RELATIVE POSITION ESTIMATION FOR INTERVENTION-CAPABLE AUVS BY FUSING VISION AND INERTIAL MEASUREMENTS

Andreas Huster and Stephen M. Rock
Aerospace Robotics Lab, Stanford University
{huster,rock}@arl.stanford.edu

Abstract

This paper proposes a position estimator that fuses monocular vision with accelerometer and gyro measurements to generate a direct, relative, 6DOF position estimate between an Autonomous Underwater Vehicle (AUV) and a stationary object of interest. This type of estimator is useful for intervention-capable AUVs, which need to control their position relative to objects in their environment in order to accomplish tasks like station-keeping, vision-based object modeling/reconstruction, and manipulation. Various vision-only systems have been used to estimate relative position, but these are often difficult to implement in real underwater environments. The system we propose relies on vision to generate relative position information, but also fuses inertial rate sensors to reduce the amount of information that needs to be extracted from the vision system. The result is a system that is simpler and more robust than a vision-only solution. However, the use of inertial rate sensors introduces several issues. The rate measurements are subject to biases, which need to be estimated to prevent the accumulation of unbounded drift as the measurements are integrated. The observability of state estimates requires sufficient camera motion. Finally, the estimation problem contains significant non-linearities. The paper compares two non-linear estimation techniques, the EKF and the two-step estimator, in the context of this estimation problem and presents a simulation of a simplified problem to highlight the advantages of the two-step estimator.

1 Introduction

This paper describes a position sensing system for intervention-capable AUVs. This is a new class of AUV, with unique sensing requirements, that will be able perform various autonomous manipulation tasks, such as placing sensors and retrieving samples. These are complex tasks composed of modeling, planning and execution phases during which the AUV requires accurate control (or at least knowledge) of its position and orientation relative to the object of interest. If the AUV cannot lock its position (e.g., by thrusting into the sea floor) during an autonomous manipulation task, it requires a real-time estimate of the relative 6DOF (degrees of freedom) position between the AUV and the object of interest. This paper proposes a sensing strategy that enables AUV control relative to objects in its environment.

Some AUVs already perform relatively simple intervention tasks, like sampling of volcanic glass with the Autonomous Benthic Explorer (ABE) [11] and docking with the Odyssey AUV [3, 9]. More complex intervention tasks, such as grasping of unknown and unprepared objects, are much more difficult to implement.

Several position sensing systems have been used to control AUVs. Sonar beacons deployed and surveyed prior to the AUV mission, as well as Inertial Navigation Systems (INS) coupled with periodic position fixes to correct for accumulated drift (e.g., from GPS), have been used for AUV navigation. *Simultaneous Localization and Mapping* (SLAM) is useful when no external position measurements are available [2, 10]. Several vision-based positioning systems have been demonstrated for control of AUVs relative to planar surfaces [6, 7, 8].

However, autonomous manipulation tasks require a direct 6DOF position measurement of the object of interest. Vision-based techniques (e.g., binocular stereo or structure from motion) provide relative position information and are often used for manipulation tasks. But most vision-only 6DOF positioning algorithms are difficult to implement in real underwater environments because they require a large number of trackable image features or rely on feature correspondences in multiple cameras. Underwater manipulation tasks may occur in environments with only a small number of good visual features, given the constraints on lighting and visibility imposed by real underwater environments. Robust tracking of features and reliable correspondences are difficult to achieve.

The system we propose relies on vision to generate relative position information, but also fuses inertial rate sensors to reduce the amount of information that needs to be extracted from the vision system. The result is a

system that is simpler and more robust than a vision-only solution. Our system requires only a bearing measurement to a fixed point (usually associated with the object to be manipulated) which can be obtained by tracking a single visual feature. Compared to the vision-only algorithms, tracking only the best feature in a monocular vision system can be much faster and more reliable.

The measurements from monocular vision and inertial rate sensors complement each other well. The motion of the camera between successive images generates a baseline for range computations by triangulation. Inertial rate sensors, whose acceleration and angular rate measurements can be integrated to obtain vehicle velocity, position and orientation, can account for the 6DOF motion of the AUV along this baseline. When these measurements are fused, the relative position between the AUV and the object can be computed. A key benefit of this system is that, after initialization, the inertial rate sensors continue to maintain a useful estimate of relative position during vision drop-outs (e.g., occlusions, lack of correspondence). Furthermore, both inertial rate sensors (for navigation) and monocular vision systems (for science purposes) are already common AUV sensors.

However, the use of inertial rate sensors introduces a major issue into the design of the system. Like other dead-reckoning sensors, inertial rate sensors suffer from bias and random noise errors, which lead to unbounded drift in the integrated quantities. While more expensive sensors are associated with less drift, we envision the use of low-cost inertial sensors, which are subject to significant drift errors. Therefore, an estimator to resolve the relative AUV position as well as the sensor biases and drift errors is required.

Observability of the states to be estimated is a critical issue because the problem is non-linear and observability depends on the motion of the camera. During camera translation directly towards or away from the feature, the estimator has no new information with which to improve its range estimate. Only camera motions transverse to the feature direction provide new information for the range estimate. As a result, the estimator requires *sufficient* transverse camera motion in order to produce useful position estimates. In some cases, extra camera maneuvers are required to improve observability. This presents an interesting conflict between trajectories which are designed to complete the manipulation task and special camera maneuvers required to ensure estimator observability.

The estimation problem contains two significant non-linearities. The first is related to the rotational degrees of freedom of the AUV and the second is caused by the camera's projection of the three-dimensional world onto the 2D image plane. As a result, the dynamics and measurement equations are non-linear and depend on the actual

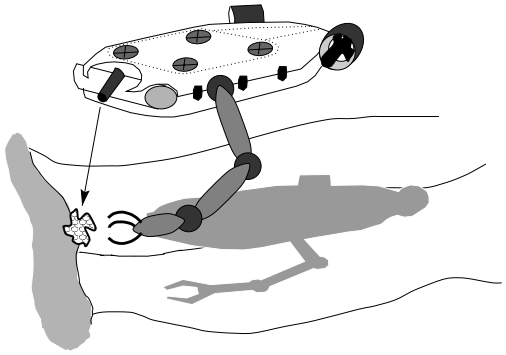

Figure 1: Relative Position Estimator Scenario

state of the system. In fact, the estimator exploits the non-linearity of the problem to observe the range to the feature. As motion of the AUV modifies the system state, the measurement equations change, and new measurements (i.e., bearings to the object from new viewpoints) make the range to the object observable.

