Shape, Motion, and Inertial Parameter Estimation of Space Objects using Teams of Cooperative Vision Sensors

by

Matthew D. Lichter

Bachelor of Science with Honors, Mechanical Engineering
The Pennsylvania State University, 1999

Master of Science, Mechanical Engineering
Massachusetts Institute of Technology, 2001

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

at the

Massachusetts Institute of Technology

February 2005

Signature of Author .....
Department of Mechanical Engineering
October 8, 2004

Certified by .....
Steven Dubowsky
Professor of Mechanical Engineering
Thesis Supervisor and Committee Chairman

Accepted by .....
Professor Lallit Anand
Chairman, Committee on Graduate Studies

Shape, Motion, and Inertial Parameter Estimation of Space Objects using Teams of Cooperative Vision Sensors

by

Matthew D. Lichter

Submitted to the Department of Mechanical Engineering
on October 8, 2004, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Mechanical Engineering

ABSTRACT

Future space missions are expected to use autonomous robotic systems to carry out a growing number of tasks. These tasks may include the assembly, inspection, and maintenance of large space structures; the capture and servicing of satellites; and the redirection of space debris that threatens valuable spacecraft. Autonomous robotic systems will require substantial information about the targets with which they interact, including their motions, dynamic model parameters, and shape. However, this information is often not available a priori, and therefore must be estimated in orbit.

This thesis develops a method for simultaneously estimating dynamic state, model parameters, and geometric shape of arbitrary space targets, using information gathered from range imaging sensors. The method exploits two key features of this application: (1) the dynamics of targets in space are highly deterministic and can be accurately modeled; and (2) several sensors will be available to provide information from multiple viewpoints. These features enable an estimator design that is not reliant on feature detection, model matching, optical flow, or other computation-intensive pixel-level calculations. It is therefore robust to the harsh lighting and sensing conditions found in space. Further, these features enable an estimator design that can be implemented in real-time on space-qualified hardware.

The general solution approach consists of three parts that effectively decouple spatial- and time-domain estimations. The first part, referred to as kinematic data fusion, condenses detailed range images into coarse estimates of the target's high-level kinematics (position, attitude, etc.). A Kalman filter uses the high-fidelity dynamic model to refine these estimates and extract the full dynamic state and model parameters of the target. With an accurate understanding of target motions, shape estimation reduces to the stochastic mapping of a static scene.

This thesis develops the estimation architecture in the context of both rigid and flexible space targets. Simulations and experiments demonstrate the potential of the approach and its feasibility in practical systems.

Thesis Supervisor: Steven Dubowsky, Professor of Mechanical Engineering

ACKNOWLEDGEMENTS

I am grateful for the opportunity to work on such interesting projects with such a great group of people. I would like to thank the Japan Aerospace Exploration Agency (JAXA) for its support of this work, particularly Hiroshi Ueno, Dr. Yoshiaki Ohkami, and Dr. Mitsushige Oda for their continuing support and contributions to the program. I thank the National Defense Science and Engineering Graduate (NDSEG) Fellowship and the MIT Rosenblith Fellowship for their support of my studies during much of my time at MIT. I would like to thank all the members of the Field and Space Robotics Laboratory over the last five years, for helping to provide such an enjoyable and stimulating work environment. I thank Dr. Karl Iagnemma and the committee members, Professor George Barbastathis and Professor Sanjay Sarma, for their important guidance during the development of this work. Finally I would like to thank Professor Dubowsky for his support and mentorship of my studies at MIT. His critical evaluation of my work has been invaluable to my development as a researcher and engineer.

On a personal level, I would like to thank all the friends and family who have lent their support over the years here. My family has always been supportive and has encouraged me in everything I have ever done. I thank Tom McGuire and Tom Lancaster for five great and memorable years as roommates. I am grateful for the friendship of Matt Spenko, one of the few constant faces in a large laboratory that is a sea change of personalities every year. His optimism and perseverance through the ups and downs of research is contagious. Finally, I thank Brittany for her constant love and encouragement.

CONTENTS

FIGURES

TABLES

CHAPTER

1

INTRODUCTION

1.1 Motivation

Autonomous robotic systems are expected to play an increasing role in future orbital space operations. At present, operations such as satellite servicing and space structure assembly are carried out by human astronauts and require substantial extra-vehicular activity (EVA). Such activity is costly from an economic standpoint and poses considerable risk to human life, as underscored by the recent Space Shuttle Columbia tragedy. Further, many of these tasks tend to be repetitive and might be handled more efficiently and precisely using semi-autonomous robotic systems. For these reasons, the international space community is striving to increase the use of robotic systems in space [7, 36, 51, 55, 62, 75, 79, 80].

One type of space mission that would rely heavily on robotic systems is the assembly, maintenance, and inspection of large space structures, such as the International Space Station (ISS) or the JAXA Space Solar Power System (SSPS) [51, 61, 75, 80]. The amount of assembly time required and the repetitiveness of tasks often mandates a robotic solution. Robotic systems might be required to transport structural modules to a construction site, join subsections together using manipulators, and inspect the processes to ensure proper assembly.

Autonomous systems are also being considered for the refueling, repair, and reorbiting of satellites [36, 37, 38, 55, 57]. Historically, this type of operation has been performed manually by astronauts for only the highest value satellites such as the Hubble Space Telescope[1] (see Figure 1.1). The use of autonomous systems could potentially reduce the economic and human costs associated with servicing missions, thus allowing a greater number of satellites to be repaired rather than replaced.

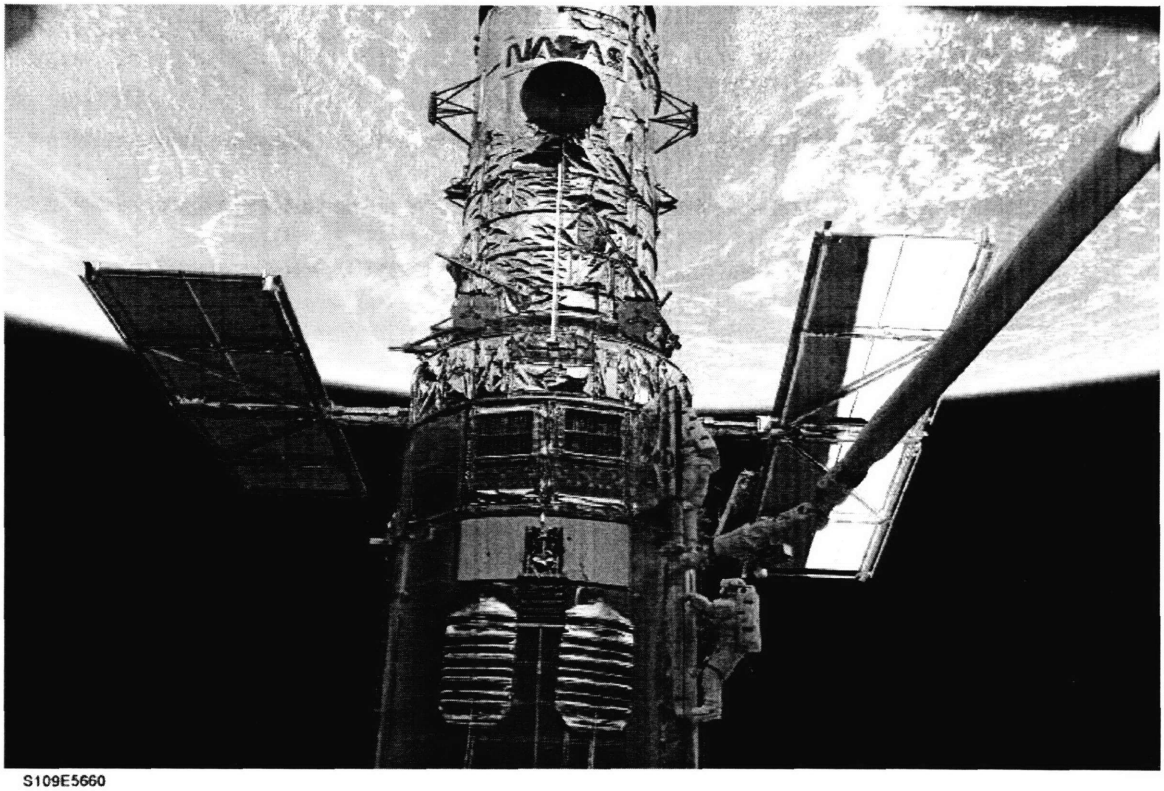


Figure 1.1. Hubble Space Telescope (HST) being serviced by astronauts during the March 2002 Hubble Servicing Mission (STS-109) (NASA photograph [58]).

A third potential application for autonomous systems might be the mitigation of space debris, in which low-cost disposable robots capture and redirect the orbits of large

[1] In light of the recent Columbia tragedy, even the final service mission for the HST was deemed too costly and has been canceled.

debris that threaten important spacecraft [42, 49, 50]. The current method for dealing with potential debris collisions involves changing the orbit of the spacecraft, a very costly procedure that expends large quantities of fuel and human resources, and interrupts communications and experiments in progress. Consider the following statement from the NASA Space Shuttle Mission Summary, STS-108, December 2001 [69]:

> *"Flight controllers planned slight changes to [Space Shuttle] Endeavour's departure from the [International Space] Station Dec. 15, allowing time for a small jet firing by the Shuttle to boost the Station's future path away from a piece of space debris that could pass near the complex. Mission Control was notified that a spent Russian rocket upper stage launched in the 1970s could pass within three miles of the Station if Endeavour did not perform the engine firing. With the Shuttle reboost, the Station was predicted to pass more than 40 miles away from the debris. Because the scheduled reboost used additional propellant, Endeavour did not perform a full-circle flyaround of the Station after undocking. Instead, the Shuttle undocked from the Station, performing a quarter circle flyaround of the complex to a point about 400 feet directly above the Station where it fired its engines in a final separation burn at 12:20 a.m. EST, beginning its departure from the orbiting outpost."*

The fuel cost of deflecting the orbit of a small piece of debris might be orders of magnitude smaller than that of a very large spacecraft or structure. Therefore, even with associated hardware costs, a debris-deflecting robotic system might be an economically viable alternative to current practices [59].

Each of these mission types involves substantial physical interaction between robot(s) and target. To achieve sufficient safety margins and robustness while minimizing fuel use and interaction forces, autonomous systems will require accurate information about their targets [36, 37, 38, 55, 57]. A robot will need to understand the shape or geometry of its target so that appropriate contact and servicing points can be located. To plan a graceful capture of a tumbling satellite, a robotic system will need to know the pose and relative velocity of the target. To assemble structural modules, a robot will need to understand their vibrational behavior. If some type of model-predictive planning or control is used for any of these missions, the dynamic model parameters will be needed (e.g. center of mass, principal inertial axes, modal frequencies, etc.).

This information, however, is often unknown to the robotic systems a priori. In the case of damaged satellites or debris, the shape and mass parameters of the target may be uncertain or unknown. Many targets do not have the ability to communicate their motions to the robotic system due to a lack of sensing or communication hardware (e.g. non-instrumented structural modules, malfunctioning satellites, debris, etc.). Therefore, a critical task for autonomous space robots is the on-orbit estimation of motion, dynamic model parameters, and shape of targets, using some type of remote sensing.

This thesis will explore the combined estimation of dynamic state, model parameters, and geometric shape of space targets using vision-type sensors. Throughout this thesis, the term *vision-type sensor* is used to refer to any type of 2- or 3-D imaging system, including monocular cameras, stereo cameras, laser rangefinders, laser radar, volume holography, etc.

1.2 Background Literature

The fundamental problem addressed by this thesis is the simultaneous estimation of both the structure of an object and its motion relative to observers. This section presents literature related to this problem.

1.2.1 Shape Estimation from Pixel-Level Details

Estimation of the shape of a static scene using information embedded in 2-D images is a topic that has been studied thoroughly by many researchers. The solutions are often referred to as "shape from X" methods. For example, many have studied the estimation of shape from shading [22, 32, 46, 81] and shape from texture [3, 24, 34, 41]. Intensity gradients and textural distortions in the 2-D image provide cues about the geometric gradients of the underlying surfaces with respect to the camera. The methods are often formulated in a differential equation framework, in which shape estimates are derived by integrating these gradients and applying appropriate boundary conditions.

While effective in many important practical applications, these methods are not well-suited to the space applications addressed in this thesis. Pixel gradient computations are highly sensitive to pixel-level noise induced by the harsh lighting environment of space. Intense sunlight and strong shadows create high-contrast scenes that wash out subtle textural details in images. Reflective spacecraft surfaces such as solar panels and metallic foils also degrade the performance of many of these methods, reducing their efficiency, accuracy, or robustness. Therefore, this approach to shape estimation, while very useful for many important applications, cannot be robustly applied to the space applications considered here.

1.2.2 Shape Estimation from Triangulation

Another approach to the shape estimation problem employs triangulation rather than pixel-level cues. Shape from stereo [4, 11, 21, 33, 54] and shape from multiple views [76, 88] are closely related methods that rely on multiple camera viewpoints to perform triangulation. These algorithms typically locate pixels in different images that correspond to the same point in the environment (referred to as the *correspondence problem*) and compute their separation distance (*disparity*) in the image plane. This information, combined with knowledge of the camera focal properties and the relative kinematics between viewpoints, is used to compute the point's depth from the cameras. Accurate knowledge of the relative pose between the viewpoints and lens distortions is critical to achieving accuracy in range measurements. Stereo imaging is widely popular in practical applications and work has been done to make the algorithms work in real-time [56, 82, 87] and in challenging lighting situations [39].

Shape from structured light [13, 16, 18, 86] and shape from shadows [8, 26, 65] are also popular approaches based on triangulation. By knowing the direction or location of an external light source (laser stripes, patterned light, etc.), and observing how it projects onto the scene, a 3-D shape estimate can be built. For example, shadows in an environment "slice" through objects, creating detectable curves in the image plane. As the light source moves, a series of curves can be assembled into a surface map of the scene using knowledge of the relative kinematics between camera and light source. Many laser-based technologies incorporate this principle by raster-scanning a scene with a narrow laser beam, recording its reflected location in the image plane, and triangulating the position of that point using the known kinematics between camera and laser.

Triangulation-based methods tend to be more robust in harsh lighting situations than the pixel-level methods described in the previous section. One reason for this is that they do not require computing intensity gradients between pixels or detecting textural

patterns. They also tend to be less computationally intensive. However, they do require more hardware and can still be confused by reflective surfaces [39], which are common in space applications since most spacecraft are covered by wrinkled metallic films and reflective solar panels. Any estimator that uses these methods in space should therefore be robust to substantial noise in the range measurements.

1.2.3 Motion Estimation with Known Shape or Features

The other half of the problem addressed by this thesis involves motion estimation. Motion estimation of known objects is another very well-studied problem in the literature. Solution methods can be categorized in a number of ways, but generally involve matching features in the observations to a known model. These features may be artificial (fiducials) or natural (edges, corners, etc.), and may consist of points, lines, surfaces, or volumes. The matching can be done in the 2-D image plane or in 3-D space. This approach has been discussed in the context of space applications [37, 38, 73].

These methods, while useful for many practical applications, can only be applied to a small subset of the applications addressed in this thesis. Many of the robotic tasks discussed earlier involve unknown or uncertain objects. Accurate a priori geometric models of the targets might not be available for space debris, damaged satellites, or thermally warped structures. Furthermore, these methods require the detection and tracking of features, a task that is often difficult in the harsh lighting environment of space.

1.2.4 Simultaneous Shape and Motion Estimation: Mesh-Based Methods

While many researchers have studied the estimation of shape with known motion and vice versa, fewer have explored the simultaneous estimation of both, although some

solutions do exist. Many of these methods start with detailed shape meshes of the target or scene (possibly obtained using any of the methods described above). Moving the sensor and/or target provides a series of overlapping meshes. The meshes are then stitched together to determine relative camera motions and provide a more complete shape estimate [6, 15, 27, 28, 31, 90]. The stitching is typically done off-line via numerical optimization or recursively with a Kalman filter.

While useful for certain terrestrial applications, these methods are not well-suited to the space applications discussed here. They are typically computationally intensive, using iterative point-matching routines or very high-dimensional Kalman filters to stitch the meshes. This precludes their use in real-time space systems, which typically have limited computation (often an order of magnitude slower than modern desktop PCs).

1.2.5 Simultaneous Shape and Motion Estimation: Feature-Based Methods

Rather than estimate detailed shape meshes directly, many simultaneous estimation methods rely on continuous tracking of high-level features. By maintaining an inventory of detected features, the methods also estimate high-level geometric structure. Typically a Kalman filter is used to efficiently estimate both the feature locations and the motion parameters in a joint framework.

Many feature-based methods have examined the estimation of an unknown object moving with respect to a fixed observer. Natural or artificial features of the object (e.g. corners, edges, fiducials, etc.) are located and tracked over time to understand the high-level motions and the general structure of the rigid target. Some methods employ physics-based dynamic models [9, 10, 89], while others do not [43, 67, 91].

Perhaps the most widespread use of feature-based methods in recent years has been in the area of Simultaneous Localization and Mapping (SLAM) [19, 48]. Here the

task is to construct a feature map of a (typically) static environment while constantly localizing the moving sensors with respect to this map. This is a slight reformulation of the previous methods; the camera is moving instead of the target object.

Again, while effective for many important applications, these methods are not well-suited to the problem addressed by this thesis since they are dependent upon feature detection. The harsh lighting, occlusions, and reflective materials found in orbit will make the reliable detection and correspondence of features nearly impossible at times. Additionally, these methods by themselves do not provide a detailed estimate of the shape of the target; they provide only a sparse set of feature points pertaining to the object, and therefore do not fully address the estimation requirements of the applications considered here. Increasing the number of feature points in order to improve shape detail results in a very high-dimensional Kalman filter, which quickly saturates the limited computational resources of space systems.

1.3 Contributions of this Thesis

This thesis addresses the simultaneous estimation of shape, motion, and dynamic model parameters of targets in orbit using range imaging sensors. It solves the problem by exploiting two unique features of the application. The first is that dynamics of bodies in the orbital environment are highly deterministic and hence can be accurately modeled. The second is that several cooperative sensors are available to observe the target from multiple vantage points. These two attributes allow a robust and efficient solution that is not reliant on subtle pixel-level details, feature detection, or model matching. Further, they allow the estimator to have low computational burden, potentially enabling its implementation in real-time on space-qualified hardware.

This thesis describes a general estimation approach and develops it in the context of representative space robotic applications. Theoretical development as well as simulation and experimental results are presented.

1.4 Thesis Organization

The thesis has seven chapters. This chapter presents motivation and background for the thesis. Chapter 2 provides a detailed technical discussion of the problem and introduces the general solution approach of the thesis. Chapters 3 and 4 implement the approach in the context of rigid-body motions. Chapter 3 provides theoretical development while Chapter 4 presents simulation and experimental studies. Chapters 5 and 6 present the estimation architecture implemented in the context of vibrational motions. Chapter 5 provides theoretical development while Chapter 6 presents simulation and experimental evaluations. Chapter 7 summarizes the thesis and suggests avenues for future research.

CHAPTER

2

GENERAL SOLUTION APPROACH

2.1 Problem Statement and Assumptions

This thesis explores the use of vision-type sensors to simultaneously estimate motion, dynamic model parameters, and geometric shape of space targets. This section states the general considerations and assumptions that apply to the applications considered in this work.

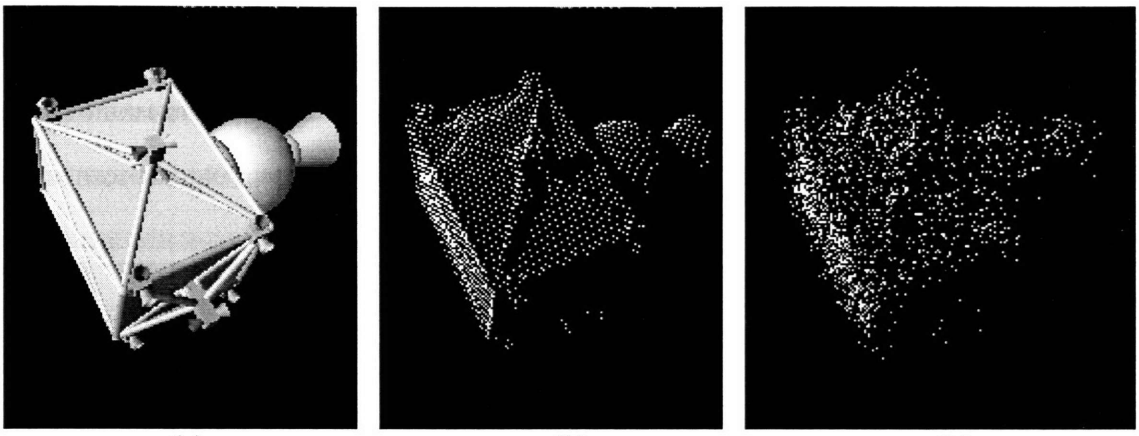
Harsh lighting conditions such as intense sunlight, dark shadows, and reflective surfaces may degrade image quality considerably. This lighting environment induces substantial noise in the 2- or 3-D images. Sensory data may be absent altogether from certain regions of the target due to visual occlusions or poor image quality. Many of these effects may change as the target moves. For example, features may move in and out of shadow as the target tumbles through space, yielding inconsistent observations over time. Therefore it may be difficult to reliably compute image correspondences or track features as the target moves.

There may be a high degree of uncertainty in the target prior to rendezvous. The motions of the target are very likely to be unknown ahead of time. The geometry and dynamic model parameters of the target might also be unknown or uncertain. For example, a priori shape models and parameters may be unavailable for damaged satellites, space debris, and thermally warped structures.

Computational resources are expected to be very limited. Space-qualified computation hardware is often an order of magnitude slower than that found in a modern desktop PC. Further, autonomous space systems must budget their processor time amongst a wide array of tasks, including communications, actuation, decision-making, and housekeeping functions. Only a fraction of the processing time may be available for the sensing tasks described in this thesis.

It is assumed that a dynamic model exists that faithfully reflects the dominant mechanics of the target. The parameter values for the model may be uncertain or unknown, but the equation structure itself is assumed to be correct. There may be uncertainties in the model due to external disturbances and unmodeled effects. These are assumed to be an order of magnitude smaller than the dominant effects included in the model.

Three-dimensional range images are assumed to come from vision-type sensors such as stereo cameras, laser range finders, LADAR, volume holography, etc. [29, 82, 84]. Each sensor is assumed to take a 3-D "snapshot" of the target object, where each pixel in the image has an associated depth measurement. The data sampling is assumed to be near-instantaneous with respect to target dynamics (i.e. images are not blurred). The output from each sensor consists of a cloud of points, ideally located on the visible surface of the target (see Figure 2.1). In practice, noise is present in these measurements. Each data point is assumed to have error in the range (z) direction and errors in the focal plane (x, y) directions, which vary among sensor implementations (see Figure 2.2). It is assumed that the noise statistics are reasonably well characterized and understood.



(a) *(b)* *(c)*

Figure 2.1. Raw sensory data (single sensor): (a) target object; (b) ideal range image; (c) noisy range image.

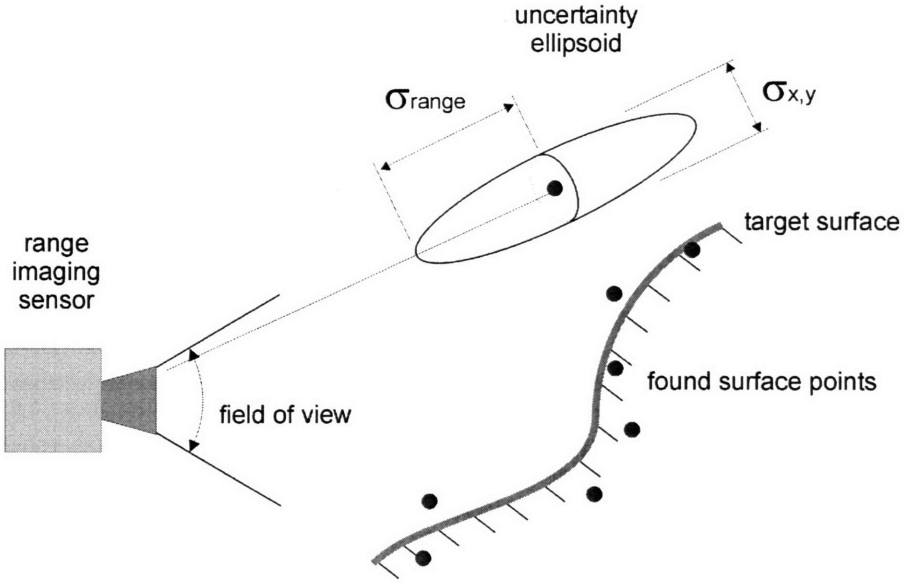


Figure 2.2. Sensory noise in range images.

Several cooperating sensors might be used to observe a space target, as shown in Figure 2.3. For such situations, it is assumed that the relative pose between sensors is well known, so that data may be expressed in a common reference frame. Range image errors due to sensor alignment uncertainty are assumed to be small compared to other sources of sensory error. The sensors are assumed to be synchronized so that the

collective range image corresponds to approximately one instant in time. In other words, synchronization errors are assumed to be negligible with respect to the target dynamics and sensor sample rate.

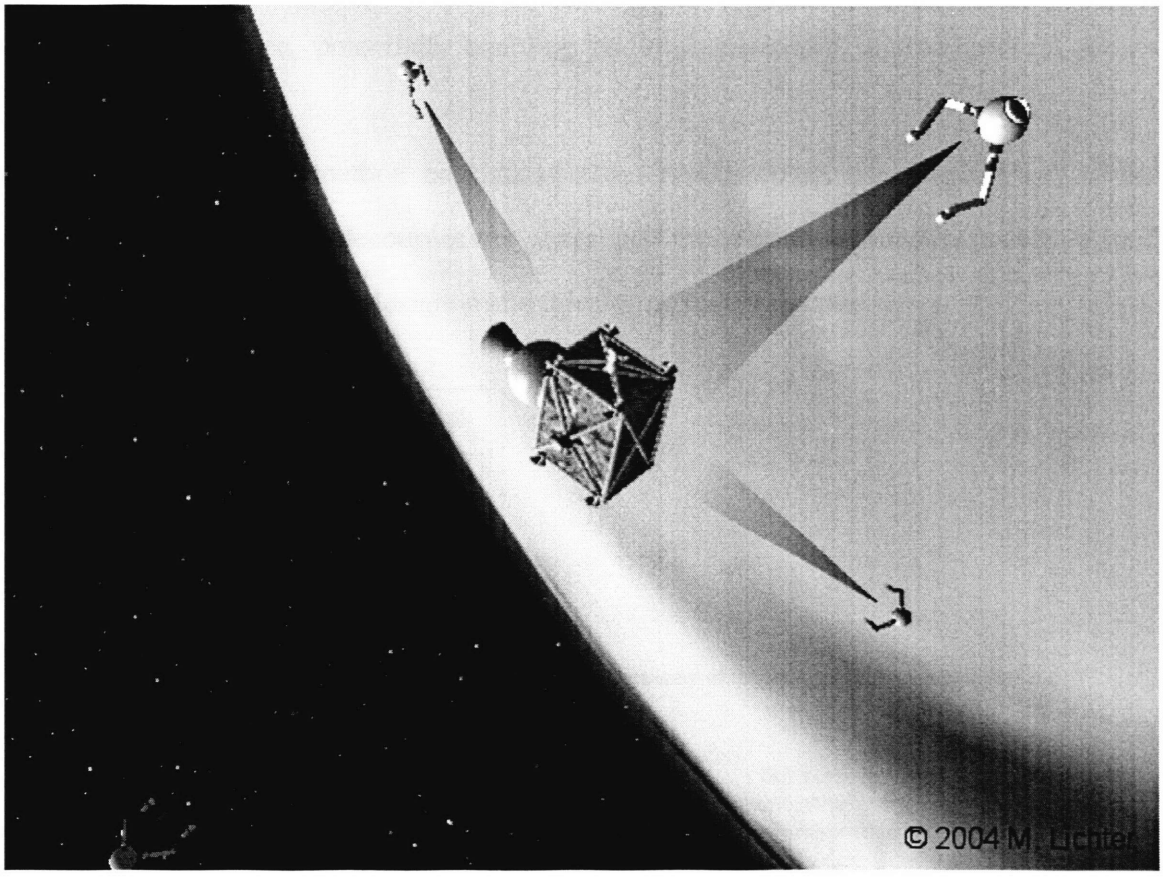


Figure 2.3. Cooperative observation: four sensors distributed about a target object. Image made using OpenGL [68].

2.2 Solution Approach

As stated in the introduction, this thesis addresses the simultaneous estimation of shape, motion, and dynamic model parameters of targets in orbit using range imaging sensors. It solves the problem by exploiting two unique features of the application. The first is that dynamics of bodies in the orbital environment are highly deterministic and

hence can be accurately modeled. The second is that several cooperative sensors are available to observe the target from multiple vantage points. These two attributes allow a robust and efficient solution that is not reliant on subtle pixel-level details, feature detection, or model matching. Further, they allow the estimator to have low computational burden, potentially enabling its implementation in real-time on space-qualified hardware.

The estimation method consists of three distinct parts (see Figure 2.4). This modularized architecture decouples the shape and the motion estimation problems, and allows its components to be designed and developed independently.

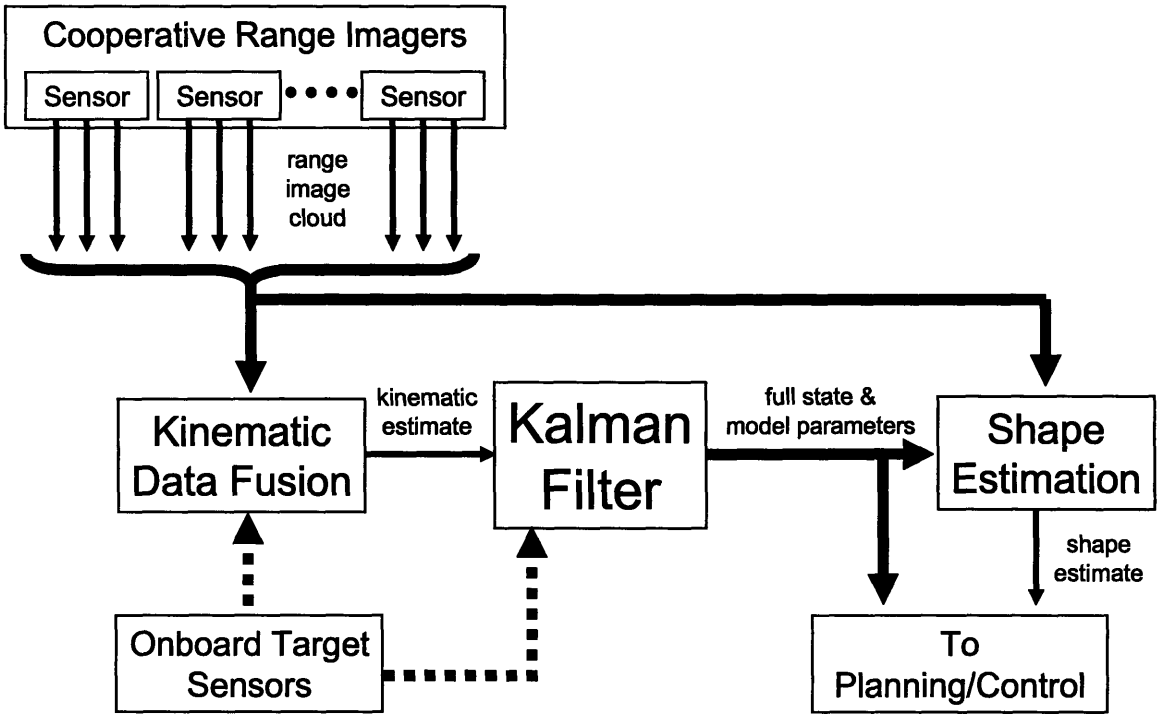


Figure 2.4. Estimator components showing information flow.

The first part of the architecture is referred to here as *kinematic data fusion*. This step is a data reduction process that condenses noisy, pixel-level sensory data into coarse,

high-level position information of the target at each sample time. For example, this step might use a range image cloud to compute the rough position and attitude of a target (Chapter 3), or the approximate modal coefficients of a vibrating structure (Chapter 5). For kinematic data fusion, the design emphasis is on speed and robustness rather than high accuracy.

These coarse kinematic estimates are then viewed as *surrogate measurements* that feed into a mechanics-based Kalman filter. Exploiting the high-fidelity dynamic model of the target, the Kalman filter is able to remove substantial errors in the surrogate measurements and extract the full dynamic state and model parameters of the target. It is computationally efficient for two reasons. First, a Kalman filter is an efficient, recursive, linear estimator. Second, the state estimate is of low dimension, since it involves only high-level motion information (pose and velocity) and a few dynamic model parameters. No pixel details, shape meshes, or large feature sets are estimated in the Kalman filter.

The final part is a *shape estimator*, which uses pixel-level information in conjunction with motion estimates from the Kalman filter. With an accurate understanding of target motions, raw sensory data can be fused across time steps into a target-fixed reference frame. Thus, a dynamic mapping problem turns into a static one. Further, this fusion can be performed in a statistically rigorous manner and can incorporate sensor uncertainty models if desired.

This estimation architecture is applicable to the most basic scenario in which no a priori knowledge of the target is available. It can easily be extended to make use of additional information when available. For example, a priori knowledge of target shape can be used to initialize the shape estimate. A priori estimates of target motions and parameters can be used to initialize the Kalman filter. If the target is able to communicate motion information (e.g. from onboard gyros, accelerometers, etc.), then

this information can be fused in a statistically optimal manner in either the Kalman filter or the kinematic data fusion step (depending on the type of information provided).

In the following chapters, the details of this estimation architecture are discussed in the context of representative space applications.

CHAPTER

3

ESTIMATION OF RIGID BODY MODES: THEORETICAL DEVELOPMENT

3.1 Problem Statement

Many future space missions hope to use autonomous robotic systems for the manipulation and servicing of satellites and space structure components in orbit [7, 36, 37, 38, 55, 57, 79]. For such missions, it is critical that the robotic systems have some knowledge of the dynamics and geometry of the targets with which they interact. Since this information is often not known a priori, accurate sensing and estimation of this information is an important task. The estimation problem addressed in this chapter is:

> *Estimate the position, attitude, rotational and translational velocities, location and relative magnitudes of the principal inertial axes, and surface geometry of a rigid uncontrolled spacecraft, using data gathered from cooperative 3-D range imaging sensors.*

3.1.1 Constraints and Assumptions

To reiterate from the previous chapter, explicit a priori information regarding the target's geometry and mass parameters may be uncertain or nonexistent. In some cases, only order of magnitude bounds on this information may be available.

For the rigid body estimation problem, the target is assumed to be a single body of approximately constant shape. That is, vibration amplitudes are assumed to be relatively small compared to the dominant target geometry. The effects of fuel sloshing are assumed to be negligible. The target is also assumed to rotate less than 90 degrees between sensor sample times.

3.1.2 Solution Approach

This problem will be solved using the general method introduced in Section 2.2. The first step, referred to as kinematic data fusion, will involve coarsely estimating the position and attitude of the target using range image data (Section 3.2). A Kalman filter will then exploit its accurate model of the target dynamics to filter noise from these estimates and extract the full state and parameters of the target (Sections 3.3 and 3.4). Finally, with accurate knowledge of the target's motions, shape estimation reduces to the stochastic mapping of a static scene (Section 3.5). This chapter provides theoretical development for this estimator.

3.2 Kinematic Data Fusion: Coarse Pose Estimation

As shown in Figure 2.4 (page 26), the first part of the estimation architecture must convert range image data into a rough pose estimate of the target at each sample time. This step is essentially a prefilter, condensing detailed visual information into high-level pose information (surrogate measurements) that can be digested easily by the Kalman filter (Sections 3.3 and 3.4).

This type of problem is common in the literature, and numerous methods have been demonstrated. However, while many of these methods are effective for particular terrestrial applications, they are not ideally suited to the application addressed here. For

example, one might fit an a priori geometric model to the data, using landmarks or contour-fitting [37, 38, 73, 89, 91]. Many published methods focus primarily on accuracy. However, they are often computationally intensive as the number of data points or complexity of the model grows. Landmark detection and model matching also tend to be unreliable under harsh space lighting and reflective satellite materials. Finally, such approaches do not address situations in which no good model exists, which is often the case with damaged spacecraft and debris.

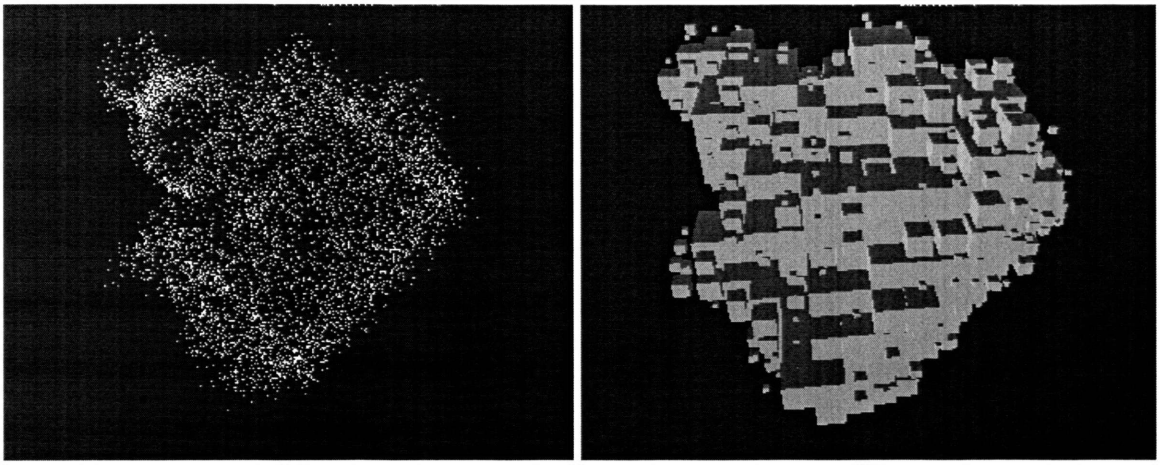
For this problem an alternative method is needed. This method should not require a priori models. It should be robust to the lighting environment of space. Most importantly, it should focus on computational speed and robustness rather than accuracy, with the expectation that the Kalman filter – exploiting the high-fidelity dynamic model – will remove errors later.

3.2.1 A Computationally Simple Pose Estimator

A unique aspect of this application is that teams of sensors can be used to gather information from different locations. This is an important feature that can be exploited to greatly simplify pose estimation in many cases.

