Intertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter

Eric Foxlin
Research Laboratory of Electronics
Massachusetts Institute of Technology

Abstract

Current virtual environment and teleoperator applications are hampered by the need for an accurate, quick-responding head-tracking system with a large working volume. Gyroscopic orientation sensors can overcome problems with jitter, latency, interference, line-of-sight obscurations, and limited range, but suffer from slow drift. Gravimetric inclinometers can detect attitude without drifting, but are slow and sensitive to transverse accelerations. This paper describes the design of a Kalman filter to integrate the data from these two types of sensors in order to achieve the excellent dynamic response of an inertial system without drift, and without the acceleration sensitivity of inclinometers.

1. Introduction

One of the key technological challenges in virtual environment, teleoperator, and augmented reality systems is head-tracking. Noise and latency in the data output by most current magnetic, acoustic, and optical head-tracking systems cause the objects in the virtual world to appear jittery and to swim about their correct stationary positions during head movements. Range limitations prohibit the use of VR for applications such as out-door operations training or building walkthroughs. Interference and distortions, particularly in magnetic systems, can cause user disorientation [1-3].

In order to overcome problems of limited range, portability, and line-of-sight restrictions, some kind of self-contained sourceless tracking system would be highly desirable. A purely inertial tracker would have the additional advantages of nearly instantaneous measurement, availability of motion derivatives for prediction, superb resolution/negligible jitter, and immunity to all forms of interference.

The operating principles for measuring orientation and position of a moving body using only gyroscopes and accelerometers have been well established in the field of Inertial Navigation Systems(INS) [4-9]. The variant called strapdown INS measures the orientation of a body by integrating the angular rates from three orthogonal rate gyros affixed to the body, starting from a known initial orientation. This orientation subsystem is referred to as an Attitude and Heading Reference System (AHRS). To get position, 3 linear accelerometers, also affixed to orthogonal axes of the moving body, measure the total acceleration vector of the body relative to inertial space. This acceleration vector can be converted from body coordinates to earth coordinates using the known instantaneous orientation of the body determined by the AHRS. Position is then obtained by subtracting off the effect of gravity from the measured acceleration and then performing double integration starting from a known initial position.

Drift in the determination of orientation by the AHRS results from gyro biases, which lead to a linear drift rate after single integration. If the startup bias can be measured and nulled, the worst case drift rate is determined by the bias stability, which ranges from about 1°/second for inexpensive silicon micromachined gyros to 0.001°/hour for sophisticated inertial navigation gyros. The best gyros of a practical size for head-tracking have a bias stability on the order of Earth's rotation rate of 15°/hour. Much less expensive and smaller are miniature vibrating element gyros with bias stabilities of several degrees/minute and worse. Drift in the measurement of linear displacement is a far more difficult problem due to the double integration of acceleration, and is not addressed in this paper.

An inertial head-tracker has been developed by the author at MIT, concentrating first on the more tractable problem of 3-DOF orientation tracking [10]. The first prototype consisted of three orthogonal angular rate sensors together with a two-axis fluid inclinometer for drift compensation. The outputs of the angular rate sensors were integrated to obtain orientation, and the orientation was occasionally reset by the fluid inclinome-

0-8186-7295-1/96 $5.00 © 1996 IEEE
Proceedings of VRAIS '96

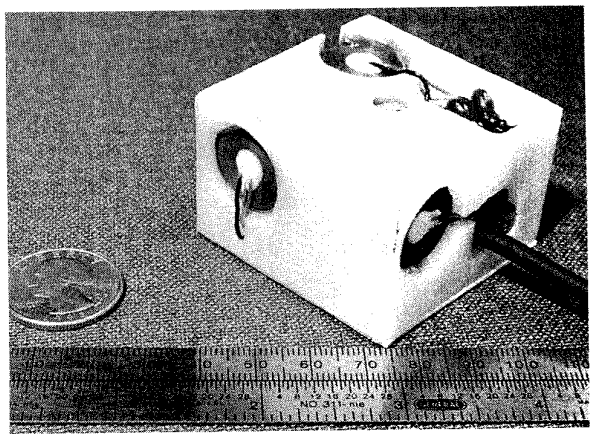

Figure 1: MIT inertial tracker 2nd prototype

ter to correct for the slow drift of the gyros. Due to the relatively high performance of the rate transducers used in that prototype, even this simple sensor fusion algorithm was able to achieve orientation tracking performance of <1 ms latency, 0.008° r.m.s. noise, and 0.5° absolute static and dynamic accuracy in pitch and roll [11]. At 1 lb., the prototype was still a little large for practical head-tracking applications.

A second prototype of the MIT Inertial Tracker, shown in Figure 1, has now been built which incorporates tiny low-cost solid-state rate gyros, a two-axis fluid inclinometer, and a two-axis fluxgate compass. It weighs only 3.5 ounces, can be comfortably worn on a head-mounted display, and uses low-cost sensors so that it can be developed into a competitive commercial head-tracking product. However, the miniature low-cost rate gyros have so much higher hysteresis, nonlinearity and bias instability that the simple ad hoc drift correction algorithm used in the previous prototype does not lead to sufficiently accurate results. This paper concerns the design of a more sophisticated sensor data fusion scheme, based on Kalman filtering, which makes the best use of all the data available from both types of sensors and thereby achieves a lower mean squared orientation estimation error than the ad hoc method. To be useful, the filter must be able to run in real time on an inexpensive 486-class microprocessor, so considerable effort is invested in formulating a minimum-order Kalman filter and implementing it efficiently.

The main contributions of this paper are 1) an analysis of the literature about related Kalman filter applications, 2) an exposition of the modeling decisions that were made to formulate the filter, which will help others to frame the questions necessary to apply Kalman filtering to similar problems, 3) an example of the use of Friedland's separate-bias Kalman filter formulation, which has not been previously applied in synthetic environment tracking work, and 4) a very effective adaptive algorithm for adjusting the Kalman filter parameters to the instantaneous motion characteristics. This paper focuses more on filter design and implementation than validation, and no effort is made to formulate an optimal filter and compare the performance of the reduced-order filter to the optimal filter in simulation.

2. Kalman filtering

Consider a dynamic system which can be modeled by a n-by-1 state vector $\mathbf{x}$ obeying a discrete-time (DT) evolution equation

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k \quad (1)$$

where $\mathbf{A}$ is an n-by-n state transition matrix, $\mathbf{B}$ is an n-by-p matrix and $\mathbf{u}$ is a p-by-1 vector of known system inputs, and $\mathbf{w}$ is an n-by-1 process noise vector with covariance matrix $\mathbf{Q}_k$. (Note that lower-case bold letters, Greek or Roman, denote vectors, and upper-case bold letters denote matrices.) Suppose there are indirect measurements of the state vector available at each time k, and that they can be expressed as an m-by-1 measurement vector

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \quad (2)$$

where $\mathbf{C}$ is an m-by-n system observation matrix, and $\mathbf{v}$ is an m-by-1 measurement noise vector with covariance $\mathbf{R}_k$. A Kalman filter is a recursive algorithm for computing an estimate $\hat{\mathbf{x}}$ of state which is optimal in the sense of least square error under certain circumstances. One form of the DT Kalman filter, used in Section 4.5, is

$$\hat{\mathbf{x}}_{k+1} = \mathbf{A}\hat{\mathbf{x}}_k + \mathbf{B}\mathbf{u}_k + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{C}\mathbf{A}\hat{\mathbf{x}}_k) \quad (3)$$

where the Kalman gain matrix $\mathbf{K}$ is computed from the estimation error covariance matrix, $\mathbf{P}$, according to