Various methods exist to handle non-linear estimation problems. In this paper, we consider estimators based on extensions to the popular Kalman Filter. The most widely used is the Extended Kalman Filter (EKF), which linearizes the dynamics and measurement equations of non-linear systems in order to take advantage of the Kalman Filter equations. Although the EKF works very well for a wide range of applications, it comes with no guarantees and can lead to very poor performance. An alternative to the EKF is the two-step estimator [5], which reformulates the problem so that all the measurement equations become linear and all of the non-linearities appear in the dynamics. At the expense of added complexity, the two-step estimator can produce much better estimates.

This paper compares the ability of the EKF and the two-step estimator to fuse monocular vision and inertial rate sensors and to estimate relative position. We will derive an EKF and two-step estimator for a simplified (2D) problem and present a simulation that examines the accuracy of the state and covariance estimates and the tendency of the estimators to diverge. The simulation indicates that in the context of this problem, the two-step estimator performs much better.

The rest of the paper is organized as follows. Section 2 defines the estimation problem in terms of the system states, dynamics, measurements, and noise models. Section 3 shows how to construct an EKF and a two-step estimator to solve the 2D version of this problem. Section 4 presents and compares the results from a simulation of both estimators. Section 5 states our conclusions and outlines future work.

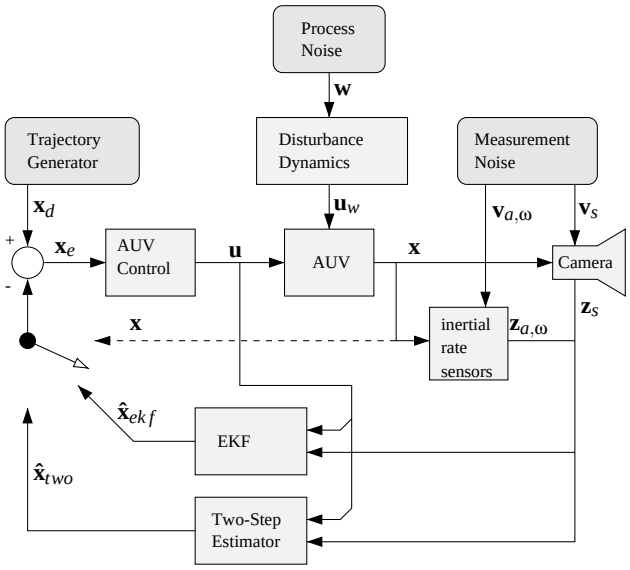

Figure 2: Block Diagram with AUV Controller, Sensors, and Estimators

2 Problem Statement

This section defines the geometry of the estimation problem, the sensor measurements used to update the state estimate, the coordinates that represent the state of the system, and the dynamics associated with these states. Figure 1 shows an intervention-capable AUV which has extended its manipulator arm to grasp a stationary object in the environment. For the grasping task to succeed, the AUV requires a relative position measurement of the object in AUV coordinates. A camera on the front of the AUV is tracking the object and inertial rate sensors are reporting the AUV's acceleration and angular velocity.

The block diagram in Figure 2 shows how the AUV, its controller, sensors, estimators and the environment interact. The camera measurement $\mathbf{z}_s$ and inertial rate measurements $\mathbf{z}_a$ and $\mathbf{z}_\omega$, along with the control commands $\mathbf{u}$ are passed to an estimator, which computes a state estimate $\hat{\mathbf{x}}$. The difference $\mathbf{x}_e$ between the desired trajectory $\mathbf{x}_d$ and the state estimate is used by the controller to generate the next $\mathbf{u}$. The control command and any disturbances $\mathbf{u}_w$ drive the AUV dynamics to generate a new state $\mathbf{x}$. In general, $\mathbf{x}$ is not available for AUV control. The diagram shows a dashed line to indicate that this truth measurement can only be used for simulations.

2.1 Geometry

The essential geometry of the estimation problem is represented in Figure 3. Frame **O** indicates the inertial reference frame and $\mathbf{p}$ indicates the position of the stationary feature. Let $\mathbf{q}$ be the position of the camera, which

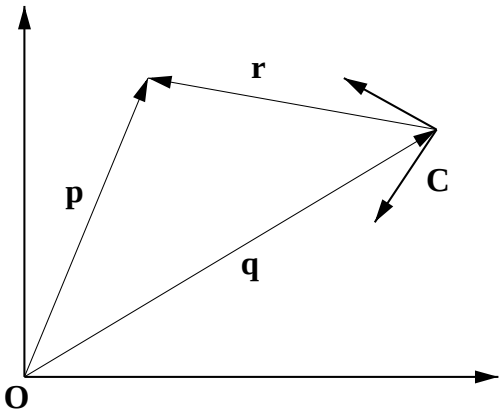

Figure 3: Geometry of the Estimation Problem

is attached to the body frame **C**. In principle, the camera and the inertial sensors can be mounted anywhere on the AUV as long as their relative position and orientation are known. To simplify the discussion, we assume that all the sensors are colocated at $\mathbf{q}$. R_{oc} is the rotation matrix from the camera to the inertial frame and ${}^C\boldsymbol{\omega}$ is the associated rotational velocity expressed in the camera frame. We use a preceding superscript C to indicate that a vector is resolved in Frame **C** instead of Frame **O**.