If several sensors are distributed about the target, they will provide a noisy 3-D cloud of points as shown in Figure 3.1a. A rough estimate of target position could be found by computing the geometric centroid of this cloud. However, this is not robust since sensors that happen to be close to the target will provide a higher density and larger number of sample points than sensors located farther away. Centroid computations would thus be biased towards closer sensors. One way to compensate for this is to discretize the point cloud into voxels (volume elements), with each voxel having an occupancy level proportional to the number of sample points found within it (see Figure

3.1b). The occupancy values can then be saturated at some pre-defined threshold to reduce the bias effects of closer sensors.



(a) *(b)*

Figure 3.1. (a) Raw sensory data; (b) voxel representation.

If the target is rigid, this voxel representation is approximately constant in gross shape even as the target moves. Tracking the centroid and principal geometric axes of the voxel image could therefore provide a simple way to coarsely track the target's pose. These quantities can be computed in an analogous manner to computing the center of mass and principal inertial axes of a solid body. That is, the centroid location $\vec{r}_m$ of the voxel image is computed as

$$\vec{r}_m = \frac{\sum_i \alpha_i \cdot \vec{r}_i}{\sum_i \alpha_i} \tag{3.1}$$

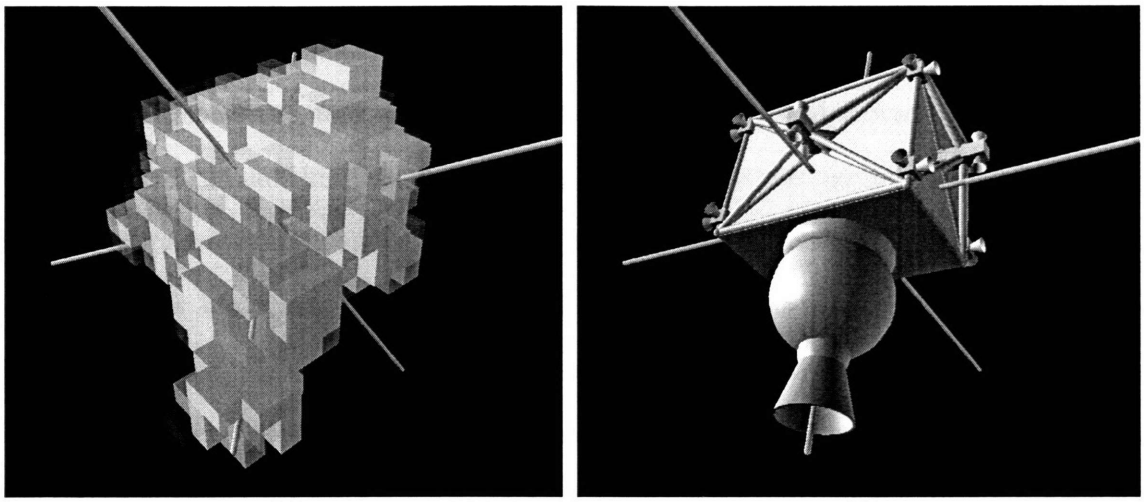
where α_i is the occupancy value of the i^{th} voxel and $\vec{r}_i$ is the position of that voxel in global coordinates. Similarly, second moments of the voxel image can be computed, analogous to inertia computations for a solid body:

$$
\begin{aligned}
J_{xx} &= \sum_i \alpha_i \left\{ \left(r_i^{(y)} - r_m^{(y)}\right)^2 + \left(r_i^{(z)} - r_m^{(z)}\right)^2 \right\} \\
J_{yy} &= \sum_i \alpha_i \left\{ \left(r_i^{(x)} - r_m^{(x)}\right)^2 + \left(r_i^{(z)} - r_m^{(z)}\right)^2 \right\} \\
J_{zz} &= \sum_i \alpha_i \left\{ \left(r_i^{(x)} - r_m^{(x)}\right)^2 + \left(r_i^{(y)} - r_m^{(y)}\right)^2 \right\} \\
J_{xy} &= J_{yx} = -\sum_i \alpha_i \left(r_i^{(x)} - r_m^{(x)}\right)\left(r_i^{(y)} - r_m^{(y)}\right) \\
J_{yz} &= J_{zy} = -\sum_i \alpha_i \left(r_i^{(y)} - r_m^{(y)}\right)\left(r_i^{(z)} - r_m^{(z)}\right) \\
J_{zx} &= J_{xz} = -\sum_i \alpha_i \left(r_i^{(z)} - r_m^{(z)}\right)\left(r_i^{(x)} - r_m^{(x)}\right)
\end{aligned}
\qquad (3.2)
$$

The superscripts on r_i and r_m denote the x, y, or z component of those vectors. The second moments are assembled into a matrix whose eigenvectors are computed, satisfying

$$
[J] = \begin{bmatrix} J_{xx} & J_{yx} & J_{zx} \\ J_{xy} & J_{yy} & J_{zy} \\ J_{xz} & J_{yz} & J_{zz} \end{bmatrix} = [R_m] \cdot [\Lambda] \cdot [R_m]^T . \qquad (3.3)
$$

Here J is the geometric moment matrix (analogous to an inertia matrix), Λ is the diagonal eigenvalue matrix of J, and R_m is the eigenvector matrix. This computation locates the principal geometric axes of the voxel image (see Figure 3.2). The rotation matrix R_m describes the attitude of the principal geometric axes with respect to the global reference frame.



(a) *(b)*

Figure 3.2. (a) Voxel grid and its computed principal geometric axes; (b) these axes superimposed on the actual target.

It is important to note that these geometric axes do not correspond to the actual center of mass or principal inertial axes of the target; they are based merely on superficial geometry of the body. However, since both reference frames are fixed to the target, the kinematic relation between the geometric axes and the principal inertial axes remains constant (see Figure 3.3).

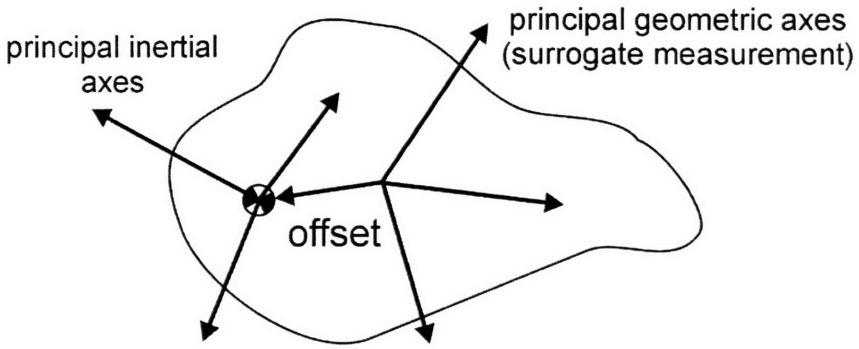


Figure 3.3. A constant rotational and translational offset exists between principal geometric axes and principal inertial axes of the target.

The quantities $\vec{r}_m$ and R_m represent the outputs of the kinematic data fusion step. They are coarse surrogate measurements of target position and attitude that are used as inputs to the Kalman filter (Sections 3.3 and 3.4). This is not the only way of providing these surrogate measurements. The simple method presented here is used to emphasize the point that kinematic data fusion need not be a highly accurate or computation-intensive process. It will be seen shortly that the Kalman filter can remove substantial noise from these surrogate measurements and that the estimation architecture as a whole can perform well with even the simplest of kinematic data fusion methods.

3.2.2 Implementation Notes and Degenerate Conditions

There is an important implementation detail to note. Before using this information in the Kalman filter, an ambiguity must be removed from the solution of R_m. First, the eigenvalues are sorted in ascending order and the columns of R_m are reordered appropriately. This keeps the x, y, z axes from transposing as the target rotates. Second, the rotation matrix R_m must map to a right-handed coordinate system. This is tested by computing the determinant of R_m; it is 1 for a right-handed frame, -1 for a left-handed frame. If R_m is found to be left-handed, then one of the columns of R_m should be negated. At this point, the ambiguity has been reduced to four possible solutions of R_m: the correct one, and three others, each of which are a 180-degree rotation about one of the three axes. To disambiguate between the four, one simply checks the dot products of each column of R_m with the corresponding columns of the previously computed R_m from the last time step. Since it was assumed that the target rotates less than 90 degrees per sample time, the dot products between corresponding columns of two consecutive rotation matrices must be positive. If any dot products are negative, then the corresponding column of the current R_m should be negated.

The computationally simple method presented here has an important degeneracy, which occurs when the target has a high degree of axial symmetry. In such cases, two or three of the eigenvalues of J will have approximately the same value, which under-constrains the solution of Equation (3.3). For example, a long cylinder has only one obvious geometric axis (the other two can be specified arbitrarily as long as they form an orthogonal basis), and a sphere has no obvious geometric axes. This problem can be handled by introducing more information (higher-order moments, color information from the sensors, etc.) to constrain the solution of Equation (3.3).

All further discussion will assume that an unambiguous pose estimate can be obtained, either by the non-degeneracy of the target, or through the incorporation of additional information.

3.3 Kalman Filtering: Rotational Estimation

The Kalman filter forms the core of the estimation architecture, using the surrogate measurements along with an accurate dynamic model to estimate the full dynamic state and inertial parameters of the target. The dynamic state consists of rotational and translational positions and velocities. External forces and torques on the target are assumed to be negligible. Gravity gradient torques and orbital mechanics effects can be incorporated into the model; however their contribution is typically small over short time intervals. In practice it is usually sufficient to model them as process noise in the Kalman filter. Because the rotational and translational dynamics are decoupled, this estimation can be performed using two distinct Kalman filters.

The parameters to be estimated include the principal inertias of the target (relative magnitudes only) and the pose offset between the principal inertial axes and the principal geometric axes (see Figure 3.3). As discussed in the previous section, the geometric axes

computed during kinematic data fusion do not correspond to the principal inertial axes of the target (see Figure 3.3). However, since both sets of axes are fixed to the body, the offset between the frames is constant and can be parameterized for estimation in the Kalman filter.

3.3.1 Rotation Representation

For the rotational estimation, spatial rotations will be represented not using rotation matrices, but with unit quaternions (Euler parameters). This simplifies many of the dynamics equations and provides a near-minimal framework for dealing with spatial rotations. Further, the unit quaternion has only a single normality constraint, which is much easier to enforce than the orthonormality of rotation matrices. Euler angles are not considered due to their well-known singularity problems. Other rotational representations exist [5, 63], but these are more complex and are unnecessary for this problem. Here, the quaternion representation is sufficient and robust.

Note that the invertible mapping from rotation matrices to quaternions is one-to-two, and thus some simple logic must be used in filter implementations to remove ambiguity from any conversions. The mapping of a quaternion to a rotation matrix is given by

$$[R(\vec{q})] \equiv \begin{bmatrix} \left(q_0^2 + q_1^2 - q_2^2 - q_3^2\right) & 2(q_1 q_2 - q_3 q_0) & 2(q_1 q_3 + q_2 q_0) \\ 2(q_1 q_2 + q_3 q_0) & \left(q_0^2 - q_1^2 + q_2^2 - q_3^2\right) & 2(q_2 q_3 - q_1 q_0) \\ 2(q_1 q_3 - q_2 q_0) & 2(q_2 q_3 + q_1 q_0) & \left(q_0^2 - q_1^2 - q_2^2 + q_3^2\right) \end{bmatrix}.$$

The shorthand notation $R(\vec{q})$ is used here to mean "the rotation matrix representation of quaternion $\vec{q}$". The inverse of this mapping (i.e. from rotation matrix to quaternion) will be denoted in shorthand as

$$q(R) \equiv R^{-1}(R).$$

For brevity, the lengthy conversion formula is not presented here[1]. The quaternion result of a concatenation of rotations can be computed using quaternion multiplication [70]. Quaternion multiplication is non-commutative, and will be denoted here by the operator $\otimes$:

$$\vec{c} = \vec{a} \otimes \vec{b} \equiv \begin{bmatrix} b_0 & -b_1 & -b_2 & -b_3 \\ b_1 & b_0 & b_3 & -b_2 \\ b_2 & -b_3 & b_0 & b_1 \\ b_3 & b_2 & -b_1 & b_0 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{Bmatrix}. \tag{3.4}$$

This equation is equivalent to the statement, "A rotation $\vec{c}$ is formed by rotating first by $\vec{a}$ and then by $\vec{b}$."

3.3.2 Notation

Variables to be used in the rotational estimator are listed here for convenience.

- $\vec{\omega}$ angular velocity vector in global (inertial) coordinates
- $\vec{\omega}_b$ angular velocity vector in principal inertial axis reference frame
- $\vec{q}_b$ quaternion describing attitude of principal inertial axes
- $\vec{q}_g$ quaternion describing attitude of principal geometric axes
- $\vec{q}_m$ surrogate measurement of $\vec{q}_g$ (from kinematic data fusion)
- $\vec{q}_d$ quaternion describing rotational offset between geometric and inertial axes
- $\vec{I}$ normalized 3 - vector containing the principal inertias
- $\vec{q}_I$ re - parameterization of $\vec{I}$ into quaternion space

[1] Interested readers may consult [70] for details.

The state vector and the measurement vector for the rotational Kalman filter are denoted respectively as

$$\vec{x} \equiv \begin{Bmatrix} \vec{\omega} \\ \vec{q}_g \\ \vec{q}_d \\ \vec{q}_I \end{Bmatrix} \qquad \vec{y} \equiv \{\vec{q}_m\}. \tag{3.5}$$

All variables listed above that are not part of $\vec{x}$ or $\vec{y}$ are considered intermediate variables, whose relation to the state vector will be described in the following sections. Note that the variables used in the state vector $\vec{x}$ must reside in coordinate systems that cannot be changed by the filter. For example, the state vector $\vec{x}$ must use the globally-referenced angular velocity vector $\vec{\omega}$ rather than the body-referenced velocity vector $\vec{\omega}_b$, since the body-fixed reference frame is defined by the variables $\vec{q}_g$ and $\vec{q}_d$, which are initially unknown to the estimator and are estimated.

3.3.3 Measurement Model

Since the Kalman filter uses quaternions rather than rotation matrices, it will use the quaternion parameterization of the rotation matrix computed from Equation (3.3):

$$\vec{q}_m \equiv q(R_m). \tag{3.6}$$

The quantity $\vec{q}_m$ is a coarse surrogate measurement of the target's principal geometric axes. Sensor noise, artifacts due to voxelization, and other phenomena will all add noise to this surrogate measurement. In other words, the measurement $\vec{q}_m$ is given as the attitude of the true geometric axes $\vec{q}_g$ concatenated with a small noise rotation $\vec{q}_w$:

$$\vec{q}_m = \vec{q}_g \otimes \vec{q}_w . \tag{3.7}$$

The noise quaternion $\vec{q}_w$ is a random variable, whose expected value[1] represents a zero rotation (i.e. $E[\vec{q}_w] = \{1 \quad 0 \quad 0 \quad 0\}^T$). Equation (3.7) represents the measurement model to be used by the rotational Kalman filter.

3.3.4 Parameter Estimates

The rotational filter will estimate the relative attitude between the principal geometric axes and the principal inertia axes (see Figure 3.3). This rotational offset is defined so that

$$\vec{q}_g \equiv \vec{q}_b \otimes \vec{q}_d \tag{3.8}$$

where $\vec{q}_b$ is the unit quaternion describing the attitude of the principal inertial axes of the target and $\vec{q}_d$ is the unit quaternion describing the offset rotation between the two axes.

The filter will also estimate the relative magnitudes of the principal inertias, denoted by the 3-vector $\vec{I}$. Since only relative magnitudes are observable, this vector must be normalized[2]. Further, this vector has a natural constraint in that the sum of any two elements cannot be less than the third. Graphically, the inertia vector $\vec{I}$ can be viewed as residing in a closed set on the unit sphere (see Figure 3.4).

[1] The expectation and covariance of a unit quaternion random variable is defined in Appendix A.

[2] If known torques were applied to the target, then the absolute values would be observable. This simplifying case will not be discussed here.

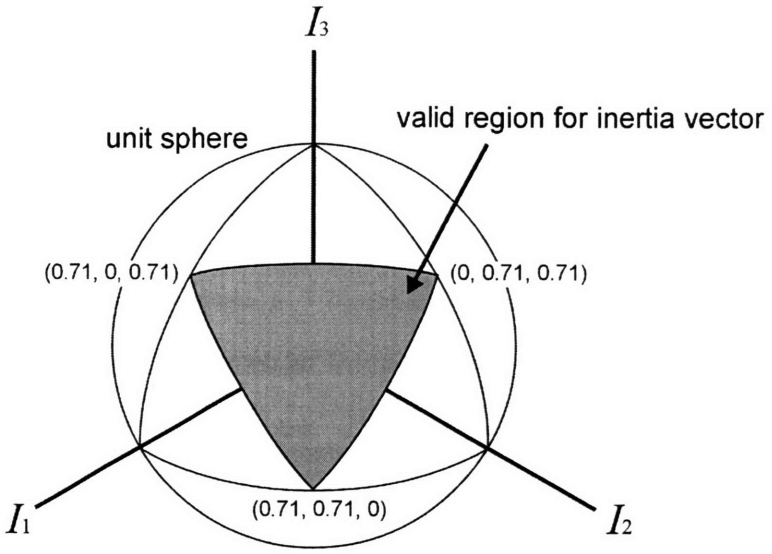


Figure 3.4. The inertia vector resides in a closed set on the unit sphere.

These constraints pose a challenge to the Kalman filter that can be eliminated through a re-parameterization onto an unbounded space. One way to do this is to parameterize the inertias with a unit quaternion[1], $\vec{q}_I$. The mapping from $\vec{q}_I$ to $\vec{I}$ is given by

$$\begin{Bmatrix} I_1 \\ I_2 \\ I_3 \end{Bmatrix} = \begin{Bmatrix} |z_2| + |z_3| \\ |z_1| + |z_3| \\ |z_1| + |z_2| \end{Bmatrix} \tag{3.9}$$

where

[1] The normality constraint on $\vec{q}_I$ should not be cause for concern. An implementation of the Kalman filter will soon be seen that is designed specifically for use with unit quaternions, and since other portions of the Kalman state involve quaternions (i.e. $\vec{q}_g$ and $\vec{q}_d$), there is no additional overhead in using a quaternion to parameterize the inertia vector here.

$$\begin{Bmatrix} z_1 \\ z_2 \\ z_3 \end{Bmatrix} = [R(\vec{q}_I)] \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 2(q_1 q_3 + q_2 q_0) \\ 2(q_2 q_3 - q_1 q_0) \\ \left(q_0^2 - q_1^2 - q_2^2 + q_3^2\right) \end{Bmatrix}_I .$$

This mapping can be interpreted as follows. A unit z-vector is rotated through space by quaternion $\vec{q}_I$ to obtain a 3-dimensional unit vector, $\vec{z}$. The vector $\vec{I}$ is computed using the absolute values of the components of $\vec{z}$, implicitly satisfying the natural constraints on the inertia vector. This mapping is C-zero continuous (due to the absolute value operator) and was observed to yield good performance. Alternatively, the absolute value operators could be replaced with smooth (e.g. quadratic) functions to yield higher-order continuity in the mapping function; however this did not seem necessary for any of the studies conducted here.

3.3.5 Process Model

In the absence of external torques, a rigid body's velocity and attitude follow the dynamics given by

$$\frac{d}{dt}\begin{Bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{Bmatrix}_b = \begin{Bmatrix} \frac{I_2 - I_3}{I_1}\omega_2\omega_3 \\ \frac{I_3 - I_1}{I_2}\omega_3\omega_1 \\ \frac{I_1 - I_2}{I_3}\omega_1\omega_2 \end{Bmatrix}_b \tag{3.10}$$

$$\frac{d}{dt}\begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix}_b = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix}_b \begin{Bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{Bmatrix}_b \equiv \frac{1}{2}[\Omega_b]\cdot\vec{q}_b \tag{3.11}$$

where $\vec{q}_b$ is the quaternion rotation from the global frame to the body's principal inertial axes, $\vec{\omega}_b$ is the angular rate about the body's principal axes, and $\vec{I}$ is the principal

inertias of the body [70]. Note that since the parameters $\vec{q}_d$ and $\vec{q}_I$ are constants, their process model is trivial.

Note the use of intermediate variables in these equations. The variables $\vec{q}_b$ and $\vec{I}$ map to the state vector using Equations (3.8) and (3.9), respectively. The velocity vector $\vec{\omega}_b$ is related to the state estimate through the coordinate transformation

$$\vec{\omega} \equiv [R(\vec{q}_b)]\vec{\omega}_b . \tag{3.12}$$

3.3.6 Kalman Filter Implementation

Equations (3.6)-(3.12) describe the measurement and process models for the state vector $\vec{x} = \{\vec{\omega} \quad \vec{q}_g \quad \vec{q}_d \quad \vec{q}_I\}^T$ and measurement vector $\vec{y} = \{\vec{q}_m\}$, and define the mappings to and from the intermediate variables $\vec{I}$, $\vec{\omega}_b$, and $\vec{q}_b$. This problem cannot be solved using the standard Kalman filter. First, the process and measurement models are nonlinear. Second, the state and measurement vectors contain unit quaternions, which have normality constraints and cannot be added using component-wise vector addition.

However, Kalman filtering of unit quaternions is a well-studied problem, due to the number of applications involving the estimation of rotations. Recent developments have provided effective and robust solution methods [17, 44, 45, 53]. Rather than obscure the focus of this chapter, the extensive details of the filter implementation are provided in Appendix A.

The result is that a Kalman filter can be designed and implemented to properly handle the unique constraints posed by this problem. Chapter 4 will demonstrate the effectiveness of this filter through simulation and experimental studies.

3.3.7 Degenerate Conditions

There are a few degenerate conditions that should be noted. The first occurs when two or more principal inertias of the target are identical. In this scenario, the principal axes are undefined by definition, since an infinitude of orientations of the principal frame yield a diagonal inertia matrix. In other words, there are multiple solutions to the estimation problem. In practice, however, this does not pose a problem since the Kalman filter will converge to one possible solution and remain there.

A second degenerate condition occurs if the target has no rotational velocity. In this case, the inertias and the kinematic offset are unobservable. As a result, the attitude of the principal inertial axes will be unknown. However, the principal geometric axes will still be observable and therefore the motions of that target-fixed reference frame can still be tracked.

If the target is spinning about a single axis (a flat spin), relative inertias are again unobservable. However, some inertia information can be extracted. The major or minor[1] principal inertial axis must be coincident with the angular velocity vector, and the center of mass must lie on this axis. This vector is observable in global coordinates. Again, the attitude of the principal inertial axes will be uncertain, but the principal geometric axes will be available to track the object.

From a practical standpoint, the lack of inertia information in the second and third cases is not a major concern. Inertia information will be available if and only if it is needed to predict the target trajectory, which is the ultimate goal to the mission planner. Furthermore, capturing a docile target might be the preferred scenario to capturing a wildly tumbling object. Therefore inducing a multi-axis tumble in the target by external

[1] In practice, it is likely to be the major axis rather than the minor axis [70].

means in order to observe inertial information might be counterproductive to greater mission objectives.

The one degenerate case which poses a significant challenge is the case of a *near*-flat spin. In this case, if the small off-axis rotations (wobbles) are of the same order of magnitude as measurement noise, the estimator might interpret them as such and converge to a flat spin estimate. This is an important degeneracy because it is fairly common in practice and is the only situation in which target pose is incorrectly identified (albeit with small error).

3.4 Kalman Filtering: Translational Estimation

The other half of the Kalman filter design involves translational state and parameter estimation. Although the translational and rotational dynamics are decoupled, it will be seen that one parameter in the translational estimation will rely on a reasonable estimate of the attitude of the principal geometric axes, $\vec{q}_g$. For this reason, rotational estimation updates must precede translational updates at each time step.

Again, external forces and torques on the target are assumed to be negligible. Orbital mechanics effects can be incorporated into the model; however their contribution is very small over short time intervals. In practice it is usually sufficient to model them as process noise in the Kalman filter.

3.4.1 Notation

Variables to be used in the translational estimator are listed here for convenience.

$\vec{v}$ translational velocity of target mass center, in global (inertial) coordinates

$\vec{r}$ cartesian position of target mass center, in global coordinates

$\vec{r}_g$ cartesian position of target geometric centroid, in global coordinates

$\vec{r}_m$ surrogate measurement of $\vec{r}_g$ (from kinematic data fusion)

$\vec{r}_d$ translational offset between $\vec{r}$ and $\vec{r}_g$, in principal geometric axes ref. frame

The state vector and the measurement vector for the translational Kalman filter are denoted respectively as

$$\vec{x} \equiv \begin{Bmatrix} \vec{v} \\ \vec{r} \\ \vec{r}_d \end{Bmatrix} \qquad \vec{y} \equiv \{\vec{r}_m\}. \tag{3.13}$$

3.4.2 Kinematic and Dynamic Relations

In the absence of external forces and orbital mechanics effects, a rigid body moving in space follows the dynamics given by

$$\frac{d}{dt}\begin{Bmatrix} v_x \\ v_y \\ v_z \\ r_x \\ r_y \\ r_z \end{Bmatrix} = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & 0 & & & \\ 1 & & & 0 & & \\ & 1 & & & 0 & \\ & & 1 & & & 0 \end{bmatrix} \begin{Bmatrix} v_x \\ v_y \\ v_z \\ r_x \\ r_y \\ r_z \end{Bmatrix} \tag{3.14a}$$

or in discrete time,

$$\begin{Bmatrix} v_x \\ v_y \\ v_z \\ r_x \\ r_y \\ r_z \end{Bmatrix}_{k+1} = \begin{bmatrix} 1 & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ \Delta t & & & 1 & & \\ & \Delta t & & & 1 & \\ & & \Delta t & & & 1 \end{bmatrix} \begin{Bmatrix} v_x \\ v_y \\ v_z \\ r_x \\ r_y \\ r_z \end{Bmatrix}_k \tag{3.14)b}$$

where $\vec{v}$ and $\vec{r}$ are the translational velocities and position of the center of mass in the inertial frame.

The geometric centroid of the target reference frame, $\vec{r}_g$, is related to the center of mass through the kinematic relation

$$\vec{r}_g = \vec{r} + \left[R(\vec{q}_g)\right] \vec{r}_d \tag{3.15}$$

where $\left[R(\vec{q}_g)\right]$ is the rotation matrix describing the attitude of the principal geometric axes, and $\vec{r}_d$ is a translational offset parameter to be estimated.

The surrogate measurement $\vec{r}_m$ given by Equation (3.1) is an approximate measurement of the geometric centroid of the target. That is,

$$\vec{r}_m = \vec{r}_g + \vec{r}_w = \vec{r} + \left[R(\vec{q}_g)\right] \vec{r}_d + \vec{r}_w \tag{3.16}$$

where $\vec{r}_w$ is a zero-mean noise vector in $\Re^3$.

3.4.3 Discrete-Time Kalman Filter Implementation

The translational estimation problem is relatively simple compared to the rotational estimation. The state and measurement equations are all linear and mostly constant. Estimation can be carried out efficiently using a standard discrete-time Kalman

filter [12]. The ideal process and measurement models are compiled from Equations (3.14)-(3.16) as

$$\vec{x}_k \equiv \begin{Bmatrix} \vec{v} \\ \vec{r} \\ \vec{r}_d \end{Bmatrix}_k = \begin{bmatrix} I & 0 & 0 \\ [\Delta t \cdot I] & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{Bmatrix} \vec{v} \\ \vec{r} \\ \vec{r}_d \end{Bmatrix}_{k-1} \equiv [A] \cdot \vec{x}_{k-1} \tag{3.17}$$

$$\vec{y}_k \equiv \{\vec{r}_m\}_k = \begin{bmatrix} 0 & I & R(\hat{q}_g)_k \end{bmatrix} \begin{Bmatrix} \vec{v} \\ \vec{r} \\ \vec{r}_d \end{Bmatrix}_k \equiv [C_k] \cdot \vec{x}_k \tag{3.18}$$

where the I and 0 (identity and zero) matrices are 3-by-3. The constant state transition matrix is denoted by $[A]$ and the measurement matrix is denoted by $[C_k]$. At each time step only the $R(\hat{q}_g)_k$ portion of $[C_k]$ needs to be updated, based on the latest estimate of the quaternion $\vec{q}_g$ from the rotational filter.

The measurement covariance matrix for the Kalman filter should approximately describe the coarseness of the surrogate measurement:

$$\Lambda_{yy} = E\left[(\vec{r}_m - \vec{r}_g)(\vec{r}_m - \vec{r}_g)^T\right].$$

The a posteriori state estimate should be initialized to an unbiased value, which in the absence of additional information is given as

$$\hat{x}_0 = E[\vec{x}_0] = \begin{Bmatrix} \vec{0} \\ \vec{y}_0 \\ \vec{0} \end{Bmatrix}.$$

The initial estimation covariance is given by

$$\Lambda_{xx} = E\left[(\vec{x}_0 - \hat{x}_0)(\vec{x}_0 - \hat{x}_0)^T\right].$$

Any reasonable values should be acceptable here. By its simple nature and linear models, the translational estimation generally converges very quickly.

The rest of the filter implementation is classical and will be omitted for brevity. Interested readers may consult literature such as [12] for further details.

3.5 Shape Estimation: Stochastic Mapping of Static Scene

With a good estimate of target motion, the problem of shape estimation is greatly simplified. As seen in Figure 2.4 (page 26), shape estimation uses both the raw sensory data as well as refined information from the Kalman filters. By understanding the motions of the target with respect to the sensors, the problem transforms from the estimation of a dynamic scene into a static one, simplifying the problem dramatically.

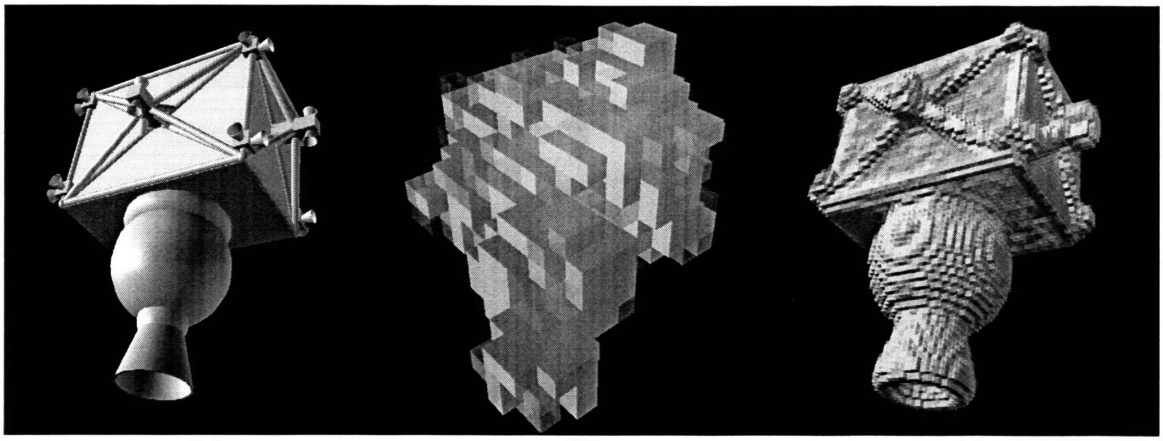
This problem is essentially a stochastic mapping problem. Numerous methods have been developed for the mapping of static scenes [71, 77]. This thesis will illustrate one simple approach to demonstrate the general concept, although more sophisticated methods could be used if necessary.

3.5.1 A Simple Recursive Shape Estimator

As was done in the kinematic data fusion algorithm (Section 3.2.1), a voxel grid will be used to represent the shape of the target. However, this grid will be aligned in a body-fixed reference frame[1] rather than a global frame. This will allow data from different sample times to be fused into a common map even as the target moves. This

[1] It is best to use the principal geometric axes reference frame (defined by $\vec{q}_g$ and $\vec{r}_g$, Sections 3.3 and 3.4), since their estimates are relatively stable throughout estimation.

grid will have a much finer resolution as well, since it effectively uses information from all time samples. Figure 3.5 shows a comparison between the voxel grids used by the kinematic data fusion algorithm and the shape estimator.



(a) (b) (c)

Figure 3.5. (a) Target object; (b) coarse grid used by kinematic data fusion; (c) refined probabilistic grid used by shape estimator.

Each voxel in the refined map will represent the probability that the location is occupied by the target. At each time step, a probabilistic map can be built based on the current data and sensor uncertainty models [77]. Figure 3.6 shows a 2-D illustration of the process, where white points indicate range image data and squares represent individual voxels. Higher probability of occupancy is indicated by lighter shading. In this illustration, the probabilistic map indicates the possibility that the target exists in voxels near the measured points, since each data point was sensed with some uncertainty (see Figure 2.2, page 24).

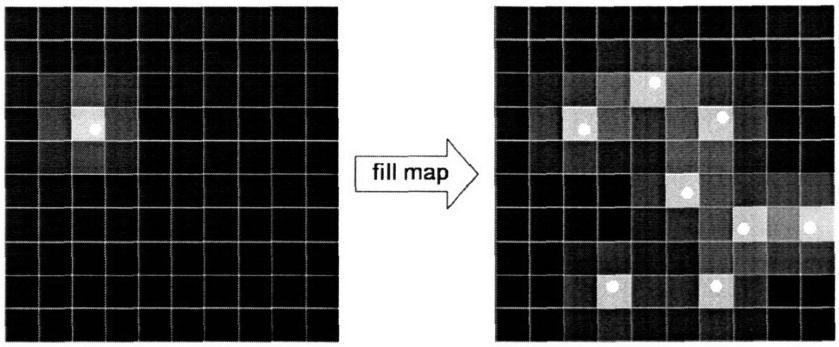


Figure 3.6. Filling a probabilistic voxel map using range data and sensor uncertainty models.

After building a map at each sample time, the maps could be combined into a cumulative map that incorporates information from all time steps. For computational efficiency, this could be a recursive process that updates an existing map using current data. In this way, only a cumulative voxel map would need to be stored, rather than a large (and growing) array of data points from all sample times. One way of implementing this concept is to scale the existing map by a forgetting factor and increment the result using the latest range data (see Figure 3.7). This method would preferentially weight new data over old data.

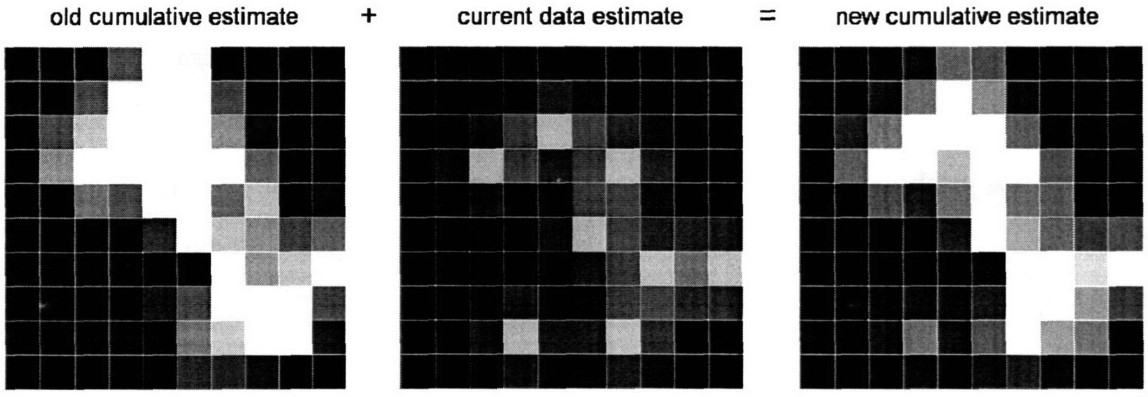


Figure 3.7. Recursive update of shape estimate.

This is a rather simple method, used primarily to illustrate the concept, although it does provide good shape estimates in practice. Certainly, more advanced methods could be used if necessary to more rigorously handle sensor uncertainty and other subtle effects [71, 77, 78].

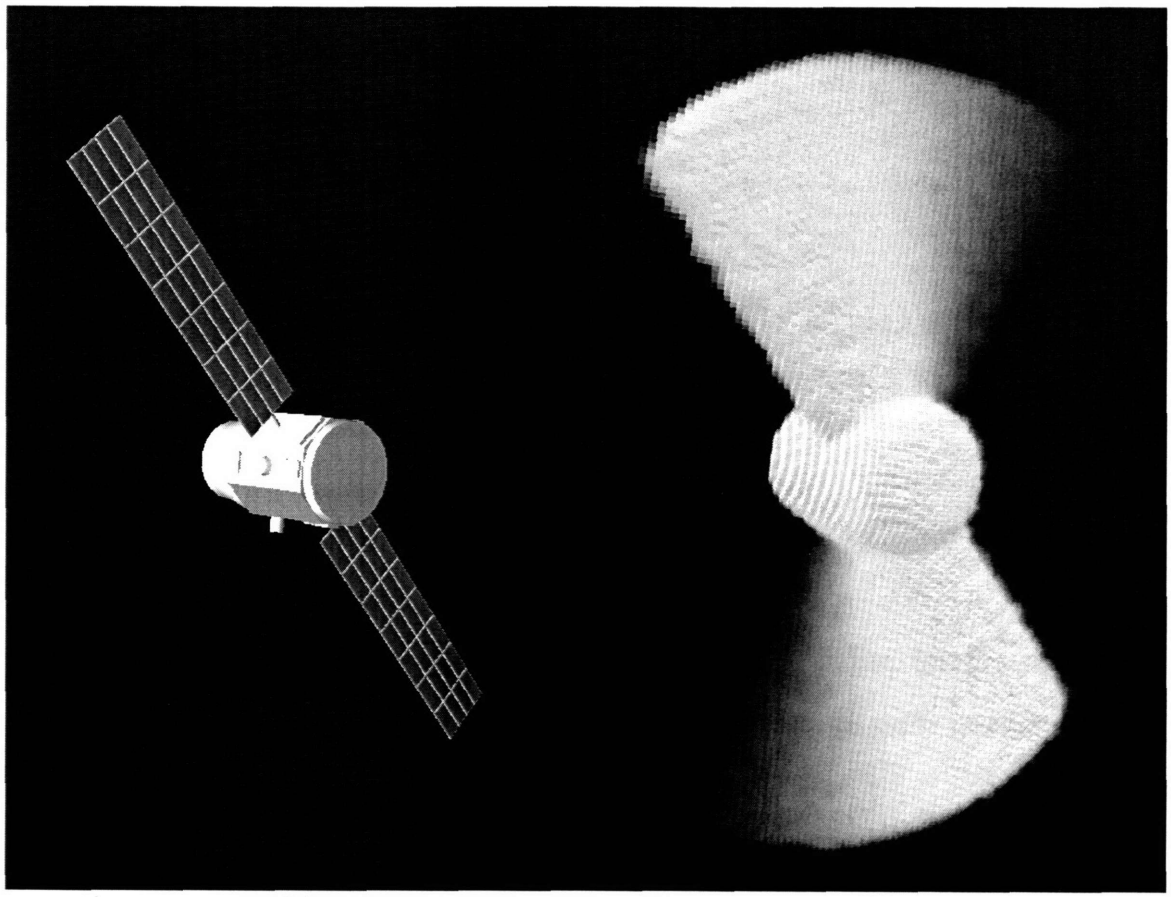
3.5.2 Computation of Uncertainty

It may be desirable to quantify the uncertainty in the shape estimate, to see if it has converged to a stable estimate. One metric for quantifying this uncertainty is the Shannon entropy metric [66, 78]

$$U(t) \equiv -\sum_i [\rho_i \ln(\rho_i) + (1-\rho_i)\ln(1-\rho_i)] \tag{3.19}$$

where ρ_i indicates the probability of occupancy of the i^{th} voxel. The value of U is large when many voxels have an intermediate likelihood value (maximum uncertainty), and small when most voxels are at the extremes (minimum uncertainty, i.e. $\rho_i = 0$ or $\rho_i = 1$).

