2011

Nonlinear Estimation And Control For Assistive Robots

Zhao Wang
University of Central Florida

Nonlinear Estimation and Control for Assistive Robots

by

Zhao Wang
B.E. Zhejiang University, 2008

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2011

Major Professor: Aman Behal

ABSTRACT

In this thesis, we document the progress in the estimation and control design of a smart assistive robot arm that can provide assistance during activities of daily living to the elderly and/or users with disabilities. Interaction with the environment is made challenging by the kinematic uncertainty in the robot, imperfect sensor calibration, limited view of angle as well as the fact that most activities of daily living are generally required to be performed in unstructured environments.

For monocular visual systems, range (or depth) information is always crucial for target modeling and system control design. In the first part of my thesis, a novel and effective method is developed to estimate the range information in perspective vision systems by observing the 2-D image information and known motion parameters. We have considered the presence of noise in the image space measurements and kinematic uncertainty in the motion parameters. Simulation and experiment results show the advantage of our algorithm in comparison with other approaches.

In the second part of the thesis, Lyapunov-based design techniques are utilized to propose a 2.5D visual servoing controller which stabilizes the robot end-effector pose while satisfying practical constraints on the sensing and the actuation. First, a nominal feedback controller is introduced which is then modified through an optimization-based approach in order to

satisfy the constraints related to limited camera field-of-view and size of actuation. In the absence of actuator constraints, the proposed control law yields semi-global asymptotic (exponential) stability. When actuator constraints are introduced, the result is local asymptotic stability with known bounds on the region of attraction. Simulation and experimental results demonstrate the effectiveness of the proposed control methodology.

To my parents and my wife

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my advisor, Dr. Aman Behal, for his insightful guidance that is so helpful to the work I have conducted in University of Central Florida. He is incessantly encouraging, supportive, and ready to provide practical help and advice, from basic theoretical details to higher level research. I am also grateful to my committee members: Dr. Ladislau Boloni, Dr. Michael Haralambous, for inspiring discussions on my research.

I would like to thank Dr. Dae-Jin Kim, for his generous and invaluable help, heartwarming friendship and the happy times we had together during the first two years of my graduate study.

Finally, I would like to express my eternal gratefulness to my parents and my wife Panpan Hu. Their unconditional love and support from my parents have fueled me with all the confidence and courage to complete my master degree. My wife Panpan Hu is always there supporting me, caring for me, and bringing joy to my life. Truly, this thesis would not be possible without them.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In this part of the thesis, we have proposed a new method to estimate the range information by using a modified particle filter (PF) algorithm. The motivation for using a particle filter based approach is the presence of both additive and non-additive uncertainty in the motion model which may not lend itself to a Kalman filter or EKF approach. This modified particle filter algorithm introduces a two-level coarse and fine filtering approach which is designed to be computationally efficient. In our system model, we not only consider the noise signal in image space but also the kinematic uncertainty in the motion parameters which is substantial when working with practical robots that have gearing and transmission. In contrast to small noise signal in the image plane in previous works, we introduce a larger noise signal to simulate feature point mismatching and other extreme situations. Simulation and experimental results show the strong performance of the proposed algorithm in comparison with standard particle filter and control theoretic techniques.

The work in this part of the thesis aims to design a control strategy based on HVS that utilizes a Lyapunov-based design approach to guarantee stability – concurrently, robust performance is ensured by employing an optimization approach that minimizes deviation of

the controller from a nominal design while enforcing motion constraints related to the size of the sensor and the finite actuation velocity at the end-effector. The amalgamation of the Lyapunov based design process along with the optimization strategy leads to a simple feedback controller in which the size of the various components of the inputs (*i.e.,* end-effector velocities) are adjusted in an on-line fashion based on the constraints. The stability result achieved is semi-global asymptotic (exponential) regulation when the size of the actuation velocity is not constrained. When finite actuation is considered, the stability result is local asymptotic stability with the ability to explicitly compute an upperbound on the size of the region of convergence.

CHAPTER 2

BACKGROUND

Range estimation in monocular vision system is a popular topic in the field of robotics and machine vision. For example, in visual servoing problems, unavailability or imprecision of range information could lead to possible failure of the visual servoing. In position-based visual servoing (PBVS), we use feature information from the image to estimate the position of the target in the camera coordinates. Then, this estimated position is used to compute error between the current and desired camera position in the world coordinates (or task space) [1]. So, it needs both image information and range information to reconstruct 3-D model for the target. For image-based visual servoing (IBVS), although the control law is designed in the image space, the measurements needed are purely feature point positions in the image plane. We still need an estimation of range information for correctly calculating the interaction matrix, which is also referred to as the image Jacobian. The most common way is using a constant value in the control design. But there is no guarantee for global stability in IBVS. In hybrid visual servoing, taking 2.5-D visual servoing as example, the control is designed in both image space and task space. Although, by introducing extended image coordinates, we don't need to directly calculate range information, the estimation of

unknown depth value d* is still needed in the control design. In [2], a stability bound was set for estimation error in the 2.5-D visual servoing system. Feature points may be out of field of view due to the bad estimation of range information too, which means the control loop is no longer closed. But for the most commonly used 2-D monocular visual servoing system, the range/depth information is unattainable in a single frame. So, an estimation algorithm for range information is essential.

In the last decade, numerous works have addressed the range estimation problem in different systems and fields. It is a classic problem in machine vision and other engineering fields. To solve this problem, extended Kalman filter (EKF) has been used early on by Matthies [3], Sridhar [4, 5, 6], and Hung [7]. Later on, some of the researchers tried to solve this problem by using state observer. Jankovic and Ghosh proposed a new recursive formulation named identifier based observer (IBO) in [8]. The proposed nonlinear filter (IBO) is based on a parameter identifier considered in model reference adaptive systems. Chen and Kano proposed another nonlinear observer combining sliding-mode method, adaptive method, and discontinuous observer techniques in [9], and tested it both on motion parameter invariant and variant systems. Dixon et al. designed a fourth-order observer [10], which can exactly determine the depth information rather than the approximately determined solution of [9]. In [11], Karagiannis and Astolfi designed a reduced-order nonlinear observer for perspective system. It is considerably simpler to implement because it is a first-order system with only one parameter. In recent years, particle filter approach has been used to solve position estimation problem. Recently, Davison et al. have introduced a depth

parameterization method to deal with low parallax in case of feature initialization and distant feature points in monocular SLAM system with EKF framework [12, 13, 14, 15]. In [16], Pupilli and Calway concentrated on estimating camera position by using particle filter. In comparison with SLAM, our problem is more concentrating on precisely estimating range information. In [17], a Gaussian probabilistic model has been proposed to estimate depth information of the feature points.

During the last decade, visual servoing for robots has been investigated by many researchers because robots may have to deal with extrinsic uncertainties such as unstructured and/or dynamic environments, or intrinsic limitations that emanate from the design and construction of the robot. Intrinsic limitations such as kinematic uncertainties are common issues for many mobile, lightweight robots that pervade our lives and interact with and/or share their habitat with humans. Specifically, researchers have conducted extensive studies on eye-in-hand configured camera systems in which visual servoing is performed by a camera held by the robot end-effector. This configuration allows the robot to have visibility in a neighborhood of the end effector, *i.e.,* the robot senses the environment around where it needs to act. However, a disadvantage is that sensor and actuator constraints – specifically, limited field-of-view (FOV) of the camera as well as finite actuation available at the end-effector – can lead to end-effector motions that can cause a target to exit the camera's sensing domain, thereby, leading to loss of feedback which can result in controller failure.

Several innovative approach has been suggested by researchers to alleviate the limited FOV issues with traditional position-based visual servoing (PBVS) and image-based visual

servoing (IBVS) [24][25]. In [26], a partitioned approach to IBVS control is adopted by decoupling approach-axis rotational and translational components of the control from the remaining degrees of freedom, while a potential function that repels feature points from the boundary of the image plane is utilized to guarantee that all features remain in the image throughout the entire trajectory. In [27], a path-following IBVS controller is developed that utilizes a potential function to incorporate motion constraints. In [28], [29], and [30], researchers employ a specialized potential function, namely navigation function, to drive the robot arm to the desired configuration while keeping the feature points in the field of view. In [31], a novel qualitative visual servoing scheme is proposed and integrated into the classic IBVS control law to solve the FOV problem. In [32], using PBVS, an approach powered by iterative computation of motion trajectory has been adopted to resolve the visibility issue. In [33], circular-like rotation and translation trajectories are designed to ensure shorter displacements while ensuring visibility. A modified invariant visual servoing scheme driven by visibility changes of features and a smooth task function has been described in [34]. In [35], a PBVS controller coupled with path planning in the image space was utilized to keep the object in the field of view.

Recently, a path planning method for visual servoing under the visibility constraint has been formulated as a convex optimization problem in [36]. This was followed in [37] by introduction of a boosting algorithm which integrates IBVS and PBVS controls. Other methods have also been proposed that rely on switching techniques. In [38], switching is between PBVS and backward motion of the camera. In [39], the control switches to PBVS

from IBVS when the configuration is close to an image singularity or a local minimum. To the best of our knowledge, detailed stability analysis for multiple switchings between IBVS and PBVS was carried out for the first time in [40]. The results achieved were: (a) local asymptotic stability with an unknown region of convergence, and (b) local stability of the pose and image errors in a well defined neighborhood of the origin. As pointed out in [40], any control strategy that relies upon IBVS is hampered by the presence of local minima and singularities – thus, the best result obtainable is local asymptotic stability.

Another approaches to counter the problems with IBVS and PBVS have been through the use of hybrid visual servoing (HVS) techniques that combine 2D (pixel-space) and 3D (task-space) feedback to guarantee stability in a large-signal sense [41, 42, 43, 44, 45]. While HVS (also known as 2.5D servoing) techniques generate intuitive motions in the task and pixel space, they still remain highly susceptible to objects exiting the camera FOV due to lack of coordination between the rotational and translational variables, *e.g.,* rotations of the in-hand camera, especially when the end-effector is close to the object, can result in large pixel errors and consequently loss of object in the visible cone if the translation velocity cannot catch up due to, say, actuation size limitations. In order to combine the advantages of the global stability of HVS along with the effectiveness of the switching approaches against limited FOV, a region-based scheme was proposed in [46] in order to switch between HVS control and pure translatory motion; however, the stability was not guaranteed through a rigorous analysis.

CHAPTER 3

PROBLEM STATEMENT

For a monocular vision system, we can observe a set of 2-D feature points in the image plane from a CCD camera across frames. Besides the image measurements, we also compute the required motion parameters (e.g., linear and/or angular velocity) across frames. Our goal is to estimate the range information (i.e., depth in 3-D space) of these feature points to reconstruct the 3-D position of the feature points in the camera coordinates using a sequence of observations.

The research objective is to drive the 6-DOF robot end-effector pose (and consequently the pose of the in-hand camera) to a setpoint while utilizing measurements of feature locations extracted from a video stream. The challenge is to rigorously ensure stability and convergence of the end-effector pose while considering limitations on the size of the video sensor as well as the size of the kinematic input, *i.e.,* the end-effector velocity vector. We note here that the end-effector velocity vector is amplitude limited for safety reasons in many applications where the robot's work space is in close proximity with a human, *e.g.,* an assistive robot mounted on the size of a patient's wheelchair. We also assume that the limits on the sensor size and the actuation are known apriori.