The position of the feature as seen by the camera is $\mathbf{r} = \mathbf{p} - \mathbf{q}$. We assume that the feature is stationary in the inertial frame, so $\dot{\mathbf{p}} = \ddot{\mathbf{p}} = 0$. Therefore, $\dot{\mathbf{r}} = -\dot{\mathbf{q}}$ and $\ddot{\mathbf{r}} = -\ddot{\mathbf{q}}$. Because of this assumption, a measurement of AUV acceleration $\ddot{\mathbf{q}}$ in inertial space is useful for estimating the relative feature position $\mathbf{r}$.

2.2 Measurements

The vision measurement $\mathbf{z}_s$ provides the projection of ${}^C\mathbf{r}$ onto the image plane, and is modeled as follows:

$$\mathbf{z}_s = \begin{bmatrix} s_x \\ s_y \end{bmatrix} = \begin{bmatrix} {}^Cr_x/{}^Cr_z \\ {}^Cr_y/{}^Cr_z \end{bmatrix} + \mathbf{v}_s$$

where $\mathbf{v}_s$ is zero-mean Gaussian noise $N(0, R_s)$. For simplicity, we assume that the camera measurement is scaled so the effective focal length is 1. With a slight abuse of terminology, we will refer to ${}^Cr_z$ as the range to the feature and to s_x and s_y as the bearing. The optical axis of the camera is aligned with the z-axis of Frame **C**.

While vision processing for underwater environments remains a challenging problem, our current research does not expand on it. Instead, we assume that a robust point-feature can be tracked, and focus instead on integrating this type of measurement into a position estimator.

The accelerometer measurement $\mathbf{z}_a$ includes the acceleration ${}^C\ddot{\mathbf{q}}$ of the AUV, the acceleration due to gravity,

a sensor bias $\mathbf{b}_a$, and sensor noise.

$$\mathbf{z}_a =^C \ddot{\mathbf{q}} + R_{co}\mathbf{g} + \mathbf{b}_a + \mathbf{v}_a$$

We assume that $\mathbf{v}_a$ is $N\ (0, R_a)$ noise. $R_{co} = R_{oc}^T$ is the rotation matrix from inertial to camera coordinates and $\mathbf{g} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$ is the acceleration due to gravity in inertial coordinates.

The gyro measurement includes the rotational velocity of the AUV, sensor bias $\mathbf{b}_\omega$, and sensor noise $\mathbf{v}_\omega$ modeled by $N\ (0, R_\omega)$.

$$\mathbf{z}_\omega =^C \omega + \mathbf{b}_\omega + \mathbf{v}_\omega$$

2.3 States

The state of the system is composed of the coordinates that are used to describe AUV position and to express the AUV dynamics and sensor measurements. These include the position $\mathbf{q}$, velocity $\dot{\mathbf{q}}$, and acceleration $\ddot{\mathbf{q}}$, acceleration disturbance $\mathbf{a}_w$, a quaternion λ to represent orientation, angular velocity ω, as well as the inertial rate sensor biases $\mathbf{b}_a$ and $\mathbf{b}_\omega$.

2.4 Dynamics

We assume simple $1/s^2$ dynamics for vehicle position and orientation. In inertial coordinates, these are:

$$\begin{aligned}
\frac{d}{dt}\mathbf{q} &= \dot{\mathbf{q}} \\
\frac{d}{dt}\dot{\mathbf{q}} &= \ddot{\mathbf{q}} = \mathbf{a}_w + R_{oc}(\lambda)\mathbf{u}_1 \\
\frac{d}{dt}\mathbf{a}_w &= \mathbf{w}_1 \\
\frac{d}{dt}\lambda &= E(\lambda)\,\omega \\
\frac{d}{dt}\omega &= R_{oc}(\lambda)\mathbf{u}_2 + \mathbf{w}_2 \\
\frac{d}{dt}\mathbf{b}_a &= \mathbf{w}_3 \\
\frac{d}{dt}\mathbf{b}_\omega &= \mathbf{w}_4 \\
\mathbf{u} &= \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} \\
\mathbf{w} &= \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ \mathbf{w}_4 \end{bmatrix}
\end{aligned}$$

The deterministic control input $\mathbf{u}$ is composed of the translational component $\mathbf{u}_1$ and the rotational component $\mathbf{u}_2$, both of which are expressed in AUV coordinates. The disturbances (e.g., ocean currents) and sensor biases (e.g., temperature dependence) are driven by the process noise $\mathbf{w}$, which is zero-mean Gaussian noise $N\ (0, Q)$.

We have chosen arbitrarily simple dynamics to model the disturbances, sensor biases, and sensor noise. For best performance, these models should be matched to the actual AUV, the environment in which it operates, and the specifications of the sensors that are used. However, engineering trade-offs are routinely applied to simplify the modeling task and the estimator design. For example, the Kalman Filter equations, which will be introduced in the following section, require all of the noise sources to be modeled as white Gaussian noise processes (which we have done). Our objective in this paper is to highlight significant differences between two estimation techniques and while the choice of these dynamics will affect the results, these differences are not central to our objective.

3 Estimator Design

The Kalman Filter is a widely used optimal estimator for linear systems with white Gaussian noise models. It updates the state and covariance estimate recursively in real time as new measurements are generated.

The Extended Kalman Filter (EKF), which is commonly applied to problems in which some of the dynamics or measurement equations are non-linear, provides a solution which is easy to implement, but fails to generate good results in the context of this problem. The EKF linearizes dynamics and measurements around the current state estimate to take advantage of the Kalman Filter equations, which accept only linear equations. However, linearization can introduce estimation errors. These are especially problematic in this case because estimation requires motion, which itself has to be estimated. Any biases that the estimates accumulate can continue to propagate to future estimates. As a result, the EKF often diverges, generates a poor covariance estimate, and results in large estimation errors.

Enforcing linear measurement equations by choosing appropriate coordinates to represent the system state provides an interesting solution to this problem. By using linear measurements, the Kalman measurement update equations require no approximations and introduce no bias errors due to linearization. This is the defining characteristic of the two-step estimator, which is described in detail in [5]. In the Bearings-Only Tracking community, a similar concept has been used to provide improved tracking estimates of targets observed with a bearing-only sensor [1].