If the Kalman filter fails, the pose estimate $\hat{q}_g$ diverges from the true value, causing the shape estimate to "smear" (see Figure 3.8). Voxel probabilities tend towards intermediate values, causing the uncertainty metric to rise. Thus, monitoring the uncertainty metric may provide some level of on-line divergence checking for the entire estimator as a whole.



(a) (b)

Figure 3.8. Kalman filter failure causes the shape estimate to "smear" and become less certain: (a) actual target; (b) poor shape estimate due to Kalman filter failure.

CHAPTER

4

ESTIMATION OF RIGID BODY MODES: SIMULATION AND EXPERIMENTAL STUDIES

4.1 Introduction

This chapter presents simulation and experimental studies for the rigid body estimator developed in the previous chapter. The purpose was to study the performance and to explore practical challenges of the estimator.

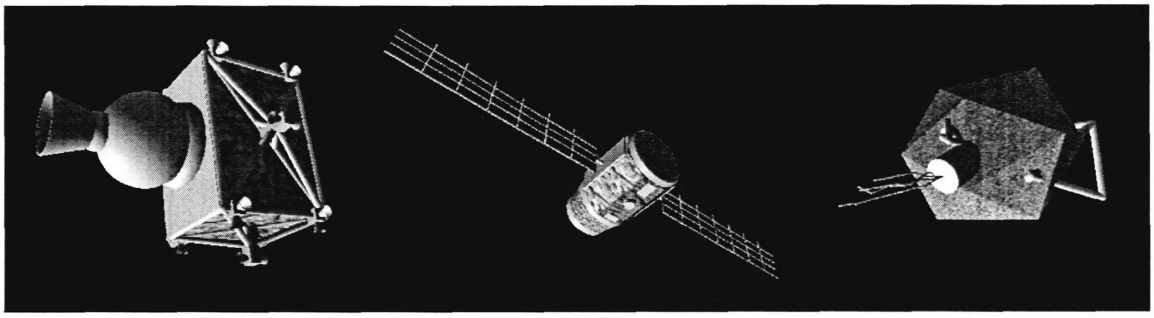
This chapter is divided into main two sections: simulation results (Section 4.2) and experimental results (Section 4.3). Within each section, results are presented for each of the three estimation modules: kinematic data fusion, Kalman filtering, and shape estimation.

4.2 Simulation Results

4.2.1 Simulation Environment

Computer simulations were used to predict theoretical performance of the estimator. Representative space objects were built into a virtual environment that was created in C++ using the OpenGL API [68] (see Figure 4.1). Rotational and translational target dynamics were accurately simulated using physics-based models. All model

parameters and initial conditions could be systematically varied to represent a wide range of scenarios.



(a) *(b)* *(c)*

Figure 4.1. Simulated space targets used in simulation studies: (a) satellite 1; (b) satellite 2; (c) space debris.

Simulated sensors were placed within this environment and their range image data was synthesized[1] (see Figure 4.2 and Figure 4.3). Sensors were given a field of view of 45 degrees and a resolution of 200 by 200 pixels. Typically, each sensor provided one to two thousand data points from the target at each sample time. Gaussian white noise[2] was added to the synthetic data, with standard deviation in the range measurement being anywhere between 0 and 10 percent of the measurement itself, depending on the study. Standard deviation in the focal plane measurements was set at one-half pixel for most studies. These values are similar to those found in practical sensors proposed for space applications [29, 84, 82].

[1] The algorithm for synthesizing range images is presented in Appendix B.

[2] The Gaussian distribution might not be a good model for some sensors. Estimator performance for non-Gaussian sensor noise was explored to some extent in the experimental studies.

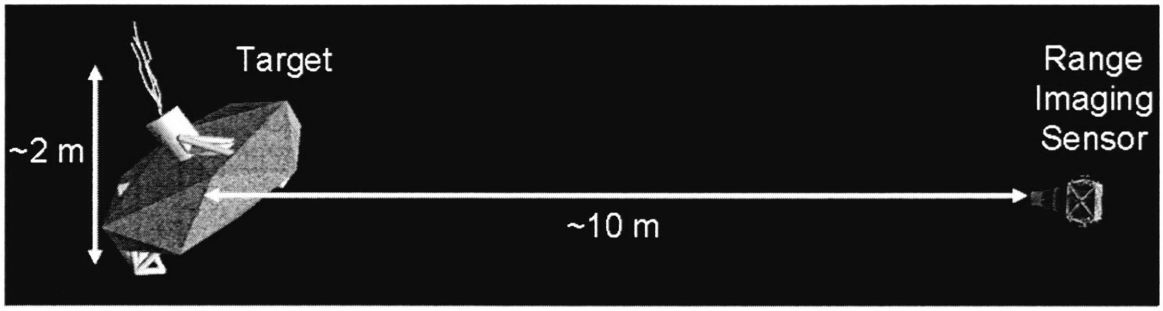


Figure 4.2. Simulated sensing environment.

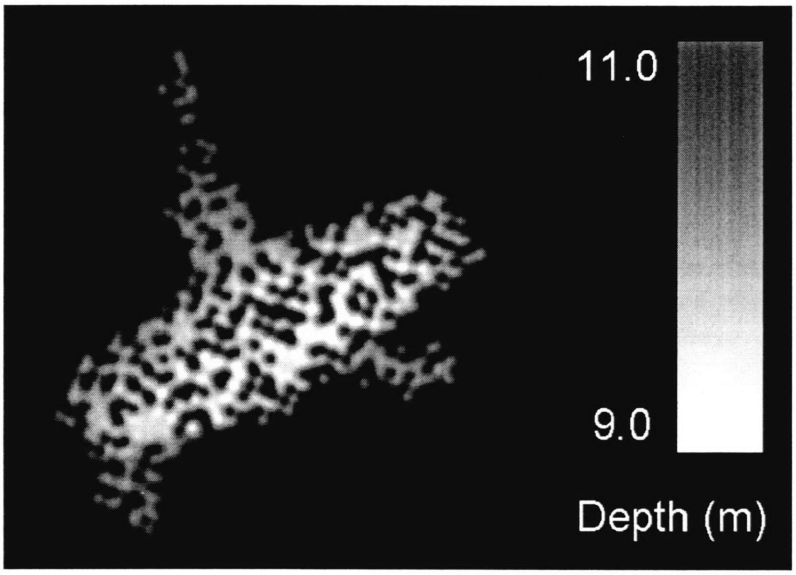


Figure 4.3. Synthetic range image.

Cooperative sensors were distributed uniformly about the target. Figure 4.4 shows the relative sensor positions used in simulation and experimental studies and the denoted names for these arrangements.

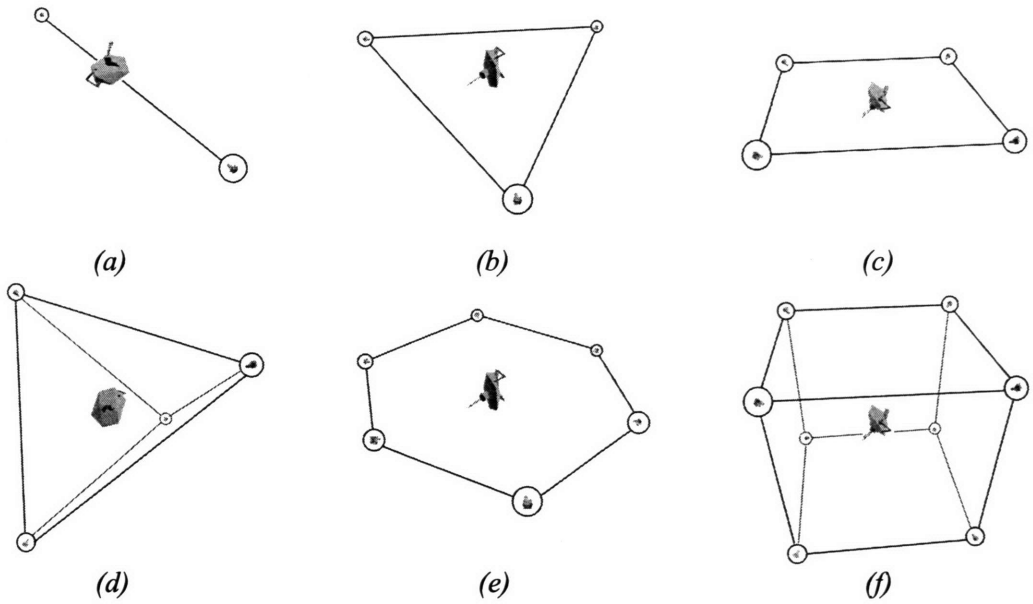
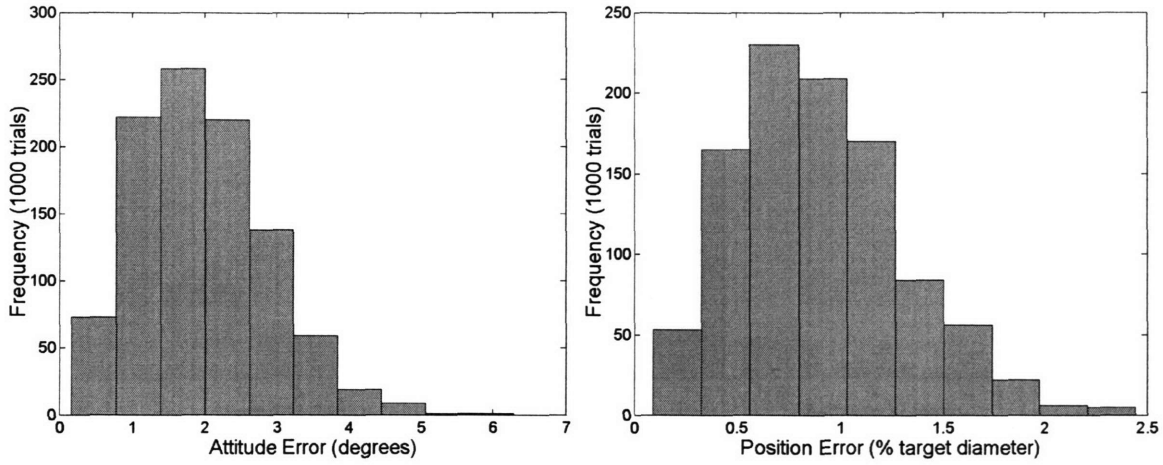


Figure 4.4. Sensor arrangements used in simulation and experimental studies: (a) 2-collinear; (b) 3-planar; (c) 4-planar; (d) 4-tetrahedral; (e) 6-planar; (f) 8-cubic. Sensors are circled and target is located in the center of the arrangement.

4.2.2 Kinematic Data Fusion

Monte Carlo simulations were used to characterize the errors in the kinematic data fusion algorithm (see Section 3.2.1). For these studies, a simulated target was placed in random poses, range data was synthesized, and the kinematic data fusion error was recorded. Figure 4.5 shows typical error distributions for attitude and position errors, using four sensors in a tetrahedral configuration with 10 percent noise in the range direction. These are approximately chi-square distributions with three degrees of freedom. Note that the errors are well-behaved, unimodal, and have very few outliers in simulation. The shape of the distributions was observed to be very similar for all simulated targets and sensor noise levels considered.

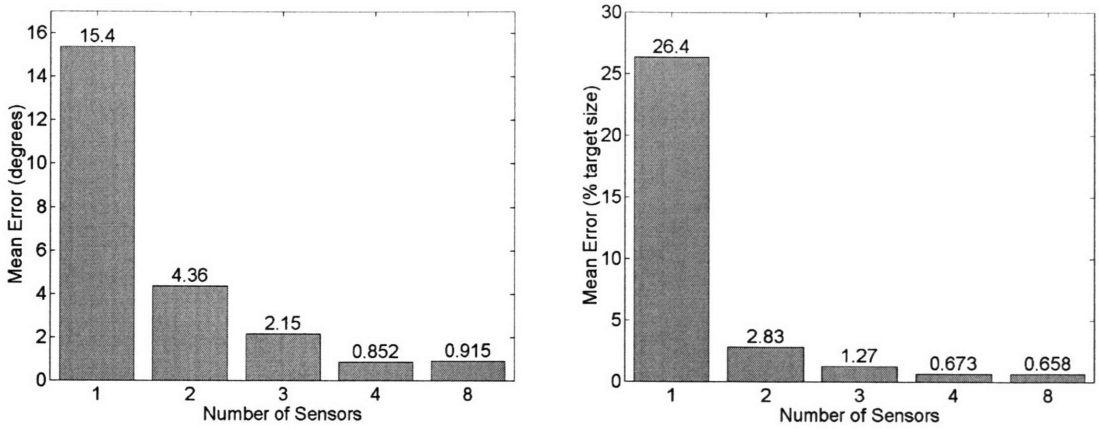


(a) *(b)*

Figure 4.5. Kinematic data fusion error distributions for satellite 1, 4-tetrahedral sensor arrangement, 10% range noise (simulation results): (a) attitude error; (b) position error.

At a systems level, it may be important to understand how estimator performance changes as a function of the number of robot team members used. Recall that the computationally simple kinematic data fusion algorithm presented in Section 3.2.1 assumed that the range image covers the entire target fairly uniformly. It would seem that decreasing the number of sensors might degrade the range image enough to invalidate this assumption.

A Monte Carlo simulation was used to quantify the relationship between number of sensors and kinematic data fusion performance. Sensors were distributed uniformly about the target using the arrangements shown in Figure 4.4. In order to estimate the maximum performance, no noise was added to range measurement. Figure 4.6 shows the mean attitude and position errors pertaining to the simulated *satellite 1* target shown in Figure 4.1a. Results for other simulated targets follow a similar trend and are provided in Table 4.1. Note the negligible performance gains when going from four sensors to eight.



(a) *(b)*

Figure 4.6. Kinematic data fusion errors vs. number of sensors, no sensor noise, satellite 1 (simulation results): (a) attitude error; (b) position error.

Table 4.1. Kinematic data fusion errors vs. number of sensors, no sensor noise (simulation results).

	Simulated target	Number of sensors / arrangement				
		1	2 collinear	3 planar	4 tetrahedral	8 cubic
Mean attitude error (°)	Satellite 1	15.4	4.36	2.15	0.85	0.91
	Satellite 2	4.75	1.12	0.75	0.63	0.69
	Debris	13.4	6.98	4.22	2.17	1.65
Mean position error (%)	Satellite 1	26.4	2.83	1.27	0.67	0.66
	Satellite 2	15.3	1.55	1.32	1.15	1.09
	Debris	18.3	1.79	0.95	0.72	0.79

The level of noise on the range images will also influence estimator performance. To quantify this relationship, a Monte Carlo simulation was created involving 11,000 trials. Standard deviation of the range noise was simulated between 0 and 10 percent of the range measurement itself. Position errors were observed to degrade only slightly as sensor noise increased. Attitude errors, by contrast, were much more sensitive to sensor noise. Figure 4.7 shows simulation results for the relationship between sensor noise and kinematic data fusion attitude errors, for each of the simulated targets shown in Figure 4.1. Note that in simulation, adding sensors seems to improve accuracy and reduce sensitivity to range image noise.

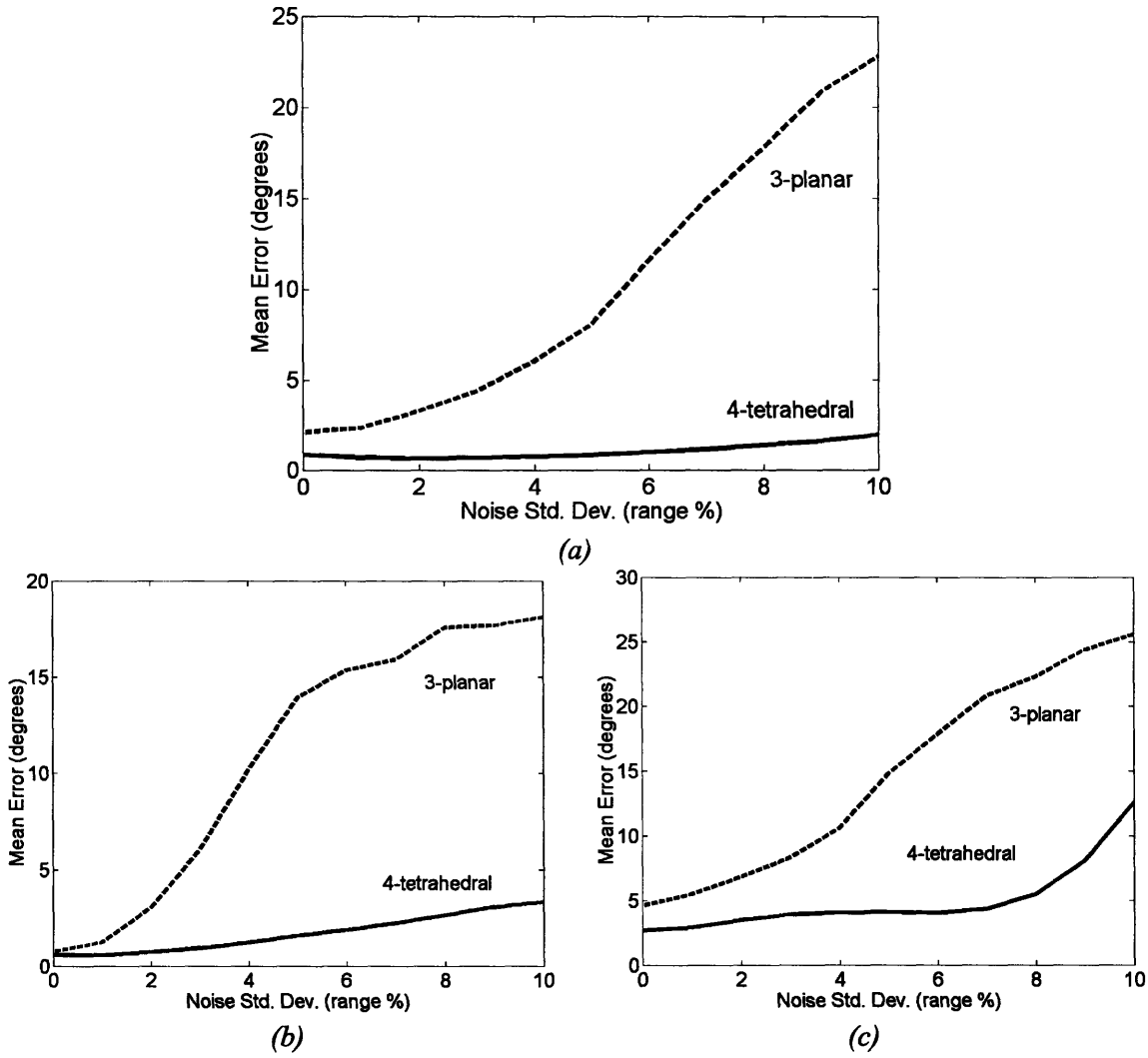


Figure 4.7. Kinematic data fusion attitude errors vs. sensory noise for various targets (simulation results): (a) satellite 1; (b) satellite 2; (c) debris.

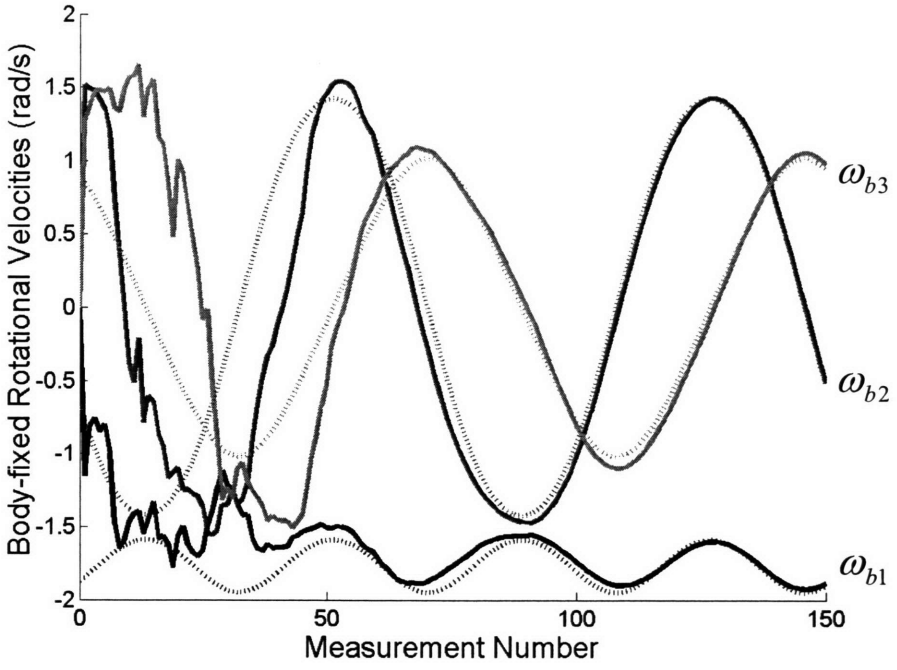
These simulation results suggest that sufficient accuracy is obtainable from the computationally simple algorithm presented in Section 3.2.1. These results also suggest that with multiple sensors, surrogate measurements are accurate to within a few degrees and a few percent of the target diameter, which is adequate for the Kalman filter. Further, these results suggest the diminishing returns of using more than four sensors. Noise in the sensors and the artifacts due to discrete sampling and voxelization mean that some error in the surrogate measurement is unavoidable. Finally, these results are a

reminder that the method described in Section 3.2.1 is inappropriate for use with a single sensor, as range data is heavily biased toward the single sensor and absent from the back side of the target. A different data fusion method would be needed if only one sensor were available.

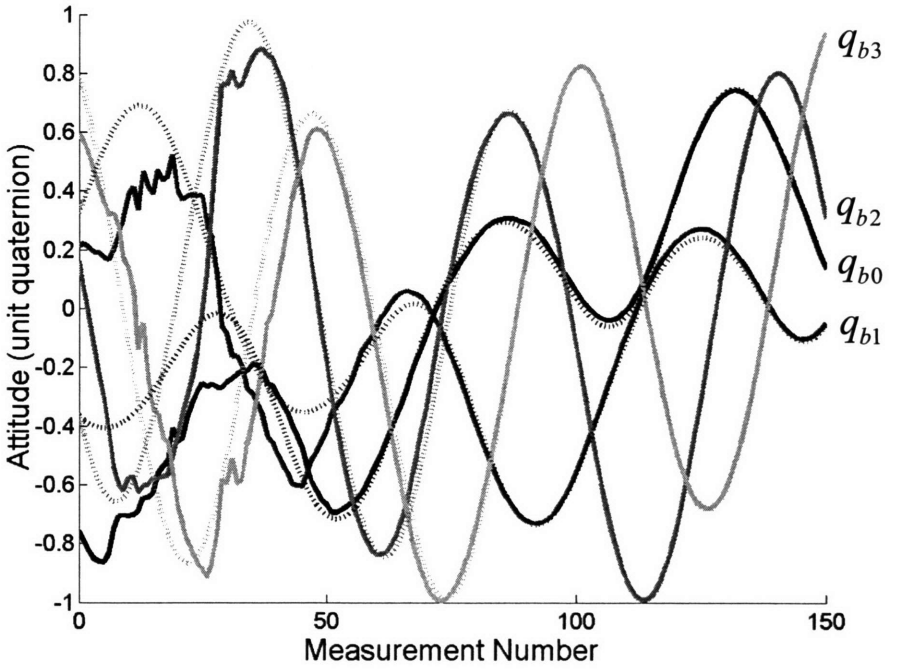
4.2.3 Kalman Filtering

The kinematic data fusion algorithm just discussed was used to provide surrogate measurements of target attitude and position to the Kalman filter. The kinematic data fusion used four sensors in a tetrahedral arrangement. Sensor noise was given a standard deviation of 1% in the range direction and 0.5 pixels in the focal plane direction. This setup provided surrogate measurements that were accurate to the order of a few degrees in attitude and a few percent of the target diameter in position.

The following figures show typical simulation results for state and parameter estimation for non-degenerate target motions. Dotted lines show the actual values of the state and parameters used in the simulation. Note that the dynamic state and parameters are estimated rather quickly, usually within one or two target rotations.

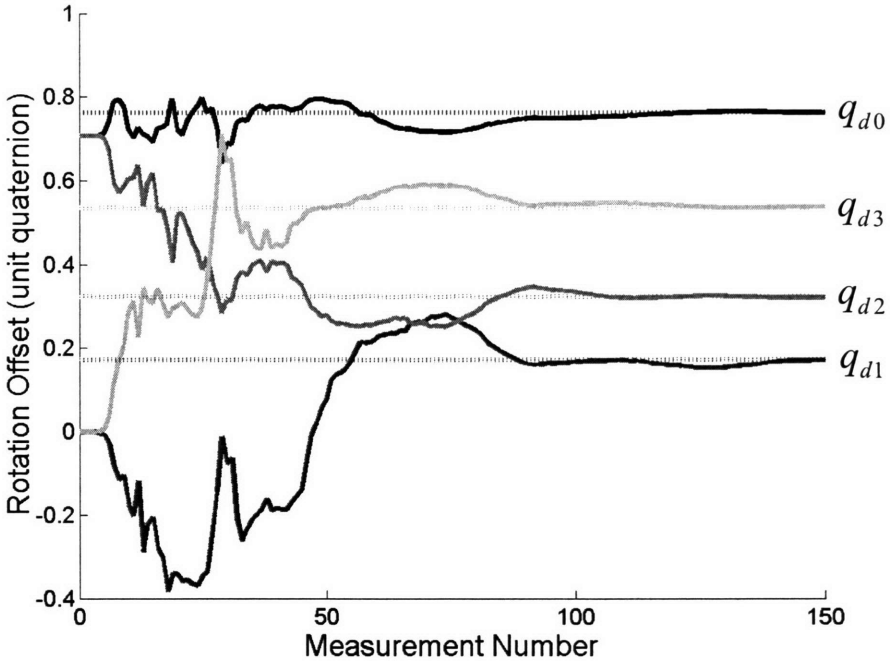


(a)

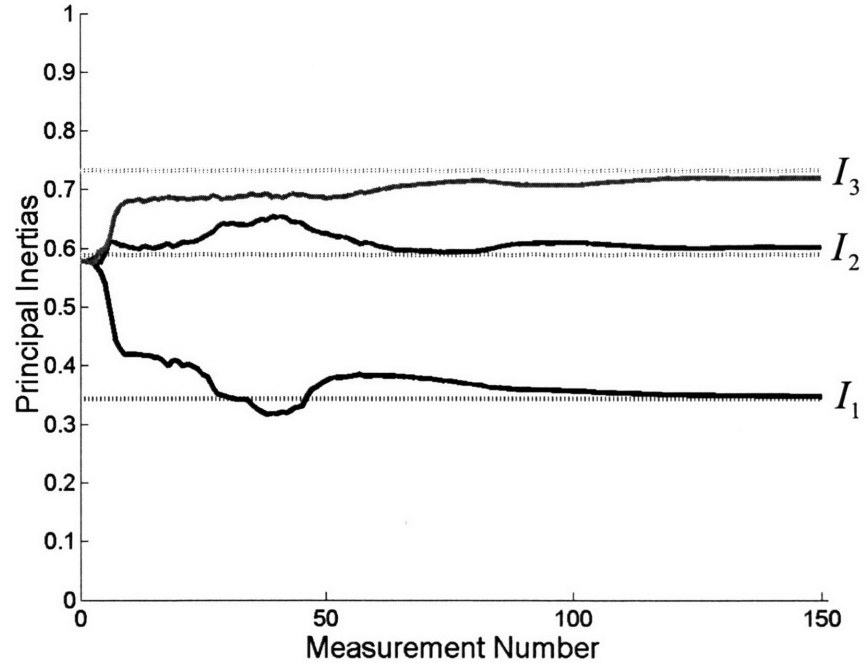


(b)

Figure 4.8. Rotational motion estimation (simulation results): (a) rotational velocity; (b) attitude quaternion. True values shown with dotted lines.

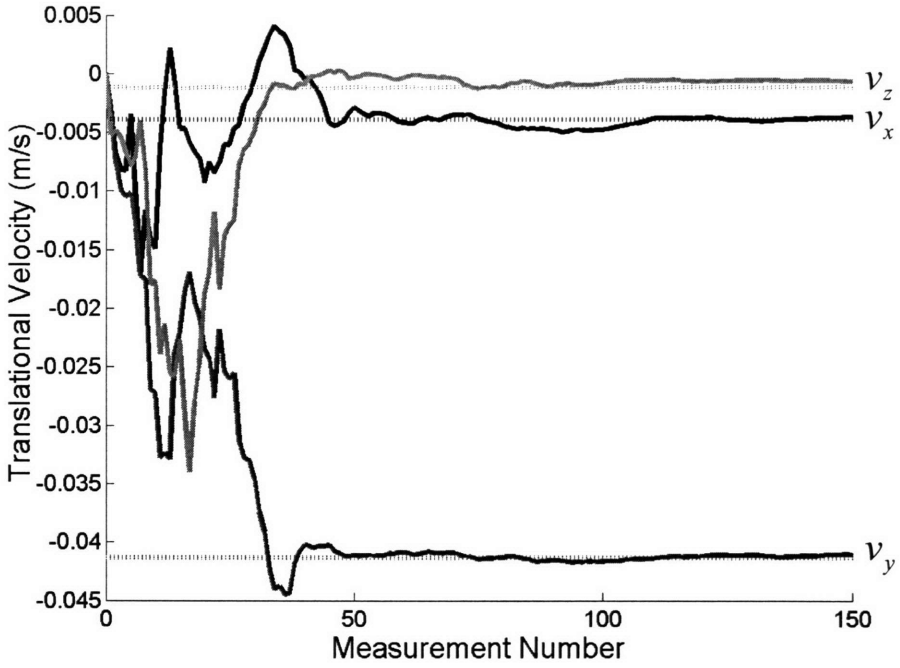


(a)

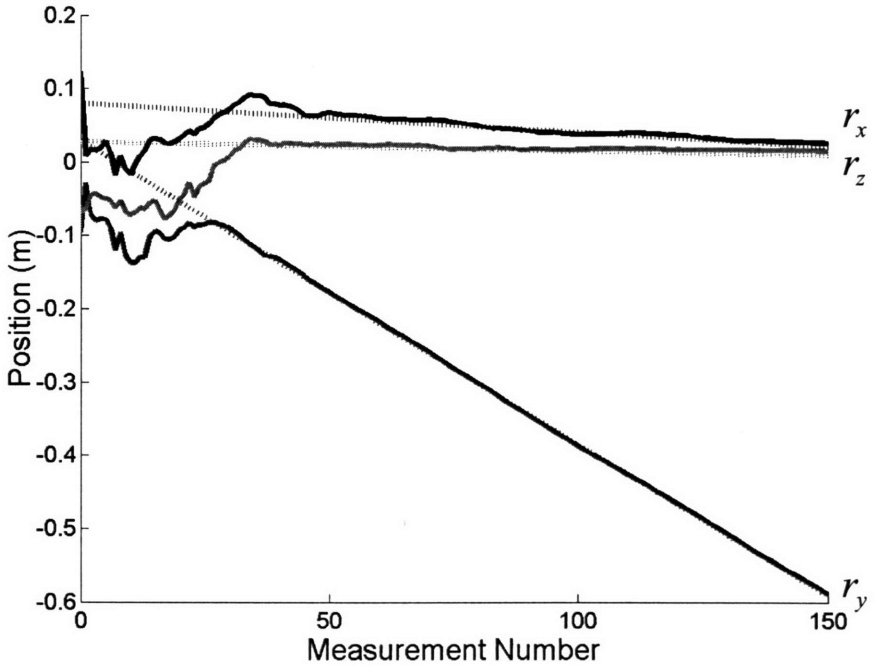


(b)

Figure 4.9. Rotational parameter estimation (simulation results): (a) rotational offset quaternion; (b) relative principal inertias. True values shown with dotted lines.



(a)



(b)

Figure 4.10. Translational motion estimation (simulation results): (a) velocity of center of mass; (b) position of center of mass. True values shown with dotted lines.

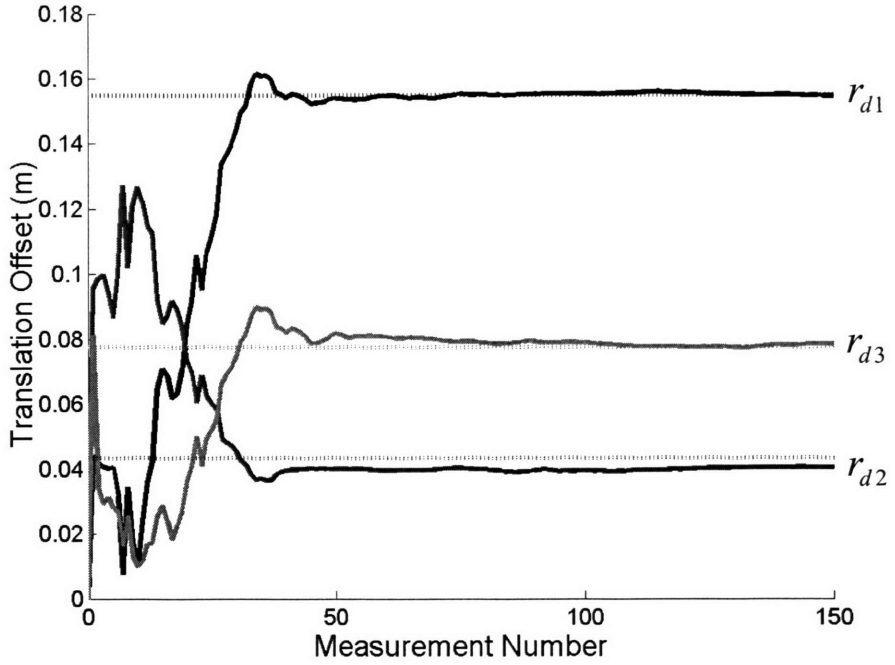


Figure 4.11. Translational parameter estimation (simulation results): translational offset. True values shown with dotted lines.

As discussed in Section 3.3.7, as the motion of the target approaches a degenerate condition (e.g. a single-axis spin), observability of the inertial parameters degrades. In other words, targets in a multi-axis tumble have rich information content embedded in their trajectory while flatly spinning targets have little. This loss of observability is manifested as sluggish convergence of the parameter estimates in the Kalman filter.

This degradation was studied using a Monte Carlo simulation involving 2500 trials. For each trial, the initial spin condition of the target was chosen randomly and the time to parameter convergence was recorded. Parameter convergence time was used as an indirect measure of observability and was defined as estimating the rotational offset parameter $\vec{q}_d$ to within 10 degrees and estimating the principal inertia vector $\vec{I}$ to within 3 degrees of the actual values. The principal inertias of the simulated target were set at $\{0.34 \quad 0.59 \quad 0.73\}^T$, a value well away from the inertia equality degeneracy. Initial angular momentum of the target was held constant among all trials so that rotation speed

would not factor into the convergence times. Again, four sensors in a tetrahedral arrangement were used, with 1% noise in the range direction.

Figure 4.12 is a scatter plot showing Kalman filter convergence time versus the spectrum of spin conditions of the target. Each point represents an individual trial from the Monte Carlo simulation. The spin condition is parameterized by the nutation angle, which is the angle between the angular momentum vector and the major principal inertial axis, at the instant of zero spin about the intermediate principal axis. A nutation angle of 0 degrees thus indicates a flat spin about the major principal axis, 90 degrees indicates a flat spin about the minor principal axis, and an intermediate angle indicates a multi-axis tumble. These results affirm that parameter estimation is slow near degenerate spin conditions and fast for targets in a multi-axis tumble. The filter could not converge in the allotted time for many trials in near-flat spins, as shown by the dark band of points along the top of the plot.

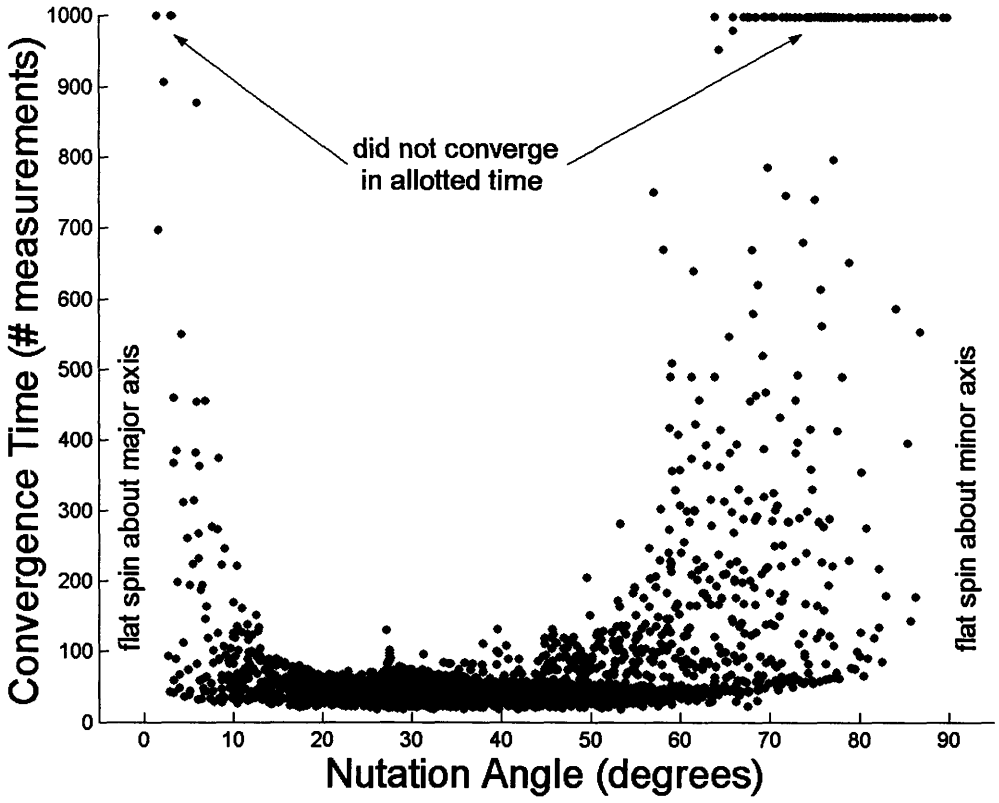


Figure 4.12. Inertia parameter convergence vs. the spectrum of target spin conditions (simulation results).