CHAPTER 4

METHODOLOGY

4.1 Particle Filter Based Range Estimation

4.1.1 System Model

In this section, a state vector is defined as the unknown 3-D position of the observed 2-D feature point and it is represented as

$$x\left(t\right)=\left[X\left(t\right),Y\left(t\right),Z\left(t\right)\right]^{T} \tag{4.1}$$

Then, an observed 2-D feature point is represented as a projection as follows

$$y(t)=\left[y_{1}\left(t\right),y_{2}\left(t\right)\right]^{T}=\left[\frac{X\left(t\right)}{Z\left(t\right)},\frac{Y\left(t\right)}{Z\left(t\right)}\right]^{T} \tag{4.2}$$

Projected representation of feature point requires knowledge of camera's internal parameters, i.e., the initial measurement of feature point in the pixel coordinates is defined as follows

$$p\left(t\right)=\left[u\left(t\right),v\left(t\right),1\right]^{T} \tag{4.3}$$

where $u(t)$ and $v(t)$ stand for location of feature point in the image plane in x and y directions, respectively. From initial measurement, an observation vector in the homogeneous

coordinates can be computed using

$$[y_1(t), y_2(t), 1]^T = A^{-1} * p(t) \tag{4.4}$$

where A is a 3×3 constant, known, and invertible intrinsic camera calibration matrix [2].The aforementioned state and observation vectors are used to formulate a system model as follows

$$\begin{aligned} \dot{x}_t &= f_t(x_t, \theta_t, v_t) \\ y_t &= h_t(x_t, n_t) \end{aligned} \tag{4.5}$$

where x_t denotes the unknown state vector, θ_t is the known motion parameters to system, y_t is the observation, while v_t and n_t represent the uncertainty in motion parameters and the image measurement, respectively. Generally, the system can be modeled as a linear or nonlinear setup. In this problem, the unknown state vector is the 3-D position of the feature point in the camera coordinates, the motion parameters are the translational and rotational velocity vectors, while the observation is the 2-D position of the feature point in the current scene.

A state space model, which models the time evolution of the system, is described in the form of differential or difference equation. As shown in [9], a deterministic general affine model for our system can be described as

$$\begin{bmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} * \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{4.6}$$

where $a_{ij}(t)$ and $b_i(t)$ are known motion parameters. Frequently, motion parameters show variations for a variety of reasons [21]. Among them, kinematic uncertainty caused by gearing

and transmission of the robot system mostly affects state space model, so we can represent a more general probabilistic state space model as follows

$$\begin{bmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \end{bmatrix} = \begin{bmatrix} \bar{a}_{11} & \bar{a}_{12} & \bar{a}_{13} \\ \bar{a}_{21} & \bar{a}_{22} & \bar{a}_{23} \\ \bar{a}_{31} & \bar{a}_{32} & \bar{a}_{33} \end{bmatrix} * \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} + \begin{bmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \end{bmatrix} \tag{4.7}$$

where the noise signal is represented by additive noise signals in the parameters $a_{ij}(t)$ and $b_i(t)$, which are defined as

$$\begin{aligned} \bar{a}_{ij} &= a_{ij} + v_{a,ij} \\ \bar{b}_{ij} &= b_{ij} + v_{b,ij} \end{aligned} \tag{4.8}$$

An observation model is defined by the property of the physical system. In this problem, an observation model is formed as a perspective vision system [9] and is defined as

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{Z(t)} & 0 & 0 \\ 0 & \frac{1}{Z(t)} & 0 \end{bmatrix} * \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} + n_t \tag{4.9}$$

where n_t is additive measurement error vector. This error is caused by multiple reasons such as quantization errors, incorrect detection of feature point, and mismatched feature point with reference template image. Although quantization errors can be modeled as a uniformly distributed random signal, the error caused by mismatched feature points is dominant – these feature point mismatches can be adequately modeled using a Gaussian distribution, n_t, which is assumed to be Gaussian random variable with known mean and variance. We

note here that the non-additive nature of noise in (4.7) (i.e., uncertainty is multiplied with state) motivates us to pursue a particle filter approach in lieu of the commonly employed Extended Kalman Filter (EKF) approach. More specifically, we are suggesting that this nonlinearity may result in the evolving probability distribution for the state vector to be not Gaussian even if the underlying uncertainty vectors are Gaussian.

4.1.2 Two-level Particle Filter Approach

As previously stated, we adopt a particle filter approach to estimate the range information. Fundamentally, the particle filter (PF) algorithm is a Bayesian estimation of states of a given system where the posterior probabilistic density function (pdf) of the state is updated by a Monte Carlo method [19]. In the particle filter approach, a prior pdf can be represented by a set of random samples (i.e., particles) as

$$p\left(x_{0:t}|y_{1:t}\right) \approx \sum_{i=1}^{N} w_t^i \delta(x_{0:t} - x_{0:t}^i) \tag{4.10}$$

In this range estimation problem, particles are considered as a set of 3-D points in the camera coordinates. Here, x_t^i denotes the i-th particle at time t while wit is an associated important weight of this particle. Also, y_t stands for observation at time t. The PF algorithm consists of two steps, namely, 'prediction' and 'update' [18]. Given the probability $p(x_{t-1}|y_{1:t-1})$ based on all the available observations from time 1 to time $t-1$, the prediction

step is performed using the evolution function to predict the prior pdf at time t as

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1}, \theta_{t-1}) p(x_{t-1}|y_{1:t-1}) dx_{t-1} \tag{4.11}$$

Then, the state x_t can be updated using Bayes rule with the predicted prior pdf and current observation of the feature points in the image plane as

$$p(x_t|y_{1:t}) = \frac{p(y_t|x_t) p(x_t|y_{1:t-1})}{p(y_t|y_{1:t-1})} \tag{4.12}$$

Throughout this part of thesis, a basic procedure for PF described in [18] is followed with our modification. An observation vector y_t is defined in the homogenous coordinates which arises from the projection of 3-D point in the camera coordinates. In view of this perspective projection, each observation vector can be mapped to 3-D line, not a 3-D point in the camera coordinates. Thus, it is easy to see that the homogeneous coordinates from projection of 3-D points experience a 'projection ambiguity' phenomenon. Fig. 4.1 shows a consequence of the projection ambiguity phenomenon in our experiment. Trajectory A shows the projection of the true state in the image space – the initial 3-D coordinates in camera-space for this point are $x_a = \begin{bmatrix} 87, & 41, & 305 \end{bmatrix}^T$. Trajectories B and C represent the projections of two particles initialized at $x_b = \begin{bmatrix} 100, & 47, & 350 \end{bmatrix}^T$ and $x_c = \begin{bmatrix} 90, & 43, & 305 \end{bmatrix}^T$. All three particles are driven by the same evolution model. It's clear to see that all three projected trajectories twist with each other between "Marker 1" and "Marker 2". Although particle C is much closer to the true state than particle B in the 3-D coordinates, the algorithm can only update according to the ambiguous 2-D projection. Thus, the estimation will initially be poor since both particles B and C will evolve with similar weights owing to their projections being

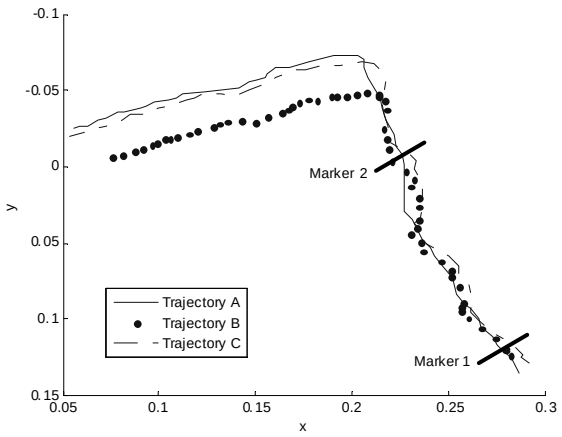


Figure 4.1: Projection ambiguity phenomenon

similar. We remark here that the above situation does not imply that the ordinary PF algorithm is unable to resolve this projective ambiguity. i.e., modulo a singular feature point located at the so-called focus of expansion, the system is observable.

Another drawback of the particle filter approach is that a large number of particles are needed to estimate high-dimensional states, which implies very low computation efficiency in high order system and poor performance for on-line estimation problem [22]. For example, if we draw N samples in each dimension of 3-D space, the total number of the particles will be N^3. The corresponding computation complexity is $O(N^3)$. Fig. 4.2 shows the relationship between the number of particles and estimation result. Thus, the standard PF method meets a dilemma in solving range estimation problem for perspective systems. To represent the posterior pdf $p(x_t|y_{0:t})$ in a 3-D space and estimate it precisely, a large number of particles

are needed. However, it means we expect to see lower computation efficiency. Our goal in this work is to find a modified algorithm that leads to efficient estimation.

As a remedy for the above issues, we propose a two-level approach. The coarse-level is a 1-D (Z axis) PF to coarsely estimate the depth of the feature points. Based on the result from the coarse-level, a 3-D fine-level particle filter was formed in a small region with relatively small number of particles. The employment of the coarse-level particle filter allows us to reduce the number of particles while the fine-level particle set helps to achieve a satisfactory estimation result. Fig. 4.3 shows that by using the two-level PF as described, the importance weight of the incorrect particle (from the discussion above) drops much more quickly than the conventional PF approach [18] which implies that the proposed approach allows for faster ambiguity resolution. In the next section, we will demonstrate that our modified PF can converge faster with a small number of particles. To achieve the best estimation result for the coarse-level particle filter, firstly, coarse-level particles $x_0^{1,i}$ are generated from the observations y_1 and y_2 in the first frame (see (4.4)) by considering a uniform distribution of Z_0^i as

$$\begin{aligned} Z_0^i &\sim U(Z_{\min,} Z_{\max}) \\ x_0^{1,i} &= [y_1 Z_0^i, y_2 Z_0^i, Z_0^i], \quad \text{for } i = 1, 2, ...N \end{aligned} \tag{4.13}$$

By choosing the coarse level particles along the homogeneous line, whose projection on the image plane is $\begin{bmatrix} y1 & y2 \end{bmatrix}$, the coarse-level PF will give us a coarsely estimated range information. Since all coarse level particles are equally likely distributed, the fine-level particles are initially drawn around any randomly chosen coarse level particle. At succeeding

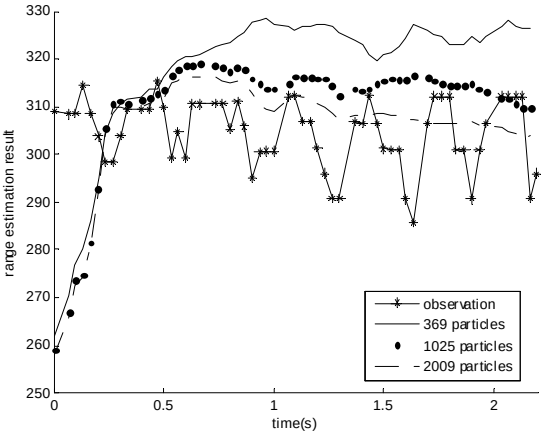


Figure 4.2: Relationship between the number of particles and estimation result using a conventional PF

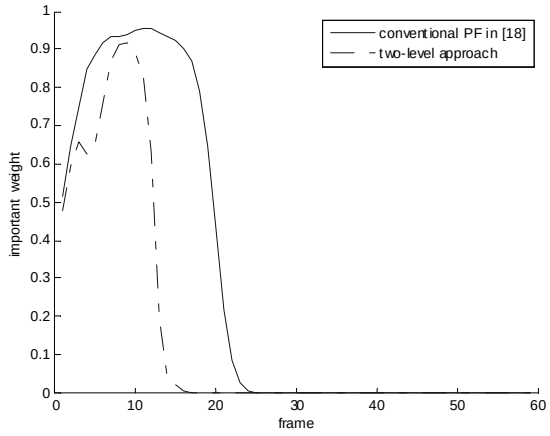


Figure 4.3: Performance Comparison

frames after the first, fine level particles need to be reinitialized under the following condition. If there is a change in the index of the coarse level particle with the highest weight between the current and the previous frames, a reset is triggered and fine-level particles are uniformly drawn from a region around this coarse-level particle with the highest importance weight. In our case, we define this region as a cuboid in the camera coordinates centered at $x_t^{1,max}$ (which is defined as the coarse particle with the highest weight). The size of this cuboid is dependent on the noise signal in the image plane, kinematic uncertainty in the motion parameters, and the estimation precision of the coarse-level filter.