The cost of this method appears in the system dynamics, which are now non-linear and more complex. Our

implementation uses a Runge-Kutta integration to propagate the modified state estimate and covariance forward in time. The burden of the non-linearity has shifted from the measurement update, where is causes biases, to the time update, where it can be handled more accurately with computational methods.

A consequence of this approach is that the system is automatically partitioned into instantaneously unobservable (e.g., range) and observable (e.g., bearing) states, which helps to prevent ill-conditioned covariance matrices. Of course, the purpose of the estimator is to make all states observable by incorporating camera motion.

In this section, we derive the EKF and two-step estimator equations for a 2D version of the problem. In two dimensions, the representation of rotation is much simpler (only one angle required and no singularities), the overall number of states is reduced, but most of the important aspects of the 3D problem are retained.

The state vector $\mathbf{x}$ for the 2D problem has eleven entries as follows:

$$\mathbf{x} = \begin{bmatrix} q_x \\ \dot{q}_x \\ a_{w,x} \\ b_{a,x} \\ q_z \\ \dot{q}_z \\ a_{w,z} \\ b_{a,z} \\ \lambda_{in} \\ \omega \\ b_\omega \end{bmatrix}$$

We use $\mathbf{q}$ instead of $\mathbf{r}$ to simplify the presentation. The estimate is still a relative measurement because we can choose $\mathbf{p}$ in the algorithm. If we choose $\mathbf{p} = 0$, then $\mathbf{r} = -\mathbf{q}$.

We represent rotation with

$$\begin{aligned} \lambda &= \begin{bmatrix} \lambda_c \\ \lambda_s \end{bmatrix} = \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix} \\ R_{oc}(\lambda) &= \begin{bmatrix} \lambda_c & -\lambda_s \\ \lambda_s & \lambda_c \end{bmatrix} \\ \frac{d}{dt}\lambda &= E(\lambda)\omega = \begin{bmatrix} -\lambda_s \\ \lambda_c \end{bmatrix} \omega \end{aligned}$$

At any time, only one of the λ-coordinates is included in the state vector. The other one can be computed from the constraint $1 = \lambda_c^2 + \lambda_s^2$ and knowledge of its sign. The sign of the coordinate with the larger magnitude does not change rapidly because its magnitude is greater than or equal to $\sqrt{0.5}$. This should be the excluded coordinate. If the absolute value of the excluded coordinate becomes less than 0.5, the two coordinates need to be exchanged to ensure that the sign of the excluded component is well defined. To handle this bookkeeping complexity, we introduce a convention that allows us to write the estimator equations without dealing with separate cases. If $|\lambda_c| < |\lambda_s|$, then

$$\begin{aligned} \lambda_{in} &= \lambda_c \\ \lambda_{out} &= \lambda_s \\ \lambda_1 &= -\lambda_s \end{aligned}$$

If $|\lambda_s| < |\lambda_c|$, then

$$\begin{aligned} \lambda_{in} &= \lambda_s \\ \lambda_{out} &= \lambda_c \\ \lambda_1 &= \lambda_c \end{aligned}$$

The dynamics of the 2D system can now be expressed in matrix form:

$$\dot{\mathbf{x}} = A(\lambda)\mathbf{x} + B_u(\lambda)\mathbf{u} + B_w\mathbf{w}$$

$$A(\lambda) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_u(\lambda) = \begin{bmatrix} 0 & 0 & 0 \\ \lambda_c & -\lambda_s & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \lambda_s & \lambda_c & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_w = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

3.1 EKF Equations

The Kalman Filter [4] consists of a time-update, which advances the estimates forward in time ($\hat{\mathbf{x}}_{i-1}$ to $\bar{\mathbf{x}}_i$), and a measurement update which incorporates the current measurements into the estimate ($\bar{\mathbf{x}}_i$ to $\hat{\mathbf{x}}_i$). Associated with these estimates are covariance matrices:

$$\begin{aligned} M_{x,i} &= E\bar{\mathbf{x}}_i\bar{\mathbf{x}}_i^T \\ P_{x,i} &= E\hat{\mathbf{x}}_i\hat{\mathbf{x}}_i^T \end{aligned}$$

3.1.1 Time Update

The time update for the EKF is computed using a discrete-time formulation, where T is the sample interval.

$$\begin{aligned} \bar{\mathbf{x}}_i &= \Phi_i\hat{\mathbf{x}}_{i-1} + \Gamma_{u,i}\mathbf{u}_i + \Gamma_{w,i}\mathbf{w}_i \\ M_{x,i} &= \Phi_i P_{x,i-1}\Phi_i^T + \Gamma_{w,i}\frac{Q}{T}\Gamma_{w,i}^T \end{aligned}$$

The matrices Φ_i, $\Gamma_{u,i}$, and $\Gamma_{w,i}$ can be computed from $A(\hat{\lambda}_{i-1})$, $B_u(\hat{\lambda}_{i-1})$, B_w and T using a zero-order hold technique.

3.1.2 Measurement Update

The measurement equations in Section 2.2 can be expressed as

$$\begin{aligned} \mathbf{z}_i &= \begin{bmatrix} z_{s,i} \\ \mathbf{z}_{a,i} \\ z_{\omega,i} \end{bmatrix} + \mathbf{u}_z + \mathbf{v}_i = f(\mathbf{x}_i) + \mathbf{u}_z + \mathbf{v}_i \\ \mathbf{u}_z &= \begin{bmatrix} 0 & \mathbf{u}_1^T & 0 \end{bmatrix}^T \end{aligned}$$

where $\mathbf{v}_i$ is $N(0,R)$ white Gaussian noise and

$$\begin{aligned} f(\mathbf{x}) &= \begin{bmatrix} {}^C r_x/{}^C r_z \\ R_{co}(\mathbf{a}_w + \mathbf{g}) + \mathbf{b}_a \\ \omega + \mathbf{b}_\omega \end{bmatrix} \\ \begin{bmatrix} {}^C r_x \\ {}^C r_z \end{bmatrix} &= R_{co}(\mathbf{p} - \mathbf{q}) \\ R &= \begin{bmatrix} R_s & 0 & 0 \\ 0 & R_a & 0 \\ 0 & 0 & R_\omega \end{bmatrix}. \end{aligned}$$

