Relative Position Estimation for AUVs by Fusing Bearing and Inertial Rate Sensor Measurements

Andreas Huster, Eric W. Frew and Stephen M. Rock
Aerospace Robotics Lab, Stanford University, CA, 94305
and Monterey Bay Aquarium Research Institute, Moss Landing, CA, 95039
{huster,ewf,rock}@arl.stanford.edu

Abstract—**This paper describes a relative position sensing strategy that fuses monocular vision (a bearing measurement) with accelerometer and rate gyro measurements to generate an estimate of relative position between a free-floating underwater vehicle and a stationary object of interest. This type of position estimate is a core requirement for intervention-capable autonomous underwater vehicles. These vehicles can perform autonomous manipulation tasks, during which the vehicle needs to control its position relative to objects in its environment. For free-floating underwater vehicles, camera motion is generally unknown and must be estimated together with relative position. Various vision-only systems have been used to estimate relative position and camera motion, but these are 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 potentially 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 when the measurements are integrated. This problem is non-linear, which presents several challenges in the estimator design. Finally, sufficient camera motion is required for the estimator to converge, which necessitates the design of a suitable trajectory.

This paper discusses some of the implementation challenges, outlines an estimation algorithm that is uniquely adopted for this sensor fusion problem, develops a method to generate useful vehicle trajectories, and presents some results from laboratory experiments with a testbed manipulator system. For these experiments, the estimator was implemented as part of a closed-loop control system that can perform an object pick-up task.

I. Introduction

Relative position estimation and control are core requirements for *intervention-capable* autonomous underwater vehicles (AUVs), an emerging class of underwater vehicles capable of performing autonomous manipulation tasks, such as placing sensors and retrieving samples. This paper focuses on a new relative position sensing system based on a simple, robust vision system and inertial rate sensors (accelerometers and rate gyros). The goal is to develop an overall solution that is sufficiently robust to function in real, operational underwater environments, which pose significant challenges for the implementation of vision-based sensing.

The sensing system fuses bearing information (e.g., from monocular vision) with inertial rate sensor measurements to estimate relative position, velocity and attitude. The estimate is computed in real time and is suitable for closed-loop control. This system relies on very simple vision requirements: tracking a single feature in a single camera image. The underwater environment exacerbates the typical challenges of identifying good visual features, establishing feature correspondences and robust tracking. Reducing the vision requirements by integrating inertial rate sensors has the potential to produce a more robust underwater sensing system than typical vision-only techniques.

Our previous work has demonstrated that monocular vision is a very robust and versatile sensor in the ocean environment (e.g., [1] for vision-based ROV position control and [2] for sea floor mapping). This has motivated us to explore monocular vision as a position sensor to enable autonomous manipulation tasks.

The bearing 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 velocity, position and attitude, can account for the 6-DOF motion of the camera along this baseline. When these measurements are fused, the relative position between the observer[1] 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 sensors on underwater vehicles.

Inertial rate sensors suffer from bias and random noise errors that lead to unbounded drift during a simple integration of the measurements to account for observer motion. While greater accuracy can be achieved with more expensive sensors, our approach is compatible with low-cost inertial rate sensors, which are subject to significant drift errors. We have designed an estimator that explicitly accounts for these inertial rate measurement errors.

The estimation problem has to handle two significant nonlinearities. The first is related to the rotational motion of the observer 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 nonlinear and depend on the actual state of the system. In fact, the estimator exploits the non-linearity of the problem to resolve the range to the feature. As motion of the observer 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.

The ability of this estimator to converge depends on the motion of the observer. For example, during camera translation directly towards the feature, the estimator has no new information with which to improve its range estimate. Only camera motions transverse to the feature direction provide observability for the range estimate. However, motion directly toward the object is typically required to complete a manipulation task.

[1] In this paper, observer refers to the system that includes the sensors and the estimator, and not the estimation algorithm itself.

0-7803-7534-3/$10.00

Fig. 1. OTTER: a small underwater vehicle operated in MBARI's test tank

The paper presents a method to generate trajectories that blend these competing objectives. The method is based on an optimization that attempts to minimize the predicted error covariance, subject to completing the manipulation task.

This paper presents three significant extensions of our previous work. In [3], we argued that standard non-linear estimation tools fail to provide adequate solutions for this sensor fusion problem and explored an alternative approach for a simplified (2D) problem. In [4], we discussed a trajectory generator for a relative position estimator that uses measurements from a bearing sensor and assumes known observer motion. In [5], we presented a laboratory testbed and demonstration task to evaluate our sensing system. In this paper, we first outline a new algorithm that can estimate relative position, velocity, attitude, and inertial rate sensor biases in the context of the full 3D problem. Second, we develop a method to design trajectories for this problem. Third, we demonstrate the effectiveness of this relative position sensing strategy with a laboratory experiment performing an object pick-up task using real sensor measurements.

Section II defines the sensor fusion problem and presents models for the vision and inertial rate sensor measurements, the dynamics, and disturbances. Section III discusses relevant problems in non-linear estimation and Section IV describes our estimator design. Section V develops a method to design useful observer trajectories.

We have conducted our initial experiments in the laboratory with a fixed-base 7-DOF manipulator arm, described in Section VI. This platform provides a truth measurement and can be used to investigate competing approaches, to simulate different disturbance environments, and to quantify performance. Section VII demonstrates the effectiveness of our estimator design by combining the estimator with a trajectory and a controller to build a system that can accomplish a simple manipulation task. It also quantifies the performance of the estimator.

