Kalman Filtering for an Aided Inertial Navigation System

by

David A. Korka

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical Engineering and Computer Science

and

Master of Engineering in Electrical Engineering and Computer Science
at the

Massachusetts Institute of Technology

May 21, 1999
[June 1999]

Author _____
David A. Korka
Department of Electrical Engineering and Computer Science

Approved by _____
Dr. Jamie Anderson
Senior Member, Technical Staff, Charles Stark Draper Laboratory
Thesis Supervisor

Certified by _____
Professor George Verghese
Professor, Massachusetts Institute of Technology
Thesis Supervisor

Accepted by _____
Professor Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Kalman Filtering for an Aided Inertial Navigation System

by

David A. Korka

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Electrical Engineering and Computer Science

and

Master of Engineering in Electrical Engineering and Computer Science

May 21, 1999

ABSTRACT

This thesis develops a Kalman filter which integrates the inertial navigation system of the Vorticity Control Unmanned Undersea Vehicle (VCUUV) with redundant navigation-sensor measurements. The model for the Kalman filter uses redundant measurements in a feedback loop to better estimate navigation variables. Using outputs from the Inertial Measurement Unit (IMU) and from a depth sensor, a velocity sensor and a magnetometer, a Kalman filter is developed. Actual test runs on the VCUUV prove the new system superior to the previously used open-loop navigation system.

Thesis Supervisor:	Jamie M. Anderson
Title:	Senior Member Technical Staff, Charles Stark Draper Laboratory
Thesis Supervisor:	George Verghese
Title:	Professor of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

ACKNOWLEDGMENT

May 21, 1999

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Research and Development #99-927.

_____

(author's signature)

I'd like to thank everyone who helped me with this thesis, especially Abigail Mieko Vargus for the hours she spent proofreading.

(This page intentionally left blank.)

Table of Contents

Table of Figures

List of Tables

List of Equations

Chapter 1 Introduction

1.1 The VCUUV Project

The Vorticity Control Unmanned Undersea Vehicle (VCUUV) is a research project being developed at the Charles Stark Draper Laboratory. The VCUUV is an experiment in underwater propulsion. Previous work by David Barrett suggests that fish-like propulsion reduces drag on underwater vehicles, allowing for a more efficient means of propulsion (Barrett). The VCUUV was built for two reasons: first, to determine if fish-like propulsion is, in fact, more efficient than the traditional propeller-based forms of underwater propulsion; second, to explore maneuverability and stability of vorticity-control propelled vehicles.

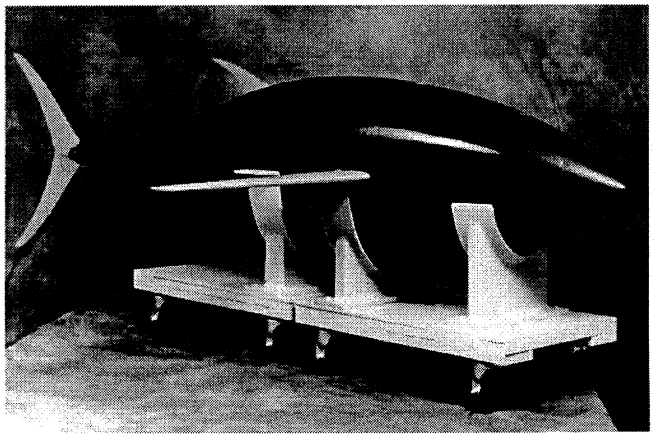

Figure 1: The VCUUV

(Draper Lab Photograph)

The VCUUV was modeled after a yellowfin tuna (see Figure 1). It is eight feet long with a rigid front and a flexible tail. The rigid front section is airtight and contains the electrical systems and the hydraulic pump. The flexible tail consists of a four-degree-of-freedom robot arm covered by an exoskeleton and a neoprene skin (see Figure 2). The links of the tail are actuated by hydraulic cylinders. The rear section of the vehicle floods with water when the vehicle is submerged. The vehicle has been carefully weighted so that when it is submerged and the tail is flooded, the VCUUV is neutrally buoyant.

Propulsion occurs when the electronics package commands the servos to move in a way that creates a traveling wave propagating down the length of the tail. This traveling-wave motion creates forward thrust.

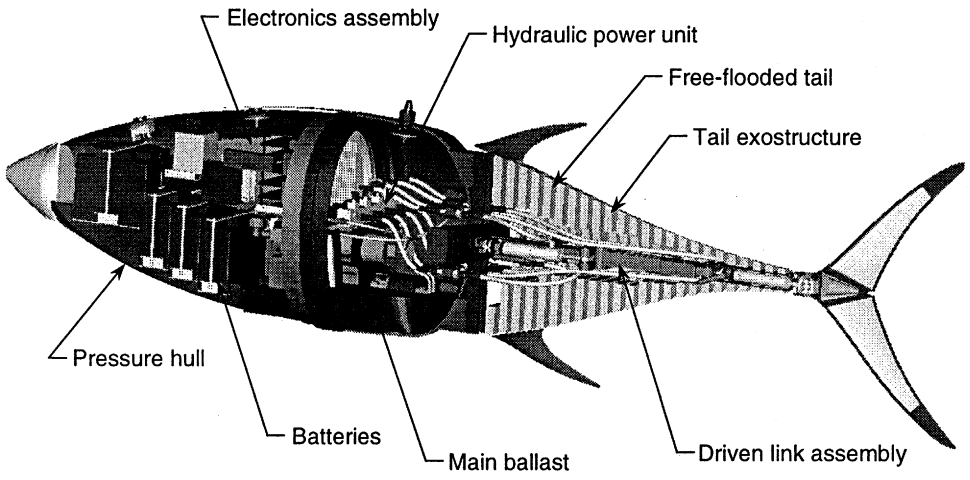

Figure 2: CAD Drawing of the VCUUV

(Draper Laboratory Drawing)

1.2 The Navigation Problem

To answer the question of propulsion efficiency, the vehicle's underwater velocity must be determined. One commonly used way to estimate velocity in land and air vehicles is to use GPS measurements. Since the radio waves that carry GPS signals cannot penetrate water, however, GPS can not be used in a vehicle that is completely submerged. Therefore, another means of navigating must be used for the VCUUV. Examples of underwater navigation systems are acoustic location systems and inertial navigation systems.

In an acoustic location system, sound waves are used to calculate the navigation variables. Two different types of acoustic location systems exist. First, sonar can be used to identify known underwater landmarks to aid positioning. Alternatively, acoustic waves can be used to communicate with fixed buoys in order to triangulate position. Regardless of the type of acoustic system used, external aid is required, either knowledge of the environment (underwater landmarks) or additional external equipment (buoys).

Inertial navigation systems (INS) use internal measurements from an inertial measurement unit (IMU) to compute the navigation variables. The IMU measures the vehicle's accelerations and rotations. By integrating the accelerations twice, the vehicle's position relative to the starting position can be calculated. The entire INS can be mounted on the vehicle, with no external equipment or knowledge required.

Because the VCUUV was designed to be an autonomous vehicle, an inertial navigation system was the best choice for navigation. An INS is not without its problems, however. Due to the multiple integrations involved in the inertial navigation calculations, the error in an INS grows very quickly. The IMU that the VCUUV's INS utilizes is not accurate enough to limit these errors to an acceptable level for this project. The purely inertial solution had errors in velocity that were greater than 10 m/s after 1 minute, and errors in position that were greater than 1 km after 1 minute. Therefore, a purely inertial solution failed.

The VCUUV, fotunately, has other sensors that can be used for navigation. These include a magnetometer, a forward-velocity sensor, and a depth sensor. By combining these sensor outputs with the INS calculations, a much better estimate of navigation variables can be achieved. This type of hybrid system is called an aided inertial navigation system. The objective of this thesis is to design such a system for the VCUUV, through the synthesis of the INS with the other, redundant navigation sensors.

1.3 Thesis Overview

The purpose of this thesis is to develop and implement an algorithm that integrates the various navigational sensor outputs of the VCUUV into an estimate of the vehicle's speed, position, and heading. More specifically, a Kalman Filter will be used to integrate the various measurements obtained from the IMU, magnetometer, and depth sensor to compute the trajectory of the VCUUV. A description of the contents of each of the chapters follows.

Chapter Two, *Data Acquisition*, discusses the acquisition of navigational data. This includes a description of each of the sensors and brief discussion of the signal processing involved in eliminating noise in the system.

Chapter Three, *Navigation Filter Design,* develops the algorithm used for navigation. Starting from the open-loop inertial calculations, it discusses the various issues of integrating the navigational sensors with an Extended Kalman Filter.

Chapter Four, *Performance Evaluation*, presents results of running the navigation algorithm on both simulated data and real data.

Chapter Five, *Conclusion*, recaps the highlights of the navigation filter developed. It also mentions further experiments that could be run to improve the navigation of the VCUUV.

Chapter 2 Data Acquisition

2.1 Sensors

The VCUUV contains several sensors that detect navigational data. These sensors are an inertial measurement unit (IMU), a magnetometer, a depth sensor, and a forward-velocity sensor. Figure 3 shows where each of these devices is located on the VCUUV.

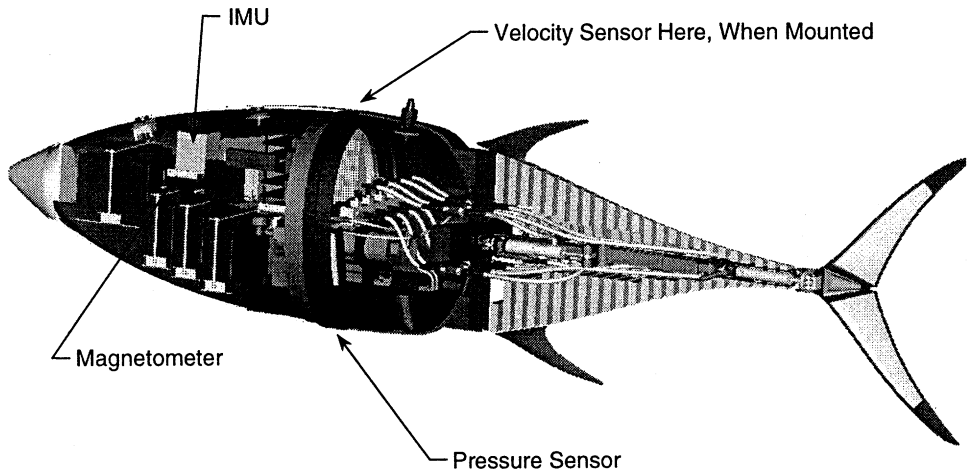

Figure 3: Location of Navigation Sensors on the VCUUV

2.1.1 Inertial Measurement Unit (IMU)

The IMU used in the VCUUV is the MotionPak made by Systron Donner. It consists of three accelerometers, three rate gyros, and an internal temperature sensor. The accelerometers and gyros are oriented along three orthogonal axes. This allows the IMU to measure all rotations and accelerations the vehicle undergoes. The temperature sensor is included because the rate gyros are very sensitive to temperature. To get accurate turn-rate measurements, the temperature must be measured so its effects can be subtracted

from the measurements reported by the IMU. The manufacturer's calibration data for the IMU is shown in Table 1.

Table 1: Manufacturer Calibration Data for the MotionPak

	Angular x-axis	Angular y-axis	Angular z-axis	Linear x-axis	Linear y-axis	Linear z-axis
Range	±200°/s	±200°/s	±200°/s	±5g	±5g	±5g
Scale Factor	12.551 mV/(°/s)	12.518 mV/(°/s)	5.000 mV/(°/s)	1.503 V/g	1.503 V/g	1.504 V/g
Bias (@22°C)	+0.16°/s	+0.07°/s	+0.09°/s	+3.20mg	+5.19mg	+3.20mg
Temp. Perform.	< 3°/s	< 3°/s	< 3°/s	-8µg/°C	-25µg/°C	-4µg/°C
Alignment Error	0.24°	0.64°	0.41°	0.11°	0.12°	0.15°
Bandwidth	78 Hz	75 Hz	74 Hz	901 Hz	885 Hz	981 Hz
Damping	0.69	0.68	0.66	0.89	0.83	0.92
Noise(10-100 Hz)	1mV RMS	1mV RMS	.4mV RMS	.7mV RMS	.9mV RMS	.7mV RMS

The IMU is mounted in a nonstandard orientation. The y-axis of the IMU points to the rear of the vehicle. The z-axis points down. The x-axis points to the vehicle's right. All rotations are positive in the clockwise direction. The standard underwater coordinate system has the x-axis pointing to the front of the vehicle, the y-axis pointing to the vehicle's right, the z-axis pointing down, and all rotations are positive in the clockwise direction. A software fix makes the appropriate changes to the IMU output signals, so that for the rest of this thesis, the standard underwater body coordinate system will apply. Refer to Figure 4 for an illustration of the IMU's orientation and the standard underwater coordinate system.

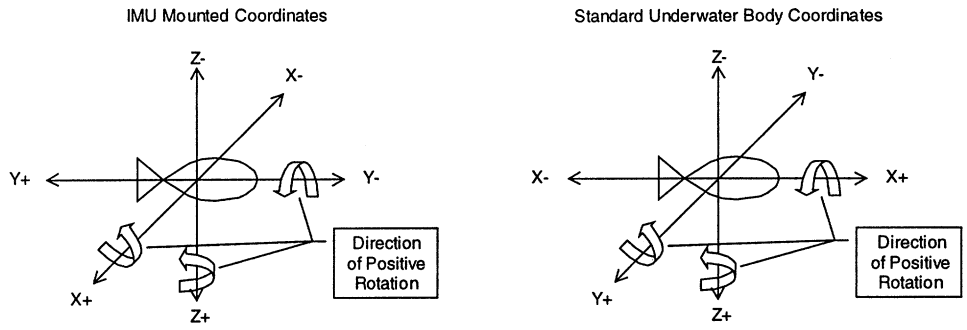

Figure 4: IMU Orientation and Underwater Body Coordinate System

All seven sensors in the IMU have analog outputs. Because the navigation algorithm is implemented on a computer, these signals must be converted to the digital domain. The conversion is achieved with analog-conditioning circuitry and analog-to-digital chips (Ads) on the signal-conditioning board. Section 2.2, Signal-Conditioning Board, discusses this process in detail.

The IMU output is corrupted by electromagnetic interference (EMI), which distorts the analog signals. The EMI comes from the DC/DC converters in the VCUUV's power systems. Additionally, the gyro outputs are very dependent on temperature. Any miscalibration of the temperature relationship will yield errors.

2.1.2 Magnetometer

The VCUUV utilizes the Vector 2XG, a two-axis magnetometer made by Precision Navigation Inc. The output of the magnetometer is inaccurate if the sensitive plane of the device is not parallel to the surface of the Earth. Therefore, the magnetometer is mounted on two gimbals, which allow it to swing up to 15 degrees to remain parallel to the Earth's surface. The magnetometer digitally reports the heading at approximately 1 Hz. The magnetometer has a resolution of 1 degree and an RMS noise of 2 degrees. Errors in this sensor's output occur when the vehicle pitches or rolls beyond 15 degrees, when the sensor rocks on its gimbals due to vehicle movements, and from noise in the magnetometer measurements themselves.

2.1.3 Depth Sensor

The depth gauge is a model 93-015S sealed-gauge pressure sensor, made by EG&G IC Sensors. The depth sensor has a linear range of 0-15 psi (corresponding to 0-33.75 feet of water). The specifications for the sensor claim +/- 0.25% accuracy, or +/- 1 inch. Scale-factor error, bias error, and changes in air pressure affect errors in this sensor's output. The errors due to bias error and changes in air pressure are eliminated by holding the vehicle just under the surface and setting the zero-depth DC value at the start of every trial.