For the EKF, we need to linearize $f(\mathbf{x})$ using a first-order Taylor expansion about $\bar{\mathbf{x}}$. This is based on an assumption that higher-order terms are small.

$$\begin{aligned} f(\mathbf{x}) &\cong f(\bar{\mathbf{x}}) + C(\bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}}) \\ C(\mathbf{x}) &= \frac{\partial f}{\partial \mathbf{x}} \end{aligned}$$

$$= \begin{bmatrix} -r_z/{}^C r_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \lambda_c & -\lambda_s & 0 \\ 0 & 1 & 0 & 0 \\ r_x/{}^C r_z^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \lambda_s & \lambda_z & 0 \\ 0 & 0 & 1 & 0 \\ a_z/\lambda_1 & -a_x/\lambda_1 & \frac{{}^C r_x^2 + {}^C r_z^2}{{}^C r_z^2 \lambda_1} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T$$

where

$$\begin{bmatrix} a_x \\ a_z \end{bmatrix} = R_{co}(\lambda)\left(\begin{bmatrix} a_{w,x} \\ a_{w,z} \end{bmatrix} + g\right)$$

Finally, we can write the EKF measurement update equations:

$$\begin{aligned} L_i &= M_{x,i}C(\bar{\mathbf{x}}_\mathbf{i})^T\left(R + C(\bar{\mathbf{x}}_\mathbf{i})M_{x,i}C(\bar{\mathbf{x}}_\mathbf{i})^T\right)^{-1} \\ \hat{\mathbf{x}}_i &= \bar{\mathbf{x}}_i + L_i(\mathbf{z}_i - f(\bar{\mathbf{x}}_i) - \mathbf{u}_z) \\ P_{x,i} &= (I - L_i C(\bar{\mathbf{x}}_\mathbf{i}))M_{x,i} \end{aligned}$$

3.2 Two-Step Estimator Equations

Before deriving the two-step estimator equations, we need to define a modified state vector $\mathbf{y}$ that leads to linear measurement equations.

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} s_x \\ v_x \\ a_x \\ b_{a,x} \\ {}^C r_z \\ v_z \\ a_z \\ b_{a,z} \\ \lambda_{in} \\ \omega \\ b_\omega \end{bmatrix} \\ v_x &= {}^C\dot{q}_x \\ v_z &= {}^C\dot{q}_z \\ \begin{bmatrix} a_x \\ a_z \end{bmatrix} &= R_{co}(\lambda)\left(\begin{bmatrix} a_{w,x} \\ a_{w,z} \end{bmatrix} + g\right) \end{aligned}$$

We can construct an invertible function F to relate the $\mathbf{x}$ and $\mathbf{y}$ states:

$$\begin{aligned} \mathbf{y} &= F(\mathbf{x}) \\ \mathbf{x} &= F^{-1}(\mathbf{y}) \end{aligned}$$

The two steps of this estimator refer to computing $\mathbf{y}$ using the Kalman Filter measurement update equations in a first step and then using F^{-1} to compute $\mathbf{x}$ in a second step.

The measurement equations from Section 2.2 can be expressed in terms of $\mathbf{y}$:

$$\begin{aligned} z_s &= s_x + v_s \\ \mathbf{z}_a &= \begin{bmatrix} a_x + b_{a,x} \\ a_z + b_{a,z} \end{bmatrix} + \mathbf{u}_1 + \mathbf{v}_a \\ z_\omega &= \omega + b_\omega + v_\omega \end{aligned}$$

3.2.1 Time Update

The dynamics for $\mathbf{y}$, derived according to the rules in [5], are as follows:

$$\begin{aligned} \dot{\mathbf{y}} &= \tilde{A}(\mathbf{y}, \mathbf{u}) \\ \dot{P}_y &= \left.\frac{\partial \tilde{A}}{\partial \mathbf{y}}\right|_{\mathbf{y}} P_\mathbf{y} + P_\mathbf{y} \left.\frac{\partial \tilde{A}}{\partial \mathbf{y}}\right|_{\mathbf{y}}^T + \tilde{B} Q \tilde{B}^T \end{aligned}$$

$$\tilde{A}(\mathbf{y}, \mathbf{u}) = \begin{bmatrix} \frac{(s_x v_z - v_x)}{C r_z} + \left(s_x^2 + 1\right)\omega \\ a_x + \begin{bmatrix} -\lambda_c & -\lambda_s \end{bmatrix} \mathbf{g} + v_z \omega + u_{1,x} \\ a_z \omega \\ 0 \\ -v_z - s_x {}^C r_z \omega \\ a_z + \begin{bmatrix} \lambda_s - \lambda_c \end{bmatrix} \mathbf{g} - v_x \omega + u_{1,z} \\ -a_x \omega \\ 0 \\ \lambda_1 \omega \\ u_2 \\ 0 \end{bmatrix}$$

$$\tilde{B}(\lambda) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \lambda_c & 0 & \lambda_s & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -\lambda_s & 0 & \lambda_c & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For the two-step estimator, we perform the time update ($\hat{\mathbf{y}}_{i-1}$ to $\bar{\mathbf{y}}_i$ and $P_{y,i-1}$ to $M_{y,i}$) with a Runge-Kutta numerical integration technique.

3.2.2 Measurement Update

Because the coordinates of $\mathbf{y}$ are chosen so that the measurement equations are linear, the measurement update ($\bar{\mathbf{y}}_i$ to $\hat{\mathbf{y}}_i$ and $M_{y,i}$ to $P_{y,i}$) is identical to the Kalman Filter.

$$\begin{aligned} C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T \\ L_i &= M_{y,i} C^T \left(R + C M_{y,i} C^T\right)^{-1} \\ \hat{\mathbf{y}}_i &= \bar{\mathbf{y}}_i + L_i \left(\mathbf{z}_i - C \bar{\mathbf{y}}_i - \mathbf{u}_z\right) \\ P_{y,i} &= (I - L_i C) M_{y,i} \end{aligned}$$

4 Simulation Results

We have used a numerical simulation to compare the 2D versions of the EKF and the two-step estimator. This section describes the simulation and presents the results.