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{C}^T\left[\mathbf{C}\mathbf{P}_k\mathbf{C}^T + \mathbf{R}_k\right]^{-1} \quad (4)$$

and $\mathbf{P}$ is updated according to the Ricatti equation:

$$\mathbf{P}_{k+1} = \mathbf{A}[\mathbf{I} - \mathbf{K}_k\mathbf{C}]\mathbf{P}_k\mathbf{A}^T + \mathbf{Q}_{k+1}. \quad (5)$$

The Kalman filter is very useful for combining data from several different indirect and noisy measurements to try to estimate variables which are not directly measurable. Thus, while the gyroscopes measure orientation by integrating angular rates, and the inclinometer and compass provide a different noisy and sloshy but drift-free measurement of orientation, the Kalman filter weights the two sources of information appropriately to make the best use of all the data from each. If the model in (1) and (2) is a simplification of the actual physical system, the resulting reduced-order Kalman filter (ROKF) will not be optimal, but will often perform almost as well as the full-order Kalman filter. This property is exploited in this paper without any attempt to evaluate the performance of the ROKF. If the system dynamics are nonlinear, it is possible to linearize about a nominal or actual trajectory and run a Kalman filter on the linearized system. This is

the basis of the extended Kalman filter (EKF) and the complimentary Kalman filter developed in Section 4.2. A discussion of Kalman filtering can be found in [12].

3. Literature analysis

In applying Kalman filtering to the inertial orientation tracking problem there is considerable freedom in system modeling - what physical variables to assign to the state vector **x**, what measurements are in the measurement vector **y**, and what matrices **A**, **B**, **C**, **Q**, and **R** most accurately describe the system given those choices. A literature search was conducted to see how other authors have used Kalman filters to estimate orientation from the outputs of 3 strapdown gyros. The 7 most relevant references found are reviewed in this section. Two come from vehicle navigation, two from robotics, and three from virtual environments.

An early maritime navigation work by Bona and Smay [13], summarized in [12], is of interest because it shows how to reset gyro biases based on indirect measurements (position errors that result from them) and provides a now-common Markov model of gyro bias evolution. The dynamic system model details how the position errors evolve in response to the gyro biases, and how the gyro bias Markov components evolve in response to the process noise.

The most relevant reference found in the aeronautics literature was Koifman and Merhav's description of an autonomously aided strapdown attitude reference system [14]. Here, an autopilot is created with three low-cost rate gyros with time-varying biases on the order of 0.1°/s. The measurements fed into the Kalman filter are from the three gyros, a magnetic compass, altimeter, and airspeed sensor. The state vector contains 16 elements: 3 linear velocities, 3 angular velocities, 3 orientation Euler angles, altitude, 3 wind gust velocity components and 3 gyro biases. The state transition matrix is obtained by linearizing the system differential equations which encompass the aircraft equations of motion as well as the kinematic Euler equations (6). In contrast to Bona and Smay, the gyro biases are considered piecewise constant, and the corresponding diagonal covariance elements are simply reset whenever a change detection algorithm suspects that the gyro biases may have changed. It is also instructive to note that the full order 16-dimensional system could not be run in real time, so they reduced the state to 11 elements and were then able to achieve about 20 updates per second with minimal loss in accuracy. The measurement vector consists of the three gyros and the airspeed sensor.

Barshan and Durrant-Whyte [15] investigated the use of a solid-state gyroscope for mobile robotics applications. They paid particular attention to the gyroscope error model, and came up with an exponential curve to fit the changes in bias as the gyroscope warms up. They then implemented a Kalman filter for estimating a single rotation angle Φ, with a state vector $\left[\Phi \;\; \dot{\Phi} \;\; \ddot{\Phi} \;\; \dddot{\Phi} \;\; \varepsilon_{\Phi} \;\; \varepsilon_{\dot{\Phi}}\right]^T$ and a state transition matrix that propagates the truth states $\Phi, \dot{\Phi}, \ddot{\Phi}, \dddot{\Phi}$ and error states $\varepsilon_{\Phi}, \varepsilon_{\dot{\Phi}}$ completely independently. The only system observation is the single rate gyro measurement, so the system is not observable, and the angular position error covariance grows unbounded. However, it is demonstrated that the gyro drift error grows at a rate 5 times slower when using the exponential gyro error model.

A paper on mobile robot attitude estimation by Vaganay et al [16] provides the only example in the literature in which gyroscope drift is compensated using two accelerometers, and is therefore particularly germane to this drift-free head-tracking application. The Kalman filter model is very unusual and results in a state vector of surprisingly low dimension. The integration of angular rates is done outside of the Kalman filter, and is treated as part of a measurement system that provides gyroscopically determined measurements of pitch and roll, θ_g and ψ_g, which are complimented by gravimetric measurements of θ and ψ from the accelerometers. The state contains θ and ψ and the pitch and roll drift rates, and the transition matrix used in the Kalman filter is just the identity. This is the leanest Kalman filter conceivable, as even the kinematics of Euler angle integration are not modeled, but the performance reported is nearly comparable to the other methods. No details are given about the determination of **Q** and **R**.

Azuma and Bishop developed a Kalman filter to use inertial sensors together with an optical head-tracker to predict head motion in HMD applications [17]. The approach is different from the preceding papers, and also from the application developed in this paper, because the gyroscope rate signals are not integrated to obtain orientation. Instead, the orientation is obtained from the optical head-tracker, and the angular rates are fused with this in the Kalman filter to yield improved predictions. The state vector contains a quaternion specifying orientation, the angular rates in body axes, and the angular accelerations in body axes. The measurement consists of the quaternion measured by the optical tracker, and the angular rates measured by the gyros. The **Q** and **R** matrices are determined off-line using Powell's method on prerecorded datasets to find the parameters that give the best performance. Prediction was accomplished by extrapolating forward in time, using the angular velocity and acceleration estimates in the state vector.

Emura and Tachi likewise used gyros to augment the dynamic performance of an existing head-tracker, but in this case the tracker was magnetic instead of optical [18, 19]. The state vector contains orientation (Euler angles in the first paper were replaced with a quaternion in the second) and angular velocities. The measurement vector measures all elements of the state, using a Polhemus magnetic tracker to measure orientation and gyros to measure the angular rates. A novel aspect of the Kalman filter structure is the use of two different types of measurement update step: a 3-dimensional measurement used most of the time, when only gyro data is available, and a 6-dimensional measurement used when the Polhemus data is available as well. **Q** and **R** were found empirically, using a high-precision mechanical tracker as a reference to measure remnant errors.

4. System modeling and filter design

4.1 State and measurement vectors

The first step in modeling is to decide what to put in the state and measurement vectors. Since the basic purpose of the Kalman filter is to estimate orientation, it is a given that it will be included in the state vector. Indeed, all the authors except [13] include it, although they are split between quaternion and Euler angle representations. In the interest of smaller state dimension (i.e. faster computation), this implementation uses Euler angles. The aeronautics convention is used, where ϕ, θ, and ψ, called yaw, pitch and roll respectively, represent positive rotations about the z, y, and x body axes in turn, with the positive x-axis pointing forward, positive y pointing right, and positive z pointing down. There is a singularity in the Euler angle representation at $\theta = \pm 90°$, but this was not found to produce any noticeable disturbances in practice.

