}(t)$ with respect to the eigenvalue 1. Based on (7-8), $tr(\bar{R}) = tr(ARA^{-1}) = tr(R)$, where $tr(\cdot)$ denotes the trace of a matrix. Since $R(t)$ and $\bar{R}(t)$ are similar matrices, the relationship between $(q_0(t), q_v^T(t))^T$ and $(\bar{q}_0(t), \bar{q}_v^T(t))^T$ can be determined as

$$\bar{q}_0 = q_0 \quad \bar{q}_v = \frac{\|q_v\|}{\|Aq_v\|}Aq_v \triangleq \gamma Aq_v, \quad (7-18)$$

where $\gamma(t) \in \mathbb{R}$ is a positive, unknown, time-varying scalar that satisfies the following inequalities (see Appendix F)

$$\underline{\zeta}_\gamma < \gamma(t) < \bar{\zeta}_\gamma, \quad (7-19)$$

where $\underline{\zeta}_\gamma, \bar{\zeta}_\gamma \in \mathbb{R}$ are positive bounding constants. The inverse of the relationship between $\bar{q}_v(t)$ and $q_v(t)$ in (7-18) can be developed as

$$q_v = \frac{1}{\gamma}A^{-1}\bar{q}_v = \frac{1}{\gamma} \begin{bmatrix} \frac{1}{a_{11}}\bar{q}_{v1} - \frac{a_{12}}{a_{11}a_{22}}\bar{q}_{v2} - \left(\frac{a_{13}}{a_{11}} - \frac{a_{12}a_{23}}{a_{11}a_{22}}\right)\bar{q}_{v3} \\ \frac{1}{a_{22}}\bar{q}_{v2} - \frac{a_{23}}{a_{22}}\bar{q}_{v3} \\ \bar{q}_{v3} \end{bmatrix}. \quad (7-20)$$

7.3.2 Translation Error System

The translation error, denoted by $e(t) \in \mathbb{R}^3$, is defined as

$$e(t) = p_e(t) - p_e^*, \quad (7-21)$$

where $p_e(t), p_e^* \in \mathbb{R}^3$ are defined as

$$p_e = \begin{bmatrix} u_i & v_i & -\ln(\alpha_i) \end{bmatrix}^T \quad p_e^* = \begin{bmatrix} u_i^* & v_i^* & 0 \end{bmatrix}^T, \quad (7-22)$$

where $i \in \{1, \dots, 4\}$. The translation error $e(t)$ is measurable since the first two elements are image coordinates, and $\alpha_i(t)$ is obtained from the homography decomposition as described in Appendix A. The open-loop translation error system can be obtained by taking the time derivative of $e(t)$ and multiplying the resulting expression by z_i^* as [89]

$$z_i^* \dot{e} = -\alpha_i A_e v_c + z_i^* A_e (A^{-1} p_i)^\times \omega_c, \quad (7-23)$$

where $v_c(t) \in \mathbb{R}^3$ defines the linear velocity of the camera expressed in \mathcal{F} , and $A_e(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$A_e = \begin{bmatrix} a_{11} & a_{12} & a_{13} - u_i \\ 0 & a_{22} & a_{23} - v_i \\ 0 & 0 & 1 \end{bmatrix}.$$

To facilitate the control development, the translation error system can be linearly parameterized as

$$z_i^* \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = -\alpha_i \begin{bmatrix} a_{11}v_{c1} + a_{12}v_{c2} + v_{c3}(a_{13} - u_i) \\ a_{22}v_{c2} + v_{c3}(a_{23} - v_i) \\ v_{c3} \end{bmatrix} + z_i^* \begin{bmatrix} \bar{Y}_1(u_i, v_i, \omega_c) \bar{\phi} \\ \bar{Y}_2(u_i, v_i, \omega_c) \bar{\phi} \\ \bar{Y}_3(u_i, v_i, \omega_c) \bar{\phi} \end{bmatrix}, \quad (7-24)$$

where $\bar{Y}_i(\cdot) \in \mathbb{R}^{1 \times m}, i = 1, 2, 3$, are known regressor vectors that do not depend on the calibration parameters, and $\bar{\phi} \in \mathbb{R}^m$ is a vector of constant unknown parameters.

7.4 Control Development

7.4.1 Rotation Control Development and Stability Analysis

Based on the relationship in (7-18), the open-loop error system in (7-15), and the subsequent stability analysis, the rotation controller is designed as

$$\omega_{c1} = -k_{\omega 1} \bar{q}_{v1} = -(k_{\omega 11} + 2) \bar{q}_{v1} \quad (7-25)$$

$$\omega_{c2} = -k_{\omega 2} \bar{q}_{v2} = -(k_{\omega 21} + k_{\omega 22} + 1) \bar{q}_{v2}$$

$$\omega_{c3} = -k_{\omega 3} \bar{q}_{v3} = -(k_{\omega 31} + k_{\omega 32} + k_{\omega 33}) \bar{q}_{v3},$$

where $k_{\omega i} \in \mathbb{R}, i = 1, 2, 3$ and $k_{\omega ij} \in \mathbb{R}, i, j = 1, 2, 3, j \leq i$, are positive constants. The expressed form of the controller in (7-25) is motivated by the use of completing the squares in the subsequent stability analysis. In (7-25), the damping control gains $k_{\omega 21}, k_{\omega 31}, k_{\omega 32}$ are selected according to the following sufficient conditions to facilitate the subsequent stability analysis

$$\begin{aligned} k_{\omega 21} &> \frac{1}{4} k_{\omega 1}^2 \frac{\bar{\zeta}_{a12}^2}{\underline{\zeta}_{a11} \underline{\zeta}_{a22}} \\ k_{\omega 31} &> \frac{1}{4} k_{\omega 1}^2 \frac{1}{\underline{\zeta}_{a11}} \left(\frac{\bar{\zeta}_{a12} \bar{\zeta}_{a23}}{\underline{\zeta}_{a22}} + \bar{\zeta}_{a13} \right)^2 \\ k_{\omega 32} &> \frac{1}{4} k_{\omega 2}^2 \frac{\bar{\zeta}_{a23}^2}{\underline{\zeta}_{a22}}, \end{aligned} \quad (7-26)$$

where $\underline{\zeta}_{a11}, \bar{\zeta}_{a11}, \underline{\zeta}_{a22}, \bar{\zeta}_{a22}, \bar{\zeta}_{a12}, \bar{\zeta}_{a13}$ and $\bar{\zeta}_{a23}$ are defined in (7-2) and (7-3), and $k_{\omega 11}, k_{\omega 22}, k_{\omega 33}$ are feedback gains that can be selected to adjust the performance of the rotation control system.

Proposition 7.1: Provided the sufficient gain conditions given in (7-26) are satisfied, the controller in (7-25) ensures asymptotic regulation of the rotation error in the sense that

$$\|q_v(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty. \quad (7-27)$$

Proof: Let $V_1(q_v, q_0) \in \mathbb{R}$ denote the following non-negative function:

$$V_1 \triangleq q_v^T q_v + (1 - q_0)^2. \quad (7-28)$$

Based on the open-loop error system in (7-15), the time-derivative of $V_1(t)$ can be determined as

$$\dot{V}_1 = 2q_v^T \dot{q}_v - 2(1 - q_0)\dot{q}_0 = q_v^T \omega_c = q_{v1}\omega_{c1} + q_{v2}\omega_{c2} + q_{v3}\omega_{c3}. \quad (7-29)$$

After substituting (7-20) for $q_v(t)$ and substituting (7-25) for $\omega_c(t)$, the expression in (7-29) can be simplified as

$$\begin{aligned} \gamma \dot{V}_1 = & - \left(k_{\omega 11} \frac{1}{a_{11}} \bar{q}_{v1}^2 + k_{\omega 22} \frac{1}{a_{22}} \bar{q}_{v2}^2 + k_{\omega 33} \bar{q}_{v3}^2 \right) - \frac{1}{a_{11}} \left[\bar{q}_{v1}^2 - k_{\omega 1} \frac{a_{12}}{a_{22}} \bar{q}_{v1} \bar{q}_{v2} + k_{\omega 21} \frac{a_{11}}{a_{22}} \bar{q}_{v2}^2 \right] \\ & - \frac{1}{a_{11}} \left[\bar{q}_{v1}^2 + k_{\omega 1} \left(\frac{a_{12} a_{23}}{a_{22}} - a_{13} \right) \bar{q}_{v1} \bar{q}_{v3} + k_{\omega 31} a_{11} \bar{q}_{v3}^2 \right] \\ & - \frac{1}{a_{22}} \left[\bar{q}_{v2}^2 - k_{\omega 2} a_{23} \bar{q}_{v2} \bar{q}_{v3} + k_{\omega 32} a_{22} \bar{q}_{v3}^2 \right]. \end{aligned} \quad (7-30)$$

After completing the squares on each of the bracketed terms in (7-30), the expression in (7-30) can be written as

$$\begin{aligned} \gamma \dot{V}_1 = & - \left(k_{\omega 11} \frac{1}{a_{11}} \bar{q}_{v1}^2 + k_{\omega 22} \frac{1}{a_{22}} \bar{q}_{v2}^2 + k_{\omega 33} \bar{q}_{v3}^2 \right) \\ & - \frac{1}{a_{11}} \left[\left(\bar{q}_{v1} - k_{\omega 1} \frac{a_{12}}{2a_{22}} \bar{q}_{v2} \right)^2 + \frac{a_{11}}{a_{22}} \left(k_{\omega 21} - \frac{1}{4} k_{\omega 1}^2 \frac{a_{12}^2}{a_{11} a_{22}} \right) \bar{q}_{v2}^2 \right] \\ & - \frac{1}{a_{11}} \left[\left(\bar{q}_{v1} + \frac{1}{2} k_{\omega 1} \left(\frac{a_{12} a_{23}}{a_{22}} - a_{13} \right) \bar{q}_{v3} \right)^2 \right. \\ & \left. + a_{11} \left(k_{\omega 31} - \frac{1}{4} k_{\omega 1}^2 \frac{1}{a_{11}} \left(\frac{a_{12} a_{23}}{a_{22}} - a_{13} \right)^2 \right) \bar{q}_{v3}^2 \right] \\ & - \frac{1}{a_{22}} \left[\left(\bar{q}_{v2} - \frac{1}{2} k_{\omega 2} a_{23} \bar{q}_{v3} \right)^2 + a_{22} \left(k_{\omega 32} - \frac{1}{4} k_{\omega 2}^2 \frac{a_{23}^2}{a_{22}} \right) \bar{q}_{v3}^2 \right]. \end{aligned} \quad (7-31)$$