In both coarse and fine-particle filter algorithm, assuming that we have a posterior pdf $p(x_{t-1}|y_{0:t-1})$ of the state at time $t-1$, the first step is to draw samples from the importance function as

$$p(x_t|x_{t-1}, \theta_{t-1}) \tag{4.14}$$

where θ_{t-1} represents the motion parameters (4.5) of the system. Thus, in this implementation, the prior is being utilized as the importance function from which it is easy to draw samples. As we have shown in (4.8), the noise signal from kinematic uncertainty affects motion parameters of the state space model. In the literature, this noise signal v_a and v_b can be modeled using a Gaussian distribution with zero mean as

$$\begin{aligned} v_{a,ij} &\sim N(0, (k_a * a_{ij})^2) \\ v_{b,i} &\sim N(0, (k_b * b_i)^2) \end{aligned} \tag{4.15}$$

where the variance of this uncertainty is a function of motion parameters of the state space model in (4.7) – here, k_a, for example, represents the percentage noise in the signal. In

consideration of this noise signal, the final motion parameters can be represented as

$$\begin{aligned} \bar{a}_{ij} &\sim N(a_{ij}, (k_a * a_{ij})^2) \\ \bar{b}_i &\sim N(b_i, (k_b * b_i)^2) \end{aligned} \tag{4.16}$$

Next, the prior pdf of particle filters can be generated using (4.7)

In the update step, the predicted prior pdf of previous step is corrected via measurements. To this end, the important weights of the particles in the prior distribution will be updated based on the current observation through following equation

$$\bar{w}_t^i \propto p(y_t|x_t^i) w_{t-1}^i \tag{4.17}$$

where $p(y_t|x_t)$ is defined as observer posterior [18] which can be derived from observation model (see (4.9)). As similarly done for the uncertainties in the motion parameters, n_t is assumed to be a Gaussian random signal with known mean and variance. To get the analytical form of the observer posterior, the image measurement error n_t is defined by the following form

$$p(n_t) = \frac{1}{(2\pi)\,|K|^{1/2}} \exp\left(-\frac{1}{2} n_t^T K^{-1} n_t\right) \tag{4.18}$$

Furthermore, without loss of generality, it is assumed that the error distribution of each dimension of the image plane shows probabilistically independent behavior. Hence, the covariance matrix K is defined as

$$K = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \tag{4.19}$$

Based on the probabilistic distribution (4.18), the observer posterior can be defined as

$$p(y_t|x_t^i) = \frac{1}{2\pi|K|^{1/2}} \exp\left(-\frac{1}{2}\left(y - h_t\left(x_t^i\right)\right)^T K^{-1}\left(y - h_t\left(x_t^i\right)\right)\right) \tag{4.20}$$

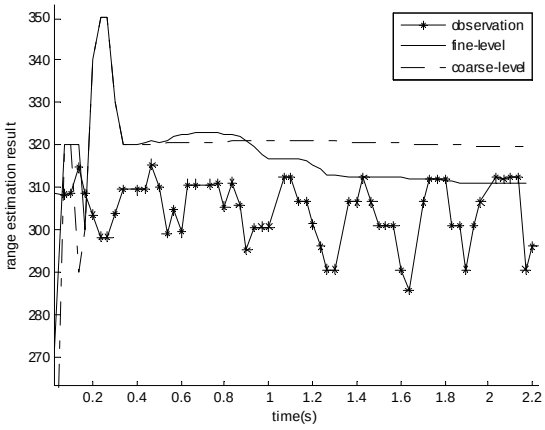


Figure 4.4: Performance comparison between the coarse and fine-level filter

At time t, the associated weights of the updated particles are normalized using

$$w_t^i = \frac{\bar{w}_t^i}{\sum_{i=1}^{N} \bar{w}_t^i} \tag{4.21}$$

Both coarse and fine-level particle sets will be updated until the end of the estimation process. Thus, time coherency is maintained for both the filters throughout the entire process – the only exception to this is when the fine particle filter gets reinitialized as described above. Fig. 4.4 shows a simulation example showing the time evolution of the coarse and fine filters. The dash-dot line shows the coarsely estimated depth. By sampling around this coarse result, fine-level particle filter (solid line) compensates the error, and eventually approaches the true state.

4.1.3 Two-Level Particle Filter Framework

Based on the aforementioned analysis, we propose a particle filter framework for range estimation as follows:

1) Initialization:

Draw particles for set $x_{1,i}$, $x_{2,i}$ as Fig. 4.2.

2) Prediction:

For $i = 1, \ldots, N_1$ and $j = 1, \ldots, N_2$, draw particles via

$$x_t^{1,i} \sim p(x_t^1 | x_{t-1}^1, \theta_{t-1}) \tag{4.22}$$

$$x_t^{2,j} \sim p(x_t^2 | x_{t-1}^2, \theta_{t-1}) \tag{4.23}$$

3) Update:

For $i = 1, \ldots, N_1$ and $j = 1, \ldots, N_2$, compute

$$w_t^{1,i} = p(y_t | x_{t-1}^1) w_{t-1}^{1,i} \tag{4.24}$$

$$w_t^{2,j} = p(y_t | x_{t-1}^2) w_{t-1}^{2,j} \tag{4.25}$$

4) Fine Level Particle Reinitialization

Find i_{max} in the coarse-level in the current frame, if i_{max} not equal to i_{max} in last frame, reset fine-level particles.

5) Resampling

Both coarse and fine-level particles are resampled whenever degeneracy phenomenon is detected [18].

4.1.4 Simulation and Experiment Results

Prior to experiment, we implemented the proposed algorithm using MATLAB^{TM} and relevant toolboxes in standard Windows XP operating system on Intel Pentium^{TM} IV 4GHz, 1GB memory configured system. In the simulation, the sampling frequency of modified particle filter algorithm is set to 25[Hz]. The number of particles in coarse-level is $N_1 = 50$ while the number of particles in fine-level is $N_2 = 343$. We chose two different observer-based range estimation approaches available in literature, namely, the fourth-order observer of [10] and the reduced-order observer of [11], and compare our results with them in the simulation environment. The following motion model is chosen in consideration of its use in many other relevant works [9, 10, 11]

$$\begin{bmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \end{bmatrix} = \begin{bmatrix} -0.2 & 0.4 & -0.6 \\ 0.1 & -0.2 & 0.3 \\ 0.3 & -0.4 & 0.4 \end{bmatrix} * \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.25 \\ 0.3 \end{bmatrix} \tag{4.26}$$

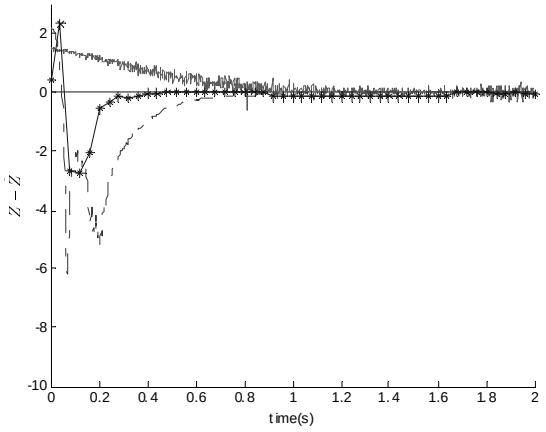


Figure 4.5: Comparison with observers in [10] and [11]

Here, the initial condition of the state is given as

$$[X(0), Y(0), Z(0)]^T = [1, 1.5, 2.5]^T \tag{4.27}$$

To generate the results shown below, λ is set equal to 20 for the method in [11] while all the parameters in [10] are utilized as reported therein.

With 1% image measurement error and no kinematic uncertainty, all the methods show satisfactory convergence in estimation of range information as shown in Fig. 4.5. (Starred line: the proposed algorithm; Dash-dot line: fourth-order observer [10]; Solid line: reduced-order observer [11]. No kinematic uncertainty is considered. Only 1% image measurement error is considered. $k_a = k_b = 0.00, \sigma_x = \sigma_y = 0.01$). The proposed algorithm has the fastest convergence; moreover, the observer based methods show noisy performance. Next, we performed a comparison when 10% kinematic uncertainty is present along with higher percentage of image measurement error (5%). The reduced-order observer [11] shows very

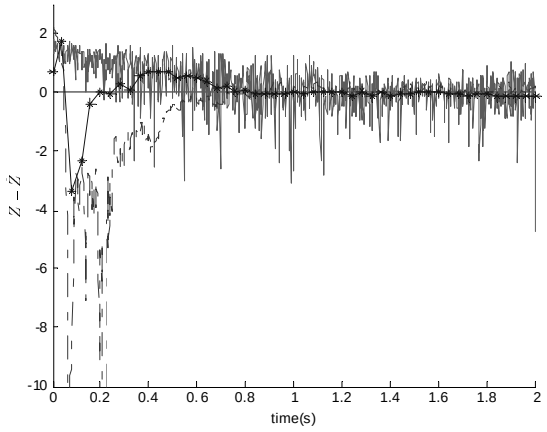
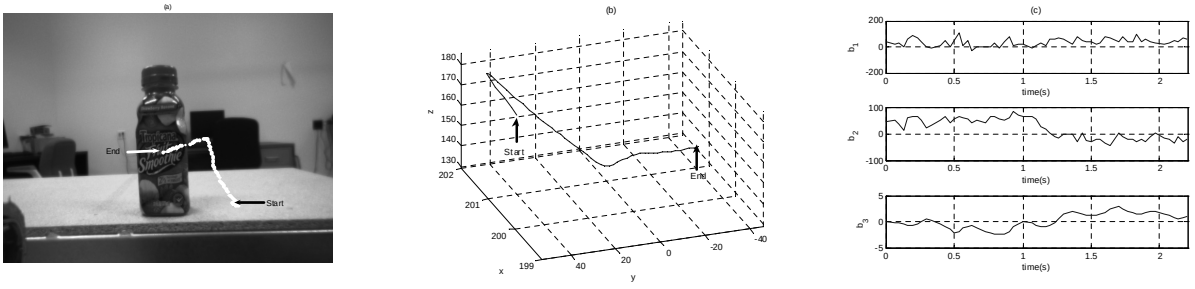


Figure 4.6: Comparison with observers in [10] and [11]

unreliable performance as shown in Fig. 4.6 (Starred line: the proposed algorithm; Dash-dot line: fourth-order observer [10]; Solid line: reduced-order observer [11]. 10% kinematic uncertainty and 5% image measurement errors are considered. $k_a = k_b = 0.10, \sigma_x = \sigma_y = 0.05$). Furthermore, smaller transient error and faster convergence rate for the two-level PF compared to the scheme in [10] implies that the proposed method is more feasible even under non-ideal conditions.