Finally, Section VIII states our conclusions and outlines future work. Specifically, we are now preparing for an underwater vehicle demonstration on OTTER (see Fig. 1 and [6]), a small AUV operated in a test tank at MBARI, the Monterey Bay Aquarium Research Institute.

II. Sensor Fusion Problem

A. Estimation Scenario

Fig. 2 shows a stationary object and a moving observer, composed of a camera and inertial rate sensors. The camera is tracking the object and the inertial rate sensors are reporting the observer's acceleration and angular velocity. The purpose of the estimator is to determine the relative position and velocity between the object and the observer.

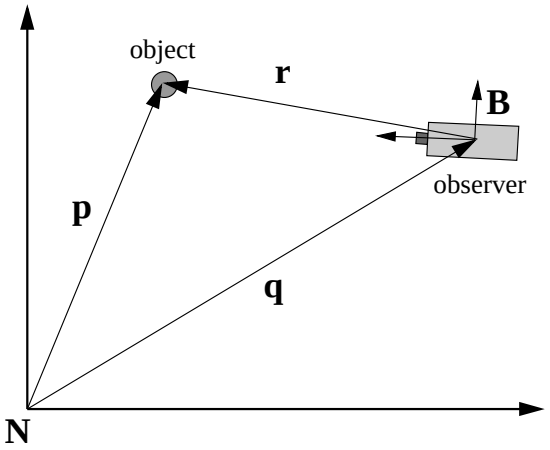

Fig. 2. Geometry of the estimation problem.

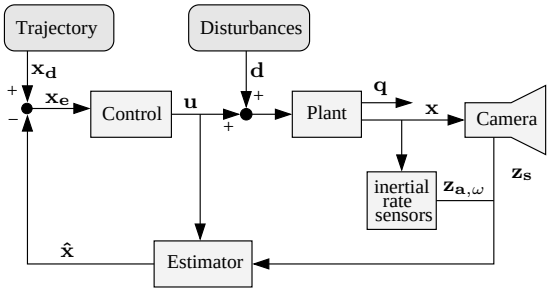

Fig. 3. Block Diagram showing the Estimator, Trajectory, Controller, Plant and Sensors.

Frame $\mathbf{N}$ is the inertial frame and $\mathbf{p}$ indicates the position of a feature on the stationary object tracked by the camera. Let $\mathbf{q}$ be the position of the observer body frame $\mathbf{B}$. To simplify the discussion, we assume that the camera and the inertial rate sensors are all co-located at $\mathbf{q}$. In practice, any known position offset between the sensors and the origin of frame $\mathbf{B}$ can be incorporated into the algorithm. The z-axis of the body frame is aligned with the optical axis of the camera. R is the rotation matrix that transforms a vector resolved in inertial coordinates to the body frame. $\boldsymbol{\omega}$ is the associated rotational velocity resolved in the body frame.

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

We use the superscript B (e.g., ${}^B\mathbf{r} = R\mathbf{r}$) to indicate that the vector is resolved in Frame $\mathbf{B}$ instead of inertial coordinates.

B. System Block Diagram

The block diagram in Fig. 3 shows the entire system composed of the estimator, trajectory, plant and sensors. The camera measurement $\mathbf{z_s}$ and inertial rate sensor 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 $\mathbf{d}$ drive the plant dynamics to generate a new state $\mathbf{x}$.

C. Sensor Models

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

$$
\begin{aligned}
{}^B\mathbf{r} &= R\left(\mathbf{p}-\mathbf{q}\right) && (1)\\
\rho &= {}^Br_z && (2)\\
\mathbf{z}_s &= \begin{bmatrix} s_x \\ s_y \end{bmatrix} = \frac{1}{\rho}\begin{bmatrix} {}^Br_x \\ {}^Br_y \end{bmatrix} + \mathbf{v}_s && (3)
\end{aligned}
$$

where $\mathbf{v}_s$ is zero-mean Gaussian noise. For simplicity, we assume that the camera measurements are normalized so the effective focal length is 1. We will refer to ρ 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 $\mathbf{B}$.

The models for the inertial rate sensors are derived from those in [7]. The accelerometer measures specific force, which includes the acceleration $\ddot{\mathbf{q}}$ of the observer and a component due to gravity. The measurement $\mathbf{z}_a$ also includes sensor biases $\mathbf{b}_a$ and sensor noise $\mathbf{v}_a$.

$$\mathbf{z}_a = R\left(-\ddot{\mathbf{q}} + \alpha\mathbf{g}\right) + \mathbf{b}_a + \mathbf{v}_a \tag{4}$$

We assume that $\mathbf{v}_a$ is zero-mean Gaussian noise. $\mathbf{g} = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$ is the acceleration due to gravity in inertial coordinates. $\alpha \approx 1$ is a scale factor that captures variations in the size of the gravity component and any scale factor errors induced by the sensor. This scale factor is important because the gravity component in the measurement is much larger than the actual acceleration for typical underwater vehicles. Therefore, small errors in the gravity model can lead to large errors in the apparent observer acceleration.

The rate gyro measurement includes the rotational velocity $\boldsymbol{\omega}$ of the observer, sensor biases $\mathbf{b}_\omega$, and zero-mean Gaussian sensor noise $\mathbf{v}_\omega$.

$$\mathbf{z}_\omega = \boldsymbol{\omega} + \mathbf{b}_\omega + \mathbf{v}_\omega \tag{5}$$

We use random walk models to capture the dynamics of the inertial rate sensor parameters.

$$
\begin{aligned}
\frac{d}{dt}\mathbf{b}_a &= \mathbf{n}_a && (6)\\
\frac{d}{dt}\mathbf{b}_\omega &= \mathbf{n}_\omega && (7)\\
\frac{d}{dt}\alpha &= n_\alpha && (8)
\end{aligned}
$$

$\mathbf{n}_a$, $\mathbf{n}_\omega$ and n_α are Gaussian white noise driving terms.