The data are derived from a Monte Carlo simulation with 1000 runs. Each run is based on the same reference trajectory, but uses random initial conditions, disturbances, and measurement noise. The control commands $\mathbf{u}$ and sensor measurements $\mathbf{z}_{s,a,\omega}$ are provided as inputs for the estimators, which compute estimates of the true system state (see Figure 2). The sample interval was set to $T = 0.1\,s$ and the duration of each experiment is $20\,s$.

To simplify the comparison between the two estimators, we avoid closing the loop on an estimated state by using the true state $\mathbf{x}$ of the system to compute control commands. Using the output from one of the two estimators to control the AUV might add an advantage or handicap over the other estimator. Our simulation does not capture the effect of closing the loop around the estimators nor does it examine the effect of using different reference trajectories. These are topics we hope to address in future work.

The goal of the simulation is to highlight generic properties of the EKF and two-step estimators in the context of fusing vision and inertial measurements to estimate relative position. Therefore, we use simple noise models with arbitrary parameters. While the results will depend on these choices, our experiments show that some generic properties are present in all useful cases. For this simulation, we chose uncorrelated process noise for which each component has a variance $\sigma_w^2 = 0.0001$. The sensor noise is also uncorrelated with

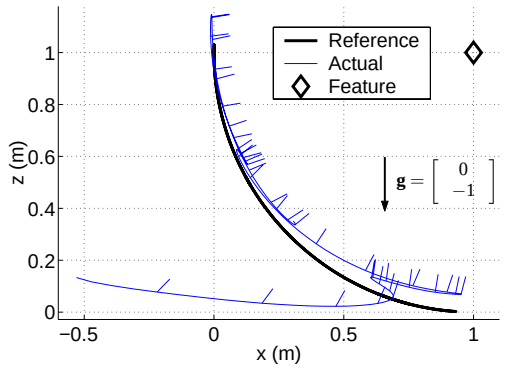

Figure 4: A 2D Simulation showing the Actual and Reference Trajectory and the Visual Feature

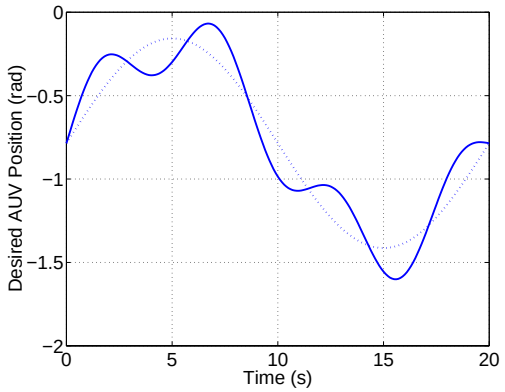

Figure 5: Desired AUV Position Expressed as an Angle along an Arc. The dotted line was considered, but the solid line was actually used.

variance $\sigma_v^2 = 0.0001$ for each component. Both R and Q are diagonal matrices with 0.0001 along their diagonal.

Figure 4 shows a 2D world with a visual feature, reference trajectory, and actual AUV trajectory for one run of the simulation. The initial conditions for the AUV are drawn from a zero-mean Gaussian distribution, except for the rotation, which is centered on $\theta = -\pi/4$ to point the camera towards the feature. (Note that the optical axis of the camera is aligned with the z-axis of Frame **C**.) The variance of all of the components in the initial condition is $\sigma_{x,t=0}^2 = 0.1$, except for the position components $\sigma_{q,t=0}^2 = 0.2$ and the rotation $\sigma_{\theta,t=0}^2 = 0.05$. The short lines on the actual trajectory indicate the optical axis (viewing direction) of the camera at 5 second intervals.

The design of an optimal desired trajectory for the AUV is a problem that we have not yet studied, so we chose an *ad hoc* trajectory for this simulation. The desired AUV position lies on an arc of radius 1 *m* around the feature. It rotates the AUV through a quarter turn while

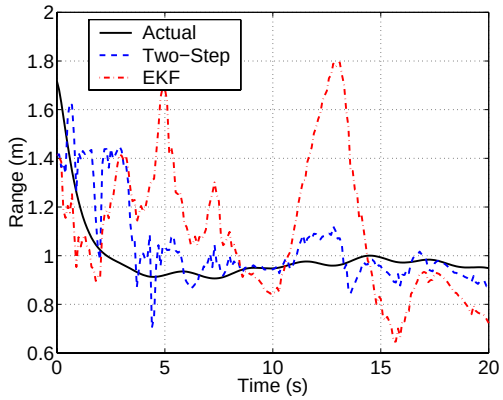

Figure 6: Range Estimate

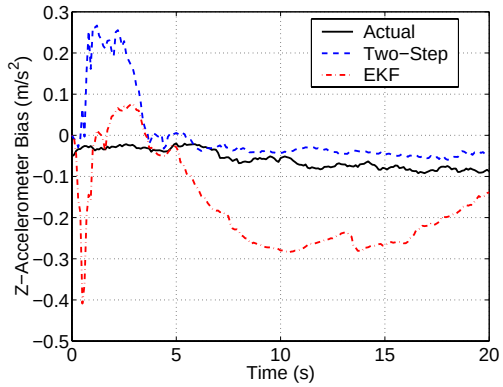

Figure 7: Z-Accelerometer Bias Estimate

observing the feature, which helps with the identification of both accelerometer biases. Figure 5 shows the desired AUV position as an angle along this arc. The dotted line would start the AUV at the midpoint of the arc, move it towards the lower right side, then to the upper left side, and finally back to the midpoint using a sine wave trajectory. However, we actually use the solid line, which generates better results because it has greater variation of AUV velocity and acceleration.