Provided the sufficient gain conditions given in (7-26) are satisfied, then (7-31) can be upper bounded as

$$\gamma \dot{V}_1 < - \left(k_{\omega 11} \frac{1}{a_{11}} \bar{q}_{v1}^2 + k_{\omega 22} \frac{1}{a_{22}} \bar{q}_{v2}^2 + k_{\omega 33} \bar{q}_{v3}^2 \right). \quad (7-32)$$

Based on (7-19), the inequality in (7-32) can be further upper bounded as

$$\dot{V}_1 < - \frac{1}{\zeta_\gamma} \left(k_{\omega 11} \frac{1}{a_{11}} \bar{q}_{v1}^2 + k_{\omega 22} \frac{1}{a_{22}} \bar{q}_{v2}^2 + k_{\omega 33} \bar{q}_{v3}^2 \right). \quad (7-33)$$

The Lyapunov function given in (7-28) and its time derivative in (7-33) can be used to conclude that $q_v(t), q_0(t) \in \mathcal{L}_\infty$ and $\bar{q}_v(t) \in \mathcal{L}_2$ (of course, $q_v(t), q_0(t) \in \mathcal{L}_\infty$ by definition also). The expressions in (7-18) and (7-20), and the fact that $\bar{q}_v(t) \in \mathcal{L}_2$, can be used to conclude that $q_v(t) \in \mathcal{L}_2$. Since $q_v(t), q_0(t) \in \mathcal{L}_\infty$, then $R(t), \bar{R}(t), \bar{q}_v(t)$ and $\bar{q}_0(t) \in \mathcal{L}_\infty$. Hence, (7-25) can be used to conclude that $\omega_c(t) \in \mathcal{L}_\infty$. Based on the rotation error system in (7-15), $\dot{q}_v(t), \dot{q}_0(t) \in \mathcal{L}_\infty$; hence, $q_v(t), q_0(t)$ are uniformly continuous. Barbalat's lemma [96] can now be used to conclude that $\|q_v(t)\| \rightarrow 0$ as $t \rightarrow \infty$.

7.4.2 Translation Control Development and Stability Analysis

For completeness of the result, the same translation controller as in Chen et al. [89] is provided. After some algebraic manipulation, the translation error system in (7-24) can be rewritten as

$$\begin{aligned} \frac{z_i^*}{a_{11}} \dot{e}_1 &= -\alpha_i v_{c1} + Y_1(\alpha_i, u_i, v_i, \omega_c, v_{c2}, v_{c3}) \phi_1 \\ \frac{z_i^*}{a_{22}} \dot{e}_2 &= -\alpha_i v_{c2} + Y_2(\alpha_i, u_i, v_i, \omega_c, v_{c3}) \phi_2 \\ z_i^* \dot{e}_3 &= -\alpha_i v_{c3} + Y_3(\alpha_i, u_i, v_i, \omega_c) \phi_3, \end{aligned} \quad (7-34)$$

where $\phi_1 \in \mathbb{R}^{n_1}$, $\phi_2 \in \mathbb{R}^{n_2}$, and $\phi_3 \in \mathbb{R}^{n_3}$ are vectors of constant unknown parameters, and the known regressor vectors $Y_1(\cdot) \in \mathbb{R}^{1 \times n_1}$, $Y_2(\cdot) \in \mathbb{R}^{1 \times n_2}$, and

$Y_3(\cdot) \in \mathbb{R}^{1 \times n_3}$ satisfy the following equations (see Appendix G):

$$\begin{aligned} Y_1 \phi_1 &= -\alpha_i \frac{a_{12}}{a_{11}} v_{c2} - \alpha_i \frac{(a_{13} - u_i)}{a_{11}} v_{c3} + z_i^* \bar{Y}_1(u_i, v_i, \omega_c) \frac{\bar{\phi}}{a_{11}} \\ Y_2 \phi_2 &= -\alpha_i \frac{a_{23} - v_i}{a_{22}} v_{c3} + z_i^* \bar{Y}_2(u_i, v_i, \omega_c) \frac{\bar{\phi}}{a_{22}} \\ Y_3 \phi_3 &= z_i^* \bar{Y}_3(u_i, v_i, \omega_c) \bar{\phi}. \end{aligned}$$

The control strategy is to design $v_{c3}(t)$ to stabilize $e_3(t)$, and then design $v_{c2}(t)$ to stabilize $e_2(t)$ given $v_{c3}(t)$, and then design $v_{c1}(t)$ to stabilize $e_1(t)$ given $v_{c3}(t)$ and $v_{c2}(t)$. Following this design strategy, the translation controller $v_c(t)$ is designed as [89]

$$\begin{aligned} v_{c3} &= \frac{1}{\alpha_i} \left(k_{v3} e_3 + Y_3(\alpha_i, u_i, v_i, \omega_c) \hat{\phi}_3 \right) \\ v_{c2} &= \frac{1}{\alpha_i} \left(k_{v2} e_2 + Y_2(\alpha_i, u_i, v_i, \omega_c, v_{c3}) \hat{\phi}_2 \right) \\ v_{c1} &= \frac{1}{\alpha_i} \left(k_{v1} e_1 + Y_1(\alpha_i, u_i, v_i, \omega_c, v_{c2}, v_{c3}) \hat{\phi}_1 \right), \end{aligned} \quad (7-35)$$

where the depth ratio $\alpha_i(t) > 0 \forall t$. In (7-35), $\hat{\phi}_1(t) \in \mathbb{R}^{n_1}$, $\hat{\phi}_2(t) \in \mathbb{R}^{n_2}$, $\hat{\phi}_3(t) \in \mathbb{R}^{n_3}$ denote adaptive estimates that are designed according to the following adaptive update laws to cancel the respective terms in the subsequent stability analysis

$$\dot{\hat{\phi}}_1 = \Gamma_1 Y_1^T e_1 \quad \dot{\hat{\phi}}_2 = \Gamma_2 Y_2^T e_2 \quad \dot{\hat{\phi}}_3 = \Gamma_3 Y_3^T e_3, \quad (7-36)$$

where $\Gamma_1 \in \mathbb{R}^{n_1 \times n_1}$, $\Gamma_2 \in \mathbb{R}^{n_2 \times n_2}$, $\Gamma_3 \in \mathbb{R}^{n_3 \times n_3}$ are diagonal matrices of positive constant adaptation gains. Based on (7-34) and (7-35), the closed-loop translation error system is

$$\begin{aligned} \frac{z_i^*}{a_{11}} \dot{e}_1 &= -k_{v1} e_1 + Y_1(\alpha_i, u_i, v_i, \omega_c, v_{c2}, v_{c3}) \tilde{\phi}_1 \\ \frac{z_i^*}{a_{22}} \dot{e}_2 &= -k_{v2} e_2 + Y_2(\alpha_i, u_i, v_i, \omega_c, v_{c3}) \tilde{\phi}_2 \\ z_i^* \dot{e}_3 &= -k_{v3} e_3 + Y_3(\alpha_i, u_i, v_i, \omega_c) \tilde{\phi}_3, \end{aligned} \quad (7-37)$$

where $\tilde{\phi}_1(t) \in \mathbb{R}^{n_1}$, $\tilde{\phi}_2(t) \in \mathbb{R}^{n_2}$, $\tilde{\phi}_3(t) \in \mathbb{R}^{n_3}$ denote the intrinsic calibration parameter mismatch defined as

$$\tilde{\phi}_1(t) = \phi_1 - \hat{\phi}_1(t) \quad \tilde{\phi}_2(t) = \phi_2 - \hat{\phi}_2(t) \quad \tilde{\phi}_3(t) = \phi_3 - \hat{\phi}_3(t).$$

Proposition 7.2: The controller given in (7–35) along with the adaptive update law in (7–36) ensures asymptotic regulation of the translation error system in the sense that

$$\|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty.$$

Proof: Let $V_2(e, \tilde{\phi}_1, \tilde{\phi}_2, \tilde{\phi}_3) \in \mathbb{R}$ denote the following non-negative function:

$$V_2 = \frac{1}{2} \frac{z_i^*}{a_{11}} e_1^2 + \frac{1}{2} \frac{z_i^*}{a_{22}} e_2^2 + \frac{1}{2} z_i^* e_3^2 + \frac{1}{2} \tilde{\phi}_1^T \Gamma_1^{-1} \tilde{\phi}_1 + \frac{1}{2} \tilde{\phi}_2^T \Gamma_2^{-1} \tilde{\phi}_2 + \frac{1}{2} \tilde{\phi}_3^T \Gamma_3^{-1} \tilde{\phi}_3. \quad (7-38)$$

After taking the time derivative of (7–38) and substituting for the closed-loop error system developed in (7–37), the following simplified expression can be obtained:

$$\dot{V}_2 = -k_{v1} e_1^2 - k_{v2} e_2^2 - k_{v3} e_3^2. \quad (7-39)$$

Based on (7–38) and (7–39), $e_1(t), e_2(t), e_3(t) \in \mathcal{L}_\infty \cap \mathcal{L}_2$, and $\tilde{\phi}_i(t), \hat{\phi}_i(t), i = 1, 2, 3 \in \mathcal{L}_\infty$. Based on the assumption that $\underline{\zeta}_{z_i} < z_i(t) < \bar{\zeta}_{z_i}$, the expression in (7–35) can be used to conclude that $v_c(t) \in \mathcal{L}_\infty$. Based on the previous stability analysis for the rotation controller, $\omega_c(t) \in \mathcal{L}_\infty$; hence, (7–37) can be used to conclude that $\dot{e}_1(t), \dot{e}_2(t), \dot{e}_3(t) \in \mathcal{L}_\infty$ (i.e., $e_1(t), e_2(t), e_3(t)$ are uniformly continuous). Barbalat's lemma [96] can now be used to show that $e_1(t), e_2(t), e_3(t) \rightarrow 0$ as $t \rightarrow \infty$. ■

Based on Propositions 7.1 and 7.2, the main result can be stated as follows.

Theorem 7.1: The controller given in (7–25) and (7–35) along with the adaptive update law in (7–36) ensures asymptotic translation and rotation regulation in the sense that

$$\|q_v(t)\| \rightarrow 0 \quad \text{and} \quad \|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty,$$

provided the control gains satisfy the sufficient conditions given in (7-26).

Proof: See proofs in Propositions 7.1 and 7.2.