All the remaining references except [16] also include angular rates in the state vector and gyroscopic angular rate measurements in the measurement vector. This is very natural, as it allows the Euler angle integration kinematics [20],

$$\dot{\boldsymbol{\theta}}(t) = \mathbf{W}_B(\boldsymbol{\theta}(t))\,\boldsymbol{\omega}(t)$$

$$\boldsymbol{\theta}(t) \equiv \begin{bmatrix} \psi(t) \\ \theta(t) \\ \phi(t) \end{bmatrix}, \qquad \boldsymbol{\omega}(t) \equiv \begin{bmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{bmatrix} \tag{6}$$

$$\mathbf{W}_B(\boldsymbol{\theta}(t)) \equiv \begin{bmatrix} 1 & \sin\psi(t)\tan\theta(t) & \cos\psi(t)\tan\theta(t) \\ 0 & \cos\psi(t) & -\sin\psi(t) \\ 0 & \sin\psi(t)/\cos\theta(t) & \cos\psi(t)/\cos\theta(t) \end{bmatrix}$$

to be incorporated into the system dynamics model, and allows the gyro measurements to be utilized in the obvious way - as measurements. However, while it is obvious from (6) how the derivatives of the orientation state elements will be computed from the state, how shall the derivatives of the angular velocity components depend on state? Some authors [18, 19] simply assume zero dependence, i.e. constant angular rates. Some process noise is added to the angular accelerations to allow for non-constant angular rates, but in reality the angular accelerations would not be very much like white noises, so this model cannot be very optimal. Other authors [15, 17] augment the state vector with $\dot{\boldsymbol{\omega}}$, which changes the model to an assumption of constant angular acceleration. The difference between the true $\ddot{\boldsymbol{\omega}}$ and the assumed $\ddot{\boldsymbol{\omega}} = \mathbf{0}$ is closer to white noise. Further derivatives, as in [15], make the model even more accurate, but lead to an unreasonably large state vector.

For most accurate estimation, the equations of motion of the body being tracked should be included in the system dynamics model (1). For example, in [14] the angular accelerations of the aircraft depend precisely, through well-known aircraft equations of motion, on quantities in the state vector and aileron positions, which are known inputs. Unfortunately, head accelerations are driven by muscle forces - an unknown input - so head dynamics are not modeled in the current system.

In inertial navigation applications, such as [13-15], gyro bias terms are usually included in the state vector. This is very important where the only aiding comes from sparse or indirect sources such as occasional position fixes. In this case, it is desired to milk as much accuracy as possible out of the gyro integration algorithm, and time-varying gyro biases are the largest source of error. Our state vector is therefore augmented with the three gyro bias terms $\delta\omega_x$, $\delta\omega_y$, and $\delta\omega_z$.

4.2 Complimentary Kalman filter

While most of the references above used a Kalman filter to directly estimate the state variables of orientation and it's derivatives, it is common in inertial navigation systems to instead use a complimentary Kalman filter which operates only on the errors in those primary state variables [12].

The direct Kalman filter block diagram in Figure 2 has $\boldsymbol{\omega}$ measured by the gyros and $\boldsymbol{\theta}$ measured by the aiding sensors all as measurement inputs. The Euler angle integration of (6) is then accomplished as part of the prediction step inside the Kalman filter block. The complimentary Kalman filter is shown in Figure 3. Here, the integration of the Euler angles is performed outside of the Kalman filter, in the block labeled "attitude computation". One advantage of this structure is that it guaran-

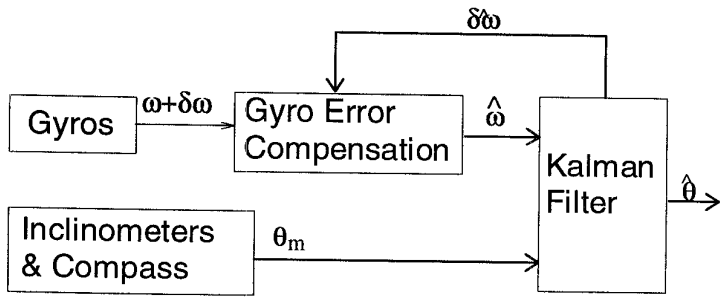

Figure 2: Direct Kalman filter for orientation

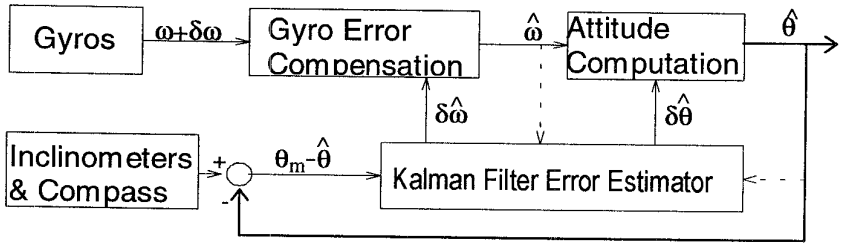

Figure 3: Complimentary Kalman filter for orientation

tees that the rapid dynamic response of the inertial system will not be compromised by the Kalman filter. Another advantage is that the gyro rates are not treated as measurements, so it is unnecessary to include $\boldsymbol{\omega}$ in the state vector. Since the head dynamics are not being modeled in this implementation, $\boldsymbol{\omega}$ is excess baggage, and by removing it from $\mathbf{x}$ the dimension is reduced from 9 to 6, with more than a three-fold computational savings. The following sections, therefore, will strive to develop a complimentary Kalman filter to estimate

$$\delta\mathbf{x} = \begin{bmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{\omega} \end{bmatrix} \equiv \begin{bmatrix} \delta\psi & \delta\theta & \delta\phi & \delta\omega_x & \delta\omega_y & \delta\omega_z \end{bmatrix}^T \quad (7)$$

using

$$\delta\mathbf{y} = \begin{bmatrix} \psi_{inclinometer} - \hat{\psi} & \theta_{inclinometer} - \hat{\theta} & \phi_{compass} - \hat{\phi} \end{bmatrix}^T \quad (8)$$

as the measurements, where $\delta\boldsymbol{\theta}$ represents the error in the output of the attitude computer, and $\delta\boldsymbol{\omega}$ represents the gyro biases.

4.3 DT nonlinear attitude computation

A Kalman filter which operates on the errors of the INS attitude computer must mimic the noise-free error dynamics of the attitude computation. This section derives the attitude integration algorithm, Section 4.4 linearizes the attitude algorithm to obtain the error dynamics, and Section 4.5 describes a complimentary EKF to operate on the errors of the attitude computation with the computational complexity of the EKF reduced by applying Friedland's separate bias formulation.

The continuous-time (CT) nonlinear differential equation which the attitude computer must integrate was given in (6). To derive the DT attitude computation from it, it is useful to approximate the evolution of $\boldsymbol{\theta}(t)$ over a short time interval by its Taylor series expansion

$$\boldsymbol{\theta}(t+\Delta t) = \boldsymbol{\theta}(t) + \dot{\boldsymbol{\theta}}(t)\,\Delta t + \ddot{\boldsymbol{\theta}}(t)\frac{\Delta t^2}{2} + \ldots \quad (9)$$

The number of terms which must be retained depends on the size of Δt. For a first order integration algorithm (retaining only the first two terms), the error per step will be mostly due to the third term, which is of order $\omega^2\Delta t^2/2$. Therefore,

$$\text{error rate} \approx \frac{1}{2}\omega^2\Delta t.$$

For typical peak head velocities of about 6 radians/sec and a timestep of 0.003 sec, this yields an error rate of about .05 rad/s (about 3°/s) which is unacceptable. Retaining the third term, the error rate will be dominated by the fourth term, or order $\omega^3\Delta t^3/6$, so

$$\text{error rate} \approx \frac{1}{6}\omega^3\Delta t^2.$$

For the same ω and Δt the error rate would be about 0.0003 rad/s, or about 1°/min. Since the low-cost gyros are unlikely to have performance much better than this, a second order integration algorithm was selected.