2.1.4 Velocity Sensor

The velocity sensor uses a small propeller mounted on the top of the VCUUV to measure forward velocity. The propeller's turn rate is measured with four magnets attached to its shaft at 90-degree intervals, a fixed Hall-effect sensor, and a PIC (a microcontroller). As the VCUUV moves through the water, the propeller and its shaft spin. Each magnet triggers the Hall-effect sensor as it passes, sending a voltage across the sensor's wires to the PIC. The PIC, located inside the hull of the VCUUV, records the number of times a magnet passes the Hall-effect sensor for each 2-second period. This number is the output of the sensor. 139 counts in one 2-second period corresponds to a vehicle speed of 1 m/s. The velocity of the VCUUV is therefore calculated by dividing the number of counts in a 2-second period by 139. Calibration of this sensor is discussed in Appendix A: Calibration of Speed Sensor.

Sources of error for this sensor include sinusoidal fluctuations due to pitching of the vehicle, inaccurate velocity estimates during turns due to orientation of propeller, and nonlinearities in the rate of rotation of the pinwheel. One of the dynamics of the VCUUV is a 0.2 Hz pitching motion. This pitching varies in amplitude, but is often more than 5 degrees and has occasionally reached 15-20 degrees. Because the propeller is mounted more than a foot away from the axis of rotation, these pitches cause erroneous sinusoidal velocity fluctuations. During quick turns, the water may not be flowing directly at the propeller, since the propeller faces the front of the vehicle. This angled flow leads to a velocity measurement that is slower than the true velocity. Finally, the

rotation rate of the propeller may not be completely linearly related to vehicle velocity. As calibrations could only be done at speeds up to 0.38 m/s, there may be some errors in the scale factor at speeds greater than 0.38 m/s.

2.2 Signal-Conditioning Board

The signal-conditioning board scales, level-shifts, and filters the analog outputs of the IMU and the depth sensor. It does further digital signal processing on the IMU data to eliminate noise. The board then collects data from the various input sensors and sends it to the on-board Intel 486 processor for navigation processing. The scaling and level-shifting are necessary to utilize the entire dynamic range of the AD converters. The analog filters are used to eliminate some of the high-frequency noise in the signals to prevent aliasing when performing the sampling to discrete time.

2.2.1 IMU

With the current scaling and shifting factors, the range and resolution of the measurements from the IMU are shown in Table 2. A discussion of the calculation of the scaling and shifting factors can be found in Appendix B: IMU Conversion Factor Calculation.

Table 2: Range and Resolution of the Scaled and Shifted IMU

	Maximum	Minimum	Resolution
x-acceleration	4.186 m/s^2	-4.184 m/s^2	-0.0020 m/s^2
y-acceleration	4.192 m/s^2	-4.190 m/s^2	-0.0020 m/s^2
z-acceleration	14.135 m/s^2	5.765 m/s^2	-0.0020 m/s^2
x-rotation rate	-0.5092 rad/s	-0.5090 rad/s	-2.5e-004 rad/s
y-rotation rate	-0.5103 rad/s	-0.5101 rad/s	-2.5e-004 rad/s
z-rotation rate	-1.2769 rad/s	-1.2763 rad/s	-6.2e-004 rad/s
Temperature	404.2 K	404.0 K	0.2 K

Note that the range of the z-acceleration does not include 0. As the vehicle is always approximately vertical, this dynamic range is centered around 9.95 m/s^2. This center

allows the best resolution possible because it accounts for gravity. The value of 9.95 m/s^2 was the nearest that could be achieved with the available hardware to the 9.8 m/s^2 magnitude of gravitational acceleration.

Because filtering is done in both the digital and analog realms, the primary purpose of the analog filter is to prevent aliasing when the signals are sampled. Because the VCUUV operates in a medium that damps most high-frequency motions, and because the VCUUV does not make any fast motions, the 0-10 Hz frequency range holds all the relevant data. Therefore, the analog filters were designed with a single pole at 20Hz. Because the sampling frequency is 2 kHz, any high frequency noise is attenuated by 34 dB before it is aliased.

Once the IMU data is converted to the digital domain, it is further filtered on a Motorola 68322 processor before being decimated to 50 Hz, the fastest speed at which data can be sent from the 68322 to the 486.

2.2.2 Depth Sensor

After conditioning by analog circuitry and conversion to the digital domain, the depth sensor has a resolution of 7.5 mm/quantum. The sensor is accurate to within +/- 0.25%. This leads to an accuracy of +/- 3.5 quanta, or 1 inch.

Chapter 3 Navigation Algorithm Design

3.1 Purely Inertial Navigation

Initially, a purely inertial solution to the navigation problem was proposed. This approach involved utilizing the outputs of an inertial measurement unit to calculate relative position, velocity and heading. This was the approach that Mohan Guranathan used in his thesis (Guranathan).

3.1.1 Inertial Navigation Equations

A simple pair of vector differential equations governs inertial navigation (VanBronkhorst). These equations are shown in Equation 1.

$$\dot{\vec{v}} = \vec{a} + g - 2(\vec{\Omega} + \vec{\rho})x\vec{v}$$
$$\dot{\vec{p}} = \vec{v}$$

Equation 1: Differential Equation Governing Navigation in an Inertial System

The position equation is very straightforward. The derivative of the position is simply the velocity. The velocity equation states that the derivative of velocity is the acceleration of the vehicle (a + g) minus the Coriolis correction. The Coriolis correction is comprised of two terms, the correction due to the Earth's rotation and the correction due to the rotation of the navigation coordinate system with respect to inertial space. These two terms are defined in Equation 2 and Equation 3, respectively.

$$\vec{\Omega} = |\Omega| \begin{bmatrix} \cos(\lambda) \\ 0 \\ \sin(\lambda) \end{bmatrix} \qquad \text{Where } |\Omega| = \text{Earth's rotation rate}, \ \lambda = \text{latitude of the vehicle position}$$

Equation 2: Correction due to Earth's Rotation With Respect to Inertial Space

$$\vec{\rho} = \left|\frac{1}{r}\right| \begin{bmatrix} v_E \\ -v_N \\ -v_E \tan(\lambda) \end{bmatrix}$$

Where r = radius of the Earth
v = velocity of the vehicle
λ = latitude of the vehicle position

Equation 3: Correction Due to Rotation of the Navigation Coordinate System With Respect to Inertial Space

3.1.2 Approximations

As the IMU on the VCUUV is rather poor quality, very little accuracy is lost by assuming that the rotation rate of the Earth and the navigation coordinate system is 0. The magnitude of the Earth's rotation is 7.27×10^{-5} rad/s. The magnitude of the rotation of the navigation coordinate system, assuming a maximum vehicle velocity of 10 m/s, is 1.57×10^{-6} rad/s. For comparison, the standard deviation of the error of the gyros is 1×10^{-3} rad/s, three orders of magnitude greater.

This approximation yields the navigation equations actually used, shown in Equation 4.

$$\dot{\vec{v}} = \vec{a} + \vec{g}$$
$$\dot{\vec{p}} = \vec{v}$$

Equation 4: Navigation Equations Implemented

3.1.3 Implementation

The inertial measurement unit in the VCUUV is a strap-down system. In a strap-down system the IMU is rigidly fixed to the hull of the vehicle. This leads to the existence of two coordinate systems: body coordinates and navigation coordinates. The body coordinate system originates at the center of the vehicle. The x-axis points through the front of the vehicle. The y-axis points to the vehicle's right. The z-axis points to the bottom of the vehicle. From this definition, it is apparent that as the vehicle moves with time, the body coordinate system will also move with respect to the navigation coordinate system. The readings from the IMU correspond to these body coordinates. The body coordinate system isn't appropriate for navigation, however, because it is time-varying. Therefore, a fixed navigation coordinate system is necessary. For this project, the navigation coordinate system used has its origin where the vehicle starts its mission. The

x-axis points north. The y-axis points east. The z-axis points down. See Figure 5 for an illustration of these coordinate systems.

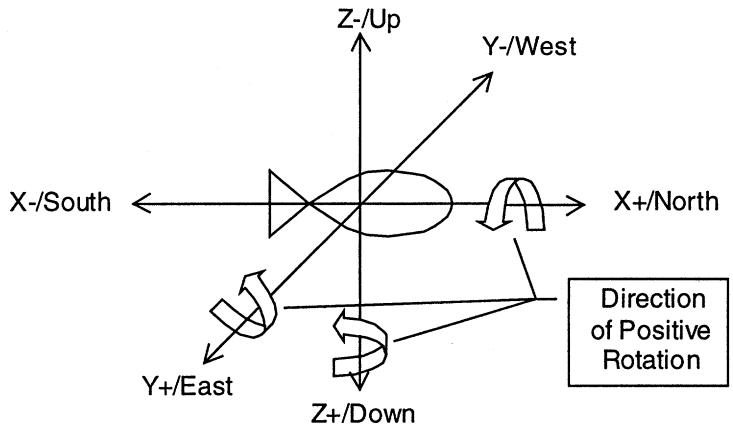

Figure 5: Coordinate Systems When the Body Coordinates Line up With the Navigation Coordinates

The process for calculating a trajectory in the navigation coordinate system is as follows:

- Create a projection matrix at each sample point in time to convert from body coordinates to navigation coordinates.
- Project the acceleration readings from the IMU into the navigation coordinate system.
- Subtract the acceleration due to gravity from the acceleration readings.
- Integrate along each axis in the navigation coordinate system to determine the velocity and position of the vehicle.

3.1.3.1 The Projection Matrix

An initial projection matrix, the *C* matrix, is calculated by using the equation listed in Equation 5 (Weisstein):

$$C_N^B = \begin{bmatrix} \cos(\theta)\cos(\phi) & \cos(\theta)\sin(\phi) & -\sin(\theta) \\ \sin(\Psi)\sin(\theta)\cos(\phi) - \cos(\Psi)\sin(\phi) & \sin(\Psi)\sin(\theta)\sin(\phi) + \cos(\Psi)\cos(\phi) & \cos(\theta)\sin(\Psi) \\ \cos(\Psi)\sin(\theta)\cos(\phi) + \sin(\Psi)\sin(\phi) & \cos(\Psi)\sin(\theta)\sin(\phi) - \sin(\Psi)\cos(\phi) & \cos(\theta)\cos(\Psi) \end{bmatrix}$$

θ – Pitch

Ψ – Roll

ϕ – Yaw

Equation 5: Projection Matrix Defined in Terms of Roll, Pitch, and Yaw

This projection matrix is then propagated by the following set of equations.

$$C_{k+1} = C_k * A_k$$

$$A_k = I_{3x3} + \frac{\sin(\sigma)}{\sigma}[\sigma x] + \frac{1-\cos(\sigma)}{\sigma^2}[\sigma x]^2$$

$$[\sigma x] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} * dt$$

$$\sigma = dt * \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2}$$

$dt = 0.02$ sec (the sampling period)

ω_i is the rotation rates about the i^{th} axis

Equation 6: Propagation Equations for the Projection Matrix

3.1.4 Problems With a Purely Inertial Solution

The main problem with a purely inertial navigation system is that the error in position grows with t^3. This error is caused by the noise of the rate-gyro readings, which leads to a random walk in the heading. These heading errors lead to inaccurate subtraction of the

gravity vector in navigation coordinates, which, in turn, results in increasingly inaccurate estimates of velocity and position.

With the quality of gyros present in the VCUUV, velocity errors of 1 m/s –or more— after just 1 minute of operation are expected. These errors are too large to be used in any meaningful calculations of propulsion efficiency. Therefore, a way to limit these errors must be found.

3.2 Kalman Filter Theory

One way to limit these errors, or at least slow their growth, is to use an algorithm known as the Kalman filter. A Kalman filter is an algorithm that takes a linear system along with measurements of components of or projections of the state of the system to estimate the current state of the system. Figure 6 shows the equations associated with the Kalman filter.

System Dynamic Model :

$$x_k = \phi_{k-1} x_{k-1} + G w_{k-1}$$
$$w_k = N(0, Q_k)$$

Measurement Model :

$$z_k = H_k x_k + v_k$$
$$v_k = N(0, R_k)$$

Initial Conditions :

$$E[x_0] = \hat{x}_0$$
$$E[\tilde{x}_0 \tilde{x}_0^T] = P_0$$

Independence Assumptions : $E[w_k v_j^T] = 0 \quad \forall \quad k \neq j$

State Estimate Extrapolation : $\hat{x}_k(-) = \phi_{k-1} \hat{x}_{k-1}(+)$

Error Covariance Extrapolation : $P_k(-) = \phi_{k-1} P_{k-1}(+) \phi_{k-1}^T + G Q_{k-1} G^T$

State Estimate Observational Update : $\hat{x}_k(+) = \hat{x}_k(-) + K_k[z_k - H_k \hat{x}_k(-)]$

Error Covariance update : $P_k(+) = [I - K_k H_k] P_k(-)$

Kalman Gain Matrix : $K_k = P_k(-) H_k^T [H_k P_k(-) H_k^T + R_k]^{-1}$

Figure 6: Discrete-Time Kalman Filter Equations

Inertial navigation, however, is not linear. Nonlinearity, however, does not mean that a Kalman filter cannot be used. By carefully formulating the problem, and making appropriate approximations, the model can be made linear. This will be discussed in Section 3.3.

3.3 Closing the Loop

After reviewing navigation literature and consulting with Tom Thorvaldsen, a navigation specialist at Draper Laboratory, the navigation scheme illustrated in Figure 7 was chosen. In this design, the trajectory is calculated using the standard open-loop design, as discussed in Section 3.1. This trajectory is compared to redundant measurements of the navigation variables provided by the aiding devices (compass, depth sensor, etc.). The differences between these estimates and the open-loop prediction are treated as the

measurements to a Kalman filter, which estimates the error in the trajectory. These errors are then fed back to the trajectory estimation process to yield more accurate trajectory estimates.

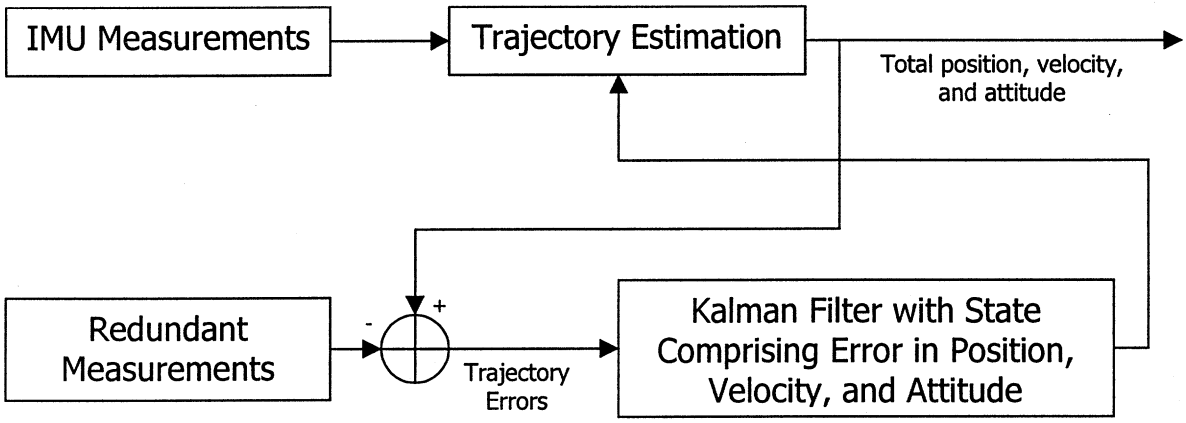

Figure 7: Block Diagram of the INS

3.3.1 Why This Implementation?

This implementation has many factors to recommend it. First, the noise overlaying the trajectory errors can reasonably be modeled as white Gaussian noise. This is critical, because the input measurements to a Kalman filter must be corrupted by only white noise. An implementation that directly used any of the navigation sensors as measurements would not have white measurement noise, eliminating the possibility of using a Kalman filter.

This implementation also has the advantage that any high frequency maneuvers, such as a rapid turn, will be fed straight through without any delay, as might occur in a different feedback setup.

3.3.2 The Kalman Filter

Error dynamics for an inertial navigation system are not truly linear. As long as the errors are small, however, the dynamics can be approximated as linear. These approximations are very accurate for attitude errors of less than 1 radian, and acceleration errors that are small compared to gravity.