D. Dynamics and Disturbances

For simplicity, we assume $1/s^2$ dynamics for observer position and a trivial actuator model. Therefore, the velocity is driven by a known control input $\mathbf{u}$ and a disturbance $\mathbf{d}$ which represents unknown external forces on the observer (e.g. ocean currents) as well as errors in the actuator model.

$$\frac{d}{dt}\dot{\mathbf{q}} = \mathbf{u} + \mathbf{d} \tag{9}$$

The disturbance is modeled by a first-order Gauss-Markov process.

$$\frac{d}{dt}\mathbf{d} = -\frac{1}{\tau}\mathbf{d} + \mathbf{n}_d \tag{10}$$

where $\mathbf{n}_d$ represents Gaussian white noise.

If $\boldsymbol{\lambda}$ represents the observer attitude, then we can write

$$\frac{d}{dt}\boldsymbol{\lambda} = E\left(\boldsymbol{\lambda}\right)\boldsymbol{\omega} \tag{11}$$

We have chosen simple models for dynamics, actuators, disturbances, sensor biases, and sensor noise. More sophisticated models should be used when they are available and performance improvements warrant them.

III. Non-Linear Estimation Issues

The design of an accurate non-linear estimator for the sensor fusion problem described in the previous section is the main focus of this research. Equations (1) to (11) include several non-linearities as a result of the rotational motion of the observer and the projection in the camera model. Therefore, the design of the sensor fusion algorithm requires a non-linear estimator, like the Extended Kalman Filter (EKF) [8]. However, while the EKF is a useful solution for many non-linear estimation problems, we have shown in [3] that the direct application of the EKF to this problem fails to generate an adequate solution.

There are many methods to design non-linear estimators (e.g., [9], which also considers a relative position estimation problem based on measurements from a bearing sensor and inertial rate sensors). However, our work has focused on appropriate modifications to the Kalman Filter (a common estimator design for linear problems). The Kalman Filter has many advantages: it is a flexible method that can handle complex problems; and the design can take advantage of a significant body of engineering solutions in the literature.

The Kalman Filter, and all of its modifications, like the EKF and those discussed below, propagate a state estimate $\hat{\mathbf{x}}_i = E\left(\mathbf{x} \mid \mathbf{z}_0 \ldots \mathbf{z}_i\right)$ and a covariance matrix $P_i = E\left(\left(\hat{\mathbf{x}}-\mathbf{x}\right)\left(\hat{\mathbf{x}}-\mathbf{x}\right)^T \mid \mathbf{z}_0 \ldots \mathbf{z}_i\right)$ forward in time according to the system dynamics (time update). New sensor measurements are used to correct the estimate and covariance (measurement update). At a given time-step i, the estimate and covariance capture all of the information about the system, including all past measurements (which is indicated by the conditioning on $\mathbf{z}_0 \ldots \mathbf{z}_i$).

This property of the Kalman Filter framework leads to a complication for this sensor fusion problem. The problem belongs to a class of non-linear estimators for which the dependence of the measurement model on the state (e.g., through observer motion) is essential to establish system observability. In this problem, the estimator has to perform an implicit triangulation using bearing measurements from two different locations which are separated by a significant period of time (i.e., the time required to traverse from the first view of the object to the second[2]). If a measurement $\mathbf{z}_j$ is useful only together with a second measurement $\mathbf{z}_k$ at a later time-step $j < i \leq k$, the information in $\mathbf{z}_j$ must be retained implicitly in the covariances established in P_i until $\mathbf{z}_k$ is incorporated. As a result, the estimator is very sensitive to errors in both the measurement and time updates for this class of problem.

[2] In reality, the estimator incorporates all bearing measurements acquired at the sample rate of the camera, not just those acquired at specific locations, but this simplified example is useful to illustrate the point.

The EKF is not a suitable approach for this problem because errors introduced by approximations in the method contaminate the covariance matrix and lead to strongly biased estimates. The first-order Taylor series expansions used to linearize the measurement equations and dynamics can introduce significant errors, especially when the estimation error is large or the estimates are biased. This distorts the record of previous measurements stored in the covariance matrix, which leads to future estimation errors. Thus, a self-reinforcing process is established in which biased estimates lead to further biases.

Many Kalman Filter modifications attempt to reduce estimation error for non-linear problems by addressing how the non-linear equations in the sensor and dynamics models are linearized. For example, the approximations of the EKF can be improved by using higher-order Taylor series expansions and more careful treatment of the stochastic differential equations [10], but the complexity of these approaches becomes prohibitive for all but the simplest problems. The choice of states used to represent the system can affect the performance of a non-linear estimator by shifting non-linearities from from one type to another or from one part of the system to another (for example, using states that yield a linear measurement update [11], [12]).

Some approaches, most notably the Unscented Kalman Filter (UKF) [13], [14] and particle filters [15] approximate the probability distribution instead of the non-linear equations. These filters represent the estimate uncertainty with a set of points chosen according to the *a priori* statistics and propagate these points through the exact non-linear equations. While particle filters are more flexible and potentially more accurate than the UKF, they use a much larger set of points and are computationally very expensive, especially for larger systems. The UKF is similar to the EKF in computational cost, but tends to generate more accurate results than the EKF.