7.5 Simulation Results

Numerical simulations were performed to illustrate the performance of the controller given in (7-25) and (7-35) and the adaptive law given in (7-36). The intrinsic camera calibration matrix is given by

$$A = \begin{bmatrix} 122.5 & -3.77 & 100 \\ 0 & 122.56 & 100 \\ 0 & 0 & 1 \end{bmatrix}.$$

The camera is assumed to view an object with four coplanar feature points with the following Euclidean coordinates (in [m]):

$$\begin{aligned} O_1 &= \begin{bmatrix} 0.1 & 0.1 & 0 \end{bmatrix}^T & O_2 &= \begin{bmatrix} 0.1 & -0.1 & 0 \end{bmatrix}^T \\ O_3 &= \begin{bmatrix} -0.1 & 0.1 & 0 \end{bmatrix}^T & O_4 &= \begin{bmatrix} -0.1 & -0.1 & 0 \end{bmatrix}^T. \end{aligned} \quad (7-40)$$

The normalized coordinates of the vanishing points were selected as

$$\begin{aligned} &\begin{bmatrix} 0.01 & 0.01 & 1 \end{bmatrix}^T & & \begin{bmatrix} 0.01 & -0.01 & 1 \end{bmatrix}^T \\ &\begin{bmatrix} -0.01 & 0.01 & 1 \end{bmatrix}^T & & \begin{bmatrix} -0.01 & -0.01 & 1 \end{bmatrix}^T. \end{aligned}$$

The initial (i.e., $p_i(0)$) and desired (i.e., p_i^*) image-space coordinates of the four feature points in (7–40) were computed as (in pixels)

$$\begin{aligned} p_1(0) &= \begin{bmatrix} 128.49 & 129.41 & 1 \end{bmatrix}^T & p_2(0) &= \begin{bmatrix} 128.80 & 119.61 & 1 \end{bmatrix}^T \\ p_3(0) &= \begin{bmatrix} 119.00 & 119.61 & 1 \end{bmatrix}^T & p_4(0) &= \begin{bmatrix} 118.70 & 129.41 & 1 \end{bmatrix}^T \\ p_1^* &= \begin{bmatrix} 136.86 & 138.68 & 1 \end{bmatrix}^T & p_2^* &= \begin{bmatrix} 138.07 & 132.17 & 1 \end{bmatrix}^T \\ p_3^* &= \begin{bmatrix} 130.97 & 131.33 & 1 \end{bmatrix}^T & p_4^* &= \begin{bmatrix} 129.84 & 137.82 & 1 \end{bmatrix}^T . \end{aligned}$$

The initial (i.e., $p_{vi}(0)$) and desired (i.e., p_{vi}^*) image-space coordinates of the four vanishing points in (7–40) were computed as (in pixels)

$$\begin{aligned} p_{v1}(0) &= \begin{bmatrix} 91.84 & 123.68 & 1 \end{bmatrix}^T & p_{v2}(0) &= \begin{bmatrix} 91.70 & 121.16 & 1 \end{bmatrix}^T \\ p_{v3}(0) &= \begin{bmatrix} 89.20 & 121.41 & 1 \end{bmatrix}^T & p_{v4}(0) &= \begin{bmatrix} 89.34 & 123.94 & 1 \end{bmatrix}^T \\ p_{v1}^* &= \begin{bmatrix} 101.19 & 101.23 & 1 \end{bmatrix}^T & p_{v2}^* &= \begin{bmatrix} 101.26 & 98.77 & 1 \end{bmatrix}^T \\ p_{v3}^* &= \begin{bmatrix} 98.81 & 98.77 & 1 \end{bmatrix}^T & p_{v4}^* &= \begin{bmatrix} 98.74 & 101.23 & 1 \end{bmatrix}^T . \end{aligned}$$

The actual value of the unknown parameter vectors ϕ_1 , ϕ_2 , and ϕ_3 are given by

$$\phi_1 = \begin{bmatrix} 0.0082 & -0.0308 & 0.8163 & 3.5110 & -0.1082 & 2.3386 & 0.0234 \\ 0.0234 & 0.0002 & 2.8648 & 0.0286 & 0.0002 & 0 & -0.0241 \\ 0.0234 & -0.0007 & -2.4117 & -0.0009 & 0 & 0.0910 \end{bmatrix}^T$$

$$\phi_2 = \begin{bmatrix} 0.0082 & 0.8159 & 3.5110 & 2.3375 & 0.0234 & 0.0002 & 0.0002 & 0 & -0.0241 \\ 0.0287 & -0.0009 & -2.9544 & 0.0234 & -0.0007 & -2.4106 & & & \end{bmatrix}^T$$

$$\phi_3 = \begin{bmatrix} 2.8648 & 0.0286 & 0.0287 & -0.0009 & -2.9544 \end{bmatrix}^T.$$

In the simulation, the initial value of the estimated parameter vectors $\hat{\phi}_1(0)$, $\hat{\phi}_2(0)$, and $\hat{\phi}_3(0)$ were selected as half of the actual value.

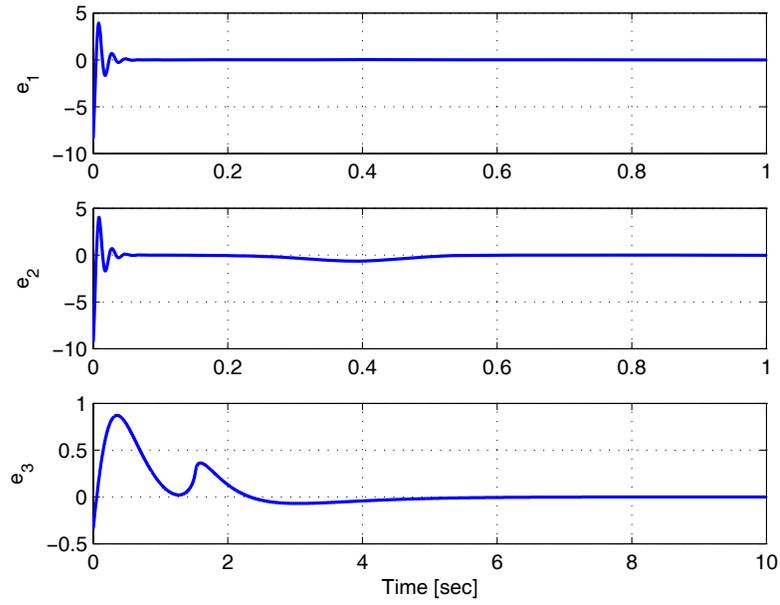
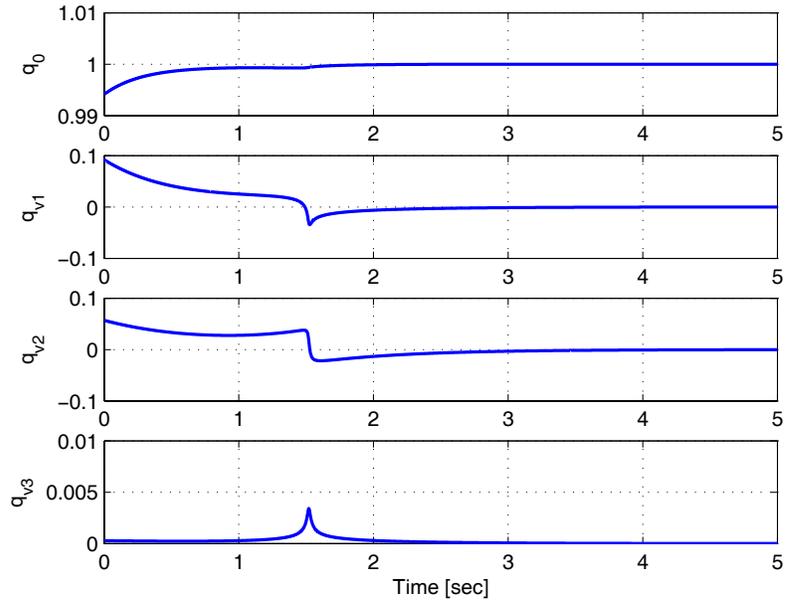
The diagonal control gain matrices K_ω in (7-25) and K_v in (7-35) were selected as

$$K_\omega = \text{diag}\{4, 2, 85\} \quad K_v = \text{diag}\{5, 5, 5\}.$$

The diagonal gain matrices Γ_1 , Γ_2 , and Γ_3 in the adaptive law (7-36) were selected as

$$\begin{aligned} \Gamma_1 &= 0.03 \times \text{diag}\{0.01, 1, 1, 1, 1, 1, 0.01, 0.01, 0.0001, 1, \\ &\quad 0.01, 0.0001, 0.0001, 0.01, 0.01, 0.01, 1, 0.01, 0.01, 0.01, 1\} \\ \Gamma_2 &= 0.03 \times \text{diag}\{0.01, 1, 1, 1, 0.01, 0.0001, 0.01, 0.01, 1, 0.0001, \\ &\quad 0.0001, 0.01, 0.01, 0.01, 1\} \\ \Gamma_3 &= 0.5 \times \text{diag}\{0.1, 0.1, 10, 0.1, 10\}. \end{aligned}$$

The resulting asymptotic translation and rotation errors are plotted in Figure 7-1 and Figure 7-2, respectively. The pixel coordinate of the four feature points is shown in Figure 7-3. The image-space pixel error (i.e., $p_i(t) - p_i^*$) is shown in Figure 7-4. The image-space trajectory of the feature points is shown in Figure 7-5, and also in Figure 7-6 in a three-dimensional format, where the vertical axis is time. The translation and rotation control outputs are shown in Figure 7-7 and Figure 7-8, respectively.

Figure 7-1: Unitless translation error $e(t)$.Figure 7-2: Quaternion rotation error $q(t)$.

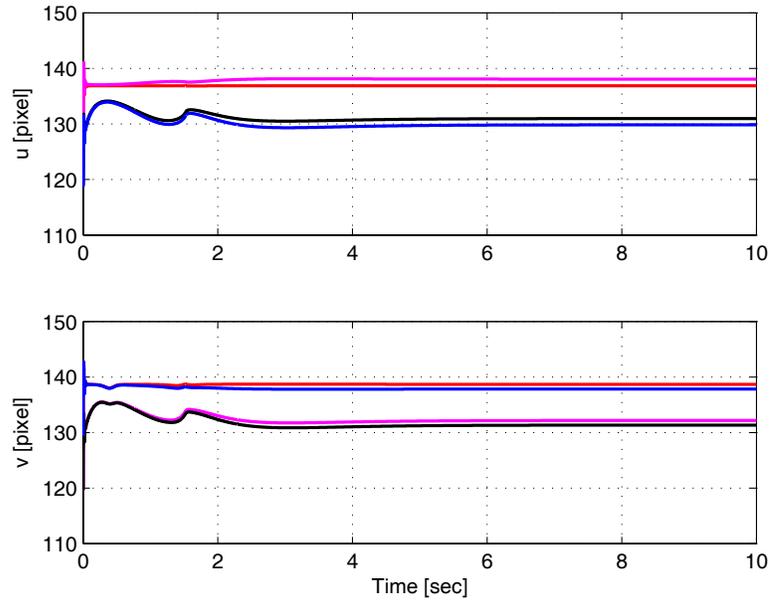


Figure 7–3: Pixel coordinate $p(t)$ (in pixels) of the current pose of the four feature points in the simulation. The upper figure is for the $u(t)$ component and the bottom figure is for the $v(t)$ component.

