Recovering Scale in Relative Pose and Target Model Estimation Using Monocular Vision

by

Michael Tribou

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A combined relative pose and target object model estimation framework using a monocular camera as the primary feedback sensor has been designed and validated in a simulated robotic environment. The monocular camera is mounted on the end-effector of a robot manipulator and measures the image plane coordinates of a set of point features on a target workpiece object. Using this information, the relative position and orientation, as well as the geometry, of the target object are recovered recursively by a Kalman filter process. The Kalman filter facilitates the fusion of supplemental measurements from range sensors, with those gathered with the camera. This process allows the estimated system state to be accurate and recover the proper environment scale.

Current approaches in the research areas of visual servoing control and mobile robotics are studied in the case where the target object feature point geometry is well-known prior to the beginning of the estimation. In this case, only the relative pose of target object frames is estimated over a sequence of frames from a single monocular camera. An observability analysis was carried out to identify the physical configurations of camera and target object for which the relative pose cannot be recovered by measuring only the camera image plane coordinates of the object point features.

A popular extension to this is to concurrently estimate the target object model concurrently with the relative pose of the camera frame, a process known as Simultaneous Localization and Mapping (SLAM). The recursive framework was augmented to facilitate this larger estimation problem. The scale of the recovered solution is ambiguous using measurements from a single camera. A second observability analysis highlights more configurations for which the relative pose and target object model are unrecoverable from camera measurements alone. Instead, measurements which contain the global scale are required to obtain an accurate solution.

A set of additional sensors are detailed, including range finders and additional cameras. Measurement models for each are given, which facilitate the fusion of this supplemental data with the original monocular camera image measurements. A complete framework is then derived to combine a set of such sensor measurements to recover an accurate relative pose and target object model estimate.

This proposed framework is tested in a simulation environment with a virtual robot manipulator tracking a target object workpiece through a relative trajectory. All of the detailed estimation schemes are executed: the single monocular camera cases when the target object geometry are known and unknown, respectively; a two camera system in which the measurements are fused within the Kalman filter to recover the scale of the environment; a camera and point range sensor combination which provides a single range measurement at each system time step; and a laser pointer and camera hybrid which concurrently tries to measure the feature point images and a single range metric. The performance of the individual test cases are compared to determine which set of sensors is able to provide robust and reliable estimates for use in real world robotic applications.

Finally, some conclusions on the performance of the estimators are drawn and directions for future work are suggested. The camera and range finder combination is shown to accurately recover the proper scale for the estimate and warrants further investigation. Further, early results from the multiple monocular camera setup show superior performance to the other sensor combinations and interesting possibilities are available for wide field-of-view super sensors with high frame rates, built from many inexpensive devices.

Acknowledgements

I would like to thank all of the people who helped to make this work possible. To my supervisors, Professor William Wilson and Professor David Wang for their invaluable guidance and advice.

Also, to the people at the University of Waterloo, especially the members of my research group, and in particular, Professor Steven Waslander for his limitless enthusiasm.

Many thanks to my parents for their encouragement and support through this journey and to my friends and family.

Dedication

This is dedicated to my family.

Contents

List of Figures

Chapter 1

Introduction

In traditional industrial robotics, the end-effector is programmed to repeat a set of tasks defined in the global frame provided that the target object's position is accurately known in this frame. Such an operation is shown in Fig. 1.1(a). This positioning technique relies on a kinematic model of the robot, which is assumed to accurately predict the robot manipulator pose given knowledge of joint parameters and link lengths. However, if the pose of either the robot, due to imperfect modelling, or the target object, due to misalignment, are not known with sufficient accuracy, operation failure will result, as shown in Fig. 1.1(b).

For mobile robots, the position and orientation of the vehicle can be estimated based on *dead reckoning* using odometry measurements and a model of the vehicle dynamics. The new pose can be calculated from a previous pose by integrating drive motor speeds over time using the relationship between wheel rotation and robot motion. However, even small amounts of noise or unmodelled disturbances accumulate to produce large positioning errors over time. Consider a wheel slipping or sliding during a maneuver. Without an external mechanism to remove the accumulated error, or drift, the positioning operation will certainly fail.

For successful robot manipulation operations, it is not the *absolute* positions of the robot end-effector and target object which are important, but rather, the *relative* position and orientation of these two entities with respect to each other. Accordingly, sensors capable of measuring this relative pose can be mounted on the robot end-effector and the operations are carried out using direct measurements of the target object position. Many researchers have considered using monocular cameras for this task. The camera is fixed to a location on the robot end-effector, as shown in Fig. 1.1(c). Using feedback from this end-effector mounted camera to control a robotic manipulator around a desired relative position and orientation with

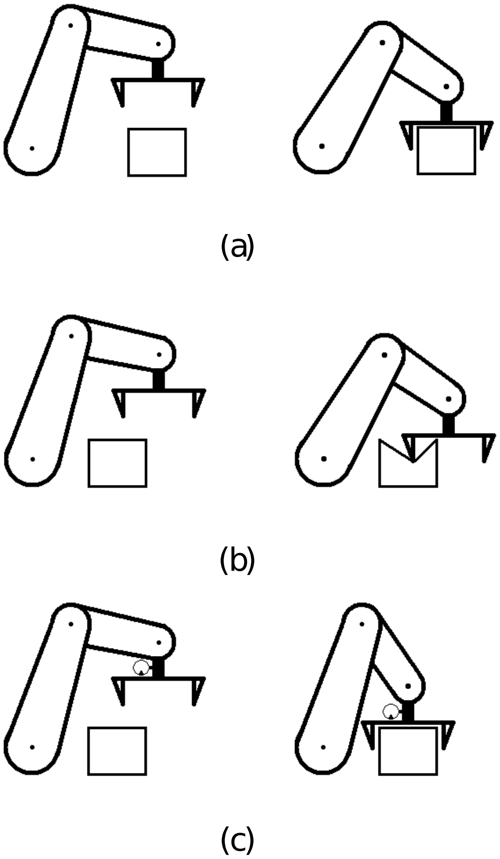


Figure 1.1: (a) Typical industrial robotic operation in which the robot attempts to pick up an object where the absolute pose of both the end-effector and the workpiece are known in the global frame. (b) If the position of the target object is not as expected, the operation will fail. (c) Mounting a camera on the end-effector, the robot measures relative pose of target object and makes the proper adjustments.

respect to a target object is known as *Visual Servoing* [7]. Consequently, robotic operations on a target workpiece can be performed when the *absolute* position and orientation of the object in a global frame are not known with confidence.

Visual servoing controllers can be classified into two broad classes, first identified by Weiss [52]: *Image-Based Visual Servoing* (IBVS) and *Position-Based Visual Servoing* (PBVS). For an overview of the two structures, with more emphasis on the IBVS methods, the reader is referred to the tutorial papers by Chaumette and Hutchinson [7] and [8]. Each approach has particular advantages and disadvantages.

The bulk of the research effort has focussed on IBVS systems, with only a handful of research groups working in the position-based visual servoing paradigm. Most of the previous research in PBVS systems has been carried out by the labs of Wilson [53], Martinet [38], and more recently, Yoon [55] and Fontanelli [23]. The work detailed in this thesis is an extension of these foundations, intended to make the position-based visual servoing control scheme more applicable to a wider array of practical applications.

1.1 Position-Based Visual Servoing

The PBVS scheme aims to track and control the relative position and orientation of a target object with respect to the robot end-effector using a fixed monocular camera [53]. The system consists of a controller and observer, which are independent, with the pose estimation component acting as the observer and providing a full state estimate as feedback to the controller. In this way, the control error signal can be formed in 3D Cartesian space.

Within the position-based visual servoing control structure, the controller relies on state estimates from the pose estimation module to regulate the robot end-effector to the desired relative configuration with respect to the target object. Traditionally, the pose estimation recursively calculates the relative position and orientation of the target object using the measurements from the monocular camera assuming that a perfect geometric model of the workpiece is available [53]. Therefore, in order for the controller to work effectively, a *known* target object model must be determined a priori. This is often cited as the principal weakness of the PBVS methods.

Some research has attempted to estimate the target object model along with the relative object pose, but with limited success [17]. The ability to estimate both the target object model and relative pose, simultaneously, would allow a position-based visual servoing system to operate effectively on a wide range of completely unknown target workpieces. The design and verification of a combined pose and target model estimation system is the focus of this thesis.

1.2 Organization

The thesis is organized in the following manner. The basic problem of using an end-effector-mounted monocular camera to estimate the relative position and orientation of a target object with known geometry is outlined in Chapter 2. In this context, the *target object model* consists of a set of point features at known locations. The parallels between the *visual servoing pose estimation* and the *mobile robot localization* problems are identified. A recursive filtering framework is outlined in which accurate, up-to-date estimates of the relative pose are recovered at each time step. Finally, an observability analysis is performed on the relative pose system to identify relative configurations of the camera and target object for which the image plane measurements do not indicate a unique relative pose.

Chapter 3 extends the problem into one of estimating both the relative pose of the target object, as well as the feature point locations on the object itself, using the single monocular camera. Current techniques from multiple research areas are outlined, and effective parameterizations from mobile robotic *Simultaneous Localization and Mapping* (SLAM) systems are adapted for use in the visual servoing system. An extended recursive framework for estimating the pose and model is detailed, and another observability analysis identifies conditions when the combined estimation is not possible. The significant result is that while such a system is able to recover a consistent solution, it is unable to recover the global scale of the environment.

Additional measurements of the target object are required to recover a solution to the combined problem which is accurate to scale. In Chapter 4, the use of supplemental sensors on the robot end-effector is considered. A framework for combining measurements from supplemental range and bearing sensors is given which will recursively fuse the data with the monocular camera measurements.

The system proposed in Chapter 4 is analyzed under a simulation environment in Chapter 5. A set of sensor combinations are tested and compared operating on a virtual target object in a robotic environment. The reliability, accuracy, and robustness of each system is evaluated.

Finally, conclusions and recommendations for future research are found in Chapter 6.

Chapter 2

Known Object Model

The *pose estimation* component of a *position-based visual servoing controller* is designed to make direct measurements of a target object's position and orientation (collectively known as *pose*) relative to the robot end-effector frame [53]. In this way, the estimator serves as an *observer* for the visual servoing controller, which subsequently uses these estimates to drive the end-effector through some *relative* trajectory with respect to the object to complete a specified task.

In this chapter, a perfect geometric *model* of the target object is assumed to be available. An *object point feature* is a visually distinguishable entity on the tracked physical object that corresponds to a unique Cartesian position with respect to a fixed local object frame and is identifiable in a camera image. When the object model is *known*, all of the 3D coordinates of the object point features are exactly known in a common local *object coordinate frame (O)*. Further, the object is a *rigid-body* and accordingly, the locations of the point features are fixed within the O frame. A valid set of features is not limited to point features. Other viable geometric features to track include, but are not limited to, edges [55] or planar surfaces. However, for the purposes of this work, only point features will be considered for tracking.

The coordinate frames used for this estimation are shown in Fig. 2.1. The *camera coordinate frame (C)* is fixed with with respect to the camera device, which is in turn fixed with respect to the robot *end-effector coordinate frame (EE)*. A fixed homogeneous transformation [49] relating the location of a point in the C frame to the EE frame, written $\mathbf{T}_C^{EE}$, can be found through camera extrinsic calibration. As a result, this allows control actions to be performed with respect to the EE frame.

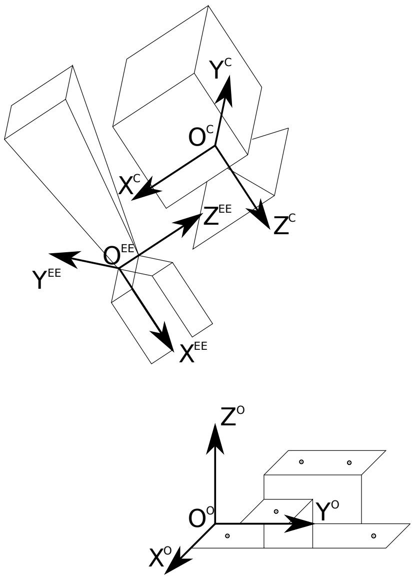


Figure 2.1: The camera frame (C) is fixed with respect to the robot end-effector frame (EE). The relative pose of the object frame (O) with respect to the C frame is the subject of the estimation.

2.1 Problem

2.1.1 Definition

A relative pose estimation system must provide accurate, real-time estimates of the relative position and orientation of the target object O frame with respect to the robot end-effector EE frame using the measurements from a set of onboard sensors. In this case, that sensor is a monocular camera mounted on the robot end-effector with an associated coordinate frame C. The relative pose estimation system must then use *image measurements* of point features on the target object and provide a solution to a set of photogrammetric equations to estimate the transformation from the O to C frames, $\mathbf{T}_O^C$. The relative pose of the target object with respect to the end-effector can then be trivially found,

$$\mathbf{T}_O^{EE} = \mathbf{T}_C^{EE}\mathbf{T}_O^C. \tag{2.1}$$

In the remainder of this chapter, the work will only consider the estimation of $\mathbf{T}_O^C$ since $\mathbf{T}_C^{EE}$ is assumed known. The three-dimensional transformation, $\mathbf{T}_O^C$, can be uniquely parameterized by six values, (three for position, three for orientation) which together form the *relative pose vector* [53],

$$\mathbf{w} = [\ X\ Y\ Z\ \phi\ \theta\ \psi\ ]^T, \tag{2.2}$$

where X, Y, Z give the Cartesion coordinates of the O frame origin expressed in the C frame and ϕ, θ, ψ represent Euler angles for roll, pitch, and yaw, with respect to the camera frame's X^C, Y^C, and Z^C axes, respectively. The coordinate transformation for a point in the O frame to the C frame, which proceeds by first rotating the point to correspond to the object frame's relative orientation, and then translating by the relative position expressed in the C frame, can be written as a matrix operation using homogeneous coordinates,

$$\begin{bmatrix} x^C \\ y^C \\ z^C \\ 1 \end{bmatrix} = \mathbf{T}_O^C \begin{bmatrix} x^O \\ y^O \\ z^O \\ 1 \end{bmatrix}, \tag{2.3}$$

where

$$\mathbf{T}_O^C = \begin{bmatrix} & X \\ \mathbf{R}_O^C & Y \\ & Z \\ \mathbf{0} & 1 \end{bmatrix}. \tag{2.4}$$

The relative orientation of the O frame with respect to the C frame is expressed by a rotation matrix and is formed using these 3-2-1 Euler angles,

$$\mathbf{R}_O^C = \mathbf{R}_{Z^C}(\psi)\mathbf{R}_{Y^C}(\theta)\mathbf{R}_{X^C}(\phi), \tag{2.5}$$

where $\mathbf{R}_{Z^C}(\psi)$ is a rotation about the Z^C-axis of the C frame by an angle ψ, and likewise for the other two parameters. Accordingly, the full rotation matrix is,

$$\mathbf{R}_O^C = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix}, \tag{2.6}$$

where $C_\phi = \cos\phi$ and $S_\phi = \sin\phi$.

Providing accurate estimates of the relative pose vector is the principal requirement of the pose estimation scheme. The next section summarizes a set of assumptions used to make such a solution tractable for most practical applications.

2.1.2 Assumptions

In order to provide real-time relative pose estimates, some explicit assumptions about the relative pose system will be made:

Assumption 1 The relative motion between the camera and target object is smooth and can be approximated by a *constant velocity model* [53] [13]. That is, the dynamics of the relative pose vector parameters can be described at time step k by,

$$\mathbf{w}_k = \mathbf{w}_{k-1} + \delta_t \dot{\mathbf{w}}_{k-1} + \eta_k^p \tag{2.7}$$

and

$$\dot{\mathbf{w}}_k = \dot{\mathbf{w}}_{k-1} + \eta_k^v. \tag{2.8}$$

where δ_t is the sampling period of the system and η_k^p and η_k^v are vectors of zero-mean Gaussian disturbance noise with known covariance.

Assumption 2 The target object is a rigid-body and defined by set of $N \geq 3$ feature points and their Cartesian coordinates in the local object frame, $\zeta_j^O = [\ x_j^O\ y_j^O\ z_j^O\ ]^T, j = 1...N$, are perfectly known.

Assumption 3 The object features points are distinguishable in the camera images and their perspective projection can be measured, up to zero-mean Gaussian measurment noise, at each time step.

Assumption 4 Both the *disturbance noise* and *measurement noise* can be represented by vectors of zero-mean Gaussian noise with known covariance. All of the individual noise processes are assumed to be uncorrelated to other noise processes and the system states at each system time step.

These assumptions do not restrict the applicability of any solution to this problem since they are generally satisfied in practical situations. Instead, the problem presented here is general and many real-world applications would allow other constraints to be asserted which would simplify the task.

Some previously proposed solutions to the problem defined in this section are detailed next.

2.2 Common Approaches

The use of visual information for estimating the relative position and orientation of two coordinate frames where a single camera is rigidly mounted within one, has received a great deal of attention by researchers across multiple areas. The two fields comprising the focus of this investigation are position-based visual servoing where it is known as *pose estimation* and mobile robotics where it is referred to as *localization* [50]. Regardless of which perspective is used, the problem is essentially one of an optimization over the state space to find the set of relative position and orientation parameters which provide the best match for a set of perspective measurements. The specific algorithms used to solve this optimization vary from one-shot batch optimizations to recursive filters. For real-time operation, filter-based recursive estimations have received the most attention since a 'near optimal' estimate can be calculated quickly using all of the available measurement data up to and including the current time step [53].

Within position-based visual servoing, the pose estimation component acts as an observer, providing the robot controller with a full-state estimate of the relative position and orientation of the target object with respect to the robot end-effector coordinate frame. The target object is free to move during the pose estimation with position-based visual servoing. Accordingly, any odometric data from the joint encoders on the robot only describe half of the relative motion to be estimated and therefore, are usually not included in the measurement space of the estimation [38] [53] [39] [28]. Instead, only the image plane measurements from a single monocular camera mounted on the robot manipulator are used to estimate the relative motion. This configuration is commonly known as *eye-in-hand*.

The definitive work on position-based visual servoing control is that of Wilson *et al.* [53]. Position and orientation (POSE) estimates of a target object with a perfectly known geometric model are recursively calculated using an Extended Kalman Filter [31] (EKF) on measurements of object point features from a monocular camera mounted on the robot end-effector. These POSE estimates are then used by the visual servoing controller to move the robot to, and remain at, the commanded relative position and orientation. The EKF framework allows for the fusion of any number of point feature image plane measurements, including properly coping with occluded features. The demonstrated system is able to track fast relative motions by operating at a 60 Hz sampling rate on the limited computing hardware of the time. A restriction of this method is the requirement for a perfect object model of the tracked workpiece. This work provides the base framework on which this study builds and will be investigated in detail starting in the next section.

Martinet *et al.* have also proposed several position-based visual servoing control schemes using pose estimates of a target object with a known model [38] [39]. The pose estimation is performed by extracting image primitives (points, lines, etc.) and using the known object model to iteratively find the object rotation and translation such that the reprojections of these features on the model best match those measured in the image. This is done by applying the DeMenthon algorithm from [16] which can determine the pose of a known object from a single image. The algorithm works by iteratively solving a set of approximate linear equations for a given camera image. The assumption is made that at least four coplanar point features are available in each image and the correspondence between these points and the target model point features is known.

Hulls *et al.* [28] build on the work of Wilson *et al.* [53] and examine the fusion of end-effector mounted camera and laser range measurements to improve the robustness and reliability of the pose estimation for tracking known object models. It is assumed that the complete surface geometry of the target object is available and, as a result, a measurement model for the laser range sensor is produced using ray-tracing techniques. With this new measurement model, the EKF was used to properly combine the two measurement sets and resulted in improved relative pose estimates of the target object. An improvement in the depth estimate of the object was obtained, which is typically a parameter with low sensitivity when using only a monocular camera [53].

Recently, Yoon *et al.* [55] recreated the work of Wilson for tracking the relative position and orientation of a target object with a single camera using the EKF for estimation. The target object model consists of edges defined by two endpoints. While the pose estimation component is similar to previous work, the feature extraction phase has been made more robust by actively searching for, and verifying, object model feature-to-measurement correspondence.

Mobile robot localization using camera sensors is a slightly simpler problem than position-based visual servo pose estimation since the robot is assumed to be moving through a static environment consisting of landmarks at known locations and the estimation is usually performed in two dimensions. As a result, the motion of the robot and camera within the world is completely described by the planar robot dynamics. This allows for a dynamic model of the mobile robot to be used along with odometry measurements from wheel encoders [4] [46] to supplement the measurement space for the estimation.

Several researchers have investigated the localization problem for mobile robots in structured environments where it is assumed that a map of the world is available. A survey of many such techniques can be found in [18]. Where position-based visual servo pose estimation tracks the motion of the target object in the robot end-effector frame, mobile robot localization tracks the inverse – the robot motion in the world frame. While it is a subtle difference, it has implications for the interpretation of the recovered parameters and the linearity of the measurement model [40]. Additionally, *features* in the visual servoing context are often referred to as *landmarks* in the mobile robotics literature [19]. They will be used interchangeably in this work.

Baumgartner *et al.* detail a 2D localization system using a robot-mounted camera to measure the bearing to point features in the environment at known locations [4]. The robotic vehicle is assumed to move in the plane and therefore only the X-Y positions and one angle θ need to be tracked to represent the robot pose at each time step. The estimation system is formulated similar to the recursive pose estimation systems and also utilizes the EKF to track the robot position. However, the robot is assumed to have a known dynamic model and provides odometry data for dead-reckoning to augment the vision-based measurements.

Mobile robots capable of visually localizing their position and orientation within a known environment have also been augmented with simple range finding devices and information from both sensors is used to help with localization and navigation tasks. In [46], ultrasonic sensors are mounted on the front of a mobile robot alongside a single camera capable of localizing the robot within the environment. While the supplementary sensors are not used directly in the localization estimates, they serve to aid in avoiding obstacles in the environment. Similar to the previous system, the estimation is performed by an EKF and the odometry measurements are used to further supplement the visual information.

If multiple robots can be used in an application, the sensory readings from each can be fused in a decentralized estimation to localize each robot with respect to some known landmarks. Nakumura *et al.* use many soccer-playing robots, each carrying a camera, and the measured azimuth angles between them as well as a few fixed landmarks (goal posts), to calculate the absolute location of each robot [44]. The geometric contraints imposed by the relative angles to each robot are solved iteratively to try and localize the robots.

A good discussion of the localization process and the application of recursive estimation techniques for the general mobile robot case can be found in Thrun *et al.* [50].

2.3 Recursive Estimation

The EKF is used extensively to provide recursive estimates from noisy measurements of the outputs of nonlinear pose estimation systems [9] [13] [17] [28] [43] [53]. In order to deploy the EKF framework, the noise processes acting on the system must be (approximately) that of zero-mean Gaussian noise. This is an assumption in the context of position-based visual servoing pose estimation. Madhusudan [36] showed that the measurement noise associated with extracting image measurements of object point features has approximately these properties and the use of the EKF is justified in practice. The following discussion is based on the work of Wilson *et al.* [53] and will attempt to follow their notation where possible.

Consider the following nonlinear discrete-time state space system,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \eta_k, \tag{2.9}$$

$$\mathbf{z}_k = \mathbf{g}(\mathbf{x}_k) + \nu_k, \tag{2.10}$$

where $\mathbf{x}_k \in \mathbb{R}^n$ is the state vector at time step k, $\mathbf{z}_k \in \mathbb{R}^m$ is the measurement vector, $\mathbf{f}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$ is the process model, $\mathbf{g}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^m$ is the measurement model, and η_k and ν_k are vectors of zero-mean Gaussian disturbance and measurement noise, respectively.

In this description, let $\hat{\mathbf{x}}_{r,s}$ be the estimate mean of the system states at time step r using measurements up to time step s. Simlarly, $\mathbf{P}_{r,s}$ is the estimate covariance at time step r using measurements up to time step s.

After initialization, the EKF proceeds recursively in two steps: a *prediction step* using the process model of the system dynamics; and a *measurement update step* that adjusts the predicted states based on the measured outputs and relative magnitudes of the disturbance and measurement noise covariances.

In the pose estimation system, the prediction step, at time step k, uses the previous estimate to predict the system states using the process model,

$$\hat{\mathbf{x}}_{k,k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1,k-1}) \tag{2.11}$$

$$\mathbf{P}_{k,k-1} = \mathbf{F}_k \mathbf{P}_{k-1,k-1} \mathbf{F}_k^T + \mathbf{Q}_k, \tag{2.12}$$

where $\mathbf{Q}_k$ is the disturbance noise covariance at the k^{th} time step, $\mathbf{F}_k$ is the linearization, through Taylor series expansion, of the process model,

$$\mathbf{F}_k = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1,k-1}}, \tag{2.13}$$

and $\mathbf{F}_k^T$ is the matrix transpose of $\mathbf{F}_k$,

Similarly, the measurement model is linearized about the current state estimate, resulting in the *measurement Jacobian*, $\mathbf{C}_k$,

$$\mathbf{C}_k = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\hat{\mathbf{x}}_{k,k-1}}. \tag{2.14}$$

The Kalman gain, $\mathbf{K}$ is calculated using this linearized model, the previous estimate covariance, and the measurement noise covariance, $\mathbf{N}_k$,

$$\mathbf{K} = \mathbf{P}_{k,k-1}\mathbf{C}_k^T(\mathbf{N}_k + \mathbf{C}_k\mathbf{P}_{k,k-1}\mathbf{C}_k^T)^{-1}. \tag{2.15}$$

Finally, the estimates and estimate covariance are updated using this gain and the innovation of the measurements – the difference between the measured and predicted outputs,

$$\hat{\mathbf{x}}_{k,k} = \hat{\mathbf{x}}_{k,k-1} + \mathbf{K}(\mathbf{z}_k - \mathbf{g}(\hat{\mathbf{x}}_{k,k-1})) \tag{2.16}$$

$$\mathbf{P}_{k,k} = \mathbf{P}_{k,k-1} - \mathbf{K}\mathbf{C}_k\mathbf{P}_{k,k-1}. \tag{2.17}$$

These two steps are applied at each time step (i.e. each camera frame) through the relative motion sequence to provide a recursive estimate of the relative position and orientation of the target object with respect to the camera frame.

The pose estimation process proceeds at each time step k as follows:

1. Capture camera image and extract image plane measurements for each of the N features on the target object.

2. Carry out the *prediction step* of the EKF to obtain the new predicted state estimate, $\hat{\mathbf{x}}_{k,k-1}$.

3. Linearize the process and measurement models about this new state estimate to find $\mathbf{F}_k$ and $\mathbf{C}_k$.

4. Find the predicted estimate covariance, $\mathbf{P}_{k,k-1}$, using $\mathbf{F}_k$.

5. Calculate the Kalman gain, $\mathbf{K}$.

6. Use the measurement update step to find a new near-optimal state estimate, $\hat{\mathbf{x}}_{k,k}$ and $\mathbf{P}_{k,k}$, based on all of the available system information up to time step k.

7. Pass these current state estimates on to the controller.

8. Repeat recursively from step 1 using these current estimates for prediction step, ($\hat{\mathbf{x}}_{k-1,k-1} = \hat{\mathbf{x}}_{k,k}$ and $\mathbf{P}_{k-1,k-1} = \mathbf{P}_{k,k}$).

This scheme has been shown to work quite well, especially when the number of features is relatively low (5-8) and the camera frame rate is high (30 - 60Hz) [53] [12] [14]. The next sections will provide the details of the process and measurement models for this system.

2.3.1 Relative Motion Dynamics

The pose estimation scheme can be represented by a nonlinear state-space system by choosing an appropriate set of states, as well as suitable process and measurement models to represent the system dynamics and outputs. As noted previously, the relative motion dynamics were assumed to be well-approximated by the constant velocity process model. This model assumes that the velocities of the six degrees of freedom are constant between samples, and integrates them into the position states accordingly.

In order to use the constant velocity process model, the system state vector will be denoted [53],

$$\mathbf{x} = [\ X\ Y\ Z\ \phi\ \theta\ \psi\ \dot{X}\ \dot{Y}\ \dot{Z}\ \dot{\phi}\ \dot{\theta}\ \dot{\psi}\ ]^T \tag{2.18}$$

and consists of the six pose parameters plus their time-derivatives.

Accordingly, the system dynamics are then,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \eta_k, \tag{2.19}$$

for the k^{th} time step, where η_k is a vector of zero mean Gaussian disturbance noise with covariance $\mathbf{Q}$, and

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{6\times6} & \delta_t \mathbf{I}_{6\times6} \\ \mathbf{0} & \mathbf{I}_{6\times6} \end{bmatrix}, \tag{2.20}$$

where $\mathbf{I}$ is the identity matrix.

While the name of this model implies that the velocities of the system do not change, in fact, the disturbance noise acts as inputs and implicitly includes the unmodeled dynamics of the system. As a result, the system noise will drive the system relative velocities, and they will not remain strictly constant. The ability of this process model to approximate the true dynamics of a physical relative motion system depends on the magnitude of the sampling period. Decreasing the sampling period reduces the apparent change in velocity between frames and the constant velocity model provides a better approximation of the motion. This implies that using cameras with high framerates will result in better relative pose estimates.

It is also possible to represent the relative motion dynamics as a *constant acceleration* process. While this would allow for improved model prediction, the performance gains are marginal when compared to the benefits of using the simpler constant velocity model and increasing the sampling rate of the system.

2.3.2 Monocular Camera Model

In order to relate the measured outputs to the system states, an appropriate *measurement model* must be identified which maps the system states at time step k, to a set of measurements, $\mathbf{z}_k$. The measurement model is derived from the physical constraints of the sensors, and as long as the sensor model can be fully described using only the system states and deterministic inputs, measurements from any number of different sensors can be *fused* together by the EKF framework [28]. Furthermore, when the covariance of the measurement noise process of each output is known, the resulting state estimate will be properly weighted based on the relative confidence levels.

In this system, the output is measured by a single monocular camera. Accordingly, a simple pin-hole camera measurement model is detailed in the following.

A *monocular camera* is a sensor which maps 3D points onto a 2D plane called the *image plane* [35], as shown in Fig. 2.2. A 3D point $\zeta^C = [\ x^C\ y^C\ z^C\ ]^T$ is mapped to a 2D point in the image plane where the line from ζ^C through the camera's *center of projection*, intersects the image plane. This intersection is at $\begin{bmatrix} f\frac{x^C}{z^C} & f\frac{y^C}{z^C} \end{bmatrix}^T$, where f is the *focal length* of the camera sensor.

Expressed in homogeneous coordinates, this projection can be written,

$$\tilde{\zeta}^I = \mathbf{\Phi}\tilde{\zeta}^C, \tag{2.21}$$

$$\begin{bmatrix} -F_x x^C \\ -F_y y^C \\ z^C \end{bmatrix} = \begin{bmatrix} -F_x & & & 0 \\ & -F_y & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{bmatrix} x^C \\ y^C \\ z^C \\ 1 \end{bmatrix}, \tag{2.22}$$

which is mapped to the image plane coordinates, in pixels,

$$[\ x^I\ y^I\ ]^T = \begin{bmatrix} -F_x\frac{x^C}{z^C} & -F_y\frac{y^C}{z^C} \end{bmatrix}^T, \tag{2.23}$$

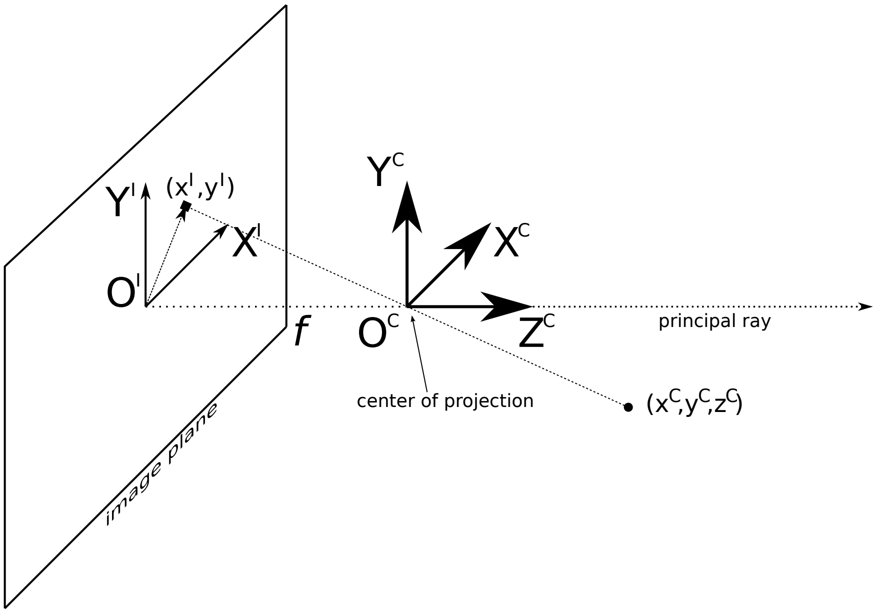


Figure 2.2: A simple pin-hole camera measurement model is used to relate the camera frame coordinates to the camera image plane coordinates for a feature point.

where $F_x = \frac{f}{P_x}$ and $F_y = \frac{f}{P_y}$ with P_x and P_y as the interpixel spacing of the camera.

The camera coordinate frame is arranged with the center of projection at the origin of the camera frame, O^C, and the *principal ray* of the camera, the ray through the camera center perpendicular to the image plane, is assumed to lie along camera's Z^C-axis.

In this work, the camera is assumed to have no skew (i.e. the image axes are perpendicular) and the *principal point*, the intersection of the principal ray and the image plane, is located at $[\ 0\ 0\ ]^T$ in the image plane.

The *measurement vector* consists of the image plane measurements at time step k of all of the visible object point features and is modelled here by,

$$\mathbf{z}_k = \mathbf{g}(\mathbf{x}_k) + \nu_k, \tag{2.24}$$

where ν_k is vector of zero mean Gaussian measurement noise with covariance $\mathbf{N}$. The output vector is composed of the projections of all of the object feature points in the camera image plane, which are calculated as a function of the state vector and the feature coordinates in the camera frame,

$$\mathbf{g}(\mathbf{x}_k) = [\ \mathbf{g}_1(\mathbf{x}_k)^T\ ...\ \mathbf{g}_j(\mathbf{x}_k)^T\ ...\ \mathbf{g}_N(\mathbf{x}_k)^T\ ]^T. \tag{2.25}$$

for all d feature points on the target object, where

$$\mathbf{g}_j(\mathbf{x}_k) = [\ x_j^I\ y_j^I\ ]^T. \tag{2.26}$$

This mapping $\mathbf{g}_j(\mathbf{x})$ is developed by transforming the known object frame coordinates of the j^{th} feature point into the camera frame,

$$\tilde{\zeta}_j^C = \mathbf{T}_O^C \tilde{\zeta}_j^O, \tag{2.27}$$

where $\mathbf{T}_O^C$ is the homogeneous transformation of the object frame with respect to the camera frame,

$$\mathbf{T}_O^C = \begin{bmatrix} \mathbf{R}_O^C & \mathbf{t}_O^C \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.28}$$

with $\mathbf{t}_O^C = [\ X\ Y\ Z\ ]^T$.

Subsequently, these camera frame coordinates are projected into the image plane, using the camera calibration matrix,

$$\begin{bmatrix} -F_x x_j^C \\ -F_y y_j^C \\ z_j^C \end{bmatrix} = \mathbf{\Phi} \mathbf{T}_O^C \tilde{\zeta}_j^O, \tag{2.29}$$

resulting in the image plane coordinates for the j^{th} feature point,

$$\mathbf{g}_j(\mathbf{x}) = [\ x_j^I\ y_j^I\ ]^T = \begin{bmatrix} -F_x \frac{x_j^C}{z_j^C} & -F_y \frac{y_j^C}{z_j^C} \end{bmatrix}^T. \tag{2.30}$$

In order to use this measurement model within the EKF measurement update step, this nonlinear function of the system states needs to be linearized about the current state estimate. The elements of the resulting measurement Jacobian can be found using the method in Appendix A.2.

Once sufficient image plane measurements are made and added to the measurement vector, the goal is to find the inverse mapping,

$$\mathbf{z} \xrightarrow{\mathbf{g}^{-1}} \mathbf{x}, \tag{2.31}$$

which will relate the measured outputs to a state vector.

2.3.3 Initialization

Due to the linearity assumptions associated with the EKF, the initialization of the estimation system plays a significant role in the performance of the relative pose tracking. The nonlinear process and measurement models dictate that the initial state estimate, $\hat{\mathbf{x}}_{0,0}$ with covariance $\mathbf{P}_{0,0}$ must define a region near the true state of the system. If this is not the case, the filter will almost certainly diverge or could converge to a suboptimal solution. However, the proximity to the true state that is required for successful tracking is not easily found and likely depends strongly on the system configuration.

Acceptable parameters of these initial estimates vary with different relative motion profiles, but can be estimated with some intuition and knowledge of the expected ranges. The 95% confidence interval of a Gaussian distribution centred about a mean, μ with a standard deviation, σ, is quantitatively expressed by $[\ \mu - 2\sigma,\ \mu + 2\sigma\ ] = [\ \mu_{min},\ \mu_{max}\ ]$. Therefore, if an approximation of this region can be found or estimated, the initial mean, μ_0, and covariance, σ_0^2, can be expressed with respect to this region,

$$\mu_0 = \frac{\mu_{min} + \mu_{max}}{2},\ \sigma_0^2 = \frac{(\mu_{max} - \mu_{min})^2}{16}. \tag{2.32}$$

An initial set of estimate parameters for all of the relative pose states and their velocities must be made and compiled into the initial state estimate mean and covariance matrix as,

$$\hat{\mathbf{x}}_{0,0} = [\ X_0\ Y_0\ Z_0\ \phi_0\ \theta_0\ \psi_0\ \dot{X}_0\ \dot{Y}_0\ \dot{Z}_0\ \dot{\phi}_0\ \dot{\theta}_0\ \dot{\psi}_0\ ]^T, \tag{2.33}$$

$$\mathbf{P}_{0,0} = \mathrm{diag}([\ \sigma^2_{X0}\ \sigma^2_{Y0}\ \sigma^2_{Z0}\ \sigma^2_{\phi 0}\ \sigma^2_{\theta 0}\ \sigma^2_{\psi 0}\ \sigma^2_{\dot{X}0}\ \sigma^2_{\dot{Y}0}\ \sigma^2_{\dot{Z}0}\ \sigma^2_{\dot{\phi}0}\ \sigma^2_{\dot{\theta}0}\ \sigma^2_{\dot{\psi}0}\ ]). \tag{2.34}$$

This initial mean and covariance are then used to seed the recursive filter and the estimation can proceed throughout the relative motion sequence.

2.4 Observability Analysis

In the previous section, the assertion was made that a certain minimum amount of information was necessary to find an inverse mapping from the measured output vector to the corresponding system states. It is important to know whether or not the system states at any time step k can be reconstructed using only the measurements up to the current time. If it is not possible to find this unique relationship, then the pose estimates may not be accurate and therefore of limited usefulness. Further, it is important that the mapping be unique since, if the robot and object are physically in one relative configuration but the estimation converges to a different configuration which also fits the measurements, this would lead to failure of the robotic operation.

With this in mind, this section investigate which relative configurations of target object and camera allow for such a unique mapping to be found. In the context of control systems, such a study is called an *observability analysis.* The process has been extensively studied for the case of *linear control systems* [45] and leads to a simple rank condition of an *observability matrix.* However, in the case of nonlinear systems, such as this one, the analysis becomes much more involved [27]. In this work, the simpler linear approximation techniques will be used to identify a subset of the configurations for which the system states are not recoverable from only the measured outputs. In other words, when the system is *unobservable.*

2.4.1 1D Pose Estimation

Consider the problem of estimating the position of a point feature with respect to a camera frame in a planar environment, where the relative position is free to vary in only one dimension. Suppose that the camera's center of projection is at the origin of the camera frame and the principal ray is aligned with the Y^C-axis, as shown in Fig. 2.3. The camera is able to measure the position of a feature point's projection onto the *image line* as the feature point moves parallel to the principal ray of the line camera.

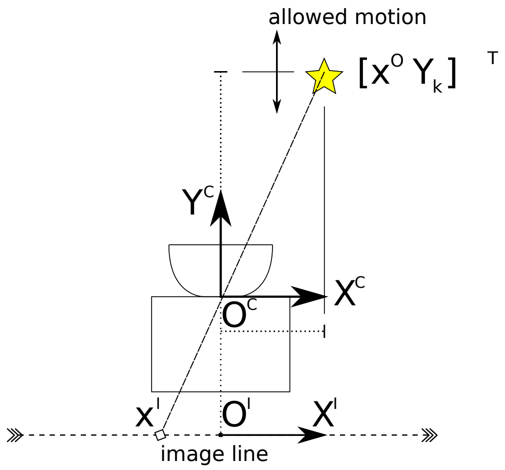


Figure 2.3: The line camera is able to measure the image line projection of the single feature point (star) as relative position between point and camera is free to vary in Y^C direction only.

It is assumed that the position of a single feature point is known in the camera frame X^C-axis, denoted x^O. It is further assumed that the relative motion is smooth and is well approximated by a constant velocity dynamic model. Finally, the focal length of the camera is assumed to be unity for simplicity, $F_x = 1$. The discrete-time system, with no disturbance or measurement noise, is written,

$$\mathbf{x}_k = [\ Y\ \dot{Y}\ ]_k^T \tag{2.35}$$

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} \tag{2.36}$$

$$z_k = g(\mathbf{x}_k) = x_k^I = -\frac{x^O}{Y_k} \tag{2.37}$$

where Y_k is the distance to the feature point along the camera Y^C-axis, $\mathbf{A}$ is defined using the system sampling period $\delta_t > 0$ by,

$$\mathbf{A} = \begin{bmatrix} 1 & \delta_t \\ 0 & 1 \end{bmatrix}, \tag{2.38}$$

and z_k is the camera image line measurement of the feature point. This measurement model is a nonlinear function of the system states.

Observability with Known Model

This section investigates the conditions when estimating the relative position (in this case, depth) of the feature point is not possible. With the nonlinear measurement model defined above, the system cannot be analyzed by immediately applying classical linear observability techniques. Instead, the nonlinear system must be linearized about an operating point. In the case of this system, there are no deterministic inputs.

The process model is already linear but the measurement model is linearized by a first-order Taylor series expansion to form the *linear output matrix*, C_k at each time step k,

$$\mathbf{C}_k = \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_k} = \left[\begin{array}{cc} \frac{x^O}{Y_k^2} & 0 \end{array} \right]. \tag{2.39}$$

Therefore, the linearized system is written,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1}, \tag{2.40}$$

$$\Delta\mathbf{z}_k = \mathbf{C}_k \Delta\mathbf{x}_k, \tag{2.41}$$

where $\Delta\mathbf{z}_k$ and $\Delta\mathbf{x}_k$ are perturbations about the operating point at which the nonlinear measurement model was linearized.

The linear system is said to be *completely observable* if and only if the the rank of $\mathbf{Q}_k$ is equal to n, the number of system states. If a system is not completely observable, it is *unobservable.*

The *linear observability at* $\mathbf{x}_k$ of this linearized relative pose system can be checked by forming the observability matrix, $\mathbf{Q}$ at each time step k,

$$\mathbf{Q}_k = \left[\begin{array}{c} \mathbf{C}_k \\ \mathbf{C}_k\mathbf{A} \end{array} \right] = \left[\begin{array}{cc} \frac{x^O}{Y_k^2} & 0 \\ \frac{x^O}{Y_k^2} & \delta_t \frac{x^O}{Y_k^2} \end{array} \right]. \tag{2.42}$$

This system is unobservable when the rank of $\mathbf{Q}_k$ is less than $n = 2$, a condition that occurs when the determinant is zero,

$$\det(\mathbf{Q}_k) = \frac{\delta_t {x^O}^2}{Y_k^4}. \tag{2.43}$$

Therefore, the system is unobservable when $x^O = 0$ or $|Y_k| \rightarrow \infty$. These conditions make intuitive sense since, when $x^O = 0$, the feature point is directly in front of the camera and moving toward or away from the feature point does not produce a change in the image projection. Further, when $|Y_k| \rightarrow \infty$, the image line projection goes to zero and finite motion in the any direction again does not produce any change in the feature point projection onto the image line.

Conversely, this result dictates that the system is *linearly observable almost everywhere* except those configurations which fall into those two sets identified.

That is, with a known x^O coordinate, the feature point depth can be tracked using the measurements from just one camera image, in almost all cases.

With this initial insight into the simple one-dimensional pose estimation system, the more complicated two-dimensional case will be investigated in the next section.

2.4.2 2D Pose Estimation

The problem is now expanded to estimating the relative position and orientation of the target object frame with respect to the camera frame in a planar environment when there are three degrees of freedom in the relative motion. The motion between the target object frame and camera is free to vary in the two translational coordinates, as well the rotation of the object frame about its origin. While this makes the problem more complicated than the one dimensional example studied in the previous section, some of the insights previously gained will assist in analyzing this case.

It is assumed that there are $N \geq 3$ feature points and the feature point coordinates, $\zeta_j^O = [\ x_j^O\ y_j^O\ ]^T$ for $j = 1,\ 2,\ \ldots\ N$, are known with respect to a common local object (O) frame. Accordingly, the relative pose of this object frame is the subject of this estimation. The further assumption that the relative motion between the camera and object is smooth, is again made, and leads to a relative motion system in the same style as the previous sections.

The state vector at time k is composed of the three pose parameters and their velocities,

$$\mathbf{x}_k = [\ X\ Y\ \theta\ \dot{X}\ \dot{Y}\ \dot{\theta}\ ]_k^T, \tag{2.44}$$

where $[\ X\ Y\ ]^T$ is the location of the object frame origin and θ is the rotation of the object frame, both with respect to the camera frame.

The discrete-time state-space model using the constant velocity is then,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1}, \tag{2.45}$$

this time with

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3\times3} & \delta_t \mathbf{I}_{3\times3} \\ \mathbf{0} & \mathbf{I}_{3\times3} \end{bmatrix}. \tag{2.46}$$

The measurement model is assumed to be that of a simple *pin-hole line camera* model depicted in Fig. 2.4. The camera's center of projection is located at the origin of the camera frame, with the principal ray along the camera Y^C-axis. Projecting

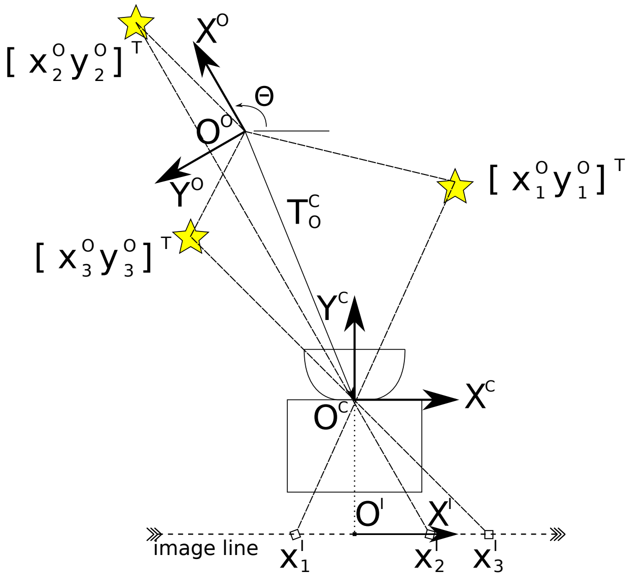


Figure 2.4: The line camera is able to measure the image line projections of the three feature points (stars) of which the coordinates in the object frame are known. From these measurements, the relative pose and orientation of the object frame with respect to the camera frame is found.

the j^{th} feature point onto the camera's image line involves finding the location of the feature point in the camera frame, then projecting onto the image line using the camera calibration matrix.

$$\begin{bmatrix} x^I \\ y^I \end{bmatrix}_j = \mathbf{\Phi}\mathbf{T}_O^C\tilde{\zeta}_j^O = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R}_O^C & \mathbf{t}_O^C \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x^O \\ y^O \\ 1 \end{bmatrix}_j \tag{2.47}$$

where R_O^C is the rotation matrix representing the relative orientation of the object frame with respect to the camera frame,

$$\mathbf{R}_O^C = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.48}$$

and

$$\mathbf{t}_O^C = \begin{bmatrix} X \\ Y \end{bmatrix}. \tag{2.49}$$

The system measurements are these image line projections of the of the j^{th} target object point feature,

$$z_j = g_j(\mathbf{x}) = -f\left(\frac{x_j^C}{y_j^C}\right) = -\left(\frac{x_j^O\cos\theta - y_j^O\sin\theta + X}{x_j^O\sin\theta + y_j^O\cos\theta + Y}\right). \tag{2.50}$$

The nonlinear measurement model can be linearized at each operating point by finding the first-order Taylor series expansion, and forming the measurement Jacobian. This allows for the direct use of linear control techniques and analysis. However, the results are then local to a small neighbourhood about the operating point.

For each element in the measurement vector, the corresponding row in the measurement Jacobian is the set of partial derivatives with respect to the system states evaluated at the operating point,

$$\mathbf{C}_k^j = \frac{\partial g_j(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_k} \tag{2.51}$$

Taking these partial derivatives,

$$\frac{\partial g_j(\mathbf{x})}{\partial X} = \frac{-1}{y_j^C} = \frac{-1}{x_j^O\sin\theta + y_j^O\cos\theta + Y} \tag{2.52}$$

$$\frac{\partial g_j(\mathbf{x})}{\partial Y} = \frac{x_j^C}{y_j^{C2}} = \frac{x_j^O\cos\theta - y_j^O\sin\theta + X}{\left(x_j^O\sin\theta + y_j^O\cos\theta + Y\right)^2} \tag{2.53}$$

$$\frac{\partial g_j(\mathbf{x})}{\partial \theta} = \frac{-\left[\begin{array}{cc} x_j^O & y_j^O \end{array}\right]\left(\left[\begin{array}{c} x_j^O \\ y_j^O \end{array}\right] + R_C^O \left[\begin{array}{c} X \\ Y \end{array}\right]\right)}{\left(x_j^O \sin\theta + y_j^O \cos\theta + Y\right)^2} \tag{2.54}$$

$$\frac{\partial g_j(\mathbf{x})}{\partial \dot{X}} = f \frac{\partial g_j(\mathbf{x})}{\partial \dot{Y}} = \frac{\partial g_j(\mathbf{x})}{\partial \dot{\theta}} = 0. \tag{2.55}$$

Accordingly, the linearized relative pose system is written,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1}, \tag{2.56}$$

$$\Delta\mathbf{z}_k = \mathbf{C}_k \Delta\mathbf{x}_k, \tag{2.57}$$

where $\Delta\mathbf{z}_k$ and $\Delta\mathbf{x}_k$ are perturbations about the operating point at which the nonlinear measurement model was linearized.

Linear Observability

Since there are three pose parameters to estimate in this problem, a necessary condition for the system to have a unique solution is that *at least* three feature point image line measurements must be observed.

Therefore, for the 2D relative pose estimation problem, consider the case when three features are observed in each frame. As a result, the measurement Jacobian, formed by the partial derivatives of the output equations with respect to the system states, $\mathbf{C}_k \in \mathbb{R}^{3\times 6}$. The measurement Jacobian at time-step k, $\mathbf{C}_k$, has the form,

$$\mathbf{C}_k = \left[\begin{array}{cc} f\mathbf{V}^{-1}\mathbf{H} & \mathbf{0} \end{array}\right]_k, \tag{2.58}$$

where

$$\mathbf{V}_k = \left[\begin{array}{ccc} y_1^{C^2} & & \\ & y_2^{C^2} & \\ & & y_3^{C^2} \end{array}\right]_k \tag{2.59}$$

and

$$\mathbf{H}_k = \left[\begin{array}{ccc} -x_1^C & y_1^C & -[\, x_1^O \; y_1^O \,][\, x_1^{C/O} \; y_1^{C/O} \,]^T \\ -x_2^C & y_2^C & -[\, x_2^O \; y_2^O \,][\, x_2^{C/O} \; y_2^{C/O} \,]^T \\ -x_3^C & y_3^C & -[\, x_3^O \; y_3^O \,][\, x_3^{C/O} \; y_3^{C/O} \,]^T \end{array}\right]_k \tag{2.60}$$

with

$$[\, x_j^{C/O} \; y_j^{C/O} \,]^T = \mathbf{R}_C^O[\, x_j^C \; y_j^C \,]^T, \tag{2.61}$$

which are the camera coordinates of the j^{th} feature point expressed with respect to the O frame.