These results demonstrate the effectiveness of the Kalman filter in removing substantial noise from the surrogate measurements and in extracting the full dynamic state and parameters of the target. The results indicate that for non-degenerate targets, full estimation of parameters and dynamics can occur in just one or two target rotations. Further, the simulation results support the assertion made in Chapter 3 that near-degenerate motions provide poor parameter observability.

4.2.4 Shape Estimation

Simulation studies were used to qualitatively assess shape estimator performance. For these simulations, four sensors were used in a tetrahedral configuration. Figure 4.13 shows typical results for the shape estimation of the three simulated targets with sensor

noise of 1% in the range direction and 0.5 pixels in the focal plane direction. Note that for clarity, only voxels with a likelihood of occupancy greater than 0.5 are shown in the shape estimates.

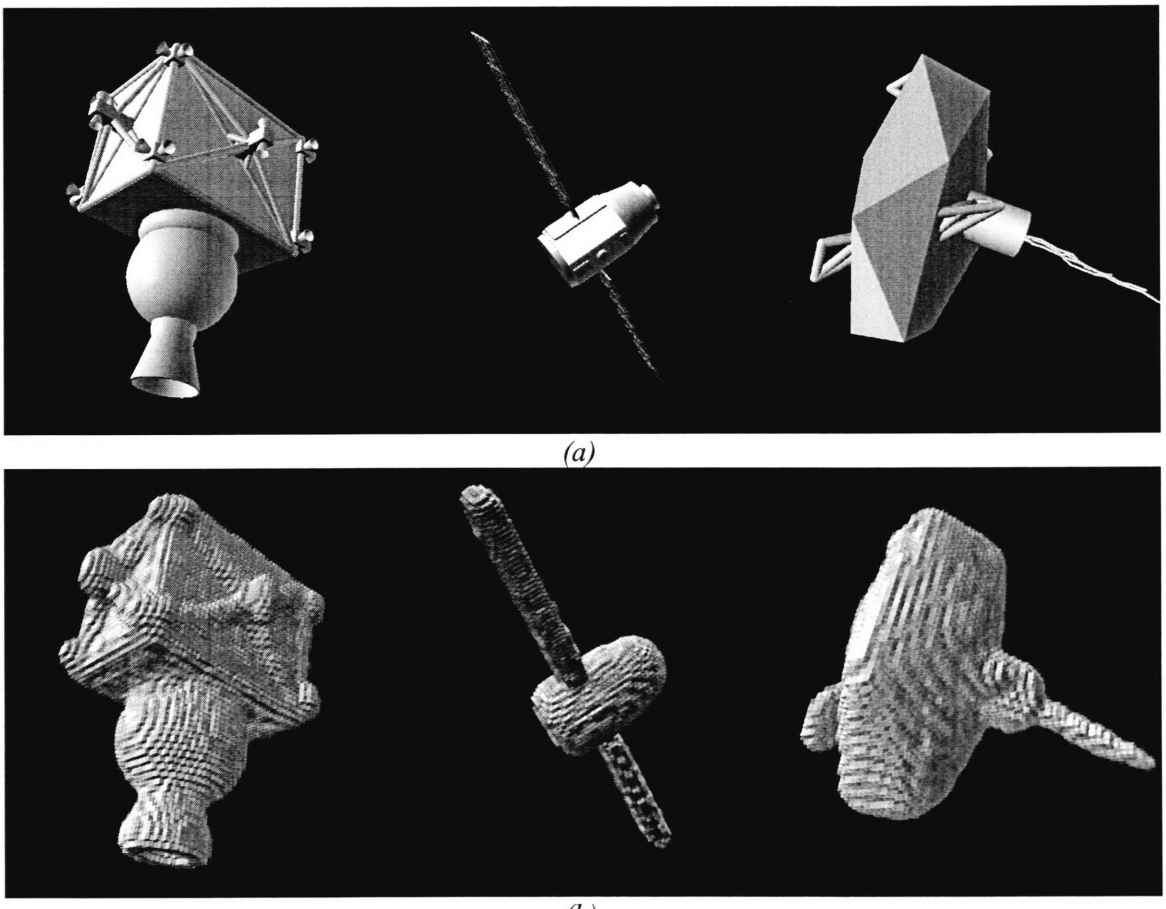
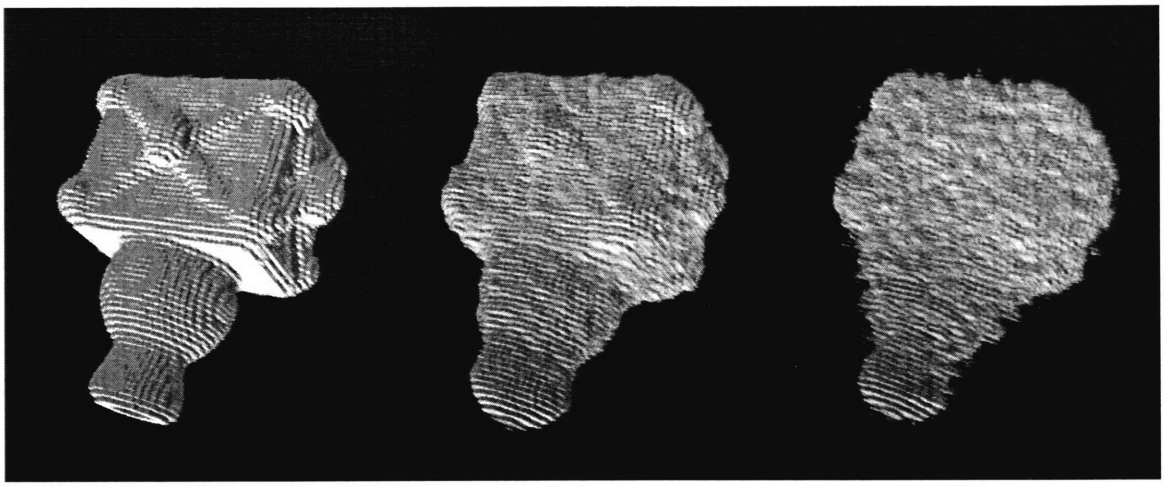


Figure 4.13. Shape estimation (simulation results): (a) simulated targets; (b) corresponding shape estimates with 1% sensor noise.

Figure 4.14 illustrates how shape estimation degrades as sensory noise increases. Sensor noise standard deviation in the range direction was specified at 1%, 3.2%, and 10%, respectively for the three images shown. Similar degradations were observed for the other simulated targets. Note that fine details become less certain as noise increases, but larger features of the target can still be mapped with reasonable certainty. These

simulation results suggest that accurate shape models can be built despite substantial sensory noise.



(a) (b) (c)

Figure 4.14. Shape estimation for different sensor noise levels (simulation results): (a) 1% sensor noise; (b) 3.2% sensor noise; (c) 10% sensor noise.

4.3 Experimental Results

4.3.1 Experimental Platform

After observing good estimator performance in simulation, a number of experimental studies were conducted to validate simulation results and to study practical challenges associated with the estimator. These studies used an experimental platform built by Dr. Vivek Sujan at the MIT Field and Space Robotics Laboratory. The platform consists of two serial-link spatial manipulators, two stereo camera heads, and a motion-controlled carriage, all of which can be mounted on a 4-foot-cube supporting structure (see Figure 4.15). Mock-ups of space targets can be mounted in the carriage and moved along one axis of translation and one axis of rotation. A movable spotlight emulates sunlight and a diffuse light box emulates earth albedo. The supporting structure is

wrapped in black felt to emulate the dark backdrop of space. All motion control and sensor processing is handled by a pair of 1-GHz Intel Pentium III computers, which communicate to each other using standard ethernet protocols. The computers interface to the manipulators and carriage using off-the-shelf motion control cards built by Servo-To-Go, Inc. [64]. The stereo heads are interfaced using IEEE 1394 firewire communication protocols.

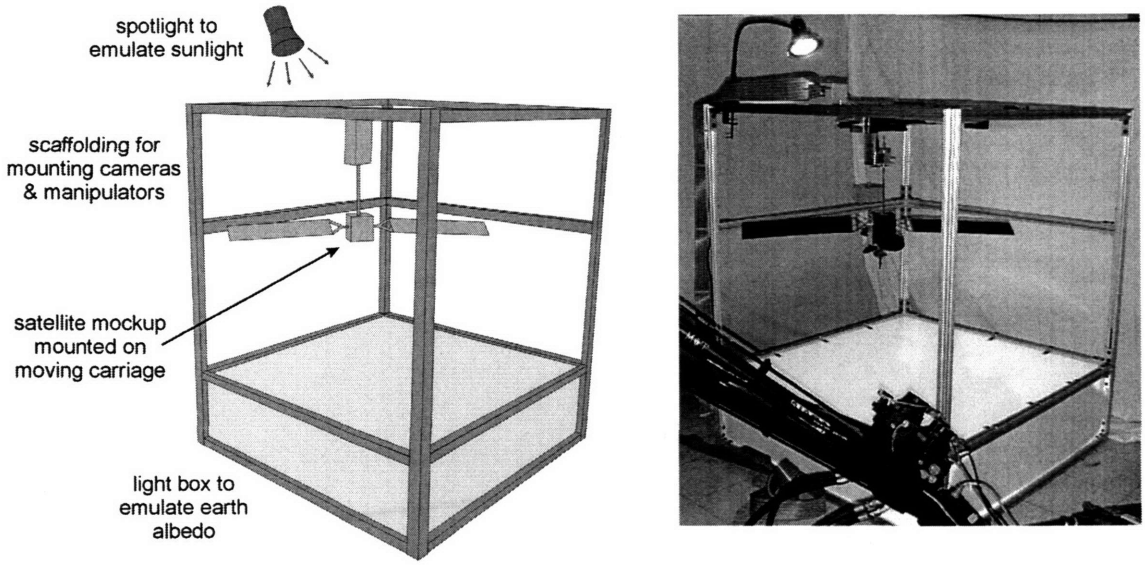


Figure 4.15. Field and Space Robotics Laboratory Experimental Testbed: (a) schematic; (b) photograph of actual system. Cameras and manipulators removed for clarity. Photo credit: V. Sujan.

The stereo camera heads, made by Videre Design [83], provide pairs of 2-D images to the central computers. 3-D range images are generated by processing the 2-D images using off-the-shelf software (SRI Small Vision System, v. 3.2b [74]). This system is able to capture 15 range images per second at a resolution of 320 by 240 pixels. Figure 4.16 shows a representative range image produced using this system. Note that like any stereo vision system, range data is noisy and may be missing from areas in

shadow or having little texture. Noise in the range direction is approximately 1 percent for ideally-textured surfaces; however this degrades by as much as an order of magnitude for poorly-textured and reflective surfaces. Non-Gaussian outliers typically comprise a small fraction (a few percent) of the entire range data cloud.

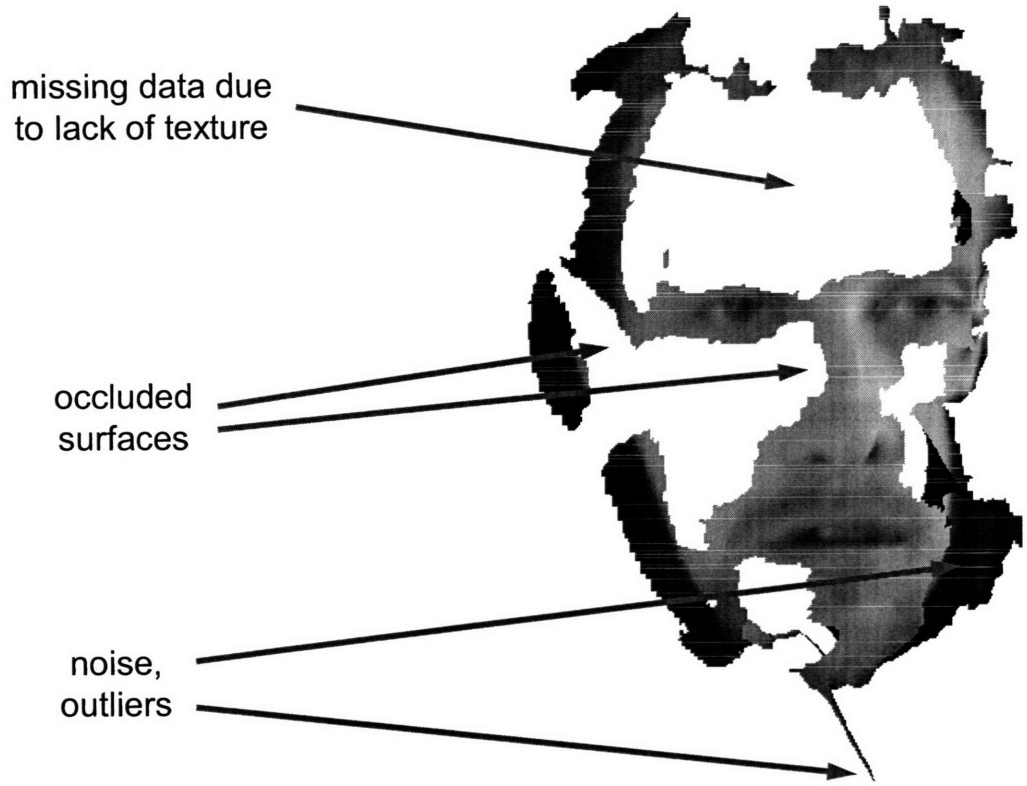
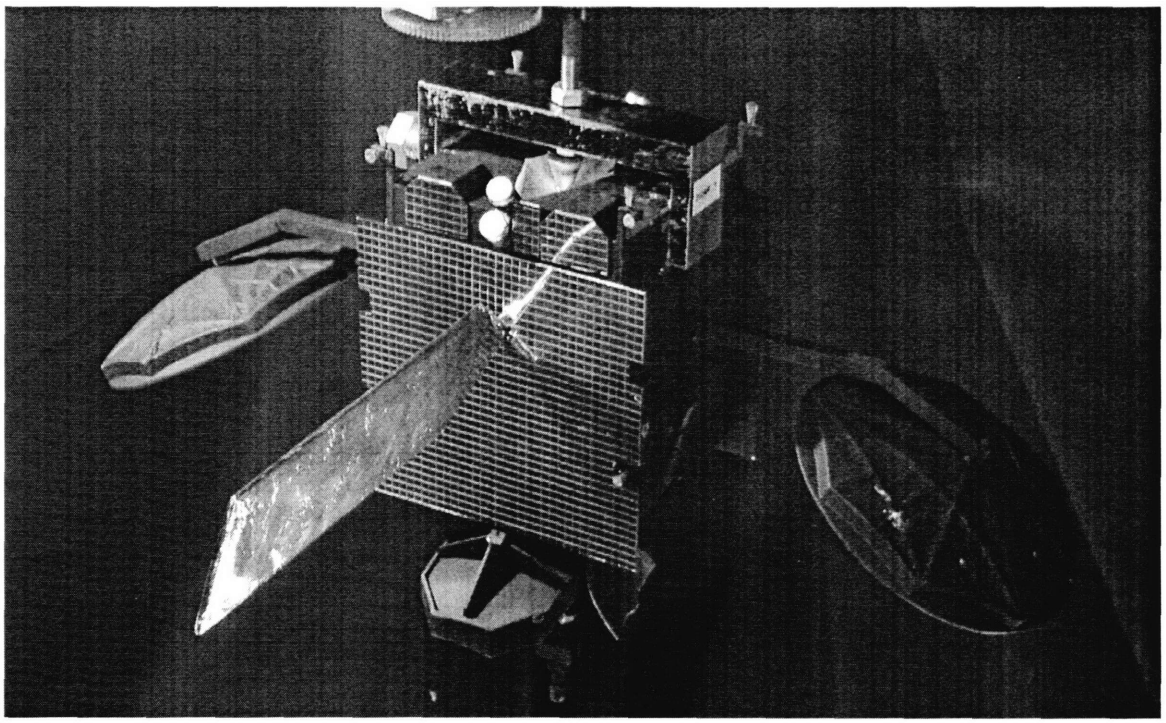
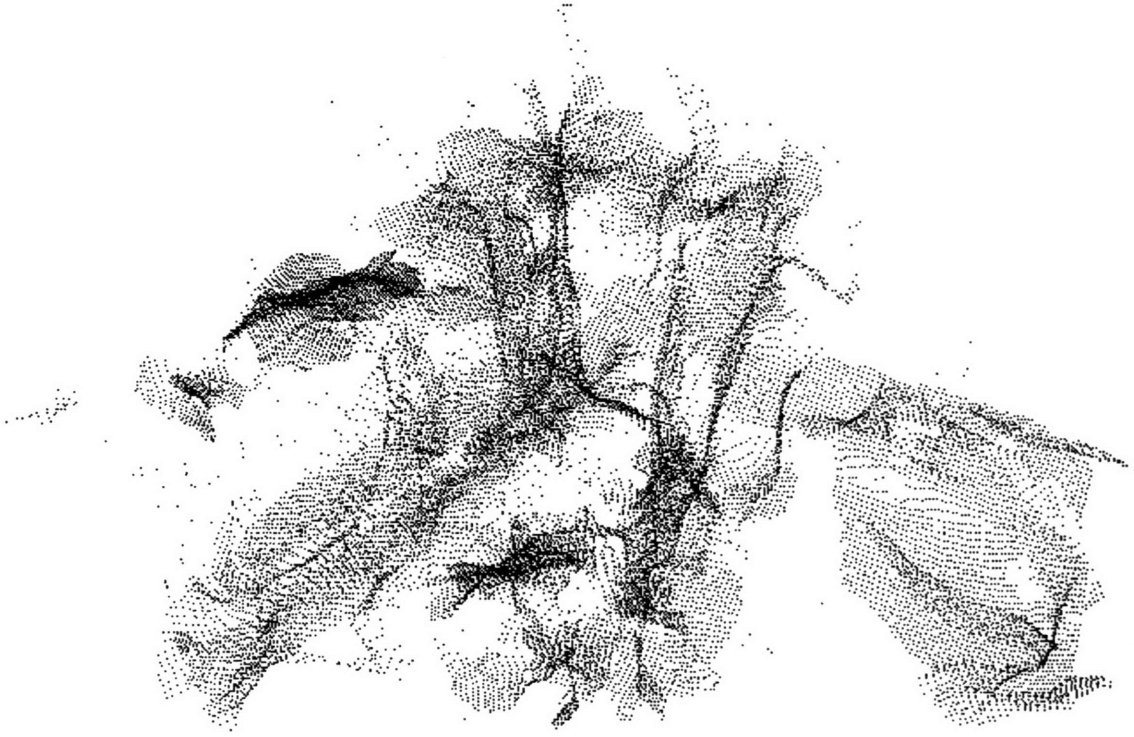


Figure 4.16. Range image of the author's face using stereo vision hardware.

A satellite mock-up was mounted in the moving carriage for the experimental trials described here (see Figure 4.17). The model was built by A&M Model Makers [1] and is a $1/24^{th}$ scale model of a Hughes 601 communications satellite. The two original solar panels from the model were removed and replaced by a single smaller one, so that the model could fit within the field of view of the stereo cameras.



(a)

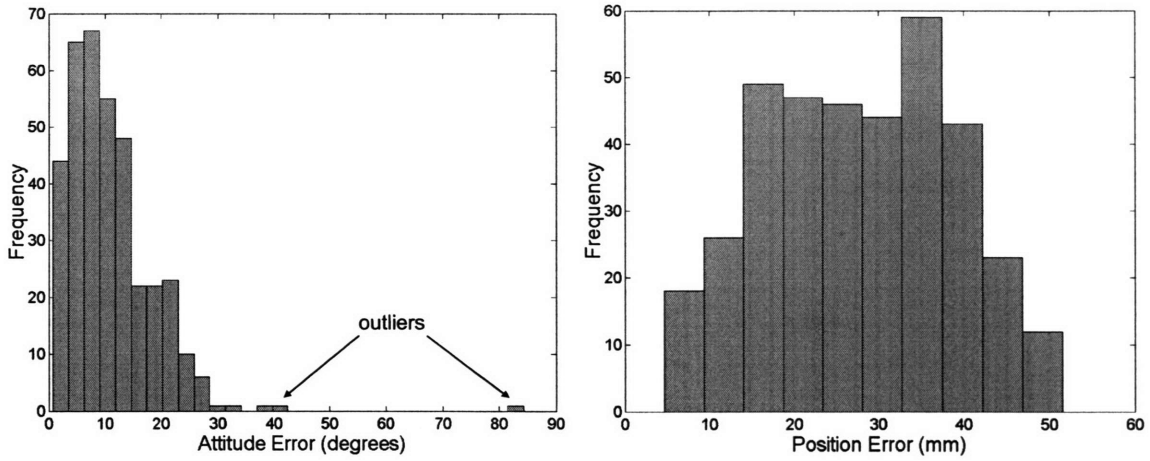


(b)

Figure 4.17. (a) Satellite mockup used in experimental studies; (b) example of range image provided by 4 sensors in tetrahedral arrangement.

4.3.2 Kinematic Data Fusion

As was done for the simulation studies, the error distributions for the kinematic data fusion algorithm were quantified, here using experimental data (see Figure 4.18). Note that the distribution shapes are similar to those observed in simulation (see Figure 4.5), although the variances are much higher with experimental data. In addition, there are a few outliers in the estimates, as seen in Figure 4.18a.



(a) *(b)*

Figure 4.18. Kinematic data fusion error distributions for the Hughes 601 satellite mock-up, 4-tetrahedral sensor arrangement (experimental results): (a) attitude error; (b) position error.

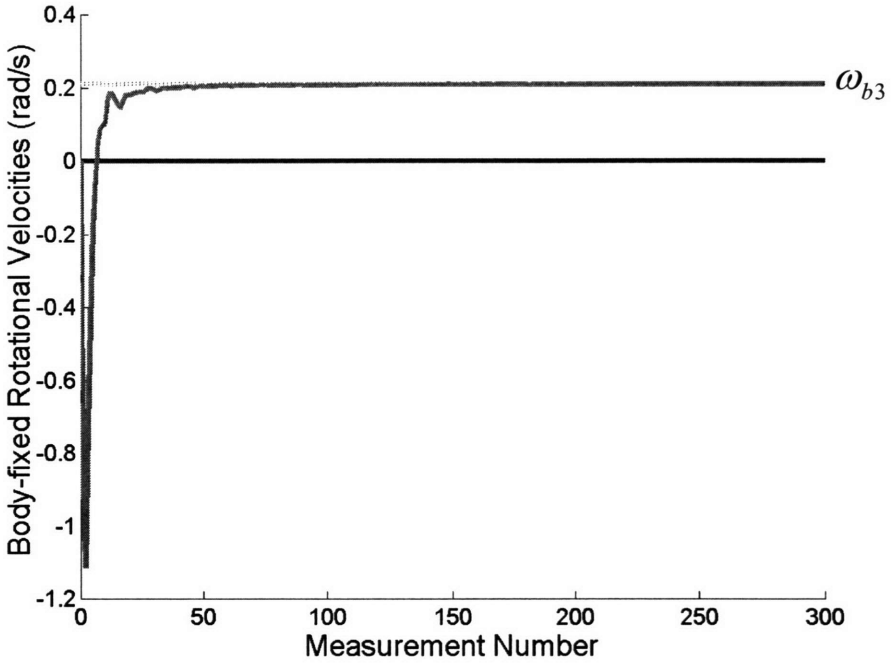
4.3.3 Kalman Filtering

The experimental platform is capable of providing only flat spin motions (i.e. no multi-axis tumbling). Consequently, inertial parameters were not observable for the experimental studies done here. Only the dynamics of the target, the location of the major principal inertial axis, and the shape were observable. The Kalman filter was modified to estimate only the observable quantities, which involved removing some elements of the state vector:

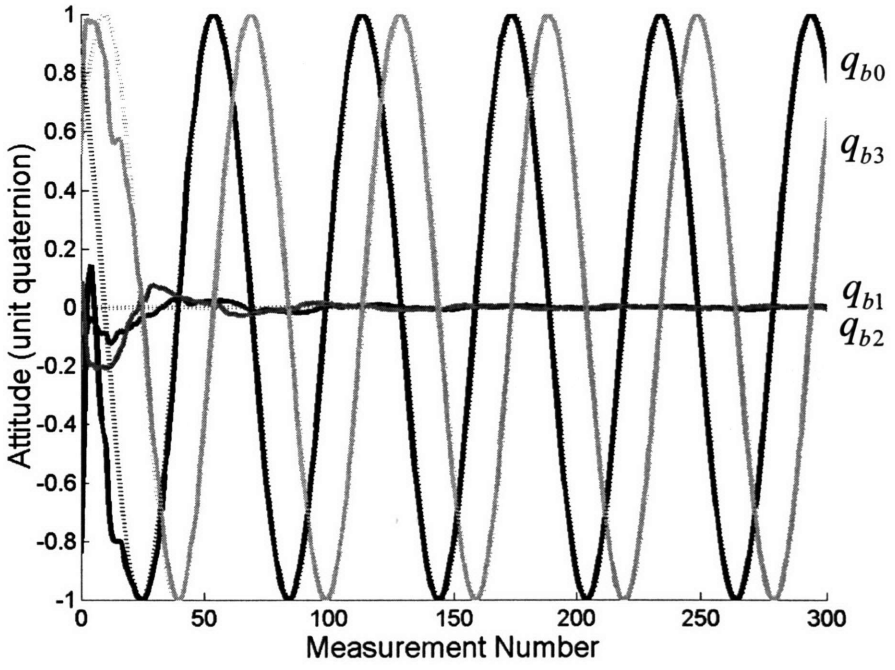
$$\vec{x}' \equiv \begin{Bmatrix} \omega_{b3} \\ \vec{q}_g \\ \vec{q}_d \end{Bmatrix}. \tag{4.1}$$

This is a truncated version of the more general state vector given by Equation (3.5).

The following figures show typical experimental results for state and parameter estimation. Like the simulation results, and despite the much higher noise in the surrogate measurements (the kinematic data fusion outputs), the state and observable parameters are estimated quickly, within one or two target rotations.

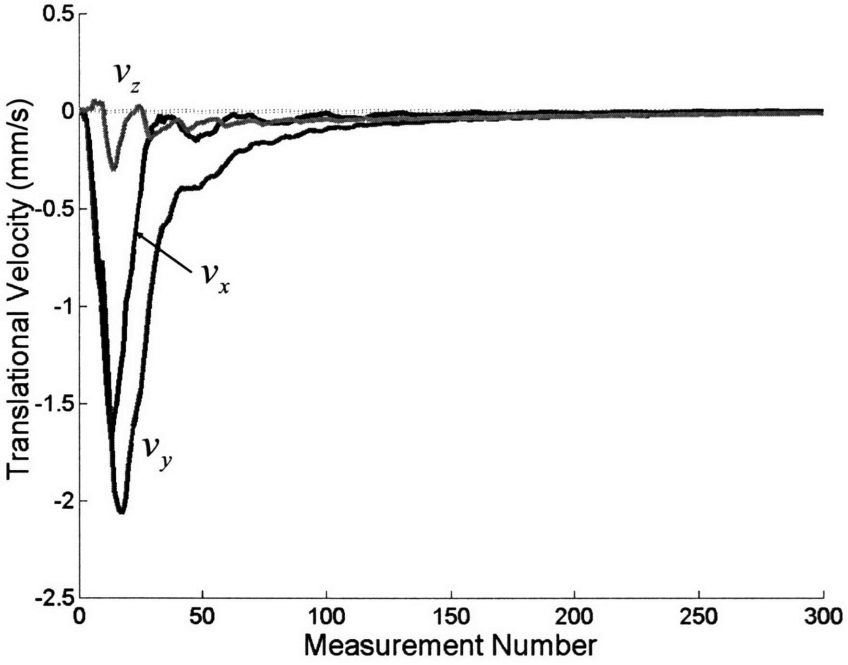


(a)

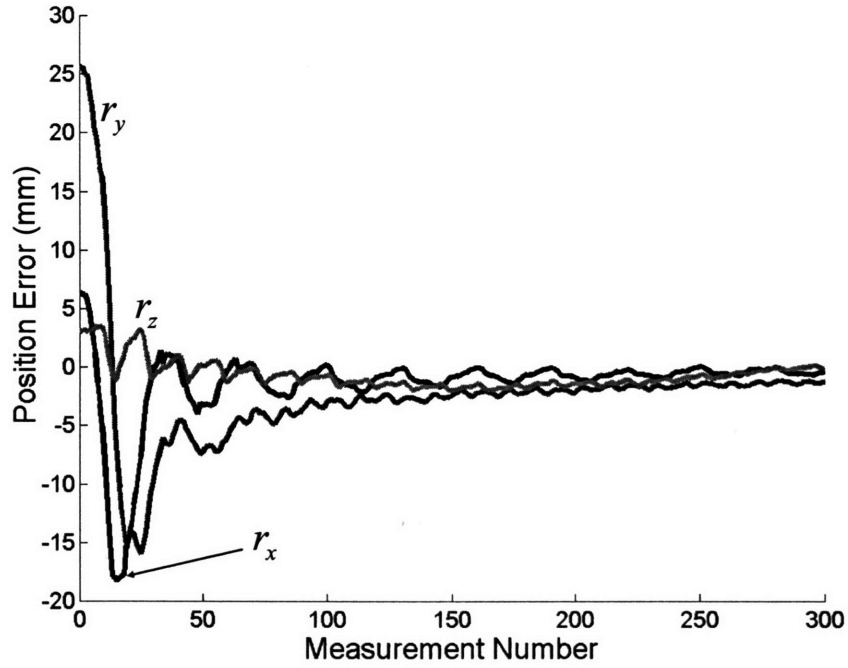


(b)

Figure 4.19. Rotational motion estimation for the Hughes 601 satellite mock-up (experimental results): (a) rotational velocity; (b) attitude quaternion. True values shown with dotted lines.

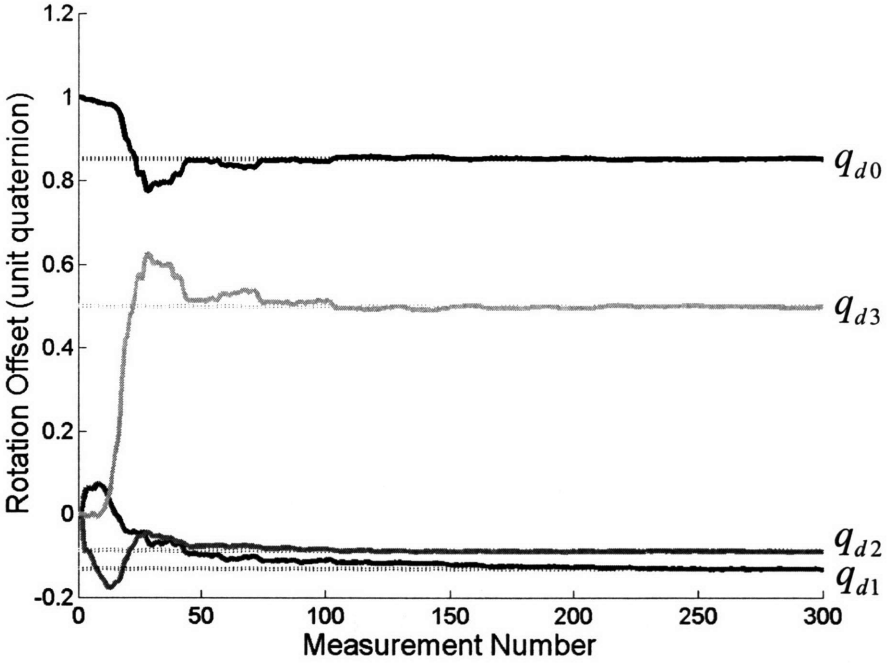


(a)

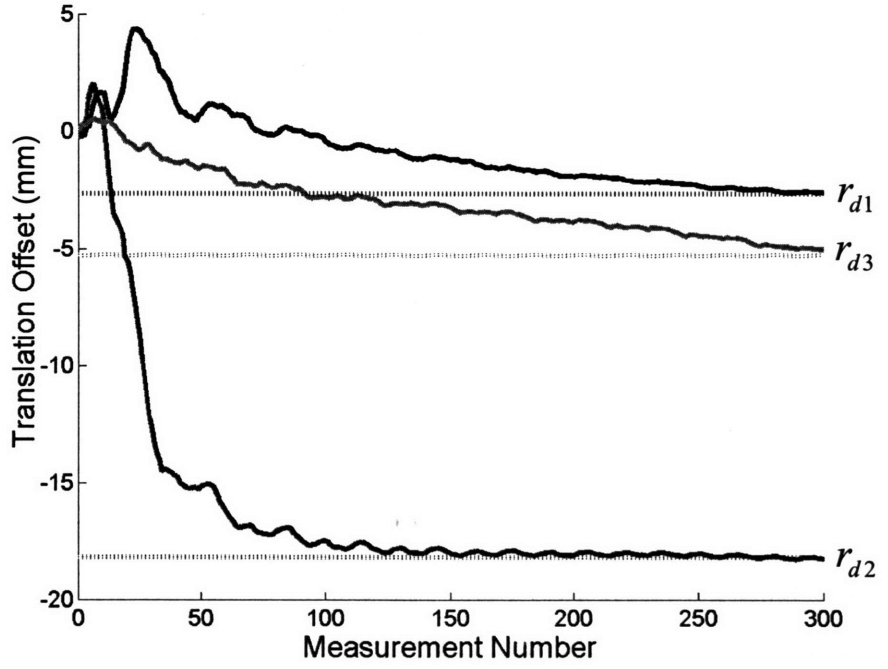


(b)

Figure 4.20. Translational motion estimation (experimental results): (a) translational velocity; (b) position. True values shown with dotted lines.



(a)

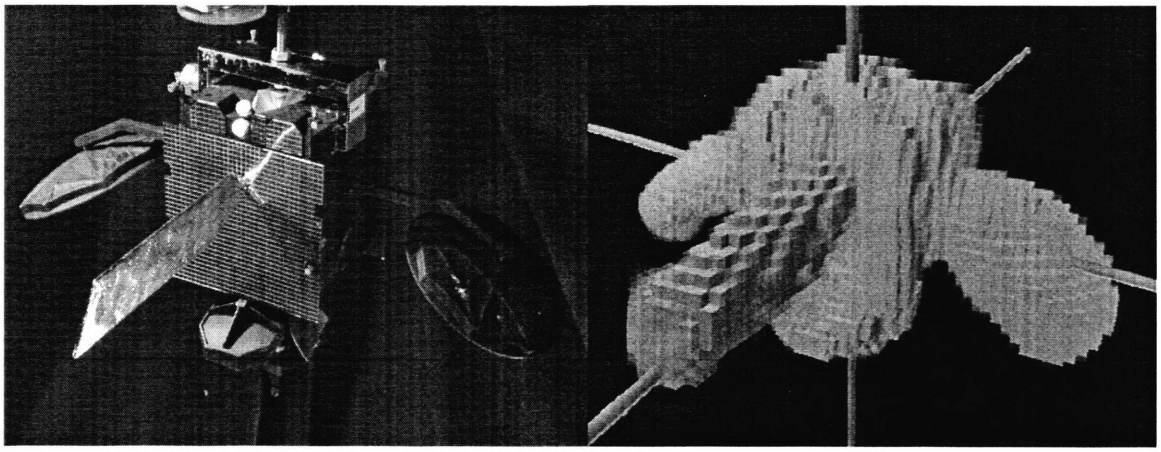


(b)

Figure 4.21. Parameter estimation (experimental results): (a) rotational offset quaternion; (b) translational offset. True values shown with dotted lines.

4.3.4 Shape Estimation

Figure 4.22 shows typical experimental shape estimates for the Hughes 601 satellite mock-up. Note the relatively accurate estimate despite large and erratic sensory noise (see Figure 4.17b). Most of the large features of the target are accurately depicted, including the reflective solar panels.



(a) *(b)*

Figure 4.22. Typical shape estimate for the Hughes 601 mock-up (experimental results): (a) actual target; (b) shape estimate.

4.3.5 Overall Performance vs. Number of Sensors

Experiments were conducted to explore estimator convergence as a function of the number and arrangement of sensors. As suggested by the simulation studies, the number of sensors affects accuracy of the kinematic data fusion algorithm, which in turn affects speed of convergence in the Kalman filter. The following studies will therefore evaluate the combined performance of the kinematic data fusion and the Kalman filter as a function of sensor arrangements.

Since orientation of the angular velocity vector was observed to be the slowest estimate to converge in experimental trials, it was used to quantify estimator performance. A series of studies were performed in which estimation error in the angular

velocity vector was recorded after the target made two full rotations. Three hundred sixty distinct trials were used for each sensor arrangement. Each trial was a combination of one of sixty initial target poses and one of six possible sensor sample rates. The sensors were allowed to sample at 10, 12, 15, 20, 30, and 60 samples per target rotation.

Figure 4.23 shows distributions of the estimation error for various sensor arrangements. These distributions approximately follow a chi-square distribution with two degrees of freedom. Performance was observed to vary little as a function of the sensor rates used. Like the simulation studies, these experiments suggest that little performance is gained using more than four sensors. Note that the estimator performed poorly when using only two sensors, which is likely due to poor kinematic data fusion.

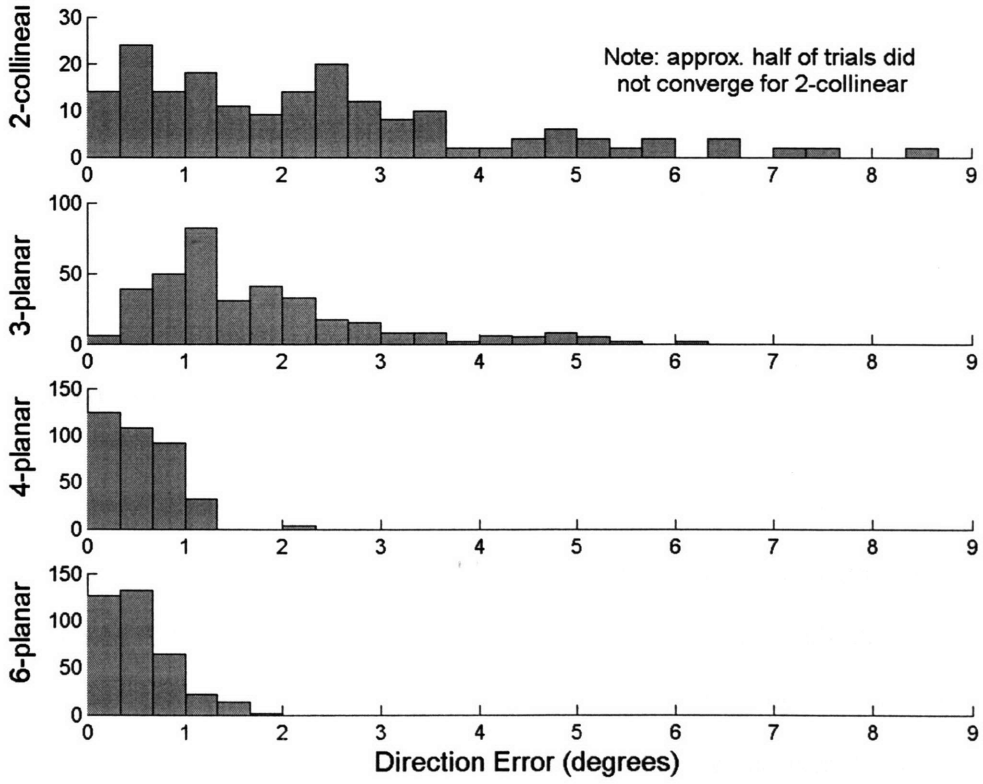


Figure 4.23. Rotational velocity estimation error distributions after two target rotations, for sensor arrangements defined in Figure 4.4 (experimental results).