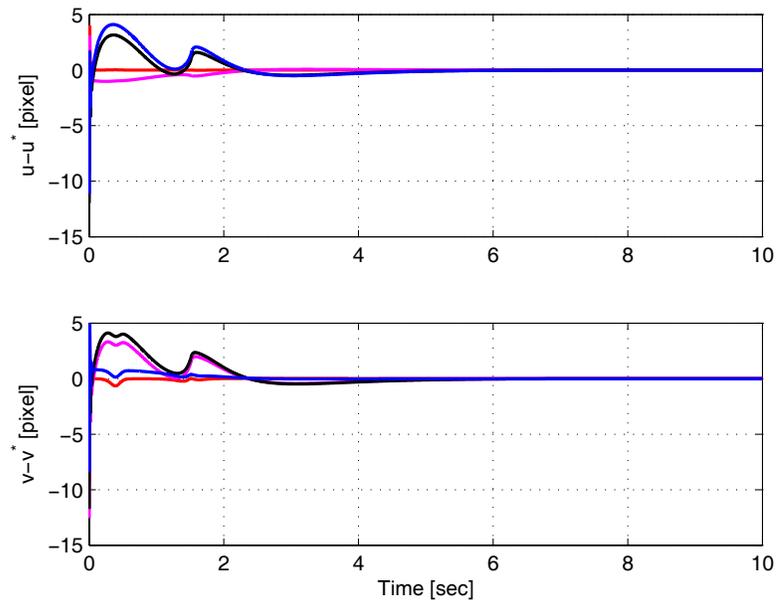


Figure 7–4: Regulation error $p(t) - p^*$ (in pixels) of the four feature points in the simulation. The upper figure is for the $u(t) - u^*(t)$ component and the bottom figure is for the $v(t) - v^*(t)$ component.

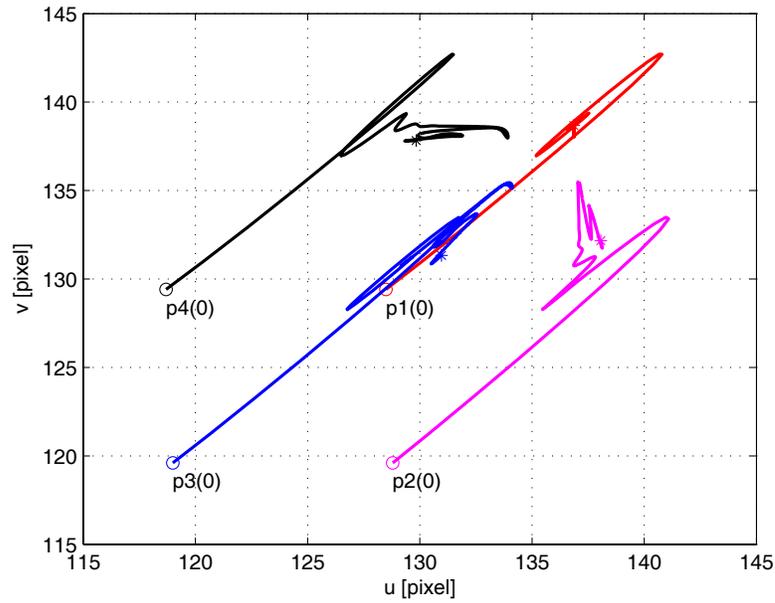


Figure 7–5: Image-space error in pixels between $p(t)$ and p^* . In the figure, “O” denotes the initial positions of the 4 feature points in the image, and “*” denotes the corresponding final positions of the feature points.

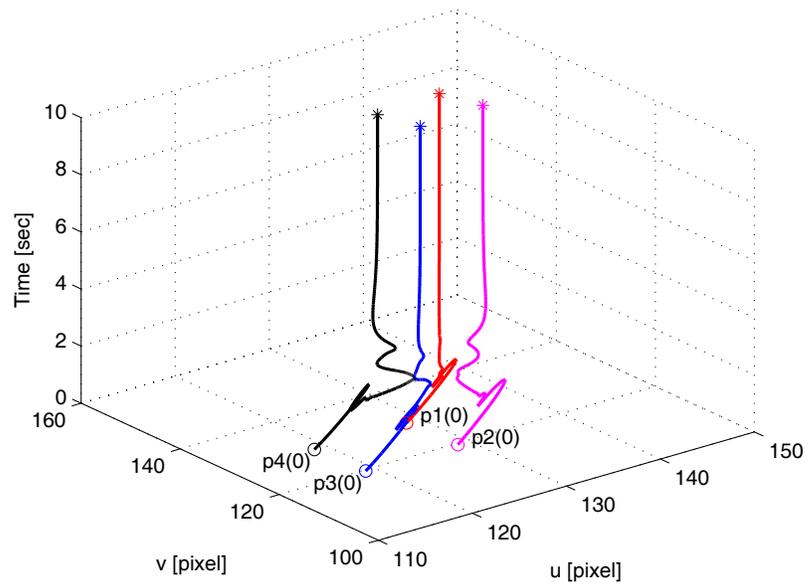
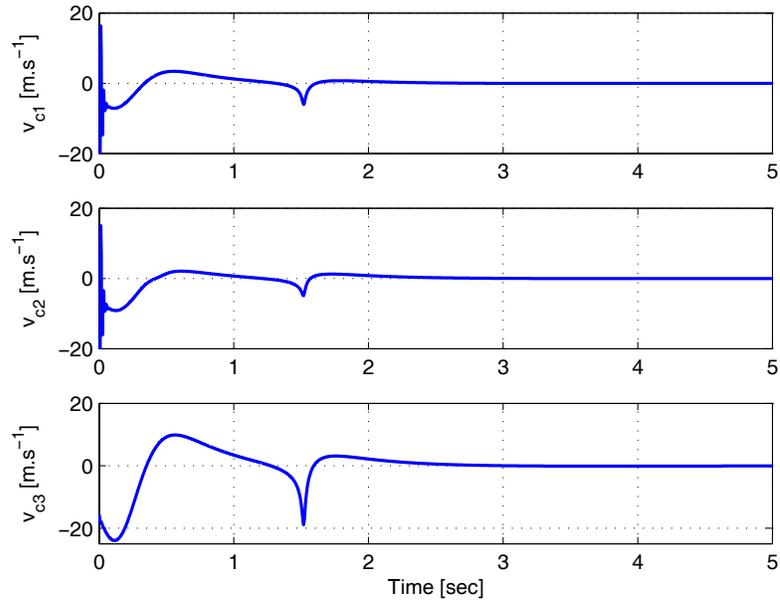
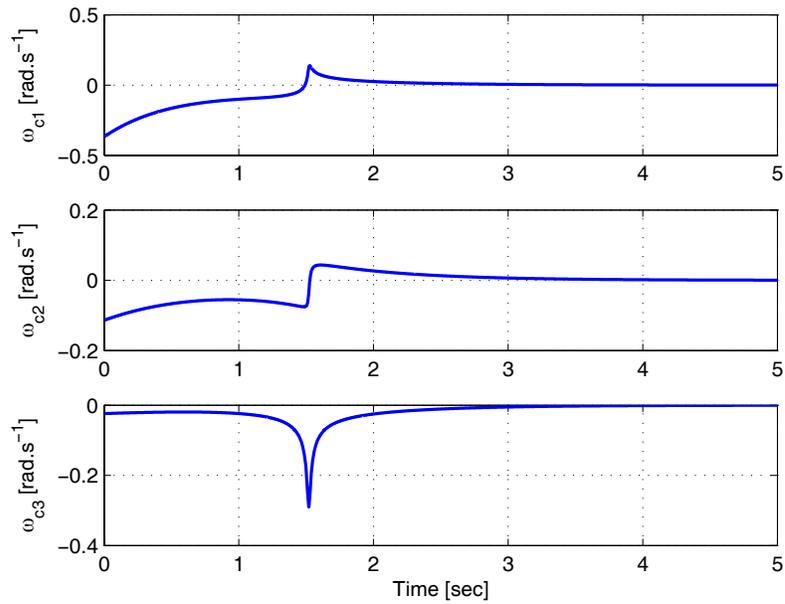


Figure 7–6: Image-space error in pixels between $p(t)$ and p^* shown in a 3D graph. In the figure, “O” denotes the initial positions of the 4 feature points in the image, and “*” denotes the corresponding final positions of the feature points.

Figure 7-7: Linear camera velocity control input $v_c(t)$.Figure 7-8: Angular camera velocity control input $\omega_c(t)$.

CHAPTER 8 CONCLUSIONS

In this dissertation, visual servo control algorithms and architectures are developed that exploit the visual feedback from a camera system to achieve a tracking or regulation control objective for a rigid-body object (e.g., the end-effector of a robot manipulator, a satellite, an autonomous vehicle) identified by a patch of feature points. These algorithms and architectures can be used widely in the navigation and control applications in robotics and autonomous systems. The visual servo control problem in this dissertation were separated into five parts: 1) visual servo tracking control via a quaternion formulation; 2) collaborative visual servo tracking control using a daisy-chaining approach; 3) visual servo tracking control using a central catadioptric camera; 4) robust visual servo control in presence of camera calibration uncertainty; and 5) combined robust and adaptive visual servo control via an uncalibrated camera.

An adaptive visual servo tracking control method via a quaternion formulation is first developed that achieves asymptotic tracking of a rigid-body object to a desired trajectory determined by a sequence of images. By developing the error systems and controllers based on a homography decomposition, the singularity associated with the typical image-Jacobian is eliminated. By utilizing the quaternion formulation, a singularity-free error system is obtained. A homography-based rotation and translation controller is proven to yield the tracking result through a Lyapunov-based stability analysis. Based on the result for the camera-in-hand configuration problem, a camera-to-hand extension is given to enable a rigid-body object to track a desired trajectory. Simulation and experiments results are provided to show the performance of the proposed visual servo controllers.

In order to enable large area motion and weaken the FOV restriction, a collaborative visual servo method is then developed to enable the control object to track a desired trajectory. In contrast to typical camera-to-hand and camera-in-hand visual servo control configurations, the proposed controller is developed using a moving on-board camera viewing a moving object to obtain feedback signals. A daisy-chaining method is used to develop homography relationship between different camera coordinate frames and feature point patch coordinate frames to facilitate collaboration between different agents (e.g., cameras, reference objects, and control object). Lyapunov-based methods are used to prove the asymptotic tracking result. Simulation results are provided to show the performance of the proposed visual servo controller. To enlarge the FOV, an alternative visual servo controller is developed that yields an asymptotic tracking result using a central catadioptric camera. A panoramic FOV is obtained by using the central catadioptric camera.