Theorem 1. *Assuming $f > 0$ and $\delta_t > 0$, the linearized system, (2.56) and (2.57), is unobservable if and only if:*

(i) $\begin{bmatrix} x_1^{C^2} + y_1^{C^2} \\ x_2^{C^2} + y_2^{C^2} \\ x_3^{C^2} + y_3^{C^2} \end{bmatrix}_k^T \begin{bmatrix} x_2^C y_3^C - x_3^C y_2^C \\ x_3^C y_1^C - x_1^C y_3^C \\ x_1^C y_2^C - x_2^C y_1^C \end{bmatrix}_k = 0,$

or

(ii) $\exists j \in \{1, 2, 3\}$ *s.t.* $y_{j,k}^C \to \infty$.

Proof. The observability matrix at time step k, for a linear control system is formed by stacking products of the measurement matrix and powers of the process matrix,

$$\mathbf{Q}_k^T = \left[\mathbf{C}_k^T \ (\mathbf{C}_k\mathbf{A})^T \ ... \ (\mathbf{C}_k\mathbf{A}^{(n-1)})^T\right]^T. \tag{2.62}$$

For the linearized relative pose system, the observability matrix is,

$$\mathbf{Q}_k = f \begin{bmatrix} \mathbf{V}^{-1}\mathbf{H} & \mathbf{0} \\ \mathbf{V}^{-1}\mathbf{H} & \delta_t \mathbf{V}^{-1}\mathbf{H} \end{bmatrix}_k. \tag{2.63}$$

This matrix loses rank, and the linear system is unobservable, if and only if the determinant of the submatrix $f\delta_t\mathbf{V}_k^{-1}\mathbf{H}_k$ is zero,

$$\det(f\delta_t\mathbf{V}_k^{-1}\mathbf{H}_k) = f\delta_t\frac{\det(\mathbf{H}_k)}{\det(\mathbf{V}_k)} = 0. \tag{2.64}$$

Therefore, the system is unobservable if and only if,

$$\det(\mathbf{V}_k) \to \infty, \tag{2.65}$$

or

$$\det(\mathbf{H}_k) = 0. \tag{2.66}$$

These conditions are equivalent to,

$$\det(\mathbf{V}_k) = \left(y_1^{C^2} y_2^{C^2} y_3^{C^2}\right)_k \to \infty, \tag{2.67}$$

which indicates that the camera frame Y^C-axis coordinate of any of the three feature points goes to infinity. Additionally,

$$\det(\mathbf{H}_k) = \begin{bmatrix} x_1^{C^2} + y_1^{C^2} \\ x_2^{C^2} + y_2^{C^2} \\ x_3^{C^2} + y_3^{C^2} \end{bmatrix}_k^T \begin{bmatrix} x_2^C y_3^C - x_3^C y_2^C \\ x_3^C y_1^C - x_1^C y_3^C \\ x_1^C y_2^C - x_2^C y_1^C \end{bmatrix}_k = 0, \tag{2.68}$$

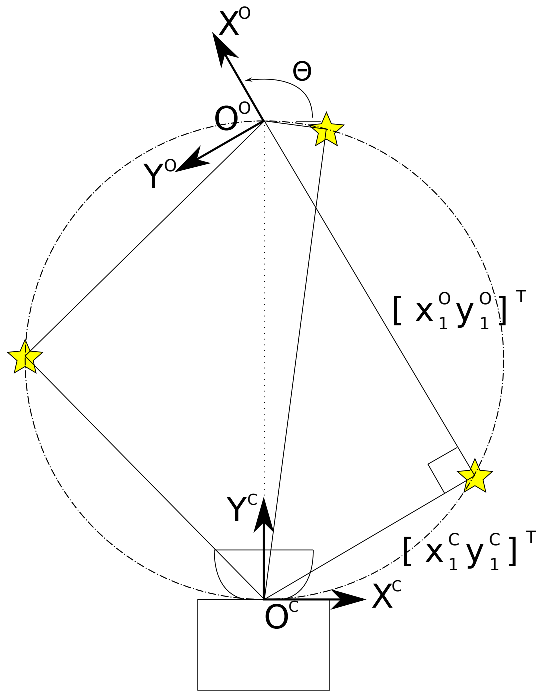


Figure 2.5: The zero-determinant of the $\mathbf{H}_k$ matrix specifies a circle through the three features and the camera center of projection.

which dictates that the rest of the unobservable configurations occur due to the camera center of projection lying on the *feature circle* $\mathcal{F}$, shown in Fig. 2.5. The feature circle is the *circumscribed circle* through the three feature points in the camera frame. When (2.68) is satisfied, this circle passes through the point $[\ 0\ 0\ ]^T$, the camera center. □

These necessary and sufficient conditions for the system being unobservable can be interpreted physically to recognize the situations that should be avoided to ensure the relative pose can be tracked successfully. The first condition in (i) is satisfied when the point features and camera center of projection lie on a common circle, and the second condition, (ii) is true when any of the point features are infinitely far away from the camera in the Y^C-axis.

This result is consistent with the work of Bonnifait and Garcia [5] who consider the observability of estimating the pose of a planar mobile robot with respect to three beacons at known locations. Importantly, their analysis excludes the use of deterministic inputs, which would include odometric information. They provide a geometric interpretation for the result dictating that the robot pose is constrained, by the relative observation angles between a pair beacons, to lie on a circle. The three pairs of beacons lead to three circles which intersect at the robot location.

The system becomes unobservable when these three circles merge. The positions where this happens are on the circle defined by the three beacons.

From a practical point-of-view, measurements from three non-collinear feature points will make the system observable if the distance to the object is sufficient to keep the camera off of this *feature circle*.

Adding a fourth feature to the system will make it completely observable as long as this feature does not fall on the *circumscribed circle* through the first three features. This excludes common shapes like rectangles. In fact, this is true for any number of supplemental features which together form a *cyclic polygon*, as will be shown in the next section.

Unobservable Configurations

For any three feature points with coordinates, $[\ x_j^O\ \ y_j^O\ ]^T, j = 1, 2, 3$, a *feature circle*, $\mathcal{F}$, circumscribed through all three points can be found. It was shown in the previous section that if the camera frame origin also lies on $\mathcal{F}$, the system is unobservable suggesting that the measurements do not imply a unique state. However, there is no information about the cause and nature of these singularities.

This section demonstrates that the feature point projections onto the image line when the camera center lies on $\mathcal{F}$, are not unique and, in fact, there is a continuum of configurations at which the feature points will have the same projection on the image line. This means, at least in theory, that the system could reach such a singular configuration, and move along the singular trajectory without an observer being able to detect the changes of the true state. In the case of an EKF, this would almost certainly lead to filter divergence and tracking failure. The relationship between the feature locations, the image projections and the relative object pose are investigated in order to characterize these singular configurations.

When the camera frame origin lies on $\mathcal{F}$, the relative pose system can be parameterized as shown in Fig. 2.6. Note that the symbols representing these angles in the figure are temporarily redefined in this subsection.

Since $\mathcal{F}$ passes through the feature point and the camera origin at an unobservable configuration, the triangle formed by the those two points and the circle $\mathcal{F}$ centre is isosceles with two side lengths r, which is the radius of $\mathcal{F}$. Using geometric identities, the angles, ρ and β, can be written as,

$$\rho = \psi - \phi \tag{2.69}$$

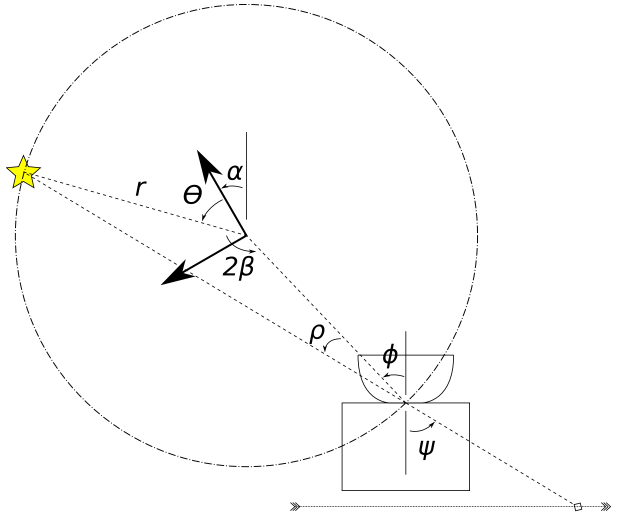


Figure 2.6: An unobservable configuration showing one feature point. Using trigonometric identities facilitates parameterizing the unobservable configurations of the camera and target object.

$$\beta = \frac{1}{2}\left(\pi - \theta - \alpha + \phi\right). \tag{2.70}$$

The motion of the feature along the projection ray through the center of projection can be interpreted as the result of a *slider crank mechanism* [26] with rigid arms of length r between the camera center, the circle center, and the feature point. Writing the constraint equations for such a physical system,

$$r \cos \rho - r \sin \beta = 0 \tag{2.71}$$

and

$$r \sin \rho - r \cos \beta = 0. \tag{2.72}$$

The equations can be expanded and simplified to solve for α,

$$\alpha = 2\psi - \phi - \theta. \tag{2.73}$$

For a set of three feature points, three equations of this form result,

Solving the pairs of equations for the projection ray angles of feature points i and j where $i \neq j$,

$$\psi_i = \psi_j + \frac{1}{2}(\theta_i - \theta_j). \tag{2.74}$$

The resulting equations describe a set of three slider crank mechanisms with a common first link between the camera center and the center of $\mathcal{F}$. In the current form, the projection ray angle, ψ_i of one of the feature points can be specified, as well as the angle, ϕ to the center of $\mathcal{F}$. If more features are added to the system which lie on $\mathcal{F}$, there will still be one degree of freedom.

As an example, two configurations which result in the same image projections are shown in Fig. 2.7. The angle ϕ is specified with values 0 and $\frac{\pi}{4}$. This illustrates why the system goes unobservable in these configurations. There is an entire set of configurations for which the resulting measurements will be identical, so one cannot discriminate between them using this information. This insight will allow the more complicated 3D case to be studied with this focus since a direct interpretation of the determinant of the larger observability matrix is much more complex.

In the context of any state observer, the fact that multiple configurations can result in the same measurements means there the inverse mapping from the image line measurements to the system states is *not unique*. However, similar to the one dimensional case studied previously, the system is completely observable almost everywhere. Additionally, the chances of exactly landing in a singular configuration is low due to the presence of measurement noise. More likely, the system will become extremely sensitive near these regions, rather than going unobservable.

Since the object model is completely known, it would be possible to detect if a singular configuration is being approached. If possible, the camera could be actively controlled away from these configurations. Alternatively, redundant features would allow the estimator to choose features such that the unobservable configurations have a less profound effect on the estimation.

The unobservable configurations explored in the planar pose estimation case are a subset of those of the full three dimensional estimation. The analysis that follows attempts to extend this approach to all six degrees of freedom.

2.4.3 3D Pose Estimation

Now that the lower dimensional cases have been explored, the full six degree of freedom relative pose estimation is considered. The target object is now free to undergo translation and rotation in 3D space, and the relative pose vector, $\mathbf{w}$ is given in (2.2).

As before, it is assumed that the target object model point feature locations are perfectly known in the O frame and the relative motion between the camera and object is smooth. Accordingly, the dynamic model of the system was given in Section 2.3.1 and the monocular camera measurement model in 2.3.2. However, in

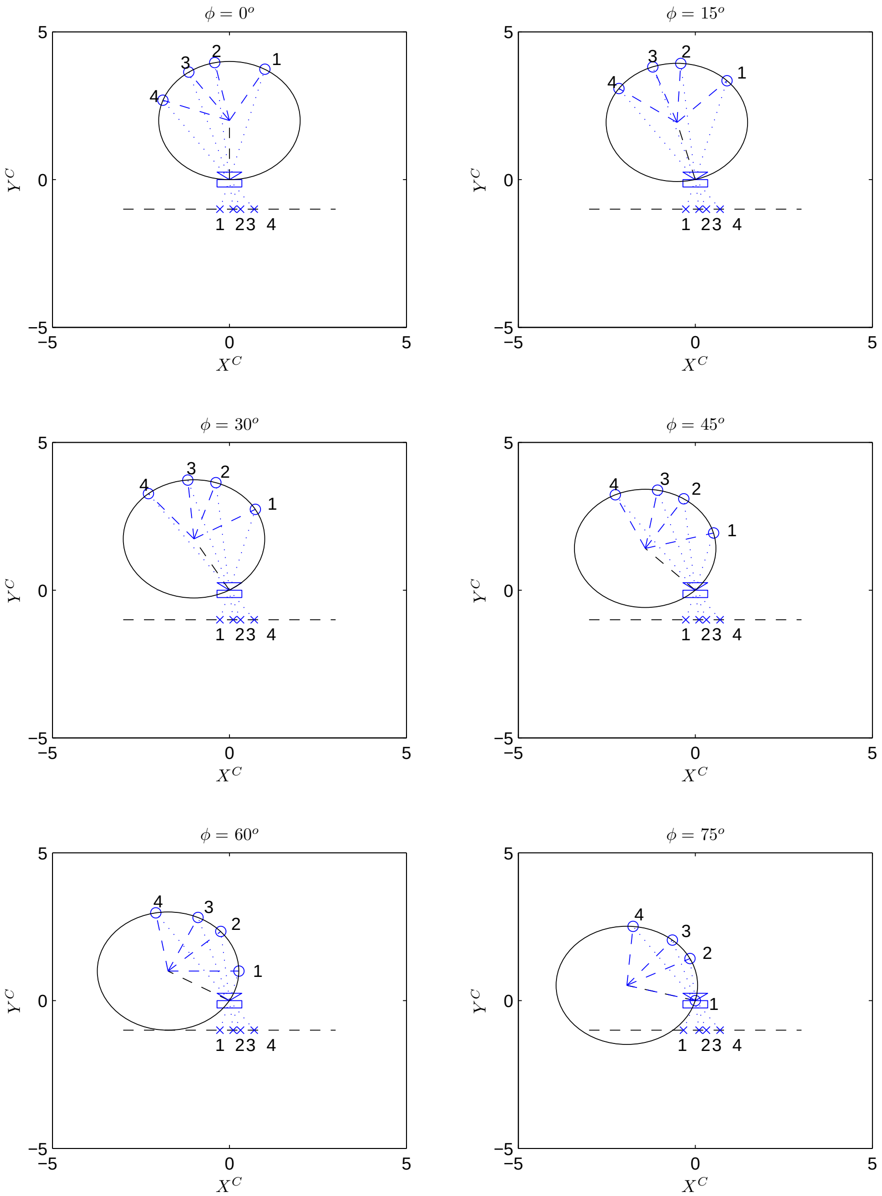


Figure 2.7: A set of distinct relative positions and orientations of the target object which all result in the same projections of the feature points on the camera image line.

this analysis, it will be assumed that these processes are not subject to noise. Each target object feature point observation results in two measurements corresponding to the horizontal and vertical coordinates on the camera's image plane.

To apply the linear control systems observability conditions, the nonlinear measurement model needs to be linearized, resulting in the measurement Jacobian, C_k,

$$\mathbf{C}_k = \begin{bmatrix} \mathbf{H}_k & \mathbf{0} \end{bmatrix}, \tag{2.75}$$

where

$$\mathbf{H}_k = \begin{bmatrix} \frac{\partial \mathbf{g}_{1_x}(\mathbf{x})}{\partial X} & \frac{\partial \mathbf{g}_{1_x}(\mathbf{x})}{\partial Y} & \frac{\partial \mathbf{g}_{1_x}(\mathbf{x})}{\partial Z} & \frac{\partial \mathbf{g}_{1_x}(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{g}_{1_x}(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{g}_{1_x}(\mathbf{x})}{\partial \psi} \\ \frac{\partial \mathbf{g}_{1_y}(\mathbf{x})}{\partial X} & \frac{\partial \mathbf{g}_{1_y}(\mathbf{x})}{\partial Y} & \frac{\partial \mathbf{g}_{1_y}(\mathbf{x})}{\partial Z} & \frac{\partial \mathbf{g}_{1_y}(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{g}_{1_y}(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{g}_{1_y}(\mathbf{x})}{\partial \psi} \\ \frac{\partial \mathbf{g}_{2_x}(\mathbf{x})}{\partial X} & \frac{\partial \mathbf{g}_{2_x}(\mathbf{x})}{\partial Y} & \frac{\partial \mathbf{g}_{2_x}(\mathbf{x})}{\partial Z} & \frac{\partial \mathbf{g}_{2_x}(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{g}_{2_x}(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{g}_{2_x}(\mathbf{x})}{\partial \psi} \\ \frac{\partial \mathbf{g}_{2_y}(\mathbf{x})}{\partial X} & \frac{\partial \mathbf{g}_{2_y}(\mathbf{x})}{\partial Y} & \frac{\partial \mathbf{g}_{2_y}(\mathbf{x})}{\partial Z} & \frac{\partial \mathbf{g}_{2_y}(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{g}_{2_y}(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{g}_{2_y}(\mathbf{x})}{\partial \psi} \\ \frac{\partial \mathbf{g}_{3_x}(\mathbf{x})}{\partial X} & \frac{\partial \mathbf{g}_{3_x}(\mathbf{x})}{\partial Y} & \frac{\partial \mathbf{g}_{3_x}(\mathbf{x})}{\partial Z} & \frac{\partial \mathbf{g}_{3_x}(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{g}_{3_x}(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{g}_{3_x}(\mathbf{x})}{\partial \psi} \\ \frac{\partial \mathbf{g}_{3_y}(\mathbf{x})}{\partial X} & \frac{\partial \mathbf{g}_{3_y}(\mathbf{x})}{\partial Y} & \frac{\partial \mathbf{g}_{3_y}(\mathbf{x})}{\partial Z} & \frac{\partial \mathbf{g}_{3_y}(\mathbf{x})}{\partial \phi} & \frac{\partial \mathbf{g}_{3_y}(\mathbf{x})}{\partial \theta} & \frac{\partial \mathbf{g}_{3_y}(\mathbf{x})}{\partial \psi} \end{bmatrix}_k, \tag{2.76}$$

and $\mathbf{g}_{j_x}(\mathbf{x})$ indicates the X^I coordinate of the image plane projection of the j^{th} feature point, and similarly for $\mathbf{g}_{j_y}(\mathbf{x})$ and the Y^I coordinate.

Similar to the planar case, the observability matrix at time step k has the form,

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{H}_k & \mathbf{0} \\ \mathbf{H}_k & \delta_t \mathbf{H}_k \end{bmatrix}. \tag{2.77}$$

This observability matrix will have full rank of $n = 12$, when $\mathbf{H}_k$ has rank 6. Therefore, the system is unobservable when the rank of $\mathbf{H}_k$ is less than 6. The expression for the determinant of this dense 6×6 matrix quickly becomes very complex.

Michel *et al.* investigate the singular configurations for the relative pose estimation system when three feature points of known geometry are observed in a single camera image plane [41]. The measurement Jacobian is interpreted as a measure of the *optical flow* of the image plane coordinates of the feature points. In this way, the unobservable configurations of the target object are those which result in no motion of the image projections. This is analogous to the feature points remaining on the same projection rays through the camera's center of projection.

The results of their analysis dictates that the system is singular when the camera's center of projection lies on the cylinder with circular cross section defined by the three feature points, and with axis perpendicular to the plane containing

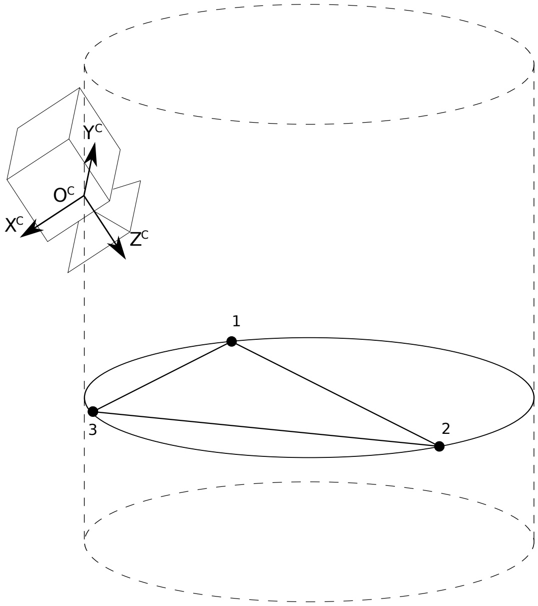


Figure 2.8: The pose estimation system with three feature points is unobservable when the camera center lies on the cylinder formed by the feature points. The circular cross-section is circumscribed through the points and the cylinder axis is perpendicular to the plane defined by the three points.

the three feature points [41]. An example configuration is illustrated in Fig.2.8. This geometric result is a generalization of the planar case studied in the previous section, for which the singular configurations are in a circle created by the three feature points. In that case, the camera lies *on the plane* defined by the feature points, and therefore on the cylinder, as well.

Using the intuition gained from the previous planar case, it is apparent that the system is unobservable when the image plane coordinates of the target object feature point projections are not unique to a particular relative pose for a given target model. The following will detail the initial investigation into this problem when the target object consists of three feature points.

Any 3D point which lies on the line from a particular set of image plane coordinates, through the center of projection, will be projected onto those image coordinates. For the three feature points, let $\mathbf{r}_1$, $\mathbf{r}_2$, and $\mathbf{r}_3$ represent those three lines or rays, respectively. As a result, a particular relative configuration of the target object leads to an unobservable system if there is a different relative pose which places the feature points on those same projection lines.

For any pair of feature points, ζ_i and ζ_j, keeping these two feature points on the same projection line, amounts to a rotation with axis parallel to the plane defined by the respective rays, $\mathbf{r}_i$ and $\mathbf{r}_j$. With a rigid-body, the distance between feature points ζ_i and ζ_j is fixed.

Consider the relative pose of a target object with respect to the camera frame. A distinct configuration, in which the image plane projections of the three feature points are the equivalent to the original, can be constructed in the following manner, shown in Fig. 2.9. Choose a feature point and move it along its projection line, here called $\mathbf{r}_1$. Rotate the target object around that feature point about an axis perpendicular to the plane defined by $\mathbf{r}_1$ and $\mathbf{r}_2$ to align the second feature with $\mathbf{r}_2$. There are two possible solutions imposed by the length constraint. Now, two of the three feature points are aligned with their original projection rays. For the final feature point, the first two feature points will remain on their projection rays if the target object is rotated about an axis with direction parallel to $\zeta_2 - \zeta_1$. The set of possible resulting positions for ζ_3 form a circle in 3D space. The second configuration will have the same measurements as the original when this circle and $\mathbf{r}_3$ intersect. If this is not the case, the new position of ζ_1 along $\mathbf{r}_1$ must be adjusted until the resulting circle and $\mathbf{r}_3$ have a common point.

Using a simulation developed in *MATLAB*, it has been observed that a set of three feature points will have the same image plane measurements in multiple target object configurations for almost all starting relative configurations. That is, it is nearly always possible to find distinct relative poses of the same target object which lead to the feature points lying on the same projection rays. For a set of image plane measurements of three feature points, there are up to four possible relative configurations [41].

It is important to point out that while multiple configurations lead to the same measurements, the measurement Jacobian only goes singular at the cylindrical set above. This is essentially the difference between *tracking* and *detecting*. Assuming that the initial condition is correct, the pose can be *tracked* from this point onwards. Accordingly, the three dimensional relative pose estimation system with a known target object model consisting of three feature points is *observable in almost all configurations*. However, with no prior information about the previous pose, the relative pose of the target object cannot be *detected* uniquely using only three feature point measurements.

The results presented in this section are only preliminary for this analysis. A better means of parameterizing the relative poses for a particular model which lead to the same measurements has yet to be identified. Additionally, adding a fourth feature point to the target object seems to make the system observable in practice.

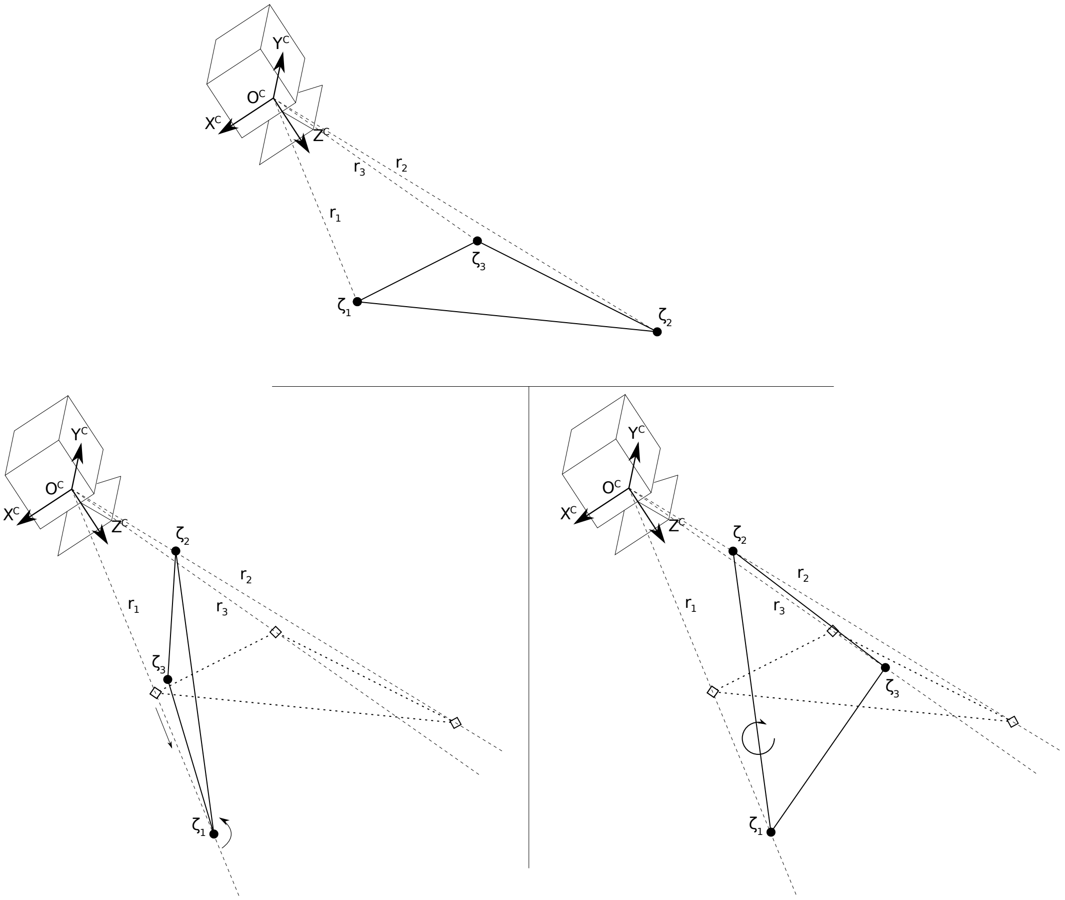


Figure 2.9: In most cases, multiple relative configurations for a set of image measurements can be found by moving ζ_1 along $\mathbf{r}_1$, rotating within the plane defined by $\mathbf{r}_1$ and $\mathbf{r}_2$ to align ζ_2 with $\mathbf{r}_2$, and then rotating around the axis $\zeta_2 - \zeta_1$ to align ζ_3 with $\mathbf{r}_3$.

It is noted in [41] that four coplanar feature points will lead to a unique relative configuration if one exists.

Chapter 3

Unknown Object Model

In the previous chapter, it was assumed that a perfect model of the target object was known a priori. Consequently, all of the Cartesian positions of the object feature points were perfectly known with respect to a common local object (O) coordinate frame. It was determined that, *in almost all configurations* of camera and target object, the relative pose was theoretically recoverable at each time step using only measurements from a single monocular camera from three point features on the object.

In this chapter, this known model assumption is relaxed and the feasibility of a combined estimation of both the relative pose *and* target object model is investigated. Such an estimation would facilitate a greater number of application areas where the target object model, on which an operation is being performed, is not well known, or even completely unknown.

It is well understood in the computer vision community that when bearing-only measurements, such as those from a monocular camera, are used to estimate both the relative pose and target object model geometry, only an up-to-scale solution can be recovered [9] [35] [15]. As shown in Fig. 3.1, a scaled version of the true solution will be found that is accurate up to a single degree of freedom which pertains to the global scale. Another way of saying this is that the solution is estimated up to a similarity transform [35]. As a result, without a priori information about this scale, any estimation of both relative pose and target model feature locations using only the monocular camera image plane measurements would not be able to recover the global scale of the relative motion and object model.

For position-based visual servo control, the scale of the relative motion and target object is particularly important since operations are to be performed on the

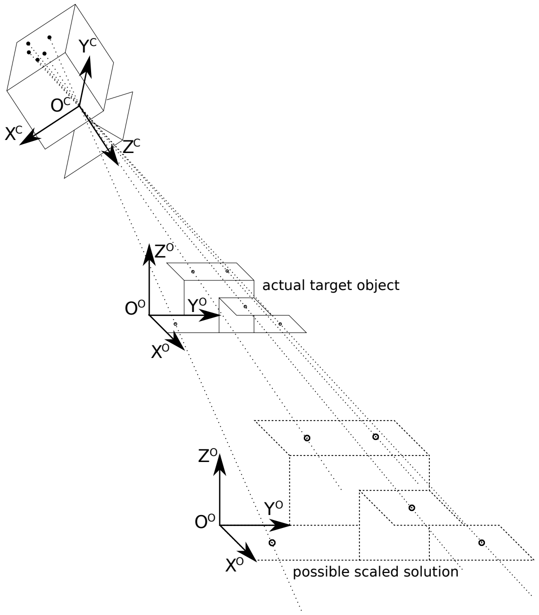


Figure 3.1: Since the monocular camera can only measure the bearing to the target object feature points, the recovered solution will be unable to disambiguate the scale of the environment.

workpiece. Therefore, supplemental information or measurements which include scale information must be added to the estimation to allow for operations to be carried out on target objects with unknown models.

In this chapter, the combined relative pose and target model estimation with a single monocular camera will be considered to determine why this deficiency exists and help suggest some possible solutions.

3.1 Problem

3.1.1 Definition

Similar to the case when the object model was known, the goal of this estimation is primarily to recover an accurate, real-time relative position and orientation estimate of the target object frame with respect to the camera frame from observations as a sequence of camera images. However, the locations of the object feature points in a local object frame, $\zeta_j^O = [\ x^O\ y^O\ z^O\ ]_j$, are not known a priori. As a result, these model parameters now must be estimated alongside the pose parameters which constitute the relative pose vector, $\mathbf{w}$.

The quality of the pose estimates depends critically on the accuracy of the target object model, which in turn depends on the relative pose estimates. As a result, the pose and model must be estimated simultaneously.

3.1.2 Assumptions

As in the known object model case, some assumptions on the system are made to facilitate the combined estimation:

Assumption 1 The relative motion between the target object and camera is *smooth* and can be approximated by a constant velocity model. Since both the robot and target object are free to move, this general motion model is used and the modelling error is represented as disturbance noise.

Assumption 2 The target object is defined by set of $N \geq 3$ point features, $[\ x^O\ y^O\ z^O\ ]_j^T,\ j = 1, 2, ..., N$ which are fixed with respect to a common local object coordinate frame. That is, the target object is a *rigid body.*

Assumption 3 Object features points are uniquely distinguishable in the camera images and their respective image plane coordinates can be measured at each time step in an image sequence. The correspondance between image plane measurements and each of the point features is available at each time step. There are a large number of techniques for solving this problem in the literature, but it will not be dealt with in this work.

Assumption 4 Both the *disturbance noise* and *measurement noise* can be represented by vectors of zero-mean Gaussian noise with known covariance. All of the individual noise processes are assumed to be uncorrelated to other noise processes and the system states at each system time step.

3.2 Common Approaches

While the problem of estimating both the relative pose and target object model has received only a small amount of attention in the visual servoing context, it has been extensively studied both in the mobile robotics and computer vision fields.

In computer vision, the process is called *Structure From Motion* (SFM) in which the 3D structure of an environment is reconstructed based on 2D camera images from unknown locations [15]. An overview of several different SFM approaches and algorithms is available in [35].

In the following two sections, an overview is presented regarding current solutions proposed by researchers in the mobile robotics and visual servoing domains. Both groups attempt to solve the problem mentioned previously, but until recently, the link between them has not been fully recognized and exploited. This review will attempt to acknowledge the strengths and weaknesses of the current techniques in anticipation of combining the strengths of each into a common framework.

3.2.1 Simultaneous Localization and Mapping

Within mobile robotics, the process is called *Simultaneous Localization and Mapping* (SLAM) and involves a mobile robot being placed at an unknown location in an unknown environment and incrementally building a consistent map of the environment while concurrently localizing itself within it [20]. A good overview of the current state of the art, as well as the outstanding issues is given by Durrant-Whyte and Bailey in [20] and [3].

The formulation of SLAM studied in this work is the estimation-theoretic or EKF-based approach identified in [19]. In this framework, the solution to the navigation problem can be found recursively and the uncertainties in the system and resulting estimates can be represented as random variables with a statistical basis. This approach closely follows that of the traditional position-based visual servoing pose estimation [53] in which the localization problem is formulated as a stochastic process and recursively estimated using an EKF.

The system is formed by augmenting the vehicle position and orientation states with the map feature point parameters. A robotic platform moving through the environment is able to make measurements of the relative location between the individual landmarks (which are assumed stationary) and the vehicle itself [19]. By making successive observations over time at different locations, the algorithm concurrently builds a statistical map of the feature locations and tracks the robot's location within it. At each time step k, the state estimate is represented by a Gaussian random variable with associated mean, $\hat{\mathbf{x}}_k$, and covariance matrix, $\mathbf{P}_k$.

The steady-state behaviour of this estimate is studied by Dissanayake *et al.* in [19]. Given the similarity with the EKF-based relative pose and target model estimation problem, these results apply equally well in the position-based visual servoing context. Mathematical proofs are presented for the following properties of the SLAM solution [19]:

1. The determinant of any submatrix of the map covariance matrix decreases monotonically as observations are made throughout the robot trajectory.
2. The estimates of the feature locations become fully correlated as the number of successive measurements goes to infinity.
3. An individual feature's estimate covariance is bounded below by only the covariance of the initial vehicle location, as the number of successive measurements tends to infinity.

The critical insights offered by these results are that the correlation between the feature location estimates only grows over time, and that it is possible to recover the *relative* locations of features within the map with perfect accuracy, while the absolute location is limited by the initial vehicle location uncertainty [19]. This makes intuitive sense since without measurements of the *absolute* position of the robot, the entire relative map can be shifted in an absolute coordinate frame without changing the measurements.

For pose estimation with position-based visual servoing, these results mean that an accurate target object model can be found concurrently while estimating the relative pose of that object with respect to the robot end-effector. Since the absolute position of the target object in the absolute world frame is not important for performing relative operations, the resulting estimate should perform flawlessly.

While the SLAM process applies most generally to mobile robots equipped with any number of different types of sensors, the use of cameras as the primary sensor has been investigated by several researchers.

SLAM with Monocular Camera

Deans and Hebert [15] investigate 2D SLAM using bearing-only sensors mounted on a mobile robot. After making the connection with SFM techniques, they suggest using a hybrid approach using a combined EKF and Bundle adjustment technique. Bundle adjustment is a batch optimization technique widely used in the computer vision community for SFM [15]. More significant than the proposed algorithm itself is the insight offered by Deans and Hebert into the structure of the bearing-only SLAM problem. Specifically, a single image from a monocular camera is able to only determine the ray on which a feature point lies and in the two dimensional case, there are four *gauge freedoms* in the estimation. That is, the solution can be translated, rotated or scaled and the resulting measurements will be unchanged [35]. To overcome this limitation, an absolute coordinate frame and scale must be imposed as a set of constraints for the solution to be unique. Further, they assert that odometric data is sufficient to disambiguate the scale of the environment, but a bias in the odometry cannot be corrected by the bearing-only measurements. These insights are key to successfully recovering the global scale of the solution in the next chapter.

The first significant application of the SLAM framework using a single monocular camera was by Davison in [12]. The full 3D position and orientation of a handheld camera with respect to an (almost) unknown environment is estimated in real-time, as well as the locations of a large set of feature points within the environment. The estimations are performed by an EKF, and the algorithm includes the ability to add or remove feature points from the map as new landmarks become available or others are no longer stable during the estimation. The use of a camera with a wide-angle lens is suggested in [13] and the process is further refined in [14].

Eade and Drummond have developed a monocular camera SLAM algorithm [21] based on the *FastSLAM* filter proposed by Montemerlo *et al.* [42]. The *FastSLAM* algorithm is a variation of the Particle Filter [50] which capitalizes on the structure of the SLAM problem to decrease the computational requirements for maintaining accurate estimates. As a result, maps with large numbers of landmarks can be maintained at reduced computational cost compared with EKF-based solutions [50]. The difficulty with this method comes when trying to maintain a sufficient number of particles to adequately represent the probability distributions over the entire state space. Even if there are a large number of particles, there may not be any particles near the correct state. This is called *particle deprivation* [50]. Despite this, Eade and Drummond show good results for large maps of features with a system operating in real-time.

Davison's work of EKF-based monocular SLAM has since been improved, first by Montiel *et al.* [43]. Instead of parameterizing the map feature point locations by their Cartesian parameters in the world frame, it is suggested that tracking the location of the camera's center of projection where the 3D point was first observed, the azimuth and altitude for the observation ray and the inverse of the depth along that ray to the feature point, provides a better representation. Using these six parameters for each feature point is called *inverse depth parameterization.* The bearing-only nature of the camera sensor means that the greatest uncertainty associated with a feature point upon the first observation is in the depth parameter. By using the *inverse* of the depth, the resulting measurement model is shown to be significantly more linear than using a Cartesian representation [43]. As a result, the Gaussian distributions representing the state estimates are more accurately propogated through the transformations in the measurement update step of the EKF. Using this parameterization allows for new feature points to be added to the estimation upon the first observation, and even permits the use of feature points at an infinite distance from the camera.

In a more recent work, an alternative parameterization using an *inverse scale parameter* along with a vector bearing to the feature point has been proposed by Marzorati *et al.* [40]. Instead of requiring six parameters, each feature point requires only four and the estimation proceeds in a *robocentric* manner. This is the same relative position and orientation parameterization as is traditionally used in position-based visual servoing pose estimation [53], and is said to produce a measurement model with better linearity properties. Further, it is asserted that the inverse depth parameterization from Montiel *et al.* suffers from underestimation of the feature location uncertainty.

The second improvement to Davison's framework is detailed in Civera *et al.* [9]. The significant contribution of this work is the explicit realization that the global scale of the map and robot pose cannot be recovered from monocular camera measurements alone. More significantly, it is demonstrated that the estimation can proceed with no a priori information about the environment or initial robot location and the resulting solution will be consistent in an up-to-scale manner. In other words, the solution will almost certainly have the wrong global scale and with only the monocular camera measurements, this parameter will not converge to the correct value.

Where the previous methods try to map and localize a robot in an environment defined by a set of discrete features, *Trajectory-Oriented SLAM* [3] attempts to directly align sensed data. Each pose estimate has an associated scan of data from, for example, a laser scanner or camera. Each of these poses are aligned to determine a global map. An example of such a system using camera data is found in [34]. Konolige *et al.* attempt to match frames from a robot-mounted camera at each time step to map dense environments and localize the robot within it. They propose schemes for selecting which information should be kept from the large quantity available. Without an effective means of dealing with the information, the memory space requirements for this method will grow unbounded as the estimation proceeds [3].

3.2.2 Visual Servoing Pose Estimation

Concurrently, researchers in the visual servoing domain have sought to augment traditional methods to accomodate for unknown target objects. The original division between image-based and position-based methods is now being blurred with the development of so-called hybrid methods. This section will detail the recent developments in all three of these visual servoing flavours.

Several authors have attempted to add an explicit pose estimation phase to the original image-based visual servoing architecture. Hafez *et al.* have added a Gaussian Particle filter in order to estimate the depth of a feature observed in the image plane [25]. With image-based techniques, it is important to have an accurate estimate of a feature's depth in order to obtain a valid image Jacobian for controlling in the image plane. For each feature on the target object, a distribution of particles is drawn from the conical region projecting out from the camera's center of projection through the observed feature. Subsequent observations are used to prune out particles at the wrong depth estimates until the resulting distribution collapses to a near-Gaussian shape. At this point, an estimate of the depth of the feature can be extracted and it is claimed that the stability of image-based methods can be improved.

This technique suffers from the same problems as all bearing-only estimations where the solution may be consistent but to the wrong scale. There are no supplemental measurements used to recover the scale and the estimates will likely diverge from the true state.

Recently, much of the work in image-based visual servoing has changed to a *hybrid* technique called *2 1/2 D visual servoing* [37]. This approach does not require an explicit 3D model of the target object, but instead calculates a *partial pose estimate* between the current and desired image plane configurations. Subsequently, the relative position and orientation controllers can be decoupled similar to position-based visual servoing control schemes. The scale of the target object cannot be recovered from the measurement space and instead must be initialized to an approximate value prior to starting an operation. This technique also requires at least 8 feature points for non-coplanar target objects [37], compared with 3 for other methods.

For position-based visual servoing, this combined estimation problem was investigated by Deng *et al.* in [17]. In this work, only monocular camera measurements were used with two decoupled EKFs. One is designed to estimate the relative pose of the object frame with respect to the camera frame and the other to estimate the target object model feature locations. The two EKFs proceed in lock-step with the results of one utilized by the other in the next time step. Further, two methods are proposed to initialize the estimation. The first assumes that the target object is stationary and uses the robot kinematics and joint encoder readings to calculate changes in the relative pose, allowing for triangulation of feature point locations from image-plane measurements. The second method assumes that the relative depth of the object is approximately known and that the object is close to planar. However, since the system uses only image-plane measurements in the estimation, the scale of the object and relative motion will not be recovered accurately. The system is able to converge to a consistent solution, but it will rarely be to the correct scale.

In [23], Fontanelli *et al.* use a position-based visual servoing switching controller to explore and navigate a mobile robot around unknown indoor environments. Significantly, this work acknowledges the parallels between the SLAM and position-based visual servoing pose estimation problems, and directly applies techniques from SLAM to estimate the environment map. The global scale is recovered using the odometric data from the robot dynamic model and therefore the environment is assumed to be static. While calling this particular approach visual servoing is a slight misnomer (this is, in fact, closer to monocular camera SLAM used as an estimator for the robot controller), it is the recognition of the link between the fields which is valuable.

3.3 Recursive Estimation

In the previous chapter, it was assumed that the target model parameters were known a priori. However, in this chapter, these parameters are initially unknown and must be estimated online along with the system states. The system, therefore, becomes a *combined state and parameter estimation*.

The unknown parameters can be appended to the original state vector and treated as *static states*. This new system is now a state estimation and can be solved recursively using the EKF framework, as before. The process and measurement models used in this combined estimation are similar to the relative pose estimation system, but some former parameters are now states.

3.3.1 Relative Motion Dynamics

While the geometry of the target object model is not known at the beginning of the estimation, the object itself is assumed to be a rigid body and therefore, its position and orientation in the camera coordinate frame can be represented by three position and three rotational parameters comprising the relative pose vector, $\mathbf{w}$.

As with the previous estimation when the object model was known, the relative motion dynamics are assumed to be well-represented by a constant-velocity model. Accordingly, the system state vector is composed of the six pose parameters and their associated velocities. Therefore, let $\mathbf{x}_k = [\ X\ Y\ Z\ \phi\ \theta\ \psi\ \dot{X}\ \dot{Y}\ \dot{Z}\ \dot{\phi}\ \dot{\theta}\ \dot{\psi}\ ]_k^T$ be the state vector at time step k.

The dynamics of the system can then be written in discrete-time state-space form as a function of the states,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \eta_k, \tag{3.1}$$

where η_k is again a vector of zero mean Gaussian disturbance noise with covariance $\mathbf{Q}$, and the process matrix has the form,

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{6\times 6} & \delta_t \mathbf{I}_{6\times 6} \\ \mathbf{0} & \mathbf{I}_{6\times 6} \end{bmatrix}, \tag{3.2}$$

where δ_t is sampling period. At this point, the process model is identical to that of the known object model estimation studied in the previous chapter. This is not surprising since the model parameters do not affect the relative motion dynamics between the target object and camera frames. However, in order to estimate the model parameters in addition to the system states, the parameters will be appended to the state vector and the process matrix will change slightly, as will be shown in the next section.

3.3.2 Cartesian Feature Parameterization

For this combined estimation, both the system states, as well as the model parameters are estimated simultaneously. The model parameters are the object feature point locations within the local object frame as shown in Fig. 3.2. The parameters for each feature are appended to the state vector and therefore each feature that is included in the estimation increases the length of the state vector by three elements. Augmenting the state vector to include the parameters this way is analogous to treating the parameters as *static* states. The new augmented state vector

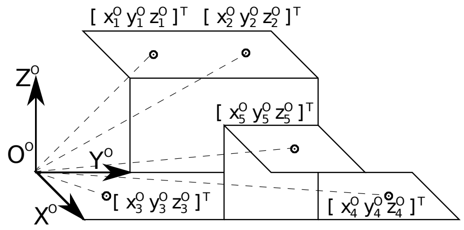

$[\ x_1^O\ y_1^O\ z_1^O\ ]^T$
$[\ x_2^O\ y_2^O\ z_2^O\ ]^T$
Z^O
$[\ x_5^O\ y_5^O\ z_5^O\ ]^T$
O^O
Y^O
$[\ x_3^O\ y_3^O\ z_3^O\ ]^T$
X^O
$[\ x_4^O\ y_4^O\ z_4^O\ ]^T$

Figure 3.2: In the Cartesian object feature point parameterization, the point feature locations (black dots) are expressed by their coordinates in the local object coordinate frame for an example target object.

is $\tilde{\mathbf{x}} = [\ \mathbf{x}^T\ x_1^O\ y_1^O\ z_1^O\ ...\ x_j^O\ y_j^O\ z_j^O\ ...\ x_N^O\ y_N^O\ z_N^O\ ]^T$ with the feature point parameters appended for the N feature points.

The process matrix must also be adjusted to accomodate the dynamics of these augmented states. The dynamics of the states representing the model parameters are defined to be static. When these parameters are added to the relative motion dynamics, this makes the full system dynamics,

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_{k-1} + \tilde{\eta}_k, \tag{3.3}$$

where $\tilde{\eta} = [\eta_k^T\ \mathbf{0}_{3N\times 1}^T]^T$ is a vector of zero-mean Gaussian disturbance noise with covariance $\mathbf{Q}$, and

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{3N\times 3N} \end{bmatrix}, \tag{3.4}$$

with $\mathbf{A}$ as defined previously in equation (3.2).

By specifying that the model parameters are not subject to disturbance noise, these appended states should converge to the correct values and remain there once the initial covariance, specified at the initialization phase, is reduced with subsequent measurement update steps of the EKF. This is the difference between the relative pose velocity states and the model parameter states – the relative velocities are subject to disturbance noise which dictates that they will not necessarily converge to one value and remain there. The disturbance noise serves to perturb these values at each time step by injecting some uncertainty at the prediction step that the measurement update step can use to affect change in their values. With the model parameters, on the other hand, their covariance will be monotonically decreasing [19] with successive measurements and will eventually become fixed. This behaviour can lead to challenges when recovering the correct global scale.

3.3.3 Single Monocular Camera Model

The measurement model utilized in the estimation is once again that of the pin-hole camera subject to zero-mean Gaussian measurement noise,

$$\mathbf{z}_k = \mathbf{g}(\mathbf{x}_k) + \nu_k, \tag{3.5}$$

but the object feature point positions, $\zeta_j^O = [\ x^O\ y^O\ z^O\ ]_j^T$, are now part of the state vector.

The camera frame is defined as previously in Section 2.3.2, for each feature point j,

$$\mathbf{g}_j = \left[\ \frac{x^i}{z^i}\ \frac{y^i}{z^i}\ \right]_j, \tag{3.6}$$

where

$$\begin{bmatrix} x^i \\ y^i \\ z^i \end{bmatrix}_j = \mathbf{\Phi}\mathbf{T}_O^C\tilde{\zeta}_j^O. \tag{3.7}$$

Since the EKF is used to estimate both the relative pose and target model parameters, the measurement model must be linearized at every time step as part of the measurement update step. The pin-hole camera equations are linearized about the current state estimate, which in this case, contains the current estimate of the target model. As a result, the partial derivative of the output equations with respect to the each model parameter is calculated and becomes another column in the measurement Jacobian, $\mathbf{C}_k$. The expressions for the elements of $\mathbf{C}_k$ are derived in Appendix A.3 using the *MATLAB* software package.

3.3.4 Inverse Depth Feature Parameterization

Within the SLAM framework, Montiel *et al.* in [43] present a useful parameterization for feature points measured using bearing-only sensors which more effectively reflects the relative uncertainty arising from these measurement systems, compared with the Cartesian feature point parameterization.

Instead of parameterizing the object feature point positions using the three Cartesian values, they are represented in the local object frame as the sum of an *initial observation point* and an *observation ray*,

$$\begin{bmatrix} x_j^O \\ y_j^O \\ z_j^O \end{bmatrix} = \begin{bmatrix} p_j^O \\ q_j^O \\ r_j^O \end{bmatrix} + \frac{1}{\tau_j}\begin{bmatrix} \cos\gamma_j \sin\chi_j \\ \sin\gamma_j \sin\chi_j \\ \cos\chi_j \end{bmatrix}, \tag{3.8}$$

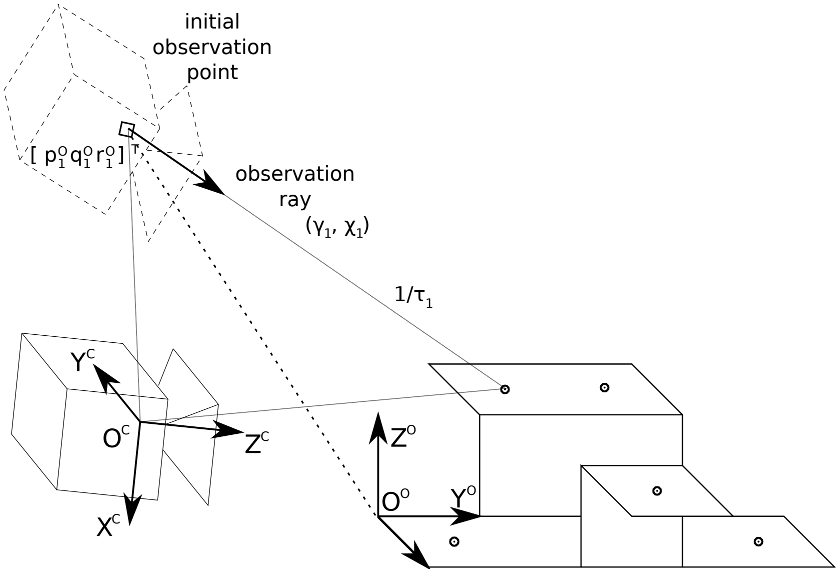


Figure 3.3: Each feature point is represented by the camera center coordinates in the object frame where the feature was first observed $(p^O_j,\ q^O_j,\ r^O_j)$, the bearing to the point $(\gamma_j,\ \chi_j)$, and the inverse of the depth along that bearing, τ_j.

where $[\ p^O_j\ q^O_j\ r^O_j\ ]^T$ is the position of the camera center in the object frame when the feature point was first observed, γ_j and χ_j are the azimuth and altitude angles respectively, to the feature point from this first observation point, and $\frac{1}{\tau_j}$ is the depth along those bearings to the feature point. It is important to note that the azimuth and altitude angles are specified with respect to the object coordinate frame. This parameterization is shown in Fig. 3.3.

By using this parameterization, the feature point can be more effectively estimated using the EKF, since the resulting measurement model has better linearity properties than when the feature points parameters are the Cartesian coordinates. Since the model linearizations are valid over a larger region, propogating the state estimates through the process and measurement equations results in more Gaussian-like distributions and the filter is able to provide more accurate estimates [43].