IV. Estimator Design

Our estimator design is based on a combination of the methods described in the previous section. We are using the Kalman Filter framework with a choice of states, $\mathbf{x}$, that leads to a linear measurement update. Insisting on a linear measurement update transfers all of the non-linearities into the state dynamics. The time update, which captures these dynamics, is implemented with the unscented transform, a component of the UKF.

Implementing a linear measurement update requires a choice of states that leads to a linear sensor model $\mathbf{z} = H\mathbf{x}$, where H is constant. Consequently, the measurement update equations are not a function of the state estimate and are therefore not susceptible to biased estimates and large initial conditions. The remaining source of error in the measurement update is the assumption that the states can be modeled as Gaussian random variables, which is not necessarily true for non-linear problems.

We have a choice in the representation of feature range ρ. Because it does not appear in the sensor model, it is not constrained by the requirement of a linear measurement update and we are free to choose a representation that improves the non-linearities in the dynamics. We represent feature range with $\zeta = 1/\rho$, which leads to low-order polynomials as the dominant non-linearities in the state dynamics. Polynomials tend to result in more accurate estimator time-updates than, for example, ratios, which are induced by representing range with ρ.

The estimator state vector is given by:

$$\mathbf{x} = \begin{bmatrix} s_x \\ s_y \\ \zeta \\ \mathbf{v} \\ {}^B\mathbf{d} \\ \mathbf{b}_a \\ \mathbf{Z} \\ \psi \\ \mathbf{b}_\omega \end{bmatrix} \quad \begin{aligned} \mathbf{v} &= \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = {}^B\dot{\mathbf{q}} = R\dot{\mathbf{q}} \\ {}^B\mathbf{d} &= R\mathbf{d} \\ \mathbf{Z} &= \begin{bmatrix} Z_x \\ Z_y \\ Z_z \end{bmatrix} = R \begin{bmatrix} 0 \\ 0 \\ \alpha \end{bmatrix} \end{aligned} \tag{12}$$

$\mathbf{Z}$ represents the direction of gravity in the body frame modified by the accelerometer scale factor α. ψ is the heading, or rotation about $\mathbf{Z}$. Together, $\mathbf{Z}$ and ψ define the attitude ($\boldsymbol{\lambda} = f(\mathbf{Z}, \psi)$). A similar representation for attitude is described in [16].

The corresponding state dynamics are given by (6), (7), and

$$\begin{aligned} \frac{d}{dt}s_x &= -v_x\zeta + s_x v_z \zeta \\ &\quad + s_x s_y \bar{\omega}_x - (1 + s_x^2)\bar{\omega}_y + s_y \bar{\omega}_z && (13) \\ \frac{d}{dt}s_y &= -v_y\zeta + s_y v_z \zeta \\ &\quad + (1 + s_y^2)\bar{\omega}_x - s_x s_y \bar{\omega}_y - s_x \bar{\omega}_z && (14) \\ \frac{d}{dt}\zeta &= v_z\zeta^2 + \zeta s_y \bar{\omega}_x - \zeta s_x \bar{\omega}_y && (15) \\ \frac{d}{dt}\mathbf{v} &= \mathbf{u} + {}^B\mathbf{d} - \bar{\boldsymbol{\omega}} \times \mathbf{v} && (16) \\ \frac{d}{dt}{}^B\mathbf{d} &= -\frac{1}{\tau}{}^B\mathbf{d} - \bar{\boldsymbol{\omega}} \times {}^B\mathbf{d} + \mathbf{n}_d && (17) \\ \frac{d}{dt}\mathbf{Z} &= -\bar{\boldsymbol{\omega}} \times \mathbf{Z} + \frac{1}{\alpha}\mathbf{Z}n_\alpha && (18) \\ \frac{d}{dt}\psi &= \frac{1}{\alpha\left(Z_y^2 + Z_z^2\right)} \begin{bmatrix} 0 & Z_y & Z_z \end{bmatrix} \bar{\boldsymbol{\omega}} && (19) \\ \bar{\boldsymbol{\omega}} &= \begin{bmatrix} \bar{\omega}_x \\ \bar{\omega}_y \\ \bar{\omega}_z \end{bmatrix} = \mathbf{z}_\omega - \mathbf{b}_\omega - \mathbf{v}_\omega && (20) \end{aligned}$$

Note that the rate gyro measurement $\mathbf{z}_\omega$ is used directly in the dynamics. The noise sources $\mathbf{v}_\omega$, $\mathbf{n}_d$, $\mathbf{n}_a$, $\mathbf{n}_\omega$ and n_α represent process noise. In practice, ψ is updated with a difference equation that avoids the singularity when $Z_y = Z_z = 0$.

We use the square-root version of the unscented transform [14] to propagate the estimate and covariance forward in time. We chose this algorithm based on accuracy, ease of implementation, and computational efficiency.

The camera measurements $\mathbf{z}_s$ and the accelerometer measurements $\mathbf{z}_a$ are incorporated with a linear measurement update. Although the accelerometer measurement could be used directly in the dynamics (as we have done for the rate gyro measurement), including it in the measurement update allows us to take advantage of the disturbance model and the known actuator commands to reduce estimation error.

The measurement update is implemented with an array algorithm [17]. The combination of an array algorithm for the measurement update and the square-root version of the unscented

transform for the time-update leads to an algorithm that operates on and stores only the square-root of the covariance, and not the actual covariance, which results in better numerical properties and reduced computational cost.