Differentiating (6) by the chain rule for partial derivatives results in

$$\ddot{\boldsymbol{\theta}}(t) = \frac{\partial}{\partial\boldsymbol{\theta}}\left[\mathbf{W}_B(\boldsymbol{\theta}(t))\;\boldsymbol{\omega}(t)\right]\dot{\boldsymbol{\theta}}(t) + \frac{\partial}{\partial\boldsymbol{\omega}}\left[\mathbf{W}_B(\boldsymbol{\theta}(t))\;\boldsymbol{\omega}(t)\right]\dot{\boldsymbol{\omega}}(t) \quad (10)$$

Defining (with time indices suppressed for brevity)

$$\mathbf{V}_B(\boldsymbol{\theta},\boldsymbol{\omega}) \equiv \frac{\partial}{\partial\boldsymbol{\theta}}\left[\mathbf{W}_B(\boldsymbol{\theta})\boldsymbol{\omega}\right] =$$

$$\begin{bmatrix} \frac{\cos\psi\sin\theta}{\cos\theta}\omega_y - \frac{\sin\psi\sin\theta}{\cos\theta}\omega_z & \frac{\sin\psi}{\cos^2\theta}\omega_y + \frac{\cos\psi}{\cos^2\theta}\omega_z & 0 \\ -\sin\psi\,\omega_y - \cos\psi\,\omega_z & 0 & 0 \\ \frac{\cos\psi}{\cos\theta}\omega_y - \frac{\sin\psi}{\cos\theta}\omega_z & \frac{\sin\psi\sin\theta}{\cos^2\theta}\omega_y + \frac{\cos\psi\sin\theta}{\cos^2\theta}\omega_z & 0 \end{bmatrix} \quad (11)$$

and approximating the derivative of $\omega(t)$ by its first difference,

$$\omega(t) \approx \frac{\omega(t+\Delta t)-\omega(t)}{\Delta t} \tag{12}$$

and substituting (11) and (12) into (10) yields

$$\ddot{\theta}(t) = \mathbf{V}_B(\theta(t),\omega(t))\ \mathbf{W}_B(\theta(t))\ \omega(t) + \mathbf{W}_B(\theta(t))\frac{\omega(t+\Delta t)-\omega(t)}{\Delta t} \tag{13}$$

Plugging (6) and (13) into (9) and rearranging terms slightly leads to

$$\theta(t+\Delta t) = \theta(t) + \mathbf{W}_B \frac{\omega(t)+\omega(t+\Delta t)}{2}\Delta t + \mathbf{V}_B\mathbf{W}_B\omega(t)\frac{\Delta t^2}{2} \tag{14}$$

which is the second order DT integration step formula implemented in the attitude computer. Since Δt remains as an explicit parameter in this formula, it is unnecessary to have constant stepsize. This eliminates the difficulties of an interrupt driven program structure that would be necessary to have constant sampling rate data acquisition.

4.4 DT linearized error dynamics

Equation (14) defines a nonlinear state propagation function $\mathbf{f}_{\Delta t}$ for the system with state vector θ and input ω:

$$\theta(t+\Delta t) = \mathbf{f}_{\Delta t}(\theta(t),\omega(t),\omega(t+\Delta t),t) \tag{15}$$

For the sake of obtaining an extended Kalman filter which can estimate both orientation errors and gyro biases, consider augmenting the state vector with ω and rewriting the system in the form

$$\begin{bmatrix}\theta(t+\Delta t)\\ \omega(t+\Delta t)\end{bmatrix} = \tilde{\mathbf{f}}_{\Delta t}\left(\begin{bmatrix}\theta(t)\\ \omega(t)\end{bmatrix}\right) + \mathbf{u}(t)$$

$$\tilde{\mathbf{f}}_{\Delta t}\left(\begin{bmatrix}\theta(t)\\ \omega(t)\end{bmatrix}\right) = \begin{bmatrix}\mathbf{f}_{\Delta t}(\theta(t),\omega(t),\omega(t+\Delta t),\Delta t)\\ \omega(t)\end{bmatrix} \tag{16}$$

$$\mathbf{u}(t) = \begin{bmatrix}\mathbf{0}\\ \omega(t+\Delta t)-\omega(t)\end{bmatrix}$$

where $\mathbf{u}(t)$ has been deviously chosen to make $\omega(t)$ evolve in accordance with the input history of the previous system. The system error dynamics can now be obtained by linearizing about the nominal trajectory $[\theta(t)\quad \omega(t)]^T$ to get

$$\begin{bmatrix}\delta\theta(t+\Delta t)\\ \delta\omega(t+\Delta t)\end{bmatrix} = \begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{0} & \mathbf{I}\end{bmatrix}\begin{bmatrix}\delta\theta(t)\\ \delta\omega(t)\end{bmatrix} \tag{17}$$

where

$$\mathbf{A} = \frac{\partial \mathbf{f}_{\Delta t}(t)}{\partial\theta(t)} = \mathbf{I} + \mathbf{V}_B\Delta t + \left[\mathbf{V}_B{}^2 + \left(\frac{\partial}{\partial\theta}\mathbf{V}_B\right)\mathbf{W}_B\omega\right]\frac{\Delta t^2}{2}$$

$$\mathbf{B} = \frac{\partial \mathbf{f}_{\Delta t}(t)}{\partial\omega(t)} = \mathbf{W}_B\Delta t + \left[\mathbf{V}_B\mathbf{W}_B + \left(\frac{\partial}{\partial\omega}\mathbf{V}_B\right)\mathbf{W}_B\omega\right]\frac{\Delta t^2}{2} \tag{18}$$

and $\mathbf{0}$ and $\mathbf{I}$ are 3-by-3 zero and identity matrices. The vector partial derivatives of $\mathbf{V}_B$ are too messy to write out in full, but the computation is straightforward and can be carried out as follows: 1) form a "row vector" of the three matrices obtained by differentiating $\mathbf{V}_B$ with respect to the first, second and third elements of the vector in the denominator of the partial derivative; 2) multiply each of these three matrices by the r.h.s. vector $\mathbf{W}_B\omega$. This results in a "row vector of column vectors", i.e. a 3-by-3 matrix.

Equation (17) gives the state transition matrix for the linearized error dynamics of the augmented system. The angular velocity errors $\delta\omega$ are principally due to gyro biases, and will be interpreted simply as gyro biases from here on. The $\mathbf{A}$ and $\mathbf{B}$ submatrices can be interpreted as describing the influence of the orientation error and gyro biases at time t on the orientation error at time $t+\Delta t$. The effect of the matrix is fairly obvious; it basically mimics the attitude computation of (14) except that the input angular velocity is due to gyro biases and the output is therefore an orientation *error*. The growth of orientation error in the absence of angular rate errors is governed by the $\mathbf{A}$ matrix. To first order $\mathbf{A} = \mathbf{I} + \mathbf{V}_B\Delta t$. The identity term maintains the previously accrued error, and $\mathbf{V}_B(\theta,\omega)$ amplifies existing orientation errors in response to motion.