The following sections define the variables used in the equations presented in Figure 6 as used in this navigation Kalman filter.

3.3.2.1 The Kalman States

Different texts present error models with widely varying numbers of states. As few as 9 states or as many as 90 states have been used. Because the IMU on the VCUUV has large gyro-drift rates and the various higher-order relationships are not known, a 90-state filter would not be beneficial—or even possible to implement. Therefore, a simple set of states that only use first-order errors is used. Equation 7 lists the state vector.

$$x = \begin{bmatrix} \text{Error in North Position} \\ \text{Error in East Position} \\ \text{Error in Down Position} \\ \text{Error in North Velocity} \\ \text{Error in East Velocity} \\ \text{Error in Down Velocity} \\ \text{Error in North Attitude} \\ \text{Error in East Attitude} \\ \text{Error in Down Attitude} \\ \text{Error in X Gyro Bias} \\ \text{Error in Y Gyro Bias} \\ \text{Error in Z Gyro Bias} \\ \text{Error in X Accelerometer Bias} \\ \text{Error in Y Accelerometer Bias} \\ \text{Error in Z Accelerometer Bias} \end{bmatrix}$$

Equation 7: The Kalman State Vector

3.3.2.2 The Dynamics Matrix

The Discrete-Time Kalman Filter is, not surprisingly, formulated for discrete-time. The movements of the VCUUV are in continuous time. However, Φ, the discrete-time state transition matrix, is equal to $e^{F(t-t')}$, where F is the continuous-time dynamics matrix. An

approximation to allow simple computation of the state transition matix is shown in Equation 8, where $dt = 0.02$ seconds, the time between samples in the discrete-time filter.

$$\Phi = I_{15x15} + F*dt$$

Equation 8: The State Transition Matrix

The F matrix is designed for a 15-state filter. A consistent set of units is used. Position and velocity are measured in meters per second. The error in attitude is measured in radians. The errors in gyro biases and accelerometer biases are measured in radians per second and meters per second^2, respectively.

With the approximations as mentioned in Section 3.1.2 and the units as described above, the following relationships are true. The error in position is simply the integral of the error in velocity. The error in the derivative of velocity, is the sum of two items: the cross product of the specific force vector and the attitude error vector; and the projection of the acceleration biases from body coordinates to navigation coordinates. Finally, the error in the derivative of the attitude error vector is the projection of the gyro biases from body coordinates to navigation coordinates. The result of combining all of these relationships is shown in Equation 9. The f_i are the components of the specific force in the north, east, and down directions. The c_i are the components of the rotation matrix that converts from body to navigation coordinates.

$$
F = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -f_D & f_E & 0 & 0 & 0 & c_{NX} & c_{NY} & c_{NZ} \\
0 & 0 & 0 & 0 & 0 & 0 & f_D & 0 & -f_N & 0 & 0 & 0 & c_{EX} & c_{EY} & c_{EZ} \\
0 & 0 & 0 & 0 & 0 & 0 & -f_E & f_N & 0 & 0 & 0 & 0 & c_{DX} & c_{DY} & c_{DZ} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{NX} & c_{NY} & c_{NZ} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{EX} & c_{EY} & c_{EZ} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{DX} & c_{DY} & c_{DZ} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Equation 9: The *F* Matrix

3.3.2.3 Process Noise

The process noise is any white noise that is added at every step of the process. For this system, there is no process noise in the calculation of the position, as the derivative of position is simply the velocity. The velocity and attitude calculations, however, necessarily include the noise from the IMU measurements of acceleration and rotation rates. These noise sources can be modeled as white Gaussian noise in body coordinates, one noise source per accelerometer and gyro. As the state variables are in navigation coordinates, these sources must be projected into navigation coordinates. This can be done by multiplying by the *C* matrix already computed in the open-loop calculation (see Equation 5 and Equation 6). The gyro bias error and the accelerometer bias error are both modeled as random constants. Because the gyro bias is very slowly changing and the duration of a mission is short, this is a good model. This modeling leads to 0 process noise for these last state variables. Putting everything together leads to the noise model described in Equation 10.

$$Gw = \begin{bmatrix} 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & C_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & C_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} 0_{3x1} \\ w_{XA} \\ w_{YA} \\ w_{ZA} \\ w_{XG} \\ w_{YG} \\ w_{ZG} \\ 0_{3x1} \\ 0_{3x1} \end{bmatrix}$$

$$w_k = N(0, \sigma_k^2)$$

Equation 10: The *G* Matrix

The Q matrix is defined as the covariance of the process noise: $Q = E[ww^T]$. In this system, it is assumed that the noise for each state variable is uncorrelated with the noise for each other state. Hence, all non-diagonal terms of this matrix are zero and the diagonal terms are simply the variances of the random variables.

Table 3 presents a comparison of theoretical values of the variance (calculated from Table 1 with the equation $Q_k = Q*dt$) to values computed from actual data. The values computed from actual data were calculated by taking the variance of a set of data where the vehicle was held motionless. This calculation assumes that the process is ergotic—ensemble averages are equal to time averages—but this assumption seems to be reasonable.

Note that the actual variances are significantly more than the theoretical values. There are two reasons for this discrepancy. First, the approximation that $Q_k = Q*dt$ is not completely accurate. Second, and probably more significant, noise is introduced throughout the transmission lines and signal processing board.

Table 3: IMU Error Variances

Sensor	Theoretical Variance	Actual Variance
X-gyro	$3.86 * 10^{-8}$ (rad/s)^2	$6.08 * 10^{-7}$ (rad/s)^2
Y-gyro	$3.88 * 10^{-8}$ (rad/s)^2	$5.15 * 10^{-7}$ (rad/s)^2
Z-gyro	$3.90 * 10^{-8}$ (rad/s)^2	$9.7 * 10^{-7}$ (rad/s)^2
X-accelerometer	$4.17 * 10^{-7}$ (m/s^2)^2	$9.7 * 10^{-6}$ (m/s^2)^2
Y-accelerometer	$6.88 * 10^{-7}$ (m/s^2)^2	$1.37 * 10^{-5}$ (m/s^2)^2
Z-accelerometer	$4.16 * 10^{-7}$ (m/s^2)^2	$3.33 * 10^{-5}$ (m/s^2)^2

As a final note, the G matrix only occurs at one point in the Kalman filter equations. This occurrence is the GQG^T term in the calculation of $P(-)$. Because C is a projection matrix, $C^T = C^{-1}$. If all the gyros are considered to have identical variances, and all the accelerometers are also considered to have identical variances, the G matrix can be completely ignored. This leads to the GQG^T matrix in Equation 11, where σ^2_A is the variance of the accelerometers and σ^2_G is the variance of the gyros.

$$GQG^T = \begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&\sigma_A^2&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&\sigma_A^2&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&\sigma_A^2&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&\sigma_G^2&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&\sigma_G^2&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&\sigma_G^2&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

Equation 11: The Process Noise Matrix

3.3.2.4 The Initial Covariance Matrix

The initial covariance matrix is calculated by the use of another Kalman filter. This process is discussed in greater detail in Section 3.4.

3.3.2.5 The Measurement Vector

Four redundant measurements of the trajectory are used in the Kalman filter. By using the compass, the error in yaw can be computed. The depth sensor can measure the error in the vertical position. The forward-velocity error can be calculated with the forward velocity sensor. Finally, the vehicle never has any lateral velocity, so a "measurement" of sideways velocity, which always equals 0, can also be included. This set of measurements yields the measurement vector seen in Equation 12.

$$z = \begin{bmatrix} \text{Error in Yaw} \\ \text{Error in Zpos} \\ \text{Error in Forward Velocity} \\ \text{Error in Sideways Velocity} \end{bmatrix}$$

Equation 12: The Measurement Vector

3.3.2.6 The Measurement Transformation Matrix

The measurement transformation matrix, H, is the matrix that transforms the Kalman state vector to the measurement vector. The first three measurements directly measure elements in the state vector. The H matrix reflects this with 1's in the appropriate columns. The velocity measures are in body coordinates, so the north, east, and down velocities must be projected back to body coordinates. This projection is accomplished by using the appropriate rows of C^T. This works because C is a projection matrix, and, therefore, its inverse is simply its transpose.

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{nx} & c_{ex} & c_{dx} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{ny} & c_{ey} & c_{dy} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Equation 13: The Measurement Transformation Matrix

3.3.2.7 The Measurement Noise Matrix

The measurement noise matrix, R, is the covariance of the measurements. The noise for each of the sensors is assumed to be uncorrelated with the noise for the other sensors. Therefore, the nondiagonal elements are 0. Each of the diagonal elements is then the variance for the respective sensor. This is shown in Equation 14.

$$R = \begin{bmatrix} \sigma^2_{Compass} & 0 & 0 & 0 \\ 0 & \sigma^2_{Depth} & 0 & 0 \\ 0 & 0 & \sigma^2_{ForVel} & 0 \\ 0 & 0 & 0 & \sigma^2_{SideVel} \end{bmatrix}$$

Equation 14: The Measurement Noise Matrix

The variances used in the measurement noise matrix are shown in Table 4.

Table 4: Sensor Noise Variances

Sensor	Variance
Compass	0.070 rad^2
Depth Sensor	0.0007 m^2
Forward Velocity Sensor	0.2 $(m/s)^2$
Side Velocity Estimate	0.1 $(m/s)^2$

3.3.3 Integration of Kalman Filter with the Open-Loop System

The Kalman filter states are fed back to the open-loop navigation calculations at a rate of 1 Hz. At that time the Kalman states are set to 0, and the appropriate adjustments are made to the open-loop navigation system.

3.4 System Initialization

One of the problems in an inertial navigation system is calculating the correct initial values for the heading. Any slight errors will propagate quickly, soon making the outputs of the navigation algorithm useless. To help solve this problem, the VCUUV is held still for a known length of time at the beginning of each trial run. As the position, velocity, and acceleration are know to be 0 (or at least very close to 0) for this time, the Kalman filter discussed above, with the added measurements of position can be used to estimate the initial vehicle orientation. This has the added benefit of calculating a covariance matrix which can be used as the initial covariance matrix in the navigation algorithm. The initial covariance matrix used for initialization is shown in Equation 15.

$$
P_0 = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{x}10^{-6} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{x}10^{-6} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{x}10^{-6} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{x}10^{-4} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{x}10^{-4} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1\text{x}10^{-4}
\end{bmatrix}
$$

Equation 15: Covariance Matrix for Initializing the Kalman Filter

Chapter 4 Performance Evaluation

4.1 Simulated Inputs

As a first pass and for debugging purposes, simulated inputs were fed to the algorithm to test its performance and accuracy. The following sections present the results of these experiments.

4.1.1 At Rest

The first test determined the algorithm's convergence when the inputs simulated a vehicle at rest. The measurements (IMU and redundant sensors) were set to 0 mean and corrupted by the expected white Gaussian noise for each type of sensor. Following are plots of the calculated position, attitude, velocity, and speed (the magnitude of the velocity vector) of the vehicle. The plots show the "true" values (the values for what the simulated vehicle actually did), the values obtained using an open-loop calculation, and the values calculated using the Kalman filter.

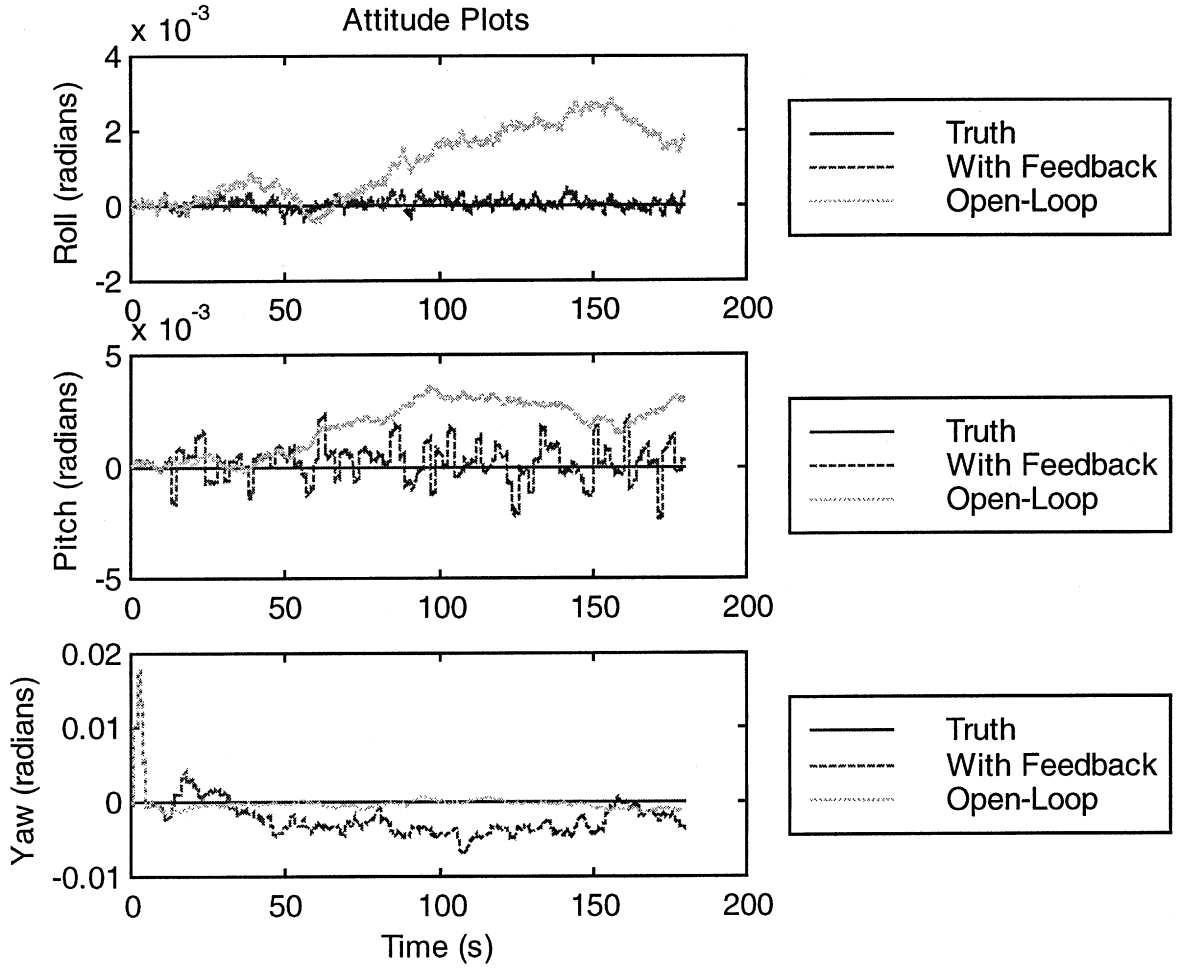

Figure 8: Attitude Plots for Simulated at-Rest Inputs

Figure 8 illustrates that both the roll and pitch of the vehicle are estimated much closer to 0 when the Kalman filter is used. This increased accuracy reduces the error in estimated acceleration. Coupled with the velocity feedback, the improved acceleration estimate serves to reduce the error in velocity, as shown in both Figure 10 and Figure 11.

As the yaw angle is decoupled from the accelerations that the IMU sees, the feedback in yaw provided by the Kalman filter is limited to the noisy compass updates which occur at a rate of 1 Hz. This leads to a larger drift in yaw angle than in the roll and pitch angles.