This estimator was designed to achieve two key properties: low estimate bias errors and accurate prediction of the error covariance. The first property is important because bias errors tend to represent the largest source of estimation error in this type of non-linear system. The second property is useful for observer control and trajectory design. For example, the size of the predicted variance for the range estimate could be used to switch between an exploratory and grasping phase of the vehicle during a manipulation task. The predicted error covariance will be used directly for the trajectory design in the next section.

V. Trajectory Design

The convergence of this estimation problem depends on the motion of the observer. In particular, only components of motion perpendicular to the line of sight to the object enable observability of range. Therefore, we need a trajectory design that ensures that the observer realizes sufficient motion for the estimator to converge. Beyond convergence, optimization of the trajectory can also improve the estimator performance. This section explores some of the issues in generating trajectories for this problem and outlines a design method.

Much of the literature on trajectory design for bearing sensors has come from the passive sonar tracking community. In [4], we have discussed this issue in the context of a bearing measurement from vision and known observer motion, with additional constraints due to non-holonomic observer dynamics and related visibility constraints. In this case, the solution to the trajectory design problem is an optimization of an observability metric over a set of feasible observer paths. Our preliminary work focused on the Fisher Information Matrix as an observability metric. But further work has shown that the predicted error covariance is a better measure of the trajectory quality because it captures the expected behavior of the estimator more accurately.

We have adopted a similar approach for our current problem, but have had to modify the implementation to account for some important differences. First, we are considering a three-dimensional problem with no kinematic constraints, so the set of feasible paths is much larger. Also, the complexity of evaluating a given path is greater. Therefore the complexity of the optimization increases dramatically and the method described in [4] becomes intractable. Second, we have to incorporate new task-specific constraints. For instance, many intervention tasks, including manipulation, require that the observer reaches a given relative position at the end of the trajectory.

In order to incorporate the task-specific constraints and to reduce the computational cost of the trajectory design, we represent the set of feasible paths with a superposition of motion primitives, each defined by parameters that can then be optimized. Using properly designed primitives ensures that only motions that satisfy the constraints, like reaching the desired final relative position, are considered in the optimization. Using primitives to optimize trajectories has already been discussed in [18].

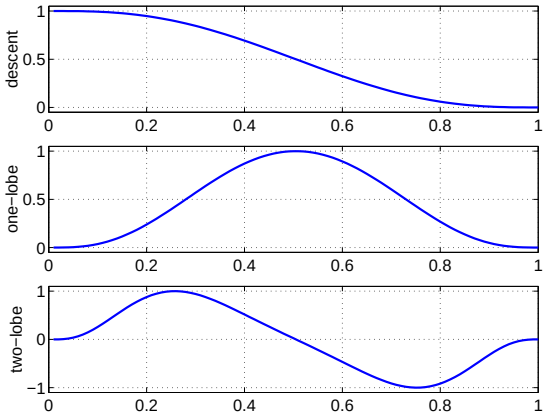

Fig. 4. Motion Primitives.

The primitives were chosen to capture a small number of useful motions. This allows the designer to express heuristics and helps to guide the optimization towards a good trajectory. Fig. 4 shows the three primitives used to assemble the set of feasible paths. The first primitive ("descent") is used to encode the forward motion of the observer that is required to complete our object pick-up task. The other two primitives ("one-lobe" and "two-lobe") are used to generate transverse observer motion. The last two primitives are motivated by much of the previous work in trajectory design for bearing sensors (e.g., [4], [19]).

The 6-DOF trajectory defines the desired observer attitude and relative position that are used by the controller to guide the robot. Each trajectory lasts for $20\,s$ and represents one attempt to pick-up an object with the manipulator. Motion in the final $4\,s$ is predefined to enable grasping of the object. The rest of the trajectory can be optimized to improve estimator accuracy while reducing the amount of observer motion.

Motion towards the object is accomplished with the descent primitive in the z-direction of the body frame (i.e., along the optical axis). Transverse motion is accomplished by superimposing the one-lobe and two-lobe primitives in the pitch, roll, and y-direction. For each primitive, the start time, total duration, and magnitude can be varied. The yaw and x-direction are held constant, which keeps the observer motion in a plane. In this way, a vector of 20 parameters is used to describe each trajectory. Based on initial experience, we defined a default parameter vector which describes a useful trajectory—we refer to this as the default trajectory.

This parameterization of candidate trajectories together with a cost function is used to perform an optimization. The cost is the linear combination of integrated predicted variance of the ζ-state of the estimator, final predicted ζ variance, final predicted bias variance, and integrated control effort. These four components are scaled based on the corresponding values for the default trajectory.

One result of the trajectory optimization is shown in Fig. 5. Each trace shows a trajectory in the x-z plane of the inertial frame. Here, the x-direction points towards the object and the z-direction points up. The optical axis of the camera is indicated by guides spaced at $0.5\,s$ intervals along the traces. The trajectory start and finish, as well as the object location, are

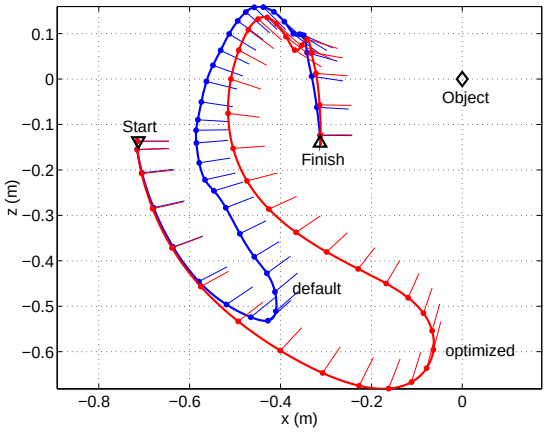

Fig. 5. Observer trajectories in the x-z plane.

defined by the problem. The plot shows the default trajectory and an optimized trajectory. The optimization represents a reduction in cost of approximately 20% relative to the default trajectory. It is a trade-off towards increased observer motion in favor of lower estimation error.