As a consequence of parameterizing the feature point with the inverse depth, feature points *at infinity*, those which are infinitely far away from the camera, can be represented with $\tau = 0$. When a feature point at infinity is tracked, only relative orientation information is available from such measurements.

Finally, the initialization of new points into the EKF framework can be done with no prior knowledge of feature locations, and on the first observation of that new feature point. Within SLAM, there is interest in being able to add new feature points during the estimation. With previous monocular camera systems, initializing a new feature point had to be done in a separate batch optimization over several observations to try and reduce the uncertainty associated with the feature point Cartesian coordinates such that the EKF could correctly estimate the location [12] [21]. Without a fairly accurate initial estimate, the filter would diverge and the tracking operation would fail.

In order to calculate these initial azimuth and altitude angles for a new feature point, j, given an initial pair of image-plane measurements, x^I_{j0} and y^I_{j0}, the bearing to the feature point in the camera frame is,

$$\mathbf{h}^C_j = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \mathbf{R}^C_O \begin{bmatrix} \frac{-x^I_{j0}}{F_x} \\ \frac{-y^I_{j0}}{F_y} \\ 1 \end{bmatrix}. \tag{3.9}$$

Then, the initial azimuth and altitude angles relative to the object frame are found by,

$$\gamma_{j0} = \tan^{-1}\left(\frac{h_y}{h_x}\right) \tag{3.10}$$

and

$$\chi_{j0} = \tan^{-1}\left(\sqrt{h_x^2 + h_y^2}\right). \tag{3.11}$$

This parameterization comes at the cost of using six parameters for each point feature instead of three as with the Cartesian coordinate case. However, once the location of the feature is known with relative certainty and the distribution associated with the estimated position becomes a sharp peak about the estimated mean, the parameterization can be reduced to only the three Cartesian coordinates in the object frame [43]. This transformation will not be considered in this work, but would aid in practical implementations of the algorithms by reducing the dimension of the state vector and therefore, the computational cost of the filter. The problem of how and when to convert the inverse depth parameters to the Cartesian representation is addressed in [10].

The strength of this parameterization is the ability to effectively represent and correctly propogate a large uncertainty in the depth of the object feature points. As a result, this parameterization will be used for the target model estimation schemes presented in the remaining sections.

3.3.5 Relative Orientation Parameterization

In many previous position-based visual servoing systems [53] [28] [17], the relative orientation of the object frame with respect to the camera frame was represented using Euler angles, (ϕ, θ, ψ) for roll, pitch, and yaw. The corresponding rotation matrix $\mathbf{R}_O^C$ is composed,

$$\mathbf{R}_O^C = \mathbf{R}(\phi, \theta, \psi) = \mathbf{R}_{Z^C}(\psi)\mathbf{R}_{Y^C}(\theta)\mathbf{R}_{X^C}(\phi), \tag{3.12}$$

where $\mathbf{R}_{Z^C}(\psi)$ indicates rotation about the camera frame Z^C-axis by an angle ψ.

Recall, when the relative motion is modeled as a constant velocity dynamical system, the process model used in the pose estimation scheme is,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \eta_k, \tag{3.13}$$

where,

$$\mathbf{x}_k = [\ X\ Y\ Z\ \phi\ \theta\ \psi\ \dot{X}\ \dot{Y}\ \dot{Z}\ \dot{\phi}\ \dot{\theta}\ \dot{\psi}\ ]_k^T, \tag{3.14}$$

and

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{6\times6} & \delta_t \mathbf{I}_{6\times6} \\ \mathbf{0} & \mathbf{I}_{6\times6} \end{bmatrix}, \tag{3.15}$$

δ_t is the sampling period, and $\mathbf{I}_{6\times6}$ is the identity matrix.

In this case, the process model is linear and can be used directly in the EKF framework without first having to linearize about the current state estimate.

However, when using Euler angles to represent the relative orientation, there are singularities when,

$$\theta = \frac{\pi}{2} + \pi n, n \in \mathbb{Z}. \tag{3.16}$$

At these configurations, the axes of the first and third rotations are aligned and the system loses a degree of freedom. A rotation matrix can be formed for all values of ϕ and ψ, however, the inverse mapping is not unique. In the context of visual servoing, this leads to the system becoming locally unobservable since the measurement Jacobian, $\mathbf{C}_k$, loses rank and the filter can diverge.

In an effort to remove this issue, Montiel *et al.* [43] and Civera *et al.* [9] in the context of mobile robot localization, have used unit quaternions to represent the relative orientation of the camera frame with respect the static world frame. This is similar to the pose representation for the position-based visual servoing.

The relative orientation of the object frame with respect to the camera frame can be parameterized by the unit quaternion [22], $\mathbf{q} = [\ q_0\ q_1\ q_2\ q_3\ ]$, where the elements are subject to the following constraints,

$$\mathbf{q} = q_0 + q_1\hat{\mathbf{i}} + q_2\hat{\mathbf{j}} + q_3\hat{\mathbf{k}}, \tag{3.17}$$

$$\hat{\mathbf{i}}^2 = \hat{\mathbf{j}}^2 = \hat{\mathbf{k}}^2 = \hat{\mathbf{i}}\hat{\mathbf{j}}\hat{\mathbf{k}} = -1, \tag{3.18}$$

$$\mathbf{q} = \cos(\frac{\alpha}{2}) + \hat{\mathbf{n}}\sin(\frac{\alpha}{2}) \tag{3.19}$$

$$q_0 = \cos(\frac{\alpha}{2}), [\ q_1\ q_2\ q_3\ ]^T = \hat{\mathbf{n}}\sin(\frac{\alpha}{2}), \tag{3.20}$$

where α is the angle of rotation about the $\hat{\mathbf{n}}$ axis.

The rotation matrix $\mathbf{R}_O^C$ can be formed using the unit quaternion representation by,

$$\mathbf{R}_O^C = \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_1q_3+2q_0q_2 \\ -2q_1q_2+2q_0q_3 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1 \\ -2q_1q_3-2q_0q_2 & -2q_2q_3+2q_0q_1 & 1-2q_1^2-2q_2^2 \end{bmatrix}. \tag{3.21}$$

The angular velocity is represented by the 3-tuple $[\ \omega_x\ \omega_y\ \omega_z\ ]^T$. The state vector at time step k for the pose estimation system with unit quaternions becomes,

$$\mathbf{x}_k = \begin{bmatrix} X & Y & Z & q_0 & q_1 & q_2 & q_3 & \dot{X} & \dot{Y} & \dot{Z} & \omega_x & \omega_y & \omega_z \end{bmatrix}_k^T. \tag{3.22}$$

This system does not suffer from the same singularities as the Euler angles-based estimation but has some disadvantages in this context. Observe that the relative pose vector now contains 13 states compared to 12 previously.

When the constant velocity model is assumed to represent the relative motion, the process model is more complicated than in the Euler angle case and is, in fact, nonlinear,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \delta_t) + \eta_k, \tag{3.23}$$

of which the quaternion transition is,

$$\mathbf{q}_k = (\mathbf{q}\mathbf{q}_\omega)_{k-1}, \tag{3.24}$$

where

$$\mathbf{q}_\omega = \begin{bmatrix} \cos\left(\frac{|\omega\delta_t|}{2}\right) & \frac{\sin\left(\frac{|\omega\delta_t|}{2}\right)}{|\omega|}[\omega_x\ \omega_y\ \omega_z] \end{bmatrix}_{k-1}^T, \tag{3.25}$$

and

$$|\omega| = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}. \tag{3.26}$$

This transition of the unit quaternion between time steps is a *nonlinear process*, and therefore, must be linearized at each filter recursion to use the EKF framework.

This, however, can lead to problems since the magnitude of the angular velocity vector, ω appears in the denominator of many of the elements of the process Jacobian, $\mathbf{F}_k$. This is problematic since zero angular velocity, $|\omega| = 0$, is a common operating condition, especially when a visual servoing controller is regulating about a fixed relative pose.

Despite this significant drawback, this parameterization has seen widespread use, especially within the SLAM community [14] [43] [9]. A simple solution used by these systems is to assign a very small value initially to angular velocity and count on the singularity never being *exactly* reached,

$$\omega_0 = [\ \epsilon\ \epsilon\ \epsilon\ ]^T,\ \epsilon > 0. \tag{3.27}$$

This is a temporary fix and there is still a real possibility of hitting the singularity in normal operating circumstances. At the very least, it could lead to numerical stability problems due to poor conditioning.

Crassidis and Markley [11] refer to other attempted solutions to this issue, but further point out that the EKF will not ensure that the unit quaternion remains normalized. Using four parameters to represent orientation means that these values will be dependent, and a parameterization with three values would be more useful. Accordingly, the authors in [11] uses Modified Rodriques Parameters (MRP) to represent attitude estimates made with an EKF.

The MRP are defined using the unit quaternion parameters as,

$$\mathbf{p} = [\ a\ b\ c\ ]^T = \frac{1}{1+q_0}[\ q_1\ q_2\ q_3\ ]^T = \hat{\mathbf{n}} \tan\left(\frac{\alpha}{4}\right), \tag{3.28}$$

of which the derivative is

$$\dot{\mathbf{p}} = \frac{1}{2}\left(\frac{1}{2}\left(1 - \mathbf{p}^T\mathbf{p}\right)\mathbf{I}_{3\times3} + [\mathbf{p}\times] + \mathbf{p}\mathbf{p}^T\right)\omega, \tag{3.29}$$

where $[\mathbf{a}\times]$ is the skew-symmetric matrix formed by vector $\mathbf{a}$,

$$[\mathbf{a}\times] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \tag{3.30}$$

and ω is the angular velocity vector as before.

Using MRP, the relative pose system, again assuming constant velocity is,

$$\mathbf{x} = \begin{bmatrix} X & Y & Z & a & b & c & \dot{X} & \dot{Y} & \dot{Z} & \omega_x & \omega_y & \omega_z \end{bmatrix}^T, \tag{3.31}$$

with dynamics,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \delta_t), \tag{3.32}$$

where the MRP transition is,

$$[\ a\ b\ c\ ]_k^T = [\ a\ b\ c\ ]_{k-1}^T + \delta_t \dot{\mathbf{p}}_{k-1}, \tag{3.33}$$

which is again a nonlinear function of the states and must be linearized about the current state estimate at each time step for use with the EKF. However, this process Jacobian, $\mathbf{F}_k$, does not lose rank when $\omega_k = \mathbf{0}$.

While the MRP eliminate the former singularities in the process model, they do suffer from another singularity which occurs when,

$$\alpha = 2\pi + 4\pi n, n \in \mathbb{Z}. \tag{3.34}$$

At this point, the denominator of the MRPs is,

$$1 + q_0 = 1 + \cos\left(\frac{\alpha}{2}\right) = 1 + \cos\pi = 0. \tag{3.35}$$

However, if the angle is kept on the interval $(-\pi,\ \pi]$, all relative orientations can be represented. Further, if the orientation MRP are reset to this interval at each timestep, then the system will never hit the singularity as long as an interframe rotation is less than π radians, or 180°. This should be a reasonable assumption since such rotations would almost certainly violate the linearity assumptions necessary to use the EKF framework in the first place. Otherwise, the other option is to increase the sampling rate of the system by using a camera with a faster frame rate.

The MRPs are better-suited for the EKF framework than unit quaternions since there are three instead of four parameters for relative orientation, the parameters are not dependent and do not need to remain normalized, and there are not singularities with zero angular velocity.

3.3.6 Object Frame Redefinition

Without any knowledge of the object model feature point locations, there are unconstrained degrees of freedom in the object frame relative pose estimate and its initial relative position and orientation with respect to the camera frame is arbitrary. This is a consequence of the gauge freedoms identified in [15] and [35]. Using only the information from the first frame of the image sequence to initialize the

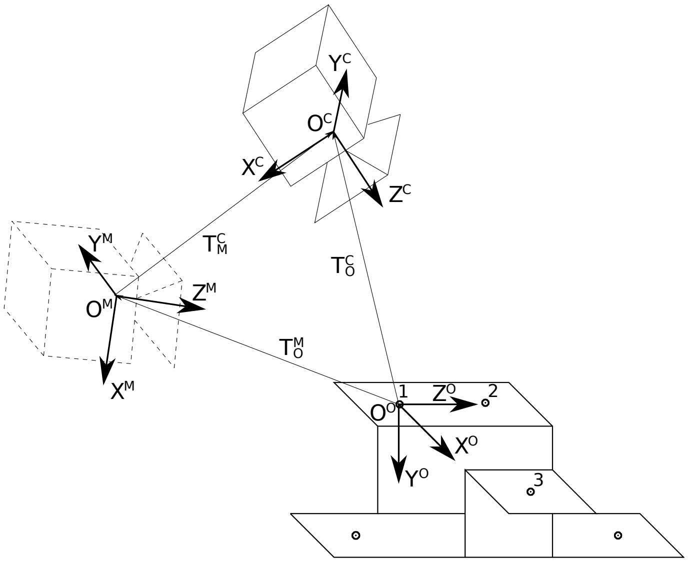


Figure 3.4: The object frame is defined with respect to three selected feature points. The Modelling frame is initially superimposed on the camera frame and the estimated target model feature point parameters are made with respect to this M frame. Furthermore, the relative pose is now the position and orientation of the M frame with respect to the C frame.

estimation does not give any measure of the depth of any of the feature points and therefore, specifying an accurate initial position of the object frame origin with respect to the camera frame is difficult. If the object frame were initially assigned some relative pose, the estimated object feature point locations may converge far from the object frame origin, which makes it awkward when representing object orientation and could amplify small rotations into large transformations due to the apparent motion of the frame origin.

Instead, the object frame can be defined with respect to feature locations on the target object [17] which would simplify specifying relative tasks. This requires the introduction of a third coordinate frame to act as an intermediary between the camera and this new object frame as depicted in Fig. 3.4. The initial feature location uncertainty is effectively propogated by the inverse depth parameterization. In order to use this representation, the position of the camera center at the first observation of the feature points needs to be captured. Accordingly, the *Modelling (M) coordinate frame* is initially assumed to be *superimposed* on the camera frame.

At the beginning of the estimation, all observed feature points are initialized to have an observation point of identically $[\ p_j^M\ \ q_j^M\ \ r_j^M\ ]^T = [\ 0\ \ 0\ \ 0\ ]^T$. The point feature locations are estimated as before, but in the M frame. Furthermore, the relative pose parameters now represent the relative position and orientation of the M frame with respect to the camera C frame. Since these two frames are *defined* as being *initially* superimposed, the initial relative pose parameters have no initial uncertainty, quantified by an associated estimate covariance of zero. Recall the result of Dissanayake *et al.* asserting that the final estimation covariance is bounded below by the initial covariance of the relative pose parameters [19]. With this representation, it is made explicit that only the relative pose is what is important and, accordingly, the final model and pose can be recovered perfectly in the limit.

The object O frame can now be redefined in terms of three point features on the target object itself with coordinates expressed in the M frame. Assume that there exists three non-collinear feature points on the target object (ζ_1^M, ζ_2^M, and ζ_3^M) for which the coordinates are expressed in the M frame. This property can be verified in the image plane since collinear points in M will be collinear in the image plane as well. The new object frame will be formed from the three feature points using the following rules: the first feature point, ζ_1^M, specifies the origin of the coordinate frame, O^O; the second, ζ_2^M, lies on the Z^O-axis; and the third feature point, ζ_3^M, lies within the X^O-Z^O plane. With these three points, all six degrees of freedom in the placement of the object frame can be fixed.

Next, define the object frame as an orthonormal basis, such that the transformation from the object frame to the M frame is written,

$$\mathbf{T}_O^M = \begin{bmatrix} \hat{\mathbf{i}}_O^M & \hat{\mathbf{j}}_O^M & \hat{\mathbf{k}}_O^M & \zeta_1^M \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.36}$$

where $\hat{\mathbf{i}}_O^M$, $\hat{\mathbf{i}}_O^M$, $\hat{\mathbf{i}}_O^M$ for the orthonormal basis as unit vectors in the directions of the X^O, Y^O, and Z^O axes respectively,

$$\hat{\mathbf{i}}_O^M = \frac{\mathbf{i}_O^M}{|\mathbf{i}_O^M|}, \tag{3.37}$$

$$\hat{\mathbf{k}}_O^M = \frac{\mathbf{k}_O^M}{|\mathbf{k}_O^M|}, \tag{3.38}$$

with,

$$\mathbf{k}_O^M = \zeta_2^M - \zeta_1^M \tag{3.39}$$

$$\mathbf{i}_O^M = (\zeta_2^M - \zeta_1^M) - ((\zeta_2^M - \zeta_1^M)^T \hat{\mathbf{k}}_O^M)\hat{\mathbf{k}}_O^M \tag{3.40}$$

$$\hat{\mathbf{j}}_O^M = \hat{\mathbf{k}}_O^M \times \hat{\mathbf{i}}_O^M. \tag{3.41}$$

In this configuration, the relative position and orientation of the new object frame with respect to the camera frame can be found by the following transformation,

$$\mathbf{T}_O^C = \mathbf{T}_M^C \mathbf{T}_O^M. \tag{3.42}$$

With this new scheme, the system produces estimates of the relative position and orientation of the modelling frame with respect to the camera frame. The object feature point locations in M are estimated as previously and the new object frame within which the robotic operations are performed, can be found at any time using the estimates of the three selected feature points to form the transformation, $\mathbf{T}_O^M$.

3.3.7 Summary

Integrating all of the modifications detailed above, the resulting combined state and parameter estimation system has a different form than previously written. While significant changes are made in the state vector itself, most of the differences with the previous system are in the *interpretation* of the parameters. This section will attempt to tie all of these changes together into a cohesive framework.

The camera (C) frame is defined to be initially superimposed with the modelling (M) frame, and the relative pose parameters encode the relative position and orientation of the M frame with respect to the C frame,

$$\mathbf{x} = [\ X\ Y\ Z\ a\ b\ c\ \dot{X}\ \dot{Y}\ \dot{Z}\ \omega_x\ \omega_y\ \omega_z\ ]^T, \tag{3.43}$$

which together can be used to define the homogeneous transformation from the M frame to the C frame,

$$\mathbf{T}_M^C = \begin{bmatrix} \mathbf{R}_M^C(a,b,c) & \mathbf{t}_M^C \\ \mathbf{0} & 1 \end{bmatrix}, \tag{3.44}$$

where

$$\mathbf{t}_M^C = [\ X\ Y\ Z\ ]^T. \tag{3.45}$$

Subsequently, the target model feature point parameters expressed in the M frame, are appended to the state vector such that they are now estimated along side the relative pose states,

$$\tilde{\mathbf{x}} = [\ \mathbf{x}^T\ ...\ p_j^M\ q_j^M\ r_j^M\ \gamma_j\ \chi_j\ \tau_j\ ...\ ]^T, \tag{3.46}$$

forming the augmented state vector for the combined state and parameter estimation system.

While the dynamics of the target object model parameters remain stationary, the relative motion dynamics, using the Modified Rodrigues Parameters, are now nonlinear,

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \eta_k \tag{3.47}$$

leading to the process matrix used at each time step k,

$$\tilde{\mathbf{A}}_k = \begin{bmatrix} \mathbf{F}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{6n \times 6n} \end{bmatrix}, \tag{3.48}$$

where

$$\mathbf{F}_k = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_k} \tag{3.49}$$

is the first-order Taylor series expansion of the relative pose dynamics, $\mathbf{f}(\mathbf{x})$.

The measurement model changes as well to reflect the new parameterizations. First, the transformation of the object feature points from the M frame to the C is written,

$$\begin{bmatrix} x_j^C \\ y_j^C \\ z_j^C \\ 1 \end{bmatrix} = \mathbf{T}_M^C \left(\begin{bmatrix} p_j^M \\ q_j^M \\ r_j^M \\ 1 \end{bmatrix} + \frac{1}{\tau_j} \begin{bmatrix} \cos\gamma_j \sin\chi_j \\ \sin\gamma_j \sin\chi_j \\ \cos\chi_j \\ 0 \end{bmatrix} \right). \tag{3.50}$$

Then, the now familiar projection onto the camera image plane is found,

$$x_j^I = -F_x \frac{x_j^C}{z_j^C} \tag{3.51}$$

$$y_j^I = -F_y \frac{y_j^C}{z_j^C}. \tag{3.52}$$

Consequently, the measurement model must now be linearized about the current state estimate, using the new parameterizations, for use in the EKF framework. Both the process and measurement Jacobians for this system are found in Appendix A.4 and Appendix A.5 respectively.

This combined relative pose and target model estimation system can now be recursively solved by the EKF using the procedure outlined in Section 2.3. The final requirement is to seed the filter with reasonable initial estimates of the unknown system.

Initialization

Similar to many of the previous localization systems, the target model can be initialized with a subset of the feature points at known locations. The specific initialization values depend on whether the target object model is assumed to be partially known or completely unknown. A known object model actually places more constraints on the *initial relative pose* estimate, whereas for a target object with a completely unknown feature point set, the choice of initial pose is arbitrary. Only the completely unknown case is investigated here.

Unknown Target Model

All of the necessary information for the initial estimate can be derived from the first set of measurements. As they were defined in Section 3.3.6, the relative pose parameters express the relative position and orientation of the modelling M frame with respect to the camera C frame. By this definition, the two frames are superimposed on the start of the estimation and all of the pose parameters are represented by Gaussian distributions with zero mean and zero covariance,

$$X_0 = 0,\ \sigma^2_{X0} = 0 \tag{3.53}$$

$$Y_0 = 0,\ \sigma^2_{Y0} = 0 \tag{3.54}$$

$$Z_0 = 0,\ \sigma^2_{Z0} = 0 \tag{3.55}$$

$$a_0 = 0,\ \sigma^2_{a0} = 0 \tag{3.56}$$

$$b_0 = 0,\ \sigma^2_{b0} = 0 \tag{3.57}$$

$$c_0 = 0,\ \sigma^2_{c0} = 0. \tag{3.58}$$

The pose velocity parameters are also set to zero-mean Gaussian distribution, but with a suitable set of covariance values to characterize the confidence.

Finding initial estimates for the feature point locations is slightly more complicated, but follows the same idea. If it is assumed that all of the object feature points are visible in the first image, the initial measurements from the first camera image provide a reasonable bearing to the feature point location. Additionally, the initial observation point for the features is identically zero, just like the initial pose parameters. As a result, five of the six feature point parameters can be initialized to reasonable values using the measurements from the first camera image alone.

$$p^M_{j0} = 0,\ \sigma^2_{p_j0} = 0 \tag{3.59}$$

$$q^M_{j0} = 0,\ \sigma^2_{q_j 0} = 0 \tag{3.60}$$

$$r^M_{j0} = 0,\ \sigma^2_{r_j 0} = 0 \tag{3.61}$$

with the initial azimuth and altitude angles γ_{j0} an χ_{j0} defined using equations (3.10) and (3.11), and the covariances associated with these angles, $\sigma^2_{\gamma_j 0}$ an $\sigma^2_{\chi_j 0}$, are set using the 95% confidence interval method.

The final feature parameter for the inverse depth, τ, is where the majority of the uncertainty resides in the single camera estimation scheme. While it is possible to specify a large uncertainty range covering from centimetres in front of the camera all the way out to infinity, using prior knowledge about the working conditions may help with convergence. Assumptions about the range of depths for object features can be quantitatively represented as a mean and covariance pair. Suppose that all of the object feature points were assumed to have an initial image depth on $[\ d_{min},\ d_{max}\ ]$ with a 95% confidence interval, where $0 < d_{min} \leq d_{max}$,

$$\tau_{min} = \frac{1}{d_{max}},\ \tau_{max} = \frac{1}{d_{min}}, \tag{3.62}$$

$$\tau_{max} - \tau_{min} = 4\sigma_{\tau_{j0}}, \tag{3.63}$$

$$\sigma^2_{\tau_{j0}} = \left(\frac{d_{max} - d_{min}}{4d_{min}d_{max}}\right)^2, \tag{3.64}$$

$$\tau_{j0} = \frac{1}{d_{max}} + 2\sigma_{\tau_{j0}} = \frac{d_{min} + d_{max}}{2d_{max}d_{min}}. \tag{3.65}$$

3.4 Observability Analysis

In order for the EKF to produce a reasonable estimate for the system states, which in this case are the relative pose, the relative velocity, and the target object model parameters, the system must be completely observable, or at least be operated in an observable region of the state space. The relative pose system uses a nonlinear measurement model and therefore, the classical linear observability concepts can not be trivially applied in this situation.

Following in the example of Vidal-Calleja *et al.* in the context of a planar SLAM system with a vehicle dynamics process model, in [51], this system is approximated as a discrete *Piece-Wise Constant System* (PWCS) [24].

3.4.1 Discrete Piece-Wise Constant Systems

A discrete-time PWCS can be written in the following form [24],

$$\mathbf{x}_k = \mathbf{A}_j \mathbf{x}_{k-1} + \mathbf{B}_j \mathbf{u}_{k-1} \tag{3.66}$$

$$\mathbf{z}_k = \mathbf{C}_j \mathbf{x}_k, \tag{3.67}$$

where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{A}_j \in \mathbb{R}^{n \times n}$, $\mathbf{u}_{k-1} \in \mathbb{R}^s$, $\mathbf{B}_j \in \mathbb{R}^{n \times s}$, $\mathbf{z}_k \in \mathbb{R}^m$, $\mathbf{C}_j \in \mathbb{R}^{m \times n}$ and $j = 1, 2, ..., r$. Within each *time segment*, j, the system matrices, $\mathbf{A}_j$, $\mathbf{B}_j$, and $\mathbf{C}_j$ are constant in their elements.

In this analysis, each time segment, j, corresponds with the time step, k. In linear systems observability analysis, the observability of a given system is not affected by the excitations or deterministic inputs. However, this is not necessarily the case for nonlinear systems [27]. The relative motion system is *autonomous* and therefore does not include deterministic inputs. The (noiseless) linear system matrices change at every time step as the linearization of the nonlinear process and measurement equations about the current state. The resulting PWCS can be written, with a slight notation change to reflect each time segment containing only one time step, as,

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{x}_{k-1} \tag{3.68}$$

$$\mathbf{z}_k = \mathbf{C}_k \mathbf{x}_k. \tag{3.69}$$

The observability of this linear system depends on the set of $\mathbf{A}_k$ and $\mathbf{C}_k$ matrices over the discrete time steps. The observability matrix at each individual time segment (time step, in this case) is,

$$\mathbf{Q}_k^T = \left[\mathbf{C}_k^T \; (\mathbf{C}_k \mathbf{A}_k)^T \; ... \; (\mathbf{C}_k \mathbf{A}_k^{(n-1)})^T\right]^T. \tag{3.70}$$

Stacking the individual observability matrices from each time segment forms the *total observability matrix* (TOM) [24] over all r segments,

$$\mathbf{Q}(r) = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \mathbf{A}_1^{n-1} \\ \mathbf{Q}_3 \mathbf{A}_2^{n-1} \mathbf{A}_1^{n-1} \\ ... \\ \mathbf{Q}_r \mathbf{A}_{r-1}^{n-1} \mathbf{A}_{r-2}^{n-1} ... \mathbf{A}_1^{n-1} \end{bmatrix}. \tag{3.71}$$

From this, it can be shown that the PWCS is *completely observable* if and only if the TOM has rank n. When the system is not completely observable, it will be said to be *unobservable*, as done previously.

Goshen-Meskin *et al.* [24] also introduce a simpler object called the *stripped observability matrix* (SOM) which drops the process matrices, $\mathbf{A}_k$, from the TOM for the case where all of the unobservable modes of the system are associated with an eigenvalue of 1. A relative pose dynamics model which uses the constant velocity approximation, such as the one used in this analysis, satisfies this requirement trivially since *all* of the modes of the system are associated with eigenvalue 1. This is evident from the structure of $\mathbf{A}_k$ in the system.

The SOM over r time segments, $\mathbf{Q}_s(r)$ is then simply,

$$\mathbf{Q}_s(r) = \begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \cdots \\ \mathbf{Q}_r \end{bmatrix}. \tag{3.72}$$

Obviously, the SOM (or TOM) where $r = 1$ reduces to the familiar linear control system observability matrix. In this single time segment case, if the matrix has rank n, the system is completely observable in the first time segment and there is no need to consider further time segments. However, if this is not the case, more time segments can be used to see if and when the PWCS becomes completely observable. Additionally, the observability matrices can be analyzed to discover which modes are unobservable and what kinds of measurements must be supplemented to make the system observable.

3.4.2 One Degree of Freedom

Recall the problem of estimating the position of a feature point with respect to a camera frame in a planar environment where the relative position is free to vary in only one dimension from Section 2.4.1. Previously, it was shown that *in almost all configurations*, when the X^C-axis coordinate of the point feature was known with certainty, it was possible to estimate the relative depth of a feature point with respect to the camera frame. The analysis is now extended to the case where the x^O parameter is unknown and thus must be estimated.

Observability with Unknown Model

If there is no prior knowledge of the camera X^C-axis position of the single feature point, it is now also part of the estimation scheme along with the Y^C-axis motion,

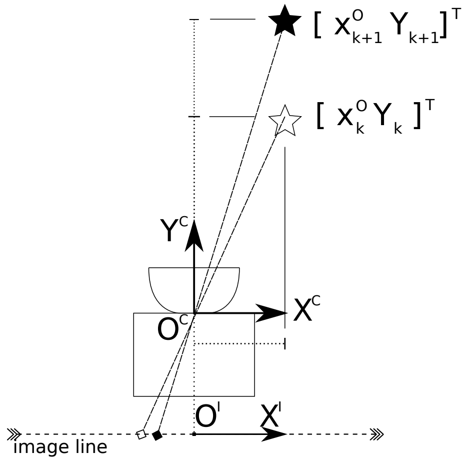


Figure 3.5: The setup for the one degree-of-freedom relative pose estimation when the feature point location is completely unknown. Two time steps of the relative motion are shown (white to black stars).

shown in Fig. 3.5. This problem becomes a *combined state and parameter estimation*. The unknown parameter, x^O is appended to the state vector to form a new augmented system where the dynamics of the parameter dictate it is stationary,

$$\tilde{\mathbf{x}} = [\ Y\ \dot{Y}\ x^O\ ]^T \tag{3.73}$$

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_{k-1} \tag{3.74}$$

$$z_k = g(\tilde{\mathbf{x}}_k) = -\left(\frac{x^O}{Y}\right)_k, \tag{3.75}$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} 1 & \delta_t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.76}$$

For this analysis, this nonlinear system will be approximated by a PWCS, as detailed previously. Accordingly, the measurement model needs to linearized at each time step, k, as follows,

$$\tilde{\mathbf{C}}_k = \frac{\partial \mathbf{g}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}_k}} = \left[\ \frac{x^O}{Y^2}\quad 0\quad -\frac{1}{Y}\ \right]_k. \tag{3.77}$$

By assuming that the nonlinear system is approximated by the PWCS, it is assumed that this linearization will remain valid over the entire time step. In the context of the PWCS, it is then assumed that each time segment consists of a single time step and the system matrices at that time segment are formed by the linearizations at that operating point which is the true system state at that time step. There is an issue with such an assumption that will be explored in the next section.

Investigating the observability over just one time step with the $\mathbf{C}_k$ found from the linearization, the observability matrix $\tilde{\mathbf{Q}}_k$ becomes,

$$\tilde{\mathbf{Q}}_k = \begin{bmatrix} \tilde{\mathbf{C}}_k \\ \tilde{\mathbf{C}}_k\tilde{\mathbf{A}} \\ \tilde{\mathbf{C}}_k\tilde{\mathbf{A}}^2 \end{bmatrix} = \begin{bmatrix} \frac{x^O}{Y_k^2} & 0 & -\frac{1}{Y_k} \\ \frac{x^O}{Y_k^2} & \delta_t\frac{x^O}{Y_k^2} & -\frac{1}{Y_k} \\ \frac{x^O}{Y_k^2} & 2\delta_t\frac{x^O}{Y_k^2} & -\frac{1}{Y_k} \end{bmatrix}. \tag{3.78}$$

For complete observablity, this matrix must have full rank. However, the rank of this matrix is never $n = 3$, as verified by observing that the matrix is rank-equivalent through elementary row operations to,

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & -\frac{Y_k}{x^O} \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.79}$$

Following the procedure in [24], two time segments of the PWCS will now be considered to see if successive measurements can produce an observable system. Notice that since the eigenvalues of $\tilde{\mathbf{A}}$ are all 1, the SOM can be utilized for the analysis rather than the more complicated TOM. The SOM is formed by stacking the two observability matrices for the two time-segments,

$$\tilde{\mathbf{Q}}(r) = \begin{bmatrix} \tilde{\mathbf{Q}}_k \\ \tilde{\mathbf{Q}}_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{x^O}{Y^2} & 0 & -\frac{1}{Y} \\ \frac{x^O}{Y^2} & \delta_t\frac{x^O}{Y^2} & -\frac{1}{Y} \\ \frac{x^O}{Y^2} & 2\delta_t\frac{x^O}{Y^2} & -\frac{1}{Y} \\ \frac{x^O}{(Y+\delta_t\dot{Y})^2} & 0 & -\frac{1}{(Y+\delta_t\dot{Y})} \\ \frac{x^O}{(Y+\delta_t\dot{Y})^2} & \delta_t\frac{x^O}{(Y+\delta_t\dot{Y})^2} & -\frac{1}{(Y+\delta_t\dot{Y})} \\ \frac{x^O}{(Y+\delta_t\dot{Y})^2} & 2\delta_t\frac{x^O}{(Y+\delta_t\dot{Y})^2} & -\frac{1}{(Y+\delta_t\dot{Y})} \end{bmatrix}_k. \tag{3.80}$$

It is then found that this matrix is rank-equivalent through row operations to,

$$\tilde{\mathbf{U}} = \begin{bmatrix} \frac{x^O}{Y^2} & 0 & -\frac{1}{Y} \\ 0 & \delta_t\frac{x^O}{Y^2} & 0 \\ \frac{x^O}{(Y+\delta_t\dot{Y})^2} & 0 & -\frac{1}{(Y+\delta_t\dot{Y})} \end{bmatrix}. \tag{3.81}$$

This indicates that the system will be completely observable when when the determinant is non-zero, where,

$$det(\tilde{\mathbf{U}}) = -\frac{\delta_t^2 x^O \dot{Y}_k}{Y_k^4(Y_k + \delta_t\dot{Y}_k)}. \tag{3.82}$$

Therefore, the system is *unobservable* when:

1. $\dot{Y}_k = 0$ – there must be some relative motion between measurements.
2. $x^O = 0$ – the feature must not be directly aligned with camera's projection ray.
3. $|Y_k| \rightarrow \infty$ and $|Y_k + \delta_t \dot{Y}_k| \rightarrow \infty$ – observations must be from a finite depth away.

The PWCS approximation of the nonlinear system is *completely observable over two timesteps* where the configurations do not fall into the sets detailed above. It should be noted that the unobservable configurations for this unknown feature location estimation include the requirements of the known object model observability criteria (from Section 2.4.1) as necessary conditions. In other words, these combined relative pose and target model estimation systems include the unobservable configurations of the relative pose estimation systems as a subset of the unobservable configurations here.

3.4.3 Shortcomings of PWCS Analysis

When using the PWCS on a nonlinear system, it is assumed that the nonlinear relative pose and target model estimation system could be effectively approximated by a linearization at each time step. In fact, this is not absolutely true. A PWCS assumes that the process and measurement matrices are a function of the time step, but for this particular system, they are also a function of the states.

Somewhat unexpectedly, the analysis in the previous section suggests that the unknown target model system is completely observable (including global scale) over two time steps. This contradicts the previous assertion that the global scale should not be recoverable using a single bearing-only measurement device. Clearly, some information is not being captured in the approximated model.

When the measurement model was linearized to form $\mathbf{C}_k$, the operating point used is the actual trajectory of the target object with respect to the camera. In other words, the *true state* at that time step. During the estimation, however, the output model is linearized about the *current state estimate.* This discrepancy is the reason why the PWCS suggests it should be possible to recover the global scale.

To see how this can occur, consider the following. For the one-dimensional estimation above, assume that the system is not affected by noise and measurements are taken from three distinct relative positions.

$$z_1 = \tilde{\mathbf{C}}_1 \tilde{\mathbf{x}}_1 \tag{3.83}$$

$$z_2 = \tilde{\mathbf{C}}_2 \tilde{\mathbf{x}}_2 \tag{3.84}$$

$$z_3 = \tilde{\mathbf{C}}_3 \tilde{\mathbf{x}}_3 \tag{3.85}$$

where

$$\tilde{\mathbf{C}}_k = \frac{\partial \mathbf{g}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}_k}} \tag{3.86}$$

with $\tilde{\mathbf{x}}_2 = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_1$, $\tilde{\mathbf{x}}_3 = \tilde{\mathbf{A}}^2\tilde{\mathbf{x}}_1$, and $\mathbf{g}(\tilde{\mathbf{x}})$ was given in Section 2.4.1.

In order to form an estimate of the system states from the measurements, an inverse mapping is required. Start by stacking the measurements into a vector,

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \hat{\mathbf{C}}\tilde{\mathbf{x}}_1 = \begin{bmatrix} \tilde{\mathbf{C}}_1 \\ \tilde{\mathbf{C}}_2\tilde{\mathbf{A}} \\ \tilde{\mathbf{C}}_3\tilde{\mathbf{A}}^2 \end{bmatrix} \tilde{\mathbf{x}}_1. \tag{3.87}$$

This augmented output matrix has the form,

$$\hat{\mathbf{C}} = \begin{bmatrix} \frac{x^O}{Y^2} & 0 & -\frac{1}{Y} \\ \frac{x^O}{(Y+\delta_t\dot{Y})^2} & \delta_t\frac{x^O}{(Y+\delta_t\dot{Y})^2} & -\frac{1}{Y+\delta_t\dot{Y}} \\ \frac{x^O}{(Y+2\delta_t\dot{Y})^2} & 2\delta_t\frac{x^O}{(Y+2\delta_t\dot{Y})^2} & -\frac{1}{Y+2\delta_t\dot{Y}} \end{bmatrix}. \tag{3.88}$$

When this $\hat{\mathbf{C}}$ matrix is invertible, the system states can be expressed as a function of the outputs,

$$\tilde{\mathbf{x}}_1 = \hat{\mathbf{C}}^{-1}\mathbf{z} \tag{3.89}$$

If the linearization of the output model is performed at an operating point which is a scaled version of the true system states, that state estimate will be scaled by an identical factor. Let the linearization occur at the operating point $\hat{\mathbf{x}} = \alpha\tilde{\mathbf{x}}$, where $\alpha \in \mathbb{R}$. The corresponding augmented output matrix $\check{\mathbf{C}}$ is a scalar multiple of $\hat{\mathbf{C}}$,

$$\check{\mathbf{C}} = \frac{1}{\alpha}\hat{\mathbf{C}} \tag{3.90}$$

and therefore the inverse will be,

$$\check{\mathbf{C}}^{-1} = \alpha\hat{\mathbf{C}}^{-1}. \tag{3.91}$$

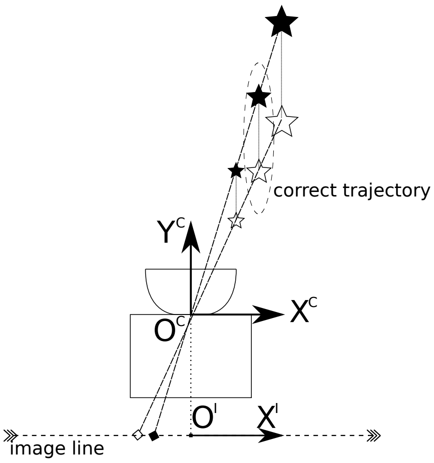


Figure 3.6: Three possible relative trajectories (from white to black stars) of the feature point with respect to the camera leading to the same image line measurements. The estimator is unable to distinguish the correct trajectory based on monocular camera measurements alone.

This result indicates that the state estimate will be a scaled version of the true states. In terms of the recursive estimation algorithms, using the current state estimate will almost certainly lead to a scaled solution for the relative pose and target model parameters. Further, if there is any error in the estimates – including round-off errors – this will lead to a change in the scale of the estimated states. There is nothing that is driving the states back to the correct global scale value.

A helpful way of looking at this situation is to think of a set of *parallel trajectories* between the camera and target object, each representing a different global scale. Fig. 3.6 shows three such two-step trajectories of a feature where the white stars are the first step and black are the second step. All three trajectories result in the same image line measurements. With perfect information about the true state and noiseless measurements, the correct trajectory is closed and the estimation will remain within it, leading to the correct solution. Estimation errors and noisy measurements can cause the states to jump between these trajectories to a new scale. There is no information in the measurements that will discriminate one such scale trajectory from the next. In this sense, more information needs to be added to the system to drive it towards the correct trajectory.

In the PWCS analysis, the *global scale is resolved* by using these true operating points. By linearizing the system about a point, the solution scale is implied by that operating point. Thus, the PWCS observability algorithm cannot directly detect all conditions when a nonlinear system is unobservable. By linearizing the system, some system information is lost. This suggests that a true nonlinear observability analysis is called for, such as is suggested in [27] investigating Lie derivatives. However, as evidenced in the previous section, there is still value in performing the PWCS analysis as it communicates a subset of the unobservable configurations without the added complexity of the nonlinear analyses.

3.4.4 2D Observability

The analysis is now extended to higher dimensions by returning to the planar relative pose system described in Section 2.4.2. Originally, the observability of the relative pose system was investigated for the case when the target object model was known with perfect certainty, and measurements of feature points were made using a single monocular camera. In this section, the target object model is assumed completely unknown. The unobservable configurations are of interest in this analysis.

Previous Results

Andrade-Cetto *et al.* assert that solving the mobile robot SLAM problem in a global reference frame is inherently unobservable [2]. Specifically, they cite to *partial observability* of the system as the cause of the results from Dissanayake *et al.* showing that the absolute accuracy of the generated map is bounded below by the initial vehicle pose uncertainty [19]. In this way, the absolute positions of the robot and the map landmarks are dependent on one another and only the combination of these values can be recovered. It is then shown that in order for the system to become fully observable, a fixed external sensor such as a GPS receiver must be added to the system, or a feature point must be selected to act as a *global localization reference.*

Vidal-Calleja *et al.* investigate the observability of mobile robot SLAM in a planar environment using a bearing-only sensor [51]. The nonlinear system, which includes a model of the vehicle dynamics, is approximated by a PWCS and the SOM is analyzed to determine its *null space.* The work looks at the observability of systems with all unknown feature point locations, some known feature points called *anchors*, and a purely relative model where the absolute positions of both the robot and feature points are disregarded. This last case is a consequence of the absolute pose of the mobile robot being unrecoverable from relative measurements of feature point locations without an external anchor of known absolute position being available [2] [19]. It is found that the final system is unobservable over two time steps if there is no translational motion, or if the robot moves directly towards an unknown feature point. In the former case, there is no *parallax* between the two observations with which to triangulate the depth of the feature point, and in the latter case, moving towards the point does not change the relative bearing to that feature point. While these results are informative, the problem is different from the relative pose estimation systems presented in this study. The principle difference being the use of the robot dynamics model, which makes these unobservable modes a subset of this which arise in the pose estimation.

These two previous works from the SLAM community rely on different process models than the generic constant velocity models used here. As a result, the observability results, while relevant to this discussion, are likely not the complete picture for this system. The next section will look at the more general results for the relative pose estimation and attempt to characterize the unobservable configurations in a geometric sense, as before.

Pose Estimation

In order to study the changes that arise when information about the target object model is not available, the system is modified in the same fashion as with the simple one dimensional system described in the previous section. As will become evident below, the additional degrees of freedom significantly complicate the observability analysis, but valuable insight can be gained in the case when the relative position and orientation of the target object model can be recovered accurately using the single camera.

A system where three feature points are observed by a line camera at each time step will again be considered, and is shown in Fig. 3.7. It is well-known that the combined estimation of both the model and relative pose can only be recovered up to a *similarity transform* [35]. As a result, a set of constraints must be imposed to allow the system to converge to a unique solution and accomodate for the *gauge freedoms* [15].

The locations of some of the feature points can be fixed with respect to the local object frame to remove these degrees of freedom, without loss of generality. It will be assumed that the first feature point is at the origin of the object frame,

$$[\ x_1^O\ y_1^O\ ] = [\ 0\ 0\ ]. \tag{3.92}$$

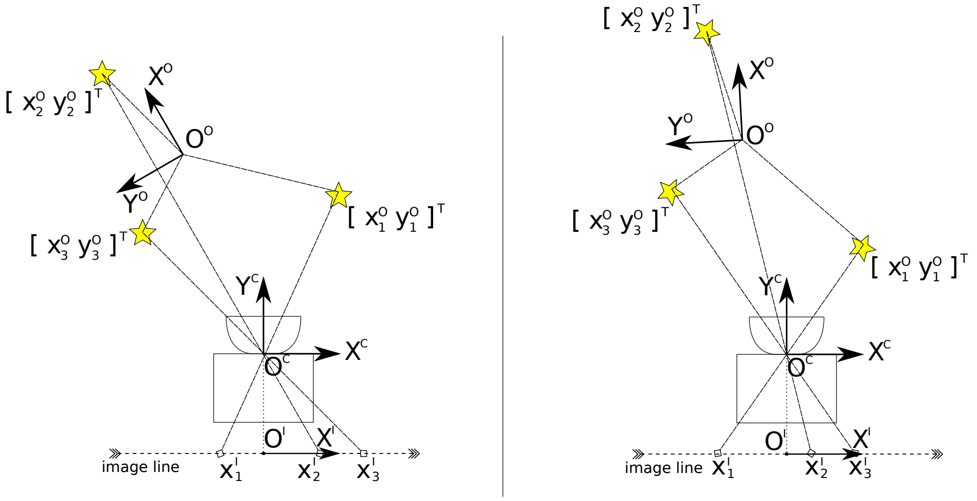


Figure 3.7: Two time steps of the planar pose estimation system with three feature points on the target object. The observabililty analysis reveals when the two sets of image line measurements are not sufficient to recover both the relative pose and model of the target object.

Additionally, it is assumed that the second feature point lies on the Y^O-axis,

$$x_2^O = 0. \tag{3.93}$$

With these three coordinates fixed, there are only three more parameters which must be estimated, and are therefore appended to the state vector,

$$\mathbf{x} = [\ X\ Y\ \theta\ \dot{X}\ \dot{Y}\ \dot{\theta}\ y_2^O\ x_3^O\ y_3^O\ ]^T. \tag{3.94}$$

These new states are considered static as reflected in the state transition for the k^{th} time step,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \eta_k, \tag{3.95}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_{3\times3} & \delta_t\mathbf{I}_{3\times3} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{3\times3} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{3\times3} \end{bmatrix}. \tag{3.96}$$

While the process model has been slightly modified to accomodate for the augmented state vector, the measurement model is unchanged, except that some of the feature point parameters are now states.

The PWCS for this planar combined relative pose and target object model estimation is,

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1}, \tag{3.97}$$

$$\Delta\mathbf{z}_k = \mathbf{C}_k\Delta\mathbf{x}_k, \tag{3.98}$$

where $\Delta\mathbf{z}_k$ and $\Delta\mathbf{x}_k$ are perturbations about the operating point at which the nonlinear measurement model was linearized.

The observability of this new system is the subject of the analysis in the following sections.

One Time Step

Consider the observability of the system using observations of the three feature points from a single time step. The linearized measurement matrix used in the PWCS, $\mathbf{C}_k$, is formed by finding the Jacobian of the measurement model evaluated at the current state, which now includes the feature point parameters,

$$\mathbf{C}_k = \frac{\partial \mathbf{g}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_k} = f\mathbf{V}_k^{-1}\left[\begin{array}{ccc} \mathbf{H}_k & \mathbf{0}_{3\times 3} & \mathbf{J}_k \end{array}\right] \tag{3.99}$$

where $\mathbf{V}_k$ is given in (2.59), $\mathbf{H}_k$ is given in (2.60), and

$$\mathbf{J}_k = \left[\begin{array}{ccc} 0 & 0 & 0 \\ x_2^{C/O} & 0 & 0 \\ 0 & -y_3^{C/O} & x_3^{C/O} \end{array}\right]_k. \tag{3.100}$$

Given the form of the measurement matrix, the product of $\mathbf{C}_k$ and the i^{th} power of $\mathbf{A}$ will be,

$$\mathbf{C}_k\mathbf{A}^i = f\mathbf{V}_k^{-1}\left[\begin{array}{ccc} \mathbf{H}_k & i\delta_t\mathbf{H}_k & \mathbf{J}_k \end{array}\right], \tag{3.101}$$

and as a result, the observability matrix at the k^{th} time step, $\mathbf{Q}_k$, is rank-equivalent through row operations to

$$\mathbf{U}_k = \left[\begin{array}{cc} \mathbf{V} & \\ & \mathbf{V} \end{array}\right]_k^{-1} \left[\begin{array}{ccc} \mathbf{H} & \mathbf{0} & \mathbf{J} \\ \mathbf{0} & \mathbf{H} & \mathbf{0} \end{array}\right]_k. \tag{3.102}$$

In this form, it is apparent that this matrix has rank($\mathbf{U}_k$) ≤ 6, which is less than the required rank of 9 for complete observability of the PWCS. Adding more unknown features will not make the system observable since each new feature adds two more states but only a single measurement at each time step. Therefore, the system is *always unobservable over one time step.*

Two Time Steps

Consider now, the observability of the system using observations of three distinct feature points over two time steps, k and $k+1$.

Theorem 2. *Assuming $f > 0$, and $\delta_t > 0$, the PWCS, (3.97) and (3.98), is unobservable over two timesteps if and only if:*

(i) $\exists j \in \{1,2,3\},\ m \in \{k, k+1\}$ *s.t.* $y^C_{j,m} \to \infty$,

or

(ii) $x^I_{1,k} = x^I_{1,k+1}$,

or

(iii) $\exists j \in \{2,3\}$ *s.t.* $\left(\mathbf{R}^C_{Ok+1}\mathbf{R}^O_{Ck}\begin{bmatrix} x^C_j \\ y^C_j \end{bmatrix}_k\right) \parallel \left(\begin{bmatrix} x^C_j \\ y^C_j \end{bmatrix}_{k+1}\right)$.

Proof. The SOM for the PWCS is constructed by stacking the two observability matrices from the successive time steps,

$$\mathbf{Q}(2) = \begin{bmatrix} \mathbf{Q}_k \\ \mathbf{Q}_{k+1} \end{bmatrix}, \tag{3.103}$$

which is rank equivalent through elementary row and column operations to,

$$\begin{aligned}\mathbf{U} &= \begin{bmatrix} \mathbf{V}_k^{-1}\mathbf{H}_k & \mathbf{0} & \mathbf{V}_k^{-1}\mathbf{J}_k \\ \mathbf{0} & \mathbf{V}_k^{-1}\mathbf{H}_k & \mathbf{0} \\ \mathbf{V}_{k+1}^{-1}\mathbf{H}_{k+1} & \mathbf{0} & \mathbf{V}_{k+1}^{-1}\mathbf{J}_{k+1} \\ \mathbf{0} & \mathbf{V}_{k+1}^{-1}\mathbf{H}_{k+1} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{V}_k & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{k+1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{V}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{V}_{k+1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{H}_k & \mathbf{J}_k & \mathbf{0} \\ \mathbf{H}_{k+1} & \mathbf{J}_{k+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_k \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_{k+1} \end{bmatrix},\end{aligned} \tag{3.104}$$

where the matrices $\mathbf{V}$, $\mathbf{H}$, and $\mathbf{J}$ are given in (2.59), (2.60), and (3.100), respectively.

The PWCS is unobservable if and only if $\mathbf{U}$ loses full rank, and the structure of this matrix dictates this happens only when *any* of the following three conditions are met,

$$\operatorname{rank}(\mathbf{U}_A) = \operatorname{rank}\left(\begin{bmatrix} \mathbf{V}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_{k+1} \end{bmatrix}^{-1}\right) < 6, \tag{3.105}$$

$$\operatorname{rank}(\mathbf{U}_B) = \operatorname{rank}\left(\begin{bmatrix} \mathbf{H}_k & \mathbf{J}_k \\ \mathbf{H}_{k+1} & \mathbf{J}_{k+1} \end{bmatrix}\right) < 6, \tag{3.106}$$

or

$$\operatorname{rank}(\mathbf{U}_C) = \operatorname{rank}\left(\begin{bmatrix} \mathbf{H}_k \\ \mathbf{H}_{k+1} \end{bmatrix}\right) < 3. \tag{3.107}$$

Consider each of these conditions individually. First, the inequality on $\mathbf{U}_A$ in (3.105) is satisfied when any of the features are infinitely far away from the line camera in the $\vec{Y}^C$-axis direction at either time step, since the zero determinant implies,

$$\left(y^C_{1,k}y^C_{2,k}y^C_{3,k}y^C_{1,k+1}y^C_{2,k+1}y^C_{3,k+1}\right)^2 \rightarrow \infty. \tag{3.108}$$

This is equivalent to (i).