For obtaining experimental results, we gathered a set of 640×480 image sequences at 30[Hz]. Stereo images were captured using a pair of Point-Grey Dragonfly2 cameras and SIFT features were matched between the stereo pairs in order to determine the ground truth depth information. Camera velocities were obtained using the measured motion of the robot end-effector.

Coarse-level particle set was chosen to contain 41 samples; fine-level particle set contained 176 samples. We compared the performance between the proposed method and the PF approach in [18] at first. We also compare the performance of fourth-order observer [10] and our two-level approach with the experimental data gathered from our system. Fig. 4.7-(a) shows the 2D trajectory (white) of feature point in the image plane, which is obtained from our CCD camera and a standard feature detector – this is used as measurement in the update step 4.17. Fig. 4.7-(b) shows the corresponding 3D trajectory of camera motion measured by the encoders in the links of the robot. This motion of the camera is used to obtain the velocities $b_i(t)$ used in (4.14) (see Fig. 4.7-(c) for a plot of these velocities). Fig. 4.8 depicts the experimental depth estimation result comparing the PF in [18] with our method. The proposed algorithm performs better than the PF approach [18] in view of convergence rate while the steady-state estimation error is nearly the same. We note here that the total number of particles employed for the proposed algorithm is 217 but is 2009 for the method in [18]. Fig. 4.9 shows the comparison between the observer based estimator in [10] and the proposed approach. It can be seen that the design from [10] cannot obtain convergence. It is especially vulnerable when there is a change in end-effector velocity (around $t = 1$[s] in the middle curve on Fig. 4.7-(c)). While it is possible to choose a bigger gain as noted in [10] to obtain theoretical asymptotic stability, this leads to poor overall stability and results in large oscillations in the range estimate. The result shown in Fig. 4.9 was the best that we could obtain after a long period of tuning a large number of gains through trial and error.



(a) (b) (c)

Figure 4.7: Comparison between Nominal and Modified control laws during simulation

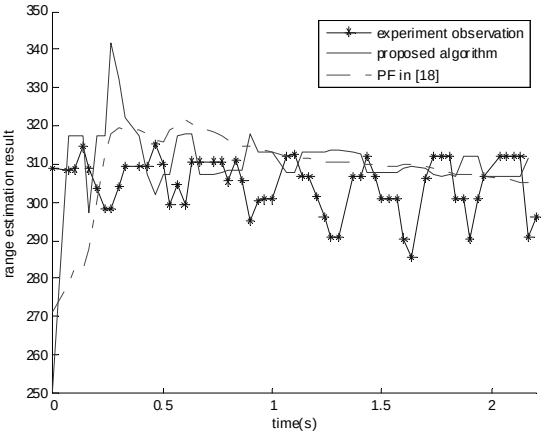


Figure 4.8: Comparison between the proposed method and particle filter approach in [18]

In comparison, the proposed approach shows rapid convergence and is not overly restrictive in terms of end-effector motion or implementation.

Both simulation and experimental results show that our proposed method outperforms the standard PF in [18] and observer based estimators proposed in [10] and [11]. Firstly, our method is very robust for the system with large image space and kinematic uncertainty. Secondly, in comparison with state observer where motion parameters are required to have some

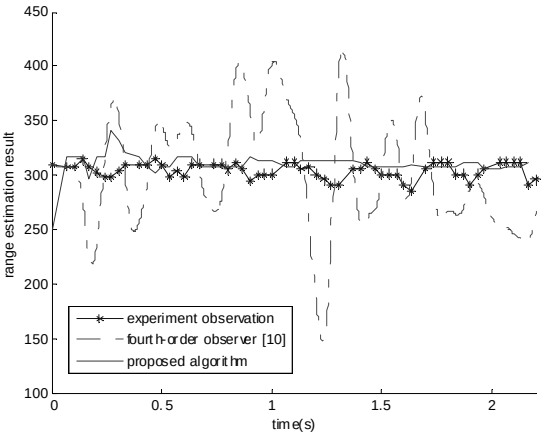


Figure 4.9: Comparison between the proposed method and observer in [10]

level of differentiability, the two-level PF approach can still work when motion parameters are non-smooth or even discontinuous. The performance and stability of the fourth-order state observer also greatly relies on the observer gains. It was hard to find a set of gains for the observer to work in all situations. Moreover, the state observers that we tested, to varying degrees, were susceptible to noise – the result of reduced order observer in [11] (while very simple to implement) was the most susceptible. All these requirements greatly restrict the application of state observers. Thirdly, comparing with standard PF in [18], our method can achieve nearly the same estimation result with smaller number of particles – a 10 fold decrease in the number of particles was achieved as discussed previously. Finally, an advantage of this approach over EKF based approaches is that it can cope with both nonlinear and non-Gaussian problems because it is derived from standard PF. In this work, the proposed method is presented as an estimation of range information for only one feature

point. Without loss of generality, this scheme could be easily extended to multiple feature points. Using three or more non-collinear 3D points, a target plane might be recovered and corresponding range information could be estimated.

4.2 Lyapunov-based Stable Visual Servoing Design

4.2.1 System Model

To setup the problem, one can begin by referring to Fig. 4.10 where a reference plane π is considered with four feature points denoted by $O_i \quad \forall i = 1, 2, 3, 4$ – here, no three of the coplanar feature points are considered to be collinear. It is noted that the classic eight-points algorithm or the Virtual Parallax method [47] can be utilized if four coplanar target points are not available, therefore, the reference plane π can denote a real or virtual plane associated with a target. The 3D position of the i^{th} feature point $\bar{\mathbf{m}}_i \in \mathbb{R}^{3\times 1}$ in the current camera coordinates $\mathcal{F}$ and $\bar{\mathbf{m}}_i^* \in \mathbb{R}^{3\times 1}$ in the desired camera coordinates $\mathcal{F}^*$, respectively, are expressed as follows

$$\bar{\mathbf{m}}_i = \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^T, \quad \bar{\mathbf{m}}_i^* = \begin{bmatrix} X_i^* & Y_i^* & Z_i^* \end{bmatrix}^T. \tag{4.28}$$

The time derivative of $\bar{\mathbf{m}}_i$ can be written as follows [42]

$$\dot{\bar{\mathbf{m}}}_i = -\mathbf{v}_c + [\bar{\mathbf{m}}_i]_\times \boldsymbol{\omega}_c \tag{4.29}$$

where $\mathbf{v}_c \in \mathbb{R}^{3\times 1}$ and $\boldsymbol{\omega}_c \in \mathbb{R}^{3\times 1}$ denote camera linear and angular velocities, while the notation $[\boldsymbol{\zeta}]_\times$ represents the skew-symmetric matrix associated with a vector $\boldsymbol{\zeta}$. One can also define the normalized coordinates $\mathbf{m}_i$ and $\mathbf{m}_i^*$ for the target points $\bar{\mathbf{m}}_i$ and $\bar{\mathbf{m}}_i^*$, respectively, in the following manner

$$\mathbf{m}_i = \frac{\bar{\mathbf{m}}_i}{Z_i} = \begin{bmatrix} \frac{X_i}{Z_i} & \frac{Y_i}{Z_i} & 1 \end{bmatrix}^T, \quad \mathbf{m}_i^* = \frac{\bar{\mathbf{m}}_i^*}{Z_i^*} = \begin{bmatrix} \frac{X_i^*}{Z_i^*} & \frac{Y_i^*}{Z_i^*} & 1 \end{bmatrix}^T. \tag{4.30}$$

Here, the normalized coordinates $\mathbf{m}_i$ can be obtained through the pixel coordinates $\mathbf{p}_i = \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^T \in \mathbb{R}^{3\times 1}$ and the global invertible transformation $\mathbf{p}_i = \mathbf{A}\mathbf{m}_i$ where $\mathbf{A} \in \mathbb{R}^{3\times 3}$ is a known, invertible, intrinsic camera calibration matrix – thus, $\mathbf{m}_i$ is considered known given measurements for the corresponding $\mathbf{p}_i$**.** Furthermore, one can define the extended image coordinates $\mathbf{m}_{e,i} \in \mathbb{R}^{3\times 1}$ and $\mathbf{m}_{e,i}^* \in \mathbb{R}^{3\times 1}$ as follows [42]

$$\mathbf{m}_{e,i} \triangleq \begin{bmatrix} \frac{X_i}{Z_i} & \frac{Y_i}{Z_i} & \log(Z_i) \end{bmatrix}^T, \quad \mathbf{m}_{e,i}^* \triangleq \begin{bmatrix} \frac{X_i^*}{Z_i^*} & \frac{Y_i^*}{Z_i^*} & \log(Z_i^*) \end{bmatrix}^T. \tag{4.31}$$

Note that the first two elements in $\mathbf{m}_{e,i}$ and $\mathbf{m}_{e,i}^*$ are the same as the ones in $\mathbf{m}_i$ and $\mathbf{m}_i^*$, which can be directly obtained from current and desired frame. After taking the time derivative of $\mathbf{m}_{e,i}$ along the dynamics of (4.29), the open-loop dynamics of the extended image coordinates for the i^{th} feature point can be obtained as follows

$$\dot{\mathbf{m}}_{e,i} = -\frac{1}{Z_i}\mathbf{L}_v^i \dot{\bar{\mathbf{m}}}_i = \begin{bmatrix} \frac{1}{Z_i}\mathbf{L}_v^i & \mathbf{L}_c^i \end{bmatrix} \boldsymbol{\tau}_c \tag{4.32}$$

where $\boldsymbol{\tau}_c = \begin{bmatrix} \mathbf{v}_c^T & \boldsymbol{\omega}_c^T \end{bmatrix}^T \in \mathbb{R}^{6\times 1}$ while $\mathbf{L}_v^i(\mathbf{m}_{e,i}) \in \mathbb{R}^{3\times 3}$ and $\mathbf{L}_c^i(\mathbf{m}_{e,i}) \in \mathbb{R}^{3\times 3}$ are defined as follows

$$\mathbf{L}_v^i = \begin{bmatrix} -1 & 0 & m_{e,i,1} \\ 0 & -1 & m_{e,i,2} \\ 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{L}_c^i = \begin{bmatrix} m_{e,i,1}m_{e,i,2} & -1-m_{e,i,1}^2 & m_{e,i,2} \\ 1+m_{e,i,2}^2 & -m_{e,i,1}m_{e,i,2} & -m_{e,i,1} \\ -m_{e,i,2} & m_{e,i,1} & 0 \end{bmatrix} \tag{4.33}$$

where $m_{e,i,j}$ denotes the j^{th} element in $\mathbf{m}_{e,i}$. For the technical details of (4.32) and (4.33), the reader is referred to [42]. Also note that $Z_i = \gamma_2^i d^*$ where d^* denotes a fixed distance to the plane π from the desired camera position; the ratio $\gamma_2^i \triangleq Z_i/d^*$ can be computed via decomposition of a Euclidean homography [42]. Furthermore, given the i^{th} feature point, the translational error $\mathbf{e}_v = \begin{bmatrix} e_{v,1} & e_{v,2} & e_{v,3} \end{bmatrix}^T \in \mathbb{R}^{3\times 1}$, which is used for quantifying the position mismatch of a feature point between coordinates $\mathcal{F}$ and $\mathcal{F}^*$, is defined as follows

$$\mathbf{e}_v \triangleq \mathbf{m}_{e,i} - \mathbf{m}_{e,i}^* \tag{4.34}$$

where $\mathbf{m}_{e,i}$ and $\mathbf{m}_{e,i}^*$ have been defined previously in (4.31). Here $e_{v,1}(t)$ and $e_{v,2}(t)$ are clearly measurable from the description above, and the depth error $e_{v,3}(t) \triangleq \log\left(\frac{Z_i}{Z_i^*}\right)$ can be computed as follows [42]

$$\frac{Z_i}{Z_i^*} = \frac{(1+\mathbf{n}^T\mathbf{x}_f)\mathbf{n}^{*T}\mathbf{m}_i^*}{\mathbf{n}^T\mathbf{m}_i}. \tag{4.35}$$

where $\mathbf{n} \triangleq \mathbf{R}\mathbf{n}^* \in \mathbb{R}^{3\times 1}$ while $\mathbf{n}^* \in \mathbb{R}^{3\times 1}$ is the unit vector normal to plane π expressed in $\mathcal{F}^*$ and $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is the rotation matrix between coordinates $\mathcal{F}$ and $\mathcal{F}^*$. $\mathbf{x}_f \in \mathbb{R}^{3\times 1}$ represents a scaled translational error. The rotation matrix $\mathbf{R}$, the normal $\mathbf{n}^*$, and the scaled translation $\mathbf{x}_f$ can be obtained from the aforementioned homography decomposition. The orientation