Our trajectory design method generates sufficient observer motion while enforcing task-specific constraints, like terminating at a given relative position. The method is based on a trajectory parameterization that uses motion primitives. We defined a cost function to balance estimator accuracy with the amount of observer motion. While we were not able to compute a global optimum because of local minima, our optimization was able to generate new trajectories that represent a significant improvement over our default trajectory.

VI. Laboratory Testbed

The goal of this research is to develop a relative position sensing strategy that is useful for intervention-capable AUVs. In fact, the sensor measurements used by our algorithm were chosen because they are typically available on these robots. However, in our initial work, we have worked with a fixed-base manipulator in the laboratory. The manipulator provides a truth measurement of position, which is usually not available when using operational vehicles.

Fig. 6 shows the K-1607 manipulator built by Robotics Research Corporation[3]. It is a 7-DOF, kinematically redundant manipulator whose endpoint can be moved to any position and attitude in its workspace. We have attached a camera and inertial rate sensors (DMU-6X Inertial Measurement Unit by Crossbow[4]) on the endpoint of the manipulator in order to demonstrate the estimator performance in the context of real sensor measurements. All of the manipulator joints are instrumented with encoders, so that the exact position of the endpoint can also be computed.

While vision processing for underwater environments remains a challenging problem, many researchers have already developed useful algorithms that can track features on various types of objects in these environments. In our current research, we assume that a point-feature can be tracked robustly, and focus instead on integrating this type of measurement into a position estimator. For this experiment, we use an infrared LED to simulate a visual feature. The camera has an infrared filter which blocks most of the ambient light. When the LED is in the camera's field-of-view, the resulting image contains a bright spot that can be tracked by simple threshold methods.

[3]http://www.robotics-research.com
[4]http://www.xbow.com

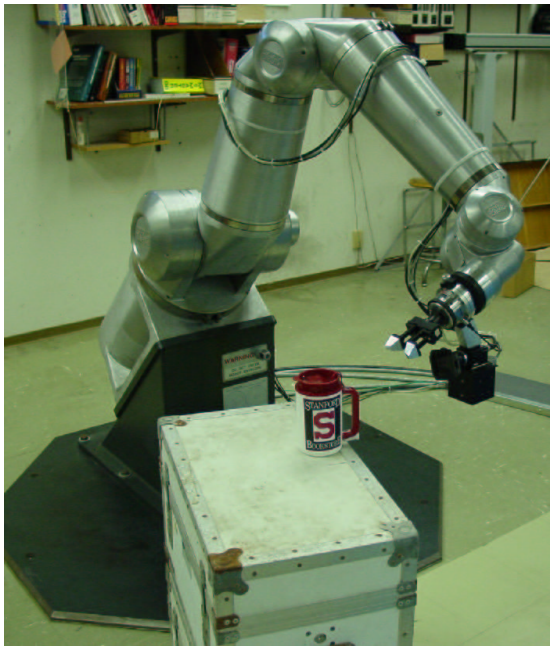

Fig. 6. RRC K-1607 7-DOF Manipulator.

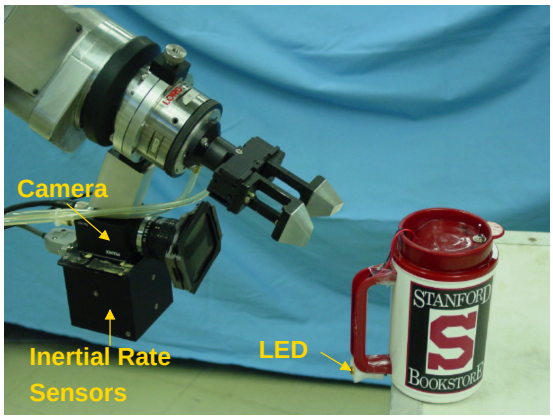

Fig. 7. Manipulator endpoint showing the camera, inertial rate sensors, and gripper and the cup with infrared LED.

We have developed a simple robotic task—picking up an object—to demonstrate the relative position estimator. The object is a large plastic cup which the robot can pick up using a pneumatic gripper. The manipulator endpoint, with the gripper, camera, and inertial rate sensors, as well as the cup are shown in Fig. 7. An LED on the handle of the cup is the only feature that the camera can see. The relative position estimate obtained by fusing the bearing to this LED with inertial rate sensor measurements is used to control the motion of the robot.

The controller in Fig. 3 is a trajectory-follower with a feedforward component to follow the nominal trajectory and a feedback component to account for deviations from the nominal. This controller uses only the state estimate and is not

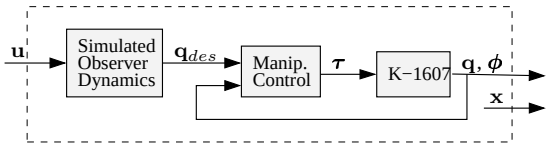

Fig. 8. Details of the Plant Implementation.

specific to the K-1607 manipulator implementation.

The purpose of this testbed system is to simulate observer motion (i.e., the "plant" in Fig. 3) at the manipulator endpoint. The input to the plant is a vector $\mathbf{u}$ of forces and torques expressed in the observer body frame. The output is composed of $\mathbf{q}$, the resulting observer position[5], and $\mathbf{x}$, the system state.