The second condition, from (3.106), specifies a rank requirement on the $\mathbf{U}_B$ submatrix. By taking advantage of the zero-elements in the matrix, the zero determinant gives the equation,

$$\begin{gathered}\left(x^{C/O}_{3,k}\; y^{C/O}_{3,(k+1)} - y^{C/O}_{3,k}\, x^{C/O}_{3,(k+1)}\right) \\ \left(x^{C/O}_{2,k}\det\left(\begin{bmatrix} \mathbf{H}^{\rightarrow 1}_k \\ \mathbf{H}^{\rightarrow 2}_{k+1} \\ \mathbf{H}^{\rightarrow 1}_{k+1} \end{bmatrix}\right) - x^{C/O}_{2,(k+1)}\det\left(\begin{bmatrix} \mathbf{H}^{\rightarrow 1}_k \\ \mathbf{H}^{\rightarrow 2}_{k} \\ \mathbf{H}^{\rightarrow 1}_{k+1} \end{bmatrix}\right)\right) = 0,\end{gathered} \tag{3.109}$$

where $\mathbf{H}^{\rightarrow i}_k$ indicates the i^{th} row of $\mathbf{H}_k$, which is related to the i^{th} feature location at time step k.

The first term in (3.109) going to zero specifies,

$$\left(\mathbf{R}^C_{Ok+1}\mathbf{R}^O_{Ck}\begin{bmatrix} x^C_3 \\ y^C_3 \end{bmatrix}_k\right) \parallel \left(\begin{bmatrix} x^C_3 \\ y^C_3 \end{bmatrix}_{k+1}\right). \tag{3.110}$$

That is, the camera frame coordinates of the third feature point at time step k, rotated by $\theta_{k+1} - \theta_k$, are parallel with the camera frame coordinates of that feature point at the $k+1$ time step. This situation occurs, for instance, when the camera undergoes a pure rotation, or when the feature points are collinear and all project to the same image line position at both time steps.

The second term of (3.109) can be expanded by first substituting equations (3.92) and (3.93) into (2.60) for both time steps, k and k+1, to find the determinants of the $\mathbf{H}$ submatrices,

$$\det\left(\begin{bmatrix} \mathbf{H}^{\rightarrow 1}_k \\ \mathbf{H}^{\rightarrow 2}_{k} \\ \mathbf{H}^{\rightarrow 1}_{k+1} \end{bmatrix}\right) = y^O_2\left(Y_kX_{k+1} - X_kY_{k+1}\right)y^{C/O}_{2,k} \tag{3.111}$$

and

$$\det\left(\begin{bmatrix} \mathbf{H}_k^{\rightarrow 1} \\ \mathbf{H}_{k+1}^{\rightarrow 2} \\ \mathbf{H}_{k+1}^{\rightarrow 1} \end{bmatrix}\right) = y_2^O \left(Y_k X_{k+1} - X_k Y_{k+1}\right) y_{2,(k+1)}^{C/O}, \tag{3.112}$$

such that the $\mathbf{U}_B$ determinant also goes to zero when,

$$\begin{aligned} y_2^O \left(Y_k X_{k+1} -\right. & \left. X_k Y_{k+1}\right) \\ & \left(x_{2,k}^{C/O} y_{2,(k+1)}^{C/O} - x_{2,(k+1)}^{C/O} y_{2,k}^{C/O}\right) = 0. \end{aligned} \tag{3.113}$$

This expression indicates that the PWCS is unobservable in three additional configurations. First, when $y_2^O = 0$, the first and second point features have the same coordinates and provide redundant measurements on the image line. Three *distinct* feature points are required for the system to be observable. The next term specifies a zero determinant, and therefore, an unobservable system, can also be caused by the first feature point positioned on the same projection ray in both time steps such that,

$$\frac{X_k}{Y_k} = \frac{X_{k+1}}{Y_{k+1}}, \tag{3.114}$$

or equivalently,

$$x_{1,k}^I = x_{1,k+1}^I, \tag{3.115}$$

meaning that the image line measurement of this feature does not change between time steps. This case is presented in (ii). Finally, the last term dictates that the second feature point has the same constraints as the third, shown before in (3.110). These conditions are encapsulated in (iii).

The final condition, regarding $\mathbf{U}_C$ in (3.107), specifies that an unobservable PWCS can be the result of the 6×3 matrix having a rank less than three. That is, no three of the six rows may be *linearly independent* if the system is unobservable. This gives 20 possible combinations of rows to check for the geometry in which the submatrices are not full rank. However, it will be shown that the condition on $\mathbf{U}_C$ implies the $\mathbf{U}_B$ condition is also met, and can therefore be disregarded. Assume that the condition (3.106) is *not* statisfied.

For this third condition, in (3.107), to be met, (2.68) dictates that the camera center is on $\mathscr{F}$ at each time step. Additionally, (3.111) and (3.112) will be zero when $y_2^{C/O} = 0$, hence the second feature point is on the same projection ray as the first feature point. These four constraints are upheld only when the second feature point is on the camera center or when all the points are collinear and on the same projection ray. These cases are impossible if (3.106) is not satisfied. Therefore, the unobservable configurations for the PWCS are completely described by the first two conditions, on $\mathbf{U}_A$ and $\mathbf{U}_B$. □

A physical interpretation for the three necessary and sufficient conditions for an unobservable PWCS is important for fully understanding the configurations that should be avoided. The first condition, (i), dictates that the system is unobservable if any of the point features are infinitely far away from the camera in the Y^C-axis at either time step. This is similar to the condition from the case when the target object model is known, but incorporates both time steps.

The second condition, (ii), indicates that the image line projection of the first feature point cannot be the same at both time steps. This is caused by the feature point moving along the original projection ray and the resulting lack of parallax which does not allow the point to be located along this ray.

The final condition in (iii) means that the PWCS is unobservable when the camera frame coordinates of the second or third feature point at time step k, rotated by the change in relative orientation, are parallel with the camera frame coordinates at time step $k+1$.

These conditions specify the situations which are to be avoided to keep the relative pose PWCS observable over the entire estimation. With this information, the combined pose and target model estimation in planar environments can be made more robust and reliable.

3.4.5 3D Observability

In the final observability analysis, the problem is expanded again to investigate the full six degree of freedom relative pose estimation with an unknown target object model measured by a monocular camera. The added degrees of freedom increase the complexity of such a study dramatically such that using the brute-force approach previously employed, seems intractable. Instead, this section will only comment on other work related to such a study, and the implications from the previous lower-dimensional cases.

A thorough work on the observability of estimating systems of this type is found in [47]. In this work, Soatto investigates the observability of the full 3D camera motion estimation when the structure of a rigid scene is initially unknown. He also points out that using state observers to recover the camera pose and scene structure, represented by a set of salient points, amounts to computing differentiations of the system output until the relationship between the output and initial state can be found. Further, when the system state vector contains only the six pose parameters, both the input (i. e. position noise) and initial state appear at the same level of differentiation. In this case, either the camera motion *or* the scene depth can be recovered, but not both.

In order to remedy this issue, the relative velocities of the pose parameters can be added to the state vector, referred to as *dynamic extension* and is equivalent to using the constant velocity model. In a similar approach to the planar observability studied previously, Soatto shows that such a system will never be linearly observable over one time step. Subsequently, Soatto then shows that the nonlinear system is not *locally observable* because the initial state of the system is *indistiguishable* with scaled versions of that initial state. This result dictates that the global scale will never be recoverable for systems using a single camera.

Soatto then looks at a different parameterization of the camera motion called the *essential model* which imposes constraints on the motion of the image points based on the rigid motion assumptions [47]. An essential model is said to be *in general position* when there are either more than 8 visible points, or a so-called *extended matix*, leading to a constraints matrix with rank 8. The major result of that work is the following theorem [47]:

Theorem 3. *An essential model is in general position* $\iff$ *there does not exist a quadric surface in* $\mathbb{R}^3$ *which contains all of the feature points and the path of the camera's center of projection.*

Furthermore, when the model is in general position, the work claims that it is possible to reconstruct the camera motion if an arbitrary scale factor is imposed. *Quadric surfaces* in $\mathbb{R}^3$ include ellipsoids, paraboloids, and hyperboloids.

Importantly, the conditions under which the *extended matrix* becomes full rank are not investigated in Soatto's work. Rather, it is shown that *if* it is full rank, then the results follow. The observability analyses in this thesis have investigated the conditions under which such a matrix could be full rank.

While Soatto identifies a large number of the unobservable configurations of the relative pose estimation system, there is more detailed information which can be inferred by looking at the previous results for the cases in this work, as a supplement.

The full linear observability matrix for the six degree of freedom case is not included in this report, but it is apparent from the previous results that the observability results from the 3D known object model observability analysis will directly apply here. Specifically, when only three feature points are measured over multiple time steps, there will be several different configurations which will result in the same feature point image plane measurements. These would represent local minima for an observer trying to find the optimal state estimates. Additionally, when the camera's center of projection lies on the cylinder with circular cross-section defined by the three feature points and projecting normal to the plane containing those points, the system will be in a singular configuration [41].

Further, from the planar case with unknown object model, motion in the direction of any one of the feature points makes the system unobservable over two time steps. Additionally, a pure rotation of the camera leads to an unobservable system.

The results in this section are preliminary and suggest an important direction for future research. Fully characterizing the relative motion and target object configurations which lead to an unobservable system, is important for understanding when such an observer can be deployed and the types of configurations which should actively be avoided to generate reliable estimates of both the relative pose and target object model.

Chapter 4

Recovering Scale

The previous chapter showed that the combined relative pose and target model estimation using a single monocular camera is *always* an unobservable system in which the global scale is ambiguous. The geometry of the target object, as well as an up-to-scale estimate of the relative pose of the target object with respect to the camera can be recovered *in almost all cases*, but the bearing-only nature of the camera sensor means that scale remains an unconstrained degree of freedom in the recovered solution.

In this chapter, the effects of adding additional sensors and the corresponding measurements to the system is considered. Specifically, those sensors which can provide measurements of the scale metric, are of interest. The ultimate goal is to come up with a combination of supplemental sensors to add to the original single monocular camera system detailed in the previous chapter, such that the resulting estimation is able to recover the scale information of both the target object model and the relative position and orientation of that target object with respect to the camera and robot end-effector.

4.1 Scale Information from Monocular Cameras

One can parameterize the relative position of the object, as well as the object feature point positions, in terms of a *normalized pose* $\check{\mathbf{x}}$, multiplied by a *global scale factor*, s [9]. Now the relative pose portion of the state vector for the combined system can be expressed,

$$\mathbf{x} = [\ X\ Y\ Z\ a\ b\ c\ \dot{X}\ \dot{Y}\ \dot{Z}\ \omega_x\ \omega_y\ \omega_z\ ]^T = [\ s\check{X}\ s\check{Y}\ s\check{Z}\ a\ b\ c\ s\dot{\check{X}}\ s\dot{\check{Y}}\ s\dot{\check{Z}}\ \omega_x\ \omega_y\ \omega_z\ ]^T. \tag{4.1}$$

Similarly, for the augmented state vector with the target model parameters appended to the state vector,

$$\tilde{\mathbf{x}} = [\ s\check{\mathbf{x}}^T \ ...\ s\check{p}_j^M \ s\check{q}_j^M \ s\check{r}_j^M \ \gamma_j \ \chi_j \ \frac{\check{\tau}_j}{s} \ ...\ ]^T, \tag{4.2}$$

and the rotation parameters are not affected by the global scale parameter.

Consider the monocular camera measurement model for the j^{th} feature point image plane projection,

$$x_j^I = -F_x \frac{x_j^C}{z_j^C} \tag{4.3}$$

$$y_j^I = -F_y \frac{y_j^C}{z_j^C} \tag{4.4}$$

where

$$\begin{bmatrix} x_j^C \\ y_j^C \\ z_j^C \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_M^C & s\check{\mathbf{t}}_M^C \\ \mathbf{0} & 1 \end{bmatrix} \left(\begin{bmatrix} s\check{p}_j \\ s\check{q}_j \\ s\check{r}_j \\ 1 \end{bmatrix} + \frac{s}{\check{\tau}_j} \begin{bmatrix} \cos\gamma_j \sin\chi_j \\ \sin\gamma_j \sin\chi_j \\ \cos\chi_j \\ 0 \end{bmatrix} \right). \tag{4.5}$$

and in this case,

$$\check{\mathbf{t}}_M^C = \begin{bmatrix} \check{X} \\ \check{Y} \\ \check{Z} \end{bmatrix}. \tag{4.6}$$

When these values are substituted into the measurement equations, the scale factor, s, multiplies both the numerator and denominator values, and as a result,

$$x_j^I = -F_x \frac{\check{x}_j^C}{\check{z}_j^C} \tag{4.7}$$

$$y_j^I = -F_y \frac{\check{y}_j^C}{\check{z}_j^C} \tag{4.8}$$

where

$$\begin{bmatrix} \check{x}_j^C \\ \check{y}_j^C \\ \check{z}_j^C \end{bmatrix} = \frac{1}{s} \begin{bmatrix} x_j^C \\ y_j^C \\ z_j^C \end{bmatrix}. \tag{4.9}$$

These measurements do not contain information about the global scale of the environment. Instead, the *ratio* of the camera frame coordinates determines the projection measurements. This concept is clear when considering the feature point projections in terms of two angles, azimuth and altitude, with respect to the camera coordinate frame. Only the *bearing* to the feature is measureable, not the distance along that bearing. Hence the name *bearing-only sensor.* This is why the inverse-depth parameterization is useful – the large uncertainty in the range to the feature point can be more effectively encoded and propogated through each filter recursion as a Gaussian distribution.

Suppose that the scale parameter, s, was appended to the state vector, similar to the feature point parameters and therefore becomes part of the estimation. Unlike the feature point parameters, the scale parameter does not appear in any of the measurement equations using the single monocular camera model. This lack of information about the scale leads to the s column entries of the measurement Jacobian being zero, and the measurement update step of the EKF is not able to map the innovation to a change in the scale parameter,

$$\frac{\partial x_j^I}{\partial s} = 0 \tag{4.10}$$

$$\frac{\partial y_j^I}{\partial s} = 0. \tag{4.11}$$

Therefore, there is no information about the scale parameter contained in the measurements and its value will not converge to the correct value. In fact, including this parameter in the estimation, leads to a measurement Jacobian with a column of zeros. This means that the augmented system is always unobservable no matter how many time steps or feature point measurements are considered. However, this is not quite a fair test since it is adding another state to the system, making it non-minimal and possibly artificially unobservable.

To recover this scale then, a measurement, or set of measurements, must be made for which the measurement model implicitly includes this scale parameter. Alternatively, it is also possible to constrain the global scale without extra measurements if prior information about the system is assumed. This is what happens in the known object model case studied in Chapter 2. The global scale is embedded in the target object model which is assumed known with perfect certainty. As a result, in almost all cases, there is only one unique set of relative pose parameters for which the feature measurements make sense.

Consequently, most monocular vision systems which attempt to simultaneously estimate the model along with the relative pose, specify the global scale by initially adding feature points with known location to constrain the estimation [21] [12] [43]. Typically, this consists of a set of four points arranged in a shape of known dimension. These features lock in both the global scale and the initial object frame configuration. As new features are added to the estimation, their location estimates must therefore, converge to the correct scale for the solution to be consistent.

By having to pre-specify the locations of a subset of the target object feature points upon initialization, the resulting systems do not accomodate for *truly unknown* target objects. In the SLAM context, where the object (the world, in this case) may consist of hundreds of feature points, the initial 'known' set may be lost as the world is traversed. At this point, since there is no scale information in the measurements of the estimated unknown feature point locations, the scale parameter can and will drift as error accumulates.

It is well-known in SLAM that *loop closure* is an important mechanism for building consistent maps [50]. When previously tracked feature points are revisited, after looping around an interior corridor, for example, the observations are propogated through the feature point correlations and act to correct drift and accumulated errors in the map estimates. With the monocular camera SLAM system, the problem is exaggerated since the scale parameter could wander away from the true value until the initially known points are revisited, instead of just *any* previous feature points.

Obviously, the requirement that a set of features be at known locations, means that information must be available at the start of the estimation. While this may be the case for some situations, there are many instances where a robot manipulator could be working within a *completely unknown environment* where the target feature locations are all initially completely unknown. Being able to build an accurate target model and then estimate its position and orientation relative to the camera would eliminate the requirement for this a priori information. Therefore, the ability to accurately recover the scale of the environment at each time step would alleviate these problems.

4.2 Current Approaches

While it is possible to impose constraints on the estimation by initializing the system with known target object feature points, truly unknown target object model estimation requires supplemental measurements over and above those from a monocular camera. In mobile robotics literature, this information is usually vehicle odometry with respect to a static environment, but some researchers have investigating adding supplemental sensors to aid in recovering the global scale. These can take the form of range measurement devices or additional monocular cameras.

4.2.1 Camera and Range Sensors

Jung *et al.* in [29] implement a range finding device for use in a SLAM context by projecting a *structured light* on the environment and measuring the resulting distortions with a monocular camera. In this case, a narrow horizontal line of infrared laser light is shone at an angle and the resulting camera image is used to detect corners in an interior environment. With a known configuration of the light beam with respect to the camera, the depths of each pixel comprising the laser line in the image can be calculated. Despite the pairing of the two sensors, the estimation, implemented using the *FastSLAM* filter [42], does not use any visual information besides that for extracting the range to the laser line. The system is limited in both accuracy and effective range of the sensors.

One of the most popular sensors used for the mobile robot SLAM problem is the 2D laser range finder (LRF). It has seen widespread use as the primary sensor on a robotic platform since it is able to measure *both* the range and bearing to landmarks by sweeping a laser beam across the environment and measuring the time-of-flight at each bearing. It is seen as a competitor sensor to the use of monocular cameras. Only a few systems have integrated the two sensors and fused the measurements to try and improve the estimation of the robot pose and map landmark locations.

An early example is the work of Castellanos *et al.* in which the explicit fusion of vision and laser measurements is used to improve the resulting estimation compared with the exclusive use of only one of the two sensors [6]. The LRF is used to detect vertical corners and semiplanes in a structured environment. These features are matched with vertical edges in the image plane of the monocular camera. This allows the two sensors to collect information about a set of features common to both devices. With both the range and bearing available to each map feature, the global scale of the solution is easily recovered.

A similar approach is taken by Yan *et al.* [54] where a monocular camera and laser range finder are used to both measure vertical edges and corners in the environment. The measurements are fused in the EKF framework by forming appropriate measurement models for each device which relate the measured edge locations to the system states. Estimations are further enhanced by the use of a vehicle dynamic model instead of the constant velocity model, imposing a static map through which the robot moves.

More recently, Amarasinghe *et al.* [1] integrate laser and camera measurements explicitly by using the laser scans only to determine the distance to vertical edge features measured in the monocular camera. As demonstrated in the two previous systems, this measurement fusion would be better accomplished by the EKF framework and it is likely that information is being lost by using these pre-combined measurements instead of the two seperate metrics. However, the availability of this range data means that the global scale is again readily recovered.

These systems suggest that the availability of this range data is sufficient to recover and maintain the proper global scale. Whether the quantity of data available from these sophisticated LRFs is necessary, is of interest in this study. A single range measurement at each system time step is used in the following sections.

4.2.2 Multiple Cameras

Other researchers have proposed to make the relative pose and target object model estimation system observable by adding additional monocular cameras to make feature measurements. It is important to realize that these are *not* stereo camera sensors since the image planes from the individual cameras are not fused to form a disparity map. Instead, they are treated as seperate sensors each measuring only the bearing to the set of feature points, and these measurements are *fused* by the estimator to recover the solution. The mechanics of such a sensor fusion framework are detailed in later sections.

An example of a mobile robot system with many monocular cameras arranged in a ring configuration is proposed by Kaess and Dellaert [30]. A planar SLAM process makes estimations of the pose and landmark map based on observations of features in all of the monocular camera image planes. Significantly, none of the monocular cameras have overlapping fields of view and the feature measurements from each are fused by the estimator. While odometry data is used to assist the estimation accuracy, this extra information is not required if the extrinsic calibration parameters between each of the monocular cameras is available. In any case, the effective field of view of the resulting *super sensor* is nearly 360 degrees and the combined system estimates are accurate in the global scale.

Sola *et al.* [48] also propose a multiple camera system in which monocular cameras are mounted on one or across mobile robots. In the single vehicle model, multiple cameras are used to estimate the robot pose, the world map, and the extrinsic calibration parameters of the monocular cameras. The authors are aware that by estimating the extrinsic calibration of the cameras, the odometry of the mobile robot is the key to recovering the global scale of the environment. Without this extra information, the estimation would be similar to multiple monocular cameras independently estimating the combined problem, for which the scale is ambiguous. The alternate way to impose a constraint on the global scale is to explicitly specify the extrinsic calibration parameters accurately.

A second setup, involving multiple robots each with a single monocular camera, works to cooperatively build a unified map [48]. The effective baseline, which now changes as the robots move, is estimated based on the monocular camera measurements and the odometric data and the resulting solution scale is adversely affected by bias in the odometry readings.

Kim *et al.* also propose a multiple robot with monocular camera system [33]. Two seperate monocular cameras are mounted on two different robots. As with most of the other previous systems, the odometric data is used with the robot dynamics, as well as having the estimation initialized with a set of *known* landmark locations. While the resulting map is accurate to the proper global scale, this fact is not solely due to using multiple cameras on multiple robots, it is the result of using known feature points and vehicle odometry. While the extra camera measurements would improve the quality of the feature point estimates, the goal of this chapter is to understand *why* a particular set of sensors will make the resulting estimation accurate to the environment scale, and what measurements are necessary to reliably and efficiently recover this information.

4.3 Supplemental Measurements

As previously mentioned, Civera *et al.* [9] show how the global scale of the world frame can be treated as a separate parameter in the monocular vision-based SLAM context, and therefore must be estimated independently using supplemental measurements, in addition to those from the camera. It is, therefore, necessary to include additional measurements which contain information about the global scale of the object and camera motion to make the complete system fully observable. However, the selection of specific sensors and measurements which are best-suited to the task is not discussed.

There are several options for supplementing the measurement space to potentially make the system observable: additional cameras with known baselines between each, ultrasonic/sodar/laser point range finders, Light Detection and Ranging (LIDAR), inertial measurement units (accelerometers, gyroscopes, magnetometers), Global Positioning System (GPS) receivers, and odometry measurements are

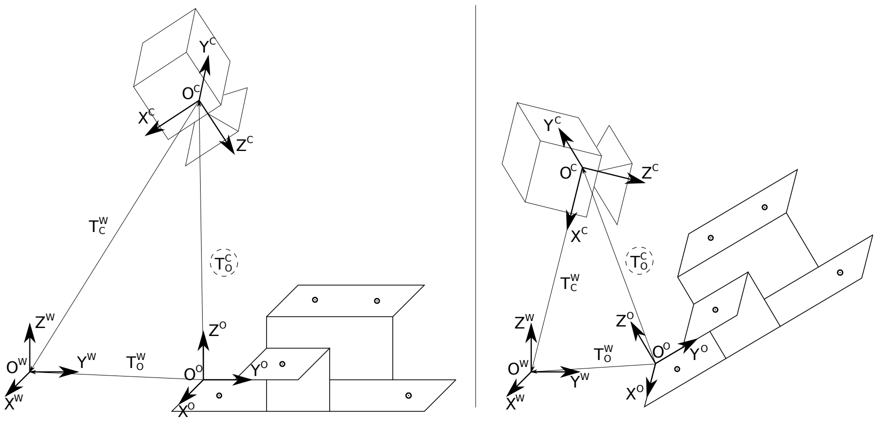


Figure 4.1: Both the camera and object are free to move in the world coordinate frame, but it is the relative pose of the two frames which must be estimated.

just a few examples. These alternatives have different characteristics which affect their suitability to the task of localization and model-building on a specific robotic platform. Additionally, there are significant differences between these types of sensors in terms of cost, weight, and performance. Moreover, the sensors provide measurements with respect to different coordinate frames.

In the pose estimation process of position-based visual servoing, it is the relative pose of the target object frame with respect to the camera frame that is used for robot feedback control purposes [53]. The target object being manipulated is assumed to be a rigid-body, but is *not assumed to be static.* That is, the point feature locations are assumed fixed with respect to the object frame, but the object frame itself is able to move freely with respect to the *world* frame. An example of both the camera and target object moving within the world coordinate frame is show in Fig. 4.1. The system is not interested in measuring the position and orientation of either frame in the world frame, only their relative pose with respect to each other. This is the most important difference between position-based visual servoing pose estimation and mobile robot localization or SLAM.

With both the object and camera free to translate and rotate in the world frame, measuring the relative motion is not possible with inertial sensors mounted on the camera since it does not provide a measure of the object's motion. The same is true for measurements from a GPS receiver, which provides positions in the Earth-Centred Earth-Fixed (ECEF) coordinate system. Without the second set of inertial or GPS measurements for the target object as well, the relative motion of the camera and object cannot be determined. As a result, using measurements of the camera movement based in the world frame would not be sufficient for the relative motion estimation since both the camera and the object may be moving. Instead, only *direct relative* measurements can be used – such as those from end-effector-mounted bearing and/or range sensors at known displacements.

The goal of this section is to identify a small set of supplemental measurements necessary to *reliably* recover the relative motion of a rigid target object with respect to the end-effector coordinate frame. These estimates, which must include the *true* global scale, provide pose measurements utilized for controlling a robotic system. Three major factors considered for any candidate sensor in this work are speed, cost, and weight. Ideally, the measuring equipment should be fast, inexpensive and light-weight.

Accordingly, this work will look at three supplemental sensors. First, a second monocular camera is added to the robot end-effector at a known position and orientation with respect to the first camera as shown in Fig. 4.2. By observing features in the two cameras, the distance to the features can implicitly be triangulated and recovered to scale. While many researchers have considered *stereo cameras*, this algorithm *does not* try to associate features to create a *disparity map* [32] before applying the measurements to the EKF. Instead, the two cameras proceed as if they were running *two separate monocular camera-based estimations* recovering the relative pose and target model up-to-scale, and then can use the information to predict when the features will be visible in each camera and resolve the scale based on measurements from both cameras. In this way, the cameras do not necessarily have to have their fields of view (FOV) overlap, and adding n cameras to the system can increase the combined FOV and framerate by n times. For this work, the dual camera system will only serve as a 'best case' scenario since it definitely represents a non-minimal set of additional measurements.

Second, a single-point reflection range-finder device operating with one of ultrasonic, sonar, or laser signals is added to the robot. This sensor will be fixed at a location on the end-effector and remain in a constant orientation relative to the camera frame as shown in Fig. 4.3. It is important that the range sensor be oriented in such a way that it is likely to strike the target object and enable measurement of the relative distance.

Third, in a slight variation of the previous device, a laser pointer is fixed to the robot end-effector at a known position and orientation, as shown in Fig. 4.4. Then, using the monocular camera, the image plane projection of the the laser dot striking the target object can be measured and the two bearings are used to triangulate the

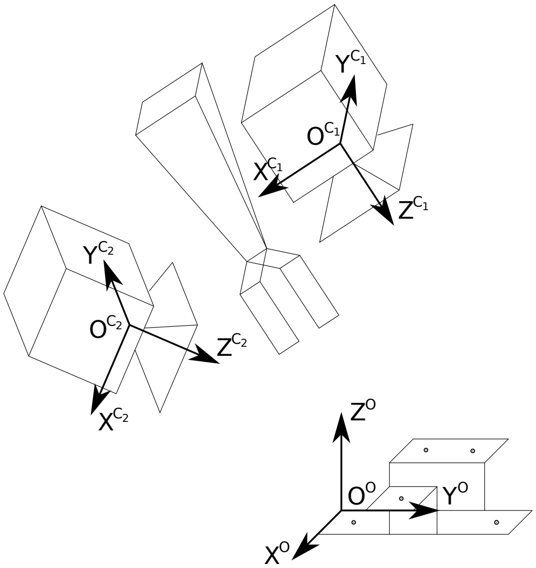


Figure 4.2: A second monocular camera is fixed to the robot end-effector and makes image plane measurements of the target object feature point locations (dots).

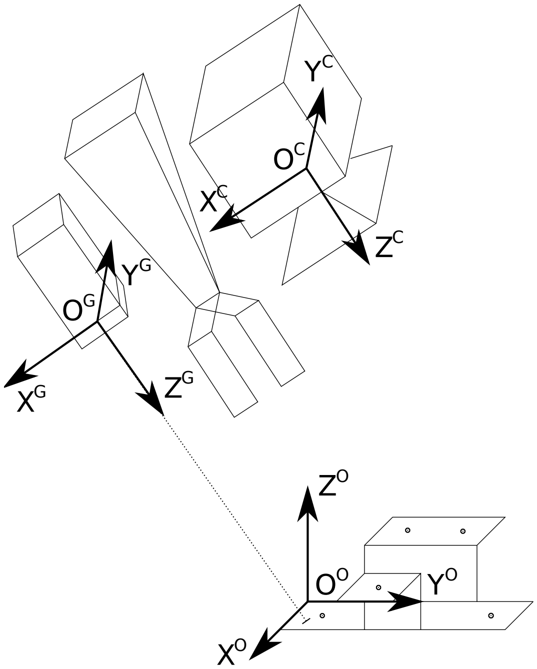


Figure 4.3: A point range sensor is mounted on the robot end-effector and the beam strikes the surface of the target object. It is able to measure the distance to this intersection at each time step.

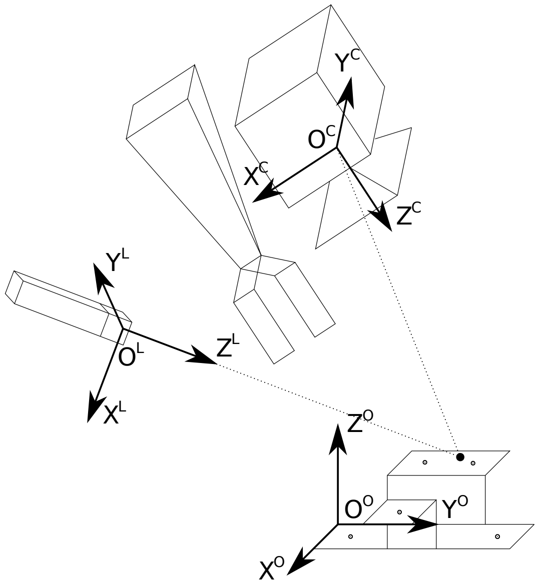


Figure 4.4: A laser pointing device is fixed to the robot end-effector at a known position and orientation such that image of where the laser beam dot strikes the target object surface provides a measure of the distance.

distance to the target object. This sensor has the advantage of being inexpensive to implement, and the measurements are available at the frame rate of the camera.

4.3.1 Additional Monocular Cameras

If the baselines between multiple cameras are accurately known from calibration and the feature points within the images can be correctly associated across camera frames and time steps in the sequence, a multiple camera setup can give an estimate of the scene depth and scale at the first observation.

In this work, an interpretation of the sensors will be used similar to [30]. Instead of actively trying to associate a feature in the two cameras in every single frame to try and determine the scene depth at *every* pixel, the cameras will operate as if each were performing a pose estimation with only one camera but the two sets of measurements will be fused by the EKF framework. To simplify implementation and facilitate comparison, it is assumed that the correspondance problem between object feature points and image plane measurements, is solved at each time step.

Consider the combined relative pose and target model estimation with a single monocular camera. The system is able to produce consistent estimates to an up-to-scale solution. By introducing the scale parameter, s, explicitly into the state vector, the uncertainty in the global scale can be quantified. When the object is observed in a single camera, the relative orientation, as well as the shape of the target object can be accurately estimated. Therefore, target object feature points are known to lie on a set of rays projecting out from the intial observation point. With the uncertainty constrained to lines in three dimensional space, a reasonable estimate of *when* the object feature should be visible in the second camera, can be asserted. Furthermore, the search region for a given feature point will be a line in the second camera image [35].

If the relative position and orientation of the second camera frame with respect to the first camera is known accurately, the scale parameter can be estimated once features have been observed in both the first and second cameras. In fact, only a single feature is needed to be observed across the two cameras to resolve the scale. As a result, only the most likely correspondences need to be considered. This can generalized for multiple cameras with known relative position and orientations, even when the fields of view do not overlap. However, in the latter case, different parts of the target object need to be visible in two different cameras at one time. This requirement is easily met for tracking feature points associated with large objects such as building interiors, in which the robot is surrounded by the object.

Suppose there are n cameras mounted at the robot end-effector. Each camera has an associated coordinate frame (C_i) in which the origin corresponds the center of projection of the camera and the Z^{C_i}-axis is aligned with the principal ray. The camera's image plane is spanned by the orthogonal X^I and Y^I axes corresponding to the horizontal and vertical image directions respectively, and run parallel with the X^{C_i} and Y^{C_i} coordinate frame axes.

To simplify the equations, the first camera will be assumed to be the primary coordinate frame in which the estimation proceeds. Accordingly, the relative pose estimates are made with respect to this C_1 frame. The position and orientation of all the camera frames are assumed known with respect to the C_1 frame and the corresponding transformations can be written as $\mathbf{T}_{C_i}^{C_1}$.

For the j^{th} feature point observed in the i^{th} camera image plane, the measurement model to relate the output to the system states is formed similar to that for the single camera system. First, the feature point location is transformed into the

C_1 frame as before,

$$\begin{bmatrix} x_j^{C_1} \\ y_j^{C_1} \\ z_j^{C_1} \\ 1 \end{bmatrix} = \mathbf{T}_M^{C_1} \left(\begin{bmatrix} p_j^M \\ q_j^M \\ r_j^M \\ 1 \end{bmatrix} + \frac{1}{\tau_j} \begin{bmatrix} \cos\gamma_j \sin\chi_j \\ \sin\gamma_j \sin\chi_j \\ \cos\chi_j \\ 0 \end{bmatrix} \right). \tag{4.12}$$

such that

$$\mathbf{T}_M^{C_1} = \begin{bmatrix} \mathbf{R}_M^{C_1}(a,b,c) & \mathbf{t}_M^{C_1} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{4.13}$$

where

$$\mathbf{t}_M^{C_1} = [\ X\ Y\ Z\ ]^T. \tag{4.14}$$

Then, the feature point is transformed into the C_i frame using the known transformation,

$$\begin{bmatrix} x_j^{C_i} \\ y_j^{C_i} \\ z_j^{C_i} \\ 1 \end{bmatrix} = T_{C_1}^{C_i} \begin{bmatrix} x_j^{C_1} \\ y_j^{C_1} \\ z_j^{C_1} \\ 1 \end{bmatrix} \tag{4.15}$$

and finally projected onto the i^{th} camera's image plane,

$$x_j^{I_i} = -F_x^{C_i} \frac{x_j^{C_i}}{z_j^{C_i}} \tag{4.16}$$

$$y_j^{I_i} = -F_y^{C_i} \frac{y_j^{C_i}}{z_j^{C_i}}. \tag{4.17}$$

The significance of these equations is the transformation, $\mathbf{T}_{C_1}^{C_i}$, which contains scale information. In fact, the image plane projection does not cause a normalization of this scale information and therefore the estimation can determine the global scale of the relative pose and target object model parameters.

If the system were parameterized with the explicit global scale s, as before, it is found that,

$$\frac{\partial x_j^{I_i}}{\partial s} \neq 0,\ i \neq 1 \tag{4.18}$$

$$\frac{\partial y_j^{I_i}}{\partial s} \neq 0,\ i \neq 1 \tag{4.19}$$

and therefore the solution scale affects the measurement and will be driven to a value which tries to minimize the sum of squared error.

An advantage of using multiple cameras instead of simple range finding devices for this estimation is an easier association of the outputs and states since the individual feature points can be measured instead of some combination of all of them. As a result, the measurement model relies on weaker assumptions and is less affected by the unknown surface geometry of the target object.

4.3.2 Point Range Finder

For the purposes of this study, a *point range finder* is a device capable of measuring the distance along a ray eminating from the origin to the first point of intersection with the surface of the target object. While there are several varieties of physical devices using different media (i. e. ultrasonic, sonar, laser, etc.), each with different beam characteristics (sonic devices tend to have cones of energy, where lasers can have much narrower focussed beams), it will be assumed that the beam is infinitely narrow such that the range to a single point straight out from the device can be measured up to some zero-mean Gaussian measurement noise.

Mathematically, the range finder device is represented by a *Range Finder (G) coordinate frame*, in which the measurement beam eminates from the origin and out along the Z^G-axis. By assuming that the relative position and orientation of the G frame with repect to the C frame is known, the range measurements can then be related to the image-plane measurements from the monocular camera. If the surface geometry of the target is known, the range measurement can be calculated as the distance to the first intersection of the range beam with a surface on the target object using ray-tracing techniques [28].

Range Finder Measurement Model

One of the main difficulties in adding a range sensor to the monocular camera-only system is fusing the range measurement data with the feature point measurements from the camera images if the target object surface geometry is unknown. The image plane measurements are in reference to point and line features which will *rarely* be exactly aligned with the range sensor. The range measurement device sends out a beam or cone of energy and measures the amount of time it takes for the the reflection off of a surface to return to the sensor. These reflecting surfaces do not necessarily have anything in common with the visual object features in the images, besides being fixed with respect to the same local coordinate frame, as is the case for a rigid-body. As a result, one cannot write a model for the range sensor in terms of the object feature points without making some assumptions about how a surface is defined with respect to the object feature points.

The problem with relying on such assumptions is that the generated model is a sparse set of point features and therefore, little information about the solidity or surface geometry of the target object can be immediately inferred from the locations of the features. It would be possible to create an Occupancy Grid Map [50] over time, but the sparsity of the features could make this a time-consuming process. However, an assumption about the surface that the range sensor is striking can be made, in order to avoid or at least delay the need for this more complicated model representation.

For example, if it is assumed that the beam of the range sensor will strike a smooth surface that is fixed in the object coordinate frame, where the surface parameters are relatively consistent between time steps, a mathematical model for the ranger's measurements can be formulated relating the system states and parameters with the range measurements. While it could be further assumed that the surface was planar, there is no reason why a more complicated surface could not be used to represent more complex surfaces. However, a flat reflection surface could be a reasonable assumption, especially if the range device is pointed at the floor, ceiling, or a wall section of a building interior. The parameters of such a surface could be appended to the state vector exactly as done for feature point coordinates.

The range sensor could be turned on and off once the global scale is determined since the camera image plane measurements will result in properly scaled models once the scale is recovered. Therefore, the range sensor measurements could be turned on in areas with known flat surfaces and then disabled once the scale is recovered. Subsequently, the system could continue to estimate the relative motion with this accurate scale information. At this point, the scale parameter may drift when only the camera is used, but the range sensor can then be enabled to capture this metric again.

This particular range measurement model is suitable for target objects which are rather large where the assumptions for consistently measuring the range to a smooth surface would remain valid. A good example of where this would be the case would be an aircraft carrier. The ship, since it is afloat, would not be a stationary object in the world frame and trying to land a helicopter on the deck, for example, requires that a *relative trajectory* down to the deck be followed precisely. With a range device reflecting off the deck, the scale of the relative pose and model of the deck could be recovered accurately without a priori information.

Conversely, for small workpiece target objects in an industrial robotics environment, this measurement model may not be ideal for recovering the scale of the environment. For parts with abrupt changes in the surface geometry, like steps, the estimation may fail or cause an inaccurate scale to be recovered. In this environment, it may be more likely that the depth variations in the object feature points are small compared to the distance from the camera to the target object. If this is the case, a different measurement model can be formulated by assuming that the range measurement is the average depth of the target object feature points. This is the measurement model that will be used for the range sensors in this work, and its derivation is the subject of the next section.

Mean Depth Measurement Model

The measurement model which relates the range measurements to the object feature points visible in the camera image plane, assumes that the range sensor measures the distance to the average feature point on the target object. Shown in Fig. 4.5, the calculation of this mean depth value, over N feature points, involves first calculating the average feature location in the modelling coordinate frame.

$$\begin{bmatrix} x_{avg}^M \\ y_{avg}^M \\ z_{avg}^M \end{bmatrix} = \frac{1}{N}\sum_{j=1}^{N}\left(\begin{bmatrix} p_j^M \\ q_j^M \\ r_j^M \end{bmatrix} + \frac{1}{\tau_j}\begin{bmatrix} \cos\gamma_j \sin\chi_j \\ \sin\gamma_j \sin\chi_j \\ \cos\chi_j \end{bmatrix}\right). \tag{4.20}$$

Next, this point is transformed into the G frame through the camera frame according to the transformation assumed known from calibration,

$$\begin{bmatrix} x_{avg}^G \\ y_{avg}^G \\ z_{avg}^G \\ 1 \end{bmatrix} = \mathbf{T}_C^G \mathbf{T}_M^C \begin{bmatrix} x_{avg}^M \\ y_{avg}^M \\ z_{avg}^M \\ 1 \end{bmatrix}. \tag{4.21}$$

Now that the mean feature position in the G frame has been found, it is only the Z^G coodinate that is of interest, representing the depth to the object,

$$g^G(\tilde{\mathbf{x}}) = z_{avg}^G. \tag{4.22}$$

The range beam travels out the Z^G-axis and is therefore the predicted distance measured by the range sensor.

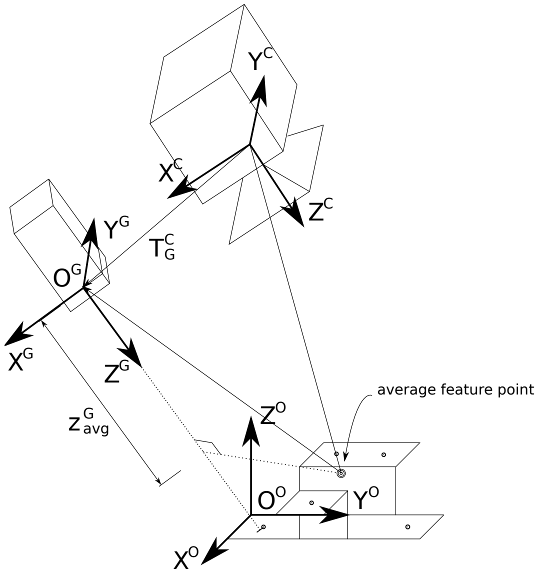


Figure 4.5: The range measurement is assumed to correspond with the average feature point depth in the G coordinate frame.

4.3.3 Laser Pointer

A variant of the simple point range-finder device, the monocular camera can be augmented with a laser pointer to provide range measurements of the target object. By fixing the laser pointer at some known position and orientation with respect to the monocular camera, the reflection of the laser light on the target object can be captured in each camera image. In this way, two bearings to the laser dot are available and the depth can then be triangulated to get a measurement of the distance to the target object as shown in Fig. 4.6.

One way to think about this process is to consider the laser pointer as a single-pixel point camera. When the laser light is on, this 'feature' is always seen at that single pixel. The monocular camera then becomes the second camera in a pseudo-stereo camera rig. In the same way that a stereo camera is able to extract depth from the two camera images, so does this device.

One difficulty arises when trying to extract the laser point from the camera image. This can be simplified in several ways. First, since the laser beam is a line, the laser dot will only show up in a certain line across the camera image. This concept is from *epipolar geometry* [35], and significantly limits the region of the image which must be searched for the laser dot. Secondly, the laser dot, usually red, can be found by colour filtering and treated as a simple blob. Thirdly, the laser light could be pulse-modulated to make extracting its location even more robust.

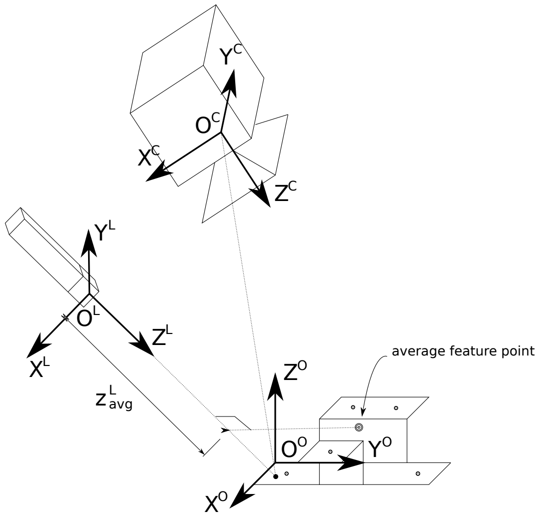


Figure 4.6: The laser pointer beam is assumed to intersect with the target object surface at the average feature point depth in the L frame.

Finally, if the laser cannot be reliably extracted from an image, the EKF framework would allow the measurement to be neglected and the estimation would proceed with the scale information only when it was available.

Similar to the point range finder, the laser pointer is modelled such that the laser beam projects from the origin of the *Laser Pointer (L) coordinate frame*, out along the Z^L-axis. The measurements are the two camera image-plane coordinates of the laser dot reflection off the first intersection with the target object's surface. If the target object's surface geometry is known, ray-tracing techniques can again be used to find the point of intersection in the camera frame, $[\ x_l^C\ y_l^C\ z_l^C\ ]^T$, and then it is projected onto the camera image plane [28].

Just like the previous range data, this measurement needs to be related to the feature point locations comprising the target object model. It is again assumed that the laser beam strikes the object at the mean feature point depth, determined from the average feature point location,

$$\begin{bmatrix} x_{avg}^L \\ y_{avg}^L \\ z_{avg}^L \\ 1 \end{bmatrix} = \mathbf{T}_C^L \mathbf{T}_M^C \begin{bmatrix} x_{avg}^M \\ y_{avg}^M \\ z_{avg}^M \\ 1 \end{bmatrix}. \tag{4.23}$$

Only the z_{avg}^L coordinate is of interest and is used to find the expected position in the camera frame,

$$\begin{bmatrix} x_l^C \\ y_l^C \\ z_l^C \\ 1 \end{bmatrix} = \mathbf{T}_L^C \begin{bmatrix} 0 \\ 0 \\ z_{avg}^L \\ 1 \end{bmatrix}. \tag{4.24}$$

Finally, the point is projected onto the camera image-plane to get the predicted output,

$$x_l^I = -F_x \frac{x_l^C}{z_l^C} \tag{4.25}$$

$$y_l^I = -F_y \frac{y_l^C}{z_l^C}, \tag{4.26}$$

such that the resulting predicted measurements become,

$$\mathbf{g}^L = \begin{bmatrix} x_l^I & y_l^I \end{bmatrix}^T. \tag{4.27}$$

4.4 Recursive Estimation

In this section, the full recursive estimation system is described for combining all of the aforementioned sensor measurements into a relative pose and target object model estimate, including global scale information. The system builds on the framework described at the end of the previous chapter and facilitates the fusion of the supplemental sensor data with the single monocular camera.

4.4.1 Relative Motion Dynamics

The relative motion between the camera and target object is once again approximated with a constant velocity model. Accordingly, the process model, as well as the system states are identical to the system in Section 3.3.7. The system states represent the relative position and orientation parameters and their time-derivatives of the modelling M frame with respect to the first monocular camera C_1 frame, as well as the feature point parameters for the N feature points on the target object, expressed in the M frame,

$$\tilde{\mathbf{x}} = [\ X\ Y\ Z\ a\ b\ c\ \dot{X}\ \dot{Y}\ \dot{Z}\ \omega_x\ \omega_y\ \omega_z \mid \ldots\ p_j^M\ q_j^M\ r_j^M\ \gamma_j\ \chi_j\ \tau_j\ \ldots\ ]^T, \tag{4.28}$$

where $j = 1\ \ldots\ N$.

The process model dictates that the relative pose parameters follow the constant velocity approximation, while the model parameters are fixed,

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{A}}_{k-1}\tilde{\mathbf{x}}_{k-1} + \tilde{\eta}_k, \tag{4.29}$$

where

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{F}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{6N\times 6N} \end{bmatrix}, \tag{4.30}$$

such that,

$$\mathbf{F}_{k-1} = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_{k-1}}, \tag{4.31}$$

is the process Jacobian for the nonlinear relative pose process model,

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \begin{bmatrix} X + \delta_t \dot{X} \\ Y + \delta_t \dot{Y} \\ Z + \delta_t \dot{Z} \end{bmatrix} \\ \mathbf{p} + \delta_t \dot{\mathbf{p}} \\ \begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} & \omega_x & \omega_y & \omega_z \end{bmatrix}^T \end{bmatrix}, \tag{4.32}$$

using MRP for relative orientation,

$$\mathbf{p} = [\ a\ b\ c\ ]^T, \tag{4.33}$$

given,

$$\dot{\mathbf{p}} = \frac{1}{2}\left(\frac{1}{2}\left(1 - \mathbf{p}^T\mathbf{p}\right)\mathbf{I}_{3\times 3} + [\mathbf{p}\times] + \mathbf{p}\mathbf{p}^T\right)\omega, \tag{4.34}$$

and

$$\omega = [\ \omega_x\ \omega_y\ \omega_z\ ]^T. \tag{4.35}$$

4.4.2 Measurement Models

While the process model for this new system is unchanged from those previous, the significant differences are found in the measurement model. Moreover, the different measurement models for each sensor all fit into the estimation framework to allow all of the data to be fused into one cohesive solution for the relative position and orientation of the target object.

Using these different models, it is possible to pick and choose which measurements are used at any one time step. The sensors, with the exception of the first monocular camera, are optional. The inclusion of any of the supplemental sensors *should* provide scaled estimates of the system solution, but adding more will provide redundancy and robustness.

With that in mind, all of the measurement models will be shown in this section to create a framework to which any number of supplemental monocular cameras, or range finding devices can be added.

Each measurement occupies one row in the output matrix, which relates the measurement with the system states. At each timestep k, the individual measurements are stacked to form the measurement vector,

$$\mathbf{g}_k(\tilde{\mathbf{x}}_k) = \begin{bmatrix} \mathbf{g}_k^C(\tilde{\mathbf{x}}_k) \\ \mathbf{g}_k^G(\tilde{\mathbf{x}}_k) \\ \mathbf{g}_k^L(\tilde{\mathbf{x}}_k) \end{bmatrix}, \tag{4.36}$$

where $\mathbf{g}_k^C(\tilde{\mathbf{x}}_k)$ are the measurements from all of the monocular cameras stacked into rows, $\mathbf{g}_k^G(\tilde{\mathbf{x}}_k)$ are the measurements from all of the point range devices, and $\mathbf{g}_k^L(\tilde{\mathbf{x}}_k)$ are measurements from laser pointer range sensors. The specific structures of these subvectors will be discussed in the following sections.

Just as before, this measurement vector needs to be linearized about the current state estimate for use in the EKF,

$$\mathbf{C}_k = \frac{\partial \mathbf{g}_k(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}|_{\tilde{\mathbf{x}}=\tilde{\mathbf{x}}_k}. \tag{4.37}$$

Each of the sensors have associated measurement noise parameters which need to be known accurately to allow the measurements and predictions to be weighted in an 'optimal' way determine the best estimate of the state variables. All of the measurement noise is assumed to be uncorrelated, zero-mean Gaussian noise, but the covariance is specified for each measurement by a diagonal covariance matrix, $\mathbf{N}_k$,

$$\mathbf{N}_k = \begin{bmatrix} \mathbf{N}_k^C & & \\ & \mathbf{N}_k^G & \\ & & \mathbf{N}_k^L \end{bmatrix}, \tag{4.38}$$

where $\mathbf{N}_k^C$ are the noise covariance statistics for the monocular camera measurements, $\mathbf{N}_k^G$ are for the point range devices, and $\mathbf{N}_k^L$ are the noise properties associated with the laser pointer measurements. The contents of these vectors for each of the sensor groups will be detailed individually the following sections.