4.5 Separate-bias Kalman filter formulation

The linear error propagation model of (17) provides the basis for a complimentary Kalman filter to estimate these errors. The model has been manipulated into a form in which the gyro biases are assumed constant, thus permitting the direct application of Friedland's separate-bias Kalman filtering results [21]. If the constant-bias model turns out to fit the gyro performance poorly, the restriction can later be ameliorated by use of an age-weighting factor. If an exponential gyro warm-up model as in [15] seems more appropriate, this can be accommodated within Friedland's formulation by replacing the identity submatrix in the state transition matrix of (17).

Switching to Friedland's notation, define an error state vector $\mathbf{x}_k \equiv \delta\theta(t_k)$ and a bias state vector $\mathbf{b}_k \equiv \delta\omega(t_k)$ where t_k is the time at the k^{th} iteration of the algorithm. An augmented state vector $\mathbf{z}_k \equiv [\mathbf{x}_k \quad \mathbf{b}_k]^T$ satisfies

$$\mathbf{z}_{k+1} = \mathbf{F}_k \mathbf{z}_k + \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{w}_k$$
$$\mathbf{F}_k = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (19)$$

The additive white noise $\mathbf{w_k}$, with variance $\mathbf{Q_k}$, only effects **x**, since **b** is assumed constant. The measurement equation is

$$\mathbf{y}_k = \mathbf{L}_k \mathbf{z}_k + \mathbf{v}_k , \quad (20)$$

where $\mathbf{v_k}$ is white noise with variance $\mathbf{Q_k}$. In Friedland's paper, $\mathbf{L}_k = [\mathbf{H}_k \quad \mathbf{C}_k]$, but in this application the measurements from the inclinometers and compass only measure **x** and not **b**, so **C** = **0** will be used throughout, resulting in a great simplification from Friedland's derivation.

Applying Kalman filtering to this model, the optimal estimate of **z** is

$$\hat{\mathbf{z}}_{k+1} = \mathbf{F}_k \hat{\mathbf{z}}_k + \mathbf{K}(k+1)\left(\mathbf{y}_{k+1} - \mathbf{L}\mathbf{F}_k \hat{\mathbf{z}}_k\right) \quad (21)$$

$$\mathbf{K}(k) = \mathbf{P}(k)\mathbf{L}^T\left[\mathbf{L}\mathbf{P}(k)\mathbf{L}^T + \mathbf{R}_k\right]^{-1}. \quad (22)$$

The Ricatti equations for the recursive computation of the estimation error covariance matrix **P**(k) needed in the Kalman gain expression can be rolled together into the single predictor-to-predictor covariance update equation:

$$\mathbf{P}(k+1) = \mathbf{F}_k\left[\mathbf{I} - \mathbf{K}(k)\mathbf{L}\right]\mathbf{P}(k)\mathbf{F}_k^T + \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}\mathbf{Q}_{k+1}[\mathbf{I} \quad \mathbf{0}]. \quad (23)$$

Partitioning **P**(k) into 3-by-3 submatrices as

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_x(k) & \mathbf{P}_{xb}(k) \\ \mathbf{P}_{xb}^T(k) & \mathbf{P}_b(k) \end{bmatrix}, \quad (24)$$

the expression for the Kalman gain, (22), may be rewritten in partitioned form as

$$\begin{bmatrix} \mathbf{K}_x(k) \\ \mathbf{K}_b(k) \end{bmatrix} = \begin{bmatrix} \mathbf{P}_x(k)\mathbf{H}^T\left[\mathbf{H}\mathbf{P}_x(k)\mathbf{H}^T + \mathbf{R}_k\right]^{-1} \\ \mathbf{P}_{xb}^T(k)\mathbf{H}^T\left[\mathbf{H}\mathbf{P}_x(k)\mathbf{H}^T + \mathbf{R}_k\right]^{-1} \end{bmatrix}. \quad (25)$$

These separate gains are used in two essentially separate Kalman filters, one for estimating **x** and one for **b**. To compute the $\mathbf{K_x}$ and $\mathbf{K_b}$ gains in (25), covariance submatrices $\mathbf{P_x}$ and $\mathbf{P_{xb}}$ are needed. These are updated by the partitioned version of (23):

$$\begin{bmatrix} \mathbf{P}_x^+ & \mathbf{P}_{xb}^+ \\ \mathbf{P}_{xb}^{T+} & \mathbf{P}_b^+ \end{bmatrix} = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \mathbf{K}_x \\ \mathbf{K}_b \end{bmatrix}[\mathbf{H} \quad \mathbf{0}]\right) \times$$
$$\begin{bmatrix} \mathbf{P}_x & \mathbf{P}_{xb} \\ \mathbf{P}_{xb}^T & \mathbf{P}_b \end{bmatrix}\begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ \mathbf{0} & \mathbf{I} \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$
$$= \begin{bmatrix} \mathbf{A}_k - \mathbf{A}_k\mathbf{K}_x\mathbf{H} - \mathbf{B}_k\mathbf{K}_b\mathbf{H} & \mathbf{B}_k \\ -\mathbf{K}_b\mathbf{H} & \mathbf{I} \end{bmatrix} \times \quad (26)$$
$$\begin{bmatrix} \mathbf{P}_x\mathbf{A}_k^T + \mathbf{P}_{xb}\mathbf{B}_k^T & \mathbf{P}_{xb} \\ \mathbf{P}_{xb}^T\mathbf{A}_k^T + \mathbf{P}_b\mathbf{B}_k^T & \mathbf{P}_b \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Thus, a plethora of 6-by-6 matrix multiplications and one 6-by-6 inversion are replaced by a somewhat greater number of 3-by-3 multiplications and one 3-by-3 inversion.

5. Implementation

Figure 4 illustrates the configuration of the hardware built to demonstrate the inertial head-attitude tracking concept. The sensors are all embedded in a specially machined 2" X 2" X 1.25" plastic block connected by a thin 10' cable to an analog signal conditioning circuit and data acquisition card in a PC.

Software was written in "C" to run on the PC and implement the basic loop shown in Figure 5.

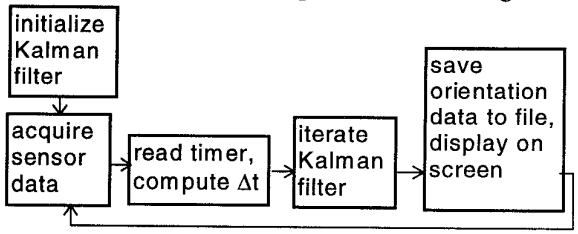

Figure 5: Inertial orientation tracker main software loop.

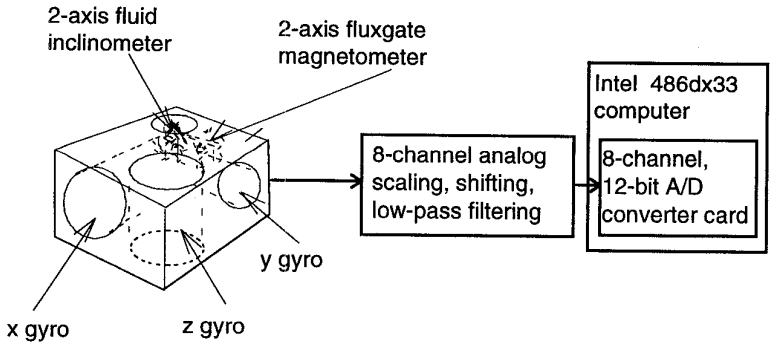

Figure 4: Orientation tracker hardware configuration.