Fig. 8 shows how the K-1607 manipulator is used to implement the plant in Fig. 3. First, arbitrary observer dynamics are simulated to generate the desired observer position $\mathbf{q}_{des}$. Currently, we simulate $1/s^2$ dynamics (which are typical for vehicles like OTTER), but more elaborate models could be used. The desired position is then passed to the manipulator controller, which computes joint torques $\boldsymbol{\tau}$ with full knowledge of the current joint angles $\boldsymbol{\phi}$. This is an accurate, high bandwidth feedback controller running at 200Hz. For the purpose of this experiment, we can assume an identity transformation from $\mathbf{q}_{des}$ to $\mathbf{q}$. Knowledge of the actual robot joint angles is used only within the plant block to simulate observer motion and as a truth measurement.

In a real underwater experiment, the observer would be subject to significant disturbances. These could be included in our laboratory testbed with suitable disturbance models. However, we are not yet injecting any artificial disturbances into the system. Our results in the next section assume that the disturbance input is zero.

VII. Experimental Results

This section presents results obtained from experiments with the K-1607 manipulator. We implemented the estimator as part of the closed-loop control of the manipulator arm, as shown in Fig. 3. The estimator, trajectory and controller for this experiment are described in the previous three sections. The experiment did not use an external disturbance input.

Before each run of the experiment, the observer (manipulator endpoint) is moved to a random initial position $\mathbf{q}_0$ and the estimator is reset to the initial estimate $\hat{\mathbf{x}}_0$, which assumes that the initial target range is $0.65\ m$ and that all other states, including inertial rate biases, are unknown. Throughout the experiment, the feature position $\mathbf{p}$ remains constant. The system then uses the current state estimate to control the observer along a precomputed trajectory (the default trajectory from Section V), which defines the desired observer attitude and relative position. If the estimate $\hat{\mathbf{x}}$ of the state $\mathbf{x}$ of the system is sufficiently accurate, the manipulator succeeds in picking up the object.

Fig. 9 shows the evolution of the relative position estimate $\hat{\mathbf{r}}$ (resolved in the body frame and computed from the state estimate $\hat{\mathbf{x}}$). For each coordinate, the plot shows an overlay of twenty runs, each representing different initial observer positions. The z-coordinate represents the range to the target while the x- and y-coordinates are parallel to the camera's image plane. The plot shows that all the runs start with the same initial estimate $\hat{\mathbf{x}}_0$, that these estimates have different transients to account for the variation in actual relative position, and that they then converge as a result of the closed loop control on the relative position estimate. Because range is poorly observable, the traces on this plot separate and converge more slowly than the other two. The final relative position estimate doesn't approach zero because the gripper (whose relative position does go to zero to pick up the object) is offset from the sensors.

[5]For the purpose of this discussion, position includes the observer attitude.

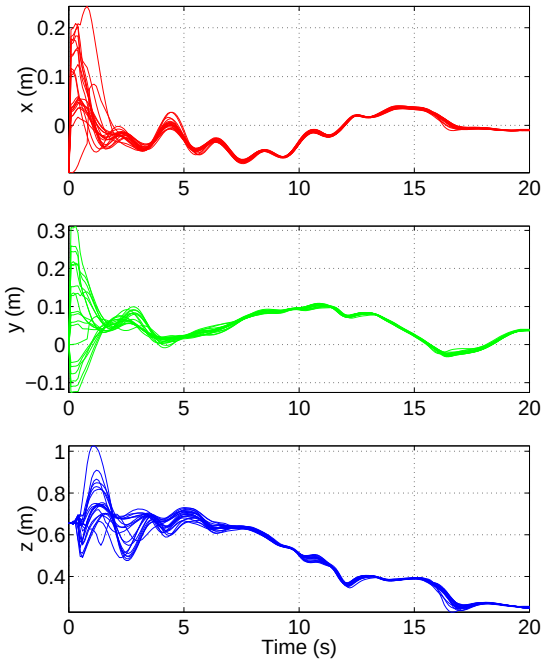

Fig. 9. Relative position estimate for twenty runs, each with different initial conditions.

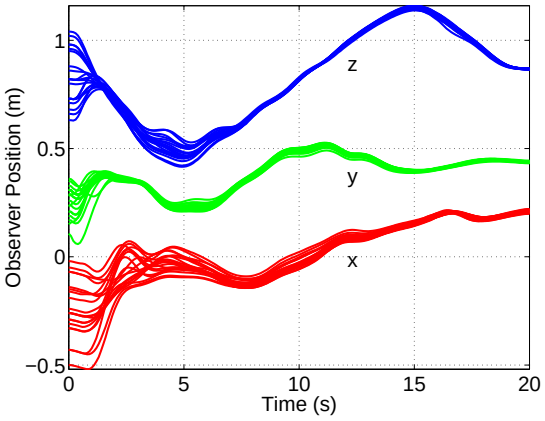

Fig. 10. Observer position for twenty runs, each with different initial conditions.

Fig. 10 overlays the actual observer position $\mathbf{q}$ (a truth value computed from the manipulator joint encoders) for the same twenty runs. The initial observer positions $\mathbf{q}_0$ range from -0.02 to $-0.50\ m$ in x, 0.11 to $0.36\ m$ in y, and 0.63 to $1.04\ m$ in z. This plot shows that the controller, which has access only to the relative position estimate and not the truth measurement $\mathbf{q}$, is successful in moving the observer from various initial positions to the same final position near the object. For all runs, the robot was able to pick up the cup. Table I shows the mean

and standard deviation of the final observer position for these runs.