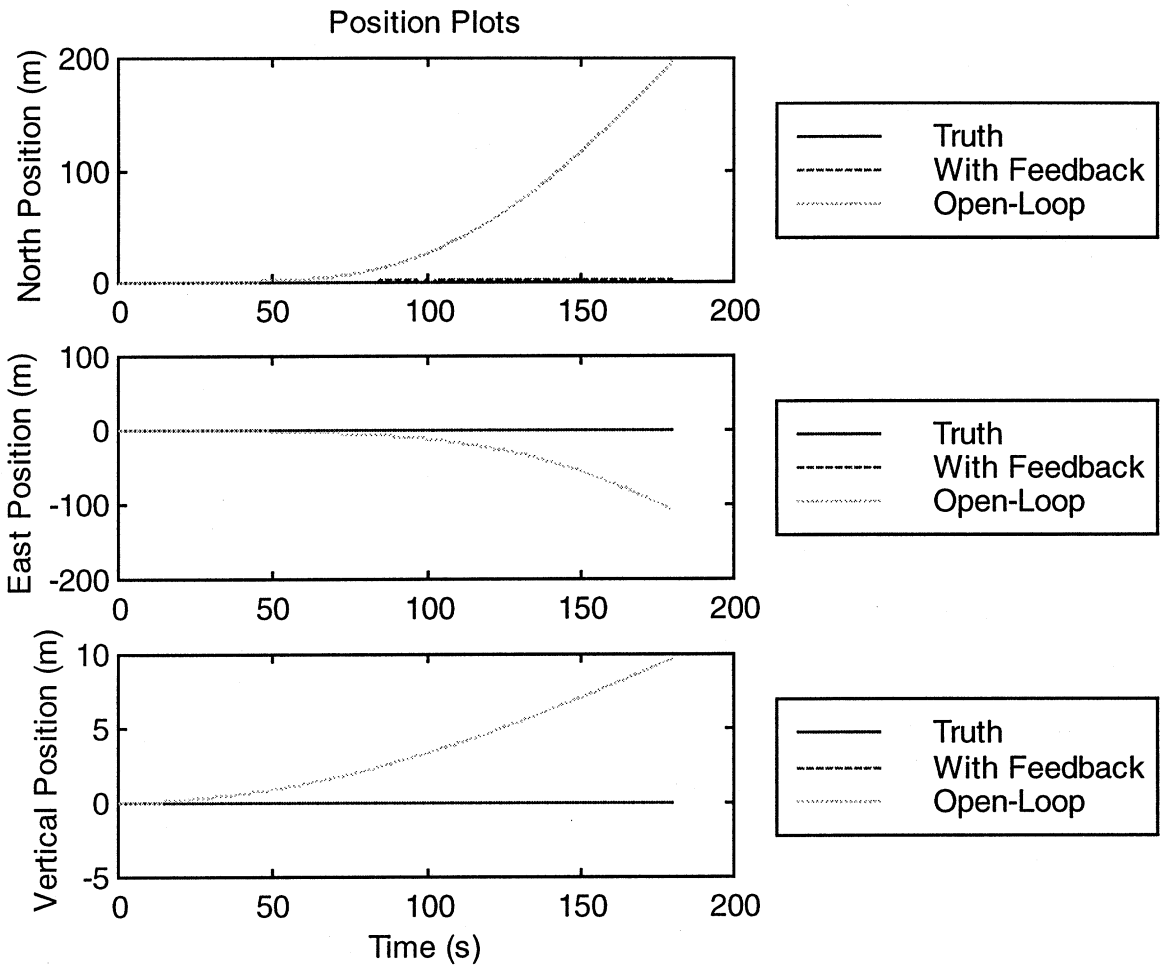

Figure 9: Position Plots for Simulated at-Rest Inputs

Figure 9 shows the position vector in terms of its north, east and down components. The origin of the vector is the initial position of the VCUUV. Figure 9 shows that the open-loop estimates of position are not accurate at all.

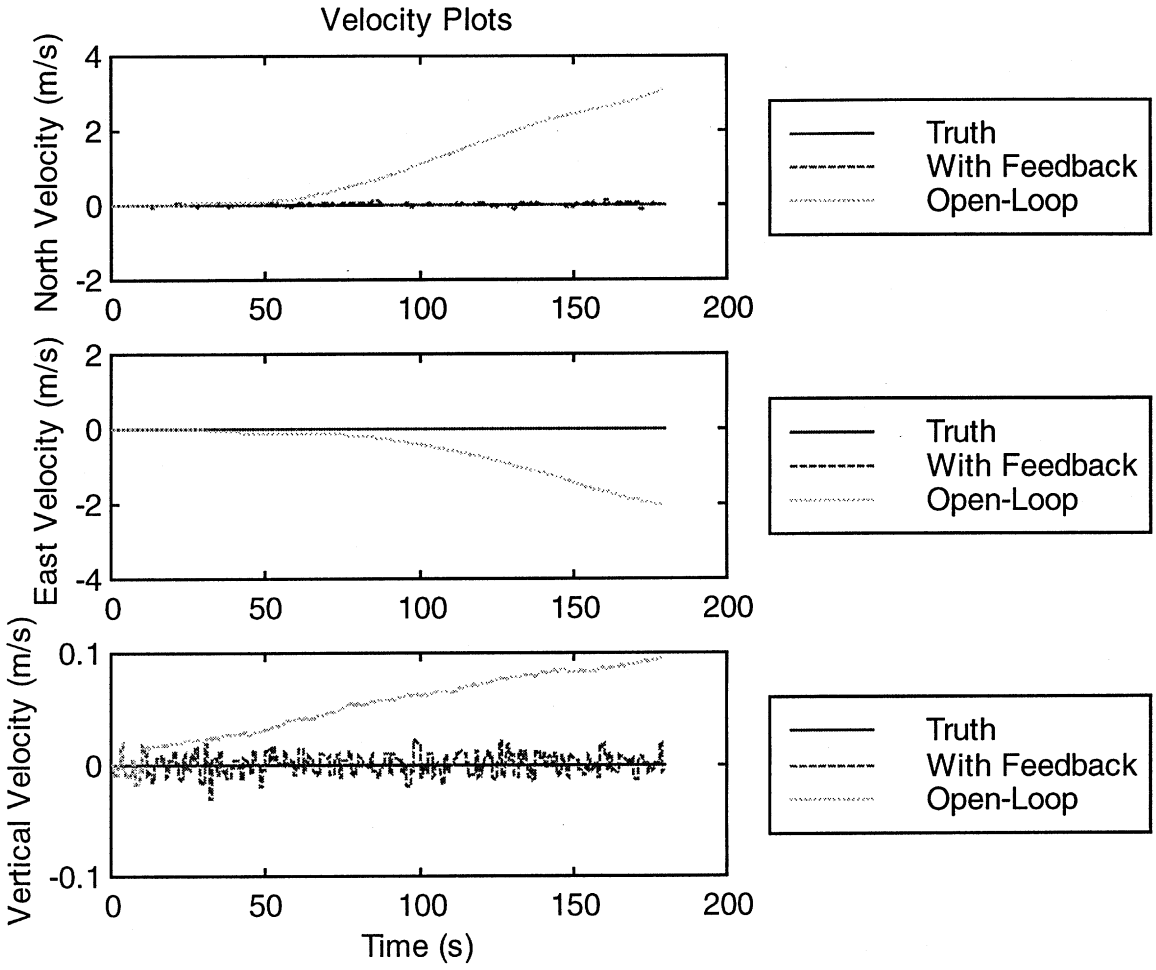

Figure 10: Velocity Plots for Simulated at-Rest Inputs

Figure 10 shows the velocity vector in terms of its north, east, and down components.

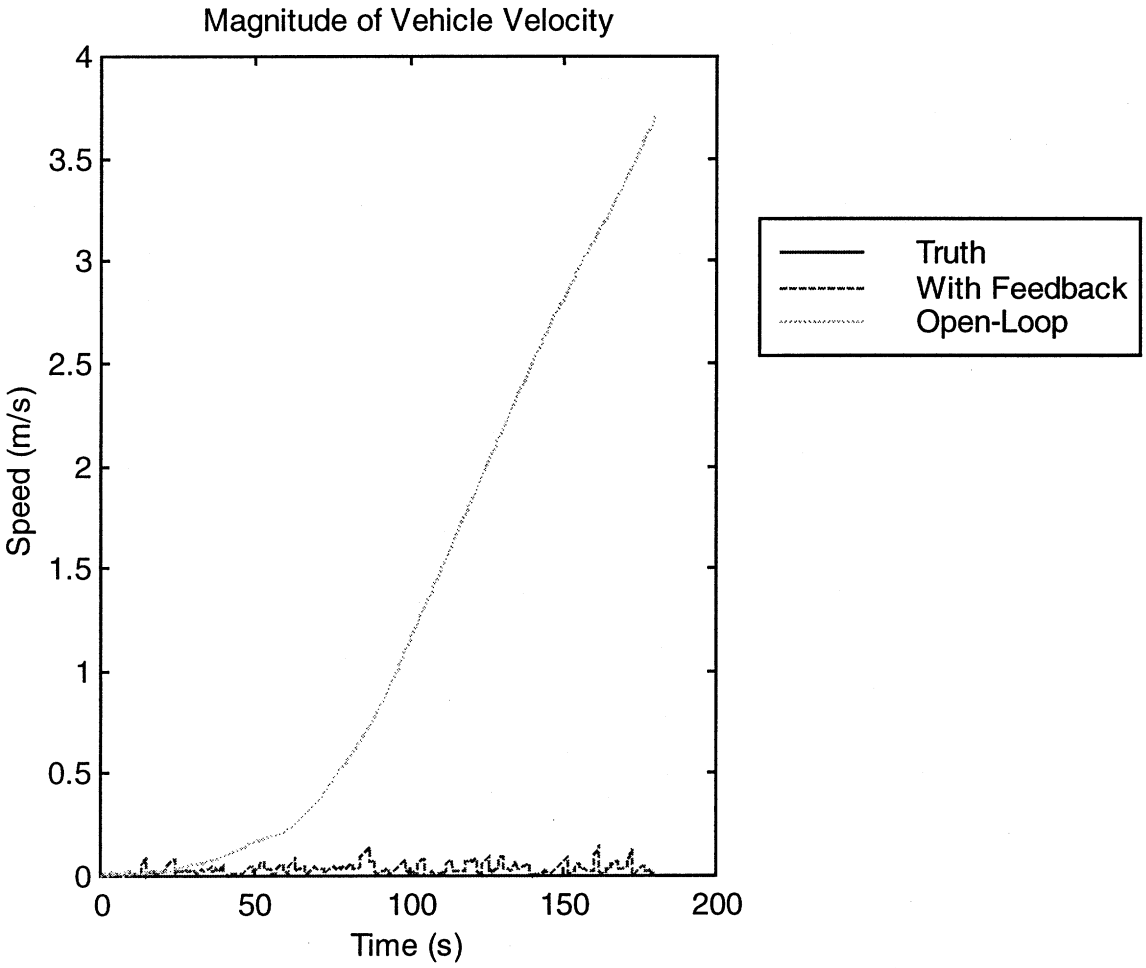

Figure 11: Speed Plot for Simulated at-Rest Inputs

Figure 11 shows the magnitude of the velocity vector. As shown in Figure 11, the error in estimated speed using the open-loop system increases drastically with time, as expected. The Kalman filter's error in estimating speed, however, does not increase with time but stays relatively reliable throughout the trial, again as expected. The figures presented in this section show that the Kalman filter limits the error growth in the simulated at-rest case.

4.1.2 Straight Forward

For the second set of tests, the simulated vehicle accelerated from rest to a constant velocity, and then decelerated to rest. The results from this test follow. Again, plots for the attitude, position, velocity, and speed are shown.

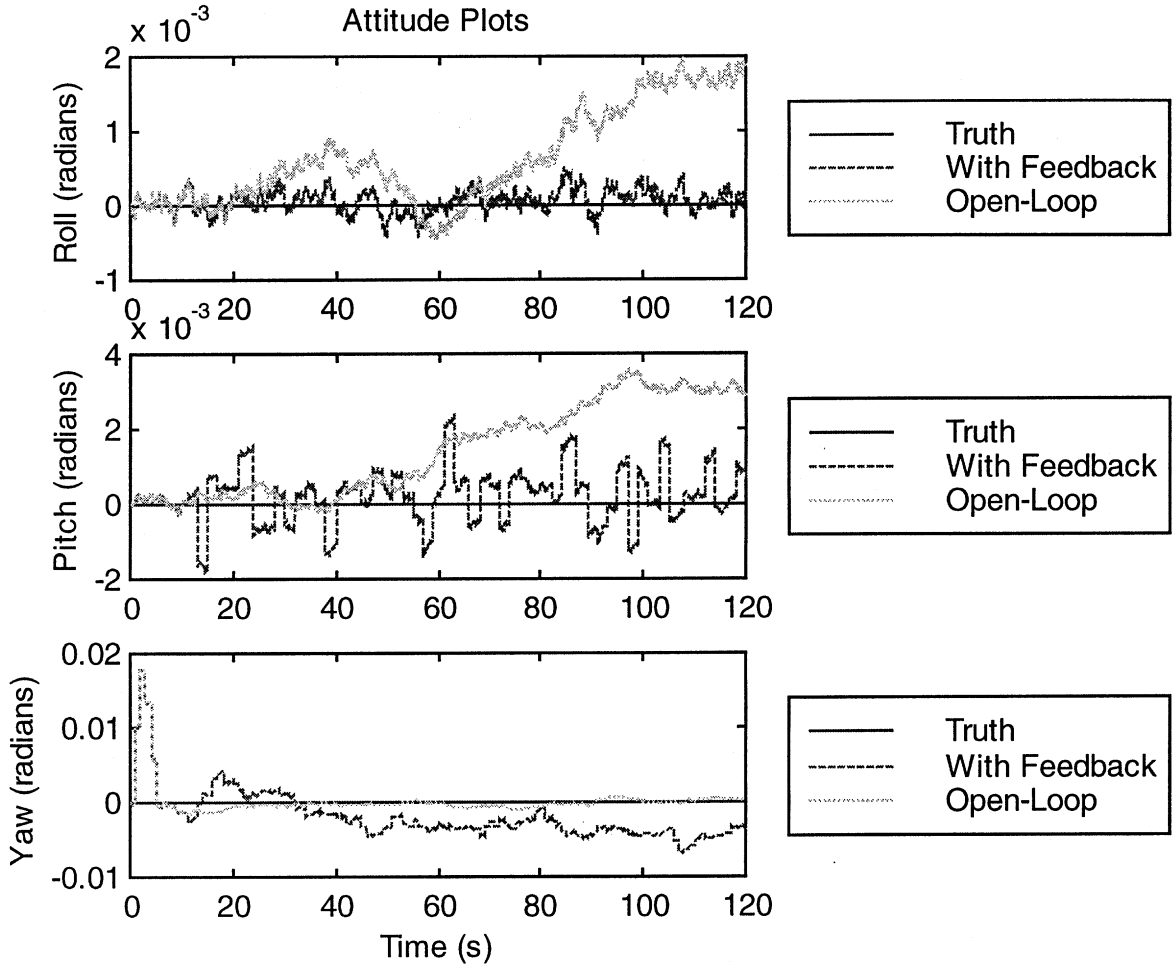

Figure 12: Attitude Plots for Simulated Moving Inputs

Again, Figure 12 shows that the Kalman filter tracks the roll and pitch accurately, but has a larger error in its estimate of yaw.