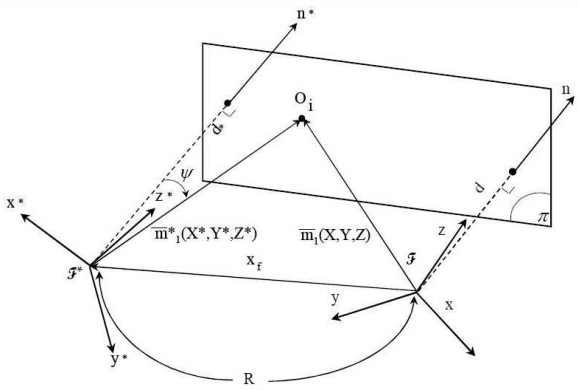


Figure 4.10: Coordinate frame relationships between the reference plane and the in-hand camera at the current and reference locations [44]

error between $\mathcal{F}$ and $\mathcal{F}^*$ can be rewritten as follows

$$\mathbf{e}_\omega = \mathbf{u}\theta \tag{4.36}$$

where $\mathbf{u} \in \mathbb{R}^{3\times 1}$ represents a rotation axis and θ denotes the rotation angle about $\mathbf{u}$. The reader is referred to [23] and [49] for details on how to obtain $\mathbf{u}$ and θ given the rotation matrix $\mathbf{R}$.

4.2.2 Control Design

Based on the definitions in (4.32), (4.34), and the time derivative of (4.34) and (4.36), the dynamics of the open-loop translation and rotation errors can be expressed as follows [48]

$$\dot{\mathbf{e}}_v = \frac{1}{\gamma_2^i d^*}\mathbf{L}_v^i \mathbf{v}_c + \mathbf{L}_c^i \boldsymbol{\omega}_c, \quad \dot{\mathbf{e}}_\omega = \mathbf{L}_\omega \boldsymbol{\omega}_c \tag{4.37}$$

where $\mathbf{L}_\omega \in \mathbb{R}^{3\times 3}$ is given as follows [42]

$$\mathbf{L}_\omega = \mathbf{I}_3 - \frac{\theta}{2}[\mathbf{u}]_\times + \left(1 - \frac{sinc(\theta)}{sinc^2\left(\frac{\theta}{2}\right)}\right)[\mathbf{u}]_\times^2 . \tag{4.38}$$

Based on the open-loop dynamics of (4.37) and the ensuing stability analysis, one can design a nominal feedback controller $\boldsymbol{\tau}_c$ in the following manner

$$\boldsymbol{\tau}_c = \mathbf{L}_\tau^i \mathbf{K}_{\boldsymbol{\tau}} \mathbf{z} \tag{4.39}$$

where $\mathbf{z} \triangleq \begin{bmatrix} \mathbf{e}_v^T & \mathbf{e}_\omega^T \end{bmatrix}^T \in \mathbb{R}^{6\times 1}$ is the composite error signal while $\mathbf{L}_\tau^i$ and $\mathbf{K}_{\boldsymbol{\tau}}$ can be defined as follows

$$\mathbf{L}_\tau^i = \begin{bmatrix} -\gamma_2^i (\mathbf{L}_v^i)^{-1} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & -\mathbf{I}_{3\times 3} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \quad \mathbf{K}_{\boldsymbol{\tau}} = \begin{bmatrix} \mathbf{K}_v & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{K}_\omega \end{bmatrix} \in \mathbb{R}^{6\times 6}$$

where $\mathbf{K}_v \in \mathbb{R}^{3\times 3}$ and $\mathbf{K}_\omega \in \mathbb{R}^{3\times 3}$ are constant, diagonal, positive gain matrices for which the bounds can be chosen to satisfy certain stability and performance criteria.

Motivated by the desire to dynamically modify the size of the control inputs based on sensor and actuator constraints, one can define diagonal weight matrices $\mathbf{W}_v \in \mathbb{R}^{3\times 3}$ and $\mathbf{W}_\omega \in \mathbb{R}^{3\times 3}$ and utilize those in order to obtain a modified controller as follows

$$\boldsymbol{\tau}_c = \mathbf{L}_\tau^i \mathbf{K}_{\boldsymbol{\tau}} \mathbf{W}_{\mathbf{c}} \mathbf{z}, \quad \mathbf{W}_{\mathbf{c}} = \begin{bmatrix} \mathbf{W}_v & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{W}_\omega \end{bmatrix} \in \mathbb{R}^{\mathbf{6\times 6}}. \tag{4.40}$$

After substituting (4.40) into (4.37), the closed-loop dynamics for the position and orientation errors can be obtained as follows

$$\dot{\mathbf{e}}_v = -\frac{1}{d^*}\mathbf{K}_v \mathbf{W}_v \mathbf{e}_v - \mathbf{L}_c^i \mathbf{K}_\omega \mathbf{W}_\omega \mathbf{e}_\omega, \quad \dot{\mathbf{e}}_\omega = -\mathbf{L}_\omega \mathbf{K}_\omega \mathbf{W}_\omega \mathbf{e}_\omega. \tag{4.41}$$

To facilitate further analysis, we define $\mathbf{w}_c \in \mathbb{R}^{6\times 1}$ as an auxiliary gain vector consisting of the diagonal entries of $\mathbf{W}_c$ such that $\mathbf{W}_c = diag\{\mathbf{w}_c\} \in \mathbb{R}^{6\times 6}$. Furthermore, we define $\mathbf{Z} = diag\{\mathbf{z}\} \in \mathbb{R}^{6\times 6}$ such that $\mathbf{W}_c\mathbf{z} = \mathbf{Z}\mathbf{w}_c$. Based upon these definitions as well as the controller definition given in (4.40), the closed-loop dynamics of the extended image coordinates for the k^{th} feature point can be succinctly obtained as follows

$$\dot{\mathbf{m}}_{e,k} = \begin{bmatrix} \frac{1}{\gamma_2^k d^*}\mathbf{L}_v^k & \mathbf{L}_c^k \end{bmatrix} \mathbf{L}_\tau^i \mathbf{K}_\tau \mathbf{Z}\mathbf{w_c} = \mathbf{A}_{m,k}\mathbf{w}_c \tag{4.42}$$

where we have utilized (4.32) and the fact that $Z_k = \gamma_2^k d^*$. In (4.42) above, $\mathbf{A}_{m,k} \triangleq \begin{bmatrix} \frac{1}{\gamma_2^k d^*}\mathbf{L}_v^k & \mathbf{L}_c^k \end{bmatrix} \mathbf{L}_\tau^i \mathbf{K}_\tau \mathbf{Z} \in \mathbb{R}^{3\times 6}$ is considered to be measurable; specifically, it is assumed that the constant d^* can be computed apriori offline or estimated online using standard depth estimation techniques, *e.g.,* [50]. Given a feature point $\mathbf{m}_{e,k}$ that lies on the j^{th} edge of the sensor boundary represented by a normal $\mathbf{n}_j$ (positive pointing outwards), one can utilize the dynamics of (4.42) to obtain the following dynamic constraint for the feature staying within the boundary

$$\mathbf{n}_j^T \dot{\mathbf{m}}_{e,k} \leq 0 \Rightarrow \mathbf{n}_j^T \mathbf{A}_{m,k}\mathbf{w}_c \leq 0 \tag{4.43}$$

where $\mathbf{n}_j \in \mathbb{R}^{3\times 1}$ is a normal vector in homogeneous coordinates whose third element is always 0 since we are only interested in the dynamics of the first two components of $\mathbf{m}_{e,k}$. Given a set of N_f such feature points lying on sensor boundaries, the constraint represented by (4.43) can be stacked for each of these feature points to obtain a set of inequality constraints as follows

$$\mathbf{B}\mathbf{w}_c \leq \mathbf{0}_{N_f\times 1} \tag{4.44}$$

where $\mathbf{B} \in \mathbb{R}^{N_f \times 6}$. Motivated by the ensuing stability analysis and the desire to introduce actuation constraints, one can bound the auxiliary gain vector $\mathbf{w}_c$ as follows

$$\boldsymbol{\varepsilon}_u \geq \mathbf{w}_c \geq \boldsymbol{\varepsilon}_l \tag{4.45}$$

where $\boldsymbol{\varepsilon}_u$, $\boldsymbol{\varepsilon}_l \in \mathbb{R}^{6\times 1}$ represent constant upper and lower bounds for the auxiliary gain vector $\mathbf{w}_c$. Furthermore, based on our desire to follow the nominal controller as allowed by the constraints (4.44) and (4.45), we define a cost function as follows

$$J \triangleq \frac{1}{2}\left(\mathbf{w}_c - \mathbf{I}_{6\times 1}\right)^T \left(\mathbf{w}_c - \mathbf{I}_{6\times 1}\right). \tag{4.46}$$

Given the constraints and the cost function represented by (4.44), (4.45), and (4.46) above, the weight optimization problem can be expressed in a standard quadratic programming form as follows

$$\min \frac{1}{2}\mathbf{x}^T\mathbf{x}, \quad \text{s.t.} \begin{cases} \mathbf{B}\mathbf{x} \leq \mathbf{b}, \\ \\ \mathbf{u}_b \geq \mathbf{x} \geq \mathbf{l}_b. \end{cases} \tag{4.47}$$

where $\mathbf{x} \triangleq \mathbf{w}_c - \mathbf{I}_{6\times 1}$, $\mathbf{b} \triangleq -\mathbf{B} \cdot \mathbf{I}_{6\times 1}$, $\mathbf{u}_b = \boldsymbol{\varepsilon}_u - \mathbf{I}_{6\times 1}$, and $\mathbf{l}_b = \boldsymbol{\varepsilon}_l - \mathbf{I}_{6\times 1}$.