To study the robust visual servo control problem, a visual servo controller is developed to regulate a camera (attached to a rigid-body object) to a desired pose in presence of intrinsic camera calibration uncertainties. A quaternion-based estimate for the rotation error system is developed that is related to the actual rotation error. The similarity relationship between the estimated and actual rotation matrices is used to construct the relationship between the estimated and actual quaternions. A Lyapunov-based stability analysis is provided that indicates a unique controller can be developed to achieve the regulation result despite a sign ambiguity in the developed quaternion estimate. Simulation results are provided to illustrate the performance of the developed controller.

A new combined robust and adaptive visual servo controller is then developed to asymptotically regulate the feature points in an image to the desired feature point locations while also regulating the six DOF pose of the camera. These dual

objectives are achieved by using a homography-based approach that exploits both image-space and reconstructed Euclidean information in the feedback loop. In comparison to pure image-based feedback approaches, some advantages of using a homography-based method include: realizable Euclidean camera trajectories; a nonsingular image-Jacobian; and both the camera pose and the feature point coordinates are included in the error system. Since some image-space information is used in the feedback-loop of the developed homography-based controller, the image features are less likely to leave the FOV in comparison with pure position-based approaches. The robust rotation controller that accommodates for the time-varying uncertain scaling factor is developed by exploiting the upper triangular form of the rotation error system and the fact that the diagonal elements of the camera calibration matrix are positive. The adaptive translation controller that compensates for the constant unknown parameters in the translation error system is developed by a certainty-equivalence-based adaptive control method and a nonlinear Lyapunov-based design approach. Simulation results are provided to show the performance of the proposed visual servo controller.

APPENDIX A
UNIT NORM PROPERTY FOR THE QUATERNION ERROR

Property: The quaternion error $(\tilde{q}_0(t), \tilde{q}_v^T(t))^T$ defined in (3–5) has a unit norm given that $q(t)$ and $q_d(t)$ are two unit quaternions.

Proof: The quaternion components in (3–5) can be expanded as

$$\tilde{q}_0 = q_0 q_{0d} + \begin{bmatrix} q_{v1} \\ q_{v2} \\ q_{v3} \end{bmatrix}^T \begin{bmatrix} q_{vd1} \\ q_{vd2} \\ q_{vd3} \end{bmatrix} = q_{v1}q_{vd1} + q_{v2}q_{vd2} + q_{v3}q_{vd3} + q_0q_{0d} \quad (\text{A-1})$$

and

$$\begin{aligned} \tilde{q}_v &= q_0 \begin{bmatrix} q_{vd1} \\ q_{vd2} \\ q_{vd3} \end{bmatrix} - q_{0d} \begin{bmatrix} q_{v1} \\ q_{v2} \\ q_{v3} \end{bmatrix} + \begin{bmatrix} 0 & -q_{v3} & q_{v2} \\ q_{v3} & 0 & -q_{v1} \\ -q_{v2} & q_{v1} & 0 \end{bmatrix} \begin{bmatrix} q_{vd1} \\ q_{vd2} \\ q_{vd3} \end{bmatrix} \\ &= \begin{bmatrix} q_0q_{vd1} + q_{v2}q_{vd3} - q_{v3}q_{vd2} - q_{v1}q_{0d} \\ q_0q_{vd2} - q_{v1}q_{vd3} + q_{v3}q_{vd1} - q_{v2}q_{0d} \\ q_0q_{vd3} + q_{v1}q_{vd2} - q_{v2}q_{vd1} - q_{v3}q_{0d} \end{bmatrix}. \end{aligned} \quad (\text{A-2})$$

Based on (A–1) and (A–2),

$$\begin{aligned} &\tilde{q}_0^2 + \tilde{q}_v^T \tilde{q}_v \\ &= (q_{v1}q_{vd1} + q_{v2}q_{vd2} + q_{v3}q_{vd3} + q_0q_{0d})^2 \\ &+ \begin{bmatrix} q_0q_{vd1} + q_{v2}q_{vd3} - q_{v3}q_{vd2} - q_{v1}q_{0d} \\ q_0q_{vd2} - q_{v1}q_{vd3} + q_{v3}q_{vd1} - q_{v2}q_{0d} \\ q_0q_{vd3} + q_{v1}q_{vd2} - q_{v2}q_{vd1} - q_{v3}q_{0d} \end{bmatrix}^T \begin{bmatrix} q_0q_{vd1} + q_{v2}q_{vd3} - q_{v3}q_{vd2} - q_{v1}q_{0d} \\ q_0q_{vd2} - q_{v1}q_{vd3} + q_{v3}q_{vd1} - q_{v2}q_{0d} \\ q_0q_{vd3} + q_{v1}q_{vd2} - q_{v2}q_{vd1} - q_{v3}q_{0d} \end{bmatrix}. \end{aligned}$$

Expanding the right side of the above equation gives

$$\begin{aligned}
& \tilde{q}_0^2 + \tilde{q}_v^T \tilde{q}_v \\
&= (q_{v1}q_{vd1} + q_{v2}q_{vd2} + q_{v3}q_{vd3} + q_0q_{0d})^2 + (q_0q_{vd1} + q_{v2}q_{vd3} - q_{v3}q_{vd2} - q_{v1}q_{0d})^2 \\
&+ (q_0q_{vd2} - q_{v1}q_{vd3} + q_{v3}q_{vd1} - q_{v2}q_{0d})^2 + (q_0q_{vd3} + q_{v1}q_{vd2} - q_{v2}q_{vd1} - q_{v3}q_{0d})^2 \\
&= q_0^2q_{vd1}^2 + q_0^2q_{vd2}^2 + q_{v1}^2q_{vd1}^2 + q_0^2q_{vd3}^2 + q_{v1}^2q_{vd2}^2 + q_{v2}^2q_{vd1}^2 + q_{v1}^2q_{vd3}^2 + q_{v2}^2q_{vd2}^2 \\
&+ q_{v3}^2q_{vd1}^2 + q_{v2}^2q_{vd3}^2 + q_{v3}^2q_{vd2}^2 + q_{v3}^2q_{vd3}^2 + q_0^2q_{0d}^2 + q_{v1}^2q_{0d}^2 + q_{v2}^2q_{0d}^2 + q_{v3}^2q_{0d}^2 \\
&= (q_0^2 + q_{v1}^2 + q_{v2}^2 + q_{v3}^2)(q_{0d}^2 + q_{vd1}^2 + q_{vd2}^2 + q_{vd3}^2).
\end{aligned}$$

Based on the fact that $q(t)$ and $q_d(t)$ are two unit quaternions (i.e., their norms are equal to 1),

$$\tilde{q}_0^2 + \tilde{q}_v^T \tilde{q}_v = 1.$$

APPENDIX B
ONE PROPERTY OF UNIT QUATERNIONS

Property: $(I_3 + q_v^\times)^{-1}q_v = q_v$.

Proof: The term $I_3 - q_v^\times$ can be expanded as

$$I_3 - q_v^\times = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & -q_{v3} & q_{v2} \\ q_{v3} & 0 & -q_{v1} \\ -q_{v2} & q_{v1} & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_{v3} & -q_{v2} \\ -q_{v3} & 1 & q_{v1} \\ q_{v2} & -q_{v1} & 1 \end{bmatrix}. \quad (\text{B-1})$$

Taking the inverse of the expanded matrix in (B-1) gives

$$\begin{aligned} (I_3 - q_v^\times)^{-1} &= \begin{bmatrix} 1 & q_{v3} & -q_{v2} \\ -q_{v3} & 1 & q_{v1} \\ q_{v2} & -q_{v1} & 1 \end{bmatrix}^{-1} \\ &= \frac{1}{q_{v1}^2 + q_{v2}^2 + q_{v3}^2 + 1} \begin{bmatrix} q_{v1}^2 + 1 & -q_{v3} + q_{v1}q_{v2} & q_{v2} + q_{v1}q_{v3} \\ q_{v3} + q_{v1}q_{v2} & q_{v2}^2 + 1 & -q_{v1} + q_{v2}q_{v3} \\ -q_{v2} + q_{v1}q_{v3} & q_{v1} + q_{v2}q_{v3} & q_{v3}^2 + 1 \end{bmatrix}. \end{aligned} \quad (\text{B-2})$$

Multiplying q_v on both sides of (B-2) gives

$$\begin{aligned} (I_3 - q_v^\times)^{-1} q_v &= \frac{\begin{bmatrix} q_{v3} (q_{v2} + q_{v1}q_{v3}) + q_{v2} (-q_{v3} + q_{v1}q_{v2}) + q_{v1} (q_{v1}^2 + 1) \\ q_{v1} (q_{v3} + q_{v1}q_{v2}) + q_{v3} (-q_{v1} + q_{v2}q_{v3}) + q_{v2} (q_{v2}^2 + 1) \\ q_{v2} (q_{v1} + q_{v2}q_{v3}) + q_{v1} (-q_{v2} + q_{v1}q_{v3}) + q_{v3} (q_{v3}^2 + 1) \end{bmatrix}}{q_{v1}^2 + q_{v2}^2 + q_{v3}^2 + 1} \\ &= q_v. \end{aligned}$$

APPENDIX C
OPEN-LOOP TRANSLATION ERROR SYSTEM

The translation error was defined as

$$e = m_e - m_{ed},$$

where $m_e(t)$ and $m_{ed}(t)$ are the extended normalized coordinates of $\bar{m}'_i(t)$ and $\bar{m}_{rdi}(t)$, respectively, and were defined as

$$m_e = \begin{bmatrix} \frac{x'_i}{z'_i} & \frac{y'_i}{z'_i} & \ln\left(\frac{z'_i}{z_{ri}^*}\right) \end{bmatrix}^T \quad m_{ed} = \begin{bmatrix} \frac{x_{rdi}}{z_{rdi}} & \frac{y_{rdi}}{z_{rdi}} & \ln\left(\frac{z_{rdi}}{z_{ri}^*}\right) \end{bmatrix}^T.$$

Differentiating $m_e(t)$ gives

$$\dot{m}_e = \frac{1}{z'_i} L'_v \dot{\bar{m}}'_i, \quad (\text{C-1})$$

where

$$L'_v = \begin{bmatrix} 1 & 0 & -\frac{x'_i}{z'_i} \\ 0 & 1 & -\frac{y'_i}{z'_i} \\ 0 & 0 & 1 \end{bmatrix}.$$

The derivative of $\bar{m}'_i(t)$ is given by

$$\dot{\bar{m}}'_i = \begin{bmatrix} \dot{x}'_i \\ \dot{y}'_i \\ \dot{z}'_i \end{bmatrix} = R' (v_c + \omega_c^\times s_i), \quad (\text{C-2})$$

where $v_c(t)$ and $\omega_c(t)$ are the linear and angular velocities of π with respect to \mathcal{I}_R expressed in \mathcal{F} . Based on (C-1) and (C-2), $\dot{m}_e(t)$ can be further written as

$$\dot{m}_e = \frac{1}{z'_i} L'_v R' (v_c + \omega_c^\times s_i). \quad (\text{C-3})$$

From (C-3), the translation error system can be obtained as

$$\dot{e} = \frac{1}{z_i'} L_v' R' (v_c + \omega_c^\times s_i) - \dot{m}_{ed},$$

and it can be further written as

$$z_{ri}^* \dot{e} = \frac{z_{ri}^*}{z_i'} L_v' R' (v_c + \omega_c^\times s_i) - z_{ri}^* \dot{m}_{ed}.$$

APPENDIX D
PROPERTY ON MATRIX NORM

Property: $\|[\xi]^\times\|_2 = \|\xi\| \quad \forall \xi \in \mathbb{R}^3$.