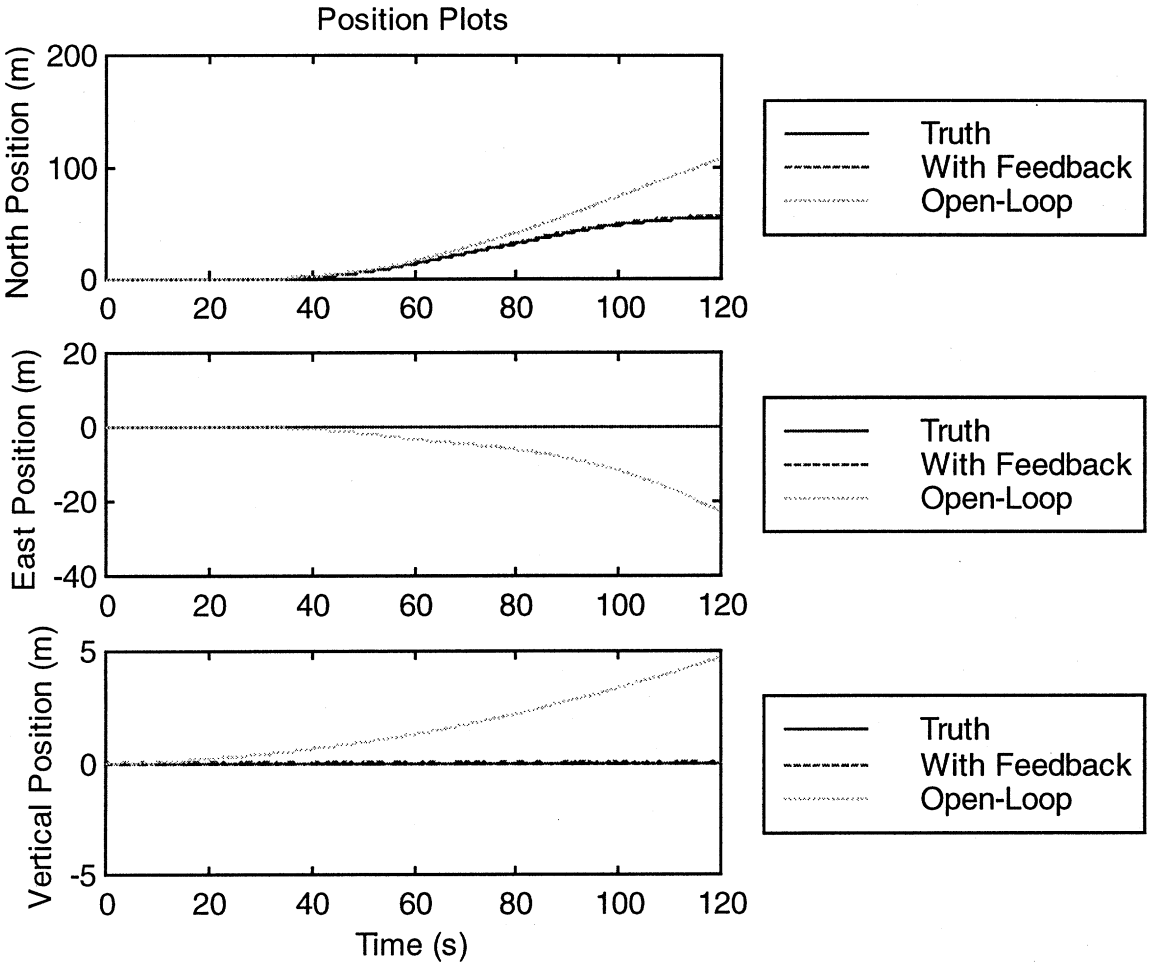

Figure 13: Position Plots for Simulated Moving Inputs

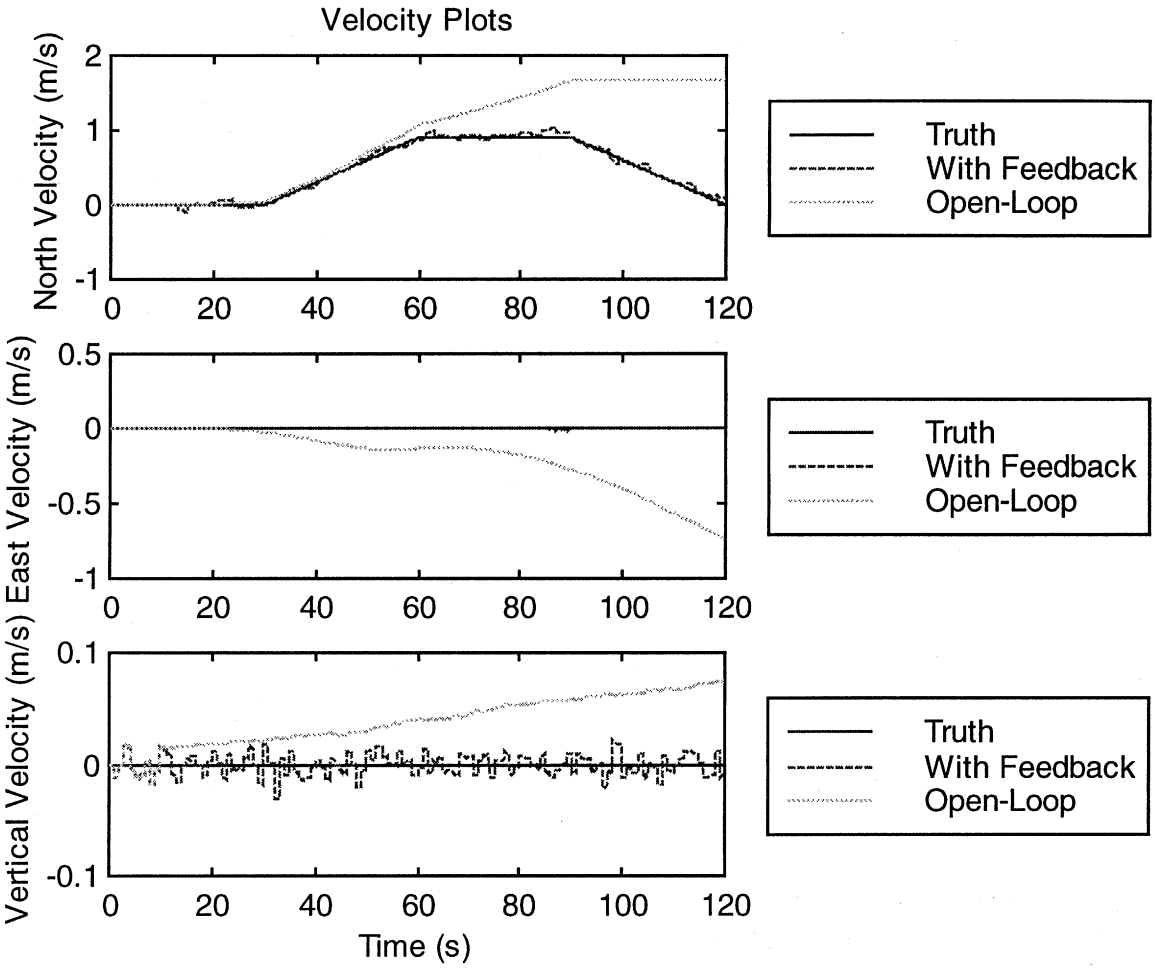

Figure 14: Velocity Plots for Simulated Moving Inputs

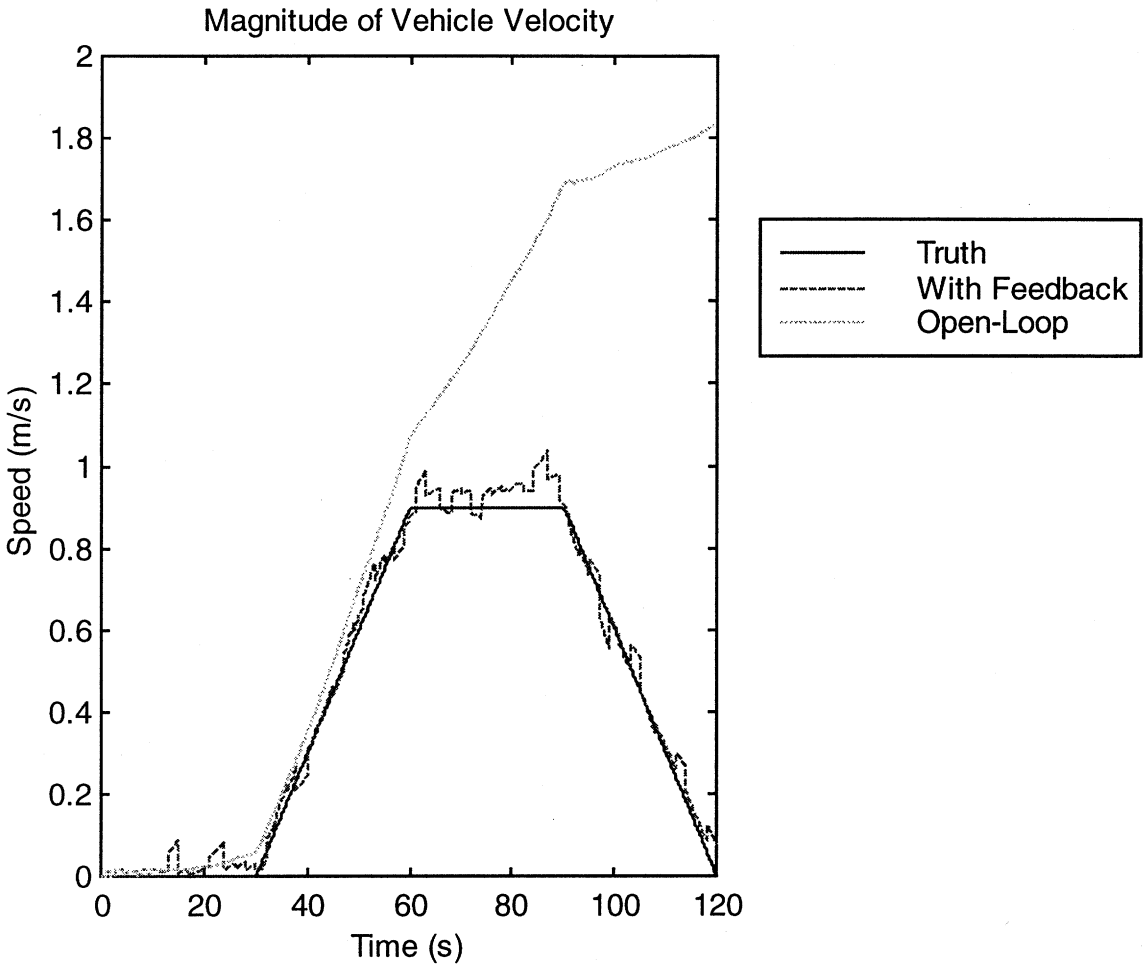

Figure 15: Speed Plot for Simulated Moving Inputs

The plots in this section reinforce the findings in Section 4.1.1 that the Kalman filter performs markedly better than the open-loop system. As shown in all the plots, the Kalman filter estimates track the "truth" very well.

4.2 Actual Inputs

After sufficient testing proved the algorithm's accuracy, the filter was run on actual data. The data was collected in the lab and on site, as described for each run.

4.2.1 At Rest

For this trial, the VCUUV simply remained motionless on a table in the lab. This experiment ensured that the algorithm performed as well on real inputs as it did on

simulated inputs. Plots of two versions of the Kalman filter are given. The first version does not use the forward velocity sensor in the feedback loop, whereas the second implementation utilizes all four "measurements." The open-loop plots had such large errors that they were not included in the figures. These errors were the result of small rotation-rate bias errors. The "truth" is not known for the attitude of the vehicle because the table is not precisely level. Because the vehicle is not moving, the "truth" for the velocity and position plots is 0. This information, however, is not included on the plots.

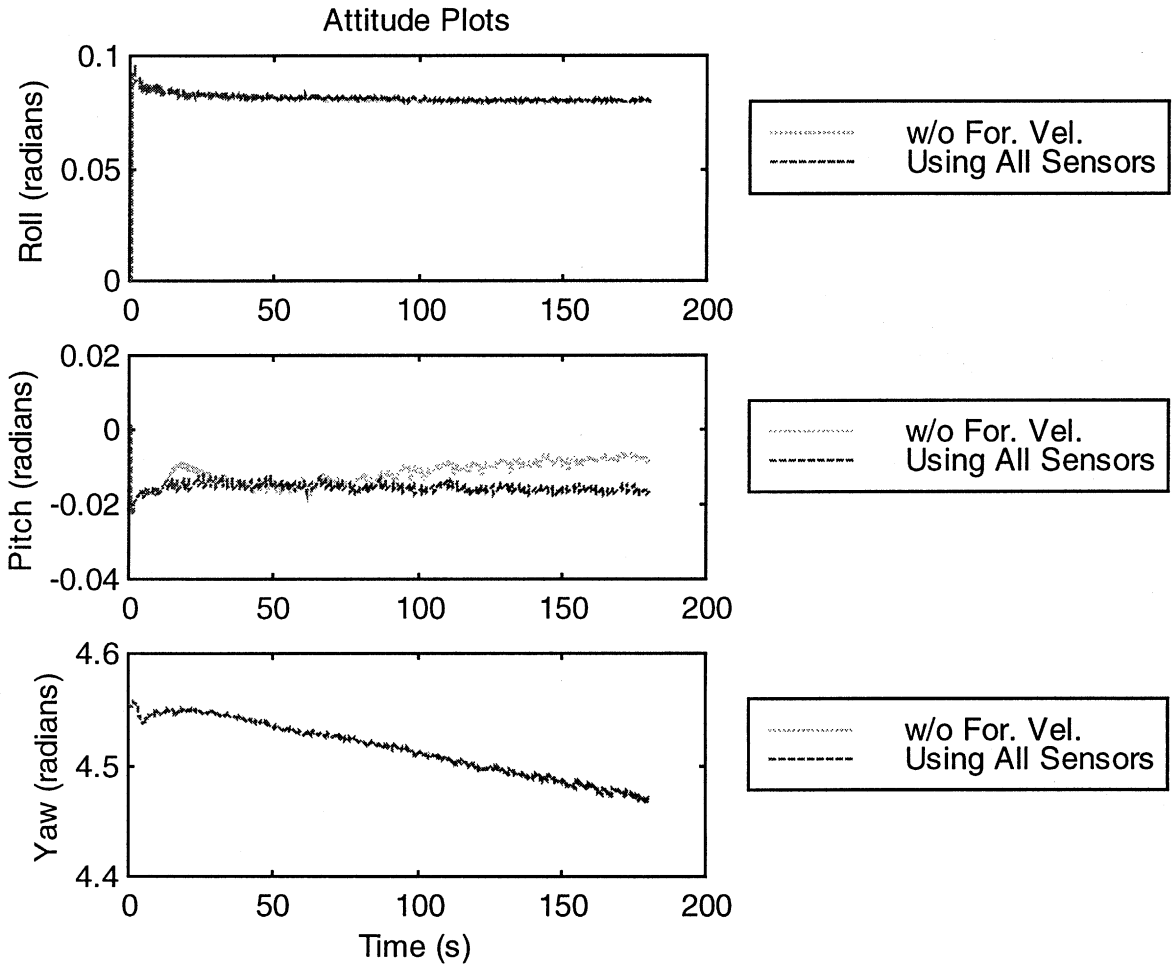

Figure 16: Attitude Plots for VCUUV At Rest

Figure 16 shows that the roll and pitch of the vehicle were not quite zero for this trial. This is reasonable because the VCUUV was resting in a cradle on a table in lab, so the IMU may not have been perfectly level.