4.2.3 Stability Analysis

Theorem 1 *Given the existence of a solution for the quadratic programming problem in (4.47), the closed-loop error system described by (4.41) is semi-globally asymptotically (exponentially) stable under unconstrained actuation while the result is local asymptotic (exponential) stability under constrained actuation.*

Proof. Let $V(t, \mathbf{z})$ denote a non-negative Lyapunov candidate function defined as

$$V(t, \mathbf{z}) = \frac{1}{2}\mathbf{e}_v^T\mathbf{e}_v + \frac{1}{2}\mathbf{e}_\omega^T\mathbf{e}_\omega = \frac{1}{2}\|\mathbf{z}\|^2 \tag{4.48}$$

where $\mathbf{z}$ has been previously defined as $\mathbf{z} \triangleq \begin{bmatrix} \mathbf{e}_v^T & \mathbf{e}_\omega^T \end{bmatrix}^T$. After differentiating (4.48) along the dynamics of (4.41) and applying the fact that $\mathbf{e}_\omega^T\mathbf{L}_\omega = \mathbf{e}_\omega^T$, one can obtain

$$\dot{V} = -\mathbf{e}_\omega^T\mathbf{K}_\omega\mathbf{W}_\omega\mathbf{e}_\omega - \mathbf{e}_v^T\mathbf{L}_c^i\mathbf{K}_\omega\mathbf{W}_\omega\mathbf{e}_\omega - \frac{\mathbf{e}_v^T\mathbf{K}_v\mathbf{W}_v\mathbf{e}_v}{d^*}. \tag{4.49}$$

After defining $\mathbf{L}_c^{i*} = \mathbf{L}_c^i\left(m_{e,i,1}^*, m_{e,i,2}^*\right)$, it is easy to see that

$$\left\|\mathbf{L}_c^i - \mathbf{L}_c^{i*}\right\| \leq \rho\left(\|\mathbf{e}_v\|\right) \tag{4.50}$$

where $\rho\left(\|\mathbf{e}_v\|\right)$ denotes a globally invertible, nondecreasing function. By applying the inequality in (4.50), $\dot{V}(t, \mathbf{z})$ can be upperbounded in the following manner

$$\begin{aligned} \dot{V} \leq \; & -\underline{K}_\omega\underline{W}_\omega\|\mathbf{e}_\omega\|^2 + \overline{K}_\omega\overline{W}_\omega\|\mathbf{L}_c^{i*}\|_\infty\|\mathbf{e}_v\|\|\mathbf{e}_\omega\| \\ & -\frac{1}{d^*}\underline{K}_v\underline{W}_v\|\mathbf{e}_v\|^2 + \overline{K}_\omega\overline{W}_\omega\rho\left(\|\mathbf{e}_v\|\right)\|\mathbf{e}_v\|\|\mathbf{e}_\omega\| \end{aligned} \tag{4.51}$$

where $\underline{W}_v$, $\underline{W}_\omega$, $\overline{W}_\omega$, $\underline{K}_v$, $\underline{K}_\omega$, and $\overline{K}_\omega$ are all constant scalars with following definitions: $\underline{W}_v = \lambda_{\min}\{\mathbf{W}_v\}$, $\underline{W}_\omega = \lambda_{\min}\{\mathbf{W}_\omega\}$, $\overline{W}_\omega = \lambda_{\max}\{\mathbf{W}_\omega\}$, $\underline{K}_v = \lambda_{\min}\{\mathbf{K}_v\}$, $\underline{K}_\omega = \lambda_{\min}\{\mathbf{K}_\omega\}$, and $\overline{K}_\omega = \lambda_{\max}\{\mathbf{K}_\omega\}$. Here we note that $\lambda_{\min}\{\cdot\}$ and $\lambda_{\max}\{\cdot\}$ denote the minimum and maximum eigenvalues for a given matrix. By using the fact that $\|a\|\|b\| \leq \|a\|^2 + \|b\|^2$, one can further upperbound (4.51) as follows

$$\dot{V} \leq -\delta_1\|\mathbf{e}_w\|^2 - \delta_2\|\mathbf{e}_v\|^2 \tag{4.52}$$

where δ_1 and δ_2 are constants of analysis that are defined as follows

$$\delta_1 \triangleq \underline{K}_\omega\underline{W}_\omega - \left(\left\|\mathbf{L}_c^{i*}\right\|_\infty^2 + 1\right), \;\; \delta_2 \triangleq \frac{1}{d^*}\underline{K}_v\underline{W}_v - \overline{K}_\omega^2\overline{W}_\omega^2\rho^2\left(\|\mathbf{e}_v\|\right) - \overline{K}_\omega^2\overline{W}_\omega^2. \tag{4.53}$$

By choosing

$$\underline{K}_{\omega} > \eta_1 \triangleq \frac{1}{\underline{W}_{\omega}} \left(\left\| \mathbf{L}_c^{i*} \right\|_{\infty}^2 + 1 \right), \tag{4.54}$$

it is easy to see that that δ_1 is positive where we have exploited the positive lower boundedness of $\mathbf{W}_{\omega}$ given in (4.45). After choosing a large enough $\mathbf{K}_{\omega}$ to satisfy (4.54), it is obvious that $\delta_2 > 0$ when the following condition is satisfied

$$\|\mathbf{e}_v\| < \eta_2 \triangleq \rho^{-1} \left(\sqrt{\frac{\frac{1}{d^*} \underline{K}_v \underline{W}_v - \overline{K}_{\omega}^2 \overline{W}_{\omega}^2}{\overline{K}_{\omega}^2 \overline{W}_{\omega}^2}} \right). \tag{4.55}$$

where $\mathbf{K}_v$ also needs to be selected large enough to ensure that

$$\underline{K}_v > \eta_3 \triangleq d^* \overline{K}_{\omega}^2 \overline{W}_{\omega}^2 / \underline{W}_v \tag{4.56}$$

such that the set $\{\mathbf{e}_v \mid \|\mathbf{e}_v\| < \eta_2\}$ is non-empty where the boundedness of the right hand side of the above expression is ensured by the bounds on the weights given in (4.45). Thus, the time derivative of $V(t, \mathbf{z})$ can be upperbounded as

$$\dot{V} \leqslant -\lambda_1 \|\mathbf{z}\|^2 \qquad \forall \ \|\mathbf{z}(0)\| < \eta_2 \tag{4.57}$$

where $\lambda_1 \triangleq \min(\delta_1, \delta_2)$ is a positive constant scalar. Furthermore, we have utilized the fact that $\|\mathbf{e}_v(0)\| < \|\mathbf{z}(0)\|$ by definition and that $\|\mathbf{z}(t)\| \leq \|\mathbf{z}(0)\|$ on account of $\dot{V}(t, \mathbf{z})$ being negative definite on the domain of interest. After exploiting the fact that $\|\mathbf{z}\|^2 = 2V$ from (4.48) and substituting in (4.57), one can solve the resulting differential inequality to obtain the following expression for $V(t, \mathbf{z})$

$$V(t, \mathbf{z}) \leqslant V(0, \mathbf{z}(0)) \exp(-2\lambda_1 t) \qquad \forall \|\mathbf{z}(0)\| < \eta_2. \tag{4.58}$$

By again utilizing the fact $\|\mathbf{z}\|^2 = 2V$ and noting that $V(t, \mathbf{z}) < V(0, \mathbf{z}(0))$ from (4.58), we can develop following upper bound for the composite vector $\mathbf{z}(t)$ as

$$\|\mathbf{z}(t)\| \leqslant \|\mathbf{z}(0)\| \exp(-\lambda_1 t) \qquad \forall \|\mathbf{z}(0)\| < \eta_2. \tag{4.59}$$

In (4.59), since η_2 can be made as large as possible by choosing $\mathbf{K}_v$ large enough, it is clear to see that $\mathbf{z}(t)$ is semi-globally asymptotically (exponentially) stable. However, the gains cannot be made arbitrarily large when considering actuator saturation, which we model as follows

$$\|\mathbf{v}_c\| \leq v_{cb} \qquad \|\boldsymbol{\omega}_c\| \leq \omega_{cb}$$

where v_{cb} and ω_{cb} stand for the finite bounds on the norm of control input signals $\mathbf{v}_c$ and $\boldsymbol{\omega}_c$, respectively. From (4.59), we know that $\mathbf{z} \in \mathcal{L}_\infty$ which implies that $\mathbf{e}_v$, $\mathbf{e}_\omega \in \mathcal{L}_\infty$. Thus, we can utilize (4.31), (4.34), and the definition of γ_2^i and $\mathbf{L}_v^i$ to prove that γ_2^i, $(\mathbf{L}_v^i)^{-1} \in \mathcal{L}_\infty$. Assuming $\gamma_2^i \left\|(\mathbf{L}_v^i)^{-1}\right\| \leq \rho$, $\|\boldsymbol{\omega}_c\|$ and $\|\mathbf{v}_c\|$ can be upperbounded according to (4.40) as

$$\begin{aligned} \|\mathbf{v}_c\| &\leq \gamma_2^i \left\|(\mathbf{L}_v^i)^{-1}\right\| \|\mathbf{K}_v\| \|\mathbf{W}_v\| \|\mathbf{e}_v\| \leq \rho \overline{K}_v \overline{W}_v \|\mathbf{e}_v\| \\ \|\boldsymbol{\omega}_c\| &\leq \|\mathbf{K}_\omega\| \|\mathbf{W}_\omega\| \|\mathbf{e}_\omega\| \leq \overline{K}_\omega \overline{W}_\omega \|\mathbf{e}_\omega\| \end{aligned} \tag{4.60}$$

where $\overline{W}_\omega$, $\overline{K}_\omega$, $\overline{W}_v$, and $\overline{K}_v$ have been defined previously. Thus, (4.60) can be utilized to prove that the actuator bounds can be satisfied as follows

$$\|\mathbf{v}_c\| \leq v_{cb} \quad \forall \ \|\mathbf{e}_v\| < \frac{v_{cb}}{\rho \overline{K}_v \overline{W}_v}, \quad \|\boldsymbol{\omega}_c\| \leq \omega_{cb} \quad \forall \ \|\mathbf{e}_\omega\| < \frac{\omega_{cb}}{\overline{K}_\omega \overline{W}_\omega}. \tag{4.61}$$

By employing the fact that $\|\mathbf{e}_\omega\|$, $\|\mathbf{e}_v\| < \|\mathbf{z}\|$ and combining the results from (4.59) and (4.61), one can obtain the following result

$$\|\mathbf{z}(t)\| \leqslant \|\mathbf{z}(0)\| \exp(-\lambda_1 t) \quad \forall \ \|\mathbf{z}(0)\| < \min\left(\eta_2, \ \frac{v_{cb}}{\rho \overline{K}_v \overline{W}_v}, \ \frac{\omega_{cb}}{\overline{K}_\omega \overline{W}_\omega}\right) \tag{4.62}$$

which proves the local asymptotic (exponential) stability result for $\mathbf{z}$ under finite actuation $\forall\ t \geq 0$. It is clear to see that (4.54), (4.55), and (4.61) offer prescriptions for choosing the bounds on the the auxiliary gain vector $\mathbf{w}_c$. ■

Remark 1 *The above stability analysis for the modified control law is also valid under nominal control by setting* $\mathbf{W}_v = \mathbf{W}_\omega = \mathbf{I}_{3\times 3}$*, i.e., in the absence of constraints, the semi-global stability result stated in Theorem 1 holds under nominal control by choosing the control gains to have bounds as follows*

$$\underline{K}_\omega > \bar{\eta}_1 \triangleq \left(\left\| \mathbf{L}_c^{i*} \right\|_\infty^2 + 1 \right), \quad \underline{K}_v > \bar{\eta}_3 \triangleq d^* \overline{K}_\omega^2. \tag{4.63}$$