The initialization block, executed once at program start-up, sets the initial state estimates and covariances as follows:

$\mathbf{x}_0$: The inclinometer is read and used to set ψ and θ. The compass, if used, determines ϕ; otherwise ϕ=0.

$\mathbf{b}_0$: The biases of all 3 gyros are measured during system calibration and stored in a file. On initialization, the file is read and $\delta\boldsymbol{\omega}$ is initialized with the stored gyro biases.

$\mathbf{P}_x(0)$: The errors in the initial determination of the Euler angles may be substantial, but they are assumed to be uncorrelated with one another: $\mathbf{P}_x(0) = \mathbf{I}$.

$\mathbf{P}_b(0)$: The gyro biases at start-up could differ substantially from the prerecorded calibration values, but the uncertainties are uncorrelated: $\mathbf{P}_b(0) = 0.1\mathbf{I}$.

$\mathbf{P}_{xb}(0)$: The initial uncertainties in orientation and gyro bias are completely uncorrelated: $\mathbf{P}_{xb}(0) = \mathbf{0}$.

The data acquisition block scans all the A/D channels in rapid succession. The new gyro readings are stored as $\omega(t+\Delta t)$ and the previous ones are moved back to $\omega(t)$. The new inclinometer and compass readings are stored in $\mathbf{y}(t+\Delta t)$. In the next block, a timestamp is obtained from the 8253 timer/counter chip on the PC motherboard. This counter is driven by a 1.19 MHz oscillator with a 65,536 divisor to generate 18.2 Hz timer ticks for BIOS and DOS time-keeping. By reprogramming the divisor it was found possible to obtain sub-microsecond timing resolution as required for inertial integration. Δt is calculated as the difference between the current timestamp and the previous one.

Next, $\omega(t)$, $\omega(t+\Delta t)$ and Δt are fed into the Kalman filter update block. $\mathbf{W}_B$ and $\mathbf{V}_B$ are computed and then used in (14) to compute the predicted $\theta(t+\Delta t)$. This corresponds to the attitude computation block in. Since the Euler angle estimates, $\hat{\theta}$ must be maintained anyway, it is convenient to subsume $\delta\hat{\theta}$ into them, and keep track of total estimates only. This does not change the filter framework developed in the previous section in any important way; it just means that $\delta\hat{\theta}(t)$ is always zero at the beginning of each iteration of the Kalman filter. At the end of the Kalman filter update cycle, $\delta\hat{\theta}(t+\Delta t)$ is used to reset $\hat{\theta}(t+\Delta t)$ and then flushed back to zero before the next cycle. Since the attitude error estimates are propagated along with the attitude estimates through the nonlinear propagation equation, the top three elements of $\mathbf{F}_k\hat{\mathbf{z}}_k$ in (21) are replaced with zeros. Since ω is *not* included in the state, the running estimates of $\delta\hat{\omega}$ must still be kept track of in the Kalman filter. They are propagated through the prediction step unchanged, as governed by the bottom three rows of $\mathbf{F}_k$. The system, then , can be thought of as a mixture of a purely complimentary Kalman filter as described in the previous section and an extended Kalman filter which keeps track of total estimates of state.

The next stage in the computational loop is to incorporate the measurements and update the error estimates as follows

$$\begin{aligned}\delta\hat{\theta}_{k+1} &= \delta\hat{\theta}_{k+1} + \mathbf{K}_x(k+1)\nu_{k+1} \\ \delta\hat{\omega}_{k+1} &= \delta\hat{\omega}_{k+1} + \mathbf{K}_b(k+1)\nu_{k+1}\end{aligned} \tag{27}$$

where ν_{k+1} is the innovations obtained by subtracting the predicted orientation estimates from the new orientation measurements. In order to calculate $\mathbf{K}_x(k+1)$ and $\mathbf{K}_b(k+1)$ with equation (25) it is necessary to first propagate the covariance submatrices using (26). Since the inclinometer and compass signals are pre-processed to give direct measurements of the Euler angles, $\mathbf{H}=\mathbf{I}$, and (26) is simplified to the following steps:

$$\begin{aligned}\mathbf{T}_1 &= \mathbf{A} - \mathbf{A}\mathbf{K}_x \\ \mathbf{T}_2 &= \mathbf{T}_1\mathbf{P}_{xb} \\ \mathbf{T}_3 &= \mathbf{B}\mathbf{T}_2{}^T \\ \mathbf{P}_b{}^+ &= \mathbf{T}_2 + \mathbf{B}\mathbf{P}_b \\ \mathbf{P}_{xb}{}^+ &= \mathbf{T}_2 + \mathbf{B}\mathbf{P}_b{}^+ \\ \mathbf{P}_x{}^+ &= \mathbf{P}_{xb}{}^+\mathbf{B}^T + \mathbf{T}_3 + \mathbf{T}_1\mathbf{P}_x\mathbf{A}^T\end{aligned} \tag{28}$$

where $\mathbf{T}_i$ are simply temporary storage matrices used to reduce the amount of redundant matrix multiplication. A small subroutine library was written, following the pointer conventions and numerical methods described in [22], to perform the necessary matrix multiplication, transposition, addition and inversion operations to carry out these steps.

5.1 The $\mathbf{Q}_k$ and $\mathbf{R}_k$ Matrices

Ideally, $\mathbf{Q}_k$ is supposed to reflect the magnitude of a white noise sequence. If all error sources in the inertial attitude system are taken care of (i.e. modeled in the state propagation matrix), then $\mathbf{w}_k$ in (19) should be entirely due to the noise floors of the angular rate sensors. In this case, it should be possible to calculate the optimal value of $\mathbf{Q}_k$ by measuring the noise covariance, $\mathbf{Q}$, of the stationary gyros in advance, then at each time step compute $\mathbf{Q}_k = \mathbf{G}_k\mathbf{Q}\mathbf{G}_k{}^T$, using $\mathbf{G}_k = \mathbf{W}_B\big(\theta(t_k)\big)$.

However, there are many nonwhite error sources besides bias, such as nonlinearity, hysteresis, misalignment, g-sensitivity, and scale factor temperature coefficient, none of which are modeled in the current implementation. The best procedure for designing a reduced-order Kalman filter under these circumstances is to use a Schmidt-Kalman filter, which eliminates the unmodeled states from the state vector, but continues to propagate their covariances in partitioned Ricatti equations and $\mathbf{Q}$ and $\mathbf{R}$ matrices. A simpler approach, which sometimes works almost as well [12, p. 397], is to just ignore the unmodeled states, but "bump up" the $\mathbf{Q}$ and $\mathbf{R}$ matrices to account for the noises in the states being discarded. This approach is taken here, except the "bumping up" is done in a very inexact way.

Without having a model of the gyro dynamics, the following error sources in the process equation (19) are assumed:

gyro noise: From an oscilloscope, for stationary gyros, $\sigma \approx 0.01$ rad/s. Covariance per step $(0.01\Delta t)^2$.

integration rule error: From the analysis in Section 4.3, $\sigma \approx \omega^3\Delta t^2$ rad/s. Covariance per step $\omega^6\Delta t^6$.

scale factor error: This is a composite of nonlinearity and temperature dependent scale factor variations. Assuming scale factor accuracy of 1% of full scale, $\sigma \approx 0.01\omega$ rad/s. Covariance per step $(0.01\omega\Delta t)^2$.