TABLE I

MEAN AND STANDARD DEVIATION OF FINAL OBSERVER POSITION

	mean (m)	standard deviation (m)
$\mathbf{q}_x$	0.211	0.006
$\mathbf{q}_y$	0.437	0.003
$\mathbf{q}_z$	0.867	0.002

The variation between the runs in Fig. 10 are due to a combination of estimation errors and controller performance. The sensors and the estimator introduce significant delays into the control loop which reduce the controller accuracy. Therefore, the results in Table I represent an upper bound on the estimation error. However, this experiment did not include a disturbance input, which would further degrade the estimator performance.

This experiment shows that the estimator is able to converge using inertial rate sensor measurements and bearing measurements to only one feature on the object. Furthermore, the estimator works well in the context of real sensors. Although the experiment uses an artificially robust vision system, the inertial rate sensors are inexpensive with higher drift rates and larger measurement noise levels than are typically assumed for AUVs.

VIII. CONCLUSIONS

In this paper, we have discussed a sensor strategy that fuses bearing measurements to a fixed object with inertial rate sensor measurements in order to estimate relative position. This capability satisfies a core requirement for *intervention-capable* AUVs: a robust, real-time estimate of the relative position between the free-floating vehicle and a stationary object.

We have defined this sensor strategy, discussed its key advantages, identified some implementation challenges, outlined an estimation algorithm, developed trajectories that generate sufficient observer motion, and performed laboratory experiments to evaluate it. The experimental results demonstrate the effectiveness of this sensor strategy in the context of real sensor measurements. We have shown that the estimator can be used as part of the closed-loop control of a manipulator arm performing an object pick-up task.

Our future work will focus on demonstrating this sensor strategy on OTTER, our test-tank AUV, and on one of MBARI's ROVs. Although we have already demonstrated this capability with the same sensors that will be available on typical underwater vehicles, we anticipate some new issues. Our experiments with the laboratory testbed have assumed a low-disturbance environment and accurate models for the observer dynamics. This assumption needs to be relaxed as we transition to a real underwater vehicle. We also need to couple this capability with a suitable vision processing algorithm that can report bearings to objects that are not specifically marked with an LED or other artificial features.

ACKNOWLEDGMENTS

This research was supported in part by the Packard Foundation under Grants 98-3816 and 98-6228.

REFERENCES

[1] K. N. Leabourne, S. M. Rock, S. D. Fleischer, and R. L. Burton, "Station keeping of an ROV using vision technology," in *Proceedings of the Oceans 97 Conference*, Halifax, October 1997, MTS/IEEE, vol. 1, pp. 634 –640.

[2] R.L. Marks, S.M. Rock, and M.J. Lee, "Real-time video mosaicking of the ocean floor," *IEEE Journal of Oceanic Engineering*, vol. 20, no. 3, pp. 229–241, July 1995.

[3] Andreas Huster and Stephen M. Rock, "Relative position estimation for intervention-capable AUVs by fusing vision and inertial measurements," in *Proceedings of the 12th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, August 2001, Autonomous Undersea Systems Institute.

[4] Eric W. Frew and Stephen M. Rock, "Exploratory motion generation for monocular vision-based target localization," in *Proceedings of the 2002 IEEE Aerospace Conference*, Big Sky, MT, March 2002.

[5] Andreas Huster and Stephen M. Rock, "Relative position estimation for manipulation tasks by fusing vision and inertial measurements," in *Proceedings of the Oceans 2001 Conference*, Honolulu, November 2001, MTS/IEEE, vol. 2, pp. 1025–1031.

[6] H. H. Wang, S. M. Rock, and M. J. Lee, "OTTER: The design and development of an intelligent underwater robot," *Autonomous Robots*, vol. 3, no. 2-3, pp. 297–320, June-July 1996.

[7] Demoz Gebre-Egziabher, *Design and Performance Analysis of a Low-Cost Aided Dead Reckoning Navigator*, Ph.D. thesis, Stanford University, 2001, Chapter 3.

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

[9] Isaac Kaminer, Wei Kang, Oleg Yakimenko, and Antonio Pascoal, "Application of nonlinear filtering to navigation system design using passive sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 1, pp. 158–172, January 2001.

[10] Peter S. Maybeck, *Stochastic Models, Estimation, and Control*, vol. 2, Academic Press, New York, 1982.

[11] Vincent J. Aidala and Sherry E. Hammel, "Utilization of modified polar coordinates for bearings-only tracking," *IEEE Transactions on Automatic Control*, vol. AC-28, no. 3, pp. 283–294, March 1983.

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

[13] Simon Julier, Jeffrey Uhlmann, and Hugh F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, March 2000.

[14] Rudolph van der Merwe and Eric A. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, Utah, May 2001, IEEE, pp. 3461–3464.

[15] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer Verlag, New York, 2001.

[16] Henrik Rehbinder and Xiaoming Hu, "Drift-free attitude estimation for accelerated rigid bodies," in *International Conference on Robotics and Automation*, Seoul, South Korea, May 2001, IEEE, vol. 4, pp. 4244 – 4249.

[17] Thomas Kailath, Ali H. Sayed, and Babak Hassibi, *Linear Estimation*, Prentice Hall, 2000.

[18] Yaakov Oshman and Pavel Davidson, "Optimization of observer trajectories for bearings-only target localization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 892–902, July 1999.

[19] J. M. Passerieux and D. VanCappel, "Optimal observer maneuver for bearings-only tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 3, pp. 777–788, July 1998.