Figures 6 to 8 show the range, z-accelerometer bias, and theta estimates for both estimators along with the truth values for one run. These three coordinates are important to consider because they are not directly observable and among the most difficult to estimate. For example, the theta estimate depends primarily on the component of the accelerometer measurement due to the gravity vector, so if the estimated accelerometer bias is wrong, this error propagates to the theta estimate. The theta estimate is used in the implicit triangulation performed by the estimator, so good range estimates depend on the quality of the theta estimate.

We will compare the two estimators based on three criteria: (1) how often the algorithm failed, (2) the qual-

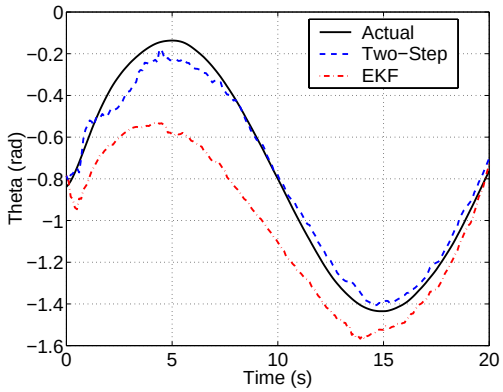

Figure 8: Theta Estimate

ity of the state estimate, and (3) their ability to predict estimation error with the covariance matrix.

During the simulation, both the EKF and the two-step estimator diverged on some of the runs. When the estimated range became too small (less than $0.1\,m$) or too large (greater than $5\,m$), the run was marked as a failure. Both algorithms encounter a singularity when the estimated range approaches zero. A range greater than $5\,m$ is unrealistic, if only because of underwater visibility constraints.

Table 1: Outcomes of the Monte Carlo Simulation

1000 Runs	EKF Success	EKF Failure
Two-Step Success	686	281
Two-Step Failure	4	29

Table 1 shows that in 686 of the 1000 runs, both algorithms succeeded in producing a useful estimate without diverging, and in 29 runs, both algorithms failed. A failure is not necessarily catastrophic, because both estimators can easily reset themselves when they have diverged (which they can detect using the rules above). However, an estimator which needs to reset less frequently is more desirable.

The table also shows that the EKF succeeded only in 4 cases when the two-step estimator failed. On the other hand, the two-step estimator succeeded 281 times when the EKF failed. These numbers indicate that the two-step estimator is much more robust than the EKF.

We used the 686 Monte Carlo runs for which both estimators succeeded for the remaining part of the analysis. Figures 9 to 11 show plots of the standard deviation of the estimation error for range, z-accelerometer bias, and theta. In all cases, the standard deviation is significantly reduced for the two-step estimator for most of the $20\,s$ experiment. At $t = 20\,s$, the standard deviation of

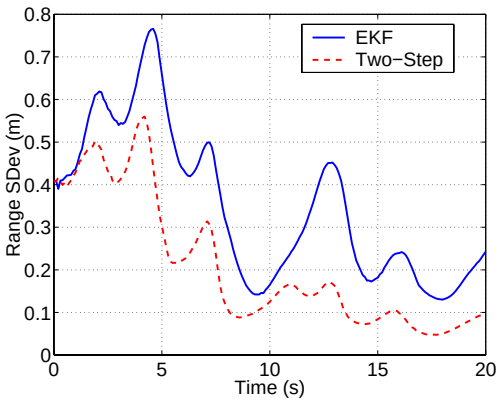

Figure 9: Monte Carlo Standard Deviation for Range

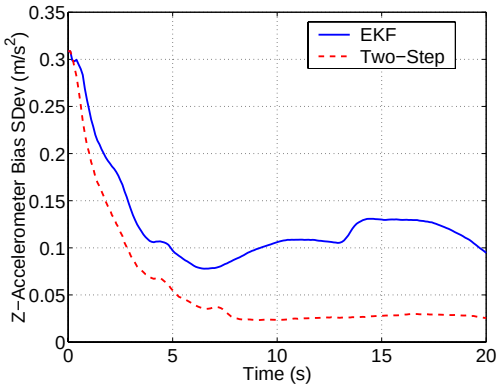

Figure 10: Monte Carlo Standard Deviation for Z-Accelerometer Bias

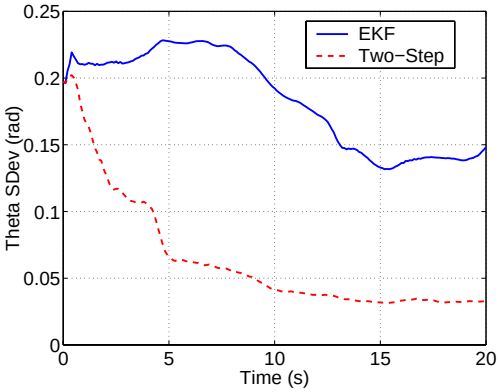

Figure 11: Monte Carlo Standard Deviation for Theta

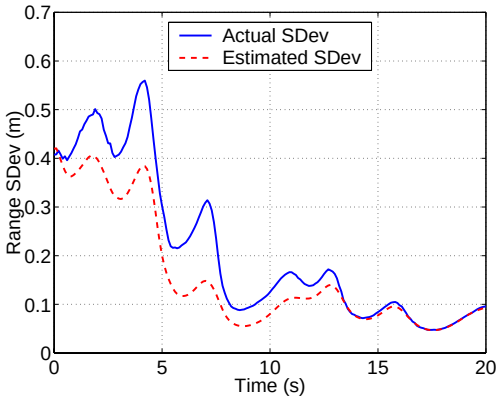

Figure 12: Comparison of Standard Deviation for the Range coordinate of the Two-Step Estimator

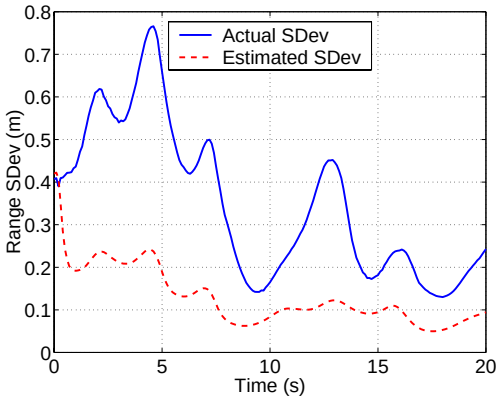

Figure 13: Comparison of Standard Deviation for the Range coordinate of the EKF

the EKF is at least twice as large for each of the three coordinates, which highlights a clear advantage of the two-step estimator.