Monocular Cameras

For a set of α monocular cameras, the measurements of the target object feature points in each image plane can be compiled into a $2\alpha N \times 1$ measurement vector as follows,

$$\mathbf{g}_k^C(\tilde{\mathbf{x}}_k) = \begin{bmatrix} \mathbf{g}_k^{C_1}(\tilde{\mathbf{x}}_k) \\ ... \\ \mathbf{g}_k^{C_\alpha}(\tilde{\mathbf{x}}_k) \end{bmatrix}, \tag{4.39}$$

where each subvector corresponds to the $2N$ feature measurements in each camera's image plane,

$$\mathbf{g}_k^{C_i}(\tilde{\mathbf{x}}_k) = \begin{bmatrix} \begin{bmatrix} -F_x^{C_i}\frac{x_1^{C_i}}{z_1^{C_i}} & -F_y^{C_i}\frac{y_1^{C_i}}{z_1^{C_i}} \end{bmatrix}^T \\ ... \\ \begin{bmatrix} -F_x^{C_i}\frac{x_N^{C_i}}{z_N^{C_i}} & -F_y^{C_i}\frac{y_N^{C_i}}{z_N^{C_i}} \end{bmatrix}^T \end{bmatrix}, \tag{4.40}$$

and $F_x^{C_i}$ and $F_y^{C_i}$ are the horizontal and vertical focal lengths for the i^{th} camera expressed in pixels.

The measurement model uses the feature point coordinates written in the C_i frame. The points are transformed from the feature coordinates in the first camera frame (C_1), in which the relative pose estimation takes place,

$$\begin{bmatrix} x_j^{C_i} \\ y_j^{C_i} \\ z_j^{C_i} \\ 1 \end{bmatrix} = \mathbf{T}_{C_1}^{C_i} \begin{bmatrix} x_j^{C_1} \\ y_j^{C_1} \\ z_j^{C_1} \\ 1 \end{bmatrix}. \tag{4.41}$$

It is assumed that the transformation between camera frames, $T_{C_1}^{C_i}$, is well known from calibration.

The coordinates of the feature point in the first camera frame (C_1) are calculated from the system states,

$$\begin{bmatrix} x_j^{C_1} \\ y_j^{C_1} \\ z_j^{C_1} \\ 1 \end{bmatrix} = \mathbf{T}_M^{C_1} \left(\begin{bmatrix} p_j^M \\ q_j^M \\ r_j^M \\ 1 \end{bmatrix} + \frac{1}{\tau_j} \begin{bmatrix} \cos\gamma_j \sin\chi_j \\ \sin\gamma_j \sin\chi_j \\ \cos\chi_j \\ 0 \end{bmatrix} \right) \tag{4.42}$$

where the transformation is given by the relative pose parameters,

$$\mathbf{T}_M^{C_1} = \begin{bmatrix} & X \\ \mathbf{R}_M^{C_1}(a,b,c) & Y \\ & Z \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4.43}$$

Obviously, in the case of measurements in the first camera, the transformation, $T^{C_1}_{C_1}$, is simply the 4×4 identity matrix. The resulting measurement models must be linearized about the current state estimates to form the measurement Jacobian. The expression for this matrix is determined for each additional camera in Appendix A.6.

Using this framework, the measurements from any number of monocular cameras can be used to estimate the relative position and orientation of the target object with respect to the robot end-effector. The difficulty arises when trying to associate the measurements of features in each camera's image plane with the correct feature parameters in the state vector. If improper correspondances are made, the estimation will be adversly affected and the tracking operation could fail.

To see how this problem could be alleviated, consider a two-camera system in which the target object is initially visible in only one camera. As the estimation progresses, the measurements from the first camera are sufficient to recover the shape of the target object feature point configuration, as well as the relative orientation of the object with respect to the camera frame. Given that the relative position of the second camera is known with respect to the first camera frame, the knowledge collected from the first camera measurements, indicates when some features of the object may *potentially* be visible in the second camera.

Since the scale parameter is the uncertain part of the system, the possible locations for the features within the second camera's image plane are restricted to a line. By searching only along these rays for the target object features, a single value of the scale parameter should provide a good fit for the set of feature measurements. The suitability of these feature correspondances in the second camera's image can be tested before the measurements are used in the estimation to make sure that the data passed to the estimation is valid. In this way, the search space for *when* and *where* to look for feature points across different cameras can be significantly reduced, and therefore the recovery of the global scale, can be improved. A planar example is shown in Fig. 4.7. Additionally, once the scale has been recovered, the movement of features from one camera's image to another's can be accurately predicted from the relative pose parameters provided the camera extrinsic parameters are properly calibrated.

The final requirement for adding the monocular camera measurements to the estimation is to assign appropriate measurement noise parameters. While these values can be influenced by the relative motion profile, they are more significantly impacted by the feature extraction method employed, and the nature of the features

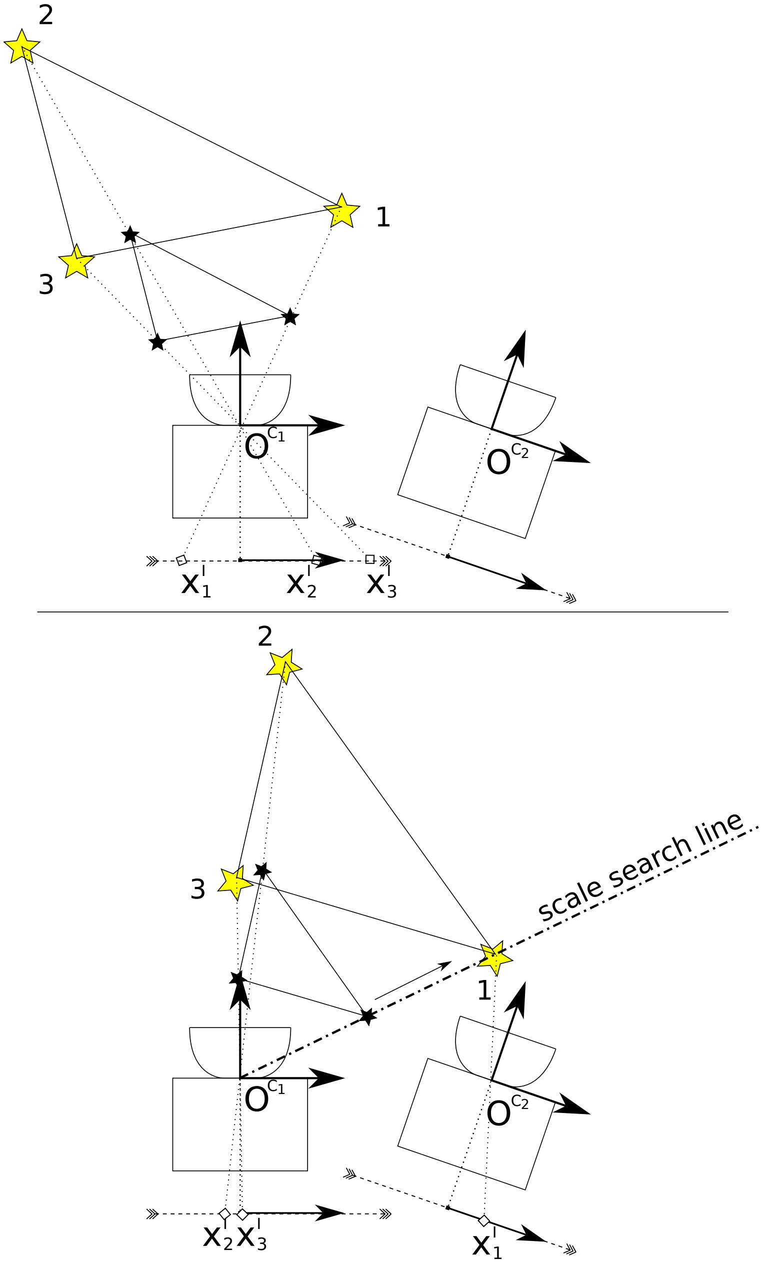


Figure 4.7: When the target object (yellow stars) is visible in only one camera, the shape and orientation (estimated at black stars) can be recovered. From this information, the *scale search line* for feature point one in the second camera specifies where to look for the feature point, and the distance along this line resolves the global scale provided that the baseline between the cameras is known. The black star estimates now move to align with the true scale and the yellow stars.

themselves.

While not a requirement for this system to operate successfully, the variance of the noise process will be assumed constant for this discussion. Accordingly, the covariance submatrix associated with the monocular camera measurements has a diagonal element corresponding to each individual feature image plane coordinate in each camera,

$$\mathbf{N}_k^C = \begin{bmatrix} \mathbf{N}_k^{C_1} & & \\ & ... & \\ & & \mathbf{N}_k^{C_\alpha} \end{bmatrix}, \tag{4.44}$$

where the diagonal submatrix for the i^{th} camera is $2N \times 2N$ such that,

$$\mathbf{N}_k^{C_i} = \begin{bmatrix} \sigma_{x_{C_i}}^2 & & & & \\ & \sigma_{y_{C_i}}^2 & & & \\ & & ... & & \\ & & & \sigma_{x_{C_i}}^2 & \\ & & & & \sigma_{y_{C_i}}^2 \end{bmatrix}. \tag{4.45}$$

Using appropriate values for the measurement noise parameters, this framework will combine the available feature point measurements from all α cameras. The next section looks at adding range measurements from point range finders to supplement the estimation system.

Point Range Finders

For a set of λ point range finders, the individual range measurements at time-step k are arranged into a $\lambda \times 1$ vector,

$$\mathbf{g}_k^G(\tilde{\mathbf{x}}_k) = \begin{bmatrix} \mathbf{g}_k^{G_1}(\tilde{\mathbf{x}}_k) \\ ... \\ \mathbf{g}_k^{G_\lambda}(\tilde{\mathbf{x}}_k) \end{bmatrix} \tag{4.46}$$

where

$$\mathbf{g}_k^{G_i}(\tilde{\mathbf{x}}_k) = z_{avg}^{G_i}. \tag{4.47}$$

The measured range is assumed to be the mean of the depth of all of the object feature points as measuremed by the i^{th} range sensor. The average depth with respect to the i^{th} range sensor is determined from the average feature location in

M which is found from the state variables,

$$z_{avg}^{G_i} = [\ 0\ 0\ 1\ 0\ ]\ \mathbf{T}_{C_1}^{G_i}\mathbf{T}_{M}^{C_1}\begin{bmatrix} x_{avg}^{M} \\ y_{avg}^{M} \\ z_{avg}^{M} \\ 1 \end{bmatrix}. \tag{4.48}$$

where

$$\begin{bmatrix} x_{avg}^{M} \\ y_{avg}^{M} \\ z_{avg}^{M} \\ 1 \end{bmatrix} = \frac{1}{N}\sum_{j=1}^{N}\left(\begin{bmatrix} p_j \\ q_j \\ r_j \\ 1 \end{bmatrix} + \frac{1}{\tau_j}\begin{bmatrix} \cos\gamma_j \sin\chi_j \\ \sin\gamma_j \sin\chi_j \\ \cos\chi_j \\ 0 \end{bmatrix}\right). \tag{4.49}$$

Similar to the monocular camera models, the coordinate transformation from the C_1 frame to the point range finder G_i frame, $\mathbf{T}_{C_1}^{G_i}$, is assumed known from calibration. The measurement Jacobian for the range sensor measurement model is developed in Appendix A.7.

The zero-mean Gaussian measurement noise associated with each range measurement is parameterized by the covariance submatrix,

$$\mathbf{N}_k^G = \begin{bmatrix} \sigma_{G_1}^2 & & \\ & \ldots & \\ & & \sigma_{G_\lambda}^2 \end{bmatrix}. \tag{4.50}$$

Significantly, the covariance values specified in this matrix are *not the true noise variance* of the range sensor itself. Instead, the measurement noise is assumed to also contain the modelling error arising from the assumption that the range measurement is the average feature point depth. Because the surface geometry is not available from feature point locations, the assumptions used to form the range measurement model will be violated when the reflection surface is not at the average feature depth. If the range sensor strikes a surface that is closer than the average feature point depth, this error acts like a bias and will lead to the recovery of a scale which is too small. Conversely, if the surface is further away than the mean feature depth, the scale will converge to a larger value than the true global scale.

By artificially inflating the noise covariance, the relative weighting of the camera and range measurements will reduce the impact of the range measurements on the state estimate. In this way, the range measurements act to slowly influence the scale of the solution and, as an added benefit, will increase the sensitivity of the estimation in the Z^{C_i}-axis direction [28].

Adding multiple range sensors should tend to find the average surface depth more effectively than a single range finder. Measuring the range to several points on the object surface may help to remove the bias and lead to a more accurate global scale of the solution. This will allow the filter to be more robust to the modelling error caused by non-ideal target object surface geometries.

Laser Pointers

A set of β laser pointing devices can be used to make range measurements via the first monocular camera by stacking the image plane measurements of the laser dots into a $2\beta \times 1$ vector,

$$\mathbf{g}_k^L(\tilde{\mathbf{x}}_k) = \begin{bmatrix} \mathbf{g}_k^{L_1}(\tilde{\mathbf{x}}_k) \\ \dots \\ \mathbf{g}_k^{L_\beta}(\tilde{\mathbf{x}}_k) \end{bmatrix} \tag{4.51}$$

where each element includes the x^I and y^I coordinates of the i^{th} laser's dot image,

$$\mathbf{g}_k^{L_i}(\tilde{\mathbf{x}}_k) = \begin{bmatrix} -F_x^{C_1} \frac{x_{l_i}^{C_1}}{z_{l_i}^{C_1}} & -F_y^{C_1} \frac{y_{l_i}^{C_1}}{z_{l_i}^{C_1}} \end{bmatrix}^T . \tag{4.52}$$

The coordinates of the average depth of the feature points from the laser pointer L_i frame are transformed into the C_1 frame via the homogeneous transformation $\mathbf{T}_{C_1}^{L_i}$, again known from calibration,

$$\begin{bmatrix} x_{l_i}^{C_1} \\ y_{l_i}^{C_1} \\ z_{l_i}^{C_1} \\ 1 \end{bmatrix} = \mathbf{T}_{L_i}^{C_1} \; diag([\ 0\ 0\ 1\ 1\ ])\ \mathbf{T}_{C_1}^{L_i} \mathbf{T}_M^{C_1} \begin{bmatrix} x_{avg}^M \\ y_{avg}^M \\ z_{avg}^M \\ 1 \end{bmatrix} . \tag{4.53}$$

The measurement model is linearized at each time step about the current state estimate to find the measurement Jacobian for use in the EKF framework. This matrix is computed as shown in Appendix A.8.

Parameterizing the noise process for these measurements is not as straightforward as for the two previous classes of devices. While the laser dots are being measured by the camera, there is significant modelling error in the range measurement similar to the point range sensor which must be represented. Additionally, while there are two measurements for each laser dot, a given relative position and orientation of the laser pointer with respect to the camera dictates that the dot can only appear on a line within the image plane and the two measurements are

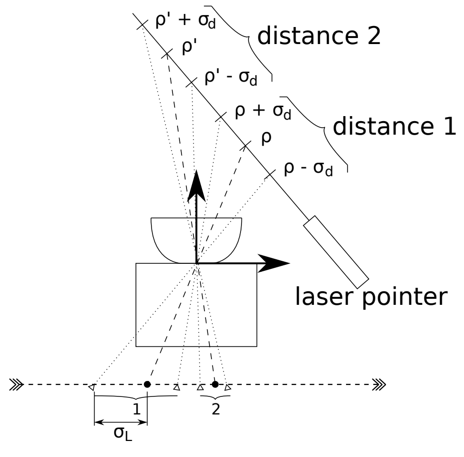


Figure 4.8: The projection of the depth uncertainty for the two different range measurements (ρ) with a standard deviation of σ_d, on to the camera image plane. Close range measurements cover larger regions of the image-plane, but the precision decreases with distance.

dependent. Accordingly, it would be possible to use only one of the measurements to estimate the depth as long as the line was not perpendicular to that image plane axis. Using the two coordinates provides some degree of redundancy in the measurement.

An additional problem arises in the sensitivity of the depth measurement. The magnitude of the noise variance changes with the depth of the laser dot. A variance of a few pixels may represent an uncertainty of a few millimeteres for close objects, but can represent tens of metres or more at long range. As a result, the noise covariance cannot be held constant over every time step for every measurement. Fig 4.8 demonstrates the sensitivity of such a sensor to the depth of the measurements.

For this device, it will be assumed that the covariance of the measurement noise is related to the measured distance, ρ, which can be found by either of the image plane coordinates of the laser dot (x_l^I, y_l^I),

$$\rho = -\frac{x_l^I Z_L + F_x X_L}{F_x \sin\theta_L + x_l^I \cos\theta_L \cos\psi_L}, \tag{4.54}$$

or

$$\rho = \frac{y_l^I Z_L + F_y Y_L}{F_y \cos\theta_L \sin\psi_L - y_l^I \cos\theta_L \cos\psi_L}, \tag{4.55}$$

where the relative position and orientation of the laser pointer L frame with respect to the first camera C_1 frame is known and parameterized by,

$$\mathbf{T}_L^{C_1} = \begin{bmatrix} & X_L \\ \mathbf{R}_{X^{C_1}}(\psi_L)\mathbf{R}_{Y^{C_1}}(\theta_L) & Y_L \\ & Z_L \\ \mathbf{0} & 1 \end{bmatrix}. \tag{4.56}$$

The laser dot location in the L frame is $[\ 0\ 0\ \rho\ ]^T$. If the uncertainty associated with the distance measure has a standard deviation of σ_d, the image plane coordinates of one standard deviation can be found by projecting $[\ 0\ 0\ (\rho{-}\sigma_d)\ 1\ ]^T$ into the camera image plane. The near end of the range leads to greater uncertainty, so it will be used so that the uncertainty in the depth is not underestimated. Therefore, the associated variance for a range measurement is found by calculating,

$$\begin{bmatrix} -F_x x_{\sigma_d}^C \\ -F_y y_{\sigma_d}^C \\ z_{\sigma_d}^C \end{bmatrix} = \mathbf{\Phi}\mathbf{T}_L^{C_1} \begin{bmatrix} 0 \\ 0 \\ \rho - \sigma_d \\ 1 \end{bmatrix}, \tag{4.57}$$

and projecting this point into the first camera's image plane to find the variance in this frame, resulting in,

$$\sigma_{x_{L_i}}^2 = \left(\frac{-F_x x_{\sigma_d}^C}{z_{\sigma_d}^C} - x_l^I \right)^2, \tag{4.58}$$

and

$$\sigma_{y_{L_i}}^2 = \left(\frac{-F_y y_{\sigma_d}^C}{z_{\sigma_d}^C} - y_l^I \right)^2. \tag{4.59}$$

The variances of each of the measurements from the β laser pointers are compiled together to form the covariance matrix,

$$\mathbf{N}_k^L = \begin{bmatrix} \sigma_{x_{L_1}}^2 & & & & \\ & \sigma_{y_{L_1}}^2 & & & \\ & & \cdots & & \\ & & & \sigma_{x_{L_\beta}}^2 & \\ & & & & \sigma_{y_{L_\beta}}^2 \end{bmatrix}. \tag{4.60}$$

4.4.3 Initialization

The initialization of the system follows directly from Section 3.3.7. When the additional range sensors are supplemented for the estimation, their initial measurements can be used to give a better starting estimate for the inverse depth parameter of the feature points than was possible in Chapter 3 when a single camera was the only device.

For the point range sensors, an initial distance measurement, ρ, plus an estimate of the depth variation of the feature point coordinates, can be used to find an appropriate initial mean and covariance for the τ_j parameter with the values,

$$d_{min} = \rho - 2\sigma_{G_i} - 2\sigma_d, \tag{4.61}$$

$$d_{max} = \rho + 2\sigma_{G_i} + 2\sigma_d, \tag{4.62}$$

where σ_{G_i} is the standard deviation for the Gaussian measurement noise associated with the i^{th} point range sensor, and σ_d is an extra term to quantify the variability in depth between all of the feature points. Since the initial inverse depth is applied to all of the feature points, using only the range measurement to quantify a high degree of confidence may be problematic if the real feature depth is far from this single range measurement. This last parameter, σ_d, can be expressed by a 95% confidence interval as well.

The same method can be used for the laser pointer initial measurements with the range calculated by the laser dot image plane coordinates using equations (4.54) or (4.55).

When a second camera is available, the feature point measurements in the two initial frames can be used to calculate an accurate initial estimate for full 3D position of all of the feature points individually. However, the mechanics of this method are outside the scope of this work.

4.5 Summary

This chapter outlined a complete framework for estimating the relative position and orientation of an unknown target object with respect to a robot end-effector. By using a monocular camera as well as a set of supplemental measurements, the relative pose and target object model can be recovered to the proper scale. The following chapter will compare the performance of this estimation using different sets of these supplemental measurement devices.

Chapter 5

Simulations

In order to validate and evaluate the algorithms listed in the previous chapters, a simulation study was conducted in the *MATLAB* software package on a virtual robotic environment. A robotic manipulator is equipped with a variety of sensors, with which, direct relative, noisy measurements of a target object are used to attempt to estimate both the relative pose and target object model. By using combinations of the available sensors, the effects of different relative measurements on the estimated parameters were evaluated for accuracy and robustness under a specified relative motion profile.

5.1 Environment

The simulation environment must provide a reasonably accurate representation of the true conditions under which the estimation proceeds and the significant dynamics of the real system should be captured. Only then can a valid comparison of each of the presented techniques be made and the resulting conclusions be asserted confidently.

The general simulation environment is shown in Fig. 5.1. The robot end-effector carries four sensors which make relative measurements of the virtual target object. In the following sections, the implementation of each of these components is detailed.

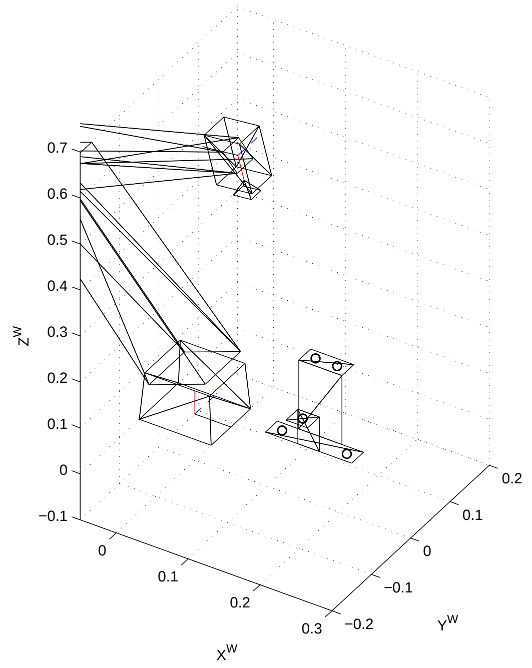


Figure 5.1: The simulation environment containing virtual robot manipulator with monocular camera fixed to end-effector and target workpiece to be tracked.

5.1.1 Target Object

In the simulation, the target object to be tracked must properly interact with the sensors mounted on the robot end-effector. There are two main classes of sensors in use for this system: the cameras are used to recognize and measure point features on the object when they are visible in their image planes; and the range sensors involve a ray striking a solid surface on the target object. As a result, information about the visible features, as well as the surface geometry of the target objectmust be available in the simulation environment.

With this in mind, objects are defined with a local coordinate frame, and consist of a set of feature *points*, assumed identifiable when projected into the camera image planes, and a mesh of *triangles* defined as a triple from a set of *vertices*. The trigonal mesh is similar to that used in computer graphics for defining and rendering three-dimensional scenes for film or interactive games. By computing the intersection of the range sensor ray and the trigonal mesh, the simulated measurement is produced by finding the distance to this point and adding a specified amount of zero-mean Gaussian noise.

Once the object geometry is defined, a scripted set of rigid motions for the target object are specified in terms of translations and rotations of the object's local coordinate frame with respect to the world frame. In these simulations, the camera position and orientation will be held constant, and therefore, the relative motion will be completely defined by the target object motion in the world frame. However, since there are no measurements made with respect to the world frame, this is just a simpler way of defining the relative motion of the system and the choice has no effect on the quality or validity of the estimation.

The virtual target object used in this experiment is shown in Fig. 5.2. It consists of five feature points distributed across three different level surfaces.

5.1.2 Sensors

The simulation environment is set up to compare the performance of the relative pose estimation when a combination of relative measurements of the target object are used. Four sensors are available in the simulations: two monocular cameras; a point laser range-finder; and a simple laser pointer. Each sensor is mounted on the end-effector of the robot with some known position and orientation with respect to the first camera frame C_1, resulting in a transformation $\mathbf{T}_S^{C_1}$ from the sensor frame

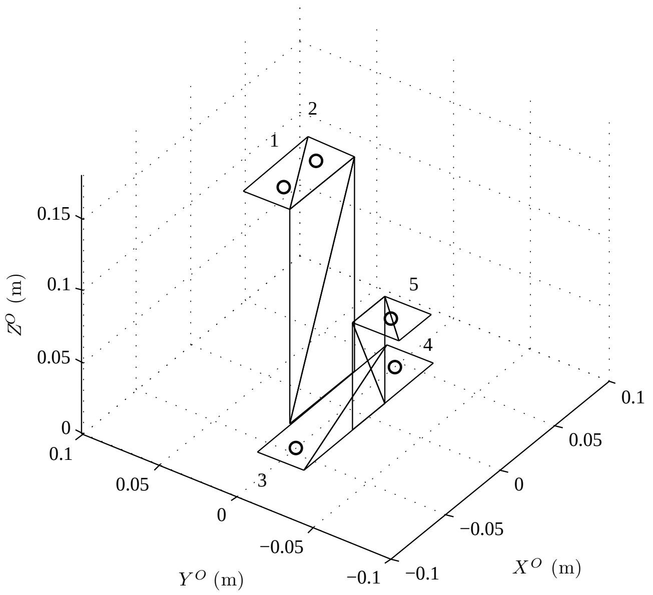


Figure 5.2: The target object used in the simulations. The five feature points (circles) are visible to the monocular cameras, and range sensor beams intersect with the trigonal mesh surfaces.

S to the C_1 frame. The relative pose estimate is always made with respect to the first camera frame, so these transformations serve to move all of the measurements to a common frame of reference.

First Camera

The first monocular camera is assumed to be used *in all simulations* and the estimation of the target object motion proceeds within this frame of reference. The *camera one frame (C_1)* is defined such that the origin is located at the center of projection, with the Z^{C_1}-axis aligned with the optical axis. The X^{C_1} and Y^{C_1} axes are arranged as previously and align with the image plane X^{i_1} and Y^{i_1} axes, respectively. Finally, the camera has an associated field of view specifying a cone within which feature points are visible in front of the camera. This is to simulate the limited view of a physical camera and to allow objects to move in and out of the image plane. When an object feature point is not visible within the camera image, the estimation must proceed without measurements of that feature point at that time step.

This camera simulates the base sensor to which the other sensors are added to provide the required supplemental measurements when estimating global scale. Since the position of the object feature points is available at each time step, the image plane projections of each feature can be calculated using a pin-hole camera model disturbed by zero-mean Gaussian noise. These noisy measurements are then passed to the estimator at each time step along the relative motion trajectory.

Second Camera

The second monocular camera is optional and provides a second image of the object feature points with a known base-line to the first camera, specified by the transformation, $\mathbf{T}_{C_2}^{C_1}$. It is assumed that the correspondance problem of which object feature points correspond to which image plane measurements has been solved by the feature extraction phase. This problem is not trivial, but it will not be considered further here.

The *camera two frame (C_2)* is defined identically to the C_1 frame but with respect to the second camera. Additionally, the measurements from this sensor are simulated in exactly the same fashion at the first camera and are available at each simulation time step.

It is important to note that this technique does not involve first creating a disparity map. Instead, the two monocular cameras operate independently and only provide raw image plane measurements of the feature points. The depth information is extracted in the filtering phase,. according to the measurement model constraints.

Point Range-Finder

As a supplement to the monocular camera, the range finder provides a measure of the global scale of the estimation. The point range-finder sensor projects a ray from the origin of the G frame along the Z^G-axis and measures the shortest distance to the intersection with the target object geometry. The associated *range-finder frame (G)* is defined such that the beam originates at the origin and the Z^G-axis is aligned with the direction of the ray. Once again, it is assumed that the relative position and orientation of this sensor with respect to the C_1 frame is known with certainty and parameterized by the transformation, $T_G^{C_1}$.

This sensor model is similar to that of a sonar or ultrasonic range finder with a cone radius of zero. Alternatively, these 'coarse' sensors may be approximated by increasing the measurement noise covariance to simulate the added uncertainty. However, the basic information provided is identical.

To simulate the measurements from such a device, the triangles specifying to object surface geometry are used sequentially to calculate the distance to an intersection with a ray, originating from the range finder frame origin down the Z axis, representing the measurement beam. It is then checked that the intersection is within the boundaries of the triangle, and the distance to the closest valid intersection is used as the measurement after adding zero-mean Gaussian noise.

Laser Pointer

While the laser pointer is not a measurement device on its own, when combined with a camera, it can be thought of as a simple stereo-vision configuration. When the base-line between the camera and laser pointer are known, $T_L^{C_1}$, the reflection of the laser pointer, the *laser dot*, can be recogized in the camera image plane, providing a bearing measurement. Along with the known bearing with respect to the laser pointer, this measurement can be used to triangulate the depth of the laser dot in the environment. If it is assumed that the reflection occurs on the target object, a rough measurement of the depth to the object can be made.

The *laser pointer frame (L)* is defined similar to the G frame with the origin as that laser ray origin and the Z^L-axis aligned with the ray itself. Since the laser dot is measured in the camera image-plane, the actual measurements will be the x_L^i, y_L^i coordinates of the image of the laser dot within the image plane.

Similar to the laser range finder above, the simulated measurement is found by first finding the closest intersection of the laser beam ray projecting out from the origin. Then, the coordinates of this intersection are transformed into the C_1 frame and projected onto the camera's image plane. Zero-mean Gaussian noise is added to these image measurements before being passed to the estimator at each time step.

5.1.3 Sensor Selection

In this analysis, the number and combinations of the sensors used will be varied. This will serve to compare the performance of the estimation of the relative pose with different combinations of sensors enabled.

To serve as the baseline, the case when only a single monocular camera is used with a known object model, will be explored first. This example closely resembles the work of Wilson *et al.* [53] detailed in Chapter 2. Next, the assumption that the target object model is known with certainty, will be relaxed, and both the target model and the relative pose will be estimated using only a monocular camera as outlined in Chapter 3. This case will demonstrate that while the shape and orientation of the target object can be found, the absolute scale of the target object model and the relative motion are not recoverable.

As discussed in Chapter 4, adding supplementary sensors with the appropriate measurement models should allow the scale to be recovered. However, the suitability and validity of the additional sensors and the corresponding measurement models, is of interest. Adding a second monocular camera adds more measurements than is strictly necessary to fully recover the global scale, but serves as a good baseline against which the simpler devices are evaluated. The combination of monocular camera and point range-finder will be investigated. Then, in an attempt to simplify the hardware required, the point range finder will be replaced with a simple laser pointer which is coupled with the monocular camera to make a different range-finder type device. The success of these last two methods depends crucially on how well the measurement model assumptions align with the true system.

5.1.4 Initialization

For the simulations detailed here, the specific initialization values depend on whether the target object model is assumed to be known or unknown. When the target object is assumed to be completely unknown at the start of the estimation, the initial distributions for the system state estimates are determined using the process described in Sections 3.3.7 and 4.4.3.

While the initialization of an estimation with a known object model is similar to the case described in Section 2.3.3, the specific process is more complicated given the framework used in the simulations. The simulations are set up to estimate both relative pose and target object model simultaneously. When the target object is known, the model parameters must be set to the proper values and fixed (i. e. given an associated covariance of zero).

Known Target Model

For this simulation study, a known target model means that the locations of the target object feature points are known with complete certainty. However, unlike the target models detailed in Chapter 2, where the features were known in a local object coordinate frame, here they are known in the modelling frame M, and the object coordinate frame O is defined by a set of three specified feature points.

As a result, the estimation of the relative pose can proceed within the same framework as for the unknown target model estimation, but the target model feature point parameters are initialized with their locations and the corresponding initial estimate covariance values are set to zero. Additionally, the object frame in this case is not defined by three features as is the case for unknown target models. While not the only possibility, here the M and O frames are superimposed and each feature point parameterization is found by,

$$p_j = 0,\ \sigma^2_{p_j0} = 0 \tag{5.1}$$

$$q_j = 0,\ \sigma^2_{q_j0} = 0 \tag{5.2}$$

$$r_j = 0,\ \sigma^2_{r_j0} = 0 \tag{5.3}$$

$$\gamma_j = \tan^{-1}\left(\frac{y_j^O}{x_j^O}\right),\ \sigma^2_{\gamma_j0} = 0 \tag{5.4}$$

$$\chi_j = \tan^{-1}\left(\frac{\sqrt{(x_j^O)^2 + (y_j^O)^2}}{z_j^O}\right),\ \sigma^2_{\chi_j0} = 0 \tag{5.5}$$

$$\tau_j = \frac{1}{\sqrt{(x_j^O)^2 + (y_j^O)^2 + (z_j^O)^2}},\ \sigma^2_{\tau_j0} = 0. \tag{5.6}$$

Subsequently, the relative pose parameters can then be initialized as shown in Section 2.3.3. While this does introduce a small amount of unnecessary computing overhead due to the model feature point parameters still being included in the state vector, it means only the relative pose states will change as a result of the measurements and the results will be identical to a system where the states consist of only the six pose parameters and their velocities as in Chapter 2.

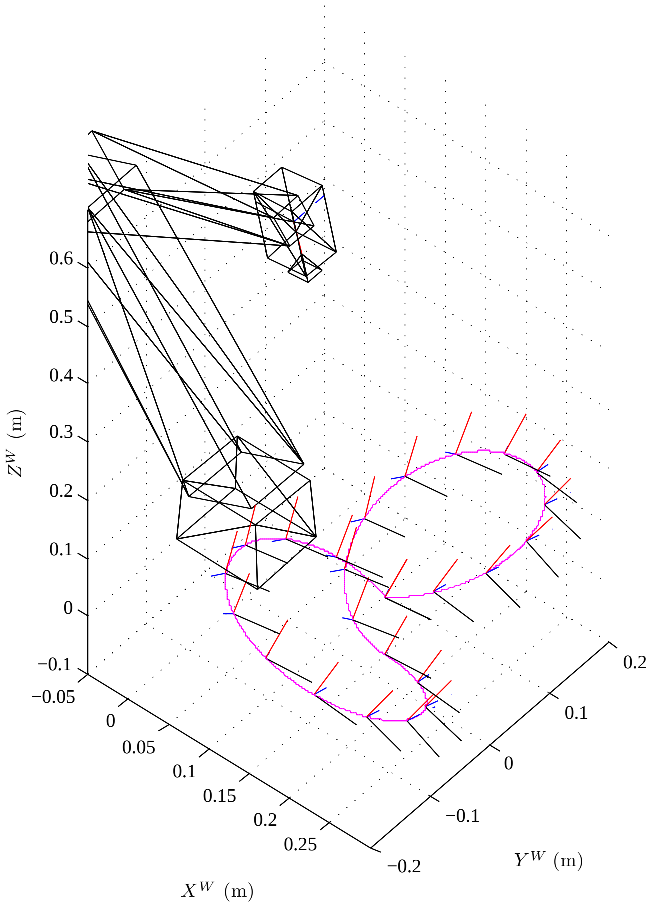


Figure 5.3: The path of the target object is a figure-eight motion in the X^W and Y^W directions with sinusoidal motion in the Z^W axes and rotational motions. The object coordinate frame at intervals are shown to characterize the rotations.

5.2 Procedure

To compare the performance of the estimator with the different sensor measurements, a series of five simulations were carried out under the same relative motion profile of the robot end-effector and the target object, which is shown in Fig. 5.3.

First, to provide a baseline, a single monocular camera is used to estimate only the relative position and orientation of a target object with a known model. For this case, only the six pose parameters and their velocities are estimated.

Second, no information on the object model is available a priori, and the relative pose and object model are simultaneously estimated using measurements from a single camera. This test case is chosen to illustrate how the combined estimation produces a solution which is consistent, but lacks the proper global scale.

Third, a second camera is added to the robot end-effector to demonstrate how two bearing-only sensors with a known baseline can recover the scene depth and scale accurately provided the correspondance problem can be solved reliably. The results from this test will provide a benchmark for comparing the simpler camera plus range sensor configurations in the cases to follow.

Fourth, the measurement space is augmented with output from a point range finder device mounted in a fixed position relative to the robot end-effector. This case will investigate whether the scale can be reliably recovered using this simple measurement device with a limited measurement model and avoiding the complications with a multiple camera setup.

Finally, the range sensor is changed to be the laser pointer and camera setup detailed previously. This setup is potentially much less expensive and its use may be justified if good performance is observed compared to the point range finder.

5.3 Results

In this section, the performance of each of the test cases will be examined in order to compare the ability of each estimator to accurately recover the relative position and orientation of the target object with respect to the robot end-effector frame.

5.3.1 Pose Estimation with Monocular Camera

In the first test case, the traditional position-based visual servo pose estimation scheme with a known target object model is simulated. The motivation of running this scenario is to provide a baseline to represent the ideal case for estimating the *relative pose* of the target object with respect to the monocular camera frame. In the subsequent cases, where the target object model is unknown, it is only reasonable to expect that the estimator would perform at this level, in the best case. It is more likely, when estimating both the relative pose and the target object model, that the relative pose estimates will be hindered by the estimation error in the target object model and vice-versa.

The simulation was run and the resulting estimated relative pose parameters are shown in Fig. 5.4. The six parameters representing the position and orientation of the target object frame are shown through the image sequence compared to the ground truth for the relative motion. While the estimation of the relative orientation is performed in MRP, these parameters have been *converted to Euler angles* to make it easier to visualize and analyze the rotations. It is clear that the estimator is able to accurately track the target object through the entire motion sequence.

A closer inspection of the estimation error, defined $\mathbf{x}_k - \hat{\mathbf{x}}_{k,k}$, is shown in Fig. 5.5. The estimation error on the relative position states is confined to sub-centimetre magnitudes, while the relative orientation estimates are accurate to less than a degree. By comparing the estimation error values to the corresponding estimated parameter, the estimation error follows the trend of the estimation itself. This is caused by the constant velocity motion assumption used to develop the process model for the system. If the relative motion dynamics of the system could be modeled perfectly, the estimation error would consist of only white noise. Since the relative motion is considered arbitrary, the constant velocity assumption is employed to deal with the general case. When the velocity of the relative motion, the slope of the estimation graph, changes quickly, the constant velocity model is a poor representation for the system dynamics, and these unmodeled dynamics appear in the estimation error. If this modeling error becomes significant, a solution may be to reduce the time step by increasing the sampling rate of the system.

A measure of the confidence with which the estimates can be used, the estimate covariance, along with the mean, parameterizes the Gaussian distribution for each state estimate. The covariance of the relative pose parameters over this simulation is shown in Fig. 5.6. Notice that the orientation parameters here are MRP.

These values rise and fall as a function of the disturbance and measurement noise, as well as the system configurations throughout the relative trajectory. The EKF prediction step directly injects the disturbance noise into the estimate covariance at each time step, while the amount to which the covariance is reduced by the measurement update step is heavily dependent on the measurement Jacobian, which is a function of the current state estimate. Different system configurations result in greater or lesser effect on the estimate covariance in this step. By quick observation, it is apparent that when the object is close to the camera, the covariance of the relative pose parameters falls dramatically, particularly the Z component.

This simulation test case confirms that when the target object model is known accurately, the relative position and orientation of the target object frame can be reliably and accurately tracked using a single monocular camera providing image plane measurements of a set of object feature points. It aligns well with the observability analysis performed in Chapter 2 which concluded that in almost all

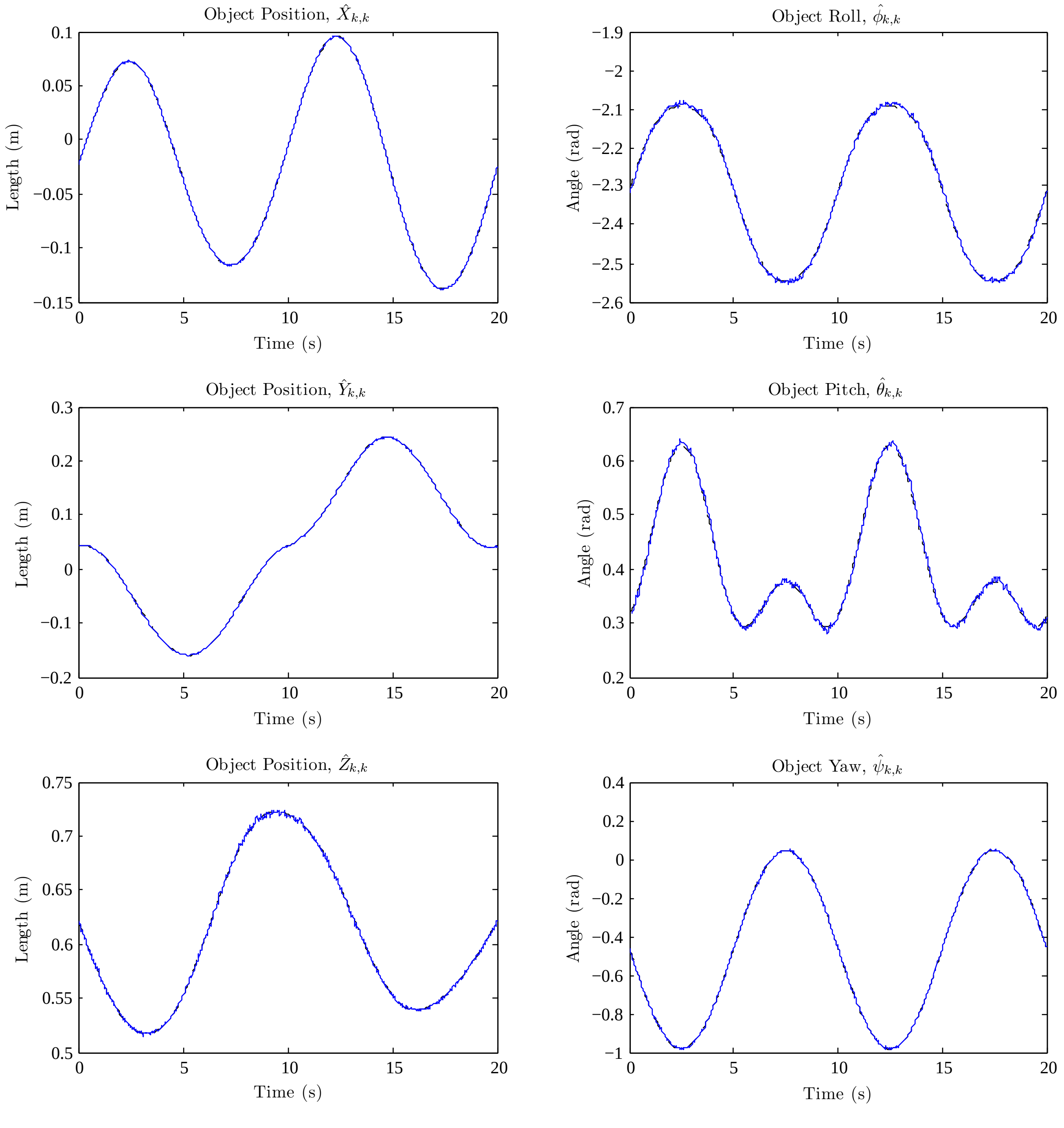


Figure 5.4: Estimates for the six relative pose parameters over the sequence using measurements from a single monocular camera with a known target object model. The estimated MRP have been converted to Euler angles. Actual values are dashed lines, estimated are solid lines.

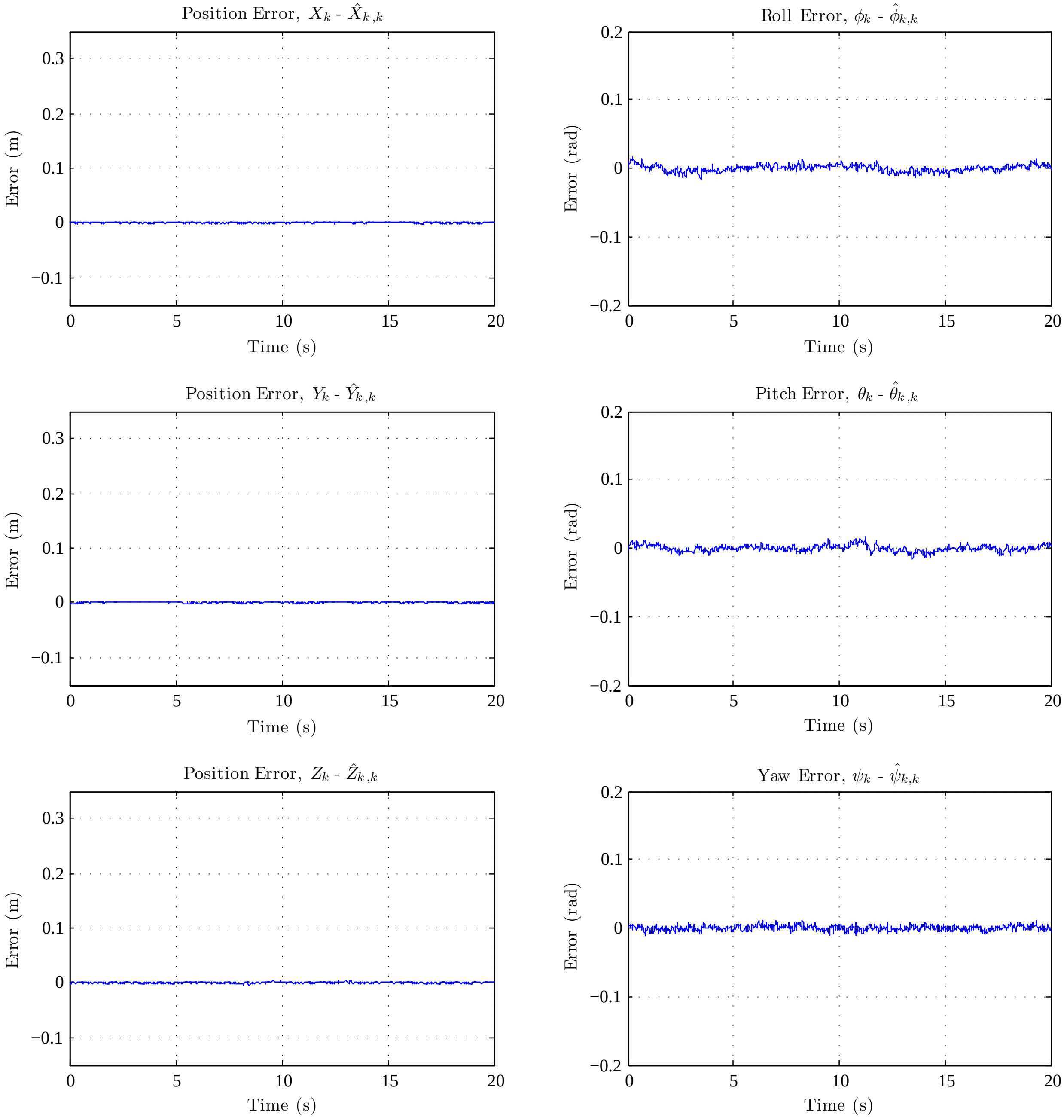


Figure 5.5: Estimation error for the six relative pose parameters through the entire sequence using measurements from a single monocular camera with a known target object model. The estimated MRP have been converted to Euler angles.

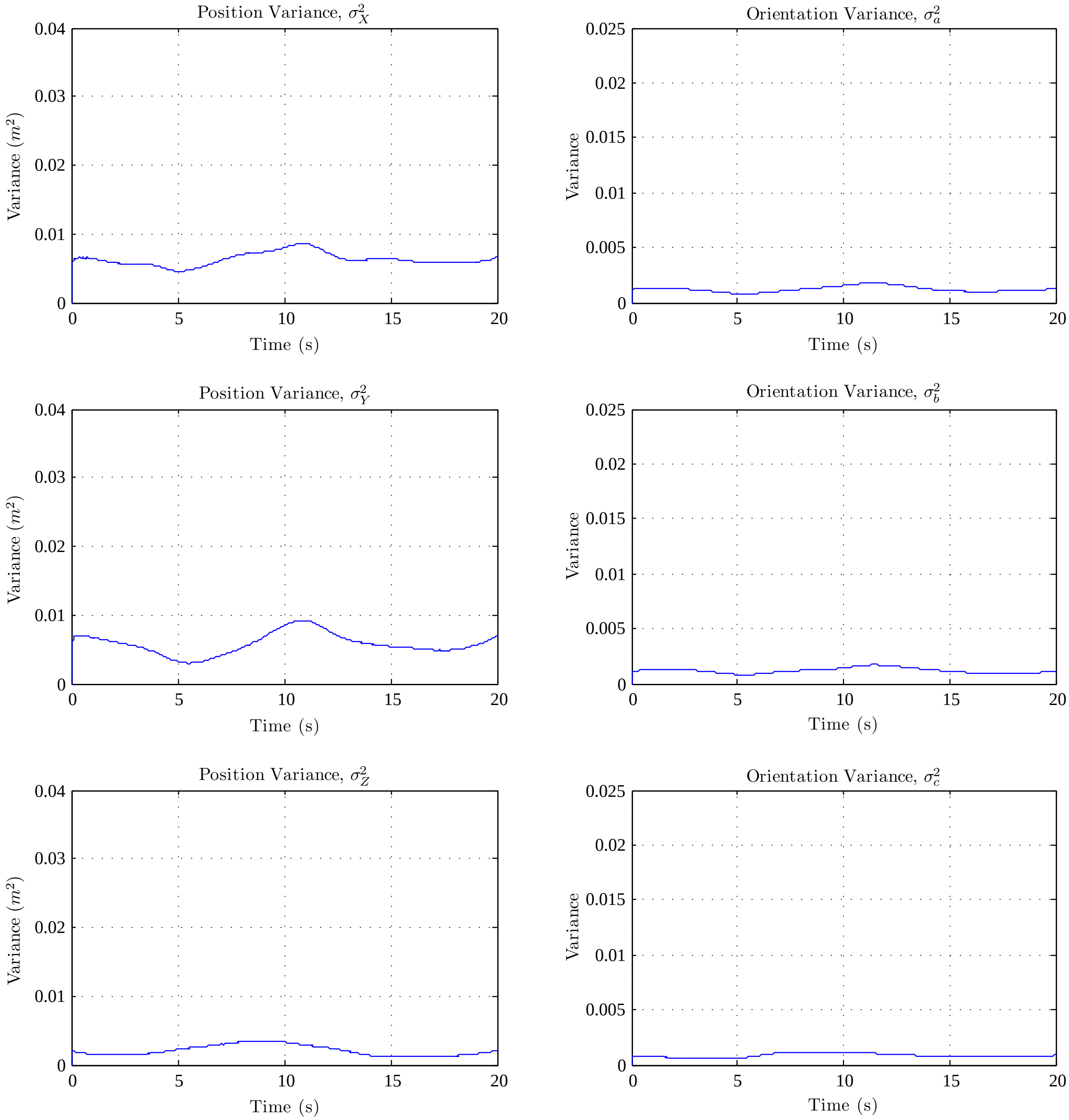


Figure 5.6: Estimate covariance for the six relative pose parameters through the entire sequence using measurements from a single monocular camera with a known target object model.

configurations, recovering the relative position and orientation of the target object should be possible.

Now, when the target object is not assumed to be known in the test cases to follow, the pose estimates can be compared to these results to determine the performance of the augmented system compared with this ideal case. Indeed, unlike the SLAM work, where recovering an accurate map of the world or object is paramount, it is the relative pose estimates that are important for control.

5.3.2 Combined Estimation with Monocular Camera

For this next test case, the only sensor used will again be a single monocular camera. However, the target object model is unknown and must therefore be estimated alongside the relative pose parameters. The target object moves in the same relative motion trajectory as the previous test and the initial setup of the system along with the initial target object model feature point estimates, as well as the final estimate of the pose and model system, are shown in Fig. 5.7.