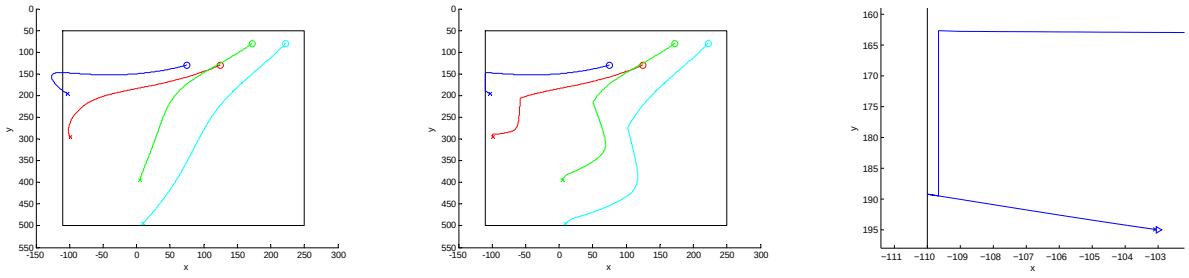
4.2.4 Results

In this section, the proposed control scheme was implemented using Simulink module in MATLAB^{TM} 7.5. A simple target object containing four coplanar feature points located on it was considered in the simulations while $d^* = 2$ was assumed to be known. The control gains $\mathbf{K}_v$ and $\mathbf{K}_\omega$ in (4.39) and (4.40) were chosen to be as follows

$$\mathbf{K}_v = diag\{0.5, 0.5, 0.5\}, \quad \mathbf{K}_\omega = diag\{0.5, 0.5, 0.5\}.$$

The lower and upper bounds $\boldsymbol{\varepsilon}_l$ and $\boldsymbol{\varepsilon}_u$ of the variable $\mathbf{w}_c$ were selected as follows

$$\boldsymbol{\varepsilon}_l = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix}^T, \quad \boldsymbol{\varepsilon}_u = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T.$$



(a) (b) (c)

Figure 4.11: Comparison between Nominal and Modified control laws during simulation

Fig. 4.11 depicts comparison between the nominal and modified control strategies in simulation. From Fig. 4.11-(a), it is clear to see that the nominal feedback controller in (4.39) was unable to keep all the feature points within the sensor boundary. On the other hand, from Fig. 4.11-(b) it is straightforward to see that the modified controller could ensure the visibility of the feature points by modifying the translation and rotation commands such that the critical feature point on the boundary was restricted to move along the boundary until such time as it was able to be driven to its setpoint (see Fig. 4.11-(c) for zoomed-in view of trajectory of critical feature point.).

A real-time implementation of the proposed control algorithm was also tested on the UCF-MANUS assistive robotic experimental testbed [51]. For obtaining experimental data, a stream of 640×480 images was captured by using a Point-Grey Dragonfly2 camera mounted on the gripper of the robot manipulator. Both the nominal and modified control algorithms were implemented in an Intel Core2 Q9450 PC with 2.75GB RAM running Windows XP operating system. In a discrete experimental implementation, it was imprudent to wait for

the feature points to reach the sensor boundary before applying the modified controller – the alternative chosen was to predict the one-step ahead position of any feature point given the 2D feature point dynamics in (4.32) and the nominal control signal given by (4.39). Thus, feature points prone to exiting were placed on the constraint stack. Furthermore, in order to account for the uncertainty in depth estimation (in this experiment, $d^* = 100$ [mm] was a coarse estimation used in the control design) as well as the imprecise prediction of feature point locations due to uncertainty in the robot model and actuator dynamics, a virtual FOV was created inside the actual FOV and was utilized in the optimization problem formulated in (4.47). The space between the virtual FOV and actual FOV served as a buffer which was needed in view of the aforementioned uncertainties as well as the fact that the implementation for the vision based feedback system had a slow bandwidth (approximately 12.5 Hz) as compared to the low-level robot controller. Thus, this buffer circumvented a practical problem of losing feature points over the implementation cycle because the feature points that escaped the virtual FOV were still visible in the real FOV and could almost always be retrieved by application of the proposed modified controller. In this experiment, the size of virtual FOV was set at 480×360. Finally, the right hand sides of (4.43) and (4.44) were chosen as a small negative number in the implementation instead of zero. Practically, this means that feature points were not allowed to travel along the sensor boundary; instead, the optimization caused them to be mildly pushed back inside once they were close to the boundary – this was seen to lead to better outcomes for retaining the feature points within the virtual FOV. It is to be noted that such a choice does not detract from the stability

analysis. The control gains $\mathbf{K}_v$ and $\mathbf{K}_\omega$ in (4.39) and (4.40) were chosen as follows

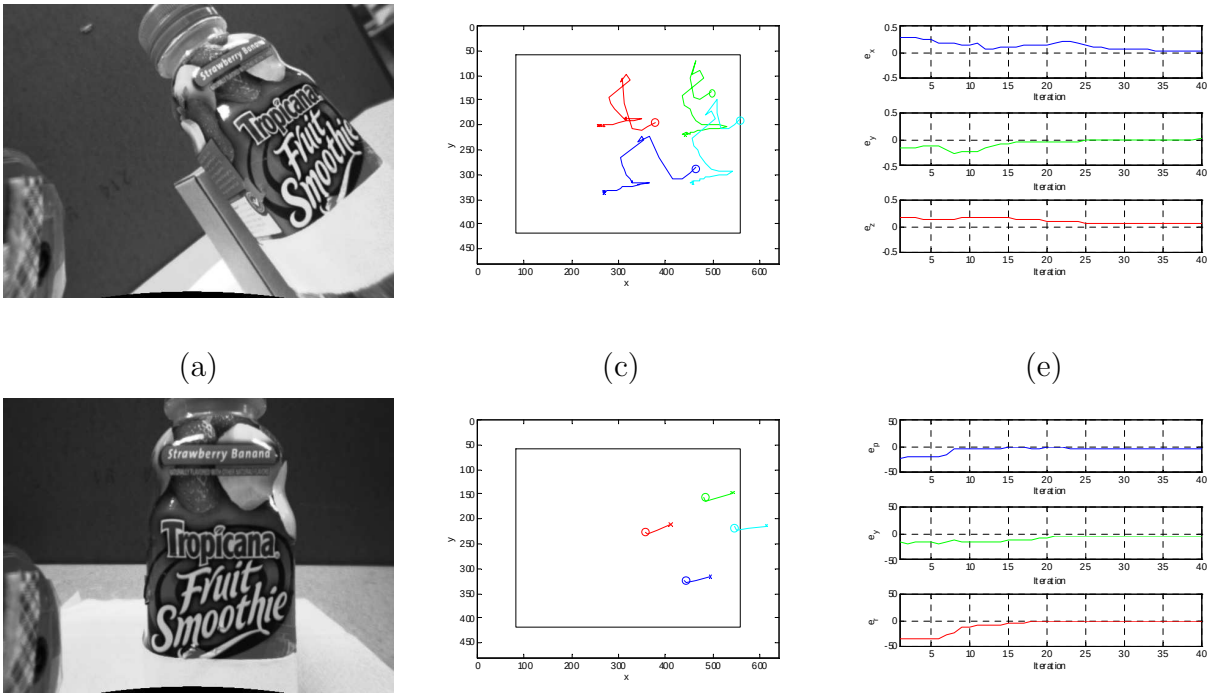
$$\mathbf{K}_v = diag\{25,\, 25,\, 25\}, \quad \mathbf{K}_\omega = diag\{0.15,\ 0.15,\ 0.15\}$$

while the lower and upper bounds $\boldsymbol{\varepsilon}_l$ and $\boldsymbol{\varepsilon}_u$ for the variable $\mathbf{w}_c$ were selected as follows

$$\boldsymbol{\varepsilon}_l = \begin{bmatrix} 0.1 & 0.1 & 0.1 & 0.02 & 0.02 & 0.02 \end{bmatrix}^T, \quad \boldsymbol{\varepsilon}_u = \begin{bmatrix} 1 & 1 & 1 & 0.8 & 0.8 & 0.8 \end{bmatrix}^T.$$

Motivated by [40], the proposed control algorithm was examined under two difficult tasks. The first task involved both translational and rotational movements in all three axes. The initial and goal frames are shown in Figs. 4.12-(a) and (b), respectively. The pixel-space trajectories for 4 feature points under the modified and nominal control laws are presented and compared in Figs. 4.12-(c) and (d), respectively. From the trajectories in Fig. 4.12-(c), it's straightforward to see that the modified control not only drove the feature points to the desired position but also kept all feature points within the virtual FOV. However, under nominal control, one feature point was driven out of actual FOV – Fig. 4.12-(d) shows the pixel trajectories up to one frame prior to controller failure due to the feature point exiting the FOV. The convergence of both translation and rotation errors, respectively, in Figs. 4.12-(e) and (f) clearly indicates the efficacy of the modified control law. In Fig. 4.12-(e), the y-axis was dimensionless while it has the unit of degrees in Fig. 4.12-(f). Note that the control objective was deemed to be achieved when all the error signals were within their preset thresholds.

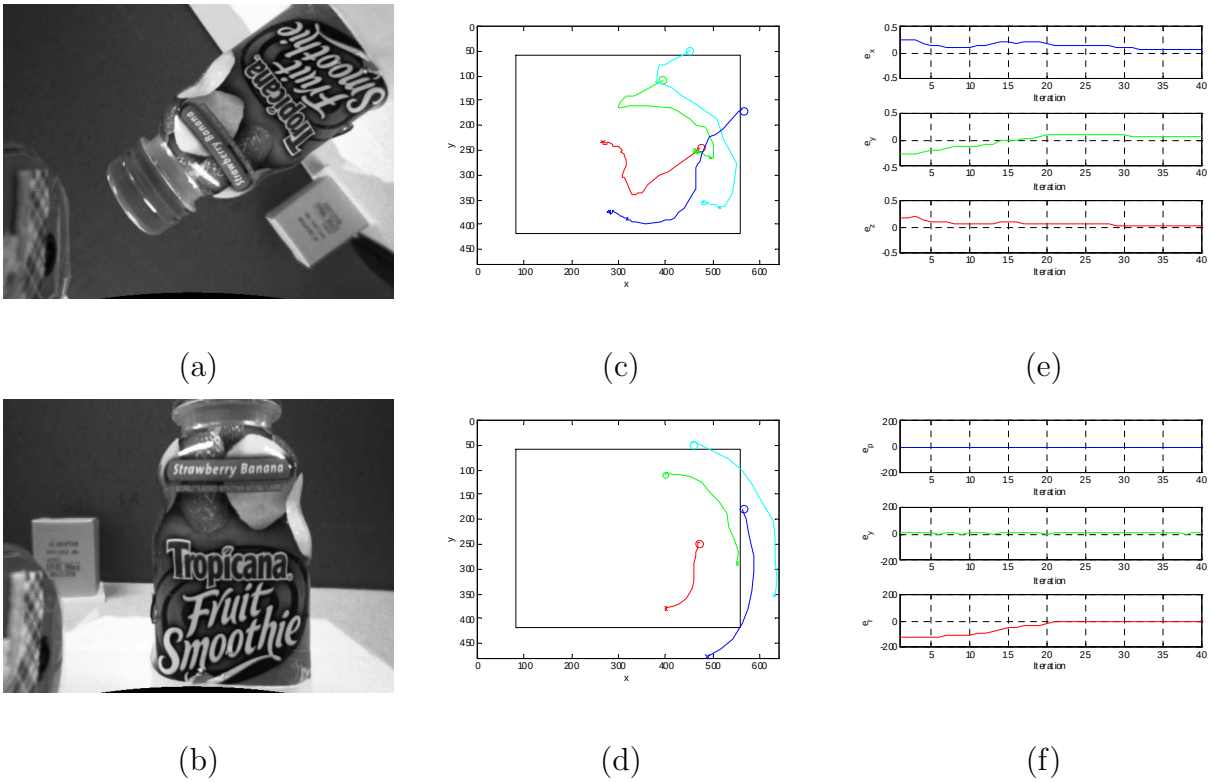
An example of large rotation about the optical axis is represented in Fig. 4.13, where the initial and goal frames are shown in Figs. 4.13-(a) and (b), respectively. The trajectories



(a) (c) (e)

(b) (d) (f)

Figure 4.12: Experimental Results for Large General Motion.