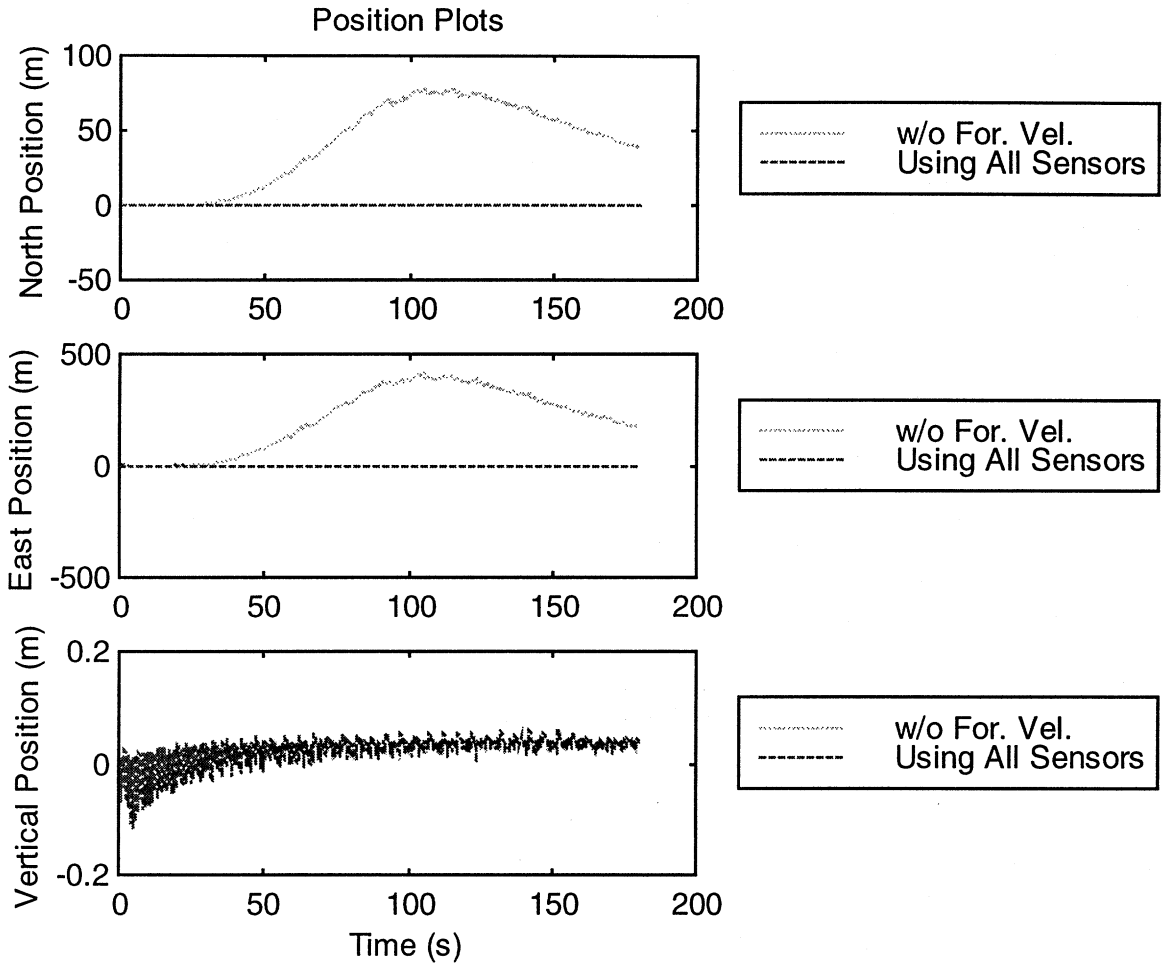

Figure 17: Position Plots for VCUUV at Rest

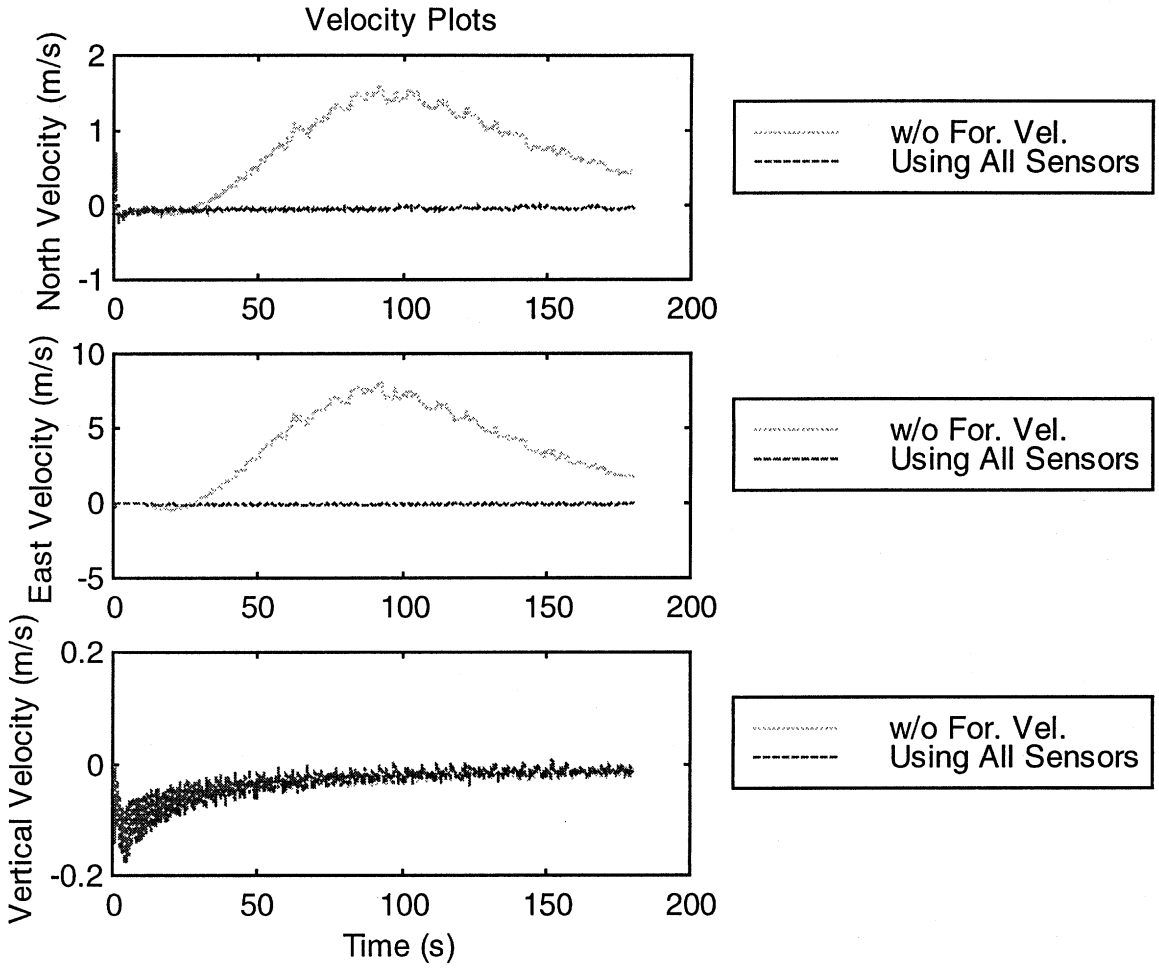

Figure 18: Velocity Plots for VCUUV at Rest

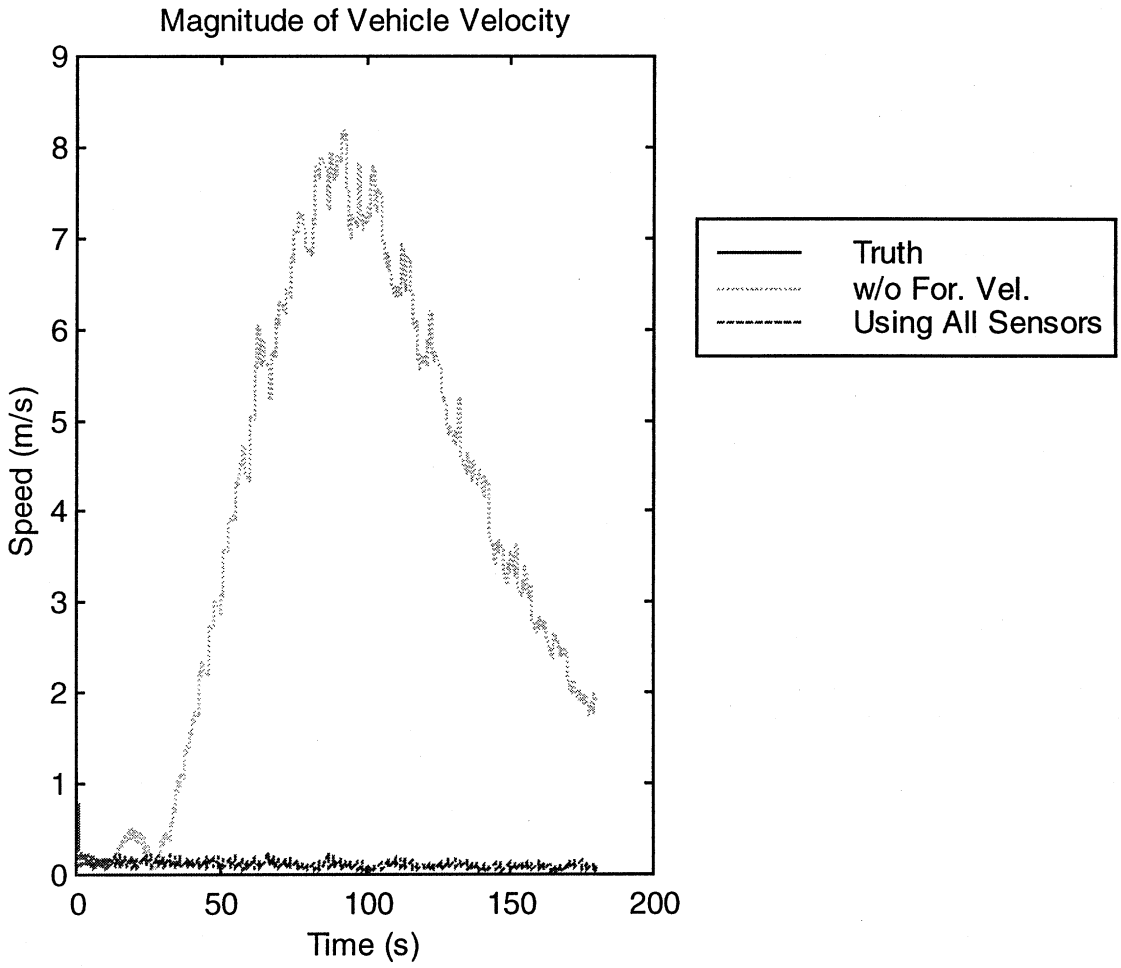

Figure 19: Speed Plot for VCUUV at Rest

While the Kalman filter that includes all four measurements performs slightly worse on the actual data than on the simulated data, it still tracks the 0 velocity of the vehicle well. The increased error is probably cause primarily by bias-estimate error. Increased bias error magnifies errors in velocity and position before the Kalman filter can correct for the errors.

The plots also show that the forward-velocity sensor is critical for limiting error in velocity and position. This finding is predictable because the forward-velocity sensor is necessary for obtaining feedback on all three dimensions of velocity.

4.2.2 Fri101

Fri101 is the data set for one of the test swims for the VCUUV. The swim was conducted in the lake at Hopkinton State Park. The trial consists of 30 seconds of free floating, 1 minute of swimming, then a period of coasting to a near stop. The following figures show the results of two versions of the Kalman filter. Again, no "truths" are shown, because the actual position of the vehicle is not known.

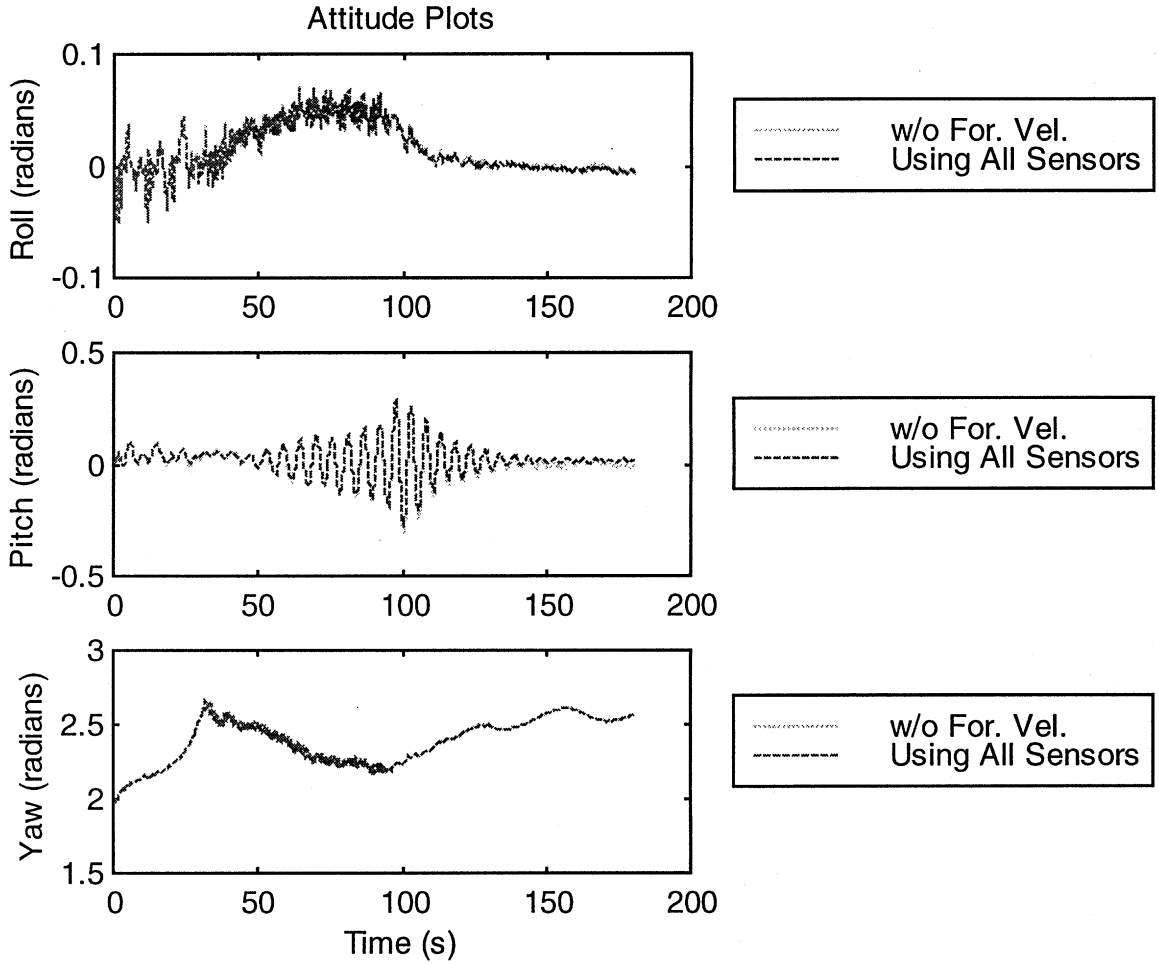

Figure 20: Attitude Plots for Fri101

The attitude plots in Figure 20 are consistent with observations made while the vehicle was swimming. The VCUUV pitches up to 20 degrees at a frequency of about 0.2 Hz.

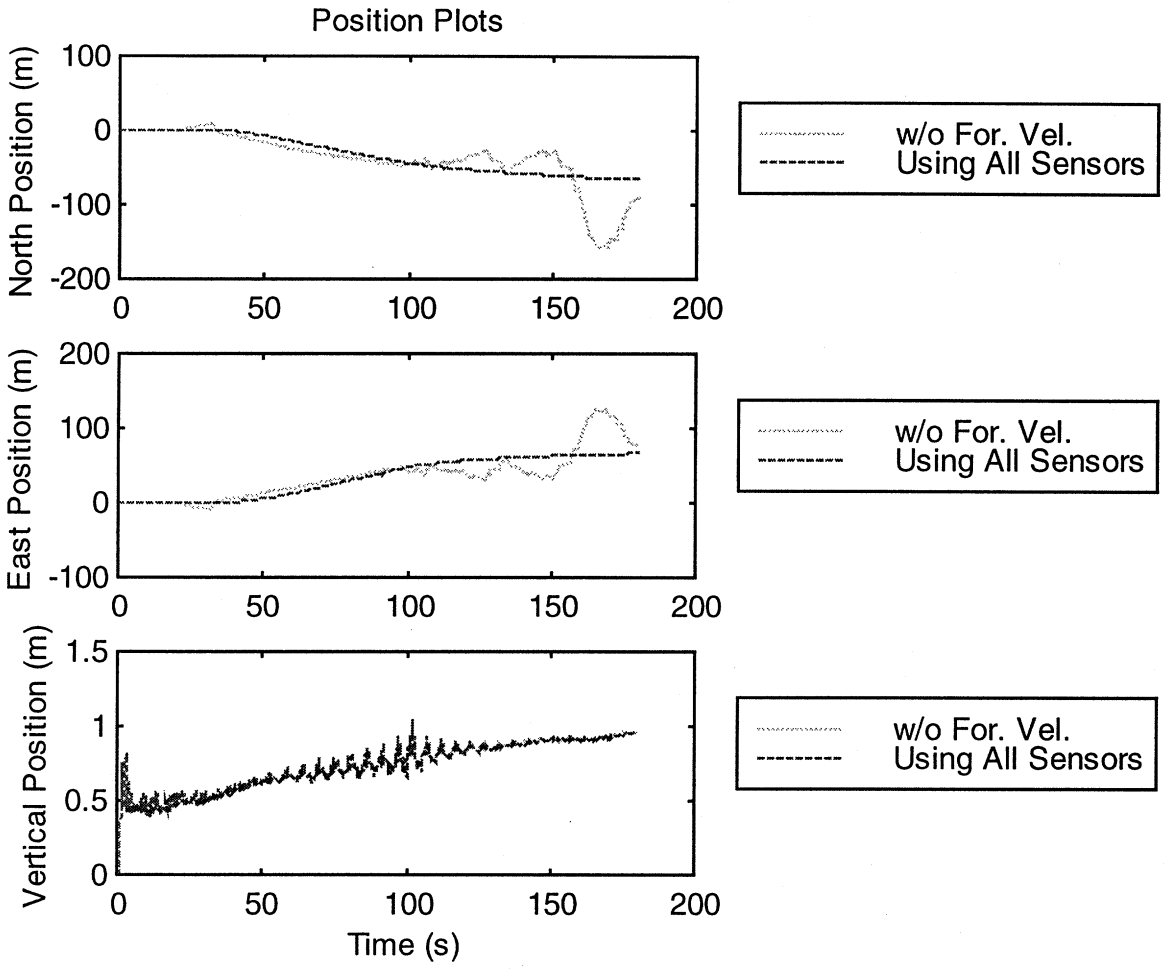

Figure 21: Position Plots for Fri101

The vertical position shown in Figure 21 shows that the VCUUV descended about 0.5 meters during the run. This agrees with observations made during the trial.