4.3.6 Computation Time

A major consideration for this application is that computational resources are very limited (see Section 2.1). Here, experiments and simulations were performed using a 1-GHz Intel Pentium III processor. On this system, the estimator consistently operated at a rate above 5.0 Hz. That is, the estimator required less than 0.2 seconds to perform kinematic data fusion on the range image, filter the results in the Kalman filter, and update the shape estimate[1]. Substantially higher speeds could probably be attained with further code optimization.

Space-qualified computation power might be an order of magnitude slower than that used here. Recall that the estimator must sample at a rate fast enough to avoid aliasing the rotational motions, which means that most practical applications would require a sensor sampling rate on the order of 0.1 to 1.0 Hz for targets spinning at rates on the order of 3 rpm. The results of simulations and experiments done here indicate that even with space-qualified hardware, the computational requirements of the estimator are manageable.

4.4 Summary

Chapters 3 and 4 have developed and demonstrated a methodology for the estimation of rigid-body motions, dynamic model parameters, and geometric shape for targets in space. The method exploits two key features of the applications discussed here, which are that the dynamics of targets are well-modeled in space, and that several cooperative sensors may be available with which to observe the target. These features

[1] This time does not include the sensors' generation of the range images, since this is implementation dependent and would be a function of the type of sensors used. The results pertain strictly to the estimation algorithms developed in this thesis.

have enabled the design of a computationally efficient and robust estimator using the architecture introduced in Chapter 2.

Chapter 3 developed the three main components of the estimator. The first component, referred to as kinematic data fusion, condenses detailed range image clouds into coarse estimates of target attitude and position. This component discretizes a range image cloud into voxels, and performs simple computations on the voxel set to determine the approximate principal geometric axes and centroid of the target. A Kalman filter was then developed to filter these coarse estimates and extract the full dynamic state and model parameters of the target. Because the rotational dynamics are nonlinear, the Kalman filter must be designed carefully. Finally, a simple method was described to illustrate the process of recursive shape estimation, which simply involves the stochastic mapping of a static scene.

This chapter has described a number of simulation and experimental studies designed to evaluate the accuracy, efficiency, and practicality of the estimator. Several different simulation and experimental studies were conducted to characterize estimation errors in the kinematic data fusion module. These studies suggested that the algorithm can provide sufficiently accurate surrogate measurements over a wide range of sensor and target conditions, although the simple design is not suited to situations involving less than three sensors. Simulation and experimental studies were then used to evaluate the Kalman filter, which was observed to perform very well. As expected, the studies also demonstrated the loss of parameter observability for targets undergoing degenerate motions. Shape estimation was observed to perform well in both simulation and experimental studies. Finally, the computational requirements of the estimator were quantified and shown to be quite reasonable for real-time implementation in practical space systems.

Modularity is one of the chief benefits of this estimation architecture. Certainly, some of the components of the estimator could be amended to deal with different scenarios. A more sophisticated shape estimator could be used instead of the very simple one illustrated here. Likewise, a different kinematic data fusion algorithm could be designed to handle situations involving only one sensor. Indeed, this could even incorporate more traditional machine vision methods if so desired. In sum, this chapter has demonstrated potential for the estimation architecture as a whole as much as it has the individual components.

CHAPTER

5

ESTIMATION OF VIBRATION MODES: THEORETICAL DEVELOPMENT

5.1 Problem Statement

Future space missions are expected to use autonomous robotic systems for the assembly, inspection, and maintenance of large space structures in orbit [7, 62, 75, 80]. Examples of such structures include the International Space Station, large synthetic aperture telescopes, and space solar power systems [52, 61]. For these missions, it is critical for the robotic systems to understand and predict the dynamics of the structures with which they interact. This information is often unavailable a priori, so accurate sensing and estimation of target dynamics and model parameters is expected to be a fundamental challenge for these missions. In the context of large flexible space targets, the general estimation problem of this thesis is stated as follows:

> *Estimate the shape and motion of a region of interest on a flexible space structure, using data gathered from one or more 3-D range imaging sensors.*

5.1.1 Structure Assumptions

The structural dynamics are assumed to be linear or weakly nonlinear. The structure's mode shapes are assumed to be well-known to the estimator a priori. In

practice, these could be found using techniques such as finite element analysis and could be computed off-line from a ground station. This modal knowledge in the estimator is assumed to be updated whenever the fundamental mode shapes change (e.g. due to a structural configuration change, added mass, etc.).

A priori uncertainty in the modal frequencies (time domain) is assumed to be bounded. This bound might be on the order of $\pm$ 20 percent of the true frequencies. The estimator should be able to accommodate this uncertainty and estimate the actual modal frequencies.

Rigid body modes are not treated here. In principle, the rigid and flexible modes could be estimated in a decoupled manner, with rigid modes estimated using techniques such as the one described in Chapter 3.

Thermal deformations are expected to be well-known or negligible. Here, this type of slowly varying deformation is treated as a quasi-static change in the equilibrium state of the structure. The estimator will only be concerned with vibrations about this equilibrium state. Thermally-induced forcing of the structure is not explored here.

Explicitly known external disturbances (e.g. robot manipulation forces) are not discussed in detail here, although in principle they could be included in the process model of the Kalman filter, described later.

5.1.2 Sensor Assumptions

To reiterate from Section 2.1, all data in the point cloud is assumed to be captured at approximately the same instant in time. In the context of vibration estimation, this means that the time delay between the first and last data point captured in the cloud is assumed to be small compared to the period of the highest natural frequency estimated.

An additional assumption for the vibration estimation case is that sensor noise is additive, white, unbiased, and roughly Gaussian. The estimator to be developed in this

chapter will be near-optimal in a minimum least square error sense if this assumption is valid.

5.1.3 Notation

Variables denoted with a subscript *i*, *j*, or *k* are considered elements of a vector. The same variable name without the subscript indicates the entire vector quantity. If the actual value of some quantity is represented by the variable *a*, then its estimated value is denoted $\hat{a}$ and its measured value is denoted $\bar{a}$.

Let the natural mode shapes of the structure be denoted as $\Phi_i(x)$ for each mode *i*. For a linear elastic system, the dynamic response $z(x,t)$ of the structural deformations is a separable function of time and space, and can be written as

$$z(x,t) = \sum_{i=1}^{m} A_i(t)\Phi_i(x) = A(t)^T \Phi(x) \tag{5.1}$$

where $z(x,t)$ is the deflection from the structure's equilibrium state, *m* is the number of modes excited in the response, and $A_i(t)$ is the i^{th} modal coefficient, which oscillates sinusoidally with the i^{th} modal frequency ω_i. If modal damping exists, these sinusoids decay exponentially with rate α_i.

5.1.4 Solution Approach

The goal is to estimate the time domain functions $A_i(t)$ for all modes of interest. This will reduce shape estimation (Section 5.4) to simply a modal reconstruction using the estimates of $A_i(t)$ and the a priori knowledge of mode shapes $\Phi_i(x)$.

Estimation of $A_i(t)$ will occur in two steps. First, modal decomposition in the spatial domain will be performed on the point cloud to arrive at a coarse estimate $\hat{\hat{A}}_i(t)$ (Section 5.2). This coarse estimate will then be filtered in the time domain using a

Kalman filter to arrive at a refined estimate $\hat{A}_i(t)$ (Section 5.3). Here, the number of hats on the variable is used to indicate the coarseness of the estimate. This chapter will now present these methods

5.2 Kinematic Data Fusion: Modal Decomposition

As shown in Figure 2.4 (page 26), the first step in the estimation process involves kinematic data fusion, in which noisy pixel-level information from the vision sensors is condensed into coarse estimates of the target's kinematic properties. In the context of vibrating space structures, this amounts to performing a modal decomposition on the visual data to find coarse estimates of the modal coefficients $A_i(t)$.

To begin, define an *inner product* (dot product) over some space X as

$$\langle a,b\rangle_X \equiv \int_X a(x)b^T(x)dx = \int_X \begin{bmatrix} a_1(x)b_1(x) & a_1(x)b_2(x) & \cdots \\ a_2(x)b_1(x) & a_2(x)b_2(x) & \\ \vdots & & \ddots \end{bmatrix} dx .$$

Let the space X be the "backbone" surface of the target structure. For example, if the structure of interest is a planar sheet of uniform thickness, then the space X is the 2-D reference surface embedded in the sheet at its equilibrium configuration, and $z(x,t)$ represents the deformation normal to the surface at some location x in X. One useful property of mode shapes is that

$$\langle \Phi_i, \Phi_j \rangle_X = \begin{cases} b_i & i = j \\ 0 & i \neq j \end{cases} .$$

That is, the mode shapes are orthogonal in the space X. If the mode shapes are normalized, then $b_i = 1$ for all i. For generality, normality will not be enforced here.

Consider a subspace $Y \subset X$, which represents a discrete, not necessarily uniform sampling of the space X (see Figure 5.1). The subspace Y represents the *sample space* in which the measurements lie, and may be changing in time.

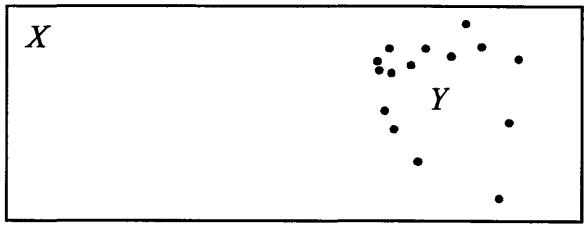


Figure 5.1. Sample space Y in complete space X.

In this discrete space, the inner product integral reduces to a summation, and the inner products of the mode shapes over the sample space Y are computed as:

$$\langle \Phi_i, \Phi_j \rangle_Y = \sum_{k=1}^{n} \Phi_i(y_k)\Phi_j(y_k) \equiv \beta_{ij}$$

where y_k is the k^{th} discrete point in the sample space Y and n is the number of discrete points in the sample space.

Define a symmetric *modal correlation matrix* M_Y to describe the inner products of the mode shapes in the sample space Y, for the m excited modes:

$$M_Y \equiv \langle \Phi, \Phi \rangle_Y = \begin{bmatrix} \langle \Phi_1, \Phi_1 \rangle_Y & \cdots & \langle \Phi_1, \Phi_i \rangle_Y & \cdots & \langle \Phi_1, \Phi_m \rangle_Y \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \langle \Phi_i, \Phi_1 \rangle_Y & \cdots & \langle \Phi_i, \Phi_i \rangle_Y & \cdots & \langle \Phi_i, \Phi_m \rangle_Y \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \langle \Phi_m, \Phi_1 \rangle_Y & \cdots & \langle \Phi_m, \Phi_i \rangle_Y & \cdots & \langle \Phi_m, \Phi_m \rangle_Y \end{bmatrix} \tag{5.2}$$

$$= \begin{bmatrix} \beta_{11} & \cdots & \beta_{1i} & \cdots & \beta_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \beta_{i1} & \cdots & \beta_{ii} & \cdots & \beta_{im} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \beta_{m1} & \cdots & \beta_{mi} & \cdots & \beta_{mm} \end{bmatrix}.$$

A similar modal correlation matrix M_X can be written for the complete space X. M_X is diagonal, and if the mode shapes are normalized, it is simply the identity matrix.

If the Y sampling is dense and uniformly distributed over X, then for some scale factor λ, $M_Y \approx \lambda M_X$. However, this thesis considers the general case in which the sample space Y is not a uniform and complete sampling of the complete structure space X. This situation could arise in practice, for example, if the sensors are focused on only a portion of the structure.

By the Cauchy-Schwarz inequality, it can be shown that

$$\beta_{ij}^{\ 2} \leq \beta_{ii}\beta_{jj}.$$

M_Y is thus guaranteed to be positive semidefinite. The semidefinite condition only arises from a pathological choice of the sample space Y such that certain modes are unobservable or undiscernable (i.e. the modes are aliased spatially). The condition number (maximum ratio of eigenvalues) of M_Y can be checked to determine the proximity to this condition. All further discussion assumes that the measurement space is not pathological and that M_Y is positive definite, well-conditioned, and invertible.

The sample space Y is likely not to be constant. For example, movement of the sensors relative to the target, noise in the sensory data, and deformations in the structure will all change the actual location of the sample points which constitute Y. For this

reason, the modal correlation matrix M_Y and the inner product operator $\langle a,b \rangle_Y$ are not constant in general and are recomputed at each time step.

The sensor measurements can be written as

$$\begin{Bmatrix} \bar{z}(y_1,t) \\ \vdots \\ \bar{z}(y_k,t) \\ \vdots \\ \bar{z}(y_n,t) \end{Bmatrix} = \begin{Bmatrix} z(y_1,t) \\ \vdots \\ z(y_k,t) \\ \vdots \\ z(y_n,t) \end{Bmatrix} + \begin{Bmatrix} e_1 \\ \vdots \\ e_k \\ \vdots \\ e_n \end{Bmatrix}$$

or in vector notation

$$\bar{z} = z + e$$

where the overbar is used to denote a measurement, $z(y_k,t)$ is the true deformation of the target at location y_k, and e_k is additive sensor noise.

The modal coefficients $A_i(t)$ can be recovered through an inner product of the mode shapes with the true deformation of the target:

$$\langle \Phi, z \rangle_Y \equiv \begin{Bmatrix} \langle \Phi_1, z \rangle_Y \\ \vdots \\ \langle \Phi_i, z \rangle_Y \\ \vdots \\ \langle \Phi_m, z \rangle_Y \end{Bmatrix}$$

$$= \begin{bmatrix} \langle \Phi_1, \Phi_1 \rangle_Y & \cdots & \langle \Phi_1, \Phi_i \rangle_Y & \cdots & \langle \Phi_1, \Phi_m \rangle_Y \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \langle \Phi_i, \Phi_1 \rangle_Y & \cdots & \langle \Phi_i, \Phi_i \rangle_Y & \cdots & \langle \Phi_i, \Phi_m \rangle_Y \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \langle \Phi_m, \Phi_1 \rangle_Y & \cdots & \langle \Phi_m, \Phi_i \rangle_Y & \cdots & \langle \Phi_m, \Phi_m \rangle_Y \end{bmatrix} \begin{Bmatrix} A_1(t) \\ \vdots \\ A_i(t) \\ \vdots \\ A_m(t) \end{Bmatrix}$$

$$= M_Y A(t)$$

and thus

$$A(t) = M_Y^{-1} \langle \Phi, z \rangle_Y .$$

Consider the use of noisy measurements in the above equation rather than the true values z.

$$\hat{\hat{A}}(t) = M_Y^{-1} \langle \Phi, \bar{z} \rangle_Y \tag{5.3}$$

$$= M_Y^{-1} \langle \Phi, z + e \rangle_Y = M_Y^{-1} \langle \Phi, z \rangle_Y + M_Y^{-1} \langle \Phi, e \rangle_Y$$

$$= A(t) + M_Y^{-1} \langle \Phi, e \rangle_Y .$$

Since it was assumed that noise e is white and unbiased, the expectation of the inner product of the noise and the mode shapes is zero. That is,

$$E\left[\langle \Phi, e \rangle_Y \right] = 0$$

and thus

$$E\left[\hat{\hat{A}}(t) \right] = A(t).$$

Equation (5.3) is therefore an unbiased estimator of $A(t)$. The actual error between the estimate and true values will be denoted by the vector w

$$w \equiv \hat{\hat{A}}(t) - A(t) = M_Y^{-1} \langle \Phi, e \rangle_Y . \tag{5.4}$$

Since it was assumed that the noise e is Gaussian and white, the error is also Gaussian since it is a summation of Gaussian random variables[1]. The covariance on this estimate is computed as

$$\Lambda_{ww} \equiv E\left[w \cdot w^T\right] = E\left[\left(\hat{\hat{A}}(t) - A(t)\right)\left(\hat{\hat{A}}(t) - A(t)\right)^T\right]$$

$$= E\left[\left(M_Y^{-1}\langle \Phi, e \rangle_Y\right)\left(M_Y^{-1}\langle \Phi, e \rangle_Y\right)^T\right]$$

$$= M_Y^{-1} E\left[\left(\langle \Phi, e \rangle_Y\right)\left(\langle \Phi, e \rangle_Y\right)^T\right] M_Y^{-1}$$

$$= M_Y^{-1} E\left[\begin{Bmatrix} \langle \Phi_1, e \rangle_Y \\ \vdots \\ \langle \Phi_i, e \rangle_Y \\ \vdots \\ \langle \Phi_m, e \rangle_Y \end{Bmatrix}\begin{Bmatrix} \langle \Phi_1, e \rangle_Y \\ \vdots \\ \langle \Phi_i, e \rangle_Y \\ \vdots \\ \langle \Phi_m, e \rangle_Y \end{Bmatrix}^T\right] M_Y^{-1}$$

$$= M_Y^{-1} E\left[\begin{bmatrix} \langle \Phi_1, e \rangle_Y \langle \Phi_1, e \rangle_Y & \cdots & \langle \Phi_1, e \rangle_Y \langle \Phi_j, e \rangle_Y & \cdots & \langle \Phi_1, e \rangle_Y \langle \Phi_m, e \rangle_Y \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \langle \Phi_i, e \rangle_Y \langle \Phi_1, e \rangle_Y & \cdots & \langle \Phi_i, e \rangle_Y \langle \Phi_j, e \rangle_Y & \cdots & \vdots \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \langle \Phi_m, e \rangle_Y \langle \Phi_1, e \rangle_Y & \cdots & \cdots & \cdots & \langle \Phi_m, e \rangle_Y \langle \Phi_m, e \rangle_Y \end{bmatrix}\right] M_Y^{-1} . \tag{5.5}$$

Consider an arbitrary term of the matrix in Equation (5.5).

[1] Even if the noise were not Gaussian, the error w would approach Gaussian by the Central Limit Theorem.

$$E\left[\langle\Phi_i,e\rangle_Y\langle\Phi_j,e\rangle_Y\right]=E\left[\left(\sum_{k=1}^{n}e_k\Phi_i(y_k)\right)\left(\sum_{l=1}^{n}e_l\Phi_j(y_l)\right)\right] \tag{5.6}$$

$$=E\left[\sum_{k=1}^{n}\sum_{l=1}^{n}e_ke_l\Phi_i(y_k)\Phi_j(y_l)\right]$$

$$=\sum_{k=1}^{n}\sum_{l=1}^{n}\Phi_i(y_k)\Phi_j(y_l)E[e_ke_l].$$

Since it was assumed that the noise e is uncorrelated with itself ($E[e_ke_l]=0$ for $k\neq l$), Equation (5.6) reduces to

$$E\left[\langle\Phi_i,e\rangle_Y\langle\Phi_j,e\rangle_Y\right]=\sum_{k=1}^{n}\Phi_i(y_k)\Phi_j(y_k)E\left[e_k{}^2\right]. \tag{5.7}$$

In the simplifying case in which the variance on the noise is approximately the same for all data points ($E\left[e_k{}^2\right]=\sigma_e{}^2$), Equation (5.7) reduces to

$$E\left[\langle\Phi_i,e\rangle_Y\langle\Phi_j,e\rangle_Y\right]=\langle\Phi_i,\Phi_j\rangle\sigma_e{}^2$$

and thus Equation (5.5) simplifies to

$$\Lambda_{ww}=M_Y{}^{-1}\begin{bmatrix}\langle\Phi_1,\Phi_1\rangle_Y & \cdots & \langle\Phi_1,\Phi_i\rangle_Y & \cdots & \langle\Phi_1,\Phi_m\rangle_Y\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \langle\Phi_i,\Phi_1\rangle_Y & \cdots & \langle\Phi_i,\Phi_i\rangle_Y & \cdots & \langle\Phi_i,\Phi_m\rangle_Y\\ \vdots & \ddots & \vdots & \ddots & \vdots\\ \langle\Phi_m,\Phi_1\rangle_Y & \cdots & \langle\Phi_m,\Phi_i\rangle_Y & \cdots & \langle\Phi_m,\Phi_m\rangle_Y\end{bmatrix}\sigma_e{}^2M_Y{}^{-1}$$

$$=M_Y{}^{-1}M_Y\sigma_e{}^2M_Y{}^{-1}$$

$$\Lambda_{ww} = \sigma_e^{\,2} M_Y^{\,-1}. \tag{5.8}$$

For the case in which the noise variance is substantially different for each data point, the equations are not as concise but Λ_{ww} is still easily solved (Section 5.5.1 discusses the more general form of Equation (5.8)). Such a situation could arise in practice if one sensor were much closer to the target than the others, in which case the noise on the data from the closer sensor would be substantially smaller than noise on the other sensors' data.

Equation (5.3) represents an easily computed coarse estimate of $A(t)$. This estimate is unbiased and has a Gaussian error distribution with statistics computed from Equation (5.8). It is a minimum-least-squares estimate of $A(t)$ using data from a single sample time. This estimate can be refined further over time using the fundamental knowledge of the dynamics of $A(t)$. The estimate $\hat{\hat{A}}(t)$ will now be viewed as a surrogate measurement of $A(t)$ that can be sent to a Kalman filter for further refinement in the time domain.

5.3 Kalman Filtering: Sinusoid Estimation

The goal now is to observe the time series $\hat{\hat{A}}(t)$ and extract a better estimate of $A(t)$, denoted $\hat{A}(t)$, using the fundamental knowledge that $A(t)$ is a weakly decaying sinusoid. One would also like to refine the estimates of the natural frequencies ω and the modal damping rates α for each mode. There are many methods to estimate a sinusoid from a noisy time series. A Kalman filter is used here for fast online estimation.

A Kalman filter uses a process model and a measurement model along with a second-order characterization of the noise statistics to update its estimate of the system

state recursively and efficiently. If the models are linear and noise is additive, white, and Gaussian, then the Kalman filter estimates are optimal in a least-squares sense.

Here, the full state to be estimated consists of $\hat{A}(t)$, its time rate of change $\hat{V}(t)$, the natural frequencies $\hat{\omega}$, and the modal damping rate $\hat{\alpha}$. If the true modal coefficients follow a decaying sine wave (of arbitrary phase φ_i), their trajectory is given by

$$A_i(t) = \exp(-t\alpha_i)\sin(t\omega_i + \varphi_i).$$

Differentiation and substitution leads to the process model

$$\frac{d}{dt}\begin{Bmatrix} A_i(t) \\ V_i(t) \\ \omega_i \\ \alpha_i \end{Bmatrix} = \begin{Bmatrix} V_i(t) \\ -\left(\omega_i{}^2 + \alpha_i{}^2\right)A_i(t) - 2\alpha_i V_i \\ 0 \\ 0 \end{Bmatrix} + v_i(t) \qquad (5.9)\text{a}$$

or in discrete time,

$$\begin{Bmatrix} A_i \\ V_i \\ \omega_i \\ \alpha_i \end{Bmatrix}_{(t+\Delta)} = \begin{Bmatrix} \exp(-\Delta\alpha_i)\cdot\left(A_i\cos(\Delta\omega_i) + \dfrac{(V_i + \alpha_i A_i)}{\omega_i}\sin(\Delta\omega_i)\right) \\ \exp(-\Delta\alpha_i)\cdot\left(V_i\cos(\Delta\omega_i) - \dfrac{\alpha_i V_i + \left(\omega_i{}^2 + \alpha_i{}^2\right)A_i}{\omega_i}\sin(\Delta\omega_i)\right) \\ \omega_i \\ \alpha_i \end{Bmatrix}_{(t)} + v_i \qquad (5.9)\text{b}$$

Note the lack of dependence on phase φ_i. If well-known external forces were applied to the structure (e.g. from robotic systems), then these would be incorporated into the process model here. Process noise is indicated by $v \equiv \{v_A \quad v_V \quad v_\omega \quad v_\alpha\}^T$, and is characterized with a covariance matrix

$$\Lambda_{vv} \equiv E\left[vv^T\right]. \tag{5.10}$$

If the process model is perfectly accurate, then process noise covariance is zero and the model is said to be deterministic. In practice, however, external disturbances and unmodeled dynamics usually exist (e.g. nonlinear elasticity, structural hysteresis, gravity gradient/orbital mechanics effects, etc.) and therefore the process noise covariance is nonzero and should be chosen to describe this uncertainty.

In the Kalman filter methodology, the process noise covariance matrix is sometimes viewed as a design variable that encapsulates uncertainty in the system dynamic model. It should be chosen carefully to obtain robust and efficient filter performance. A poor choice of the process noise covariance matrix can lead to estimator instability (for Λ_{vv} chosen too small) or sluggish performance (for Λ_{vv} chosen too large). Therefore, this matrix should be based on the best a priori information available. For example, if the natural frequencies are known to drift (e.g. due to unknown thermal effects) then the diagonal terms of Λ_{vv} corresponding to $E\left[v_\omega v_\omega{}^T\right]$ should correspond to the expected Brownian drift rate of ω.

The Kalman filter measurement model is written as

$$\hat{\hat{A}}(t) = A(t) + w \tag{5.11}$$

which is simply a restatement of Equation (5.4). The noise term w is white, unbiased, and Gaussian with covariance statistics computed from Equation (5.8) (or its more general form described in Section 5.5.1, below).

The last piece of information required by the Kalman filter is the initialization information. The initial a posteriori state estimate is given by

$$\begin{Bmatrix} \hat{A}(0) \\ \hat{V}(0) \\ \hat{\omega}(0) \\ \hat{\alpha}(0) \end{Bmatrix} = \begin{Bmatrix} \hat{\hat{A}}(0) \\ 0 \\ \omega_{exp} \\ \alpha_{exp} \end{Bmatrix} \tag{5.12}$$

where ω_{exp} and α_{exp} are the expected frequencies and damping predicted from the ground station. For lack of better information, the estimated time rate of change of $A(t)$ is initialized to the unbiased value zero.

The state estimation error is denoted

$$\theta(t) \equiv \begin{Bmatrix} \hat{A}(t) - A(t) \\ \hat{V}(t) - V(t) \\ \hat{\omega}(t) - \omega \\ \hat{\alpha}(t) - \alpha \end{Bmatrix}.$$

The initial uncertainty on the a posteriori state estimate is given by

$$\Lambda_{\theta\theta}(0) \equiv E\left[\theta(t)\theta(t)^T\right] \tag{5.13}$$

$$= \begin{bmatrix} \Lambda_{ww} & & & \\ & E\left[\{V(0)\}\{V(0)\}^T\right] & & \\ & & E\left[\{\omega - \omega_{exp}\}\{\omega - \omega_{exp}\}^T\right] & \\ & & & E\left[\{\alpha - \alpha_{exp}\}\{\alpha - \alpha_{exp}\}^T\right] \end{bmatrix}$$

where Λ_{ww} is computed from Equation (5.8) and the other terms encapsulate the uncertainty in the initial estimate. Again, these terms should be chosen judiciously based on the best information available ahead of time.

The implementation of the Kalman filter is now straightforward using Equations (5.8)-(5.13), and the details are left to the reader. Note that since the process model is

nonlinear, an extended Kalman filter, unscented Kalman filter, particle filter, or some other more general Bayesian filter must be used [12, 20, 25, 35, 40, 60, 85]. Since uncertainty in the frequencies is typically small, the extended Kalman filter [12] and unscented Kalman filter [40, 85] generally perform very well for this problem.

5.4 Shape Estimation: Modal Reconstruction

With a good estimate of the modal coefficients, shape estimation is simply a modal reconstruction using Equation (5.1):

$$\hat{z}(x,t) = \sum_{i=1}^{m} \hat{A}_i(t)\hat{\Phi}_i(x) = \hat{A}(t)^T \hat{\Phi}(x). \tag{5.14}$$

The hat notation is used on the mode shapes $\Phi(x)$ as a reminder that the estimator might not know these perfectly, since they are based on theoretical analyses performed ahead of time. The ramifications of uncertainty in mode shapes will be discussed later in Section 5.5.3.

Assuming for now that the mode shape knowledge is perfect, the uncertainty in the shape estimate due strictly to uncertainty in coefficient estimates is computed as

$$\Lambda_{zz}(x,t) \equiv E\left[\left(\hat{z}(x,t) - z(x,t)\right)\left(\hat{z}(x,t) - z(x,t)\right)^T\right]$$

$$= E\left[\left(\Phi(x)^T \theta(t)\right)\left(\Phi(x)^T \theta(t)\right)^T\right]$$

$$\Lambda_{zz}(x,t) = \Phi(x)^T \Lambda_{\theta\theta}(t)\Phi(x). \tag{5.15}$$

5.5 Practical Considerations

5.5.1 Non-uniform Sensory Noise

A derivation of Λ_{ww} for the more general case when the noise on each data point is not the same is now presented. Let the variance on each data point be denoted as

$$\sigma_{ek}{}^2 \equiv E\left[e_k{}^2\right].$$

Define a weighted sample space $\mathbb{Y}$ such that the inner product in this space is given by

$$\langle a,b\rangle_{\mathbb{Y}} = \sum_{k=1}^{n} \sigma_{ek}{}^2 a(y_k) b(y_k).$$

Thus the inner product of two mode shapes in the weighted sample space is given by

$$\langle \Phi_i, \Phi_j \rangle_{\mathbb{Y}} = \sum_{k=1}^{n} \sigma_{ek}{}^2 \Phi_i(y_k) \Phi_j(y_k) \equiv \kappa_{ij}$$

and the modal correlation matrix in this weighted space is given by

$$M_{\mathbb{Y}} = \begin{bmatrix} \kappa_{11} & \cdots & \kappa_{1i} & \cdots & \kappa_{1m} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa_{i1} & \cdots & \kappa_{ii} & \cdots & \kappa_{im} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \kappa_{m1} & \cdots & \kappa_{mi} & \cdots & \kappa_{mm} \end{bmatrix}.$$

The purpose of defining this space is that it possesses the useful property

$$E\left[\left\langle \Phi_i, e\right\rangle_Y \left\langle \Phi_j, e\right\rangle_Y^T\right] = \left\langle \Phi_i, \Phi_j\right\rangle_{\Psi}.$$

Reexamining the derivation of Equation (5.8), a more general form of the modal decomposition error covariance can be written as

$$\Lambda_{ww} = M_Y^{-1} M_{\Psi} M_Y^{-1}. \tag{5.16}$$

Note that the implementation of this more general form is slightly more complex. A different inner product operator must be defined in the software, and the variance on each point must be assessed or computed at every time step. Allocation of memory for the variance on each point might also be significant if there are thousands of data points.

5.5.2 Truncation of Estimated Modes

All discussion so far has assumed that all active modes of vibration are estimated. This might not always be the case in practice. For example, the number of pre-computed mode shapes might be truncated at some large number. This might be done to limit the amount of a priori information transmitted to and stored by the estimator, to reduce estimator computation (recall that M_Y is m-by-m and must be inverted), or simply because modes above a certain threshold frequency may be unimportant to the greater mission tasks.

Let r denote the actual number of modes excited in the system, where $r > m$. The estimator now has incomplete knowledge of the modal correlation matrix M_Y. The complete matrix is denoted with a superscript:

$$M_Y^{\,complete} \equiv \begin{bmatrix} \langle \Phi_1, \Phi_1 \rangle_Y & \cdots & \langle \Phi_1, \Phi_i \rangle_Y & \cdots & \langle \Phi_1, \Phi_r \rangle_Y \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \langle \Phi_i, \Phi_1 \rangle_Y & \cdots & \langle \Phi_i, \Phi_i \rangle_Y & \cdots & \langle \Phi_i, \Phi_r \rangle_Y \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \langle \Phi_r, \Phi_1 \rangle_Y & \cdots & \langle \Phi_r, \Phi_i \rangle_Y & \cdots & \langle \Phi_r, \Phi_r \rangle_Y \end{bmatrix}$$

which can be written in the form

$$M_Y^{\,complete} = \begin{bmatrix} M_Y & C_Y \\ C_Y^T & \tilde{M}_Y \end{bmatrix}$$

where M_Y is the modal correlation matrix for the estimated modes, $\tilde{M}_Y$ is the modal correlation matrix for the unestimated modes, and C_Y is the cross-correlation matrix between the estimated and unestimated modes.

The first m true modal coefficients are computed as

$$\langle \Phi, z \rangle_Y = M_Y A(t) + C_Y \tilde{A}(t)$$

where $\tilde{A}(t)$ denotes the true values of the residual (unestimated) modes. Rearranging yields

$$A(t) = M_Y^{-1} \left(\langle \Phi, z \rangle_Y - C_Y \tilde{A}(t) \right).$$

Estimation error is now computed as

$$w = \hat{\hat{A}}(t) - A(t) = M_Y^{-1} \langle \Phi, \tilde{z} \rangle_Y - M_Y^{-1} \left(\langle \Phi, z \rangle_Y - C_Y \tilde{A}(t) \right)$$

$$= M_Y^{-1} \langle \Phi, z \rangle_Y + M_Y^{-1} \langle \Phi, e \rangle_Y - M_Y^{-1} \left(\langle \Phi, z \rangle_Y - C_Y \tilde{A}(t) \right)$$

$$= M_Y^{-1}\langle \Phi, e\rangle_Y + M_Y^{-1} C_Y \tilde{A}(t). \tag{5.17}$$

Comparing this to Equation (5.4), in the case of modal truncation the error contains an additional *time-varying nonrandom* term which is not zero in general. Thus Equation (5.3) is now a *biased* estimator. Because the bias is nonrandom and uncorrelated with the noise, the error covariance matrix is given by

$$\Lambda_{ww} = E\left[\left(M_Y^{-1}\langle \Phi, e\rangle_Y\right)\left(M_Y^{-1}\langle \Phi, e\rangle_Y\right)^T\right] + \left(M_Y^{-1} C_Y \tilde{A}(t)\right)\left(M_Y^{-1} C_Y \tilde{A}(t)\right)^T$$

$$= \sigma_e^{\,2} M_Y^{-1} + \left(M_Y^{-1} C_Y \tilde{A}(t)\right)\left(M_Y^{-1} C_Y \tilde{A}(t)\right)^T \tag{5.18}$$

which is similar to Equation (5.8) but with an additional term due to the bias.

Under certain conditions these bias terms are negligible. That is,

$$\left(M_Y^{-1} C_Y \tilde{A}(t)\right)\left(M_Y^{-1} C_Y \tilde{A}(t)\right)^T << \sigma_e^{\,2} M_Y^{-1}. \tag{5.19}$$

This condition is satisfied in several real-world situations. First, the condition can be met if the amplitude of vibration of the residual modes is very small compared to the estimated modes. Second, the condition is met whenever the residual modes are approximately orthogonal to the estimated modes in sample space Y (i.e. $C_Y \approx 0$). This second case can occur, for example, if the sample space Y is dense and uniform over the complete space X.

In general, however, the bias terms in Equations (5.17) and (5.18) may be significant and therefore must be addressed. The bias can be mitigated a number of ways. The first and most obvious solution is to simply estimate a larger number of modes, in effect not truncating the model at all. Of course the drawback is that this

requires more computation (recall that M_Y must be inverted) and this solution becomes impractical as the number of modes increases. Further, it is still unclear just how many modes need to be estimated, since in principle the structure has an infinite number of modes.

A second solution is to place the sensors in a configuration that makes C_Y vanish, i.e. makes the unestimated modes unobservable. As mentioned above, this might be done by sampling the space X in a uniform and complete manner. However, this solution is often impractical, either because such poses are difficult to determine, or because it imposes undesirable requirements on sensor placement and motion.

A third approach is to treat the bias as random noise and let the Kalman filter remove it naturally. This may be a simple and reasonable solution for relatively small biases. However, if the biases are large, this solution severely degrades Kalman filter performance. The estimator has no information regarding the second term of Equation (5.18), since it involves unknown and unestimated information (C_Y and $\tilde{A}(t)$). Consequently, this "noise" covariance must be provided as a blind guess, which may be incorrect by orders of magnitude. If the guess is undersized, the filter could become unstable; if the guess is oversized, the estimation will be very sluggish. Further, this solution is fundamentally suboptimal, since the bias is poorly modeled by random noise.

A fourth method is to use a low-pass (anti-aliasing) filter on the surrogate measurement $\hat{\hat{A}}(t)$ immediately prior to the Kalman filter (see Figure 5.2). Since the bias term in Equation (5.17) is a linear sum of high-frequency sinusoids, a low-pass filter will effectively eliminate the bias. There are two important considerations, however. First, the anti-aliasing filter is implemented in discrete time, and thus the Nyquist sampling criterion must be met for the unestimated modes. Second, the Kalman filter must compensate the phase shift and attenuation of the desired signal $A(t)$ caused by the anti-aliasing filter. This typically involves a forward dynamic propagation and scaling using

the Kalman filter's dynamic model and latest estimate $\hat{A}(t)$. The effectiveness of this method is shown in simulation in the following chapter.

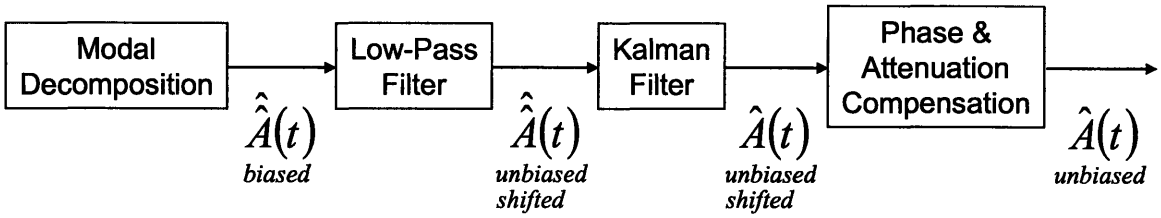


Figure 5.2. Using a low-pass filter to attenuate bias due to modal truncation.

5.5.3 Imperfect Knowledge of Mode Shapes

The discussion so far has assumed that the estimator has perfect knowledge of the mode shapes $\Phi(x)$. In practice, it is unlikely that these will be known exactly. Nonlinearities, parameter uncertainty, and modeling errors, for example, will lead to errors in the computed mode shapes. Consider the following relation between the predicted and the actual mode shapes:

$$\Gamma_k \hat{\Phi}(x) \equiv \Phi(x) + \delta_k(x) \tag{5.20}$$

where $\delta_k(x)$ represents a small error displacement function. The scaling matrix Γ_k is approximately the identity and is chosen to satisfy

$$\left\langle \hat{\Phi}, \delta_k \right\rangle_{Yk} \equiv [0]. \tag{5.21}$$

Note that in order to satisfy Equation (5.21), $\delta_k(x)$ and Γ_k will change slightly whenever the sample space Y changes. Figure 5.3 shows a graphical interpretation of these equations.