Assuming $\Delta t = 0.01$sec, and that these error sources are uncorrelated, the error covariances add up to approximately $10^{-8}(1+\omega^2+10^{-4}\omega^6)$. At each iteration of the Kalman filter software, the following algorithm is used to compute $\mathbf{Q}_k$:

1. find $\omega_{max} = \max(\omega_x, \omega_y, \omega_z)$
2. set $\sigma_w^2 = 10^{-8}(1+\omega_{max}^2+10^{-4}\omega_{max}^6)$
3. set $\tilde{\mathbf{Q}}_k = \begin{bmatrix} \sigma_w^2 & 0 & 0 \\ 0 & \sigma_w^2 & 0 \\ 0 & 0 & \sigma_w^2 \end{bmatrix}$
4. set $\mathbf{Q}_k = \mathbf{W}_B \tilde{\mathbf{Q}}_k \mathbf{W}_B^T$

This algorithm is very crude and likely to overestimate $\mathbf{Q}_k$ because it uses ω_{max} to find the variance for all three diagonal elements of $\mathbf{Q}_k$.

$\mathbf{R}_k$ is modeled in an equally sloppy manner. The measurement noise is extremely nonwhite. The major source of measurement noise for the fluid inclinometers is "slosh" caused by transverse linear accelerations. Linear motion is not included in the state vector, and therefore, this error cannot be modeled in the measurement matrix. Furthermore, the magnitude of the low-frequency "slosh" errors are sometimes extremely large: up to 1 radian. Slosh-induced inclination errors cause similarly large heading errors in the compass system. On the other hand, when the head is still, there is no slosh and the attitude angles measured by the inclinometer are very accurate. The algorithm for $\mathbf{R}_k$ is therefore designed in a heuristic way to force the Kalman filter to take good advantage of the measurements when they are likely to be meaningful, and to ignore them when they are likely to be erroneous. The basic principle is that σ_v should approach 1 when slosh is likely, and approach the static accuracy of the inclinometer/compass measurements, about 0.01 radians, when slosh is very unlikely. In the absence of a model for human head motion, it is assumed that a person cannot sustain a constant linear acceleration of the head very long with no rotation. Therefore, the longer the period of time that the head has had 1) zero angular velocity, and 2) unchanging inclinometer outputs, the higher the probability that the head is still. Based on this intuition, the algorithm used to set $\mathbf{R}_k$ is:

1. compute "stilltime", τ, since last non-zero gyro reading OR last change in inclinometer reading.
2. set $\sigma_v = 1/(1+400\tau)$
3. if $\sigma_v < 0.01$, set $\sigma_v = 0.01$
4. set $\mathbf{R}_k = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_v \end{bmatrix}$.

According to this algorithm, the measurement error covariances for the inclinometer roll and pitch range from 1, during periods of likely slosh, down to 10^{-4}, during periods of likely stillness. The covariance of the compass yaw error only comes down to 0.01, corresponding to $\sigma \approx 6°$, because even with good inclinometer information, magnetic distortions in the room make the compass this inaccurate.

6. Results

Using the $\mathbf{Q}_k$ and $\mathbf{R}_k$ matrices described above, it was found that the Kalman filter diverged within a few seconds when the sensor was still. An age weighting multiplier did not help. After much experimentation, it was found that the only way to prevent divergence is to never let the diagonal elements of $\mathbf{R}_k$ be less than 1. The algorithm for $\mathbf{R}_k$ was adjusted so that σ_ϖ ranges from 10, when $\tau=0$, to 1, when $\tau>0.2$. The base level of $\mathbf{Q}_k$ was also boosted from 10^{-8} to 10^{-4} so that the filter would still make use of the measurements with the larger measurement noise covariance. With these modifications, the filter remains stable indefinitely and succeeds in eliminating long term drift without compromising the rapid dynamic response of the inertial tracking technique. The filter can run at approximately 200 iterations/second. This is a five-fold slowdown as compared to the raw attitude computation with the Kalman filtering steps commented out. However, it is still reasonably fast and the delay can be compensated for by prediction if necessary.

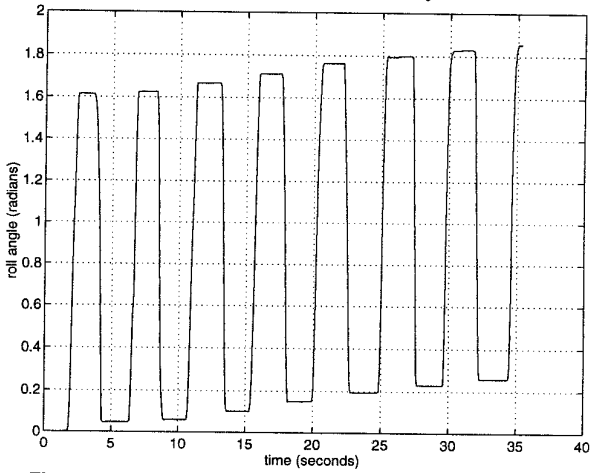

Figure 6: Test run without complimentary Kalman filter.

To demonstrate the behavior of the filter, two datasets were collected. In the first dataset, the complimentary

Kalman filter block is disabled by setting $\mathbf{K}_x$ and $\mathbf{K}_b$ equal to zero. During the test period of approximately 35 seconds, the sensor block was repeatedly turned through +90° about the roll axis and left to rest on its right side, then returned to rest in its horizontal orientation on the table. The roll Euler angle is plotted against time in Figure 6, which demonstrates the problem with unaided inertial integration: the accumulated drift error by the end of the run is about 15°. The second dataset is created by a similar motion sequence, but the Kalman filter is in effect. As Figure 7 shows, the filter incorporates the drift-free but noisy measurements from the inclinometers, and effectively compensates the drift of the inertial system. Due to the time-varying $\mathbf{R}_k$ strategy which shuts out the measurements during motion, a certain amount of error accumulates each time the sensor is rolled over and back, and the Kalman filter corrects it once the sensor returns to a stationary pose. The graph clearly shows the time-course of this corrective action.

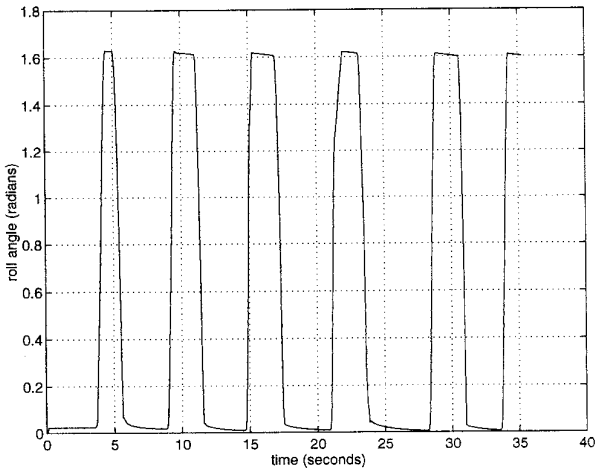

Figure 7: Test run with complimentary Kalman filter.

7. Acknowledgements

This work was supported by NASA Ames Research Center grant NCC 2-771 and a Link Foundation Fellowship, for which the author is of course very thankful.

8. References

[1] D. K. Bhatnagar, "Position Trackers for Head Mounted display Systems: A Survey," University of North Carolina, Chapel Hill TR93-010, March 1993.

[2] F. J. Ferrin, "Survey of Helmet Tracking Technologies," SPIE, vol. 1456, pp. 86-94, 1991.

[3] K. Meyer, H. L. Applewhite, and F. A. Biocca, "A Survey of Position Trackers," Presence, vol. 1, pp. 173--200, 1992.