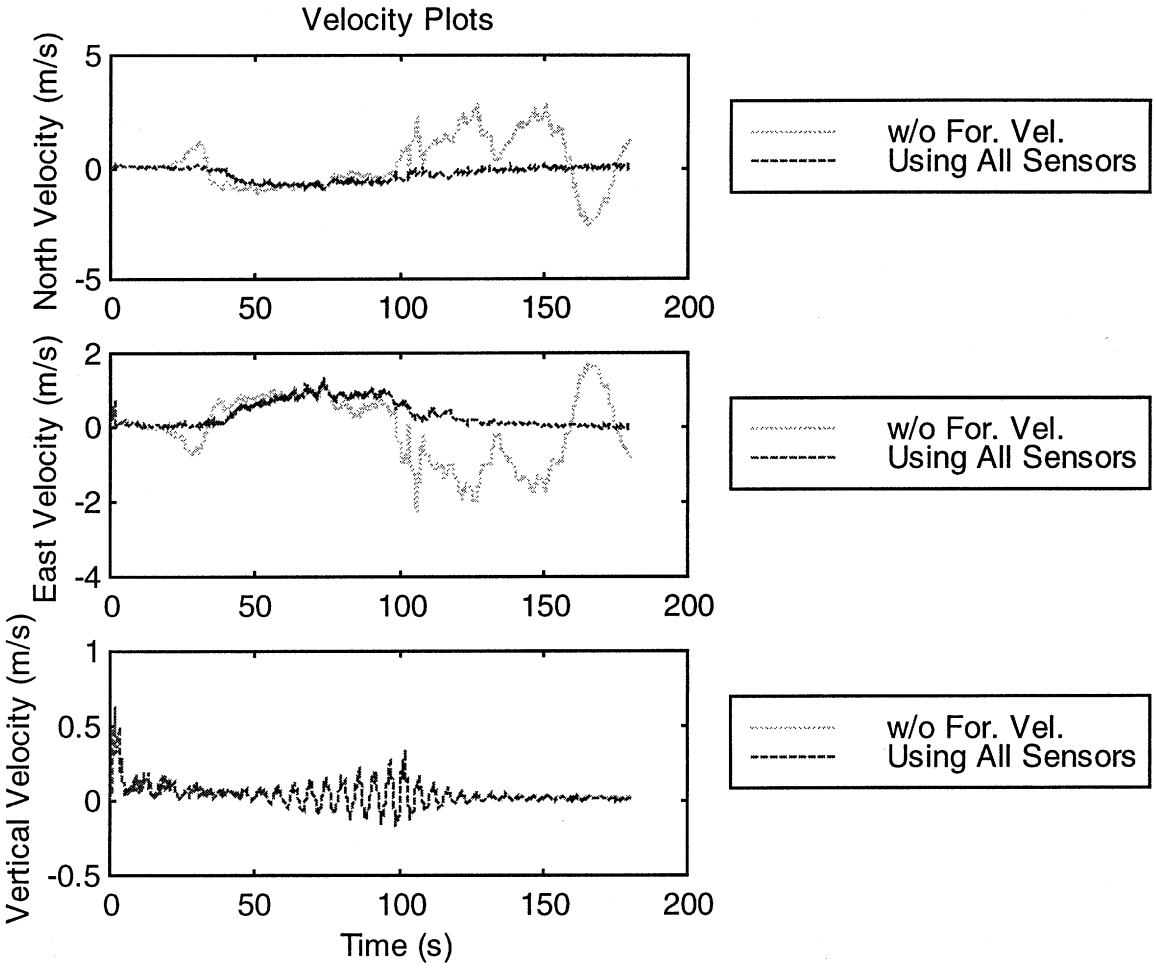

Figure 22: Velocity Plots for Fri101

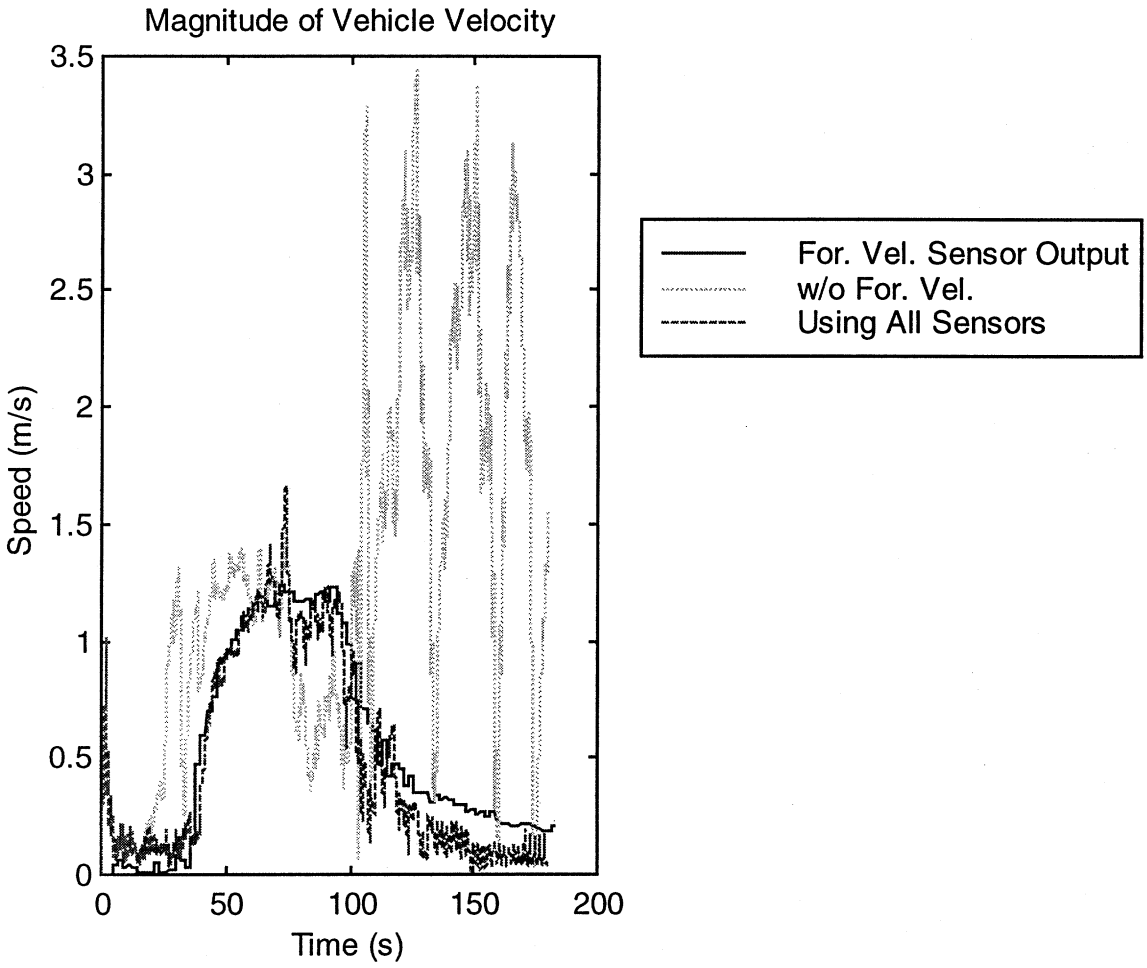

Figure 23: Speed Plot for Fri101

Figure 23 shows plots of the forward-velocity sensor output and the estimates of the speed of the VCUUV as calculated by the two versions of the Kalman filter. The estimate from the Kalman filter which includes the forward-velocity sensor agrees closely with the output of the forward velocity sensor. The plot shows that the vehicle reached speeds slightly more than 1 m/s. This estimate is consistent with estimates of the vehicle's speed obtained from swimmers in the water with the VCUUV.

Again, the forward-velocity sensor is shown to be necessary for any reasonable estimate of speed to be calculated.

4.2.3 Fri101cr3

Fri101cr3 is the data set for a run wherein the VCUUV remained at rest for 30 seconds, accelerated by swimming for 30 seconds, then locked its tail hard over, resulting in a very quick turn which reduced the vehicle's speed to almost 0 m/s. The "truth" is not known, and is therefore omitted from the following plots.

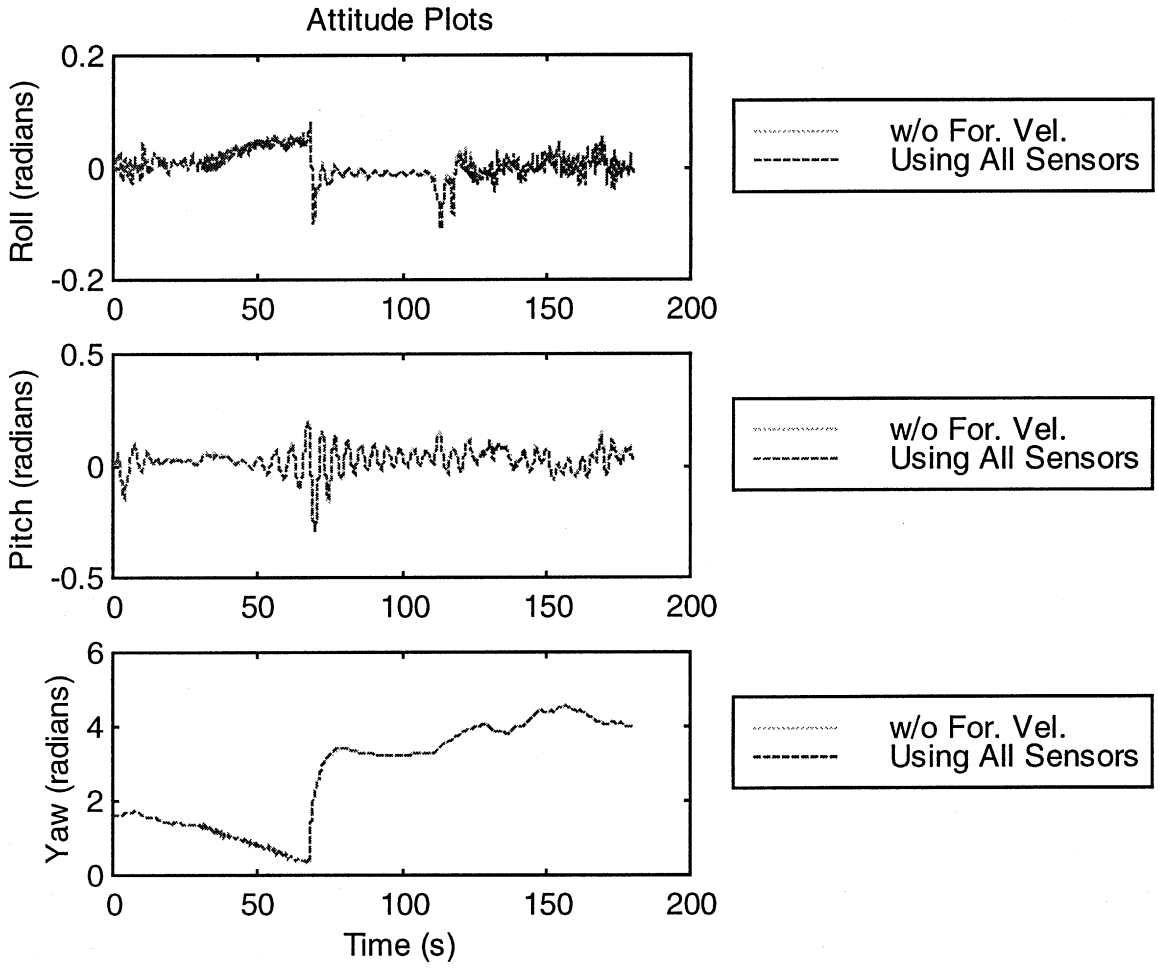

Figure 24: Attitude Plots for Fri101cr3

The plots in Figure 24 confirm what was observed at the time of the trial. The VCUUV did not pitch until after it turned. When it did turn, it rotated about 180 degrees in a very short amount of time. The vehicle was grabbed by the swimmers at about 110 seconds into the run. This explains the somewhat random looking readings in the latter parts of the attitude plots.

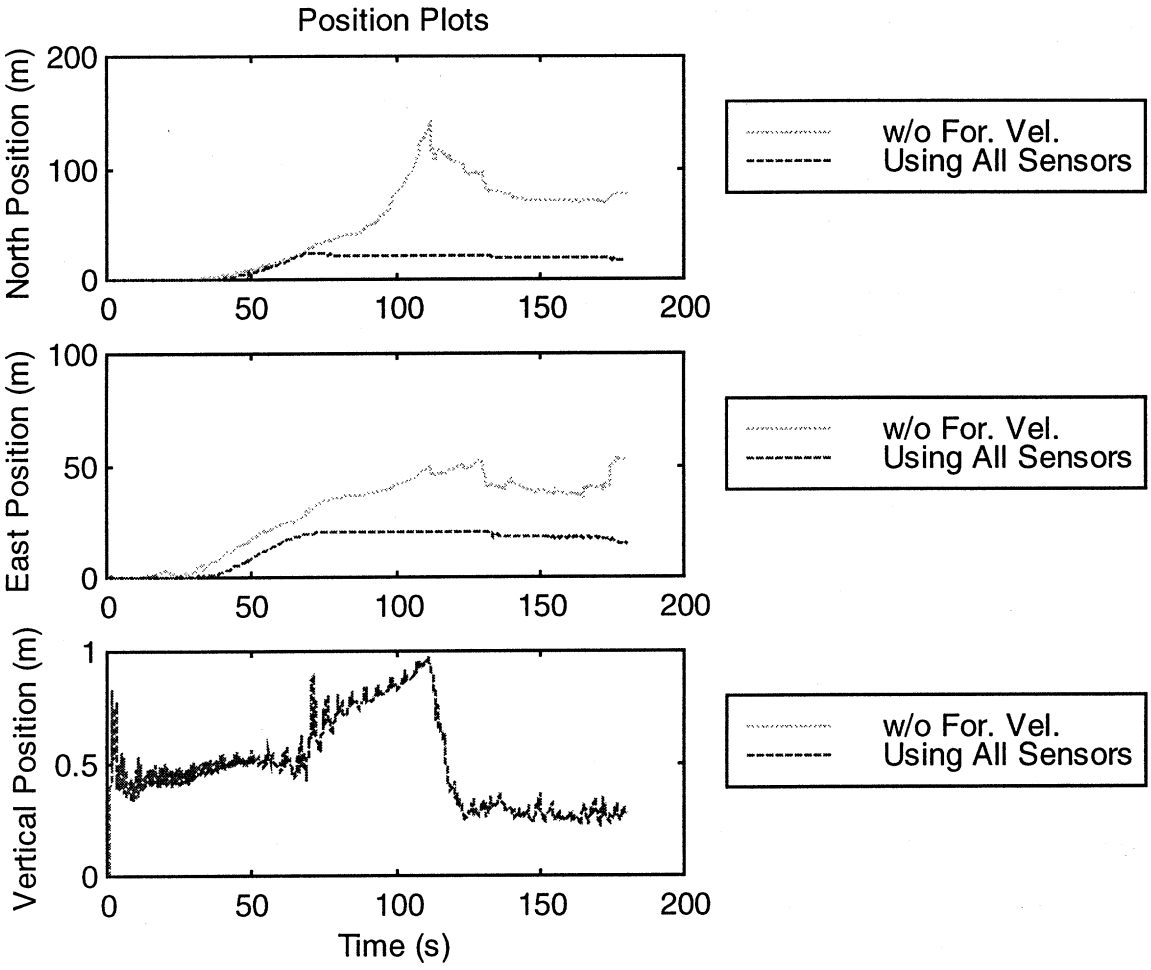

Figure 25: Position Plots for Fri101cr3

The VCUUV sank about 0.5 meters during this trial. At about 110 seconds, the vertical position shows that the swimmers grabbed the vehicle and brought it near the surface.