The covariance estimate generated by each algorithm is also a useful quantity in the control of the AUV. For example, when the standard deviation of the range is large, the algorithm indicates that the range estimate is uncertain. In that case, the trajectory generator should back up the AUV to avoid a possible collision and should start a maneuver to improve the observability of the range estimate. In this last comparison, we examine the quality of the covariance estimate for each algorithm.

The actual standard deviation σ is computed from the estimation error ($\tilde{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}$), which can only be computed during a simulation when a truth measurement is available. It is an ensemble average over N runs.

$$\sigma(t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \tilde{\mathbf{x}}_\mathbf{i}^2(\mathbf{t})}$$

The estimated standard deviation $\hat{\sigma}$ is an ensemble average of the quantity that the algorithm reports in the covariance matrix.

$$\hat{\sigma}(t) = \frac{1}{N}\sum_{i=1}^{N} \sigma_i(t)$$

Figure 12 compares these two quantities for the range estimate generated by the two-step estimator and Figure 13 shows the equivalent comparison for the EKF. Whereas $\hat{\sigma}$ is a useful predictor for σ for the two-step estimator, this is not the case for the EKF, where $\hat{\sigma}$ is much smaller than σ. This failure of the EKF to generate an accurate covariance estimate is tightly related to the inflated estimation errors plotted in Figures 9 to 11.

The simulation results in this section have shown a clear advantage in performance of the two-step estimator over the EKF. This was demonstrated by the number of failures due to divergence, by the size of the estimation errors, and the accuracy of the covariance estimate.

5 Conclusions

We have proposed a relative position estimator for intervention-capable AUVs. These AUVs will be used to execute autonomous manipulation tasks during which the AUV's position relative to a target object needs to be controlled. While a variety of position sensing systems have been used for AUV control, this application poses a unique sensor requirement: a robust, 6DOF, direct estimate of the relative position between the AUV and a stationary visual feature.

Our goal is to demonstrate this position estimator on an intervention-capable AUV. All of the assumptions and constraints that have shaped the design of the estimators are derived from such an AUV mission. However, this work is still in progress, and we have not reached the AUV testing phase.

At this time, we have defined a sensor strategy that merges vision and inertial rate sensors, discussed its key advantages, and identified some of the challenges of implementing this technique. Our initial work has focused on developing an estimator to fuse measurements from these two sensors. We have compared two estimator designs, the popular EKF and the two-step estimator, in the context of a simplified 2D version of this problem, and have concluded that the two-step estimator provides critical advantages at a cost of extra design complexity and some computational expense.

Our future work will focus on four main topics. First, the algorithm needs to be expanded to the 3D scenario. Second, we have not discussed the impact of specific disturbance dynamics and sensor noise models, and have instead assumed arbitrary models. More realistic modeling of these processes will improve the performance

of an operational system. Third, we currently use *ad hoc* trajectories to generate sufficient sensor motion. The issue of blending optimal maneuvers to improve observability with the execution of useful tasks (e.g., manipulation) remains to be explored. Finally, we expect to demonstrate the estimator as part of an AUV manipulation task.

References

[1] Vincent J. Aidala and Sherry E. Hammel. Utilization of modified polar coordinates for bearings-only tracking. *IEEE Transactions on Automatic Control*, AC-28(3), March 1983.

[2] Hans Jacob S. Feder, John J. Leonard, and Christopher M. Smith. Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research*, 18(7):650–668, July 1999.

[3] Michael D. Feezor, Paul R. Blankinship, James G. Bellingham, and F. Yates Sorrell. Autonomous underwater vehicle homing/docking via electromagnetic guidance. In *Oceans '97 MTS/IEEE Proceedings*, volume 2, pages 1137–1142, Halifax, October 1997.

[4] Arthur Gelb, editor. *Applied Optimal Estimation*. The M.I.T. Press, Cambridge, Massachusetts, 1974.

[5] N. Jeremy Kasdin. The two-step optimal estimator and example applications. In Robert D. Culp and Eileen M. Dukes, editors, *Guidance and Control 2000*, volume 104 of *Advances in the Astronautical Sciences*, pages 15–34, San Diego, CA, 2000. American Astronautical Society, Univelt Incorporated.

[6] J.-F. Lots, D. M. Lane, and E. Trucco. Application of 2 1/2 D visual servoing to underwater vehicle station-keeping. In *Oceans 2000 MTS/IEEE*, Providence, RI, September 2000.

[7] R. L. Marks, M. J. Lee, and S. M. Rock. Visual sensing for control of an underwater robotic vehicle. In *Proceedings of IARP Second Workshop on Mobile Robots for Subsea Environments*, Monterey, May 1994. IARP.

[8] S. Negahdaripour, X. Xu, and L. Jin. Direct estimation of motion from sea floor images for automatic station-keeping or submersible platforms. *IEEE Journal of Oceanic Engineering*, 24(3):370–382, July 1999.

[9] Hanumant Singh, Martin Bowen, Franz Hover, Phillip LeBas, and Dana Yoerger. Intelligent docking for an autonomous ocean sampling network. In *Oceans '97 MTS/IEEE Proceedings*, volume 2, pages 1126–1131, Halifax, October 1997.

[10] Stefan B. Williams, Paul Newman, Gamini Dissanayake, and Hugh Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1793–1798, San Francisco, CA, April 2000. IEEE.

[11] Dana R. Yoerger, Albert M. Bradley, Marie-Helene Cormier, William B. F. Ryan, and Barrie B. Walden. High resolution mapping of a fast spreading mid ocean ridge with the autonomous benthic explorer. In *Proceedings of the 11th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, August 1999. Autonomous Undersea Systems Institute.