Upon reviewing the final state estimate, the deficiencies of this method are apparent. The resulting relative pose estimates throughout the motion, are shown below in Fig. 5.8. Looking at the graphs of the relative position parameters, it can be seen that while the estimated motion has the correct trends, the scale is wrong. The estimated position, in this case, is less than the true position by a scale factor at every time step. Additionally, this scale factor is not necessarily constant, but it is not converging to the true scale. This suggests that the theoretical result, specifying that the global scale should not be recoverable by a single monocular camera alone, is correct.

While the global scale is not observable in this system, it is clear that the relative orientation of the target object certainly is recoverable with the available information. Once the shape of the object is determined, around four seconds in this case, the relative orientation parameters are very close to their ground truth values. This is expected since the global scale does not affect the relative orientation estimates. Consider the formation of the rotation matrix, for which the scale parameter does not appear in expression (4.5) as it does in the translation vector.

The estimation error for the relative pose parameters reflects the problems with recovering the global scale. These errors through time are shown in Fig. 5.9. While the error on the position states is significant due to the scale error, the up-to-scale

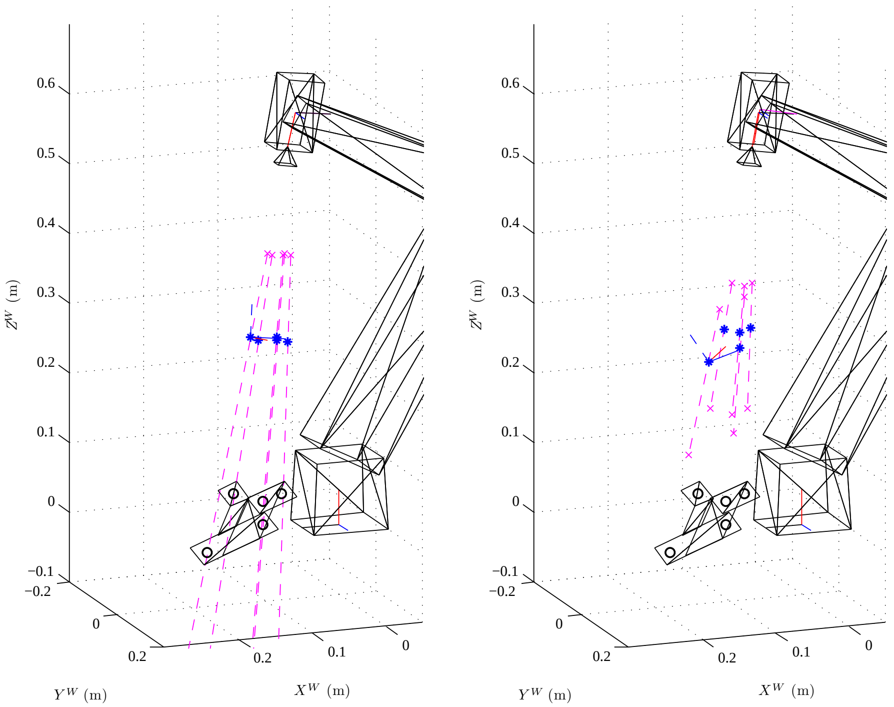


Figure 5.7: Initial (left) and final (right) state estimates for the monocular camera case with unknown target object model. Actual feature point locations are empty circles on target object, estimated locations are solid dots. The 2σ range of the feature point depth uncertainty is represented by the dashed lines.

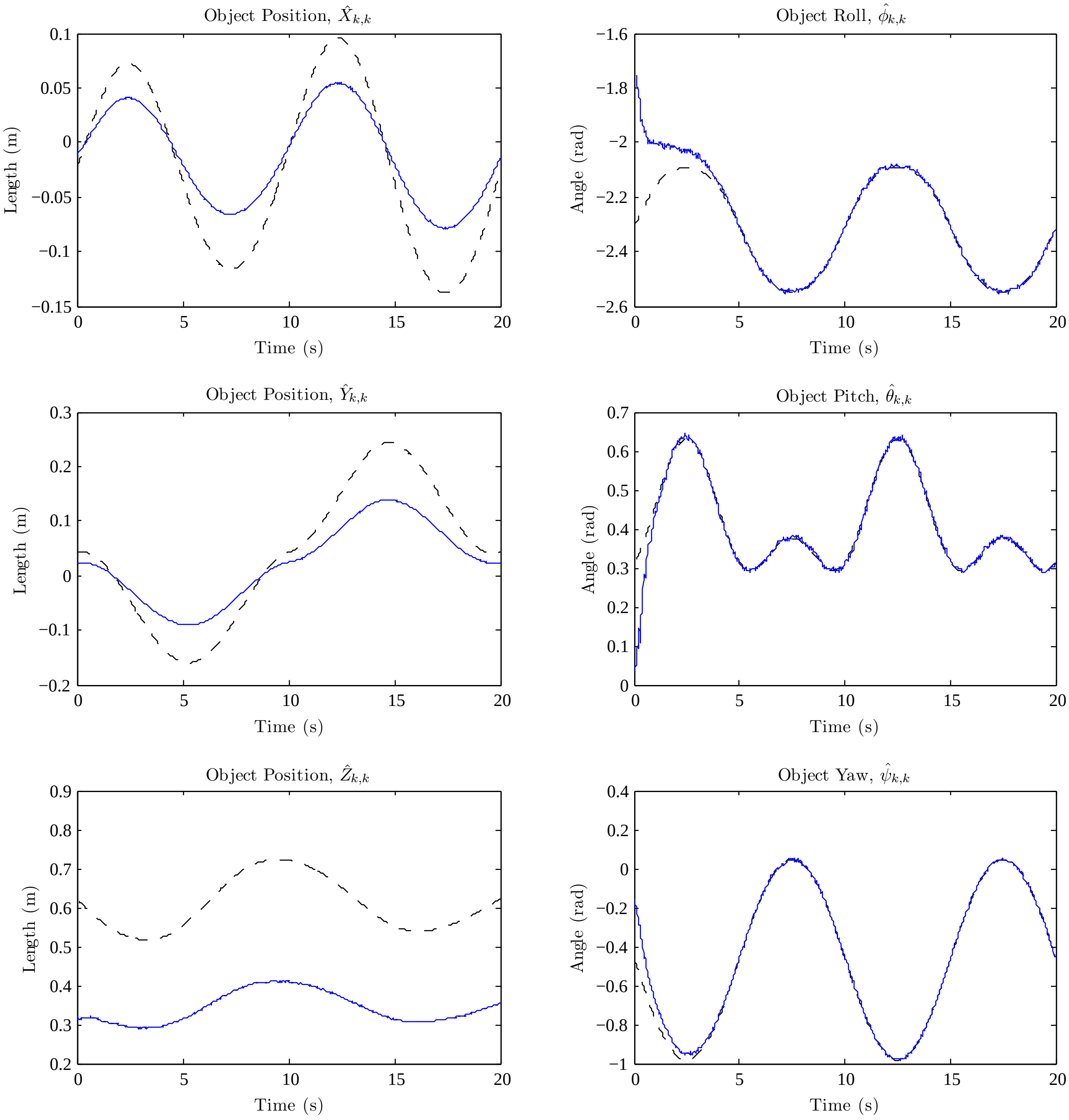


Figure 5.8: Estimates for the six relative pose parameters over the sequence using measurements from a single monocular camera with a completely unknown target object model. Actual values are dashed lines, estimated are solid lines.

solution is consistent and can be tracked successfully. This scale factor should be consistent throughout all of the position states of the system, as well as the feature position estimates at any give time step.

Estimating both the relative pose and the target object model simultaneously has a significant effect on the estimate covariance of the relative pose parameters through the relative motion. The covariances of the pose parameters are shown in Fig. 5.10. Initially, the covariance of all of the parameters grows as a result of the estimator resolving the general shape of the target object. However, just like the relative orientation estimate, once the shape is found with sufficient certainty, around four seconds, the covariance falls sharply for most of the state estimates. The large uncertainty at the beginning of the sequence is associated with the initial conditions of the feature points in which they are arranged in a spherical configuration with the camera being at the sphere's center. This is likely a very sensitive configuration for the system and this is reflected in the growing covariance.

Of secondary interest is how well the estimator is able to recover the target object geometry alongside the relative position and orientation of the target object with respect to the camera frame. Estimates of the parameters for two of the five feature points are shown in Fig. 5.11. Along the left column is the evolution of the estimates for the azimuth, altitude, and inverse depth of feature point one, and the corresponding parameters for feature two are in the right column. Feature points one and two are identified in Fig. 5.2 showing the target object.

Similar to the relative orientation parameters, the azimuth and altitude angles for both feature points are recovered accurately over the entire trajectory. This, once again, suggests that the shape of the object model is observable with the single monocular camera measurements, as expected. It is a different case for the inverse depth parameters since they implicitly rely on scale information for the system. Both parameters converge to a value which is a consistent scale factor of the true inverse depth from the ground truth data.

The estimation error for the feature parameters leads to the same conclusions, and is shown in Fig. 5.12.

The estimate covariance of the feature point parameters are shown in Fig. 5.13. Since there is no disturbance noise associated with the model parameters, the covariances associated with these estimates are monotonically decreasing as suggested in [19]. Therefore, it is a question of how quickly the covariance falls to zero to quantify the quality of the measurement information. The inverse depth parameters, even though they are known to be converging to the wrong scale, have decreasing

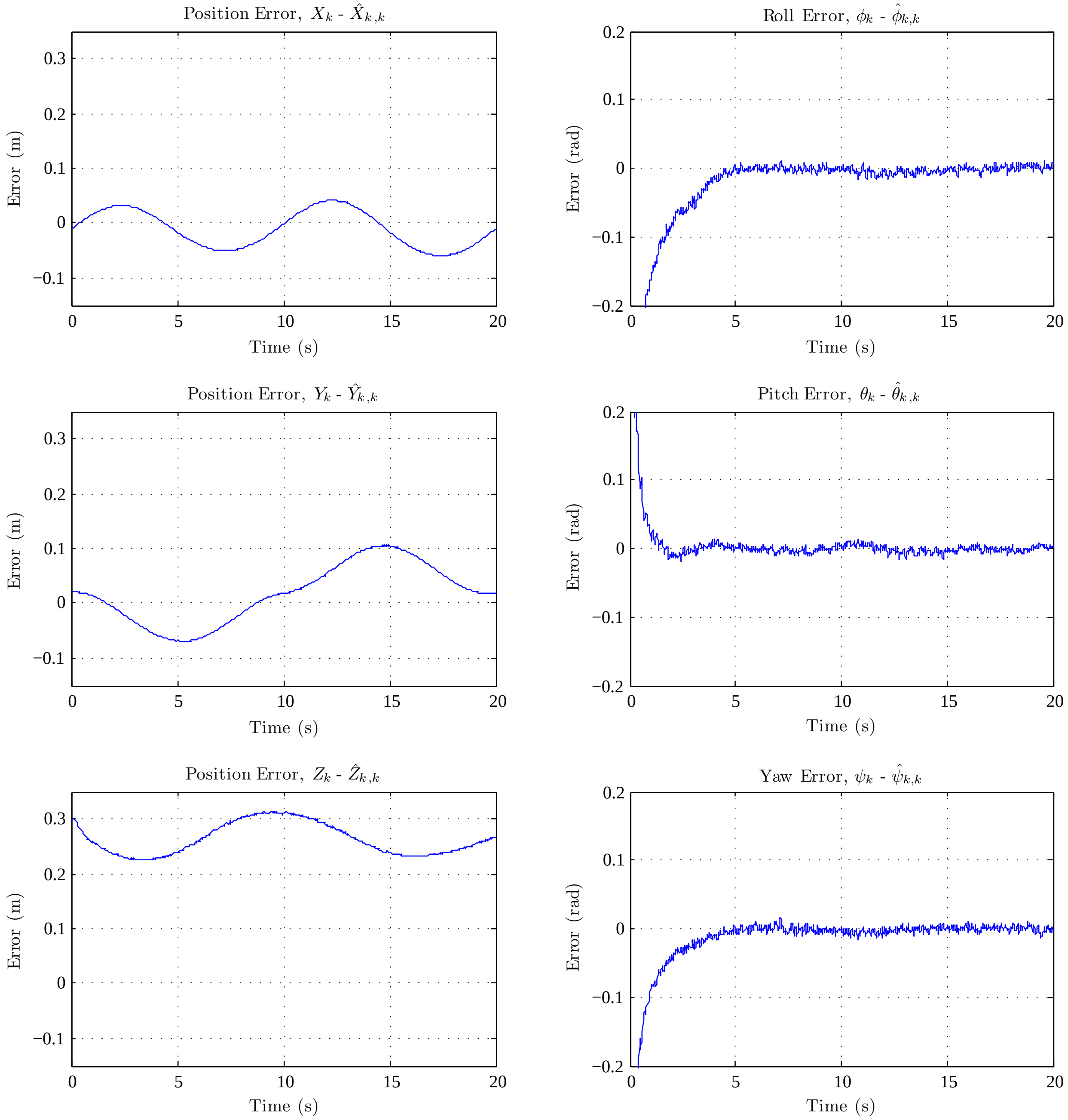


Figure 5.9: Estimation error for the six relative pose parameters through the entire sequence using measurements from a single monocular camera with a completely unknown target object model.

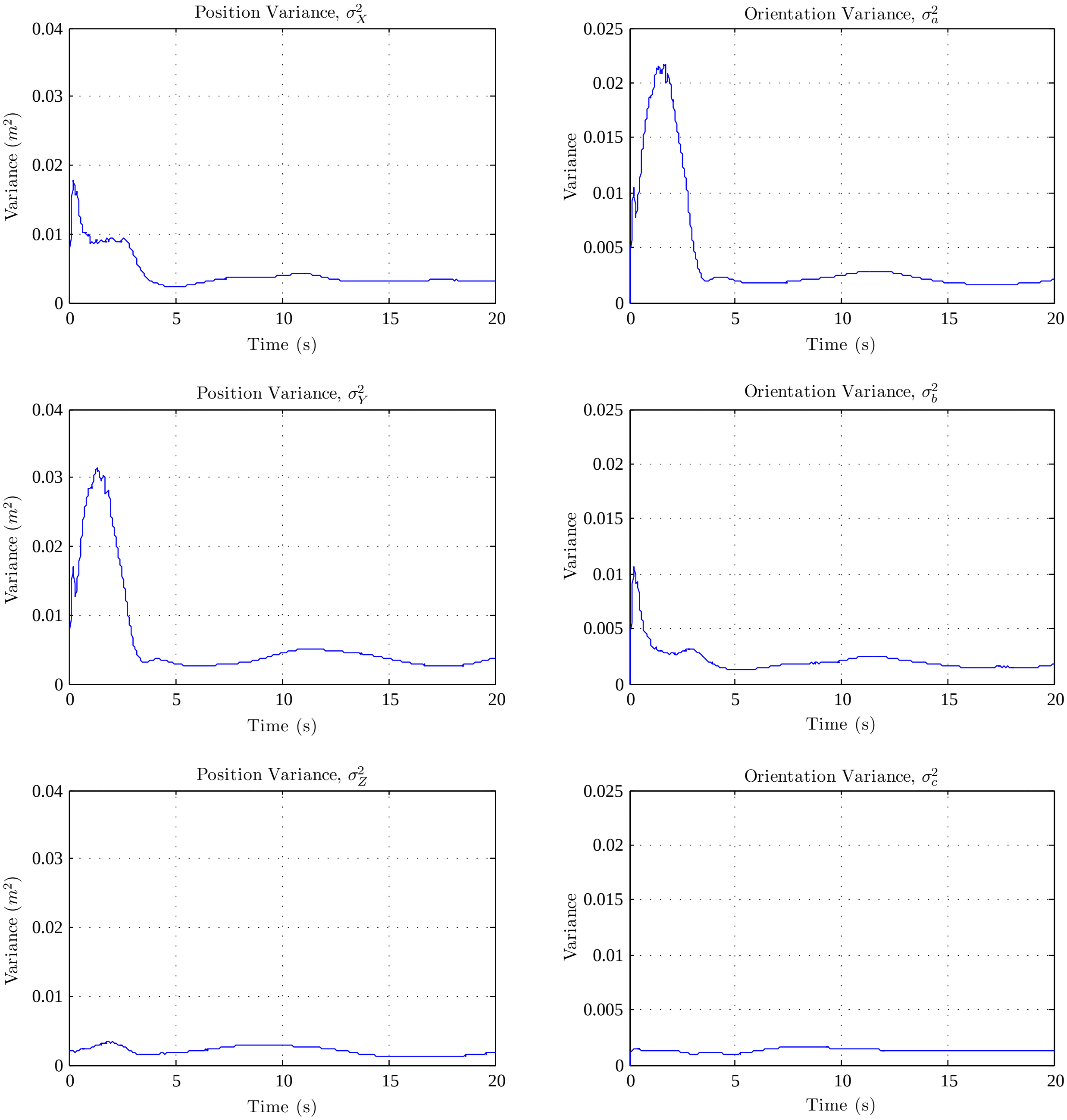


Figure 5.10: Estimate covariance for the six relative pose parameters through the entire sequence using measurements from a single monocular camera with a completely unknown target object model.

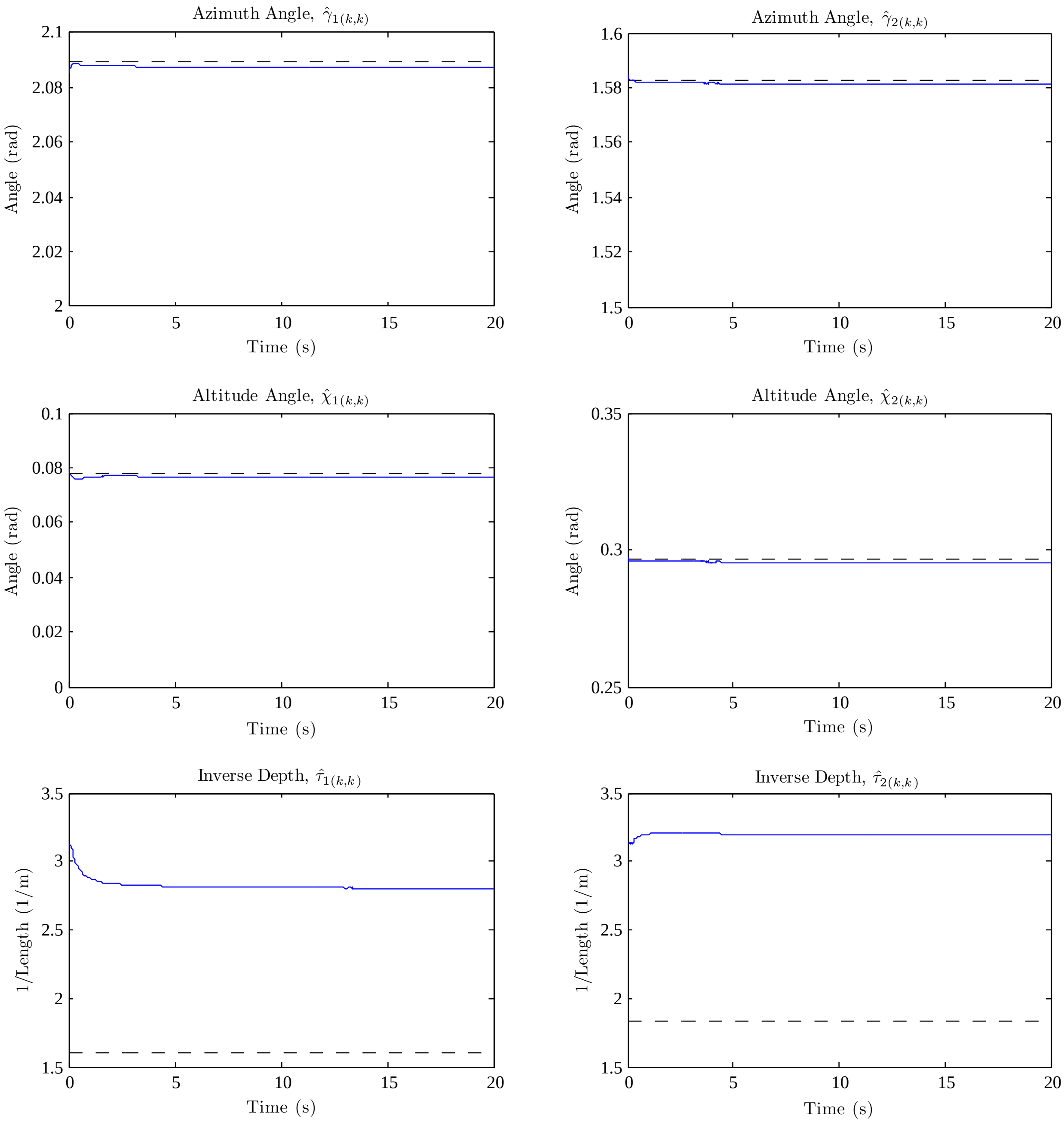


Figure 5.11: Estimates of the modelling frame locations of the first (left column) and second (right column) object feature points from measurements with a single monocular camera with a completely unknown target object model. Actual values are dashed lines, estimated are solid lines.

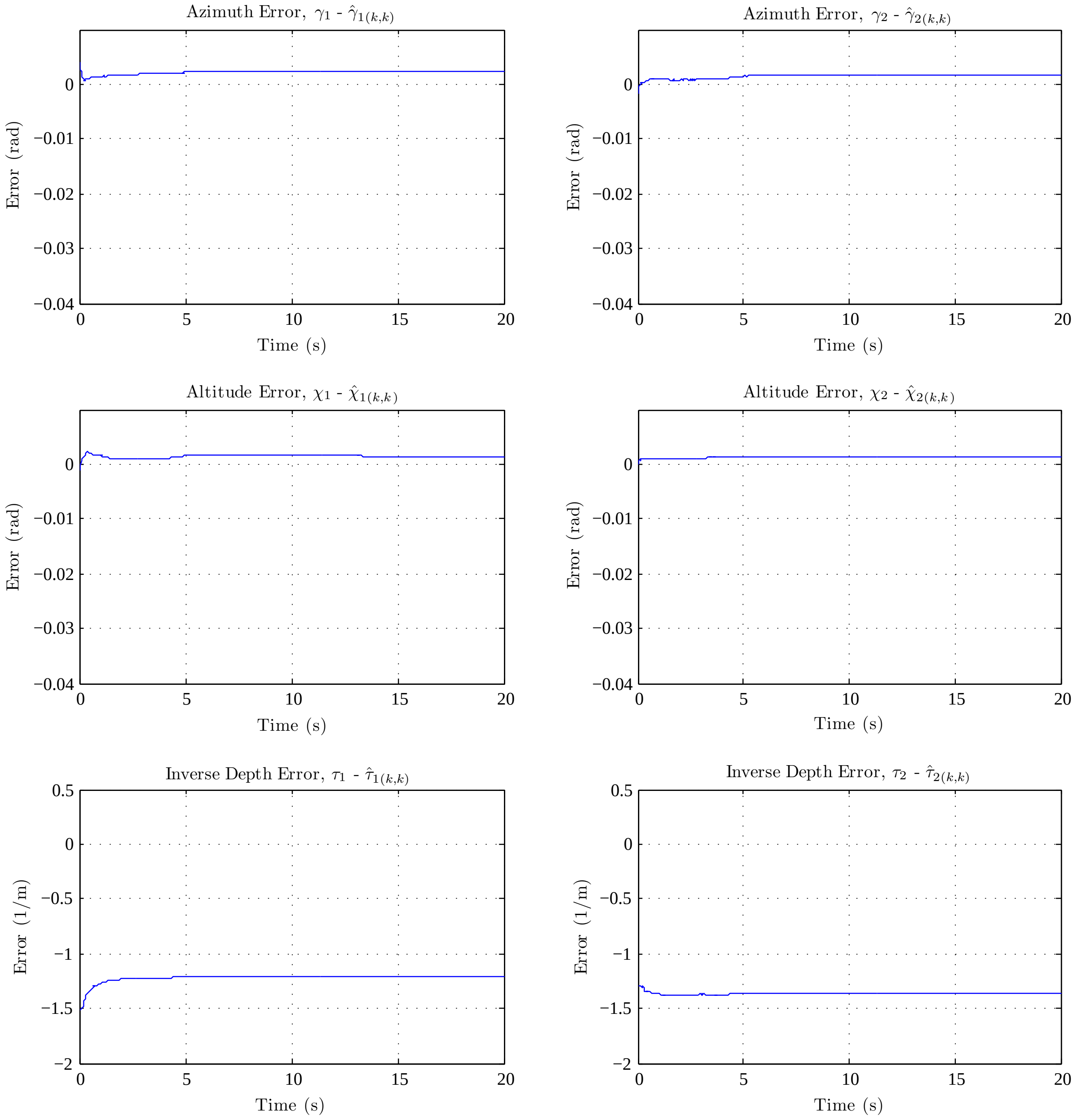


Figure 5.12: Estimation error of the modelling frame locations of the first (left column) and second (right column) object feature points from measurements with a single monocular camera with a completely unknown target object model.

covariances. This is due to the construction of the EKF. In order to apply the EKF to a system, it is assumed that the system is observable. If this were the case, and the nonlinearities do not have a significant effect, the estimate covariance matrix would be adjusted properly according the disturbance and measurement noise magnitudes.

The falling covariance of the inverse depth estimate raises an issue when adding additional measurements which contain scale information. Consider what would happen if the estimation were to initially proceed with only the monocular camera, and then the other measurements were added at a later time. In the current setup, the covariance of the model parameters would converge as shown in Fig. 5.13 and could become effectively *fixed* if the covariance falls to a point to prevent the scale measurements from adjusting the parameters to fit the *proper* scale. This suggests that it is important to keep enough 'mobility' in the system parameters to allow this information to be integrated effectively. A mechanism such as injecting noise into the relevant parameter estimates, or including a seperate scale parameter which is acted on by the range measurements alone, may provide a solution.

The test case explored here, in which a single monocular camera is used to estimate both the relative pose as well as the target object model, demonstrates the problem which the following sections will attempt to overcome. The image plane measurements of the object feature points is not sufficient to recover a solution with the proper global scale. Instead, supplemental measurements which contain scale information are necessary, and the nature of such measurements are investigated in the following test cases.

5.3.3 Combined Estimation with Two Cameras

When attempting to recover the scale of the system, an intuitive option is to add a second monocular camera to the robot end-effector at some fixed position and orientation relative to the first camera. With this baseline known, the distance to the object feature points can be triangulated, and the global scale recovered.

For this test case, a second camera is situated alongside the first monocular camera and the image plane measurements from both cameras are used in the EKF framework to estimate both the relative pose and the target object model of the system. While this method is expected to outperform the monocular camera-only tests both with and without a known target model, it represents a non-minimal measurement set. In this way, it will serve as a good baseline for comparing the

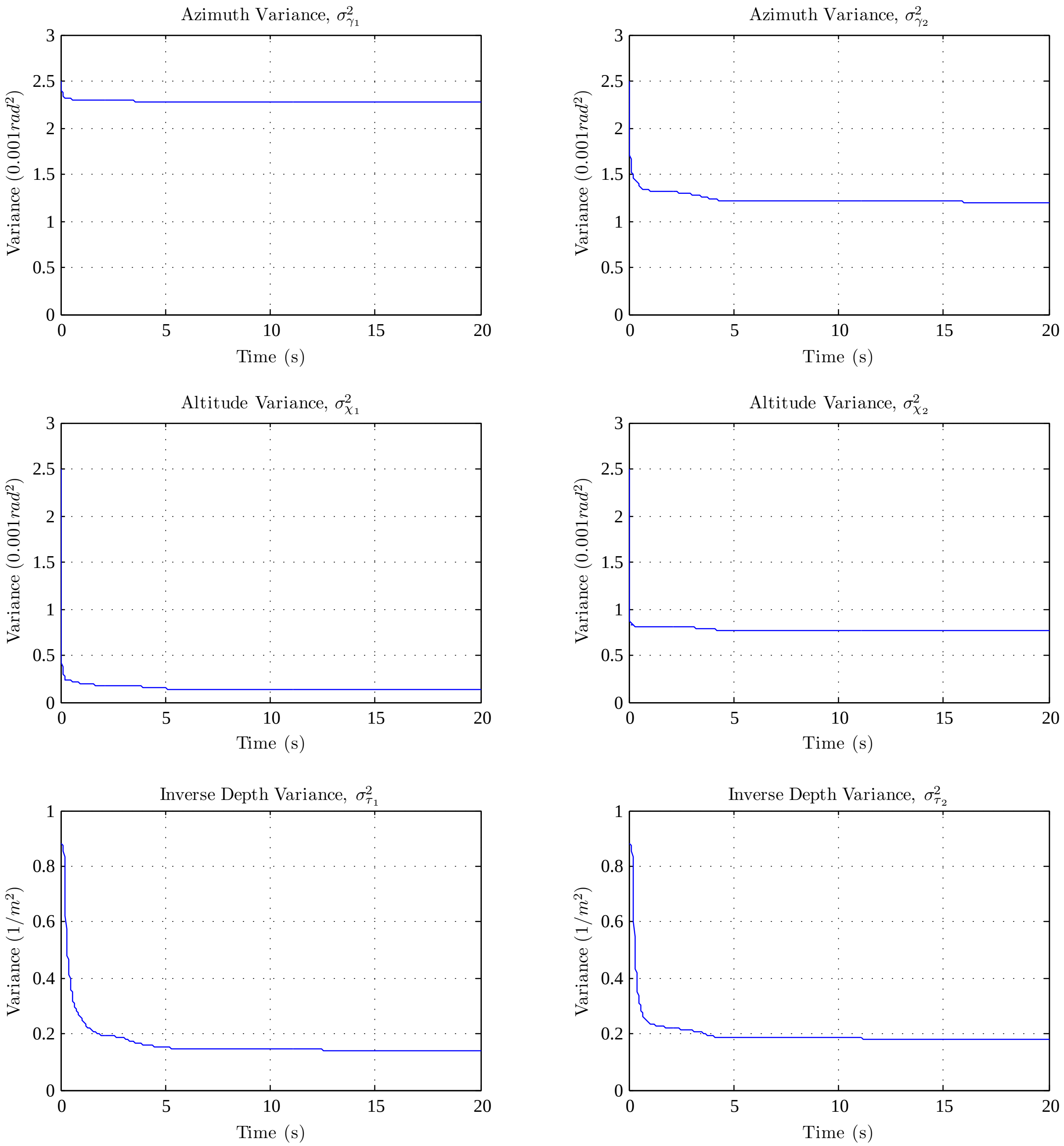


Figure 5.13: Estimate covariance of the modelling frame locations of the first (left column) and second (right column) object feature points from measurements with a single monocular camera with a completely unknown target object model.

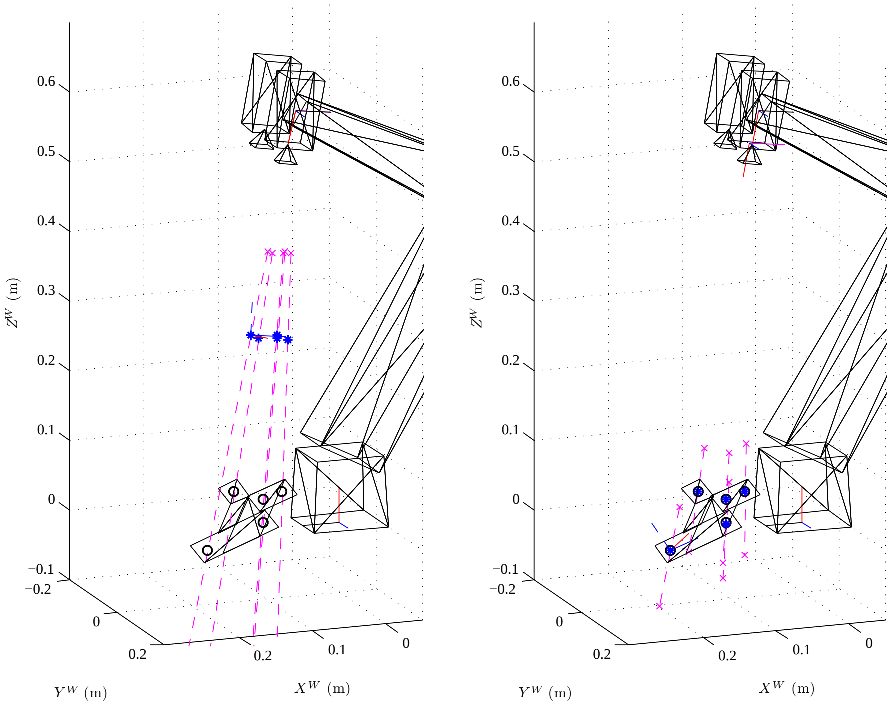


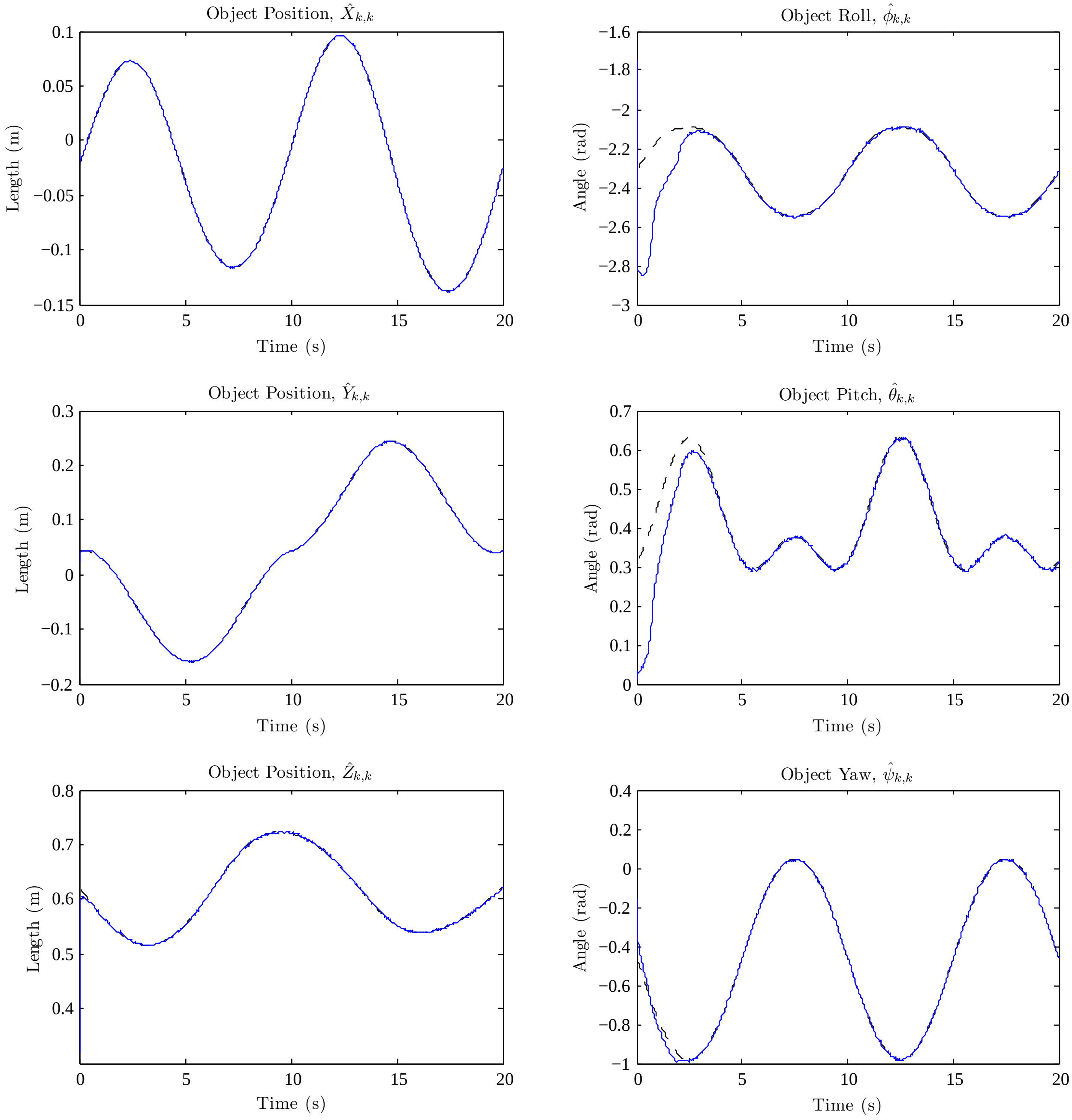
Figure 5.14: Initial (left) and final (right) state estimates for the two monocular cameras case with unknown target object model. Actual feature point locations are empty circles on target object, estimated locations are solid dots. The 2σ range of the feature point depth uncertainty is represented by the dashed lines.

performance of the camera and range finder combinations in the test cases that follow.

The initial setup for this test case and the final estimate are shown in Fig. 5.14. For the same relative motion of the target object and robot end-effector, this system generates accurate estimates of the relative pose parameters, as shown in Fig. 5.15. Initially, the object is visible in both cameras and the scale is quickly resolved. Accordingly, the relative pose estimates go towards the ground truth values.

The estimation errors for the relative pose parameters reflect this evolution, and are displayed in Fig. 5.16. Once the shape is recovered, just prior to the 5 second mark, the error drops on the position and orientation states for the rest of the trajectory.

The estimate covariances for the relative pose parameters are strongly related to the visibility of the object in both cameras. In Fig. 5.17, when both monocular cameras can see the feature points, the covariances drop sharply indicating high confidence can be placed in the resulting estimates. In fact, the covariance values



Object Position, $\hat{X}_{k,k}$

Object Roll, $\hat{\phi}_{k,k}$

Object Position, $\hat{Y}_{k,k}$

Object Pitch, $\hat{\theta}_{k,k}$

Object Position, $\hat{Z}_{k,k}$

Object Yaw, $\hat{\psi}_{k,k}$

Figure 5.15: Estimates for the six relative pose parameters over the sequence using measurements from two monocular cameras with a completely unknown target object model. Actual values are dashed lines, estimated are solid lines.

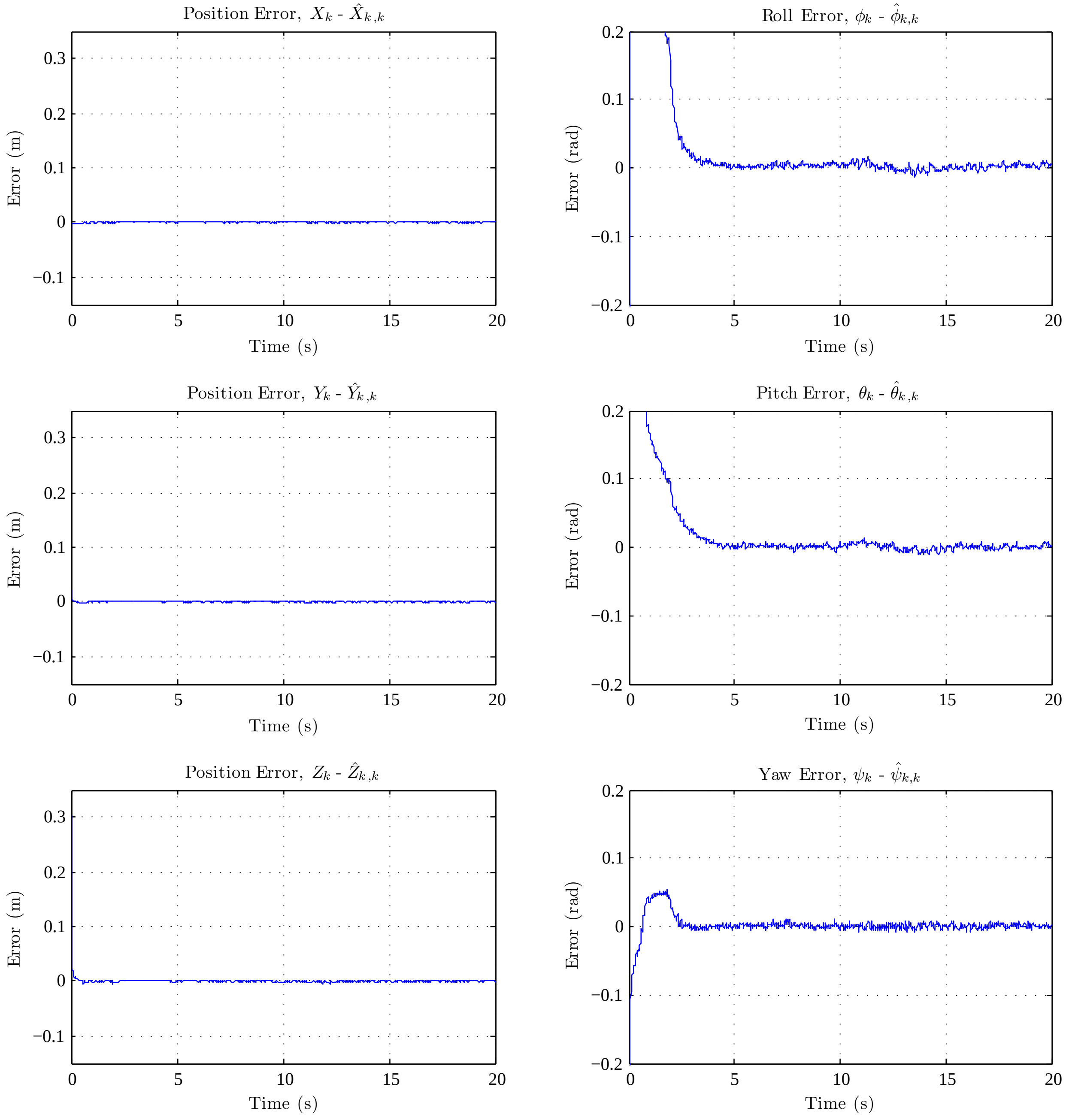


Figure 5.16: Estimation error for the six relative pose parameters through the entire sequence using measurements from two monocular cameras with a completely unknown target object model.

for the pose parameters are smaller than those in Fig. 5.6, when the target model was assumed perfectly known and the estimation proceeded with a single monocular camera.

With the measurements from the two monocular cameras, the depth of any feature point is measureable in each and every frame. As a result, the feature point parameter estimations, particularly the inverse depth, are accurate and converge to a consistent solution quickly, see Fig. 5.18. For the associated estimation errors, refer to Fig. 5.19.

Finally, the estimate covariance for the feature point parameters reflect a high and growing confidence in the parameter estimates. By effectively doubling the number of measurements available at every time step, the covariance of the feature parameters is significantly less than the same values in the case where only a single monocular camera was used to estimate both the relative pose and target object model (see Fig. 5.13).

Not surprisingly, using multiple monocular cameras to estimate both the relative position and orientation, as well as the feature point locations for a target object, results in greatly improved performance over the case when only a single camera is used. The global scale is constrained by the known baseline between the cameras, and as a result, there is only one solution such that the measurement data can be fit, in a least-squares sense, to the generated model.

While this does offer a solution to the relative pose estimation problem when the target object model is unknown, the measurement space is certainly not minimal, and the correspondance problem presents a significant challenge to make such an implementation practical. Instead, a simple range finder device will be coupled with the single monocular camera estimation scheme to provide a range measurement to the target object at each time step. With the proper measurement model, the data can be fused to form a cohesive, correctly-scaled relative pose estimate. However, certain assumptions are necessary to make such a combination possible, and it will be the suitability of these assumptions which will be verified by comparing the resulting performance with this test case.

5.3.4 Combined Estimation with Camera and Range Finder

In this test, a point range finder device is fixed in a known position and orientation with respect the the monocular camera, and supplies a single range measurement

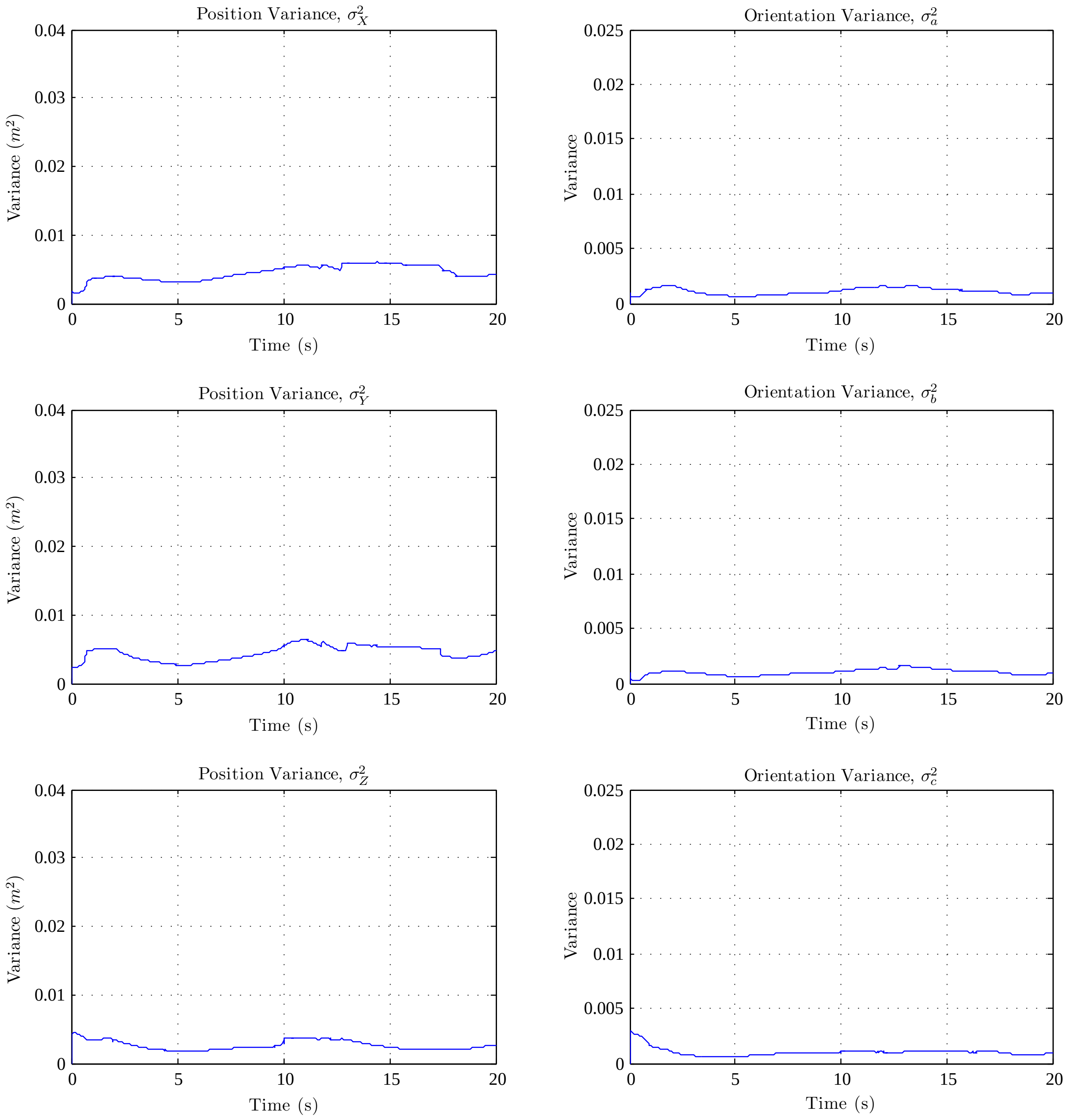


Figure 5.17: Estimate covariance for the six relative pose parameters through the entire sequence using measurements from two monocular cameras with a completely unknown target object model.

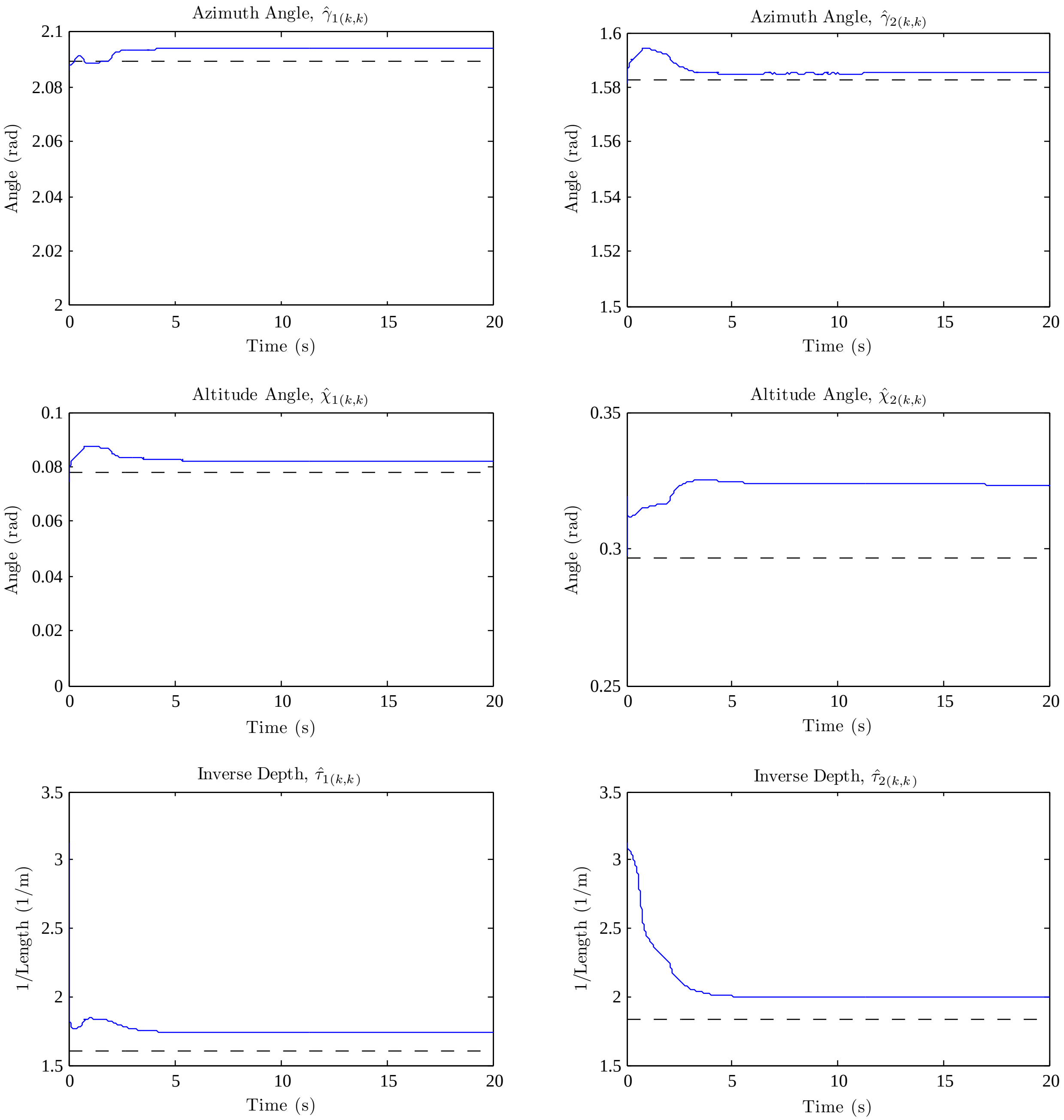


Figure 5.18: Estimates of the modelling frame locations of the first and second object feature points from measurements with two monocular cameras with a completely unknown target object model. Actual values are dashed lines, estimated are solid lines.