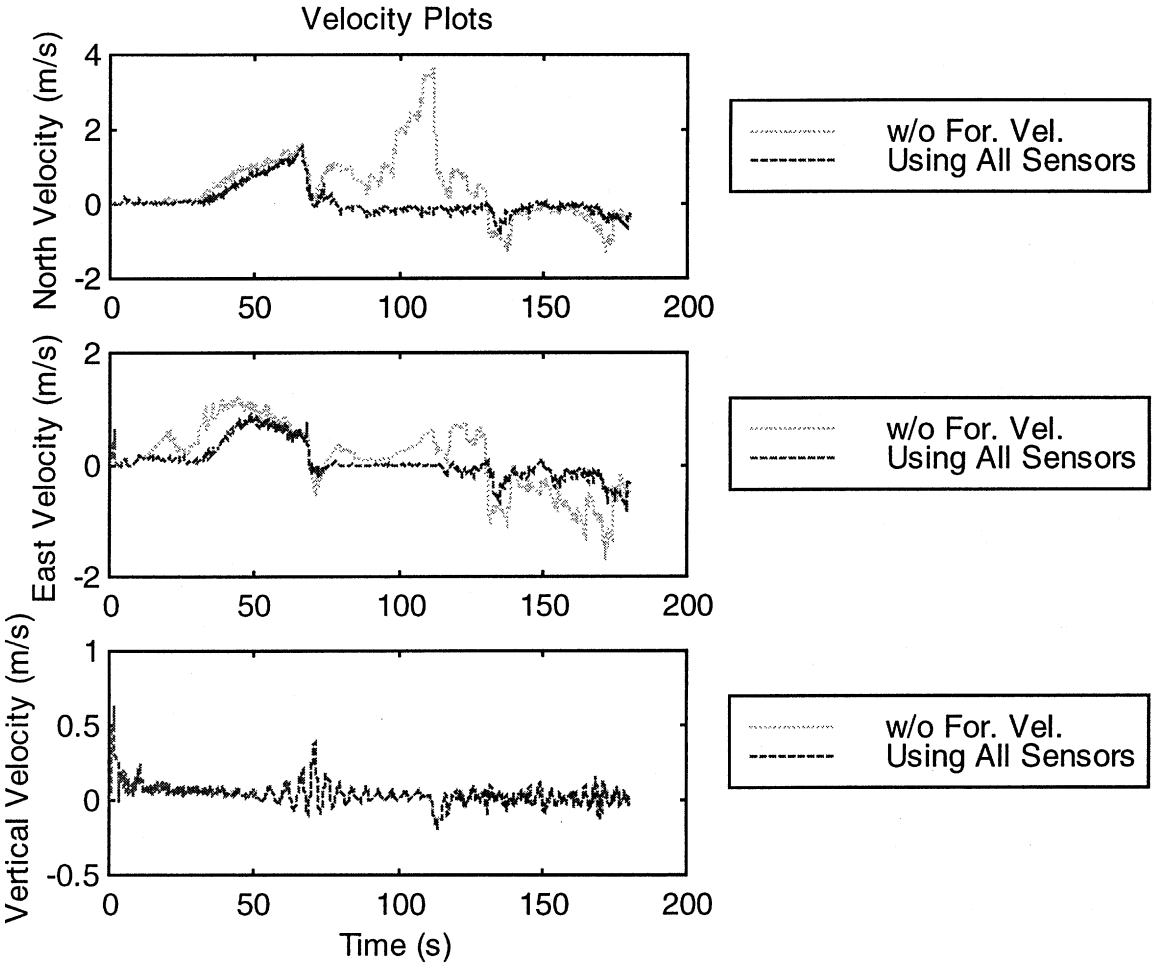

Figure 26: Velocity Plots for Fri101cr3

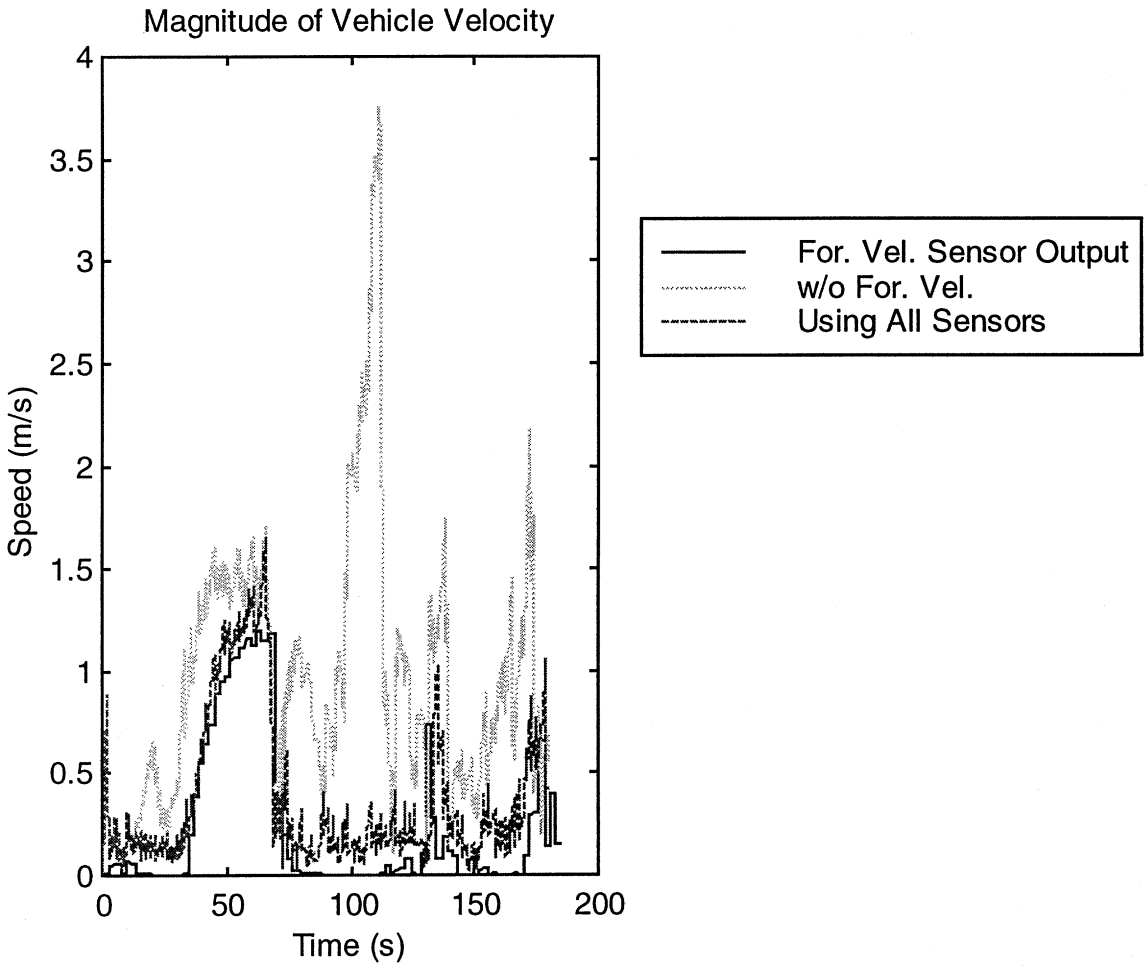

Figure 27: Speed Plot for Fri101cr3

Figure 27 is consistent with the observation that the vehicle suffered a precipitous loss of velocity during the turn, going from nearly top speed to nearly 0 m/s. The effects of the swimmers grabbing the VCUUV are also shown in the latter part of the plot.

Again, the Kalman filter without the velocity sensor does not perform well enough to be used to accurately estimate the velocity and position of the vehicle.

Chapter 5 Conclusion

5.1 Summary of Work Completed

This thesis developed a Kalman filter which can be used in conjunction with an inertial navigation system and other measures of navigation variables to estimate the attitude, position, and velocity of the VCUUV.

The Kalman filter developed for this thesis performs much better than the open-loop system that was previously in place on the VCUUV. Qualitatively, using all four measurements, velocity estimates accurate to approximately 0.5 m/s for all time can be obtained—as illustrated in Figure 23 and Figure 27. This is compared to the velocity estimate errors of the open-loop system, which are greater than 1 m/s after 30 seconds and continue to grow with time.

Without the forward-velocity sensor, however, the accuracy of the Kalman filter is seriously reduced. Velocity errors are larger initially and grow with time. While the results achieved with this reduced implementation are still an improvement over the open-loop system, the errors which result from the omission of the forward-velocity sensor make its use in estimating velocity a poor choice. The full implementation, with the forward-velocity sensor, is far preferable.

5.2 Other Applications of Work

This filter can be applied in any case where an inertial navigation system, in addition to other sensors, is used for navigation. As the Kalman filter includes some approximations, this filter is only appropriate for systems where the inertial navigation system is not depended upon for long periods of time or where great precision is not necessary. If frequent position updates, such as GPS position readings, are available, this filter can be used for autonomous vehicles with long mission lengths. An autonomous land rover or an autonomous helicopter are two examples of vehicles that could effectively use this Kalman filter in conjunction with GPS position readings.

5.3 Future Work for VCUUV

The navigation algorithm can be improved by better estimating the parameters of the IMU. By fully characterizing the temperature dependencies of the gyro biases and accelerometer biases, the errors in the open-loop estimate of the attitude can be reduced. These smaller errors will be easier for the Kalman filter to correct for, leading to better overall estimates of the velocity.

Additionally, a better way to initialize the state of the Kalman filter should be researched. In lab, when the vehicle can be held perfectly still during the initialization period, the Kalman filter can be initialized fairly accurately. However, in the field, when the VCUUV is floating during the first 30 seconds of the run, the initialization of the Kalman filter is not as accurate.

References

Barrett, David S. "Propulsilve Efficiency of a Flexible Hull Underwater Vehicle." Ph.D. Thesis. Massachusetts Institute of Technology. 1996.

Brown, Robert Grover and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Inc: New York, NY. 1992.

Gelb, Arthur (ed.). *Applied Optimal Estimation*. The M.I.T. Press: Cambridge, MA. 1989.

Grewal, Mohinder S. and Angus P. Andrews. *Kalman Filtering: Theory and Practice*. Prentice-Hall, Inc.: Englewood Cliffs, NJ. 1993.

Gurunathan, Mohan. "Guidance, Navigation and Control of a Robotic Fish." M.Eng. Thesis. Massachusetts Institute of Technology. May 1998.

Levy, Larry J. "The Kalman Filter: Navigation's Integration Workhorse." *GPS World*. September 1997, pp. 65-71.

VanBronkhorst, Alan. NATO Advisory Group for Aerospace Research and Development (ed.). "Strap-down System Algorithms." AGARD Lecture Series No. 95. *Strap-Down Inertial Systems* NATO 1978. . pp. 3-1 -- 3-22.

Welch, Greg and Gary Bishop. "An Introduction to the Kalman Filter." Department of Computer Science, University of North Carolina at Chapel Hill: Chapel Hill, NC. 1997.

Weisstein, Eric W. "Euler Angles." http://www.astro.virginia.edu/~eww6n/math/EulerAngles.html.

Strang, Gilbert. *Introduction to Linear Algebra*. Wellesley-Cambridge Press: Wellesley, Ma. 1993.

Appendix A: Calibration of Speed Sensor

See Figure 28 for a plot of the calibration runs for this sensor. Calibration was performed on a robotic gantry. Testing beyond 0.38 m/s was not possible. However, the data appears to be linear. The slope of this line is 69.5 (counts/second)/(m/s). As the PIC only reports once every two seconds, this leads to a scale factor of 139 counts/(m/s).

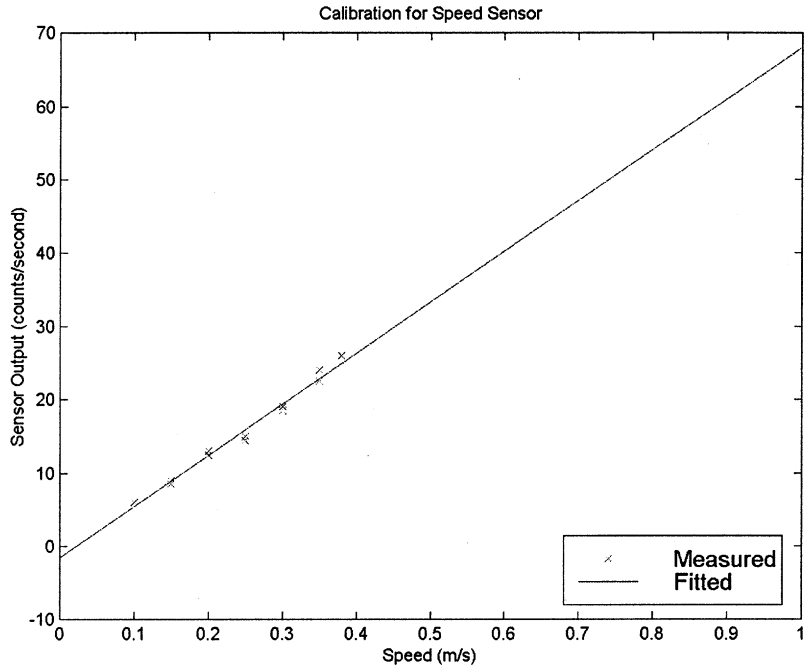

Figure 28: Speed Sensor Calibration

Appendix B: IMU Conversion Factor Calculation

By using the manufacturer-supplied scale factors, the specifications for our ADs, the analog conditioning circuitry, the conversion factors from the quanta recorded to the actual accelerations, rotation rates, and temperature were calculated. The conversion factor equations are as follows.

$$\text{Conversion(quanta} \rightarrow m/s^2) = A2D * Analog * (1/IMU) * G$$

$$\text{Conversion(quanta} \rightarrow \text{radians/s}) = A2D * Analog * (1/IMU) * (pi/180)$$

$$\text{Conversion(quanta} \rightarrow \text{Kelvin}) = A2D * Analog * (1/IMU)$$

Table 5: Definition of Conversion Factor Constants

Quanti ty	Value	Explanation
A2D	5/4096	4096 quanta spanning the 5 volts of dynamic range for the A2D
Analog	Varies from channel to channel	The scale factor of the analog amplifier
IMU	varies from channel to channel	The Systron Donner given scale factors from gravities to volts or degrees/second to volts or Kelvin to amps
G	9.8	The value used for gravity

The above conversion factors can all be found in the file Gain_calculations.m.