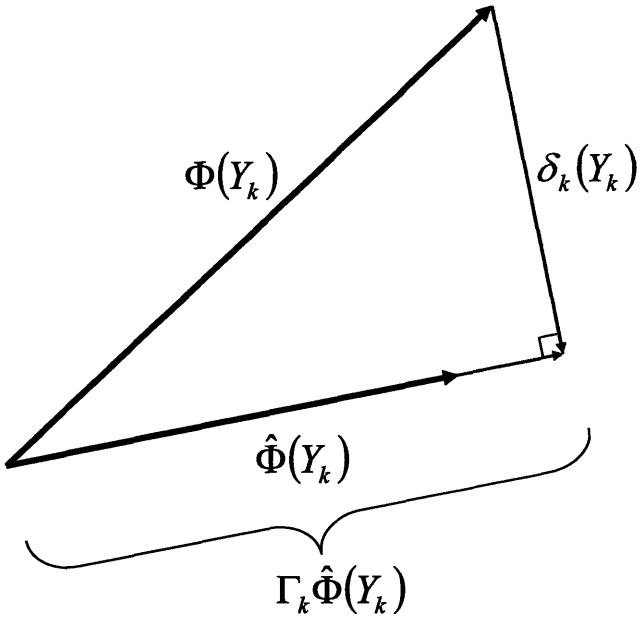


Figure 5.3. Graphical interpretation of Equations (5.20) and (5.21).

Referring to Equation (5.3), the estimate of the modal coefficients $A(t)$ will be computed as

$$\hat{\hat{A}}(t) = \hat{M}_Y^{-1}\left\langle \hat{\Phi}, \bar{z} \right\rangle_Y \tag{5.22}$$

where the approximate modal correlation matrix $\hat{M}_Y$ is computed as

$$\hat{M}_Y \equiv \begin{bmatrix} \left\langle \hat{\Phi}_1, \hat{\Phi}_1 \right\rangle_Y & \cdots & \left\langle \hat{\Phi}_1, \hat{\Phi}_m \right\rangle_Y \\ \vdots & \ddots & \vdots \\ \left\langle \hat{\Phi}_m, \hat{\Phi}_1 \right\rangle_Y & \cdots & \left\langle \hat{\Phi}_m, \hat{\Phi}_m \right\rangle_Y \end{bmatrix}.$$

The estimate of target shape is thus

$$\hat{\hat{z}}(x,t) = \hat{\hat{A}}(t)^T \hat{\Phi}(x) = \left[\hat{M}_Y^{-1}\left\langle \hat{\Phi}, \bar{z} \right\rangle_Y\right]^T \hat{\Phi}.$$

Consider the following metric for quantifying the magnitude of the shape estimation error due to imperfect mode shape knowledge:

$$\varepsilon_B \equiv \frac{\left\| E\left(\hat{\hat{z}}\right) - z \right\|_B}{\left\| z \right\|_B} \tag{5.23}$$

where the double brackets indicate the norm of the enclosed value over the subscripted space. That is,

$$\left\| a \right\|_B \equiv \sqrt{\langle a, a \rangle_B} \, .$$

In other words, Equation (5.23) defines a ratio between the RMS shape estimation error and the RMS amplitude of vibration at one instant in time. The space B could be X, Y, or any other space over which one would like to compute shape error. The denominator of Equation (5.23) can be computed as

$$\left\| z \right\|_B = \sqrt{\langle z, z \rangle_B} = \sqrt{\left\langle A^T \Phi, A^T \Phi \right\rangle_B} = \sqrt{A^T \langle \Phi, \Phi \rangle_B A} = \sqrt{A^T M_B A} \, . \tag{5.24}$$

The numerator of Equation (5.23) is computed as follows

$$\left\| E\left(\hat{\hat{z}}\right) - z \right\|_B = \sqrt{\left\langle E\left(\hat{\hat{z}}\right) - z, E\left(\hat{\hat{z}}\right) - z \right\rangle_B}$$

$$= \sqrt{\left\langle E\left(\hat{\hat{A}}\right)^T \hat{\Phi} - A^T \Phi, \cdots\cdots \right\rangle_B}$$

$$= \sqrt{\left\langle \left[\hat{M}_Y^{-1}\left\langle \hat{\Phi}, z\right\rangle_Y\right]^T \hat{\Phi} - A^T\Phi, \cdots\cdots \right\rangle_B}$$

$$= \sqrt{\left\langle \left[\hat{M}_Y^{-1}\left\langle \hat{\Phi}, \Phi\right\rangle_Y A\right]^T \hat{\Phi} - A^T\Phi, \cdots\cdots \right\rangle_B}$$

$$= \sqrt{A^T\left\langle \left\langle \Phi, \hat{\Phi}\right\rangle_Y \hat{M}_Y^{-1}\hat{\Phi} - \Phi, \cdots\cdots \right\rangle_B A}$$

$$= \sqrt{A^T\left\langle \left\langle \Gamma_k\hat{\Phi} - \delta_k, \hat{\Phi}\right\rangle_Y \hat{M}_Y^{-1}\hat{\Phi} - \left(\Gamma_k\hat{\Phi} - \delta_k\right), \cdots\cdots \right\rangle_B A}$$

$$= \sqrt{A^T\left\langle \left(\Gamma_k\hat{M}_Y - 0\right)\hat{M}_Y^{-1}\hat{\Phi} - \Gamma_k\hat{\Phi} + \delta_k, \cdots\cdots \right\rangle_B A}$$

$$\left\| \mathrm{E}\left(\hat{\hat{z}}\right) - z \right\|_B = \sqrt{A^T\left\langle \delta_k, \delta_k\right\rangle_B A} = \sqrt{A^T D_{Bk} A} \tag{5.25}$$

where the matrix D_{Bk} is defined as

$$D_{Bk} \equiv \left\langle \delta_k, \delta_k \right\rangle_B$$

Substituting Equations (5.24) and (5.25) into Equation (5.23),

$$\varepsilon_B = \sqrt{\frac{A^T D_{Bk} A}{A^T M_B A}}\,. \tag{5.26}$$

Examining order of magnitude relationships between estimation error and mode shape uncertainty reveals

$$O(\varepsilon_B) = O\left(\sqrt{\frac{A^T D_{Bk} A}{A^T M_B A}}\right) \approx O\left(\sqrt{\frac{|A| \cdot \|D_{Bk}\|_{Fro} \cdot |A|}{|A| \cdot \|M_B\|_{Fro} \cdot |A|}}\right) = O\left(\sqrt{\frac{\|D_{Bk}\|_{Fro}}{\|M_B\|_{Fro}}}\right)$$

$$= O\left(\sqrt{\frac{\|\langle \delta_k, \delta_k \rangle_B\|_{Fro}}{\|\langle \Phi, \Phi \rangle_B\|_{Fro}}}\right) = O\left(\sqrt{\frac{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}\langle \delta_{ik}, \delta_{jk} \rangle_B^2}}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}\langle \Phi_i, \Phi_j \rangle_B^2}}}\right) \approx O\left(\frac{\|\delta_k\|_B}{\|\Phi\|_B}\right) \qquad (5.27)$$

where single brackets indicate the vector norm and double brackets with subscript *Fro* indicate the Frobenius (matrix) norm.

This result suggests that estimation errors arising from imperfect modal knowledge are directly proportional to the ratio between mode shape errors and the mode shape amplitudes. Importantly, this suggests that estimator performance degrades gracefully rather than catastrophically as mode shape uncertainty grows.

5.5.4 Errors in Variables

Another implicit assumption made so far deals with the specific way in which the structure is sensed. So far it has been assumed that for each sensor data point, its location in X space is known perfectly – that is, y_k is known perfectly for the k^{th} data point. In practice, however, this may not be true. If the sensor has an oblique view of the target, or if there is noise in the focal plane direction (i.e. normal to the range direction), then y_k will not be known perfectly. Performing a regression (e.g. shape estimation) while having errors in the regressors is known as an *errors in variables* problem [14, 23].

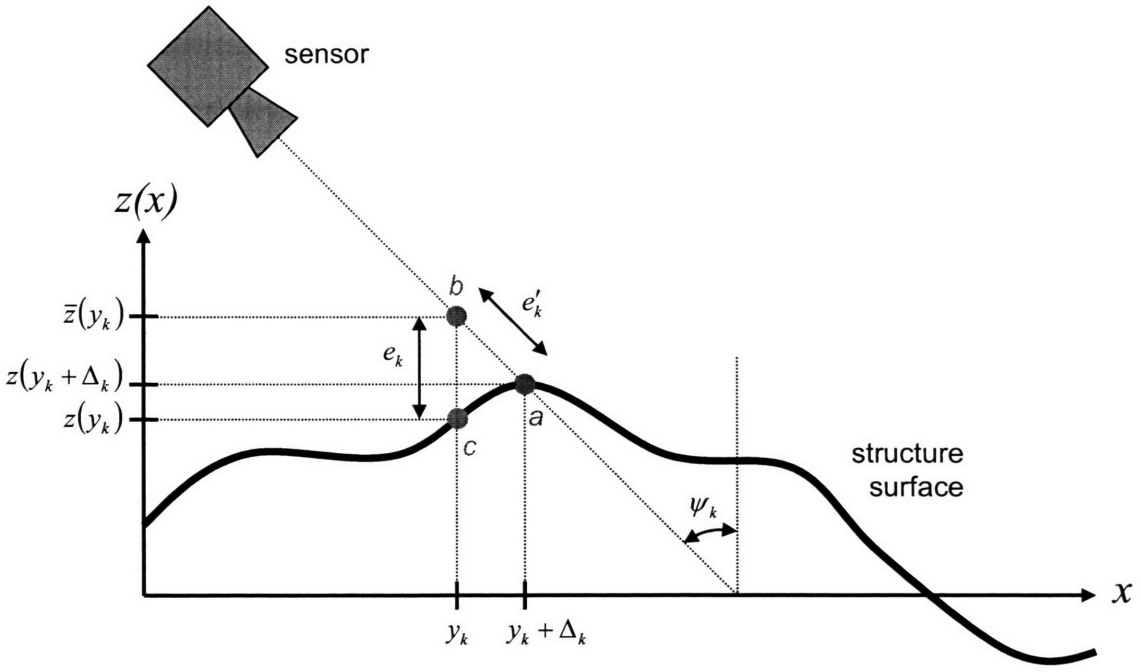


Figure 5.4. Errors in variables for a sensor viewing a structure.

Figure 5.4 illustrates a deformed structure being viewed by a range imaging sensor. The point *a* denotes the actual point on the structure being sensed. Due to noise e'_k in the range measurement, the actual point is measured to exist at point *b*[1]. Due to measurement noise and the oblique viewing angle ψ_k, the projections onto *X* space of the two points will differ. Let y_k denote the *X*-projection of the measured point *b* and $(y_k + \Delta_k)$ denote the *X*-projection of the actual point *a*. Since the value of the measurement noise e'_k is unknown, the value of Δ is also unknown.

Earlier, the measurement was modeled using the equation

$$\bar{z}(y_k) = z(y_k) + e_k$$

[1] Measurement noise in the focal plane direction has been omitted here for clarity, but could be included in the following analysis with similar effect.

where the noise term e_k was assumed to be white, Gaussian, and unbiased. Consider the validity of this statement if there are errors in variables y_k. Examination of the figure above shows that

$$e_k = e_k' \cos(\psi_k) + z(y_k + \Delta_k) - z(y_k)$$

$$= e_k' \cos(\psi_k) + \Delta_k \left(\frac{\partial z}{\partial x} \right)\Bigg|_{x=y_k} + \frac{\Delta_k^2}{2} \left(\frac{\partial^2 z}{\partial x^2} \right)\Bigg|_{x=y_k} + \ldots$$

where the latter terms come from the Taylor series expansion of $z(x)$. Noting that $\Delta_k = e_k' \sin(\psi_k)$ and ignoring higher order terms in Δ_k yields

$$e_k \approx e_k' \cos(\psi_k) + e_k' \sin(\psi_k) \left(\frac{\partial z}{\partial x} \right)\Bigg|_{x=y_k} + \tfrac{1}{2} (e_k' \sin(\psi_k))^2 \left(\frac{\partial^2 z}{\partial x^2} \right)\Bigg|_{x=y_k} \qquad (5.28)$$

If measurement noise in the range direction (e_k') is Gaussian, white, and unbiased, then the first term of Equation (5.28) will be Gaussian, white, and unbiased. The second term will also be Gaussian and unbiased. However, the magnitude of its covariance will be a function of the slope of the structure, and thus the second term is not white with respect to the deformation of the structure. Finally, the third term will be biased since it is a quadratic function of the random variable e_k'. Further, this bias is a function of the curvature of the structure at the point y_k, and thus the third term is also not white with respect to the deformation of the structure. Thus, if there are errors in the variables y_k, then technically e_k is not Gaussian, white, or unbiased.

In practice, however, Δ_k, Δ_k^2, the gradient of z, and the curvature of z are all very small values and thus

$$\left| e_k' \sin(\psi_k) \left(\frac{\partial z}{\partial x} \right)\Big|_{x=y_k} + \tfrac{1}{2} (e_k' \sin(\psi_k))^2 \left(\frac{\partial^2 z}{\partial x^2} \right)\Big|_{x=y_k} \right| << |e_k' \cos(\psi_k)|.$$

Thus in practice, the estimation error or loss of optimality due to errors in variables will typically be negligible. Further, it should be noted that a sensor viewpoint approximately normal to the space X will also virtually eliminate the non-white and bias terms, and will make $e_k \approx e_k'$.

CHAPTER

6

ESTIMATION OF VIBRATION MODES: SIMULATION AND EXPERIMENTAL STUDIES

6.1 Introduction

This chapter presents simulation and experimental studies for the vibration estimator developed in the previous chapter. These studies were performed to validate the performance and to explore practical challenges of the estimator. The chapter is divided into two main sections: simulation studies (Section 6.2) and experimental studies (Section 6.3).

6.2 Simulation Studies

6.2.1 Simulation Environment

Computer simulations were used to study performance of the vibrational estimator. Representative space structures were built into a virtual environment very similar to the one described in Section 4.2.1. The structural models used here were based on state-of-the-art deployable space structures [2]. Mode shapes, frequencies, and dynamic responses of the simulated structures were provided from finite element analysis conducted by Vickram Mangalgiri [51].

Figure 6.1 shows a structure used in simulation studies, which is representative in scale and stiffness of proposed space solar power systems [52, 61]. Each triangular element has a 200-meter side length, making the entire structure approximately 2 km by 2 km in size. The first mode of vibration has a period of approximately 40 minutes and the one-hundredth mode has a period of approximately 40 seconds. Further details of the structural model can be found in [51].

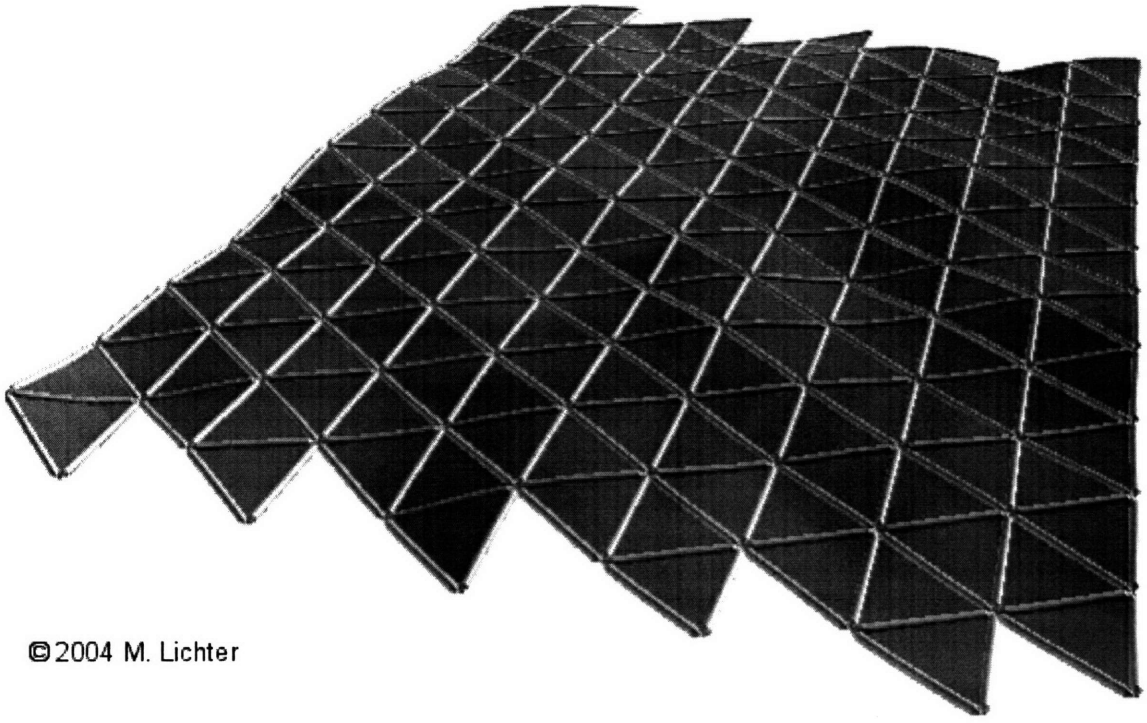


Figure 6.1. A planar space structure used in simulation studies. Vibrations occur primarily in the out-of-plane direction. Image made using OpenGL [68].

As was done in the rigid body estimator simulations, simulated sensors were placed in this environment and their range images were synthesized[1]. The sensors were given a resolution of only 30 by 30 pixels and a field of view of 80 degrees. Gaussian

[1] The algorithm for synthesizing range images is presented in Appendix B.

noise was added to the synthetic range images, with a magnitude that varied depending on the study.

6.2.2 Estimator Performance

Simulation studies were used to evaluate estimator performance. Each mode shape was scaled so that a modal coefficient of 1.0 indicated the same elastic potential energy for all modes (i.e. the modes were energy-normalized). The simulated structure was given a random excitation on the interval $0 \leq A_{i_{\max}} \leq 1$, meaning the expected energy in each mode was the same. Phase for each mode was randomly selected over the interval $0 \leq \varphi_i < 2\pi$. For initial studies, damping was set to zero, all the active modes were estimated, and the mode shapes were known perfectly by the estimator. A single simulated sensor was used to observe about one-quarter of the entire structure (see Figure 6.2). Range images were taken at a rate of one per simulated minute. Gaussian sensor noise was simulated with a standard deviation of 3% of the measurement in the range direction. This value is similar to that found in practical sensors proposed for space applications [29, 82, 84]. Note that scaling the range noise simply scales the noise on the kinematic data fusion outputs proportionally (see Equation (5.8), page 93).

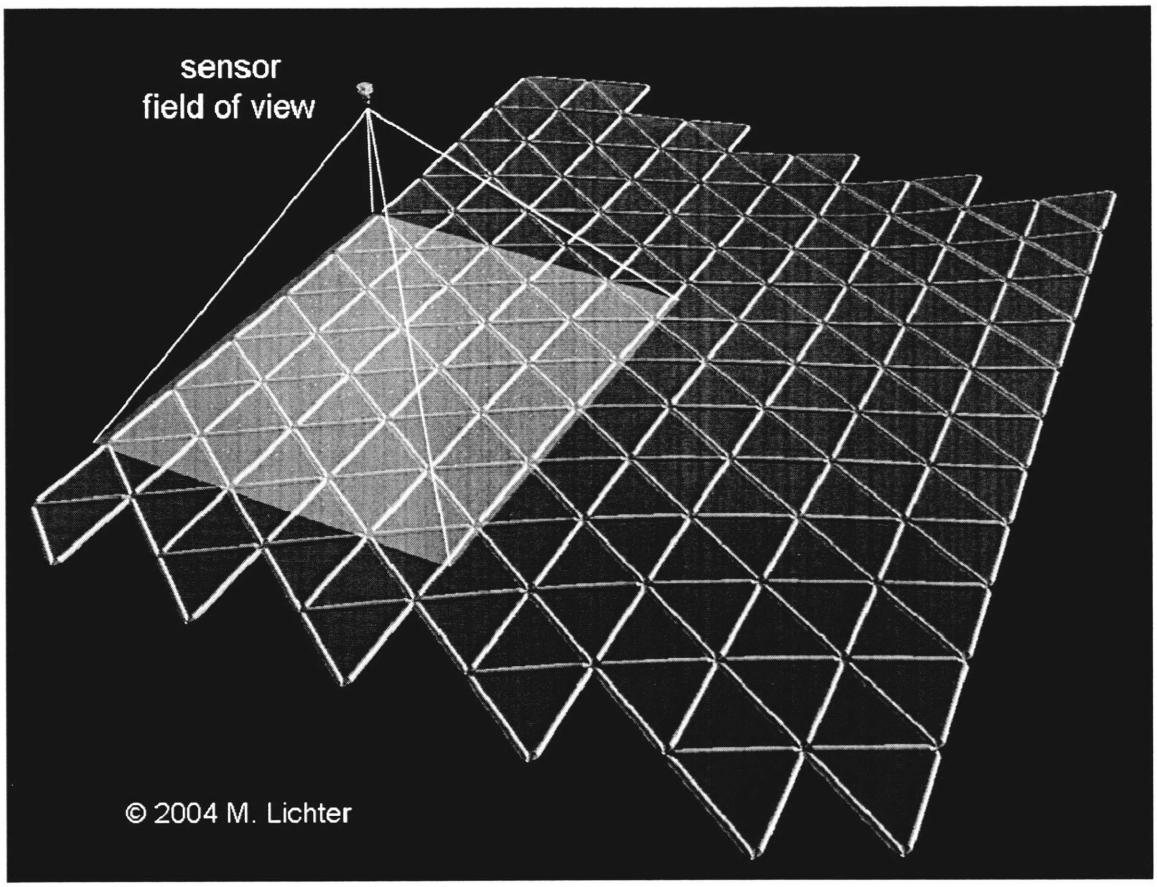


Figure 6.2. Sensor placement for simulation studies.

Figure 6.3 shows typical simulation results for the surrogate measurements (kinematic data fusion outputs) and filtered estimates of a few modal coefficients. These results are similar for all modes, regardless of number estimated, as long as the Nyquist sampling criterion is met with respect to the modal frequencies. Note that the signal-to-noise ratio is different for each mode, which is largely due to the sample space chosen, which directly affects the matrix M_y (see Equation (5.8), page 93).

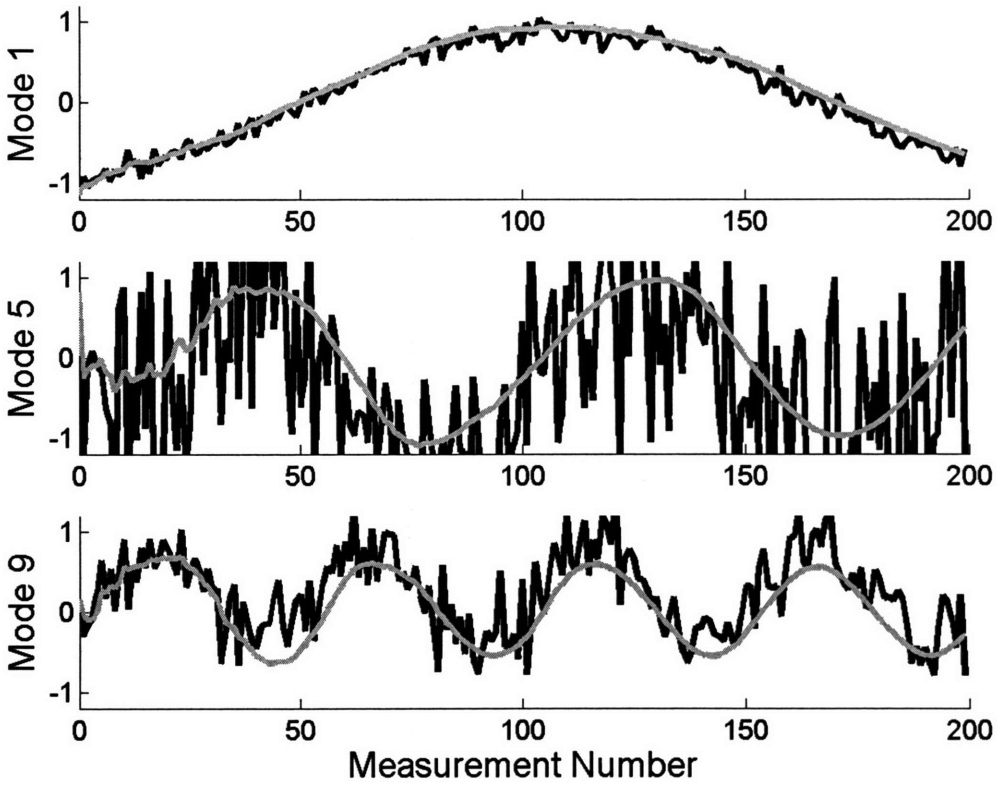


Figure 6.3. Surrogate measurements of modal coefficients (black lines), with results of Kalman filtering superimposed (gray lines) (simulation results).

Figure 6.4 shows frequency estimation histories for these simulations. Again, these results are typical for any number of modes estimated, as long as the Nyquist sampling criterion is met for each mode. Note that all frequencies are correctly estimated within a few periods of vibration.

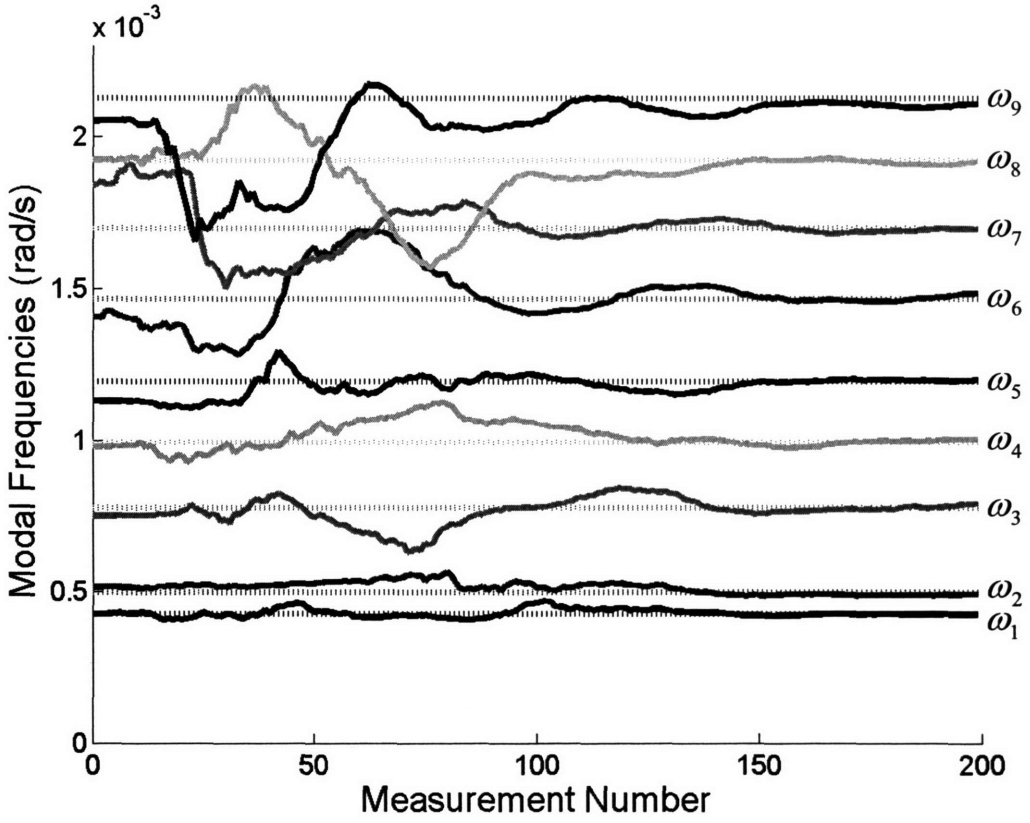


Figure 6.4. Frequency estimates (simulation results). True values shown with dotted lines.

Figure 6.5 shows overall shape estimation errors as a function of time. Shape estimation error is defined in this figure as the root-mean-square (RMS) estimation error computed over the entire structure:

$$\theta \equiv \sqrt{\frac{\int_X \left(\hat{z}(x) - z(x)\right)^2 dx}{\int_X dx}}. \tag{6.1}$$

Recall that the first mode of vibration has a period of 40 minutes. Figure 6.5 suggests that very good shape estimates can be achieved within a few periods of vibration of the first mode.

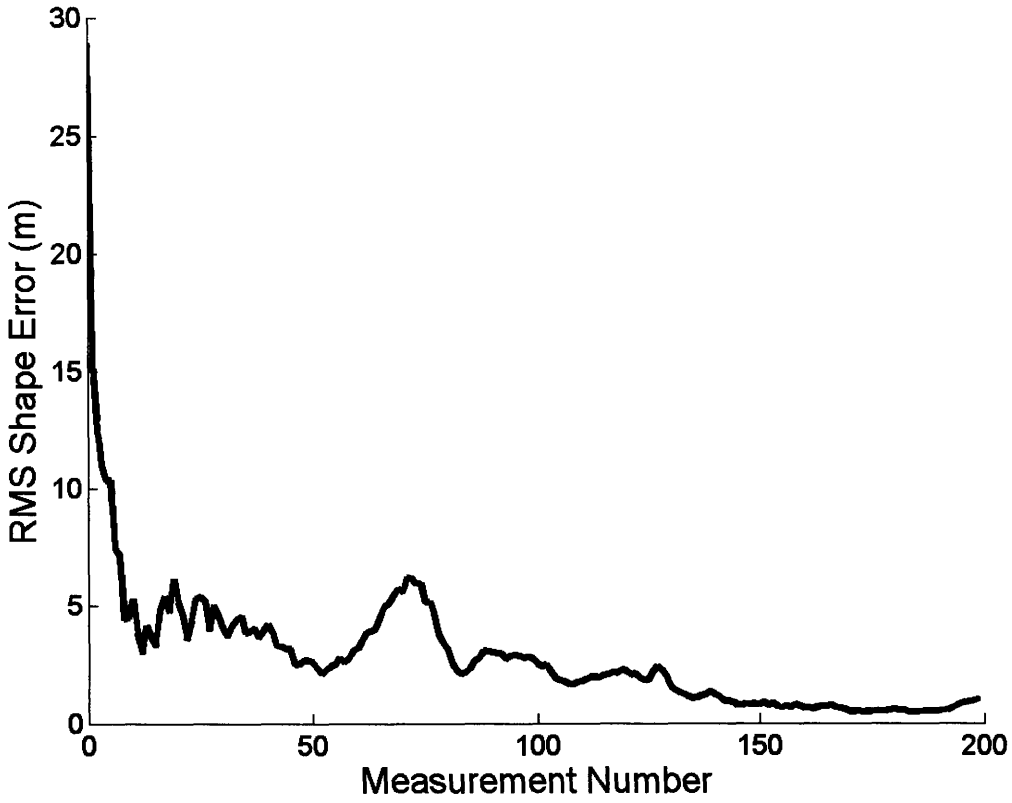


Figure 6.5. Shape estimation error vs. time, with all excited modes estimated (simulation results). Measurements are taken once per simulated minute. Amplitude of vibration of structure is on the order of 25 m.

Figure 6.6 shows a typical parameter estimation history when damping was introduced into the simulation. These results show that damping estimation requires a long time to converge. This is largely because it is a subtle effect that is only observed over many periods of vibration. Note that in this particular trial, the estimator was able to correctly identify all of the coefficients very accurately except for that of the sixth mode (denoted α_6). This misidentification is due to a very low signal-to-noise ratio in the surrogate measurement for that mode. It should be pointed out that if motions damp out before the estimator has converged, the estimates will be frozen at incorrect values, since parameters are unobservable without target motion. In the space applications discussed here, damping is likely to be very small, making this an unlikely occurrence in practice.

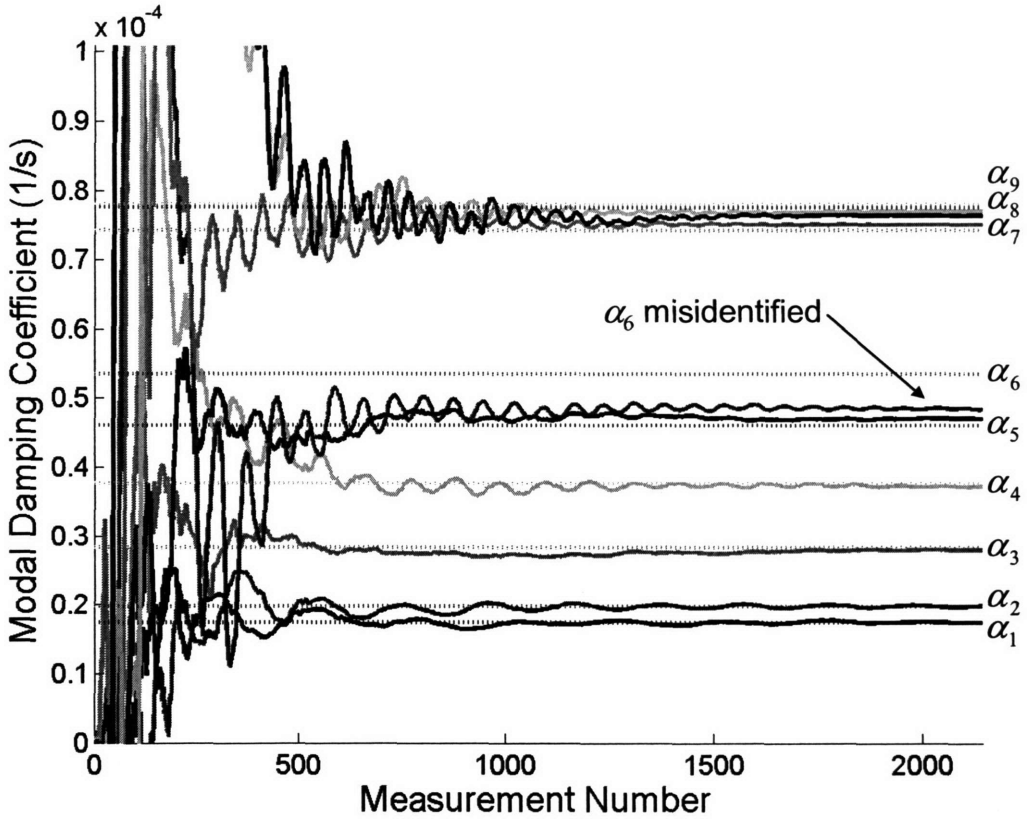


Figure 6.6. Modal damping estimation (simulation results). True values shown with dotted lines.

Simulations were used to evaluate performance when not all modes are estimated (modal truncation). In the previous chapter it was shown that ignoring modes during estimation introduces bias in the surrogate measurements that is a linear combination of the unestimated modal coefficients (Equation (5.17), page 101). It was argued that the most appropriate method for dealing with this bias is to use a low-pass filter to remove it from the surrogate measurements before it reaches the Kalman filter (see Figure 5.2, page 103).

Figure 6.7 and Figure 6.8 show filter performance for the case in which only the first six of thirty excited modes are estimated. Results are shown for the nominal estimator design and for a design that uses a low-pass prefilter prior to the Kalman filter.

For these studies, a discrete-time eighth-order low-pass Butterworth filter was used. The frequency of the last mode estimated (i.e. the sixth modal frequency) was chosen for the cutoff of the Butterworth filter. Note that some non-zero shape estimation error is unavoidable if not all modes are estimated, and this theoretical limit is represented by the dashed line in Figure 6.7. These results clearly suggest the effectiveness of using a low-pass prefilter before the Kalman filter.

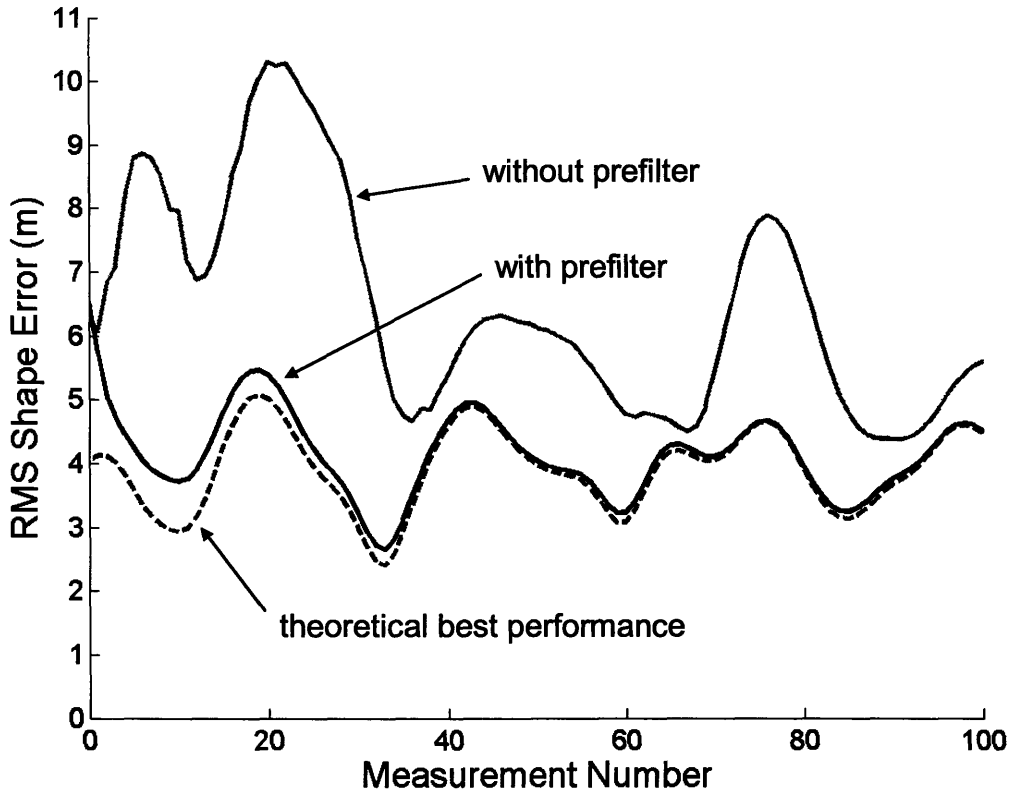


Figure 6.7. Shape estimation error vs. time, when not estimating all excited modes (simulation results). The first 6 out of 30 modes are estimated here. Amplitude of vibration of structure is on the order of 25 m.