[4] K. R. Britting, Inertial Navigation Systems Analysis. New York: Wiley-Interscience, 1971.

[5] C. Broxmeyer, Inertial Navigation Systems. New York: McGraw-Hill, 1964.

[6] J. L. Farrell, Integrated Aircraft Navigation. New York: Academic Press, 1976.

[7] A. Lawrence, Modern Inertial Technology: Springer-Verlag, 1993.

[8] R. H. Parvin, Inertial Navigation. Princeton, New Jersey: Van Nostrand, 1962.

[9] G. M. Siouris, Aerospace Avionics Systems: A Modern Synthesis. San Diego, CA: Academic Press, 1993.

[10] E. Foxlin, "Inertial Head-Tracking," M.S. Thesis, Dept. of Elec. Engineering and Comp. Sci., Mass. Inst. of Technology, Cambridge, MA, 1993

[11] E. Foxlin, "An Inertial Head-Orientation Tracker with Automatic Drift Compensation for use with HMD's," in Virtual Reality Software & Technology, G. Singh, S. K. Feiner, and D. Thalmann, Eds. Singapore: World Scientific, 1994, pp. 159-174.

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

[13] B. E. Bona and R. J. Smay, "Optimum Reset of Ship's Inertial Navigation System," IEEE Transactions on Aerospace and Electronic Systems, vol. AES-2, pp. 409-414, 1966.

[14] M. Koifman and S. J. Merhav, "Autonomously Aided Strapdown Attitude Reference System," Journal of Guidance and Control, vol. 14, pp. 1164--1172, 1990.

[15] B. Barshan and H. F. Durrant-Whyte, "Evaluation of a Solid-State Gyroscope for Robotics Applications," IEEE Transactions on Instrumentation and Measurement, vol. 44, pp. 61-67, 1995.

[16] J. Vaganay, M. J. Aldon, and A. Fournier, "Mobile Robot Attitude Estimation by Fusion of Inertial Data," in Proceedings - IEEE International Conference on Robotics and Automation, vol. 1, 1993, pp. 277-282.

[17] R. Azuma and G. Bishop, "Improving Static and Dynamic Registration in an Optical See-through HMD," in Proceedings of SIGGRAPH '94. Orlando, Florida: ACM, 1994.

[18] S. Emura and S. Tachi, "Sensor Fusion Based Measurement of Human Head Motion," in Proceedings of 3rd IEEE International Workshop on Robot and Human Communication, 1994, pp. 124--129.

[19] S. Emura and S. Tachi, "Compensation of Time Lag Between Actual and Virtual Spaces by Multi-Sensor Integration," in Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems(MFI'94), 1994, pp. 463--469.

[20] J. M. Cooke, M. J. Zyda, D. R. Pratt, and R. B. McGhee, "NPSNET: Flight Simulation Dynamic Modeling Using Quaternions," *Presence*, vol. 4, 1993.

[21] B. Friedland, "Treatment of Bias in Recursive Filtering," IEEE Transactions on Automatic Control, vol. AC-14, pp. 359-367, 1969.

[22] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Veterling, Numerical Recipes in C: Cambridge University Press, 1988.

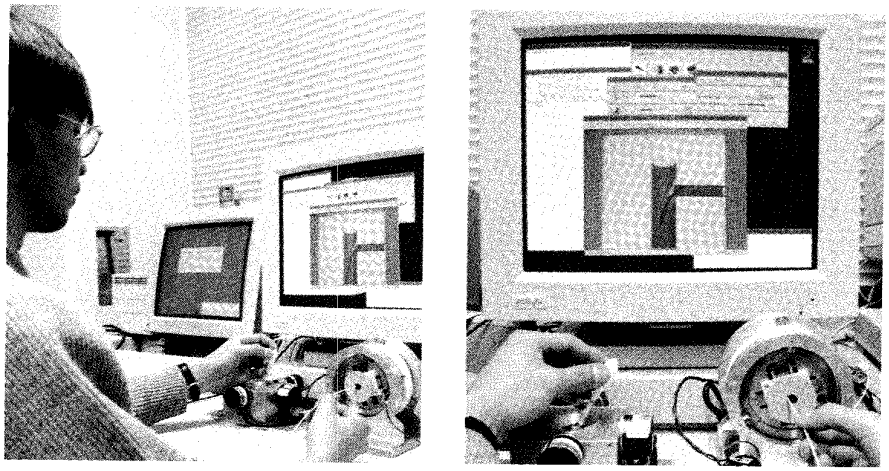

Plate 7: *Distributed Virtual...* – pg. 79; Fig. 12: Outline of the Simulator

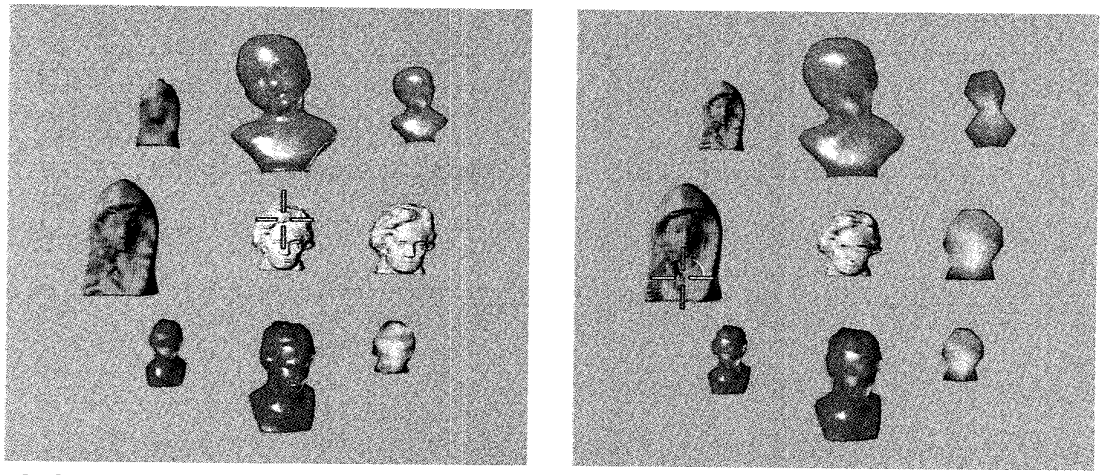

Plate 8: *Gaze-Directed...* – pg. 103; Figs.: Adaptively Rendered Images Using Two Gaze Directions

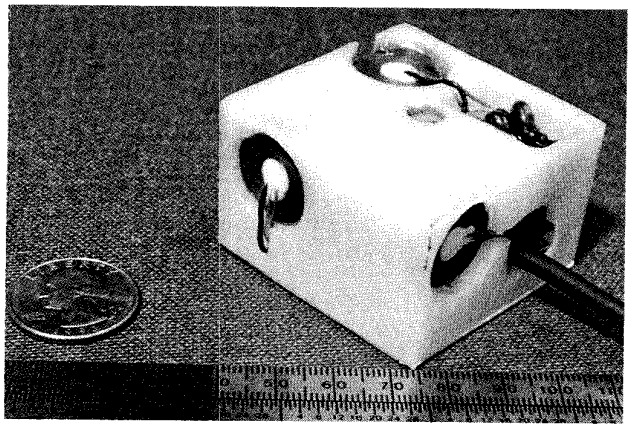

Plate 9: *Inertial Head-Tracker...* – pg. 185; Fig. 1: MIT Inertial Tracker 2nd Prototype