Proof: The spectral norm $\|A\|_2$ of a real matrix is the square root of the largest eigenvalue of the matrix multiplied by its transpose, i.e.,

$$\|A\|_2 = \sqrt{\lambda_{\max}\{A^T A\}}.$$

For any given vector $\xi \in \mathbb{R}^3$,

$$[\xi]^\times = \begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix}.$$

The Euclidean norm of $[\xi]^\times$ is given by

$$\begin{aligned} \|[\xi]^\times\|_2 &= \sqrt{\lambda_{\max}\{[[\xi]^\times]^T [\xi]^\times\}} \\ &= \sqrt{\lambda_{\max}\left\{\begin{bmatrix} \xi_2^2 + \xi_3^2 & -\xi_1\xi_2 & -\xi_1\xi_3 \\ -\xi_1\xi_2 & \xi_1^2 + \xi_3^2 & -\xi_2\xi_3 \\ -\xi_1\xi_3 & -\xi_2\xi_3 & \xi_1^2 + \xi_2^2 \end{bmatrix}\right\}} \\ &= \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2}. \end{aligned}$$

The norm of ξ is given by

$$\|\xi\| = \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2}.$$

Hence, $\|[\xi]^\times\|_2 = \|\xi\|$.

APPENDIX E
COMPUTATION OF DEPTH RATIOS

Based on (7-5) and (7-6), the following expression can be obtained:

$$p_i = \alpha_i (\bar{R} + \bar{x}_h \bar{n}^{*T}) p_i^*, \quad (\text{E-1})$$

where $\bar{R}(t)$ is defined in (7-8), and $\bar{x}_h(t) \in \mathbb{R}^3$ and $\bar{n}^*(t) \in \mathbb{R}^3$ are defined as

$$\bar{x}_h = A \frac{x_f}{d^*} \quad \bar{n}^{*T} = n^{*T} A^{-1}.$$

Using the four corresponding reference feature points, the expression in (E-1) can be written as

$$p_i = \alpha_i G p_i^* = \alpha_i g_{33} G_n p_i^*, \quad (\text{E-2})$$

where $g_{33}(t) \in \mathbb{R}$ is the (assumed w.l.o.g.) positive third row third column element of $G(t)$, and $G_n(t) \in \mathbb{R}^{3 \times 3}$ is defined as $G(t)/g_{33}(t)$. Based on (E-2), twelve linear equations can be obtained for the four corresponding reference feature points. A set of twelve linear equations can then be developed to solve for $\alpha_i(t)g_{33}(t)$ and $G_n(t)$.

To determine the scalar $g_{33}(t)$, the following equation can be used:

$$g_{33} G_n = \bar{R} + \bar{x}_h \bar{n}^{*T}, \quad (\text{E-3})$$

provided that $\bar{R}(t)$ is obtained using the four vanishing points, where

$$\bar{R} = \begin{bmatrix} \bar{r}_{11} & \bar{r}_{12} & \bar{r}_{13} \\ \bar{r}_{21} & \bar{r}_{22} & \bar{r}_{23} \\ \bar{r}_{31} & \bar{r}_{32} & \bar{r}_{33} \end{bmatrix} \quad G_n = \begin{bmatrix} g_{n11} & g_{n12} & g_{n13} \\ g_{n21} & g_{n22} & g_{n23} \\ g_{n31} & g_{n32} & 1 \end{bmatrix}$$

$$\bar{x}_h = \begin{bmatrix} \bar{x}_{h1} & \bar{x}_{h2} & \bar{x}_{h3} \end{bmatrix}^T \quad \bar{n}^{*T} = \begin{bmatrix} \bar{n}_1^* & \bar{n}_2^* & \bar{n}_3^* \end{bmatrix}.$$

To this end, let

$$\begin{aligned}
 x_1 &= \bar{x}_{h1}\bar{n}_1^* & x_2 &= \bar{x}_{h1}\bar{n}_2^* & x_3 &= \bar{x}_{h1}\bar{n}_3^* \\
 x_4 &= \bar{x}_{h2}\bar{n}_1^* & x_5 &= \bar{x}_{h2}\bar{n}_2^* & x_6 &= \bar{x}_{h2}\bar{n}_3^* \\
 x_7 &= \bar{x}_{h3}\bar{n}_1^* & x_8 &= \bar{x}_{h3}\bar{n}_2^* & x_9 &= \bar{x}_{h3}\bar{n}_3^* \\
 \mu_1 &= \frac{\bar{n}_2^*}{\bar{n}_1^*} & \mu_2 &= \frac{\bar{n}_3^*}{\bar{n}_1^*},
 \end{aligned}$$

then the following relationships can be obtained:

$$\begin{aligned}
 x_2 &= \mu_1 x_1 & x_5 &= \mu_1 x_4 & x_8 &= \mu_1 x_7 \\
 x_3 &= \mu_2 x_1 & x_6 &= \mu_2 x_4 & x_9 &= \mu_2 x_7.
 \end{aligned}$$

The elements of the matrix $g_{33}(t)G_n(t)$ in (E-3) can be rewritten as

$$\bar{r}_{11} + x_1 = g_{33}g_{n11} \quad \bar{r}_{12} + \mu_1 x_1 = g_{33}g_{n12} \quad (\text{E-4})$$

$$\bar{r}_{13} + \mu_2 x_1 = g_{33}g_{n13}$$

$$\bar{r}_{21} + x_4 = g_{33}g_{n21} \quad \bar{r}_{22} + \mu_1 x_4 = g_{33}g_{n22} \quad (\text{E-5})$$

$$\bar{r}_{23} + \mu_2 x_4 = g_{33}g_{n23}$$

$$\bar{r}_{31} + x_7 = g_{33}g_{n31} \quad \bar{r}_{32} + \mu_1 x_7 = g_{33}g_{n32} \quad (\text{E-6})$$

$$\bar{r}_{33} + \mu_2 x_7 = g_{33}.$$

The expressions in (E-4) can be combined as

$$\bar{r}_{12} + \mu_1 x_1 = \frac{g_{n12}}{g_{n11}} (\bar{r}_{11} + x_1) \quad (\text{E-7})$$

$$\bar{r}_{13} + \mu_2 x_1 = \frac{g_{n13}}{g_{n11}} (\bar{r}_{11} + x_1).$$

Similarly, the expressions in (E-5) can be combined as

$$\begin{aligned}\bar{r}_{22} + \mu_1 x_4 &= \frac{g_{n22}}{g_{n21}} (\bar{r}_{21} + x_4) \\ \bar{r}_{23} + \mu_2 x_4 &= \frac{g_{n23}}{g_{n21}} (\bar{r}_{21} + x_4).\end{aligned}\tag{E-8}$$

Since $\bar{r}_{ij}(t)$, $g_{nij}(t) \forall i, j \in \{1, 2\}$ and $\bar{r}_{13}(t)$, $\bar{r}_{23}(t)$, $g_{n13}(t)$ and $g_{n23}(t)$ are known, (E-7) and (E-8) provide four equations that can be solved to determine the four unknowns (i.e., μ_1 , μ_2 , $x_1(t)$, $x_4(t)$). After solving for these four unknowns, (E-4) and (E-5) can be used to solve for $g_{33}(t)$, and then $G(t)$ can be obtained. Given $G(t)$, (E-2) can be used to solve for $\alpha_i(t)$.

APPENDIX F
INEQUALITY DEVELOPMENT

Property: There exist two positive constants $\underline{\zeta}_\gamma$ and $\bar{\zeta}_\gamma$ such that the scaling factor $\gamma(t)$ satisfies the inequality

$$\underline{\zeta}_\gamma < \gamma(t) < \bar{\zeta}_\gamma. \quad (\text{F-1})$$

Proof: Since

$$\gamma = \frac{\|q_v\|}{\|Aq_v\|},$$

the square of $\gamma(t)$ is given by

$$\gamma^2 = \frac{q_v^T q_v}{(Aq_v)^T Aq_v} = \frac{q_v^T q_v}{q_v^T A^T A q_v}. \quad (\text{F-2})$$

Based on the fact that A is of full rank, the symmetric matrix $A^T A$ is positive definite. Hence, the Rayleigh-Ritz theorem can be used to conclude that

$$\lambda_{\min}(A^T A) q_v^T q_v \leq q_v^T A^T A q_v \leq \lambda_{\max}(A^T A) q_v^T q_v, \quad (\text{F-3})$$

where $\lambda_{\min}(A^T A)$ and $\lambda_{\max}(A^T A)$ denote the minimal and maximal eigenvalues of $A^T A$, respectively. From (F-2) and (F-3), it can be concluded that

$$\frac{1}{\lambda_{\max}(A^T A)} \leq \gamma^2 = \frac{q_v^T q_v}{q_v^T A^T A q_v} \leq \frac{1}{\lambda_{\min}(A^T A)}$$

$$\sqrt{\frac{1}{\lambda_{\max}(A^T A)}} \leq \gamma \leq \sqrt{\frac{1}{\lambda_{\min}(A^T A)}}.$$