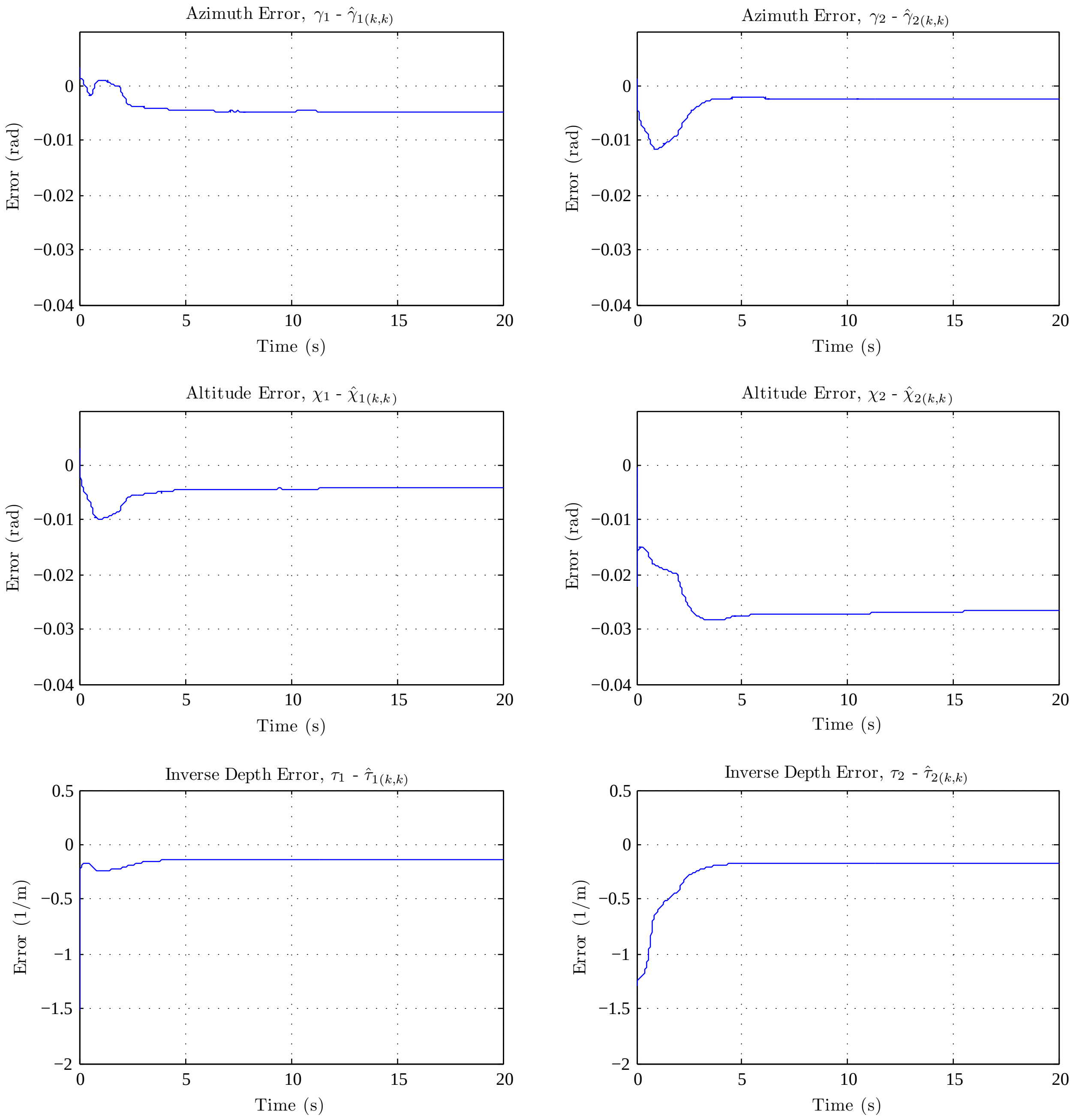


Figure 5.19: Estimation error of the modelling frame locations of the first and second object feature points from measurements with two monocular cameras with a completely unknown target object model.

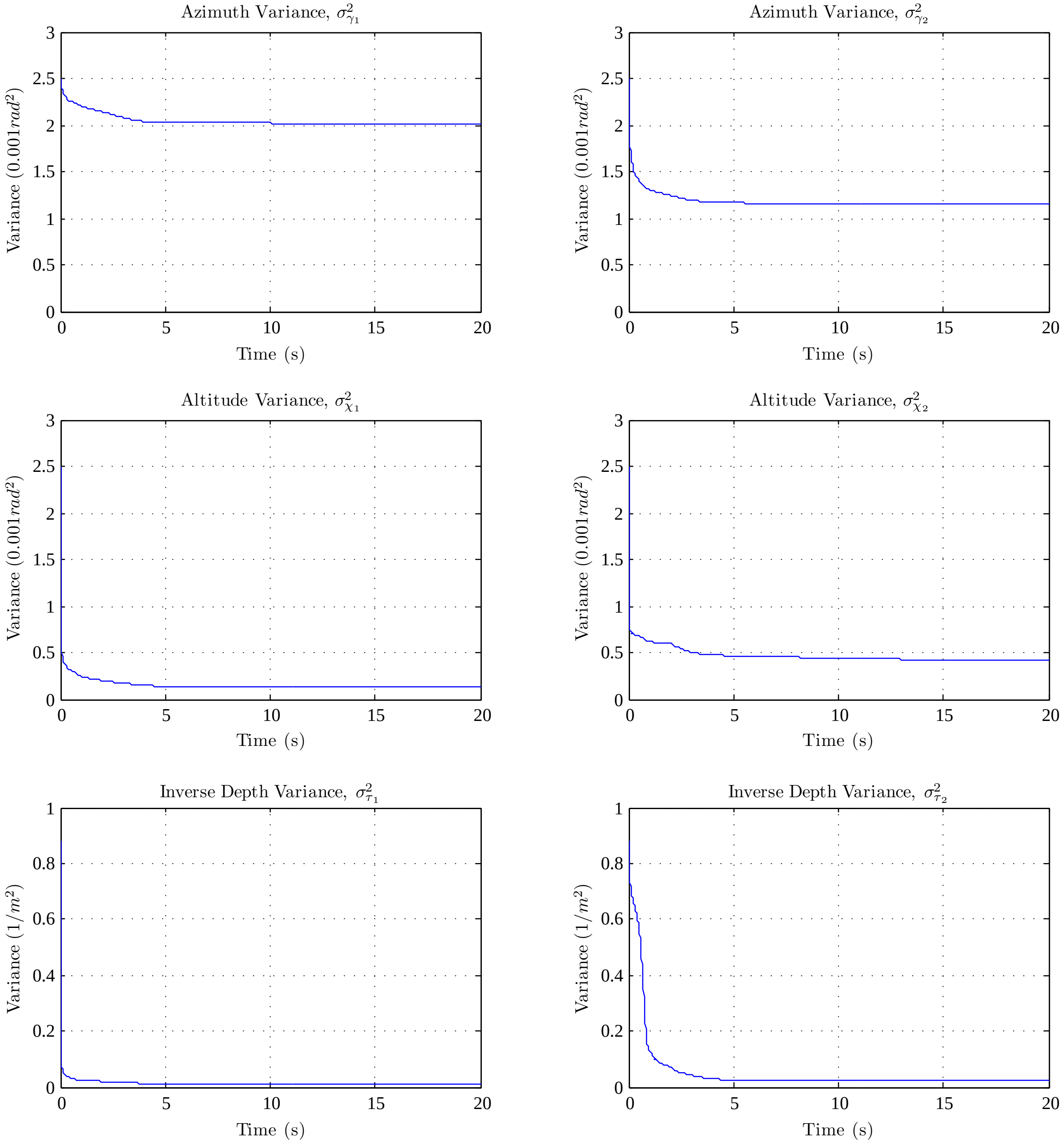


Figure 5.20: Estimate covariance of the modelling frame locations of the first and second object feature points from measurements with two monocular cameras with a completely unknown target object model.

to the point of reflection on the target object at each time step. The target object moves with the same relative motion as in the last two test cases to check whether augmenting the measurement space in this way will improve the relative pose estimates of the unknown target object.

The performance of the estimator will depend critically on the validity of the assumptions associated with the measurement model. Principally, the range finder is assumed to provide a measure of the average feature point scene depth. Since it will almost certainly not be the case that the range finder will always strike a surface at this particular depth, the measurement noise associated with the range measurement is artificially inflated above the noise from the base sensor so that the modeling error is accomodated for by (possibly biased) noise. Mathematically, increasing the measurement noise covariance amounts to de-weighting the measurement in the estimation so that is has a less significant effect relative to the other measurements available. In this case, and demonstrated by the two previous tests, more confidence can be placed in the camera measurements compared to the range finder with an approximate measurement model. The range measurements should be used sparingly to 'suggest' the correct scale instead of strongly influencing the relative pose estimates directly.

After running the estimation, the final estimate of the pose and target model is shown in Fig. 5.21. The relative pose estimates for the entire motion are shown in Fig. 5.22. When compared to the estimates from the previous test with the single monocular camera as the only measurement device, the relative pose estimation is significantly improved and it is clear the the scale of the pose is being recovered. The relative trajectory of the target object takes it in and out of the range finder's beam several times. Initially, the beam strikes the object a surface which is closer than the average depth of the feature points. This results in the scale of the solution initially being too small. Soon after, the object leaves the beam and the range measurements are not available for the estimator for this period.

This illustrates a strength of the EKF framework, by which measurements can be included or neglected when they are available and the estimation will continue with the remaining measurements. When the range measurements are lost, the current scale value is effectively *locked in* and the estimation proceeds with the vision measurements only, at that scale.

Later, the object crosses the range beam again at just before ten seconds. At this point, the scale converges further towards the proper value. In Fig. 5.23, the estimation error of the relative pose parameters indicates that the scale is wrong

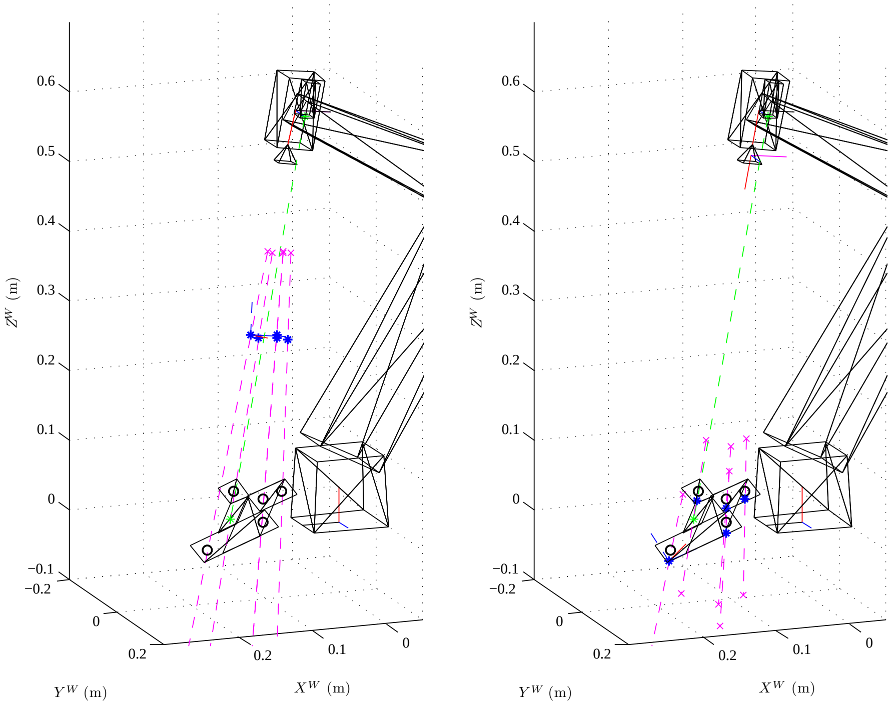


Figure 5.21: Initial (left) and final (right) state estimates for the monocular camera plus point range sensor case with unknown target object model. Actual feature point locations are empty circles on target object, estimated locations are solid dots. The 2σ range of the feature point depth uncertainty is represented by the dashed lines.

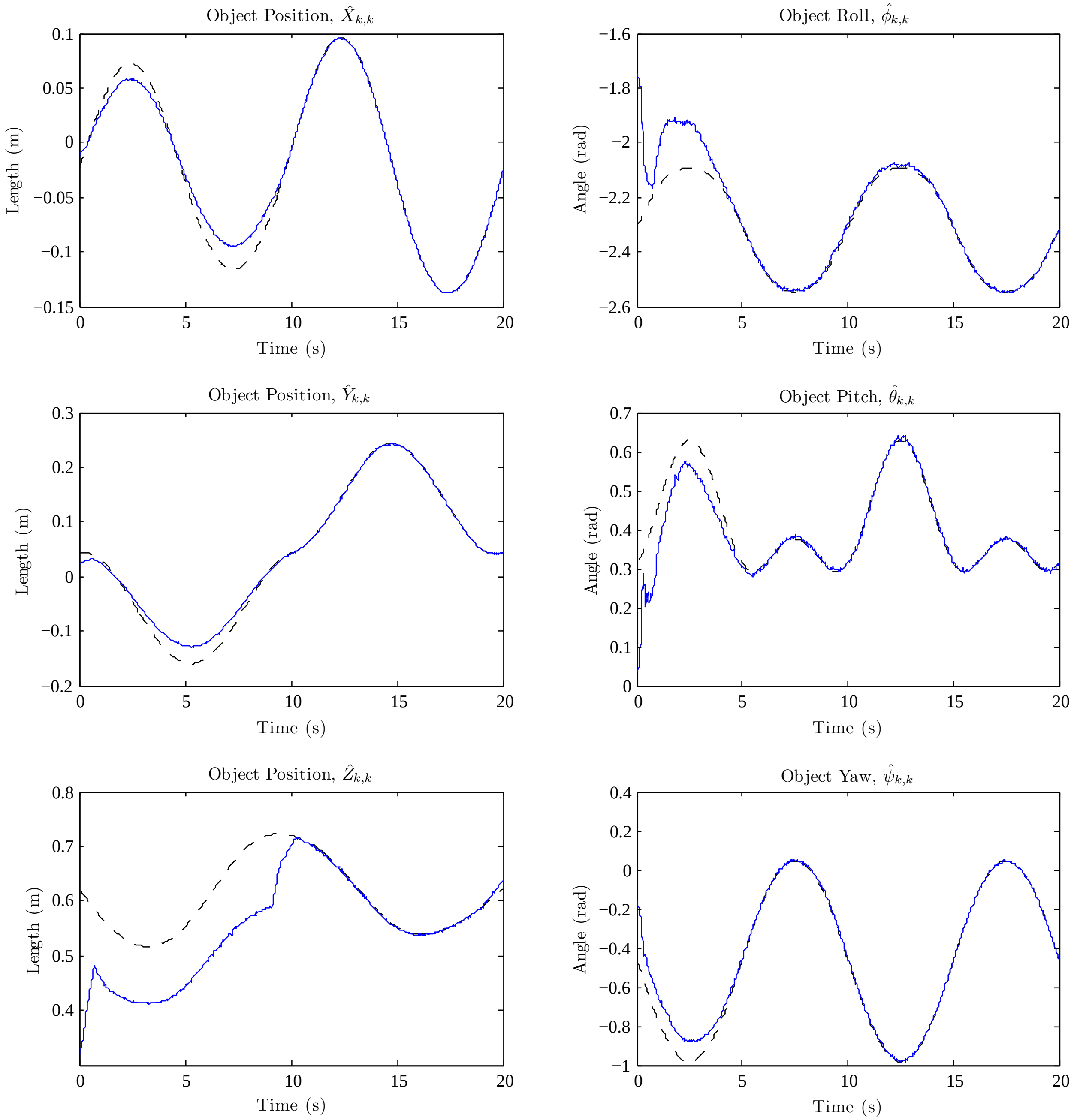


Figure 5.22: Estimates for the six relative pose parameters over the sequence using measurements from a single monocular camera and point range finder device with a completely unknown target object model. Actual values are dashed lines, estimated are solid lines.

initially, but once it converges at around ten seconds, the errors collapse to what resembles zero-mean white noise. This indicates that the previously unmodeled dynamics, in this case the model scale, are successfully extracted from the information provided by the measurements.

The estimate covariance for the relative pose parameters looks similar to that of the previous case, especially when the covariance initially increases until the shape of the target object model, and therefore the object coordinate frame, are resolved. The corresponding plots are shown in Fig. 5.24.

Given the dependance of estimating the relative pose on the estimate of the target object, one would expect that the improve performance of the pose tracking is a direct result of an improved target model estimation. Indeed, this is the case for this system, as can be seen from the estimates of the parameters for the first two features, shown in Fig. 5.25 below.

The azimuth and altitude angles for the two feature points demonstrate similar tracking performance to the monocular camera-only case. However, the estimates of the inverse depth shows a great improvement in both cases, with the values actually moving to the correct values. This is a result of the range measurements dictating the proper scale for the system. Additionally, three distinct steps can be seen in each plot representing when the object crosses the range finder's beam and range measurements become available.

Once again, estimation errors for the feature parameters are shown in Fig. 5.26. It can be seen that the error of most of the parameters is falling.

The estimate covariances of the feature parameters, in Fig. 5.27, are all monotonically decreasing, as expected. Of note, are the covariances of the inverse depth parameters, which show significant reductions during the periods corresponding to the range measurements being available. Additionally, the final covariance values are less than those from the monocular camera-only test case. The range measurements allow for the relative confidence of the parameter estimates to be greater than without them, and the covariance is smaller to indicate it.

This test shows that the performance of the relative pose estimation for an unknown target object is significantly improved when compared to estimating using measurements from a monocular camera alone. The keystone is recovering the global scale of the relative motion and the object model. This is facilitated by the range finder device, mounted on the robot end-effector, whose measurements are fused within the EKF framework using a simple measurement model.

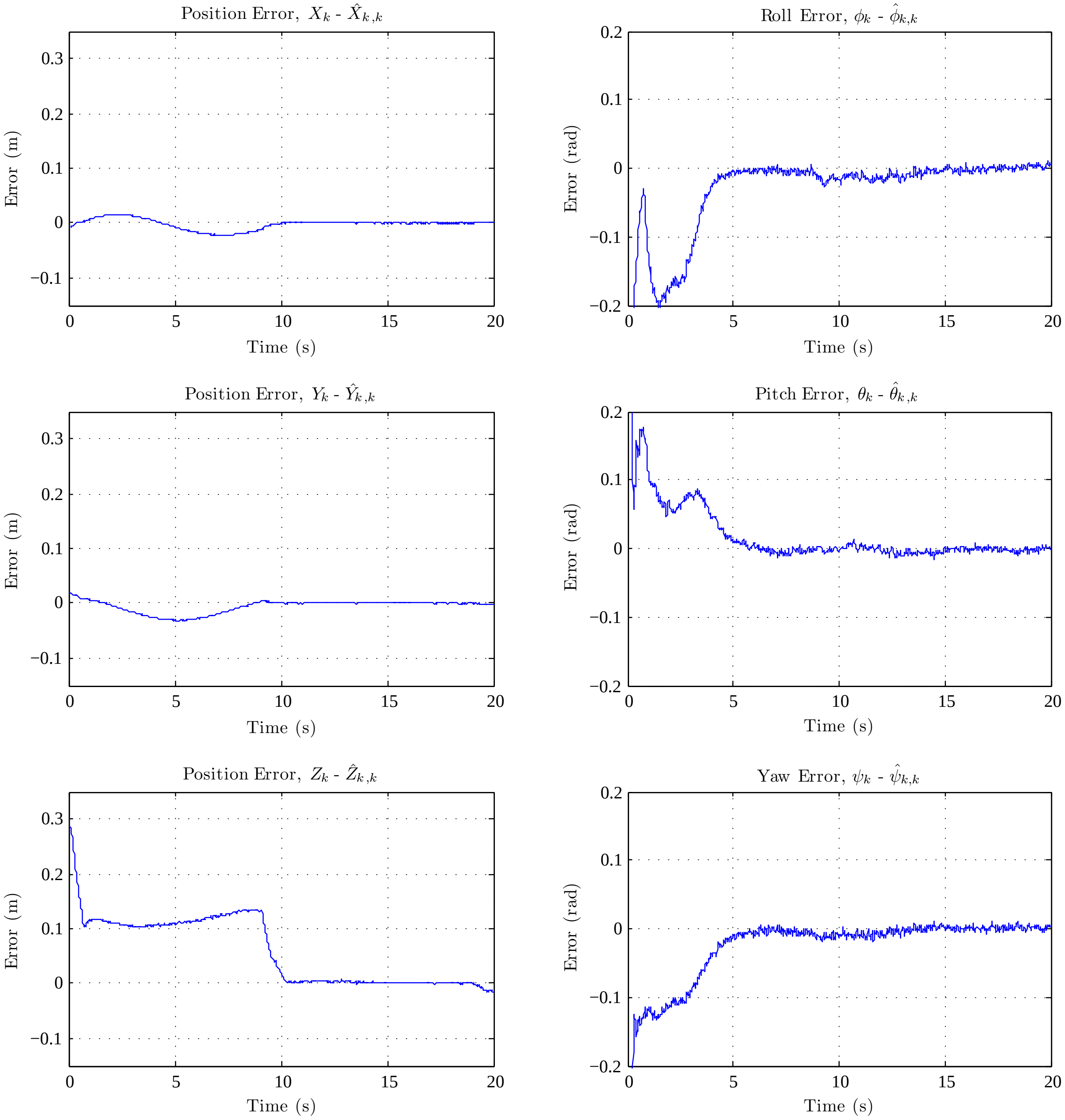
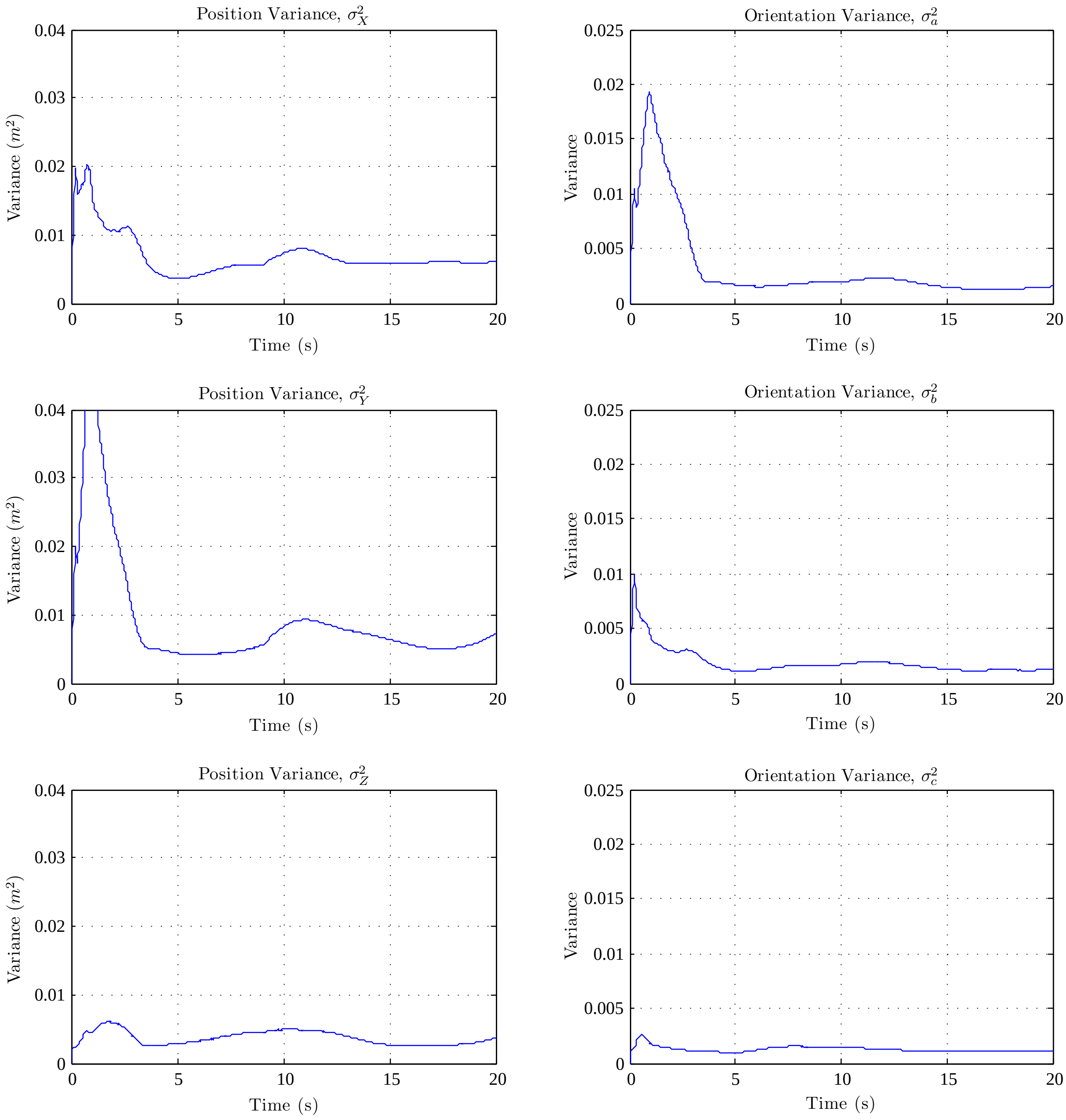


Figure 5.23: Estimation error for the six relative pose parameters through the entire sequence using measurements from a single monocular camera and point range finder device with a completely unknown target object model.


Position Variance, σ_X^2

Orientation Variance, σ_a^2

Position Variance, σ_Y^2

Orientation Variance, σ_b^2

Position Variance, σ_Z^2

Orientation Variance, σ_c^2

Figure 5.24: Estimate covariance for the six relative pose parameters through the entire sequence using measurements from a single monocular camera and point range finder device with a completely unknown target object model.

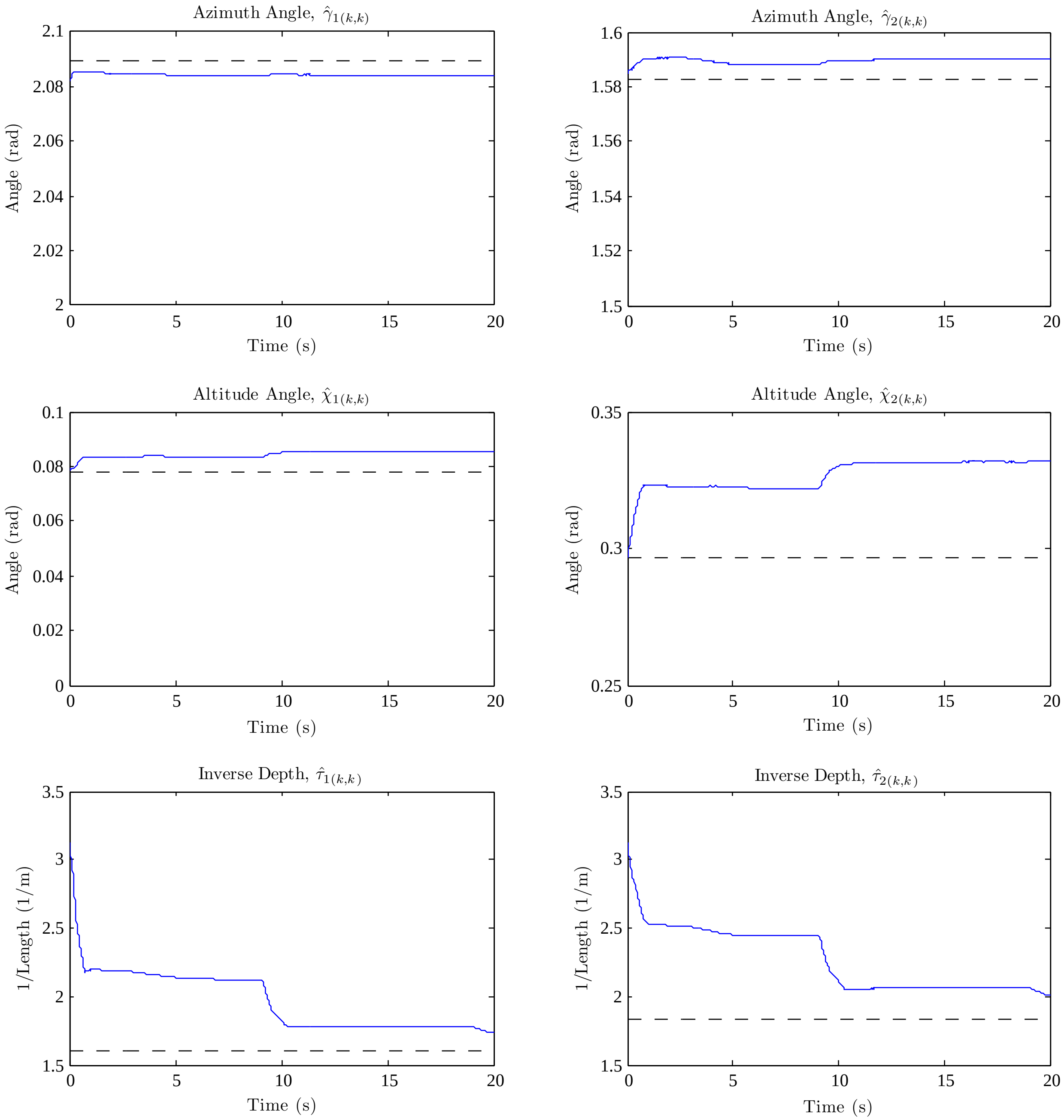


Figure 5.25: Estimates of the modelling frame locations of the first and second object feature points from measurements with a single monocular camera and point range finder device with a completely unknown target object model. Actual values are dashed lines, estimated are solid lines.

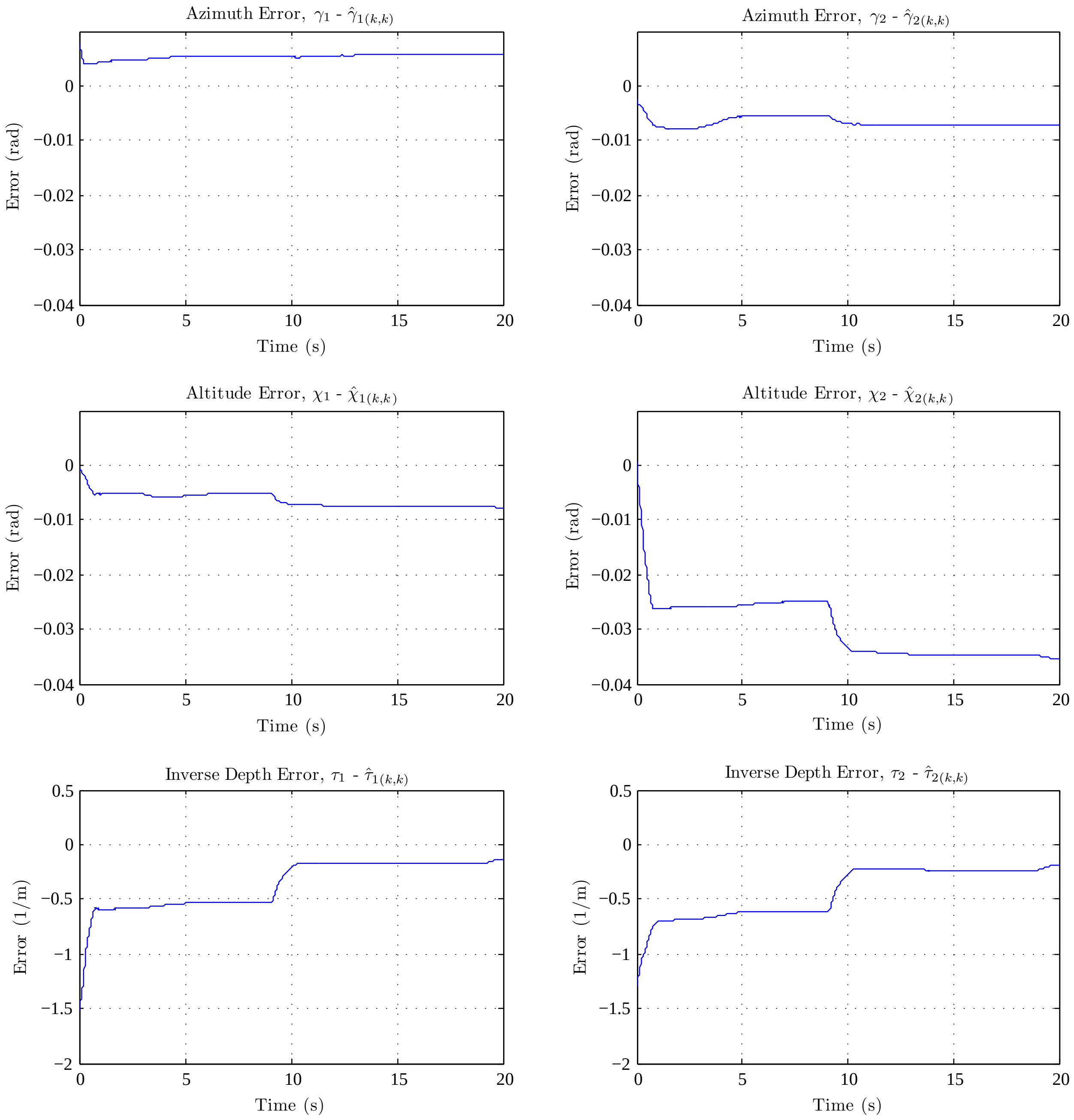


Figure 5.26: Estimation error of the modelling frame locations of the first and second object feature points from measurements with a single monocular camera and point range finder device with a completely unknown target object model.

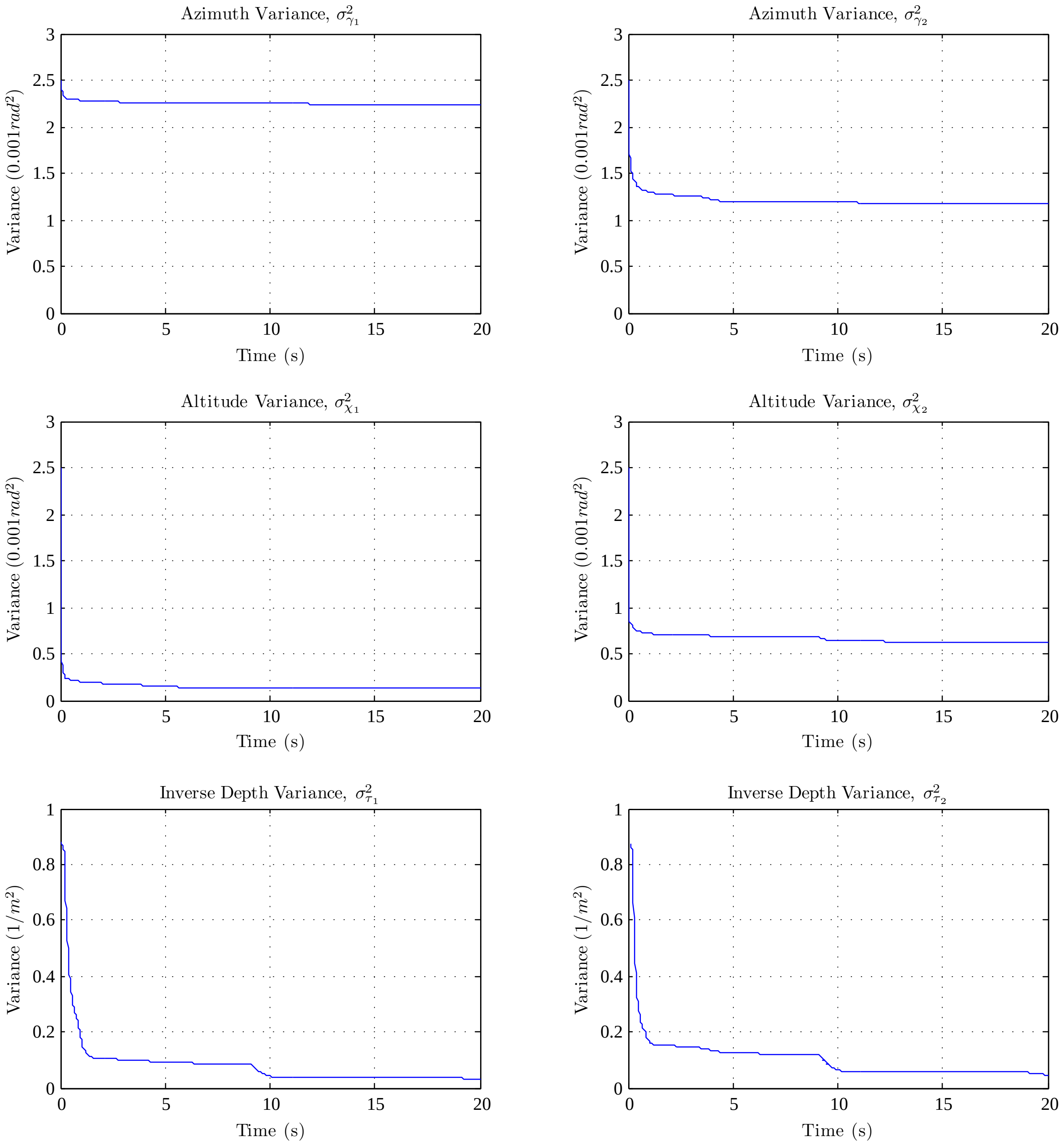


Figure 5.27: Estimate covariance of the modelling frame locations of the first and second object feature points from measurements with a single monocular camera and point range finder device with a completely unknown target object model.

This method has the disadvantage of relying on the assumption that the range beam measures the average feature point depth. This will almost never happen and approximating the modelling error as zero-mean noise may not work well if the distance measured is consistently greater or lesser than the true average. Consider an object with a convex or concave surface. If, for example, the distance measured is 90% of the true depth, the estimated scale of the environment will be 90% of the true value. For target objects where the distance to the camera is large compared to the variations in the surface depth, the solution recovered may be sufficiently accurate for certain tasks. This deficiency is important to consider for practical applications.

In the next test, a modified range finder device utilizing a laser pointer and the monocular camera with a similar measurement model, is simulated to compare the performance to the setup used in this section. If the performance is comparable, this represents an inexpensive alternative both in terms of cost and weight.

5.3.5 Combined Estimation with Camera and Laser Pointer

With the success of the previous estimation using the monocular camera supplemented by the point range finder, the current test attempts to achieve the same results with more compact and less expensive sensors. As described previously, a laser pointer is fixed at a known position and orientation with respect to the first monocular camera. The initial setup for the estimation and the final estimate are shown in Fig. 5.28.

This modified range sensor uses the same measurement model as with the point range sensor where the measured distance is assumed to be the mean feature point depth on the object. As a result, the estimation will suffer the same disadvantages concerning the unknown surface geometry assumptions.

The results of the estimation with the same relative motion profile are not as impressive as the previous system's performance. It is clear from Fig. 5.28 that the final scale of the estimate is too small. The estimated relative pose parameters though the relative motion are shown in Fig. 5.29.

When compared to ground truth, it is again apparent that the relative position parameters do not achieve the proper global scale throughout the entire motion sequence. The estimation error, shown in Fig. 5.30, shows that the estimator is unable to accurately recover any of the relative position parameters. Similar to the case when only a single monocular camera was used to estimate both the pose and

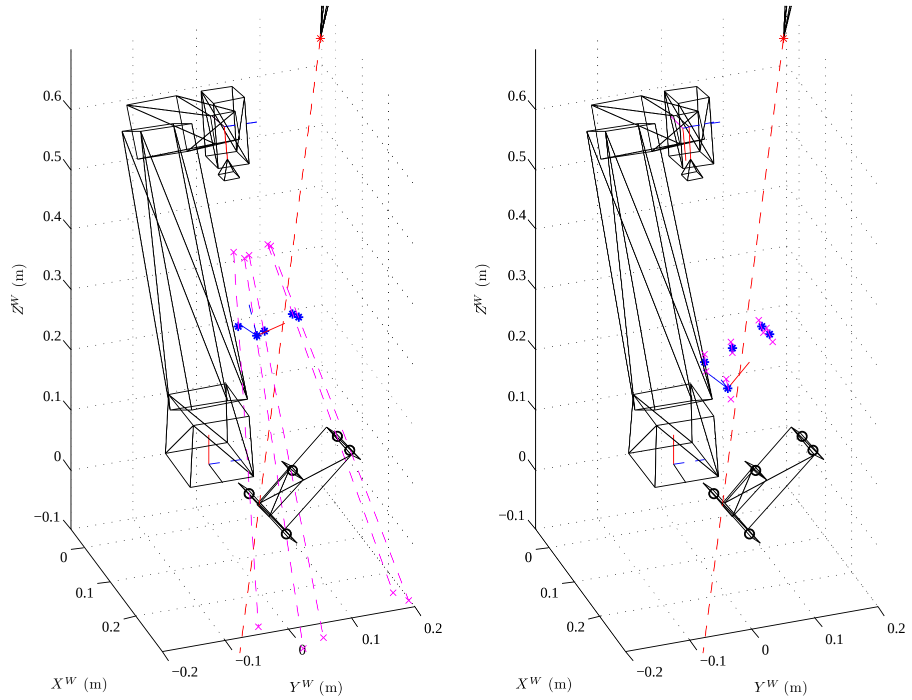


Figure 5.28: Initial (left) and final (right) state estimates for the monocular camera plus laser pointer case with unknown target object model. Actual feature point locations are empty circles on target object, estimated locations are solid dots. The 2σ range of the feature point depth uncertainty is represented by the dashed lines.

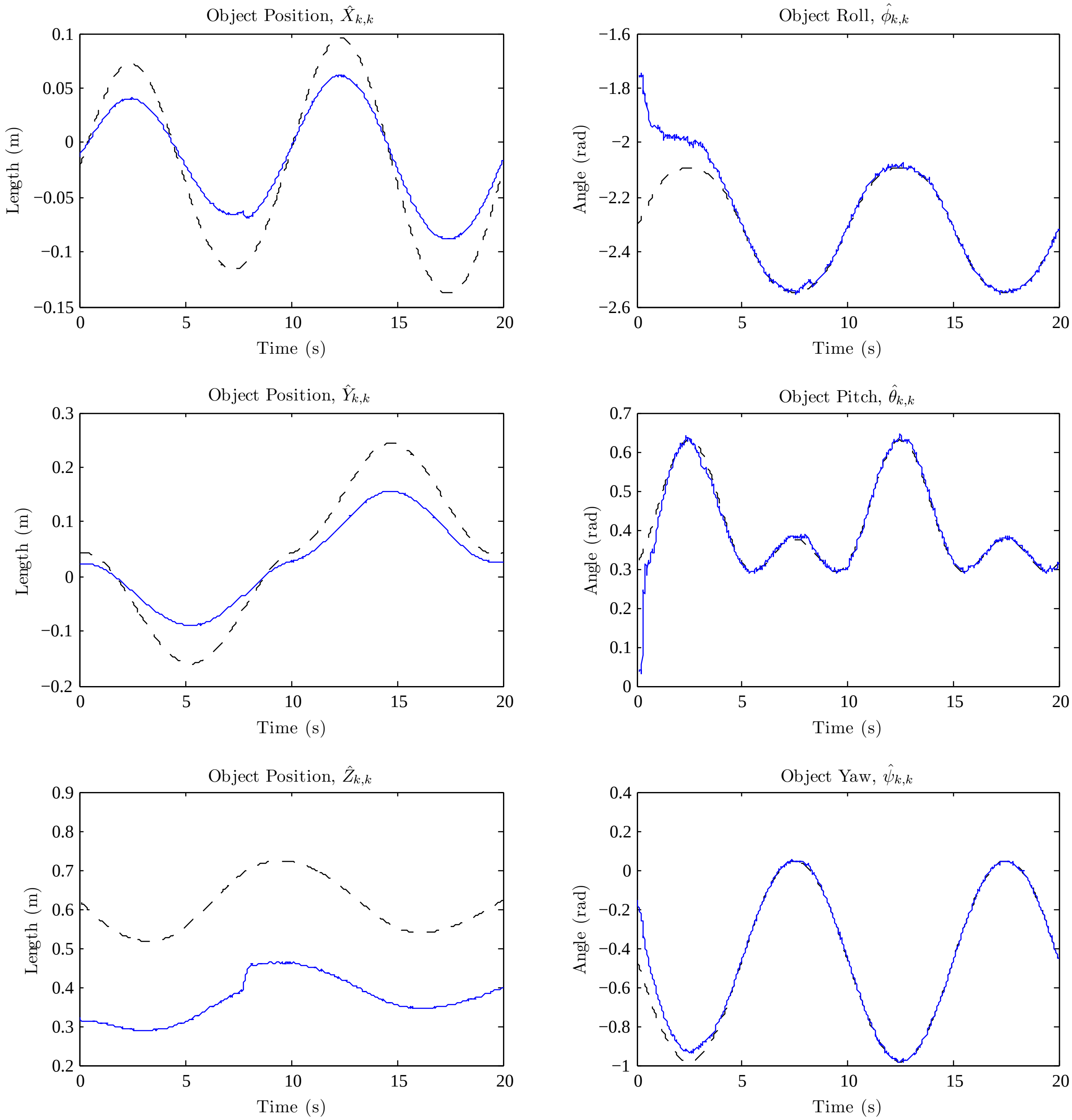


Figure 5.29: Estimates for the six relative pose parameters over the sequence using measurements from a single monocular camera and laser pointer range device with a completely unknown target object model. Actual values are dashed lines, estimated are solid lines.

model, the errors follow the trend of the estimates indicating that the recovered global scale is inaccurate.

Despite the relative pose estimates having large errors associated with the parameters, the estimate covariance for the pose parameters, in Fig. 5.31, are similar to those from the previous test case (see Fig. 5.24). This is a result of the disturbance and measurement noise processes having similar characteristics in both experiments, but slightly different state estimates used for the linearizations during the EKF recursions. Once again, this demonstrates that it is important to take care when evaluating the confidence that can be placed in an estimate from the EKF operating with a nonlinear system which is not observable at every time step (especially when the laser dot does not strike the target object).

Given the poor quality relative pose estimates, it is expected that the target object model parameters would suffer as well. Indeed, this is the case as can be seen in Fig. 5.32, showing the parameters for feature points one and two. While the azimuth and altitude angles of the observation rays perform similarly to the other test cases, the inverse depth parameters only slightly move toward their proper ground truth values.

This result is reenforced by observing the estimation error on these model parameters in Fig. 5.33.

The most important result for this estimation is seen in the covariance plots of the target object model parameter estimates in Fig. 5.34. When the laser dot strikes the target object the covariance associated with the inverse depth parameters drops significantly. This is the result of the measurement noise and sensitivity of the range measurements using the laser pointer, combined with the fact that there are two measurements at each observation (two image plane coordinates). When the uncertainty in depth is translated into image plane coordinates, the resulting covariance values are very small due to the sensitivity of the laser dot location at different measured depths. As the depth increases, the motion of the laser dot in the image plane becomes less significant. As a result, small differences in the laser dot location can mean a change in the depth measurement of several metres or more. Conversely, at short distances, the measurement is much less sensitive to changes in depth.

When the laser dot strikes the target object and a range measurement becomes available, the associated covariance found using equations (4.58) and (4.59) is small and reduces the estimate covariance accordingly. This apparent 'high confidence' leads the estimate of the inverse depth to converge quickly, even though the relative

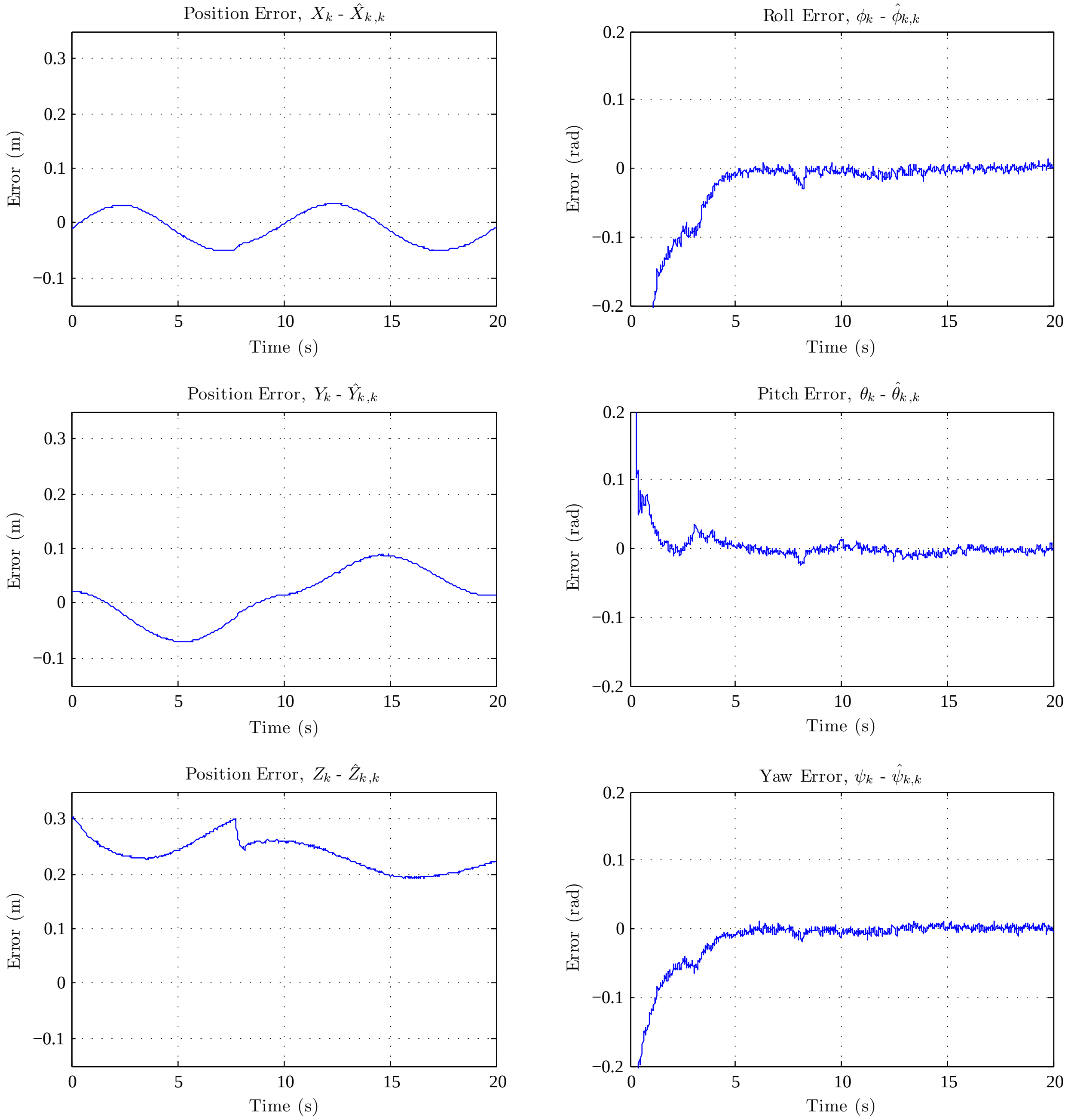


Figure 5.30: Estimation error for the six relative pose parameters through the entire sequence using measurements from a single monocular camera and laser pointer range device with a completely unknown target object model.

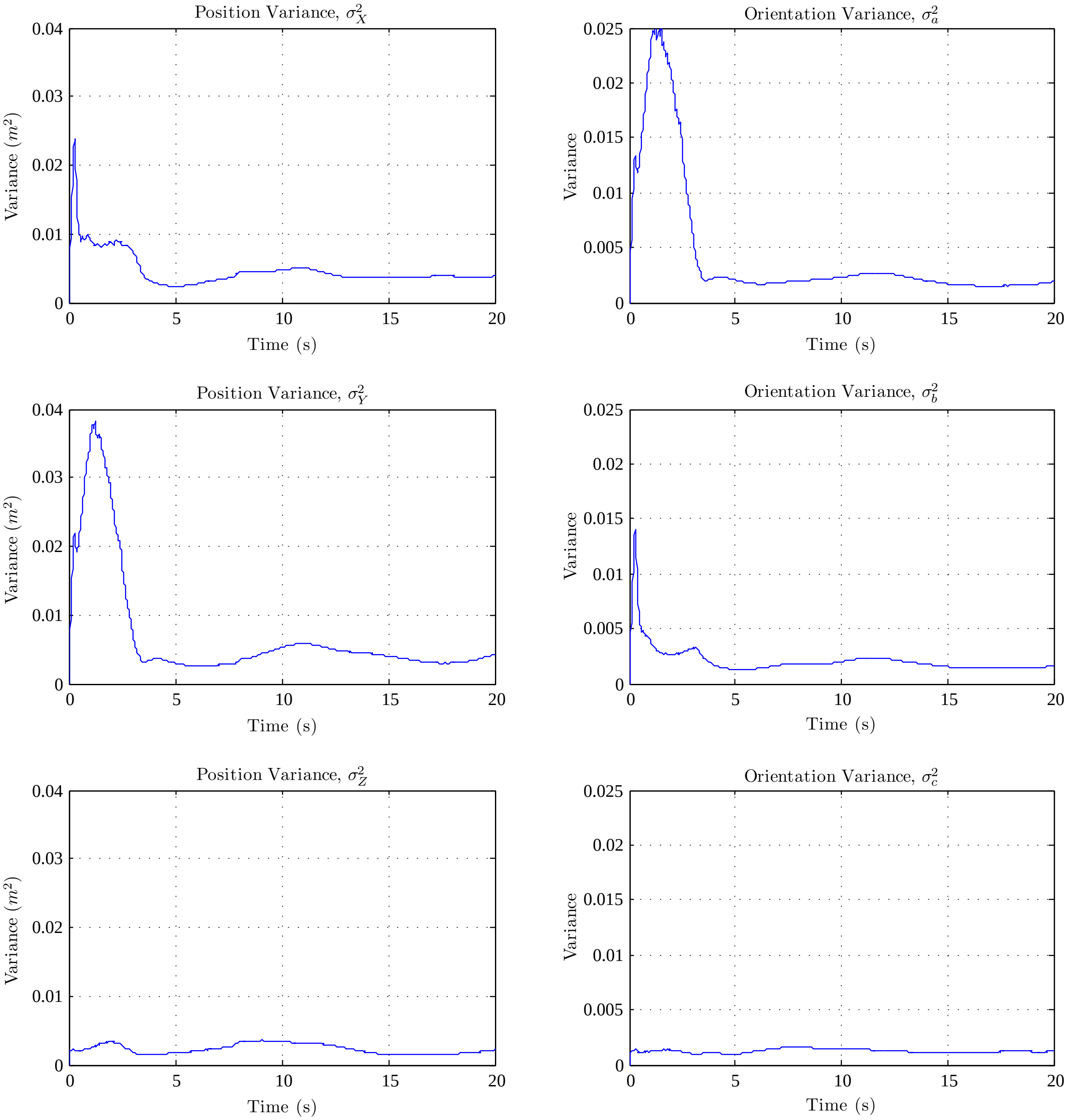


Figure 5.31: Estimate covariance for the six relative pose parameters through the entire sequence using measurements from a single monocular camera and laser pointer range device with a completely unknown target object model.