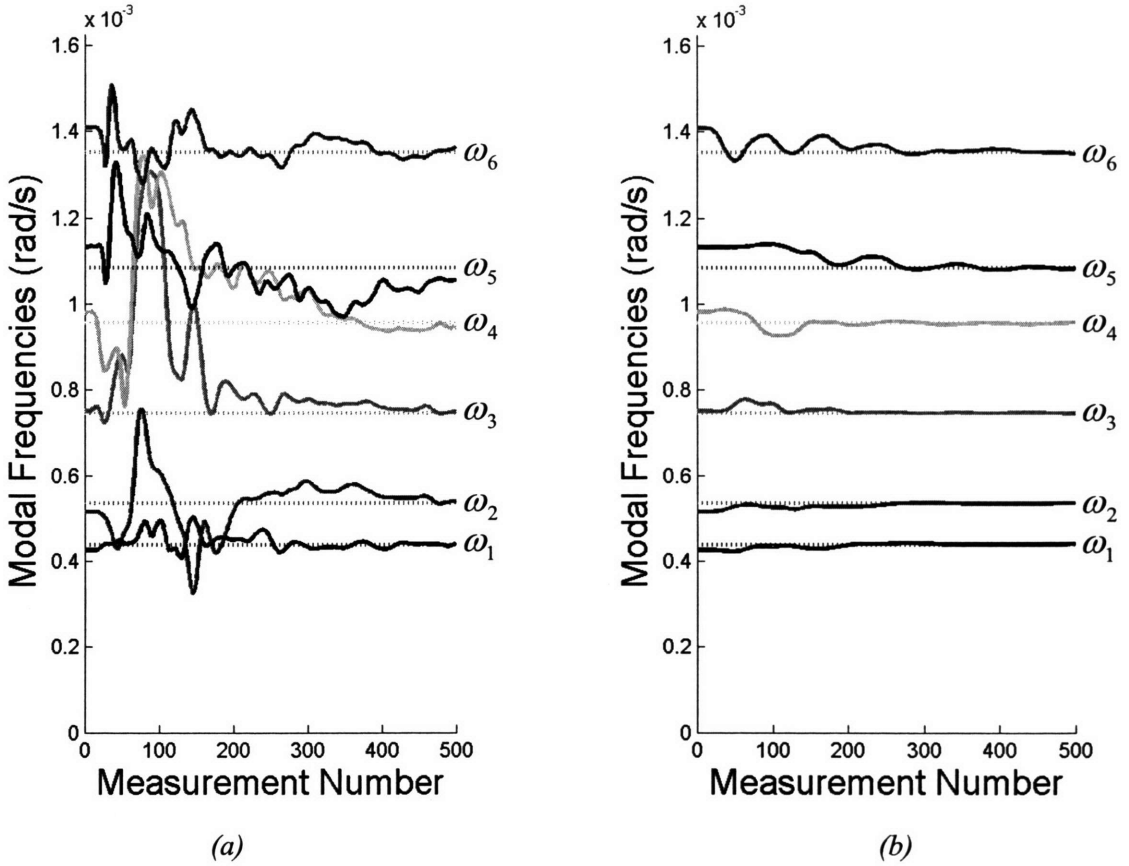


Figure 6.8. Frequency estimation when not estimating all excited modes (simulation results): (a) without prefilter; (b) with prefilter. The first 6 out of 30 modes are estimated here.

Simulations were used to evaluate the effect of sensor location and field of view on estimator performance. Single and multiple sensors were studied, viewing anywhere from 5% to 100% of the structure, all with similar results to those presented above. The simulations suggest that as long as the modes are observable[1], estimation is efficient and robust. Similarly, increasing sensory noise only scales up the noise on the kinematic data fusion outputs (surrogate measurements), which generally only slows the rate of convergence of the Kalman filter.

[1] i.e. the modal correlation matrix M_Y is well-conditioned (see Equation (5.2), page 87).

6.3 Experimental Studies

6.3.1 Experimental Platform

With good estimator performance observed in simulation, some simple experiments were conducted to assess practical challenges and qualitatively study performance using real hardware. The experimental platform described in Section 4.3.1 was used here to provide realistic sensory data to the estimator. Figure 6.9 and Figure 6.10 show a multi-degree-of-freedom structure used in the experimental studies. It is a quadruple-pendulum composed of four panels and low-friction rotational joints. The upper joint is fixed in inertial space. The system is not instrumented to provide a ground-truth of its motions, and therefore only qualitative results are discussed here.

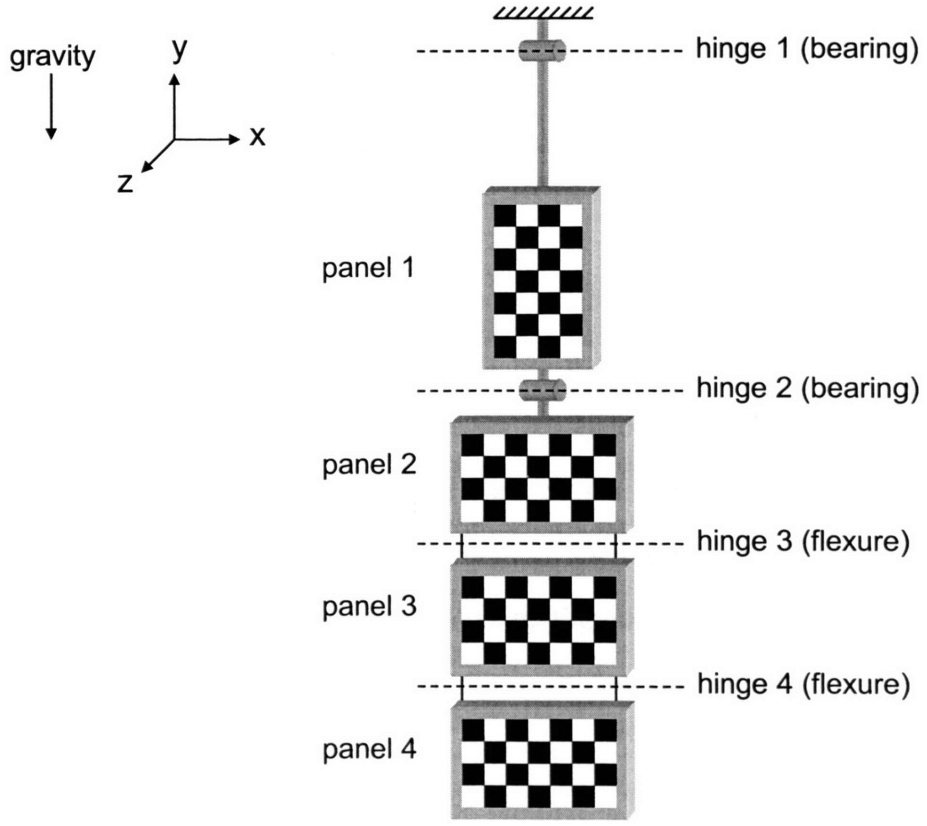


Figure 6.9. Schematic of multi-degree-of-freedom structure used in experimental studies.

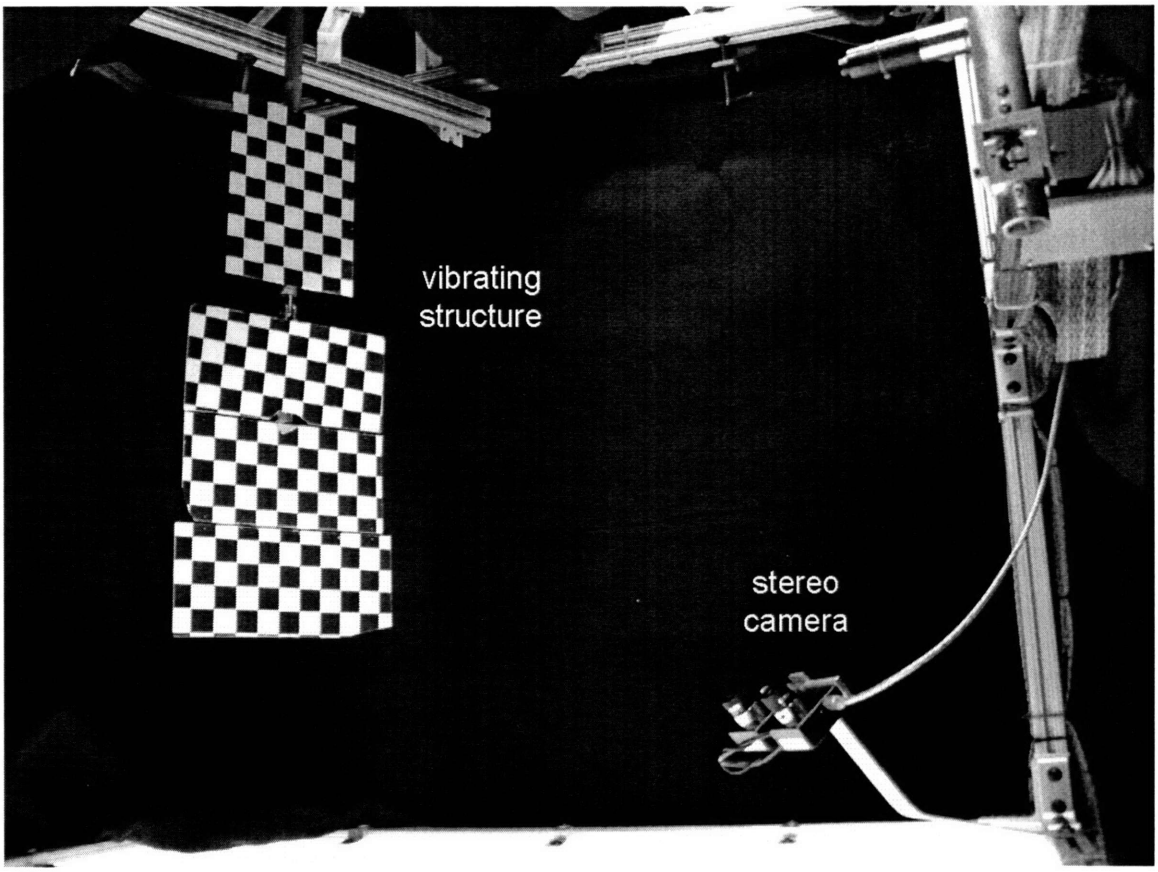


Figure 6.10. Photograph of multi-degree-of-freedom structure used in experimental studies.

The panels are textured to make them visible to the stereo cameras. It is important to note however that these textures are not used to provide feature sets for tracking. Range image points or features are *not* correlated across sample times.

Gravity provides a restoring force that causes the structure to vibrate when perturbed. The four dominant modes of vibration involve rotational motions about the four hinges, which are parallel to the x-axis in Figure 6.9. Due to backlash in the bearings, several parasitic modes exist that allow limited motions about orthogonal axes (i.e. axes parallel to the y- and z-axes).

Recall that the vibrational estimator requires a priori knowledge of the mode shapes of the structure, and a reasonable guess of the modal frequencies (within ± 20

percent). The modal frequencies were empirically derived (to one digit of precision) by timing the actual system with a stopwatch. The approximate mode shapes were provided by digitizing still images of the structure, after exciting each mode of vibration independently. The approximate camera location with respect to the structure was determined manually by visually aligning range images of the structure to geometric models. Certainly, more accurate a priori estimates of the mode shapes, frequencies, and camera pose could be provided. However, one intention of using this imprecise method was to explore the estimator's robustness to fairly high a priori uncertainty.

6.3.2 Estimator Performance

Figure 6.11 shows typical experimental results for the surrogate measurements and filtered estimates of the first three modal coefficients. The fourth mode was not estimated since it had very high damping and a frequency near the Nyquist sample rate (the stereo camera used here has a frame rate of 15 Hz).

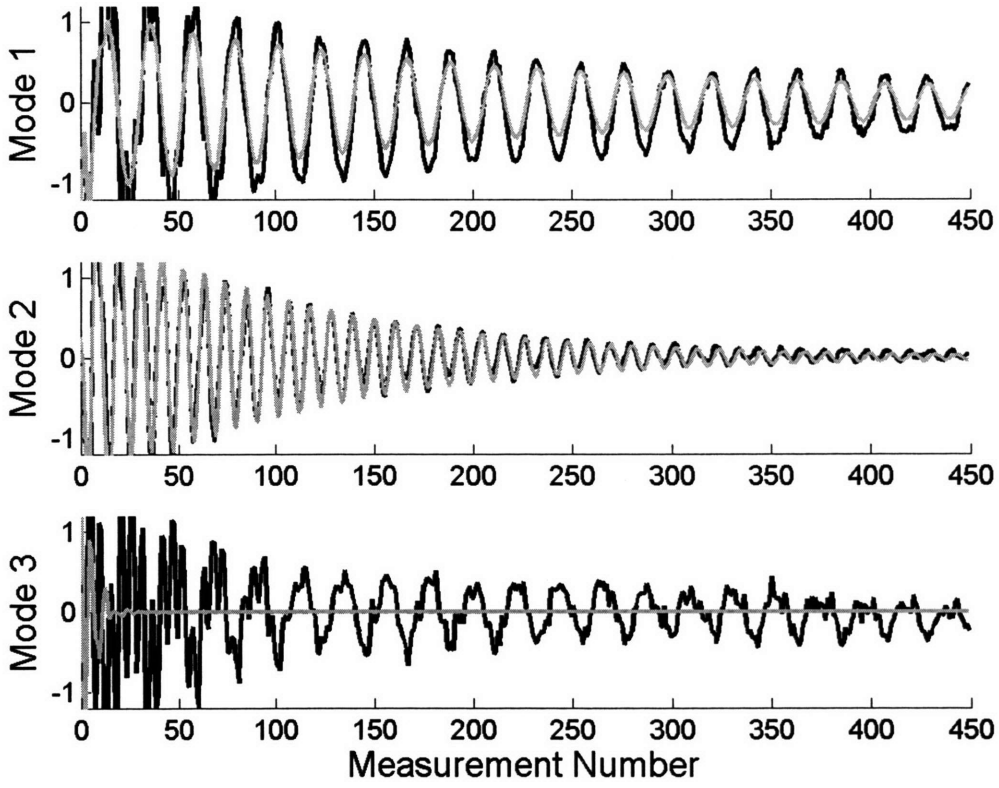


Figure 6.11. Modal coefficient measurements (black) and filtered estimates (gray) vs. time for the first three modes of vibration (experimental results).

Note that the first two modes appear to be estimated accurately, as the filtered results very closely follow the measurements in amplitude, phase, frequency, and damping. At first glance, it appears that the third mode is incorrectly estimated to be zero. However, closer inspection reveals that the surrogate measurement is dominated by a waveform having the same frequency as that of mode one, not mode three. This indicates that the mode shapes provided to the estimator were not exactly correct, and much of the a priori mode shape error happens to be aligned with the true first mode shape. In other words,

$$\left\langle \left(\hat{\Phi}_3 - \Phi_3 \right), \Phi_1 \right\rangle_Y \neq 0 .$$

During the determination of the mode shapes, the third mode was observed to damp out in less than two seconds, suggesting that the estimate shown here is correct.

Over the course of twenty experimental trials, the frequency estimates converged to $\hat{\omega} = \{4.31 \pm 0.01 \quad 8.68 \pm 0.02\}^T$ rad/s and the damping estimates converged to $\hat{\alpha} = \{0.049 \pm 0.007 \quad 0.143 \pm 0.008\}^T$ s^{-1} for the first two modes. The third mode typically damped out before the parameters could be estimated. The estimates converged in less than 90 measurements (6 seconds) for all trials. These results seem reasonable considering the fit between estimates and measurements above, and that these are within 10% of the very rough estimates provided to the estimator initially. Visual inspection also revealed good alignment of the range images with the estimated shape.

A number of experimental studies were conducted to observe the effects of poor a priori knowledge of the equilibrium configuration of the structure. Figure 6.12 shows typical experimental results for the case in which the equilibrium configuration of the structure is incorrectly provided to the estimator. It is observed that faulty equilibrium knowledge effectively biases the measurement with a DC offset, which for the example below is most obvious in mode three. Interestingly, during every experimental trial it was observed that modes one and two are still correctly estimated, despite the faulty information. The estimated frequencies and damping match those predicted in the previous experiments, and the phase and amplitude appear to match the measurements well. It should be noted, however, that the resulting shape estimates will be biased by an amount equal to the error in the a priori equilibrium shape knowledge.

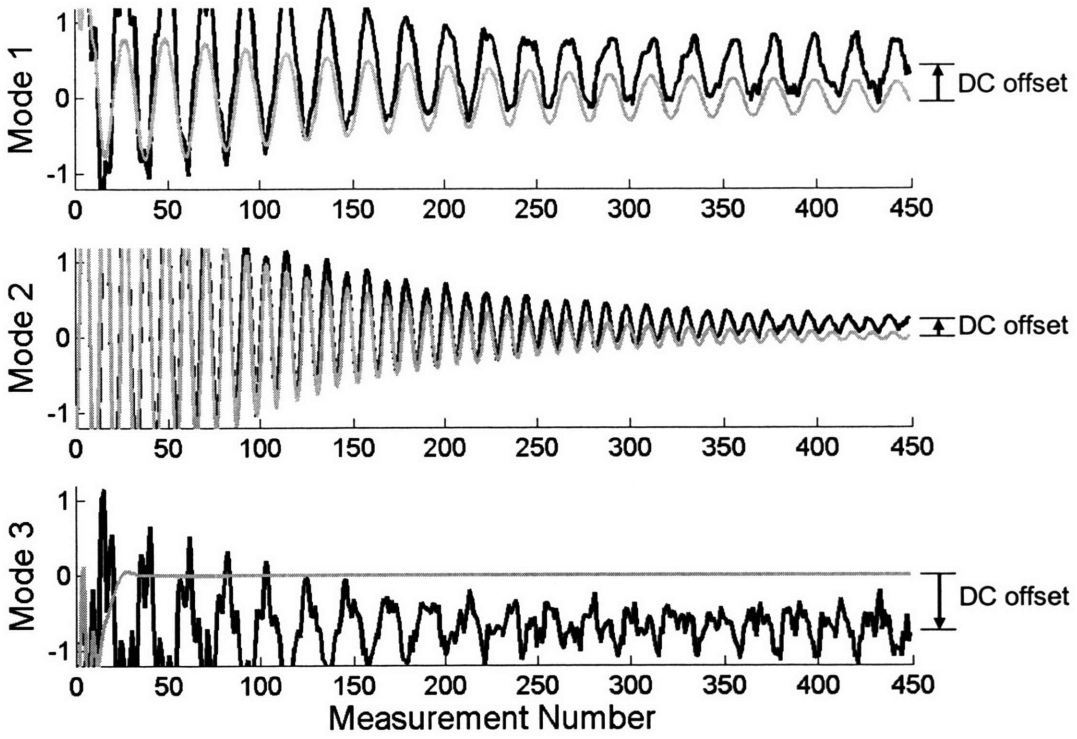


Figure 6.12. Modal coefficient measurements (black) and filtered estimates (gray) vs. time for the first three modes of vibration, when a priori knowledge of equilibrium state is poor (experimental results).

6.3.3 Computation Time

Table 6.1 shows the estimator computation time required between samples. The empirical relation describing computation time as a function of the number of range points (n) and the number of modes estimated (m) is given approximately by

$$t_{compute} \approx c_1 mn + c_2 m^3$$

where $c_1 \approx 1.4 \times 10^{-6}$ and $c_2 \approx 3.6 \times 10^{-5}$ for the studies conducted here. The cubic term is due to matrix inversions involving matrices of size proportional to m. The bilinear term is due to the inner product computation between range data and each mode shape, which involves summations over a set of size m by n.

Table 6.1. Computation time per sample, in seconds (1-GHz Intel Pentium processor).

		Number of points in range image			
		100	1000	10,000	100,000
Number of modes estimated	10	0.030	0.038	0.18	1.4
	20	0.22	0.24	0.65	4.4
	30	0.87	0.89	1.7	9.1
	40	2.1	2.2	3.5	-
	50	4.6	4.8	6.6	-

Recall that the periods of vibration for large space structures can be on the order of tens of minutes. It is practice, sensors may only need to sample at a rate on the order of once per minute in order to observe the motions of the structure. Further, it might only be necessary to observe the first ten or twenty modes in order to obtain good shape estimates. Therefore, even with space-qualified hardware, it seems that the computational requirements of the estimator are manageable for real-time implementation.

6.4 Summary

The last two chapters have developed and demonstrated a methodology for the estimation of modal vibrations using vision-type sensors. The method has exploited a key feature of space applications, which is that the dynamics of objects in space can be modeled analytically to very high accuracy. This feature has enabled an estimator design that is both accurate and robust to the challenging sensing conditions found in space.

The previous chapter developed the central theoretical methods of the estimator. A modal decomposition method was developed for partial and non-uniform visual sampling of the vibrating target. It was shown that observability of modes is easily checked by computing the condition number of a modal correlation matrix M_Y. This matrix surfaced in a number of equations, from modal decompositions to the covariance

of the resulting coefficient estimates. Kalman filtering was shown to reduce to a basic sinusoid estimation problem, and shape estimation simply involved modal reconstruction using the estimated modal coefficients. A number of special cases were then discussed from a theoretical standpoint. The effects of not estimating all excited modes was found to introduce bias to the surrogate modal measurements, which could be mitigated (among other methods) by inserting a low-pass filter between the modal decomposition and the Kalman filter. Imperfect a priori knowledge of the mode shapes was also shown to degrade shape estimates gracefully rather than catastrophically.

This chapter has explored performance of the estimator using simulation and experimental studies. Simulation studies suggested that modal amplitude, phase, frequency, and damping can all be estimated accurately even if the sensors are only viewing a small portion of the structure, and that any number of modes can be estimated as long as the Nyquist sampling criterion is met. The use of a low-pass prefilter was shown to greatly improve estimation if not all excited modes are estimated. Simple experimental studies seemed to corroborate the simulation results. Finally, it was shown that computational burden is largely a function of the number of modes estimated, and should be low enough to allow real-time implementation in practical hardware.

CHAPTER

7

CONCLUSIONS

7.1 Contributions of this Thesis

This thesis has developed an architecture for the simultaneous estimation of motion, dynamic model parameters, and shape of arbitrary space objects, using cooperative on-orbit vision sensors. The key contribution of this work is that it demonstrates that space vision systems can – and should – exploit physics-based dynamic models of their targets to improve estimator robustness and computational speed.

Chapter 1 provided motivation for the problem and discussed related literature. Future missions in space are likely to involve the robotic assembly, inspection, and maintenance of large space structures; the capture and servicing of valuable satellites; and the capture and disposal of threatening space debris. Robotic systems will require information about the motions, geometry, and dynamic model parameters of their targets. Since this information is often lacking a priori, it will need to be estimated in orbit. However, most conventional machine vision techniques, while well-suited to their particular terrestrial applications, are either not robust to the harsh lighting conditions of space or are too computationally intensive for use in space-qualified hardware.

Chapter 2 introduced a new estimation architecture for general application to space systems. This architecture decouples the shape and motion estimation problems and exploits two key features of the space applications it is intended for. These are: (1)

the dynamics of objects in space are highly deterministic and can be modeled analytically to high accuracy; and (2) several cooperative sensors are available for the collection of visual information from different perspectives. These two features enable an estimator design that is not only robust to the harsh lighting conditions of space, but is also computationally simple enough to allow real-time implementation in space-qualified hardware.

Chapter 3 provided theoretical development of this architecture in the context of rigid-body targets. It developed a computationally simple and robust *kinematic data fusion* algorithm that quickly and coarsely estimates a target's attitude and position in space relative to the observers. A Kalman filter was then developed to filter these coarse estimates and extract the full dynamic state and model parameters of the target. Finally, a simple recursive shape estimator was developed to demonstrate its function within the estimation architecture as a whole.

Chapter 4 provided simulation and experimental results for the estimation of rigid-body targets. It first described a number of Monte Carlo simulation studies used to evaluate the theoretical performance of the kinematic data fusion, Kalman filter, and shape estimator. The chapter then discussed several experimental studies used to assess feasibility and performance in practical systems. Both the simulation and the experimental studies were encouraging, demonstrating efficient, reliable performance for a variety of sensor arrangements under realistic lighting conditions. The chapter concluded with a discussion of the overall performance and feasibility of the estimator in practical systems.

Chapter 5 provided theoretical development for the estimation architecture in the context of vibrating targets. Again, it developed a kinematic data fusion methodology, which in this case involves modal decomposition. A Kalman filter was developed to estimate modal coefficients and extract frequency and damping parameters. Shape

estimation was shown to be simply a modal reconstruction. The chapter then discussed a number of practical challenges associated with the estimator and developed methods to address them.

Finally, Chapter 6 discussed a number of simulation and experimental studies used to assess the practicality and theoretical performance of the vibrating target estimator. The results demonstrated accurate estimation and reasonable computational burden over a wide range of realistic scenarios.

7.2 Suggestions for Future Work

This thesis has presented a number of simulation and experimental studies. For such a method to be implemented in real space systems it must undergo very rigorous testing on earth under the most challenging settings expected in the application. The MIT Field and Space Robotics Laboratory (FSRL) is commencing more detailed studies of this research using hardware and facilities in collaboration with its sponsor and partner, the Japan Aerospace Exploration Agency (JAXA).

Further work could be done in the estimation of rigid targets. As noted earlier, the kinematic data fusion algorithm presented here is very simple and robust but it is not intended for use with fewer than three sensors. Further research could be done to develop computationally efficient and robust algorithms for one- or two-sensor scenarios. With fewer sensors, the problem is fundamentally more challenging and a solution is not obvious at present.

The very simple shape estimator for rigid targets could be expanded or amended as necessary to incorporate more complex sensor uncertainty models. As mentioned earlier, the estimation architecture transforms a very difficult dynamic mapping problem

into a very conventional static one. Consequently, virtually any published stochastic mapping method could be applied directly to this problem with little to no alteration.

The methods developed here have assumed that range images are collected instantaneously. Some practical sensors however, notably laser range finders, employ raster scanning methods in which the first and last range points are not captured at exactly the same time. Similarly, cooperating sensors might not be synchronized, with the net result that points in the range image cloud are not all captured at the same time. The current estimator design is built on a foundation of decoupling spatial- and time-domain estimations; these sensors, however, effectively recouple space and time and therefore degrade estimator performance. Modifying the estimator to handle this recoupling would be an interesting area for future research.

REFERENCES

1 *A&M Model Makers*. http://www.ammodel.com.

2 *ABLE Engineering*. http://www.aec-able.com.

3 Aloimonos, Y. and M.J. Swain. "Shape from Texture." *Biological Cybernetics*, Vol. 58, No. 5, pp. 345-360, 1988.

4 Ayache, N.J. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*. Cambridge: MIT Press, 1991.

5 Bar-Itzhack, I.Y., and F.L. Markley. "Minimal Parameter Solution of the Orthogonal Matrix Differential Equation." *IEEE Transactions on Automatic Control*, Vol. AC-35, No. 3, pp. 314-317, 1990.

6 Besl, P.J. and N.D. McKay. "A Method for Registration of 3-D Shapes." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256, Feb. 1992.

7 Bornschlegl, E., G. Hirzinger, M. Maurette, R. Mugnuolo, and G. Visentin. "Space Robotics in Europe, a Compendium." *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics, and Automation in Space: i-SAIRAS 2003*, NARA, Japan, May 2003.

8 Bouguet, J.-Y. and P. Perona. "3D Photography Using Shadows in Dual-Space Geometry." *International Journal of Computer Vision*, Vol. 35, No. 2, pp. 129-149, Nov.-Dec. 1999.

9 Broida, T.J. and R. Chellappa. "Estimating the Kinematics and Structure of a Rigid object from a Sequence of Monocular Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, pp. 497-513, 1991.

10 Broida, T.J. and R. Chellappa. "Estimation of Object Motion Parameters from a Sequence of Noisy Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 90-99, Jan. 1986.

11 Brown, M.Z., D. Burschka, and G.D. Hager. "Advances in Computational Stereo." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 8, pp. 993-1008, Aug. 2003.

12 Brown, R.G., and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. 2nd Ed. New York: John Wiley & Sons, 1992.

13 Busboom, A. and R.J. Schalkoff. "Direct Surface Parameter Estimation using Structured Light: a Predictor-Corrector Based Approach." *Image and Vision Computing*, Vol. 14, No. 5, pp. 311-321, June 1996.

14 Carroll, R.J., D. Ruppert, and L.A. Stefanski. "Nonlinear Measurement Error Models." *Monographs on Statistics and Applied Probability*, Vol. 63, New York: Chapman and Hall, 1995.

15 Chen, C.S., Y.P. Hung, and J.B. Cheung. "Ransac-based Darces: a New Approach to Fast Automatic Registration of Partially Overlapping Range Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 11, pp. 1229-1234, Nov. 1999.

16 Chen, Z. and S.-Y. Ho. "Incremental Model Building of Polyhedral Objects using Structured Light." *Pattern Recognition*, Vol. 26, No. 1, pp. 33-46, Jan. 1993.

17 Crassidis, J.L., and F.L. Markley. "Unscented Filtering for Spacecraft Attitude Estimation." *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 4, pp. 536-542, July-August 2003.

18 de Piero, F.W. and M.M. Trivedi. "3-D Computer Vision using Structured Light: Design, Calibration, and Implementation Issues." *Advances in Computers*, Vol. 43, pp. 243-278, Academic Press, 1996.

19 Dissanayake, M.W.M.G., P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem." *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp. 229-241, Jun 2001.

20 Doucet, A., J.F.G. de Freitas, and N.J. Gordon. *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2000.

21 Faugeras, O. *Three Dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.

22 Frankot, R.T. and R. Chellappa. "A Method for Enforcing Integrability in Shape from Shading Algorithms." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 4, pp. 439-451, July 1988.

23 Fuller, W.A. *Measurement Error Models*. New York: Wiley and Sons, 1987.

24 Gårding, J. "Direct Estimation of Shape from Texture." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11, pp. 1202-1208, Nov. 1993.

25 Handschin, J.E. and D.Q. Mayne. "Monte Carlo Techniques to Estimate the Conditional Expectation in Multi-Stage Non-Linear Filtering." *International Journal of Control*, Vol. 9, pp. 547-559, 1969.

26 Hatzitheodorou, M. and J.R. Kender. "An Optimal Algorithm for the Derivation of Shape from Shadows." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 486-491, June 1988.

27 Heel, J. "Direct Dynamic Motion Vision." *Proceedings of the IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1142-47, May 1990.

28 Heel, J. "Temporal Surface Reconstruction." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 607-612, June 1991.

29 *High Resolution Stereo Camera on Mars Express*. http://berlinadmin.dlr.de/Missions/express/indexeng.shtml.

30 Horn, B.K.P. and B.G. Schunck. "Determining Optical Flow." *AI*, Vol. 17, No. 1-3, pp. 185-203, August 1981.

31 Horn, B.K.P. and J.G. Harris. "Rigid Body Motion from Range Image Sequences." *Computer Vision, Graphics, and Image Processing (CVGIP): Image Understanding,* Vol. 53, No. 1, Jan, pp. 1-13, 1991.

32 Horn, B.K.P. and M.J. Brooks, eds. *Shape from Shading*. Cambridge, Massachusetts: MIT Press, 1989.

33 Howard, I.P. and B.J. Rogers. *Binocular Vision and Stereopsis*. Oxford University Press, 1995.

34 Ikeuchi, K. "Shape from Regular Patterns." *Artificial Intelligence*, Vol. 22, No. 1, pp. 49-75, Jan. 1984.

35 Ito, K., and K. Xiong. "Gaussian Filters for Nonlinear Filtering Problems." *IEEE Transactions on Automatic Control*, Vol. 45, pp. 910-927, 2000.

36 Jacobsen, S., C. Lee, C. Zhu, and S. Dubowsky. "Planning of Safe Kinematic Trajectories for Free Flying Robots Approaching an Uncontrolled Spinning Satellite." *Proceedings of the ASME 27th Annual Biennial Mechanisms and Robotics Conference*, Montreal, Canada, September 2002.

37 Jasiobedzki, P., et al. "Model Based Pose Estimation for Autonomous Operations in Space." *Proceedings of the IEEE 1999 International Conference on Information Intelligence and Systems*, Rockville, Maryland, pp. 211-215, April 1999.

38 Jasiobedzki, P., M. Greenspan, and G. Roth. "Pose Determination and Tracking for Autonomous Satellite Capture." *Proceedings of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space: i-SAIRAS 2001*, Canadian Space Agency, St-Hubert, Quebec, Canada, June 2001.

39 Jenkin, M. and P. Jasiobedzki. "Computation of Stereo Disparity for Space Materials." *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 1998.

40 Julier, S.J., and J.K. Uhlmann. "A New Extension of the Kalman Filter to Nonlinear Systems." *Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando, Florida, 1997.

41 Kanatani, K., and T. Chou. "Shape from Texture: General Principle." *AI*, Vol. 38, pp. 1-48, 1989.

42 Kawamoto, S., K. Matsumoto, and S. Wakabayashi. "Ground Experiment of Mechanical Impulse Method for Uncontrollable Satellite Capturing." *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space: i-SAIRAS 2001*, St-Hubert, Quebec, Canada, June 2001.

43 Kim, Y.C., and J.K. Aggarwal. "Determining Object Motion in a Sequence of Stereo Images." *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 6, pp. 599-614, December 1987.

44 Kraft, E. "Ein Sensorsystem zur Bestimmung räumlicher Orientierung in Echtzeit". (A Sensor System for the Determination of Spatial Orientations in Real Time.) Thesis Diploma (in German), Universität Bonn, Physikalisches Institut, Bonn, Germany, August 2002.

45 Kraft, E. "Quaternion-based Unscented Kalman Filter for Orientation Tracking." *Proceedings of the 6th International Conference on Information Fusion*, Cairns, Australia, pp. 47-54, July 2003.

46 Lee, K.M. and C.-C.J. Kuo. "Shape from Shading with a Generalized Reflectance Map Model." *Computer Vision and Image Understanding*, Vol. 67, No. 2, pp. 143-160, Aug. 1997.

47 Lefferts, E.J., F.L. Markley, and M.D. Shuster. "Kalman Filtering for Spacecraft Attitude Estimation." *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 5, pp 417-429, September-October 1982.

48 Leonard, J.J. and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Boston, MA: Kluwer Academic, 1992.

49 Lichter, M.D. and S. Dubowsky. "Estimation of State, Shape, and Inertial Parameters of Space Objects from Sequences of Range Images." *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision*, Providence, RI, Vol. 5267, October 2003.

50 Lichter, M.D. and S. Dubowsky. "State, Shape, and Parameter Estimation of Space Objects from Range Images." *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, pp. 2974-2979, April 2004.

51 Mangalgiri, V. "Analysis for the Robotic Assembly of Large Flexible Space Structures." Master's Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2004.

52 Mankins, J.C. "A Fresh Look at Space Solar Power: New Architectures, Concepts and Technologies." *Proceedings of the 38th International Astronautical Federation Conference*, 1997.

53 Markley, F.L. "Attitude Error Representations for Kalman Filtering." *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 2, pp. 311-317, March-April 2003.

54 Marr, D. and T. Poggio. "A Computational Theory of Human Stereo Vision." *Proceedings of the Royal Society of London*, B204, pp. 301-328, 1979.

55 Matsumoto, S., S. Dubowsky, S. Jacobsen, and Y. Ohkami. "Fly-by Approach and Guidance for Uncontrolled Rotating Satellite Capture." *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Austin, TX, August 2003.

56 Mulligan, J., V. Isler, and K. Daniilidis, K. "Trinocular Stereo: A Real-Time Algorithm and its Evaluation." *International Journal of Computer Vision*, Vol. 47, No. 1-3, pp. 51-61, April-June 2002.

57 Nagamatsu, H., T. Kubota, and I. Nakatani. "Capture Strategy for Retrieval of a Tumbling Satellite by a Space Robotic Manipulator." *Proceedings of the 1996 IEEE International Conference on Robotics and Automation (ICRA 19964)*, Minneapolis, MN, pp. 70-75, April 1996.

58 *NASA Human Space Flight*. http://www.spaceflight.nasa.gov.

59 Neish, M.J. and T. Goka. "Costs and Benefits of Space Debris Mitigation." *Proceedings of the 3rd European Conference on Space Debris*, Darmstadt, Germany, March 2001.

60 Nørgaard, M., N.K. Poulsen, and O. Ravn. "Advances in Derivative-Free State Estimation for Nonlinear Systems." *Technical Report IMM-REP-1998-15*, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, 1998.

61 Oda, M., H. Ueno, and M. Mori. "Study of the Solar Power Satellite in NASDA." *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics, and Automation in Space: i-SAIRAS 2003*, NARA, Japan, May 2003.

62 Piedboeuf, J.-C., and É. Dupuis. "Recent Canadian Activities in Space Automation & Robotics – An Overview." *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics, and Automation in Space: i-SAIRAS 2003*, NARA, Japan, May 2003.

63 Ronen, M., and Y. Oshman. "Third-Order, Minimal-Parameter Solution of the Orthogonal Matrix Differential Equation." *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 3, pp. 516-521, May-June 1997.

64 *Servo-To-Go, Inc.* http://www.servotogo.com.

65 Shafer, S.A. *Shadows and Silhouettes in Computer Vision.* Kluwer Academic Publishers, 1985.

66 Shannon, C. E. "A Mathematical Theory of Communication." *The Bell System Technical Journal*, Vol. 27, pp. 379-423 and 623-656, July and Oct. 1948.

67 Shieh, J.-Y., H. Zhuang, and R. Sudhakar. "Motion Estimation from a Sequence of Stereo Images: A Direct Method." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 7, pp. 1044-1053, July 1994.

68 Shreiner, D., M. Woo, J. Nieder, and T. Davis. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.4.* 4th ed., Addison-Wesley, Reading, MA, 2004.

69 *Shuttle Mission Summary: STS-108.* http://www-pao.ksc.nasa.gov/kscpao/shuttle/summaries/sts108/.

70 Sidi, M.J. *Spacecraft Dynamics and Control: a Practical Engineering Approach.* New York: Cambridge University Press, 1997.

71 Smith, R.C., and P. Cheeseman. "On the Representation and Estimation of Spatial Uncertainty." *International Journal of Robotics Research*, Vol. 5, No. 4, pp. 56-68, 1987.

72 Soatto, S., R. Frezza, and P. Perona. "Motion Estimation via Dynamic Vision." *IEEE Transactions on Automatic Control*, Vol. 41, No. 3, March 1996, pp. 393-413.

73 Soucy, G., F.G. Callari, and F.P. Ferrie. "Estimating Pose Through Local Geometry." *Proceedings of the IEEE Second International Conference on 3-D Imaging and Modeling*, Ottawa, Canada, pp. 220-228, October 1999.

74 *SRI Small Vision System*. http://www.ai.sri.com/~konolige/svs/svs.htm.

75 Staritz, P.J., S. Skaff, C. Urmson, and W. Whittaker. "Skyworker: A Robot for Assembly, Inspection and Maintenance of Large Scale Orbital Facilities." *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA 2001)*, Seoul, Korea, pp. 4180-4185, May 2001.