APPENDIX G
LINEAR PARAMETERIZATION OF TRANSLATION ERROR SYSTEM

The regressor vectors $Y_1(\cdot) \in \mathbb{R}^{1 \times 20}$, $Y_2(\cdot) \in \mathbb{R}^{1 \times 15}$, $Y_3(\cdot) \in \mathbb{R}^{1 \times 5}$ and the constant unknown parameter vectors $\phi_1 \in \mathbb{R}^{20 \times 1}$, $\phi_2 \in \mathbb{R}^{15 \times 1}$, $\phi_3 \in \mathbb{R}^{5 \times 1}$ are given by

$$\begin{aligned}
Y_1 &= \begin{bmatrix} \alpha_i u_i v_{c3} & -\alpha_i v_{c2} & -\alpha_i v_{c3} & -\omega_{c2} & \omega_{c1} & \omega_{c1} & -v_i \omega_{c1} \\ -u_i \omega_{c1} & u_i v_i \omega_{c1} & -\omega_{c3} & v_i \omega_{c3} & -u_i^2 \omega_{c2} & u_i v_i \omega_{c2} & -u_i \omega_{c2} \\ -\omega_{c3} u_i & \omega_{c3} v_i & -\omega_{c3} & \omega_{c2} u_i & -\omega_{c2} v_i & \omega_{c2} \end{bmatrix} \\
\phi_1 &= z_i^* \begin{bmatrix} \frac{1}{a_{11} z_i^*} & \frac{a_{12}}{a_{11} z_i^*} & \frac{a_{13}}{a_{11} z_i^*} & 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13} a_{23}}{a_{11} a_{22}} & \frac{a_{13}}{a_{11} a_{22}} \\ \frac{a_{23}}{a_{11} a_{22}} & \frac{1}{a_{11} a_{22}} & \frac{a_{23}}{a_{22}} & \frac{1}{a_{22}} & \frac{1}{a_{11}^2} & \frac{a_{12}}{a_{11}^2 a_{22}} & \frac{a_{12} a_{23} - a_{13} a_{22}}{a_{11}^2 a_{22}} \\ \frac{a_{12}}{a_{11}^2} & \frac{a_{12}^2}{a_{11}^2 a_{22}} & \frac{a_{12}^2 a_{23} - a_{12} a_{13} a_{22}}{a_{11}^2 a_{22}} & \frac{a_{13}}{a_{11}^2} & \frac{a_{12} a_{13}}{a_{11}^2 a_{22}} & \frac{a_{12} a_{13} a_{23} - a_{13}^2 a_{22}}{a_{11}^2 a_{22}} \end{bmatrix}^T \\
Y_2 &= \begin{bmatrix} \alpha_i v_i v_{c3} & -\alpha_i v_{c3} & \omega_{c1} & \omega_{c1} & -2v_i \omega_{c1} & v_i^2 \omega_{c1} & -\omega_{c3} u_i & \omega_{c3} v_i & -\omega_{c3} \\ -v_i \omega_{2c} u_i & v_i^2 \omega_{2c} & -v_i \omega_{2c} & \omega_{2c} u_i & -\omega_{2c} v_i & \omega_{2c} \end{bmatrix} \\
\phi_2 &= z_i^* \begin{bmatrix} \frac{1}{a_{22} z_i^*} & \frac{a_{23}}{a_{22} z_i^*} & 1 & \frac{a_{23}^2}{a_{22}^2} & \frac{a_{23}}{a_{22}^2} & \frac{1}{a_{22}^2} & \frac{1}{a_{11}} & \frac{a_{12}}{a_{11} a_{22}} & \frac{a_{12} a_{23} - a_{13} a_{22}}{a_{11} a_{22}} \\ \frac{1}{a_{11} a_{22}} & \frac{a_{12}}{a_{11} a_{22}^2} & \frac{a_{12} a_{23} - a_{13} a_{22}}{a_{11} a_{22}^2} & \frac{a_{23}}{a_{11} a_{22}} & \frac{a_{12} a_{23}}{a_{11} a_{22}^2} & \frac{a_{12} a_{23}^2 - a_{13} a_{22} a_{23}}{a_{11} a_{22}^2} \end{bmatrix}^T \\
Y_3 &= \begin{bmatrix} \omega_{2c} u_i & -\omega_{2c} v_i & \omega_{2c} & -v_i \omega_{c1} & \omega_{c1} \end{bmatrix} \\
\phi_3 &= z_i^* \begin{bmatrix} \frac{1}{a_{11}} & \frac{a_{12}}{a_{11} a_{22}} & \frac{a_{12} a_{23} - a_{13} a_{22}}{a_{11} a_{22}} & \frac{1}{a_{22}} & \frac{a_{23}}{a_{22}} \end{bmatrix}^T.
\end{aligned}$$

REFERENCES

- [1] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651–670, 1996.
- [2] F. Chaumette and S. Hutchinson, “Visual servo control part I: Basic approaches,” *IEEE Robot. Automat. Mag.*, vol. 13, no. 4, pp. 82–90, 2006.
- [3] —, “Visual servo control part II: Advanced approaches,” *IEEE Robot. Automat. Mag.*, vol. 14, no. 1, pp. 109–118, 2006.
- [4] G. Hu, N. Gans, and W. E. Dixon, *Complexity and Nonlinearity in Autonomous Robotics, Encyclopedia of Complexity and System Science*. Springer, to appear 2008, ch. Adaptive Visual Servo Control.
- [5] N. J. Cowan and D. Koditschek, “Planar image-based visual servoing as a navigation problem,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 1720–1725.
- [6] N. J. Cowan, J. D. Weingarten, and D. E. Koditschek, “Visual servoing via navigation functions,” *IEEE Trans. Robot. Automat.*, vol. 18, pp. 521–533, 2002.
- [7] Y. Mezouar and F. Chaumette, “Path planning for robust image-based control,” *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 534–549, 2002.
- [8] E. Rimon and D. E. Koditschek, “Exact robot navigation using artificial potential functions,” *IEEE Trans. Robot. Automat.*, vol. 8, pp. 501–518, 1992.
- [9] A. Ruf, M. Tonko, R. Horaud, and H.-H. Nagel, “Visual tracking of an end-effector by adaptive kinematic prediction,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1997, pp. 893–898.
- [10] J. Chen, D. M. Dawson, W. E. Dixon, and A. Behal, “Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension,” *IEEE Trans. Contr. Syst. Technol.*, vol. 13, no. 5, pp. 814–825, 2005.
- [11] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: John Wiley & Sons Inc., 1989.
- [12] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, Massachusetts: MIT Press, 1993.

- [13] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. New York: NY: Cambridge University Press, 2000.
- [14] G. Hu, W. E. Dixon, S. Gupta, and N. Fitz-coy, “A quaternion formulation for homography-based visual servo control,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 2391–2396.
- [15] M. Shuster, “A survey of attitude representations,” *J. Astronautical Sciences*, vol. 41, no. 4, pp. 439–518, 1993.
- [16] G. Hu, S. Mehta, N. Gans, and W. E. Dixon, “Daisy chaining based visual servo control part I: Adaptive quaternion-based tracking control,” in *IEEE Multi-Conf. Syst. Control*, 2007, pp. 1474–1479.
- [17] G. Hu, N. Gans, S. Mehta, and W. E. Dixon, “Daisy chaining based visual servo control part II: Extensions, applications and open problems,” in *IEEE Multi-Conf. Syst. Control*, 2007, pp. 729–734.
- [18] S. Mehta, W. E. Dixon, D. MacArthur, and C. D. Crane, “Visual servo control of an unmanned ground vehicle via a moving airborne monocular camera,” in *Proc. American Control Conf.*, 2006, pp. 5276–5211.
- [19] S. Mehta, G. Hu, N. Gans, and W. E. Dixon, “Adaptive vision-based collaborative tracking control of an ugv via a moving airborne camera: A daisy chaining approach,” in *Proc. IEEE Conf. Decision Control*, 2006, pp. 3867–3872.
- [20] E. Malis, F. Chaumette, and S. Bodet, “2 1/2 D visual servoing,” *IEEE Trans. Robot. Automat.*, vol. 15, no. 2, pp. 238–250, 1999.
- [21] E. Malis and F. Chaumette, “Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods,” *IEEE Trans. Robot. Automat.*, vol. 18, no. 2, pp. 176–186, 2002.
- [22] Y. Fang, W. E. Dixon, D. M. Dawson, and P. Chawda, “Homography-based visual servoing of wheeled mobile robots,” *IEEE Trans. Syst., Man, Cybern. - Part B: Cybern.*, vol. 35, no. 5, pp. 1041–1050, 2005.
- [23] W. E. Dixon, A. Behal, D. M. Dawson, and S. Nagarkatti, *Nonlinear Control of Engineering Systems: A Lyapunov-Based Approach*. Birkhäuser Boston, 2003.
- [24] G. Hu, S. Gupta, N. Fitz-coy, and W. E. Dixon, “Lyapunov-based visual servo tracking control via a quaternion formulation,” in *Proc. IEEE Conf. Decision Control*, 2006, pp. 3861–3866.
- [25] P. Corke and S. Hutchinson, “A new partitioned approach to image-based visual servo control,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 4, pp. 507–515, 2001.

- [26] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, “Keeping features in the field of view in eye-in-hand visual servoing: A switching approach,” *IEEE Trans. Robot.*, vol. 20, no. 5, pp. 908–913, 2004.
- [27] Y. Mezouar and F. Chaumette, “Optimal camera trajectory with image based control,” *Int. J. Robot. Res.*, vol. 22, no. 10, pp. 781–804, 2003.
- [28] H. Zhang and J. Ostrowski, “Visual motion planning for mobile robots,” *IEEE Trans. Robot. Automat.*, vol. 18, no. 2, pp. 199–208, 2002.
- [29] J. Chen, D. M. Dawson, W. E. Dixon, and V. Chitrakaran, “Navigation function based visual servo control,” *Automatica*, vol. 43, pp. 1165–1177, 2007, to appear.
- [30] S. Benhimane and E. Malis, “Vision-based control with respect to planar and nonplanar objects using a zooming camera,” in *Proc. IEEE Int. Conf. Advanced Robotics*, 2003, pp. 991–996.
- [31] N. Garcka-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, “Continuous visual servoing despite the changes of visibility in image features,” *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1214–1220, 2005.
- [32] E. Hecht and A. Zadac, *Optics*, 3rd ed. Addison-Wesley, 1997.
- [33] C. Geyer and K. Daniilidis, “Catadioptric projective geometry,” *Int. J. Computer Vision*, vol. 45, no. 3, pp. 223–243, 2001.
- [34] S. Baker and S. Nayar, “A theory of single-viewpoint catadioptric image formation,” *Int. J. Computer Vision*, vol. 35, no. 2, pp. 175–196, 1999.
- [35] J. Barreto and H. Araujo, “Geometric properties of central catadioptric line images,” in *Proc. European Conf. Computer Vision*, 2002, pp. 237–251.
- [36] C. Geyer and K. Daniilidis, “A unifying theory for central panoramic systems and practical implications,” in *Proc. European Conf. Computer Vision*, 2000, pp. 445–461.
- [37] R. Y. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” *IEEE J. Robot. Automat.*, vol. 3, no. 4, pp. 323–344, 1987.
- [38] —, *Synopsis of recent progress on camera calibration for 3D machine vision*. Cambridge, MA, USA: MIT Press, 1989.
- [39] L. Robert, “Camera calibration without feature extraction,” *Computer Vision and Image Understanding*, vol. 63, no. 2, pp. 314–325, 1996.