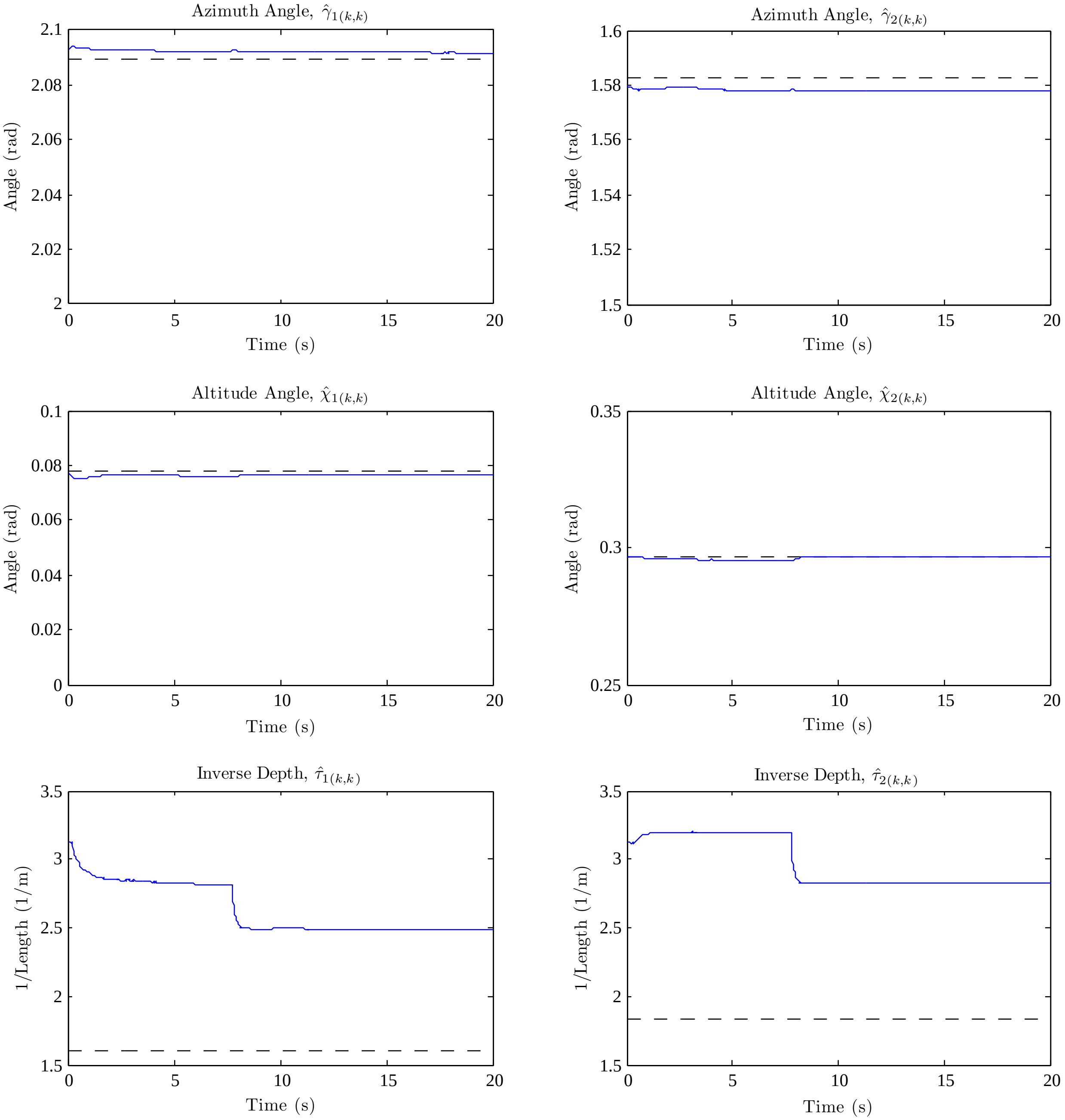


Figure 5.32: Estimates of the modelling frame locations of the first and second object feature points from measurements with a single monocular camera and laser pointer range device with a completely unknown target object model. Actual values are dashed lines, estimated are solid lines.

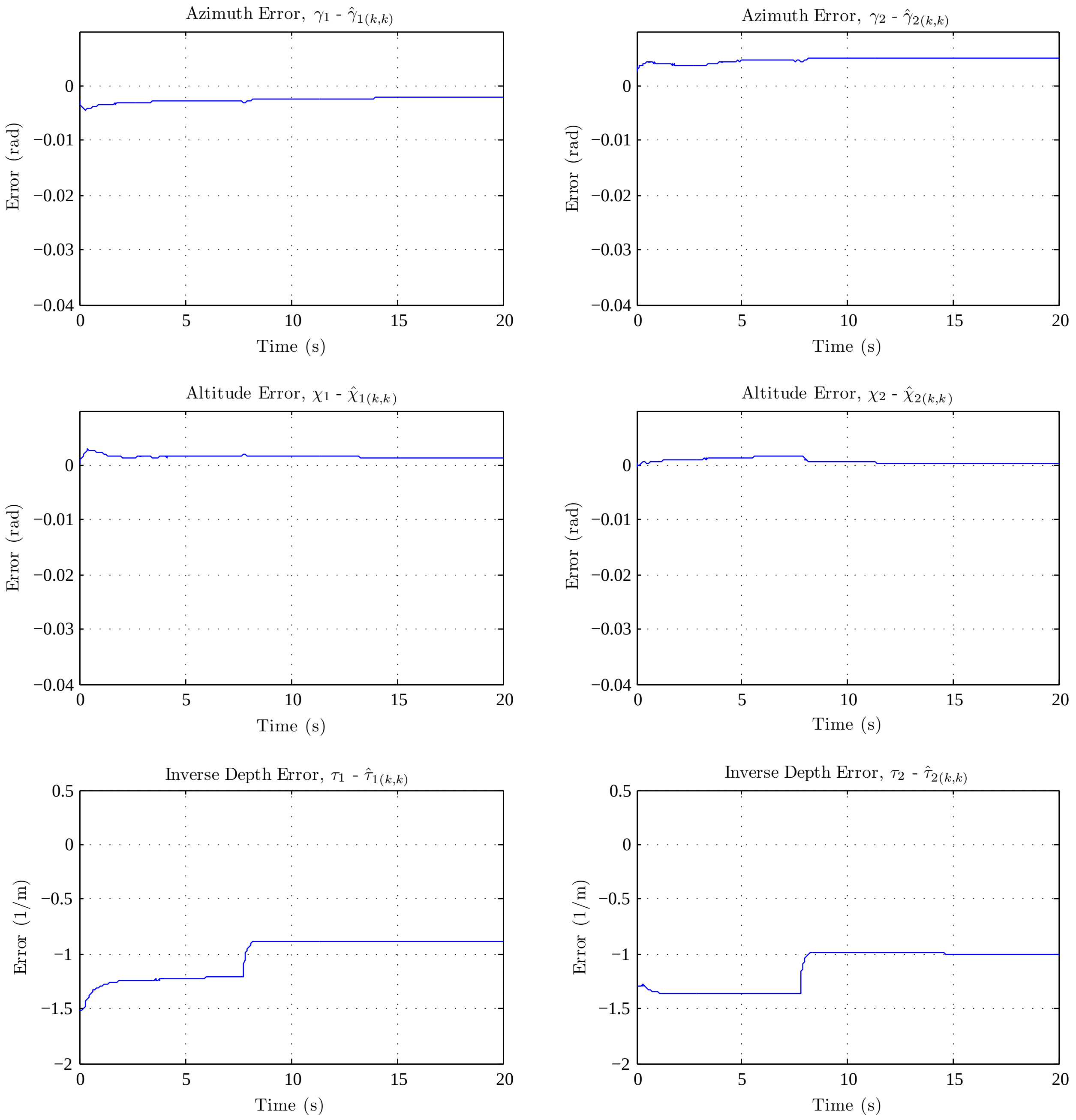


Figure 5.33: Estimation error of the modelling frame locations of the first and second object feature points from measurements with a single monocular camera and laser pointer range device with a completely unknown target object model.

depth uncertainty required to use the mean depth model is large. Projecting this large depth uncertainty in the $\mathbb{R}^3$ to $\mathbb{R}^2$ results in a small measurement uncertainty especially over long distances. Once these values have converged, the rest of the parameters will be constrained to operate at that scale. In this case, the inverse depth estimates are too large, resulting in a estimate of the entire system which is too small.

The poor performance of this estimation is caused, in part, by the sensitivity of the image plane measurements of the laser dot to changes in range. Unfortunately, there is little that can be done to improve this result. It may be possible to improve the results by increasing the baseline between the camera and the laser pointer to reduce the sensitivity at longer distances, but the complexities will persist and physically mounting the pointer that far away from the camera may be difficult and clumsy. Multiple laser pointers mounted at different relative locations may provide the redundancy required to overcome the sensitivity issues, but will not be considered further in this report.

Accordingly, the laser pointer mounted on the end-effector makes a poor range measurement device. The point range finder from the previous test case is superior since the range uncertainty can more naturally be represented in the Cartesian frame, instead of a projection onto the image plane.

5.4 Summary

In this chapter, the estimation of the relative pose of a target object with respect to the robot end-effector using different combination of relative sensors was evaluated in simulation. A complete robotic manipulator and workpiece were constructed in virtual environment and designed to properly interact with a set of sensors.

The first experiment showed when an accurate prior model of the target model is available, the relative pose estimation is accurate when using only a single monocular camera for all the measurements of target object feature points.

Next, the combined estimation of both the relative pose and the target model was run using a single monocular camera. The results demonstrated that while the recovered solution was consistent, the global scale was incorrect. This means that using a single monocular camera to estimate the relative pose of a completely unknown target object is impractical without some prior scale information.

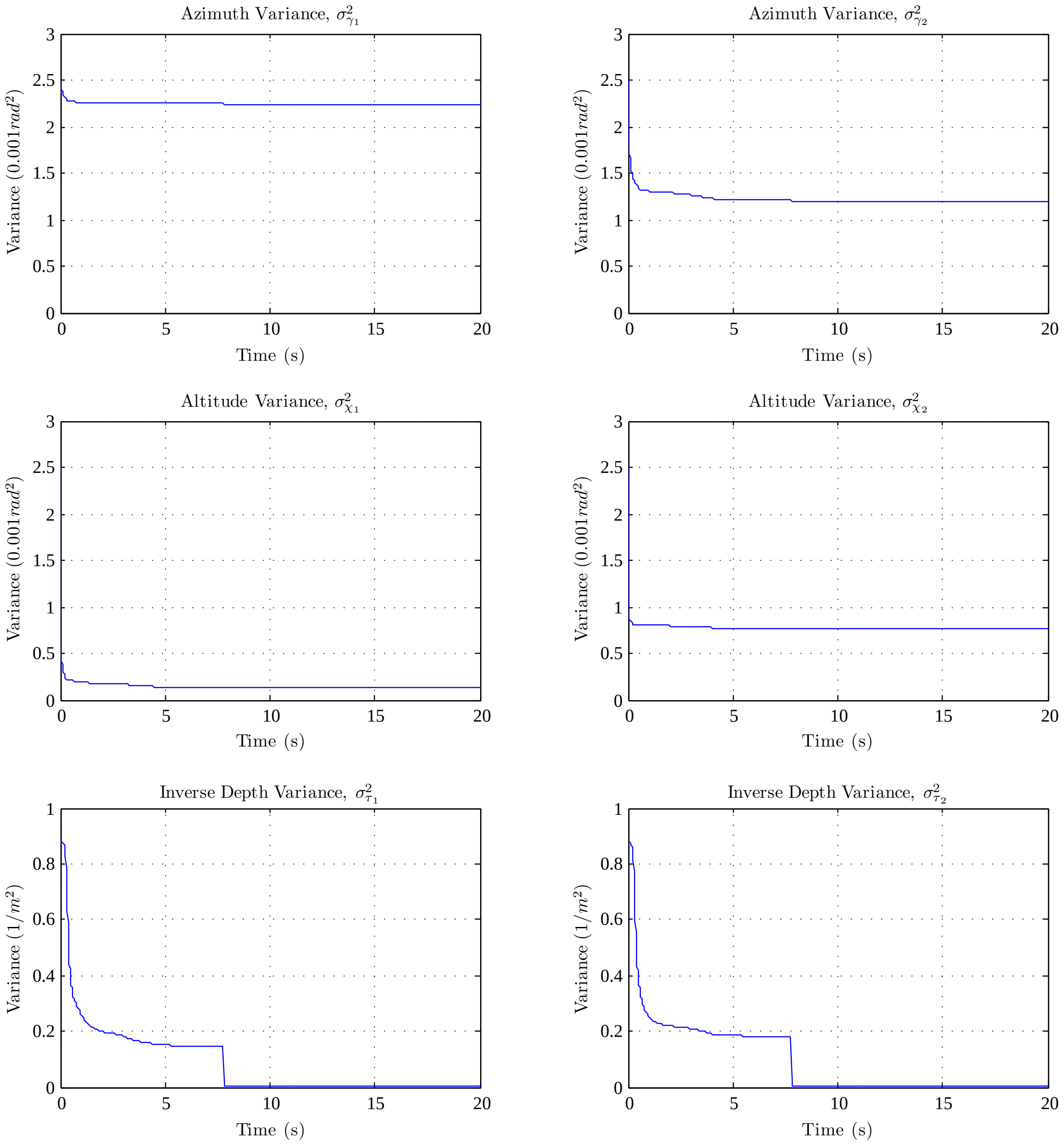


Figure 5.34: Estimate covariance of the modelling frame locations of the first and second object feature points from measurements with a single monocular camera and laser pointer range device with a completely unknown target object model.

The third experiment added a second monocular camera to the robot manipulator and it was shown that even when the target object model was initially completely unknown, the estimation error of the relative pose parameters is smaller than with a single camera and a known object model.

In an effort to utilize a smaller number of supplemental measurements, a single point range sensor was fixed to the robot manipulator alongside the single monocular camera. Good results were obtained, at the proper scale, when the measurement noise was inflated to accomodate for the modelling error of the sensor and target object surface. While it is outperformed by the dual camera system, this experiment represents a simpler, more economical solution to the problem of recovering the global scale with a single monocular camera.

The final test case replaced the point range sensor with a laser pointer fixed at a pose with respect to the monocular camera. It was found that the sensitivity of the image plane measurements and the complexities of representing the uncertainty in depth lead to a poor estimation of the target object relative pose and model parameters. While the equipment required is simpler, there are still several issues to resolve before this configuration can be reliably deployed.

Chapter 6

Conclusions and Future Work

This chapter reviews the important results from this thesis and then makes recommendations for future work and improvements to the systems proposed herein.

6.1 Conclusions

This thesis investigated the design and verification of a combined relative pose and target model estimation system for a position-based visual servoing controller.

The classical PBVS relative pose estimation scheme was outlined in which a single monocular camera mounted on the robot end-effector is used to measure the relative position and orientation of a target object with a known geometric model. An observability analysis of the resulting state space system revealed the geometric configurations of the camera and target object from which the image plane measurements cannot uniquely determine the relative pose.

Then, the estimation problem was expanded to measure the relative pose of the target object with respect to the robot end-effector when the locations of a set of target object point features are initially unknown. The resulting system treats these models parameters as augmented system states and estimates both the relative pose and target object model simultaneously. This procedure is very similar to the Simultaneous Localization and Mapping (SLAM) problem from mobile robotics literature. After recognizing this link, recent advances in the mobile robot literature are translated into the relative pose estimation framework, facilitating more effective combined estimation. Another observability analysis found a subset of the unobservable configurations of the system over multiple time steps, highlighting situations where the target model and relative pose cannot be recovered for a variety of different dimensional systems. More significantly, it was confirmed that information about the global scale of the environment cannot be recovered using on the point feature image plane measurements from a monocular camera. This indicated that supplemental relative measurements were needed to accurately recover the scale of the resulting estimation.

The specific sensors to provide the additional relative measurements were the subject of Chapter 4 and included additional monocular cameras, and two point range finding devices capable of recovering a single distance measurement to a point on the target object. A simple measurement model to facilitate augmenting the measurement space and relate the supplemental range data with the feature points visible in the monocular camera was formulated. Subsequently, a system capable of fusing the additional camera and range measurements with those from the original monocular camera was then detailed.

To test the performance of the proposed estimator with differenct combinations of additional sensors, a full set of simulation experiments are carried out with a virtual robot manipulator and target object. Five different test cases were considered for comparison. First, the classic relative pose estimation with a monocular camera and a known target object model was run to provide a baseline for the relative pose estimates assuming perfect knowledge of the target object model. Next, the combined pose and model estimation using a single camera is executed and it was demonstrated that the recovered solution, while consistent, only recovers a scaled version of the environment. Third, adding a second monocular camera to the robot end-effector was shown to improve the resulting estimate of the pose and model and lead to the recovery of an accurate global scale. Then, the combination of a monocular camera and a point laser range finder was run and found to be able to estimate the relative pose and model of the target object with reasonable accuracy. By artificially increasing the noise covariance associated with the range measurement, the modelling error can be accomodated for and the global scale of the solution can be recovered provided the measurement model assumptions are not significantly violated. Finally, the combination of a monocular camera and a laser pointer mounted at a fixed relative position was implemented to allow range measurements to be made corresponding to the laser dot location in the camera image plane. However, it was observed that this device is sensitive to noise and does not work well within the EKF framework. The recovered estimates were not correct to the proper scale.

6.2 Future Work

From the work presented in the previous chapters, it is clear that the combined relative pose and target model estimation requires more measurements than are available from only a single monocular camera. The supplemental measurements, which are fused with the original camera image plane measurements in the system presented here, provide the extra information needed to recover the correct scale of the solution. While the simulated test cases suggest that the camera and range finder combination is capable of accurately recovering this scale, the implementation of the estimator using a physical robotic system is an important next step to fully validate these theoretical and simulation results.

There are also some improvements to the range finder measurement model which may lead to reduced modelling error in the distance measurements to the target object surface. The difficulty, as noted previously, was in associating the feature point parameters with the target object surface geometry. Assuming that the laser beam strikes the target object at the mean feature point depth may not work in all cases, especially for convex or concave objects with the feature points on the target object surface. For these two object shapes, the scale of the estimate is likely to be too small and too large, respectively. It may be more effective to weight the feature points used to calculate the expected depth based on relative orientation and target object shape, which is observable using only the monocular camera measurements. Further, it may be possible to actively point the range beam at a feature point directly using the shape information.

Another possibilty for improving the range information would be to add additional range sensors to the robot end-effector. This would provide a better average of the mean feature point depth. Instead of relying on only one range measurement per time step, the multiple range measurements should relieve the effect of modelling error on the resulting estimate.

In the simulation experiments, the combination of the monocular camera and the laser pointer to provide measurements of the range to the target object, performed poorly. A more in-depth investigation into the sensitivity of the measurements and ways to improve the resulting system is required.

On the theoretical side of this work, some of the observability results can be extended to fully understand all of the configurations in which the estimators will struggle or fail to recover the correct solution for the relative pose and target object model. The case where the target object model is unknown represents a significant opportunity. Being able to fully quantify the relative configurations of the camera and target object, for which the pose and model are not recoverable, is essential in order to make assertions about the robustness and reliability of any robotic system based on this technology.

Finally, the most promising estimation results from the simulated test cases occured when the robot end-effector was fitted with two monocular cameras. When the baseline between multiple monocular cameras is known accurately from extrinsic calibration, the scale of the solution is recoverable for the combined estimation. Furthermore, the fields of views of the multiple cameras do not necessarily need to overlap, as with traditional stereo cameras. Accordingly, a set of monocular cameras can be implemented to observe point features of the target object in a potentially unlimited combined field of view, and provide independent measurements to a unifying estimation framework. If a suitably robust feature extraction scheme can be implemented in which the location of a feature point across multiple camera image planes can be searched, the pose of a robot manipulator in or around relatively large target objects could be estimated effectively. These results would also be useful when localizing a mobile robot within a map of landmarks which is built concurrently, as in a SLAM process.

While the theoretical results and proposed estimation systems in this thesis increase the applicability of the PBVS control scheme, the possiblities for future research which result are both promising and exciting. It is hoped that this work will lead to significant advances in producing truly capable autonomous robotic systems.

APPENDICES

Appendix A

System Jacobians

In this appendix, *MATLAB* code is provided to determine the symbolic Jacobians for the nonlinear process and measurement models presented in this thesis.

A.1 Utility Functions

A.1.1 rotationX

```
function R = rotationX( angle )

R = [ ...
        1   0           0 ;
        0   cos(angle)  -sin(angle) ;
        0   sin(angle) cos(angle) ;
        ];

end
```

A.1.2 rotationY

```
function R = rotationY( angle )

R = [ ...
        cos(angle)  0   sin(angle) ;
        0           1   0 ;
        -sin(angle)  0   cos(angle) ;
        ];

end
```

A.1.3 rotationZ

```
function R = rotationZ( angle )

R = [ ...
        cos(angle)  -sin(angle)  0 ;
        sin(angle) cos(angle)  0 ;
        0           0           1 ;
        ];

end
```

A.1.4 mrp2R

```
% A function to find the rotation matrix for a set of
% Modified Rodrigues Parameters
function R = mrp2R( a, b, c )

p = [ a b c ]';
S = [ ...
        0 -p(3) p(2) ;
        p(3) 0 -p(1) ;
        -p(2) p(1) 0 ;
        ];

R1 = eye(3);
R2 = ( 4*( 1 - p'*p )/( ( 1 + p'*p )^2 ) * S );
R3 = 8/(( 1 + p'*p )^2) * S*S;

R = R1 - R2 + R3;

end
```

A.2 KOM Camera Measurement Jacobian

```
% Symbolic Measurement Jacobian for a Pin-hole Camera Model
% - Relative Orientation in Euler angles (phi, theta, psi)

% By Michael Tribou -- August 2009

clear;

% System States as Symbols
% Relative Pose Parameters (Object Frame wrt Camera Frame)
syms X Y Z ph th ps real;
% Feature Point Parameters (in Object Frame)
syms xo yo zo real;

% Monocular Camera Model Parameters
% Focal Length in X^I and Y^I Directions
syms Fx Fy real;

% Transformation from Modelling to Camera Frame
Tc_m = [ ...
        rotationZ(ps)*rotationY(th)*rotationX(ph) [X Y Z]' ;
        0 0 0 1 ];

% Object Feature Point in the Object Frame
obj = [ xo yo zo ]';

% Transform the Object Feature Point into the Camera Frame
cam = simplify( Tc_m*[ obj ; 1 ] );

xc = cam(1);
yc = cam(2);
zc = cam(3);

% Project the Feature Point onto the Camera Image Plane
xi = -Fx * xc/zc;
yi = -Fy * yc/zc;
```

```
% Find the Partial Derivatives of the Image Plane Coordinates wrt the
% System States.
dxidX = diff(xi,X);
dxidY = diff(xi,Y);
dxidZ = diff(xi,Z);
dxidphi = diff(xi,ph);
dxidtheta = diff(xi,th);
dxidpsi = diff(xi,ps);

dyidX = diff(yi,X);
dyidY = diff(yi,Y);
dyidZ = diff(yi,Z);
dyidphi = diff(yi,ph);
dyidtheta = diff(yi,th);
dyidpsi = diff(yi,ps);

% Now have a symbolic representation of the Measurement Jacobian rows for
% this feature point
Cj = [      dxidX dxidY dxidZ dxidphi dxidtheta dxidpsi zeros(1,6) ;
            dyidX dyidY dyidZ dyidphi dyidtheta dyidpsi zeros(1,6) ];
```

A.3 UOM Camera Measurement Jacobian

```
% Symbolic Measurement Jacobian for a Pin-hole Camera Model
% - Relative Orientation in Euler angles (phi, theta, psi)
% - Feature Points in Cartesian Parameterization

% By Michael Tribou -- August 2009

clear;

% System States as Symbols
% Relative Pose Parameters (Object Frame wrt Camera Frame)
syms X Y Z ph th ps real;
% Feature Point Parameters (in Object Frame)
syms xo yo zo real;
```

```
% Monocular Camera Model Parameters
% Focal Length in X^I and Y^I Directions
syms Fx Fy real;

% Transformation from Modelling to Camera Frame
Tc_m = [ ...
        rotationZ(ps)*rotationY(th)*rotationX(ph) [X Y Z]' ;
        0 0 0 1 ];

% Object Feature Point in the Object Frame
obj = [ xo yo zo ]';

% Transform the Object Feature Point into the Camera Frame
cam = simplify( Tc_m*[ obj ; 1 ] );

xc = cam(1);
yc = cam(2);
zc = cam(3);

% Project the Feature Point onto the Camera Image Plane
xi = -Fx * xc/zc;
yi = -Fy * yc/zc;

% Find the Partial Derivatives of the Image Plane Coordinates wrt the
% System States.
dxidX = diff(xi,X);
dxidY = diff(xi,Y);
dxidZ = diff(xi,Z);
dxidphi = diff(xi,ph);
dxidtheta = diff(xi,th);
dxidpsi = diff(xi,ps);

dxidxo = diff(xi,xo);
dxidyo = diff(xi,yo);
dxidzo = diff(xi,zo);
```

```
dyidX = diff(yi,X);
dyidY = diff(yi,Y);
dyidZ = diff(yi,Z);
dyidphi = diff(yi,ph);
dyidtheta = diff(yi,th);
dyidpsi = diff(yi,ps);

dyidxo = diff(yi,xo);
dyidyo = diff(yi,yo);
dyidzo = diff(yi,zo);


% Now have a symbolic representation of the Measurement Jacobian rows for
% this feature point
% First part wrt pose parameters
Cj_w = [    dxidX dxidY dxidZ dxidphi dxidtheta dxidpsi zeros(1,6) ;
            dyidX dyidY dyidZ dyidphi dyidtheta dyidpsi zeros(1,6) ];

% Second part wrt feature parameters
Cj_f = [    dxidxo dxidyo dxidzo ;
            dyidxo dyidyo dyidzo ];
```

A.4 MRP Process Jacobian

```
% Symbolic Process Jacobian for a Constant Velocity Model with MRP
% - Relative Orientation in Modified Rodrigues Parameters

% By Michael Tribou -- August 2009

clear;

% System States as Symbols
% Relative Orientation Parameters (Modelling Frame wrt Camera Frame)
syms a b c wx wy wz real;
% System Sampling Period
syms dt real;
```

```
% Construct the Process Jacobian
F = sym(eye( 12 ));

% Position States (X,Y,Z)
F(1:3,7:9) = dt*eye(3);

% Partial derivatives for MRP transition
dfada = dt*((a*wx)/2 + (b*wy)/2 + (c*wz)/2) + 1;
dfadb = dt*(wz/2 + (a*wy)/2 - (b*wx)/2);
dfadc = -dt*(wy/2 - (a*wz)/2 + (c*wx)/2);
dfadwx = dt*(a^2/4 - b^2/4 - c^2/4 + 1/4);
dfadwy = -dt*(c/2 - (a*b)/2);
dfadwz = dt*(b/2 + (a*c)/2);
dfbda = -dt*(wz/2 + (a*wy)/2 - (b*wx)/2);
dfbdb = dt*((a*wx)/2 + (b*wy)/2 + (c*wz)/2) + 1;
dfbdc = dt*(wx/2 + (b*wz)/2 - (c*wy)/2);
dfbdwx = dt*(c/2 + (a*b)/2);
dfbdwy = -dt*(a^2/4 - b^2/4 + c^2/4 - 1/4);
dfbdwz = -dt*(a/2 - (b*c)/2);
dfcda = dt*(wy/2 - (a*wz)/2 + (c*wx)/2);
dfcdb = -dt*(wx/2 + (b*wz)/2 - (c*wy)/2);
dfcdc = dt*((a*wx)/2 + (b*wy)/2 + (c*wz)/2) + 1;
dfcdwx = -dt*(b/2 - (a*c)/2);
dfcdwy = dt*(a/2 + (b*c)/2);
dfcdwz = -dt*(a^2/4 + b^2/4 - c^2/4 - 1/4);

F(4:6,4:6) =    [ ...
                    dfada dfadb dfadc ;
                    dfbda dfbdb dfbdc ;
                    dfcda dfcdb dfcdc ;
                ];

F(4:6,10:12) =  [ ...
                    dfadwx dfadwy dfadwz ;
                    dfbdwx dfbdwy dfbdwz ;
                    dfcdwx dfcdwy dfcdwz ;
                ];
```

A.5 UOM Inverse Depth Measurement Jacobian

```
% Symbolic Measurement Jacobian for a Pin-hole Camera Model
% - Relative Orientation in Modified Rodrigues Parameters
% - Feature Points in Inverse Depth Parameterization

% By Michael Tribou -- August 2009

clear;

% System States as Symbols
% Relative Pose Parameters (Modelling Frame wrt Camera Frame)
syms X Y Z a b c real;
% Feature Point Parameters (in Modelling Frame)
syms p q r gm ch tau real;

% Monocular Camera Model Parameters
% Focal Length in X^I and Y^I Directions
syms Fx Fy real;

% Transformation from Modelling to Camera Frame
Tc_m = [ ...
        mrp2R(a,b,c) [X Y Z]’ ;
        0 0 0 1 ];

% Feature Point Initial Observation Ray Direction
ro = rotationZ(gm)*rotationY(ch)*[ 0 0 1 ]’;

% Object Feature Point in the Modelling Frame
obj = [ p q r ]’ + 1/tau*ro;

% Transform the Object Feature Point into the Camera Frame
cam = simplify( Tc_m*[ obj ; 1 ] );

xc = cam(1);
```

```
yc = cam(2);
zc = cam(3);

% Project the Feature Point onto the Camera Image Plane
xi = -Fx * xc/zc;
yi = -Fy * yc/zc;

% Find the Partial Derivatives of the Image Plane Coordinates wrt the
% System States.
dxidX = diff(xi,X);
dxidY = diff(xi,Y);
dxidZ = diff(xi,Z);
dxida = diff(xi,a);
dxidb = diff(xi,b);
dxidc = diff(xi,c);

dxidp = diff(xi,p);
dxidq = diff(xi,q);
dxidr = diff(xi,r);
dxidgamma = diff(xi,gm);
dxidchi = diff(xi,ch);
dxidtau = diff(xi,tau);


dyidX = diff(yi,X);
dyidY = diff(yi,Y);
dyidZ = diff(yi,Z);
dyida = diff(yi,a);
dyidb = diff(yi,b);
dyidc = diff(yi,c);

dyidp = diff(yi,p);
dyidq = diff(yi,q);
dyidr = diff(yi,r);
dyidgamma = diff(yi,gm);
dyidchi = diff(yi,ch);
dyidtau = diff(yi,tau);
```

```
% Now have a symbolic representation of the Measurement Jacobian rows for
% this feature point
% First part wrt pose parameters
Cj_w = [    dxidX dxidY dxidZ dxida dxidb dxidc zeros(1,6) ;
            dyidX dyidY dyidZ dyida dyidb dyidc zeros(1,6) ];

% Second part wrt feature parameters
Cj_f = [    dxidp dxidq dxidr dxidgamma dxidchi dxidtau ;
            dyidp dyidq dyidr dyidgamma dyidchi dyidtau ];
```

A.6 Additional Camera Measurement Jacobian

```
% Symbolic Measurement Jacobian for Additional Pin-hole Camera Models
% - Relative Orientation in Modified Rodrigues Parameters
% - Feature Points in Inverse Depth Parameterization
% - Known Extrinsic Calibration wrt Primary Camera

% By Michael Tribou -- August 2009

clear;

% System States as Symbols
% Relative Pose Parameters (Modelling Frame wrt Camera Frame)
syms X Y Z a b c real;
% Feature Point Parameters (in Modelling Frame)
syms p q r gm ch tau real;

% Monocular Camera Model Parameters
% Focal Length in X^I and Y^I Directions
syms Fx Fy real;
% Relative Location wrt Primary Camera
syms xac yac zac phac thac psac real;

% Transformation from Modelling to Camera Frame
Tpc_m = [ ...
        mrp2R(a,b,c) [X Y Z]' ;
        0 0 0 1 ];

% Transformation from Additional Camera (ac) to Primary Camera (pc)
Tpc_ac = [ ...
        rotationZ(psac)*rotationY(thac)*rotationX(phac) [xac yac zac]’ ;
        0 0 0 1 ];

% Invert the Transformation
Tac_pc = invert_transform(Tpc_ac);

% Feature Point Initial Observation Ray Direction
ro = rotationZ(gm)*rotationY(ch)*[ 0 0 1 ]’;

% Object Feature Point in the Modelling Frame
obj = [ p q r ]’ + 1/tau*ro;

% Transform the Object Feature Point into the Additional Camera Frame
cam = Tac_pc*Tpc_m*[ obj ; 1 ];

xc = cam(1);
yc = cam(2);
zc = cam(3);

% Project the Feature Point onto the Camera Image Plane
xi = -Fx * xc/zc;
yi = -Fy * yc/zc;

% Find the Partial Derivatives of the Image Plane Coordinates wrt the
% System States.
dxidX = diff(xi,X);
dxidY = diff(xi,Y);
dxidZ = diff(xi,Z);
dxida = diff(xi,a);
dxidb = diff(xi,b);
dxidc = diff(xi,c);

dxidp = diff(xi,p);
```

```
dxidq = diff(xi,q);
dxidr = diff(xi,r);
dxidgamma = diff(xi,gm);
dxidchi = diff(xi,ch);
dxidtau = diff(xi,tau);


dyidX = diff(yi,X);
dyidY = diff(yi,Y);
dyidZ = diff(yi,Z);
dyida = diff(yi,a);
dyidb = diff(yi,b);
dyidc = diff(yi,c);

dyidp = diff(yi,p);
dyidq = diff(yi,q);
dyidr = diff(yi,r);
dyidgamma = diff(yi,gm);
dyidchi = diff(yi,ch);
dyidtau = diff(yi,tau);

% Now have a symbolic representation of the Measurement Jacobian rows for
% this feature point
% First part wrt pose parameters
Cj_w = [    dxidX dxidY dxidZ dxida dxidb dxidc zeros(1,6) ;
            dyidX dyidY dyidZ dyida dyidb dyidc zeros(1,6) ];

% Second part wrt feature parameters
Cj_f = [    dxidp dxidq dxidr dxidgamma dxidchi dxidtau ;
            dyidp dyidq dyidr dyidgamma dyidchi dyidtau ];
```

A.7 Point Range Sensor Measurement Jacobian

```
% Symbolic Measurement Jacobian for Point Range Finder Model
% - Relative Orientation in Modified Rodrigues Parameters
% - Feature Points in Inverse Depth Parameterization
% - Known Extrinsic Calibration wrt Primary Camera
% By Michael Tribou -- August 2009

clear;

% System States as Symbols
% Relative Pose Parameters (Modelling Frame wrt Camera Frame)
syms X Y Z a b c real;
% Parameters for the jth Feature Point (in Modelling Frame)
syms p q r gm ch tau real;

% Point Range Finder Model Parameters
% Number and Average of Feature Points Locations (in Modelling Frame)
syms N xo_avg yo_avg zo_avg real;
% Relative Location wrt Primary Camera
syms xr yr zr phr thr psr real;

% Transformation from Modelling to Primary Camera Frame
Tpc_m = [ ...
        mrp2R(a,b,c) [X Y Z]’ ;
        0 0 0 1 ];

% Transformation from Ranger (r) to Primary Camera (pc)
Tpc_r = [ ...
        rotationZ(psr)*rotationY(thr)*rotationX(phr) [xr yr zr]’ ;
        0 0 0 1 ];

% Invert the Transformation
Tr_pc = invert_transform(Tpc_r);

% Transformation from Modelling to Range Finder Frame
Tr_m = Tr_pc*Tpc_m;

% Feature Point Initial Observation Ray Direction
ro = rotationZ(gm)*rotationY(ch)*[ 0 0 1 ]’;

% Object Feature Point in the Modelling Frame
```

```
obj = [ p q r ]’ + 1/tau*ro;

% Separate the Measurement Prediction for Jacobian wrt Pose (w) and Feature
% Parameters (f)
gw = [ 0 0 1 0 ] * ( Tr_m*[ xo_avg yo_avg zo_avg 1 ]’ );
gf = [ 0 0 1/N 0 ] * ( Tr_m*[obj; 1] );

% Find the Partial Derivatives of the Range Measurement wrt the
% System States.
dgdX = diff( gw, X );
dgdY = diff( gw, Y );
dgdZ = diff( gw, Z );
dgda = diff( gw, a );
dgdb = diff( gw, b );
dgdc = diff( gw, c );

dgdp = diff( gf, p );
dgdq = diff( gf, q );
dgdr = diff( gf, r );
dgdgamma = diff( gf, gm );
dgdchi = diff( gf, ch );
dgdtau = diff( gf, tau );

% Now have a symbolic representation of the Measurement Jacobian rows for
% this measurement
% First part wrt pose parameters
C_w = [    dgdX dgdY dgdZ dgda dgdb dgdc zeros(1,6) ];

% Second part wrt feature parameters (for each Feature Point)
C_f = [    dgdp dgdq dgdr dgdgamma dgdchi dgdtau ];
```

A.8 Laser Pointer Sensor Measurement Jacobian

```
% Symbolic Measurement Jacobian for Laser Pointer Range Model
% - Relative Orientation in Modified Rodrigues Parameters
% - Feature Points in Inverse Depth Parameterization
% - Known Extrinsic Calibration wrt Primary Camera
% By Michael Tribou -- August 2009

clear;

% System States as Symbols
% Relative Pose Parameters (Modelling Frame wrt Camera Frame)
syms X Y Z a b c real;
% Parameters for the jth Feature Point (in Modelling Frame)
syms p q r gm ch tau real;

% Laser Pointer Model Parameters
% Number and Average of Feature Points Locations (in Modelling Frame)
syms N xo_avg yo_avg zo_avg real;
% Relative Location wrt Primary Camera
syms xl yl zl phl thl psl real;

% Monocular Camera Model Parameters
% Focal Length in X^I and Y^I Directions
syms Fx Fy real;

% Transformation from Modelling to Primary Camera Frame
Tpc_m = [ ...
        mrp2R(a,b,c) [X Y Z]' ;
        0 0 0 1 ];

% Transformation from Laser (l) to Primary Camera (pc)
Tpc_l = [ ...
        rotationZ(psl)*rotationY(thl)*rotationX(phl) [xl yl zl]' ;
        0 0 0 1 ];

% Invert the Transformation
Tl_pc = invert_transform(Tpc_l);

% Transformation from Modelling to Range Finder Frame
Tl_m = Tl_pc*Tpc_m;
```

```
% Feature Point Initial Observation Ray Direction
ro = rotationZ(gm)*rotationY(ch)*[ 0 0 1 ]';

% Object Feature Point in the Modelling Frame
obj = [ p q r ]' + 1/tau*ro;

% Separate the Measurement Prediction for Jacobian wrt Pose (w) and Feature
% Parameters (f)
gw = Tpc_l*[ ...
        0 ;
        0 ;
        [ 0 0 1 0 ] * ( Tl_m*[ xo_avg yo_avg zo_avg 1 ]' ) ;
        1 ];

gf = Tpc_l*[ ...
        0 ;
        0 ;
        [ 0 0 1/N 0 ] * ( Tl_m*[obj; 1] ) ;
        1 ];

% Project the new point into the image plane of the Primary Camera
xi_w = -Fx * gw(1)/gw(3);
yi_w = -Fy * gw(2)/gw(3);

xi_f = -Fx * gf(1)/gf(3);
yi_f = -Fy * gf(2)/gf(3);

% Find the Partial Derivatives of the Image Measurements wrt the
% System States.
dxidX = diff( xi_w, X );
dxidY = diff( xi_w, Y );
dxidZ = diff( xi_w, Z );
dxida = diff( xi_w, a );
dxidb = diff( xi_w, b );
dxidc = diff( xi_w, c );

dxidp = diff( xi_f, p );
```

```
dxidq = diff( xi_f, q );
dxidr = diff( xi_f, r );
dxidgamma = diff( xi_f, gm );
dxidchi = diff( xi_f, ch );
dxidtau = diff( xi_f, tau );

dyidX = diff( yi_w, X );
dyidY = diff( yi_w, Y );
dyidZ = diff( yi_w, Z );
dyida = diff( yi_w, a );
dyidb = diff( yi_w, b );
dyidc = diff( yi_w, c );

dyidp = diff( yi_f, p );
dyidq = diff( yi_f, q );
dyidr = diff( yi_f, r );
dyidgamma = diff( yi_f, gm );
dyidchi = diff( yi_f, ch );
dyidtau = diff( yi_f, tau );

% Now have a symbolic representation of the Measurement Jacobian rows for
% this measurement
% First part wrt pose parameters
C_w = [     dxidX dxidY dxidZ dxida dxidb dxidc zeros(1,6) ;
            dyidX dyidY dyidZ dyida dyidb dyidc zeros(1,6) ];

% Second part wrt feature parameters (for each Feature Point)
C_f = [     dxidp dxidq dxidr dxidgamma dxidchi dxidtau ;
            dyidp dyidq dyidr dyidgamma dyidchi dyidtau ];
```

References

[1] D Amarasinghe, GKI. Mann, and RG Gosine. Integrated laser-camera sensor for the detection and localization of landmarks for robotic applications. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, pages 4012–4017, 2008. 83

[2] J Andrade-Cetto and A Sanfeliu. The effects of partial observability in SLAM. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 397–402. Citeseer, 2004. 69, 70

[3] T Bailey and H Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006. 41, 44

[4] ET Baumgartner and SB Skaar. An autonomous vision-based mobile robot. *IEEE Transactions on Automatic Control*, 39(3):493–502, 1994. 11, 12

[5] P Bonnifait and G Garcia. Design and experimental validation of an odometric and goniometriclocalization system for outdoor robot vehicles. *IEEE Transactions on robotics and automation*, 14(4):541–548, 1998. 28

[6] JA Castellanos, JM Martinez, J Neira, and JD Tardos. Simultaneous map building and localization for mobile robots: a multisensor fusion approach. In *1998 IEEE International Conference on Robotics and Automation, 1998. Proceedings*, volume 2, 1998. 83

[7] F Chaumette and S Hutchinson. Visual servo control. I. Basic approaches. *IEEE Robotics and Automation Magazine*, 13(4):82–90, 2006. 3

[8] F Chaumette and S Hutchinson. Visual servo control, Part II: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118, 2007. 3

[9] J Civera, AJ Davison, and JMM Montiel. Dimensionless monocular SLAM. *Lecture Notes in Computer Science*, 4478:412, 2007. 13, 38, 44, 52, 54, 79, 85

[10] J Civera, AJ Davison, and JMM Montiel. Inverse depth to depth conversion for monocular slam. In *2007 IEEE International Conference on Robotics and Automation*, pages 2778–2783, 2007. 51

[11] J Crassidis and F Markley. Attitude Estimation Using Modified Rodrigues Parameters. In *Flight Mechanics/Estimation Theory Symposium 1996 p 71-83(SEE N 96- 30487 11-13)*, 1996. 54

[12] AJ Davison. Real-time simultaneous localisation and mapping with a single camera. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 1403–1410, 2003. 15, 43, 51, 81

[13] AJ Davison, AG Cid, and N Kita. Real-time 3D SLAM with wide-angle vision. In *Proc. IFAC Symposium on Intelligent Autonomous Vehicles*, 2004. 9, 13, 43

[14] AJ Davison, ID Reid, ND Molton, and O Stasse. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007. 15, 43, 54

[15] M Deans and M Hebert. Experimental Comparison of Techniques for Localization and Mapping Using a Bearing-Only Sensor. *Lecture Notes in Control and Information Sciences; Vol. 271*, pages 395–404, 2000. 38, 41, 42, 55, 70

[16] DF DeMenthon and LS Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1):123–141, 1995. 11

[17] L Deng, WJ Wilson, F Janabi-Sharifi, MDAS. Missions, and O Brampton. Decoupled EKF for simultaneous target model and relative pose estimation using feature points. In *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, pages 749–754, 2005. 3, 13, 45, 52, 56

[18] GN Desouza and AC Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002. 11

[19] M Dissanayake, P Newman, S Clark, HF Durrant-Whyte, and M Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, 2001. 12, 41, 42, 48, 57, 69, 70, 126

[20] H Durrant-Whyte and T Bailey. Simultaneous localization and mapping (SLAM): part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006. 41

[21] E Eade and T Drummond. Scalable Monocular SLAM. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 1*, pages 469–476. IEEE Computer Society Washington, DC, USA, 2006. 43, 51, 81

[22] DH Eberly. *3D game engine design: a practical approach to real-time computer graphics*. Morgan Kaufmann Pub, 2001. 52

[23] D Fontanelli, A Danesi, FAW Belo, P Salaris, and A Bicchi. Visual Servoing in the Large. *The International Journal of Robotics Research*, 28(6):802, 2009. 3, 46

[24] D Goshen-Meskin, IY Bar-Itzhack, MIA Ind, and BGI. Airport. Observability analysis of piece-wise constant systems. I. Theory. *IEEE Transactions on Aerospace and Electronic Systems*, 28(4):1056–1067, 1992. 61, 62, 63, 65

[25] AHA Hafez and CV Jawahar. Target model estimation using particle filters for visual servoing. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4. 45

[26] EJ Haug. *Computer aided kinematics and dynamics of mechanical systems*. Allyn and Bacon Boston, 1989. 30

[27] R Hermann and A Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, 1977. 20, 62, 69

[28] CCW Hulls and WJ Wilson. Integration of camera and range sensors for 3D pose estimation in robot visual servoing. In *Proceedings of SPIE*, volume 3523, page 76. SPIE, 1998. 10, 11, 13, 16, 52, 92, 96, 104

[29] MJ Jung, H Myung, SG Hong, DR Park, HK Lee, SW Bang, Y Kiheung-up, and K Kyunggi-do. Structured Light 2D Range Finder for Simultaneous Localization and Map-building (SLAM) in Home Environments. In *Proceedings of the 2004 International Symposium on Micro-NanoMechatronics and Human Science*, page 371. Institute of Electrical & Electronics Engineers (IEEE), 2004. 83

[30] M Kaess and F Dellaert. Visual SLAM with a multi-camera rig. 2006. 84, 89

[31] RE Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. 10

[32] T Kanade, H Kano, S Kimura, A Yoshida, and K Oda. Development of a video-rate stereo machine. In *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings*, volume 3, 1995. 87

[33] GH Kim, JS Kim, and KS Hong. Vision-based simultaneous localization and mapping with two cameras. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS 2005)*, pages 1671–1676, 2005. 85

[34] K Konolige and M Agrawal. Frame-Frame Matching for Realtime Consistent Visual Mapping. In *2007 IEEE International Conference on Robotics and Automation*, pages 2803–2810, 2007. 44

[35] Y Lu, JZ Zhang, QMJ Wu, and ZN Li. A survey of motion-parallax-based 3-D reconstruction algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(4):532–548, 2004. 16, 38, 41, 43, 55, 70, 90, 95

[36] C Madhusudan. Error analysis of the Kalman filtering approach to relative position estimation using noisy vision measurements. Master's thesis, University of Waterloo, 1992. 13

[37] E Malis, F Chaumette, and S Boudet. 2 1/2 D Visual Servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, 1999. 45

[38] P Martinet, N Daucher, J Gallice, and M Dhome. Robot control using monocular pose estimation. 3, 10

[39] P Martinet and J Gallice. Position based visual servoing using a non-linear approach. In *1999 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1999. IROS'99. Proceedings*, volume 1, 1999. 10

[40] D Marzorati, M Matteucci, D Migliore, and DG Sorrenti. Monocular SLAM with Inverse Scaling Parametrization. In *Proc. of the British Machine Vision Conference*, 2008. 12, 44

[41] H Michel and P Rives. Singularities in the determination of the situation of a robot effector from the perspective view of 3 points. 1993. 33, 34, 35, 37, 78

[42] M Montemerlo, S Thrun, D Koller, and B Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence*, volume 18, pages 1151–1156, 2003. 43, 83

[43] JMM Montiel, J Civera, and AJ Davison. Unified inverse depth parametrization for monocular SLAM. *analysis*, 9:1. 13, 43, 44, 49, 50, 51, 52, 54, 81

[44] T Nakamura, M Oohara, T Ogasawara, and H Ishiguro. Fast self-localization method for mobile robots using multiple omnidirectional vision sensors. *Machine Vision and Applications*, 14(2):129–138, 2003. 12

[45] K Ogata. *Modern control engineering*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1996. 20

[46] A Ohya, A Kosaka, and A Kak. Vision-based navigation of mobile robot with obstacle avoidance by single camera vision and ultrasonic sensing. In *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, volume 2, 1997. 11, 12

[47] S Soatto. Observability/identifiability of rigid motion under perspective projection. In *In Proc. of the 33rd IEEE Conf. on Decision and Control*, 1994. 76, 77

[48] J Sola, A Monin, M Devy, and T Vidal-Calleja. Fusing Monocular Information in Multicamera SLAM. *IEEE Transactions on Robotics*, 24(5):958–968, 2008. 84, 85

[49] MW Spong and M Vidyasagar. *Robot dynamics and control*. Wiley India Pvt. Ltd., 2008. 6

[50] S Thrun, W Burgard, and D Fox. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). 2005. 10, 12, 43, 82, 93

[51] T Vidal-Calleja, M Bryson, S Sukkarieh, A Sanfeliu, and J Andrade-Cetto. On the observability of bearing only SLAM. In *2007 IEEE International Conference on Robotics and Automation (ICRA), Roma, Italia*, pages 4114–4119. IEEE, 2007. 61, 69

[52] L Weiss, AC Sanderson, and CP Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, 3(5):404–417, 1987. 3

[53] WJ Wilson, CCW Hulls, and GS Bell. Relative end-effector control using Cartesian position based visual servoing: Special section on vision-based control of robot manipulators. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, 1996. 3, 6, 8, 9, 10, 11, 13, 15, 41, 44, 52, 86, 115

[54] Z Yan, W Wei, W Ke, and X Xiao-Dong. Mobile robot indoor simultaneous localization and mapping using laser range finder and monocular vision. *Acta Automatica Sinica*, 2005. 83

[55] Y Yoon, A Kosaka, and AC Kak. A New Kalman-Filter-Based Framework for Fast and Accurate Visual Tracking of Rigid Objects. *IEEE Transactions on Robotics*, 24(5):1238–1251, 2008. 3, 6, 11