76 Stenstrom, J.R., and C.I. Connolly. "Constructing Object Models from Multiple Images." *International Journal of Computer Vision*, Vol. 9, No. 3, pp. 185-212, 1992.

77 Sujan, V., and S. Dubowsky. "Visually Built Task Models for Robot Teams in Unstructured Environments," *Proceedings of 2002 IEEE International Conference on Robotics and Automation (ICRA 2002)*, Washington, D.C., May 11-May 15, 2002.

78 Sujan, V.A. "Compensating for Model Uncertainty in the Control of Cooperative Field Robots." Ph.D. Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2002.

79 Ueno, H., S. Dubowsky, C. Lee, C. Zhu, Y. Ohkami, S. Matsumoto, and M. Oda. "Space Robotic Mission Concepts for Capturing Stray Objects." *Proceedings of the 23rd International Symposium on Space Technology and Science*, Matsue, Japan, May 2002.

80 Ueno, H., T. Nishimaki, M. Oda, and N. Inaba. "Autonomous Cooperative Robots for Space Structure Assembly and Maintenance." *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics, and Automation in Space: i-SAIRAS 2003*, NARA, Japan, May 2003.

81 Vega, O.E. and Y.H. Yang. "Shading Logic: A Heuristic Approach to Recover Shape from Shading." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 6, pp. 592-597, June 1993.

82 Vergauwen, M., M. Pollefeys, and L.J. Van Gool. "A Stereo-vision System for Support of Planetary Surface Exploration." *Machine Vision and Applications*, Vol. 14, No. 1, pp. 5-14, 2003.

83 *Videre Design*. http://www.videredesign.com.

84 Wakabayashi, Y., Y. Ohkami, M. Miyata, T. Adachi, and T. Iijima. "A Compact Laser Range Finder for Space Applications." *Proceedings of SPIE, Vol. 3714: Enabling Photonic Technologies for Aerospace Applications*, A.R. Pirich, E.W. Taylor, eds., pp. 131-138, 1999.

85 Wan, E.A., and R. van der Merwe. "The Unscented Kalman Filter." *Kalman Filtering and Neural Networks*, pp. 221-280, Edited by Simon Haykin, New York: Wiley Publishing, 2001.

86 Yang, Z. and Y.F. Wang. "Error Analysis of 3D Shape Construction from Structured Lighting." *Pattern Recognition*, Vol. 29, No. 2, pp. 189-206, Feb. 1996.

87 Yau, W.Y. and H. Wang. "Fast Relative Depth Computation for an Active Stereo Vision System." *Real Time Imaging*, Vol. 5, No. 3, pp. 189-202, June 1999.

88 Yoshida, K. and S. Hirose. "Real-Time Stereo Vision with Multiple Arrayed Camera." *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1992)*, Nice, France, pp.1765-1770, May 1992.

89 Young, G.-S.J. and R. Chellappa. "3-D Motion Estimation Using a Sequence of Noisy Stereo Images: Models, Estimation, and Uniqueness Results." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 8, pp. 735-759, Aug 1990.

90 Zhang, Z.Y. "Iterative Point Matching for Registration of Free-form Curves and Surfaces." *International Journal of Computer Vision*, Vol. 13, No. 2, pp. 115-19, Oct. 1994.

91 Zhuang, H., and R. Sudhakar. "Simultaneous Rotation and Translation Fitting of Two 3-D Point Sets." *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 27, No. 1, pp. 127-131, Feb 1997.

APPENDIX

A

UNSCENTED FILTERING OF QUATERNIONS

A.1 Background

The use of Kalman filters to estimate unit quaternions has been studied for several decades now. The classic problem with estimating quaternions in a Kalman filter framework is that the unit quaternion has a normality constraint which reduces its degrees of freedom to three.

Early approaches treated the unit quaternion as an element of $\Re^4$ and used the standard Kalman filter equations. However, this approach is suboptimal for several reasons. Fundamentally, the unit quaternion resides in S^3, not $\Re^4$. As such, component-wise vector additions (e.g. in the Kalman update step) will lead to loss of normality in resulting quaternions (i.e. the result will not be a member of S^3). Further, computation of the covariance matrix of a unit quaternion by treating it as an element of $\Re^4$ leads to a nearly singular 4-by-4 matrix, and does not capture the essential fact that the quaternion has only three degrees of freedom. Early solutions to these problems often involved brute-force renormalization at each time step and linear projections of the 4-by-4 covariance matrix onto a 3-by-3 space. However, these approaches were sub-optimal at best and often not robust.

In 1982, a breakthrough method was conceived that appropriately maintained the normality constraints and naturally provided 3-by-3 covariance matrices [47]. This

method, named the Multiplicative Extended Kalman Filter (MEKF), attacked the root of the problem by using quaternion multiplications (i.e. rotation "additions") rather than component-wise vector additions. Small rotations were mapped smoothly and without loss of information to an unconstrained $\Re^3$ space, which was compatible with the linear update step in Kalman filters. This minimum parameter representation was used for noise variables, measurement and state errors, and state updates involving unit quaternions. Covariance structures were then computed based on this minimum order representation, naturally yielding 3-by-3 covariance structures. This new method performed reliably and near-optimally compared to all previous methods. However, it still possessed the classic limitations of the extended Kalman filter (EKF), including poor performance during the initialization of the filter when state errors are large [12].

In the late 1990's, the unscented filter (UF) was conceived and developed [40, 85]. This provided a new method for propagating covariance estimates for systems with nonlinear models. The UF was generally much easier to implement than the EKF since it did not require the derivation or computation of Jacobians. It was also accurate to higher order and more robust during initialization than the EKF, and only nominally more computation intensive.

It was only shortly thereafter that several researchers, including the author of this thesis, independently linked the UF and the MEKF. The marriage of these techniques led to the first fundamentally sound linear estimator for quaternions. Edgar Kraft of the University of Bonn, Germany, appears to have the first formal documentation of the solution in his diploma thesis, published in August 2002 [44]. Kraft's first English publication of the work came in July 2003 [45]. Surprisingly, an identical solution was published by John L. Crassidis and F. Landis Markley, also in July 2003 [17]. This work was submitted for review in April 2002, well before the Kraft thesis was finished, indicating these were probably independent discoveries. Interestingly, the author of this

thesis had also derived the solution in September 2002, well before the English texts were published and without knowledge of the Kraft thesis.

This appendix will now describe the implementation of a quaternion-based unscented Kalman filter in the context of the rigid body estimation problem described in Chapter 3. This chapter is intended to support and complement the discussion found in the literature [17, 44, 45, 47, 53]. The derivation takes a somewhat different philosophical perspective than the literature, but the final solution is mathematically identical. Further, the solution is provided here in a more algorithmic manner that may simplify implementation for some readers.

A.2 Operations on Spatial Rotations

A.2.1 The Rotation Vector Parameterization

A minimum-order rotation vector $\vec{p}$ is introduced to describe rotation errors, perturbations, and increments smaller than 2π radians. This is just one of many possible three-dimensional parameterizations of rotation[1], but has several important properties that will make it useful in later sections of this appendix. The rotation vector is defined by a scalar angle ϕ and a rotation axis $\vec{e}$ of unit length:

$$\vec{p} = \phi \cdot \vec{e} \,.$$

The rotation vector lies in the unconstrained Euclidean space $\Re^3$, making it compatible with the typical Kalman filter. The magnitude of $\vec{p}$ defines the angle of

[1] Reference [53] provides a good discussion of several other useful parameterizations in addition to the rotation vector.

rotation. Scaling $\vec{p}$ scales the amount of rotation proportionally, making it is useful in interpolations. In terms of the quantities ϕ and $\vec{e}$, the unit quaternion can be written as

$$\vec{q} = \begin{Bmatrix} \cos(\phi/2) \\ e_1 \sin(\phi/2) \\ e_2 \sin(\phi/2) \\ e_3 \sin(\phi/2) \end{Bmatrix}.$$

In this way, a mapping can be defined from quaternion to rotation vector that is one-to-one and invertible for all rotations less than 2π radians. The mapping will be denoted in shorthand as

$$p(\vec{q}) \equiv \frac{\phi}{\sin(\phi/2)} \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix}$$

$$q(\vec{p}) \equiv p^{-1}(\vec{p}) \equiv \begin{Bmatrix} \cos(\phi/2) \\ \frac{\sin(\phi/2)}{\phi} p_1 \\ \frac{\sin(\phi/2)}{\phi} p_2 \\ \frac{\sin(\phi/2)}{\phi} p_3 \end{Bmatrix}$$

where $\phi = 2 \cdot \cos^{-1}(q_0) = \|\vec{p}\|$.

A.2.2 Rotation Addition and Subtraction Operators

The filter to be developed will require the addition and subtraction of small incremental rotations from a rotation estimate (typical parameterized by a unit quaternion). For convenience, a set of addition and subtraction operators will be defined here for spatial rotations. The purpose will become clear later as the filter is developed.

A rotation addition operator $\oplus$ is defined here as

$$\vec{q}_a \oplus \vec{p}_c \equiv \vec{q}_a \otimes q(\vec{p}_c) \tag{A.1}$$

which simply means that addition of spatial rotations is the (non-commutative) concatenation of the two rotations. Note that the first operand is a unit quaternion and the second is a rotation vector, and the result is a unit quaternion. Recall that the quaternion multiplication operator $\otimes$ was defined in Equation (3.4) on page 38.

Likewise, a rotation subtraction operator $\ominus$ is defined here as

$$\vec{q}_b \ominus \vec{q}_a \equiv p(\vec{q}_a^{-1} \otimes \vec{q}_b) \tag{A.2}$$

which is defined so that the following statements are equivalent:

$$\vec{q}_b \ominus \vec{q}_a = \vec{p}_c \quad \Leftrightarrow \quad \vec{q}_a \oplus \vec{p}_c = \vec{q}_b .$$

Note that the operands of the rotation subtraction operator are unit quaternions, and the result is a rotation vector. Recall that the inverse of a unit quaternion is computed by negating the last three elements.

A.2.3 Rotation Covariance

For Kalman filtering, the concept of a covariance matrix needs to be defined for rotations. Note that it is incorrect to treat the unit quaternion as a vector in $\Re^4$ and compute its covariance the traditional way:

$$\Lambda_{qq} \neq E\left[(\vec{q} - E[\vec{q}])(\vec{q} - E[\vec{q}])^T\right] .$$

Using this naïve approach would yield a 4-by-4 matrix, incorrectly indicating that the rotation possesses four degrees of freedom.

Instead, the covariance of a random rotation, parameterized by a unit quaternion $\vec{q}$, will be computed as

$$\Lambda_{qq} \equiv E\left[(\vec{q} \Theta E[\vec{q}])(\vec{q} \Theta E[\vec{q}])^T\right] \tag{A.3}$$

which is a 3-by-3 matrix. The order of terms in this equation is critical and should not be commuted. Mathematically, this is a much different equation than the one for Euclidean random variables. At a conceptual level, however, it has the same meaning.

A.2.4 Rotation Expectations and Averages

The expectation of a random rotation, parameterized by $\vec{q}$, can be defined as

$$E[\vec{q}] \equiv \arg\min_{\vec{a}} \left\| \; E(\vec{q} \Theta \vec{a}) \; \right\|$$

The weighted mean of several rotations, parameterized by unit quaternions, can be computed as the normalized weighted vector sum. Here the summing notation implies ordinary component-wise vector addition.

$$mean(\vec{q}_1, \vec{q}_2, ..., \vec{q}_N) \equiv \frac{\sum_{i=1}^{N} weight_i * \vec{q}_i}{\left\| \sum_{i=1}^{N} weight_i * \vec{q}_i \right\|}. \tag{A.4}$$

Due to the two-to-one nature of the mapping from unit quaternion to spatial rotations, this equation is only valid for quaternions within the same hemisphere on the

unit quaternion hypersphere[1]. Therefore before averaging, quaternions should be reflected onto the correct hemisphere if necessary by negating each of their elements.

A direct result of Equation (A.4) is that if a unit quaternion distribution resides on a single hemisphere, its expectation can be computed easily by taking its $\Re^4$ expectation and normalizing. This condition is applicable for this problem and thus the computation of quaternion expectations is greatly simplified.

A.2.5 Analogy in the Complex Plane

While all of this may be difficult to visualize due to high dimensionality, there is an obvious analogy in a more comprehensible space. Consider the 2-D complex plane[2]. Let $\vec{c}$ be a random complex number of unit length that parameterizes planar rotations. By its normality constraint, it has only one degree of freedom, and can be alternatively parameterized to minimum order by the angle b (see Figure A.1).

[1] To determine if two quaternions are in the same hemisphere, simply multiply one by the inverse of the other. The first element of the resulting quaternion will be positive if they reside on the same hemisphere.

[2] Technically the complex plane is just a subspace of the quaternion space.

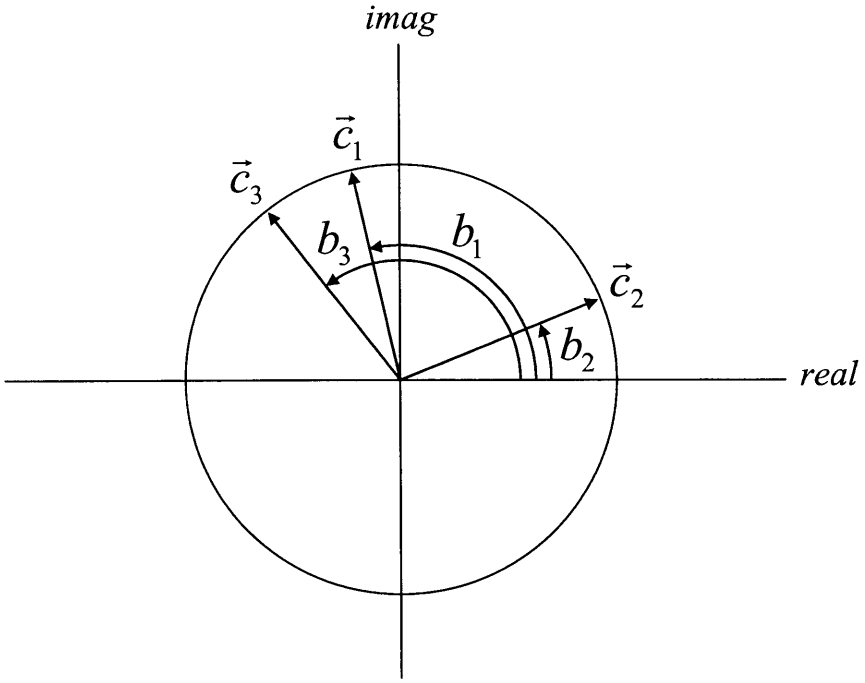


Figure A.1. Two-dimensional analogy for the unit quaternion and rotation vector.

The addition of two planar rotations parameterized by $\vec{c}_1$ and $\vec{c}_2$ is not accomplished by a component-wise addition ($\vec{c}_3 \neq \vec{c}_1 + \vec{c}_2$), but rather involves their corresponding angle parameterizations b_1 and b_2. Analogous to Equation (A.1), this can be mathematically described as

$$\vec{c}_3 = \vec{c}_1 \oplus b_2 = \vec{c}_1 \otimes \vec{c}_2$$

where the $\otimes$ indicates ordinary multiplication in the complex plane.

The expectations and averages of planar rotations can be computed as the normalized $\Re^2$ expectations and averages of their unit complex vector parameterizations, analogous to Equation (A.4).

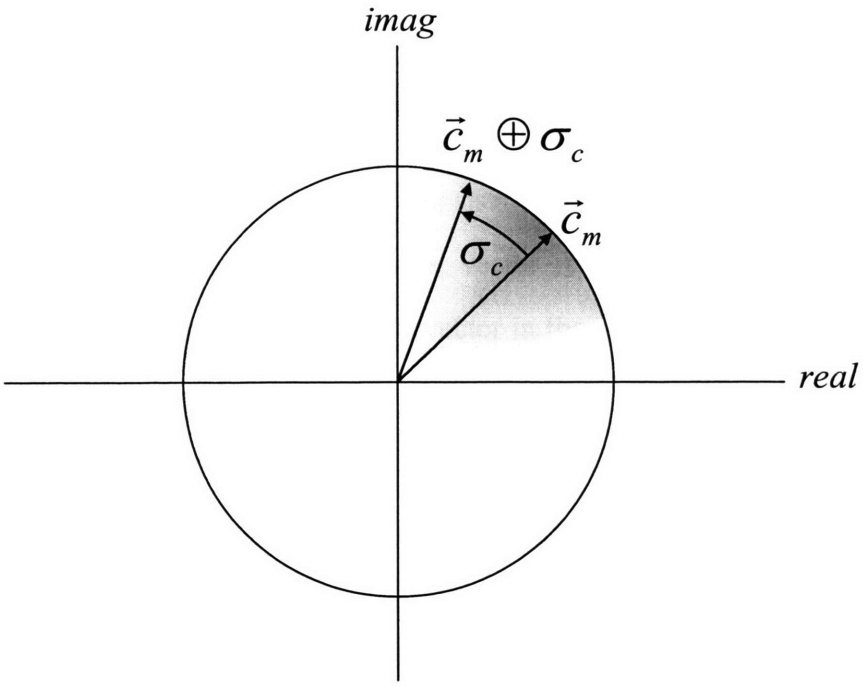


Figure A.2. Mean and standard deviation of a random planar rotation parameterized by a unit complex vector.

Figure A.2 shows a distribution of planar rotations, where the darker region indicates higher probability density. The mean rotation, parameterized by a unit complex vector, is denoted by $\vec{c}_m$. The covariance of a random planar rotation does not involve its parameterization as a complex unit vector, but rather involves the fundamental degree of freedom in the b-coordinate. The 1-by-1 covariance of planar rotations, parameterized by a unit complex vector $\vec{c}$, is given by

$$\Lambda_{cc} \equiv E\left[(\vec{c} \ominus E[\vec{c}])(\vec{c} \ominus E[\vec{c}])^T\right] = E\left[b\left((E[\vec{c}])^{-1} \otimes \vec{c}\right)^2\right]$$

which is directly analogous to Equation (A.3). Note that the standard deviation $\sigma_c \equiv \sqrt{\Lambda_{cc}}$ has the units of b-vectors (radians), not $\vec{c}$-vectors.

A.3 Operations on the State and Measurement Space

A.3.1 The Non-Euclidean State Space

The state vector to be estimated consists of a 3-D Euclidean vector and three spatial rotations. In other words, the state vector resides in $\Re^3 \times S^3 \times S^3 \times S^3$. The globally nonsingular parameterization of a vector in the state space uses quaternions and is denoted with superscript (q):

$$\vec{x}^{(q)} \equiv \begin{Bmatrix} \vec{\omega} \\ \vec{q}_g \\ \vec{q}_d \\ \vec{q}_I \end{Bmatrix} \quad \text{(15-by-1)}.$$

The minimum-order parameterization of a vector in the state space uses the rotation vector parameterization and is denoted with the superscript (p):

$$\vec{x}^{(p)} \equiv \begin{Bmatrix} \vec{\omega} \\ \vec{p}_g \\ \vec{p}_d \\ \vec{p}_{I(1,2)} \end{Bmatrix} \quad \text{(11-by-1)}.$$

Only the first two elements of $\vec{p}_I$ are included, since exactly these two degrees of freedom achieve all possible relative inertias. The third element is defined to be zero for all time:

$$p_{I3} \equiv 0 .$$

Addition and subtraction operators in the state space are now defined as follows:

$$\vec{x}_b^{(q)} = \vec{x}_a^{(q)} \oplus \vec{x}_c^{(p)} \equiv \begin{Bmatrix} \vec{\omega}_a + \vec{\omega}_c \\ \vec{q}_{ga} \oplus \vec{p}_{gc} \\ \vec{q}_{da} \oplus \vec{p}_{dc} \\ \vec{q}_{Ia} \oplus \vec{p}_{Ic} \end{Bmatrix}$$

$$\Updownarrow \qquad . \qquad \text{(A.5)}$$

$$\vec{x}_c^{(p)} = \vec{x}_b^{(q)} \Theta \vec{x}_a^{(q)} \equiv \begin{Bmatrix} \vec{\omega}_b - \vec{\omega}_a \\ \vec{q}_{gb} \Theta \vec{q}_{ga} \\ \vec{q}_{db} \Theta \vec{q}_{da} \\ (\vec{q}_{Ib} \Theta \vec{q}_{Ia})_{(1,2)} \end{Bmatrix} \equiv \begin{Bmatrix} \vec{\omega}_c \\ \vec{p}_{gc} \\ \vec{p}_{dc} \\ \vec{p}_{Ic(1,2)} \end{Bmatrix}$$

Note the equivalence of these statements, and their relation to Equations (A.1) and (A.2). Also note the parameterizations of the state vector expected by the operators.

A.3.2 The Non-Euclidean Measurement Space

As discussed in Section 3.3.3, the measurement vector is a spatial rotation, and therefore resides in S^3. The globally nonsingular parameterization of a vector in the measurement space is denoted with superscript (q) and is given by

$$\vec{y}^{(q)} \equiv \{\vec{q}_m\} \quad \text{(4-by-1)}.$$

The minimum-order parameterization of a vector in the measurement space is denoted with superscript (p) as

$$\vec{y}^{(p)} \equiv \{\vec{p}_m\} \quad \text{(3-by-1)}.$$

Like the state vector, addition and subtraction operators in the measurement space will be defined by the equivalent statements

$$\vec{y}_b^{(q)} = \vec{y}_a^{(q)} \oplus \vec{y}_c^{(p)} \equiv \{\vec{q}_{ma} \oplus \vec{p}_{mc}\}$$

$$\Updownarrow \qquad . \qquad \text{(A.6)}$$

$$\vec{y}_c^{(p)} \equiv \vec{y}_b^{(q)} \Theta \vec{y}_a^{(q)} \equiv \{\vec{q}_{mb} \Theta \vec{q}_{ma}\} \equiv \{\vec{p}_{mc}\}$$

Again, note the parameterizations expected by the operators, and the equivalence of these equations.

A.3.3 Process and Measurement Models

The dynamics of the state vector can now be written by combining the dynamics and all the mappings presented in Section 3.3, Equations (3.4)-(3.12). Let the process model of the state vector be denoted in shorthand by the discrete-time[1] model

$$\vec{x}_{k+1} \equiv f(\vec{x}_k) \oplus \vec{v} \qquad \text{(A.7)}$$

where $\vec{v}$ indicates process noise, which may vary depending on application. Similarly, let the discrete-time measurement model of the state vector be denoted in shorthand by

$$\vec{y}_k \equiv h(\vec{x}_k) \oplus \vec{w} = \vec{q}_{gk} \oplus \vec{w} \qquad \text{(A.8)}$$

where $\vec{w} \equiv p(\vec{q}_w)$ is the rotation vector parameterization of measurement error (see Equation (3.7), page 40).

[1] The continuous-time models presented by Equations (3.6)-(3.12) are smooth and easily solved online via numerical integration to yield the discrete-time model.

A.4 Unscented Kalman Filter Implementation

Due to the nonlinearities in the process and measurement models, the standard linear Kalman filter is not appropriate here. One extension of the Kalman filter for nonlinear models is the extended Kalman filter (EKF) [12]. However, implementation of the EKF is difficult here, since it requires analytical expressions for the Jacobians of Equations (A.7) and (A.8). Convergence is also weak or unstable if the parameter estimates are initialized with large error, which may occur commonly in practice. Due to its relative speed, simplicity, and demonstrated capabilities in estimation problems involving nonlinear models, the unscented Kalman filter (UKF) will be used instead [40, 85].

The UKF consists of several parts as depicted in Figure A.3. First, $(2N+1)$ points are sampled from the state space near the current a posteriori state estimate, where N is the degrees of freedom in the state vector. This sampling is deterministic and the distribution of these points is chosen so that their mean and covariance match the current state estimate and covariance. In this way, the UKF superficially resembles a particle filter; however, the points are sampled deterministically and are relatively few in number. Next, the sample points (referred to as *sigma points* in the literature) are propagated through the nonlinear process and measurement models (Equations (A.7) and (A.8)) to yield a priori (pre-update) state and measurement distributions at the next time step. These points are referred to as the *a priori sigma points*. Finally, the sample means and covariances of the a priori sigma points are computed to yield the a priori covariances, state, and measurement estimates. This information is then used in conjunction with the next measurement to perform the typical Kalman update operations.

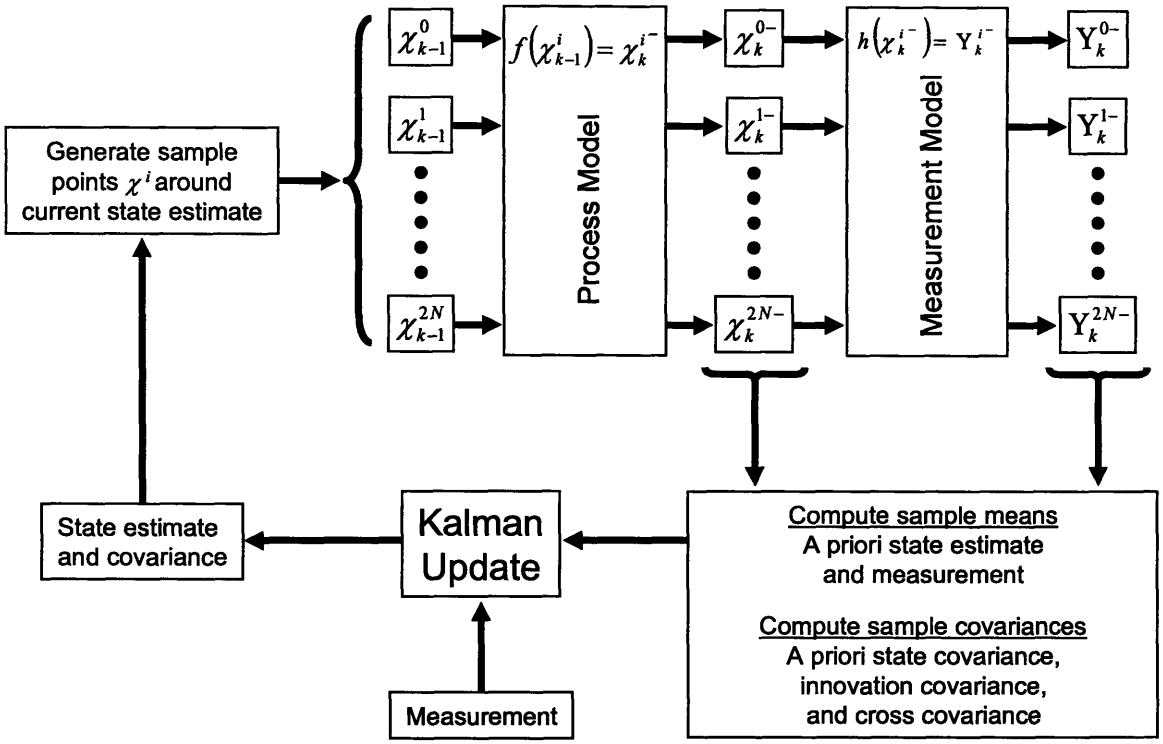


Figure A.3. Schematic for the Unscented Kalman Filter.

A.4.1 Sigma Point Generation and Propagation

The first step is to generate the set of a posteriori (post-update) sample points (sigma points) used by the unscented transform [40, 85]. First, the matrix square root of the a posteriori covariance estimate is computed[1]. Any choice of matrix square root is sufficient; here, the Cholesky decomposition method is fast and robust. The sigma points are then computed as the current estimate plus or minus each column of the square root matrix. In essence, a symmetric set of sigma points are being created that lie one standard-deviation from the current estimate. That is,

[1] Because measurement and process noise are nonlinear, one must incorporate the state augmentation methods described in [85]. For brevity, this implementation detail is not discussed here.

$$\chi_{k-1}^{i} = \begin{cases} \hat{x}_{k-1} & i = 0 \\ \hat{x}_{k-1} \oplus \left(+\sqrt{\Lambda_{xx}}\right)_{i^{th}\,column} & 0 < i \le N \\ \hat{x}_{k-1} \oplus \left(-\sqrt{\Lambda_{xx}}\right)_{(i-N)^{th}\,column} & N < i \le 2N \end{cases} \tag{A.9}$$

where χ^i denotes the i^{th} sigma point, $\hat{x}$ denotes the a posteriori (post-update) state estimate, and Λ_{xx} denotes the a posteriori state covariance. The time subscript $(k-1)$ has been omitted on some terms for clarity. The $\oplus$ operator is used here as a reminder that Equation (A.5) must be used for summing operations in the state space.

These points are now propagated through the state and measurement equations to achieve two new sets of points representing the distributions of the a priori state estimate and the a priori measurement prediction.

$$\chi_k^{i^-} = f\left(\chi_{k-1}^{i}\right) \qquad \mathrm{Y}_k^{i^-} = h\left(\chi_k^{i^-}\right) \tag{A.10}$$

A.4.2 Recombination of the Sigma Points

The next step is to compute the a priori (pre-update) state estimate and measurement prediction as the weighted mean of their respective sigma points[1]. It is critical that the appropriate averaging methods are used. The ω-component of the state vector is simply an $\Re^3$ average, but all the unit quaternions must be averaged using Equation (A.4). The a priori state estimate and measurement prediction are given by

$$\begin{aligned} \hat{x}_k^- &= weighted \cdot average \cdot of \cdot \chi_k^{i^-} \\ \hat{y}_k^- &= weighted \cdot average \cdot of \cdot \mathrm{Y}_k^{i^-} \end{aligned} \tag{A.11}$$

[1] Weights are defined in [40, 85].

where the minus (-) superscript indicates the a priori estimate. The a priori state covariance, innovation covariance, and cross covariance matrices are computed as

$$\Lambda_{xx\,k}^{-} = \sum_{i=0}^{2N} weight_i \left(\chi_k^{i\,-} \Theta \hat{x}_k^{-}\right)\left(\chi_k^{i\,-} \Theta \hat{x}_k^{-}\right)^T \quad \text{(11-by-11)}$$

$$\Lambda_{yy\,k} = \sum_{i=0}^{2N} weight_i \left(\mathrm{Y}_k^{i\,-} \Theta \hat{y}_k^{-}\right)\left(\mathrm{Y}_k^{i\,-} \Theta \hat{y}_k^{-}\right)^T \quad \text{(3-by-3)} \qquad \text{(A.12)}$$

$$\Lambda_{xy\,k} = \sum_{i=0}^{2N} weight_i \left(\chi_k^{i\,-} \Theta \hat{x}_k^{-}\right)\left(\mathrm{Y}_k^{i\,-} \Theta \hat{y}_k^{-}\right)^T \quad \text{(11-by-3)}$$

where the summing notation denotes ordinary component-wise addition of matrices. The Θ operator is a reminder that Equations (A.5) and (A.6) must be used when taking differences of state and measurement vectors. Because operations are non-commutative, the order presented here is critical.

A.4.3 Kalman Update

The state estimate and covariance is now updated as

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^{-} \oplus K_k \left(\vec{y}_k \Theta \hat{y}_k^{-}\right) \\ \Lambda_{xx\,k} &= \Lambda_{xx\,k}^{-} - K_k \Lambda_{yy\,k} K_k^T \end{aligned} \qquad \text{(A.13)}$$

where the Kalman gain K_k is given by

$$K_k = \Lambda_{xy\,k} \left(\Lambda_{yy\,k}\right)^{-1} \quad \text{(11-by-3).} \qquad \text{(A.14)}$$

Again, all summing and differencing operations in the state and measurement spaces must use Equations (A.5) and (A.6) . Operations are non-commutative and must be handled with care.

A.4.4 Initialization

With the filter built, the only remaining step is the specification of the measurement and process noise covariances, and the initialization of the state estimate and covariance. The nonlinear measurement noise is characterized by a 3-by-3 covariance matrix describing rotational measurement error.

$$\Lambda_{ww} \equiv E\left[\left(\vec{q}_m \Theta \vec{q}_g\right)\left(\vec{q}_m \Theta \vec{q}_g\right)^T\right] = E\left[\vec{w} \cdot \vec{w}^T\right] = \begin{bmatrix} \sigma_w^2 & & \\ & \sigma_w^2 & \\ & & \sigma_w^2 \end{bmatrix}$$

where σ_w^2 is the angular variance of the surrogate measurement in each rotational degree of freedom. If the pose estimator performs properly (see Section 3.2), then σ_w will be small, on the order of degrees.

Likewise, process noise covariance is defined as

$$\Lambda_{vv} \equiv E\left[\left(\vec{x}_k \Theta f\left(\vec{x}_{k-1}\right)\right)\left(\vec{x}_k \Theta f\left(\vec{x}_{k-1}\right)\right)^T\right] = E\left[\vec{v} \cdot \vec{v}^T\right].$$

The process noise covariance matrix should be chosen carefully to encapsulate the uncertainty in the dynamic model. This is somewhat implementation dependent since it describes real-world, unmodeled phenomena such as orbital mechanics, gravity gradient effects, fuel sloshing, lack of target rigidity, etc. Ultimately this choice is based on engineering judgement of the real-world environment.

In the absence of additional information, the initial a posteriori state estimate should be provided (in the nonsingular parameterization) as

$$\hat{x}_0 = \begin{Bmatrix} \hat{\omega} \\ \hat{q}_g \\ \hat{q}_d \\ \hat{q}_I \end{Bmatrix}_0 = \left\{ \begin{array}{c} \{\vec{0}\} \\ \{\vec{q}_m(t=0)\} \\ \left\{ \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array} \right\} \\ \left\{ \begin{array}{c} \cos\left(\frac{1}{2}\cos^{-1}\left(\frac{1}{\sqrt{3}}\right)\right) \\ \frac{-1}{\sqrt{2}}\sin\left(\frac{1}{2}\cos^{-1}\left(\frac{1}{\sqrt{3}}\right)\right) \\ \frac{1}{\sqrt{2}}\sin\left(\frac{1}{2}\cos^{-1}\left(\frac{1}{\sqrt{3}}\right)\right) \\ 0 \end{array} \right\} \end{array} \right\}.$$

This choice represents an unbiased first estimate. The velocity estimate is zero and the pose estimate is the first surrogate measurement. The offset quaternion is initialized to the identity (no offset). This choice of inertia quaternion maps the principal inertia estimates to equal values $\left(\frac{1}{\sqrt{3}}\right)$.

The initial covariance estimate is a diagonal matrix that quantifies the expected errors in the initial state estimate. This is a minimum-order 11-by-11 matrix:

$$\Lambda_{xx0} \equiv E\left[(\vec{x}_0 \Theta \hat{x}_0)(\vec{x}_0 \Theta \hat{x}_0)^T\right]$$

$$= \begin{bmatrix} E\left[(\vec{\omega}_0 - \hat{\omega}_0)(\vec{\omega}_0 - \hat{\omega}_0)^T\right] & & & \\ & E\left[(\vec{q}_{g0} \Theta \hat{q}_{g0})(\vec{q}_{g0} \Theta \hat{q}_{g0})^T\right] & & \\ & & E\left[(\vec{q}_d \Theta \hat{q}_{d0})(\vec{q}_d \Theta \hat{q}_{d0})^T\right] & \\ & & & \begin{bmatrix} E\left[p_{I1}^2\right] & E\left[p_{I1}p_{I2}\right] \\ E\left[p_{I2}p_{I1}\right] & E\left[p_{I2}^2\right] \end{bmatrix} \end{bmatrix}.$$

APPENDIX

B

CREATING SYNTHETIC RANGE IMAGES USING OPENGL

```
//This function synthesizes a range image of the objects that are drawn in the function
//(*ObjectsToScan)(void).  It returns the number of data points in the range image.

#include <gl/glut.h>

int Scan(
        void (*ObjectsToScan)(void),  //function that renders the scanned objects
        double scanData[][3],         //range image storage - make sure it's big enough
        )
{
    //local variables
    //****************************** be sure to initialize these ***********************
    double  invPositionMatrix[16],    //inverse of the sensor position matrix
            projectionMatrix[16],     //sensor projection matrix, describes focal props
            scanWidth, scanHeight;    //sensor's scan width and height, in pixels
    //**********************************************************************************
    double  identityMatrix[] = {1.,0.,0.,0.,  0.,1.,0.,0.,  0.,0.,1.,0.,  0.,0.,0.,1.};
    int     viewport[] = {0, 0, scanWidth, scanHeight},  //sensor viewport
            numberDataPoints = 0;                        //number of range image points
    GLfloat *depthBuffer;                                //depth buffer contents

    //allocate memory
    depthBuffer = new GLfloat[scanWidth*scanHeight];

    //it may be desirable to render to a pixel buffer rather than the main OpenGL window
    //if so, switch to the pixel buffer here
    pBuffer->Push();

    //remember the viewport variables
    glPushAttrib(GL_VIEWPORT_BIT);

        //set up a new viewport that is the size of the scan width and height
        glViewport(0, 0, (GLsizei) scanWidth, (GLsizei) scanHeight);
        glMatrixMode(GL_PROJECTION);
        glPushMatrix();

            //set up scanner focal properties
            glLoadMatrixd(projectionMatrix);
            glMatrixMode(GL_MODELVIEW);
            glPushMatrix();

                //set up scanner viewpoint to desired position
                glLoadMatrixd(invPositionMatrix);

                //draw everything the scanner should see
                glClear(GL_DEPTH_BUFFER_BIT);
                glColorMask(GL_FALSE,GL_FALSE,GL_FALSE,GL_FALSE);
                (*ObjectsToScan)();
                glColorMask(GL_TRUE,GL_TRUE,GL_TRUE,GL_TRUE);
```

```
            //grab contents of the OpenGL depth buffer
            glReadPixels(0,0,scanWidth,scanHeight,
                         GL_DEPTH_COMPONENT,GL_FLOAT,depthBuffer);

            //get x,y,z location of all scanned pixels (not including background)
            //(these are in the scanner's reference frame)
            numberDataPoints=0;
            for (int i=0;i<scanWidth*scanHeight;i++)
            {
                if (depthBuffer[i]!=1.)
                {
                    //a rendered object was found at this pixel location
                    //(i.e. pixel data is not at the far clipping plane)
                    gluUnProject((GLdouble) (i%scanWidth)+.5,
                                 (GLdouble) (i/scanWidth)+.5,
                                 (GLdouble) depthBuffer[i],
                                 identityMatrix, projectionMatrix, viewport,
                                 &scanData[numberDataPoints][0],
                                 &scanData[numberDataPoints][1],
                                 &scanData[numberDataPoints][2]);
                    numberDataPoints++;
                }
            }

            //add sensor noise here if desired
            CorruptData();

        //return to original projection and modelview matrices
        glPopMatrix();
        glMatrixMode(GL_PROJECTION);
    glPopMatrix();
    glMatrixMode(GL_MODELVIEW);

//return to original viewport
glPopAttrib();

//exit the pixel buffer, if one was used
pBuffer->Pop();

//free memory
delete [] depthBuffer;

return(numberDataPoints);
}
```