- [40] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, 1997, pp. 1106–1112.
- [41] T. A. Clarke and J. G. Fryer, "The development of camera calibration methods and models," *Photogrammetric Record*, vol. 16, no. 91, pp. 51–66, 1998.
- [42] P. F. Sturm and S. J. Maybank, "On plane-based camera calibration: A general algorithm, singularities, applications," in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, 1999, pp. 432–437.
- [43] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. IEEE Int. Conf. Computer Vision*, 1999, pp. 666–673.
- [44] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robot. and Automat.*, vol. RA-3, no. 5, pp. 404–417, 1987.
- [45] J. Feddema and O. Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Trans. Robot. Automat.*, vol. 5, no. 5, pp. 691–700, 1989.
- [46] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1991, pp. 2267–2272.
- [47] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, no. 3, pp. 313–326, 1992.
- [48] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The Confluence of Vision and Control*, ser. LNCIS Series, D. Kriegman, G. Hager, and A. Morse, Eds. Berlin, Germany: Springer-Verlag, 1998, vol. 237, pp. 66–78.
- [49] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics," in *The 3rd Int. Symp. Experimental Robotics*, 1993, pp. 182–192.
- [50] W. J. Wilson, C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 684–696, 1996.
- [51] P. Martinet, J. Gallice, and D. Khadraoui, "Vision based control law using 3D visual features," in *Proc. World Automation Congress*, vol. 3, 1996, pp. 497–502.
- [52] N. Daucher, M. Dhome, J. Lapresté, and G. Rives, "Speed command of a robotic system by monocular pose estimate," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1997, pp. 55–62.

- [53] K. Hashimoto, “A review on vision-based control of robot manipulators,” *Advanced Robotics*, vol. 17, no. 10, pp. 969–991, 2003.
- [54] K. Deguchi, “Optimal motion control for image-based visual servoing by decoupling translation and rotation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1998, pp. 705–711.
- [55] E. Malis and F. Chaumette, “2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement,” *Int. J. Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.
- [56] F. Chaumette and E. Malis, “2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 630–635.
- [57] J. Chen, W. E. Dixon, D. M. Dawson, and M. McIntyre, “Homography-based visual servo tracking control of a wheeled mobile robot,” *IEEE Trans. Robot.*, vol. 22, no. 2, pp. 406–415, 2006.
- [58] N. Gans and S. Hutchinson, “Stable visual servoing through hybrid switched-system control,” *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 530–540, 2007.
- [59] E. Malis, “Visual servoing invariant to changes in camera intrinsic parameters,” in *Proc. IEEE Int. Conf. Computer Vision*, 2001, pp. 704–709.
- [60] Y. Y. Schechner and S. Nayar, “Generalized mosaicing: High dynamic range in a wide field of view,” *Int. J. Computer Vision*, vol. 53, no. 3, pp. 245–267, 2003.
- [61] S. Hsu, H. S. Sawhney, and R. Kumar, “Automated mosaics via topology inference,” *IEEE Computer Graphics and Application*, vol. 22, no. 2, pp. 44–54, 2002.
- [62] M. Irani, P. Anandan, J. Bergen, R. Kumar, and S. Hsu, “Efficient representations of video sequences and their application,” *Signal Processing: Image communication*, vol. 8, pp. 327–351, 1996.
- [63] A. Smolic and T. Wiegand, “High-resolution image mosaicing,” in *Proc. IEEE Int. Conf. Image Processing*, 2001, pp. 872–875.
- [64] R. Swaminathan and S. Nayar, “Non-metric calibration of wide-angle lenses and polycameras,” in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, 2000, pp. 413–419.
- [65] D. Burschka and G. Hager, “Vision-based control of mobile robots,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2001, pp. 1707–1713.

- [66] Y. Mezouar, H. H. Abdelkader, P. Martinet, and F. Chaumette, “Central catadioptric visual servoing from 3D straight lines,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2004, pp. 343–349.
- [67] H. Hadj-Abdelkader, Y. Mezouar, N. Andreff, and P. Martinet, “2 1/2 D visual servoing with central catadioptric cameras,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 3572–3577.
- [68] G. Mariottini, E. Alunno, J. Piazzzi, and D. Prattichizzo, “Epipole-based visual servoing with central catadioptric camera,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2005, pp. 3516–3521.
- [69] G. Mariottini, D. Prattichizzo, and G. Oriolo, “Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 538–544.
- [70] S. Benhimane and E. Malis, “A new approach to vision-based robot control with omni-directional cameras,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 526–531.
- [71] R. Tatsambon and F. Chaumette, “Visual servoing from spheres using a spherical projection model,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 2080–2085.
- [72] ———, “Visual servoing from spheres with paracatadioptric cameras,” in *Int. Conf. Advanced Robotics*, August 2007.
- [73] J. P. Barreto, F. Martin, and R. Horaud, “Visual servoing/tracking using central catadioptric images,” in *Proc. Int. Symp. Experimental Robotics*, 2002, pp. 863–869.
- [74] K. Hosoda and M. Asada, “Versatile visual servoing without knowledge of true jacobian,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1994, pp. 186–193.
- [75] M. Jagersand, O. Fuentes, and R. Nelson, “Experimental evaluation of uncalibrated visual servoing for precision manipulation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 1997, pp. 2874–2880.
- [76] M. Shahamiri and M. Jagersand, “Uncalibrated visual servoing using a biased newton method for on-line singularity detection and avoidance,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 3953–3958.
- [77] J. A. Piepmeier and H. Lipkin, “Uncalibrated eye-in-hand visual servoing,” *Int. J. Robot. Res.*, vol. 22, pp. 805–819, 2003.
- [78] J. A. Piepmeier, G. V. McMurray, and H. Lipkin, “Uncalibrated dynamic visual servoing,” *IEEE Trans. Robot. Automat.*, vol. 24, no. 3, pp. 143–147, 2004.

- [79] R. Kelly, “Robust asymptotically stable visual servoing of planar manipulator,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 759–766, 1996.
- [80] B. Bishop and M. W. Spong, “Adaptive calibration and control of 2D monocular visual servo system,” in *Proc. IFAC Symp. Robot Control*, 1997, pp. 525–530.
- [81] L. Hsu and P. L. S. Aquino, “Adaptive visual tracking with uncertain manipulator dynamics and uncalibrated camera,” in *Proc. IEEE Conf. Decision Control*, 1999, pp. 1248–1253.
- [82] C. J. Taylor and J. P. Ostrowski, “Robust vision-based pose control,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 2734–2740.
- [83] E. Zergeroglu, D. M. Dawson, M. de Queiroz, and A. Behal, “Vision-based nonlinear tracking controllers in the presence of parametric uncertainty,” *IEEE/ASME Trans. Mechatr.*, vol. 6, no. 3, pp. 322–337, 2001.
- [84] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Behal, “Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system,” *IEEE Trans. Syst., Man, Cybern. - Part B: Cybern.*, vol. 31, no. 3, pp. 341–352, 2001.
- [85] A. Astolfi, L. Hsu, M. Netto, and R. Ortega, “Two solutions to the adaptive visual servoing problem,” *IEEE Trans. Robot. Automat.*, vol. 18, no. 3, pp. 387–392, 2002.
- [86] Y. Liu, H. Wang, C. Wang, and K. Lam, “Uncalibrated visual servoing of robots using a depth-independent interaction matrix,” *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 804–817, 2006.
- [87] Y. Fang, W. E. Dixon, D. M. Dawson, and J. Chen, “An exponential class of model-free visual servoing controllers in the presence of uncertain camera calibration,” *Int. J. Robot. Automat.*, vol. 21, 2006.
- [88] G. Hu, N. Gans, and W. E. Dixon, “Quaternion-based visual servo control in the presence of camera calibration error,” in *IEEE Multi-Conf. Syst. Control*, 2007, pp. 1492–1497.
- [89] J. Chen, A. Behal, D. M. Dawson, and W. E. Dixon, “Adaptive visual servoing in the presence of intrinsic calibration uncertainty,” in *Proc. IEEE Conf. Decision Control*, 2003, pp. 5396–5401.
- [90] B. Boufama and R. Mohr, “Epipole and fundamental matrix estimation using virtual parallax,” in *Proc. IEEE Int. Conf. Computer Vision*, 1995, pp. 1030–1036.
- [91] J. Shi and C. Tomasi, “Good features to track,” in *Proc. IEEE Int. Conf. Computer Vision Pattern Recognition*, 1994, pp. 593–600.

- [92] C. Tomasi and T. Kanade, “Detection and tracking of point features,” Carnegie Mellon University, Tech. Rep., 1991.
- [93] O. Faugeras and F. Lustman, “Motion and structure from motion in a piecewise planar environment,” *Int. J. Pattern Recognition and Artificial Intelligence*, vol. 2, no. 3, pp. 485–508, 1988.
- [94] Z. Zhang and A. R. Hanson, “Scaled euclidean 3D reconstruction based on externally uncalibrated cameras,” in *IEEE Symp. Computer Vision*, 1995, pp. 37–42.
- [95] Y. Fang, A. Behal, W. E. Dixon, and D. M. Dawson, “Adaptive 2.5D visual servoing of kinematically redundant robot manipulators,” in *Proc. IEEE Conf. Decision Control*, 2002, pp. 2860–2865.
- [96] J. J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliff, NJ: Prentice Hall, Inc., 1991.
- [97] G. Bradski, “The OpenCV library,” *Dr. Dobb’s Journal of Software Tools*, vol. 25, pp. 120, 122–125, 2000.
- [98] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, M. Booth, and F. Rossi, *GNU Scientific Library: Reference Manual*, Network Theory Ltd., Bristol, UK, 2005.
- [99] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, “Automated tracking and grasping of a moving object with a robotic hand-eye system,” *IEEE Trans. Robot. Automat.*, vol. 9, no. 2, pp. 152–165, 1993.
- [100] G. D. Hager, W.-C. Chang, and A. S. Morse, “Robot hand-eye coordination based on stereo vision,” *IEEE Contr. Syst. Mag.*, vol. 15, no. 1, pp. 30–39, 1995.
- [101] S. Wiiesoma, D. Wolfe, and R. Richards, “Eye-to-hand coordination for vision-guided robot control applications,” *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 65–78, 1993.
- [102] N. Gans, G. Hu, and W. E. Dixon, *Complexity and Nonlinearity in Autonomous Robotics, Encyclopedia of Complexity and System Science*. Springer, to appear 2008, ch. Image-Based State Estimation.
- [103] A. Almansa, A. Desolneux, and S. Vamech, “Vanishing point detection without any a priori information,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 4, pp. 502–507, 2003.

BIOGRAPHICAL SKETCH

Guoqiang Hu received his bachelor's degree in automation engineering from the University of Science and Technology of China (USTC) in 2002, and he received his Master of Philosophy in Automation and Computer-aided Engineering from The Chinese University of Hong Kong (CUHK) in 2004.

Currently, he is pursuing his PhD degree in the nonlinear control and robotics group at the University of Florida (UF) under the supervision of Dr. Warren Dixon.