(a) (c) (e)

(b) (d) (f)

Figure 4.13: Experimental Results for Large Rotation Motion.

under the modified and nominal control laws are presented and compared in Figs. 4.13-(c) and (d), respectively. From the trajectories in Fig. 4.13-(c), one can easily see that under the modified control, all feature points were kept within the virtual FOV although two of them were initially located outside it. However, the nominal controller failed to keep the feature points within the FOV due to large rotation movement resulting in controller failure. The results shown in Figs. 4.13-(e) and (f) clearly indicates the convergence of the translation and rotation error signals, respectively, under the modified control law.

CHAPTER 5

CONCLUSIONS

In this work, we have proposed a two-level particle filter algorithm to solve the range estimation for a single camera system. Compared with particle filter approach in [18] and other observer methods, our proposed method has a faster convergence rate with smaller number of particles. Simulation and experimental results strongly suggest that our proposed algorithm can estimate the range information more precisely and converge well in the presence of non-trivial image measurement error and kinematic uncertainties in the motion parameters.

In this work, we have proposed a visual servoing control law that utilizes a Lyapunov based design method and constrained optimization to drive the robot pose to a setpoint while retaining the target features in the image FOV. Motivated by the assistive robotic application at hand as well as its applicability to other robot applications, both sensor and actuator constraints have been considered. When constraints related only to the limited size of the in-hand camera FOV were considered, the proposed control law yielded a semi-global asymptotic stability result from Lyapunov analysis. However, when actuator constraints were added, the result was local asymptotic stability with known bounds on the region of attraction. The results obtained from simulation in MATLAB and experiments conducted

on the UCF-MANUS assistive robotic testbed illustrate the efficacy of the proposed visual servoing control strategy.

LIST OF REFERENCES

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.,* vol. 12, no. 5, 1996, pp. 651–670.

[2] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Trans. Robot. Automat.,* vol. 15, no. 2, 1999, pp. 79-97.

[3] L. Matthies, R. Szeliski, and T. Kanade, "Kalman filter-based algorithms for estimating depth from image sequences," *Int. J. Computer Vision,* vol. 3, 1989, pp. 209-236.

[4] B. Sridhar and A. Phatak, "Analysis of image-based navigation system for rotorcraft low-altitude flight," *IEEE Trans. Systems Man Cybernet.,* vol. 22, no. 2, 1992, pp. 290-299.

[5] B. Sridhar, R. Soursa and B. Hussein, "Passive range estimation for rotorcraft low altitude flight," *Int. J. Machine Vision Appl.,* vol. 6, no. 1, 1993, pp. 10-24.

[6] B. Sridhar, R. Soursa, P. Smith and B. Hussein, "Vision-based obstacle detection for rotorcraft flight," *J. Robotic Systems,* vol. 9, no. 6, 1992, pp. 709-727.

[7] Y. S. Hung and H. T. Ho, "A Kalman filter approach to direct depth estimation incorporating surface structure," *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 21, no. 6, 1999, pp. 570–575.

[8] M. Jankovic and B. Ghosh, "Visually guided ranging from observations of points, lines and curves via an identifier based nonlinear observer," *Syst. and Control Lett.,* vol. 25, no. 1, 1995, pp. 63–73.

[9] X. Chen and H. Kano, "A new state observer for perspective systems," *IEEE Trans. Automat. Contr.,* vol. 47, no. 4, 2002, pp. 658–663.

[10] W. Dixon, Y. Fang, D. Dawson, and T. Flynn, "Range identification for perspective vision systems," *IEEE Trans. Automat. Contr.,* vol. 48, no. 12, 2003, pp. 2232–2238.

[11] D. Karagiannis and A. Astolfi, "A new solution to the problem of range identification in perspective vision systems," *IEEE Trans. Automat. Contr.,* vol. 50, no. 12, 2005, pp. 2074–2077.

[12] A. Davison, "Real-time simultaneous localization and mapping with a single camera," in *Proc. IEEE Int. Conf. on Computer Vision,* 2003, pp. 1403-1410.

[13] J. Civera, A. Davison, and J. Montiel, "Inverse depth to depth conversion for monocular SLAM," in *Proc. IEEE Int. Conf. Robot. Auto.,* 2007, pp. 2778-2783.

[14] J. Montiel, J. Civera, and A. Davison, "Unified inverse depth parametrization for monocular SLAM," *Robot. Sci. Syst. Conf.,* Philadelphia, PA, 2006.

[15] J. Civera, A. Davison, and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *IEEE Trans. on Robotics,* vol. 24, no. 5, 2008, pp. 932–945.

[16] M. Pupilli and A. Calway, "Real-Time Camera Tracking Using a Particle Filter," in *Proc. of the British Machine Vision Conference,* 2005, pp. 519-528.

[17] A. H. Abdul Hafez and C. V. Jawahar, "Target model estimation using particle filters for visual servoing," in *Proc. IEEE Int. Conference on Pattern Recognition,* Hong Kong, pp. 2006.

[18] S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filter for on-line nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Processing,* vol. 50, no. 2, 2001, pp. 174-188.

[19] A. Doucet, S. Godsill, C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing,* vol. 10, no. 3, 2000, pp. 197-208.

[20] A. Doucet, "On sequential simulation based methods for Bayesian filtering," Tech. Rep. CUED/F-INFENG/TR.310, 1998.

[21] C. C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, "Approximate Jacobian control for robots with uncertain kinematics and dynamics," *IEEE Trans. Robot. Automat.,* vol. 19, no. 4, 2003, pp. 692–702.

[22] A. Doucet, N. de Freitas, K. Murphy, S. Russel, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," *UAI-2000.*

[23] Z. Wang, D.-J. Kim, and A. Behal, "An Optimization-based Approach for Design and Analysis of Stable 2.5D Visual Servoing under Sensor and Actuator Constraints," *Proc. of the 2010 IEEE Conference on Decision and Control*, Atlanta, GA, pp. 1650-1655, Dec. 2010.

[24] A. H. A. Hafez, and C. V. Jawahar, "Target Model Estimation using Particle Filters for Visual Servoing," in *Proc. of the 18th Int. Conf. on Pattern Recognition*, Hong Kong, pp. 651-654, August 20-24, 2006.

[25] E. Malis and P. Rives, "Robustness of image-based visual servoing with respect to depth distribution errors," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, Taipei, Taiwan, pp. 1056-1061, September 14-19, 2003.

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

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

[28] N. J. Cowan, J. D. Weingarten, and D. E. Koditscheck, "Visual servoing via navigation functions," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 521-533, 2001.

[29] N. J. Cowan, O. Shakernia, R. Vidal, and S. Sastry, "Vision-based follow-the-leader," in *Proc. of Int. Conf. on Intelligent Robots and Systems,* pp. 1796–1801, Las Vegas, NV, October 2003.

[30] J. Chen, D. M. Dawson, W. E. Dixon, and V. K. Chitrakaran, "Navigation function-based visual servo control," *Automatica*, vol. 43, pp. 1165-1177, 2007.

[31] A. Remazeilles, N. Mansard, and F. Chaumette, "A qualitative visual servoing to ensure the visibility constraint," *in Proc. of Int. Conf. on Intelligent Robots and Systems*, Beijing, China, pp. 4297-4303, 2006.

[32] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, Washington, DC, USA, pp. 1624-1629, 2002.

[33] G. Chesi and A. Vicino, "Visual servoing for large camera displacements," *IEEE Trans. on Robotics*, vol. 20, no. 4, pp. 724-735, 2004.

[34] N. Garcia-Aracil, E. Malis, R. Aracil-Santonja, and C. Perez-Vidal, "Continuos visual servoing despite the changes of visibility in image features," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1214-1220, 2005.

[35] V. Kyrki, D. Kragic, and H. Christensen, "Path planning in image space for robust visual servoing," in *Proc. Int. Conf. Intelligent Robots and Systems*, pp. 349–354, 2004.

[36] A. H. A. Hafez, A. K. Nelakanti, and C. V. Jawahar, "Path Planning Approach to Visual Servoing with Feature Visibility Constraints: A Convex Optimization based Solution," in *Proc. of Int. Conf. on Intelligent Robots and Systems*, San Diego, CA, USA, pp 1981-1986, 2007.

[37] A. Hafez, E. Cervera, and C. Jawahar, "Hybrid visual servoing by boosting IBVS and PBVS," in *Proc. of Int. Conf. on Information and Communication Technologies*, Damascus, Syria, pp. 1-6, 2008.

[38] 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. on Robotics*, vol. 20, no. 5, pp. 908-913, 2004.

[39] L. Deng, F. Janabi-Sharifi, and W. J. Wilson, "Hybrid Motion Control and Planning Strategies for Visual Servoing," *IEEE Trans. on Industrial Electronics*, vol. 52, no. 4, pp. 1024-1040, August 2005.

[40] N. Gans and S. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Trans. on Robotics*, vol. 23, no. 3, pp. 530-540, 2007.

[41] F. Chaumette, E. Malis, and S. Boudet, "2D 1/2 visual servoing with respect to a planar object," In *Proc. of the workshop on new trends in image-based robot servoing, IROS'97,* Grenoble, France, pp. 45–52, Sep.1997.

[42] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D Visual Servoing," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 238-250, Apr. 1999.

[43] Y. Fang, A. Behal, W. E. Dixon, and D. M. Dawson, "Adaptive 2.5D Visual Servoing of Kinematically Redundant Robot Manipulators," in *Proc. of the IEEE Conference on Decision and Control*, Las Vegas, NV, pp. 2860-2865, Dec. 2002.

[44] 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. on Control Systems Technology*, vol. 13, no. 5, pp. 814-825, 2005.

[45] G. Hu, W. Mackunis, N. Gans, W. E. Dixon, J. Chen, A. Behal, and D. M. Dawson, "Homography-based visual servo control with imperfect camera calibration," *IEEE Trans. Autom.Control*, vol. 54, no. 6, pp. 1318–1324, Jun. 2009.

[46] D.-J. Kim, R. Lovelett, Z. Wang, and A. Behal, "A Region-Based Switching Scheme for Practical Visual Servoing under Limited FOV and Dynamically Changing Features," *in Proc. of 2009 IASTED 14th Int. Conf. on Robotics and Applications*, Cambridge, MA, USA, 2009.

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

[48] E. Malis, "Contributions à la modélisation et à la commande en asservissement visuel," Ph.D. Dissertation, University of Rennes I, IRISA, France, Nov. 1998.

[49] R. A. Spurrier, "Comment on Singularity-Free Extraction of a Quaternion from a Direction-Cosine Matrix", *Journal Spacecrafts & Rockets*, vol. 15, pp. 255, 1978.

[50] W. E. Dixon, Y. Fang, D. M. Dawson, and T. J. Flynn, "Range identification for perspective vision systems," *IEEE Transactions on Automatic Control,* vol. 48, no. 12, pp. 2232-2238, December 2003.

[51] D.-J. Kim, R. Lovelett, and A. Behal, "Eye-in-hand stereo visual servoing of an assistive robot arm in unstructured environments," *in Proc. of IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, pp. 2326-2331, 2009.