NASA-CR-200870

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

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

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

purpose of the Kalman filter is to estimate orientation, it is a given that it will be included in the state vector.

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 $\dot{\omega}$, which changes the model to an assumption of constant angular acceleration. The difference between the true $\ddot{\omega}$ and the assumed $\ddot{\omega} = \mathbf{0}$ 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

driven by muscle forces - an unknown input - so head dynamics are not modeled in the current system.

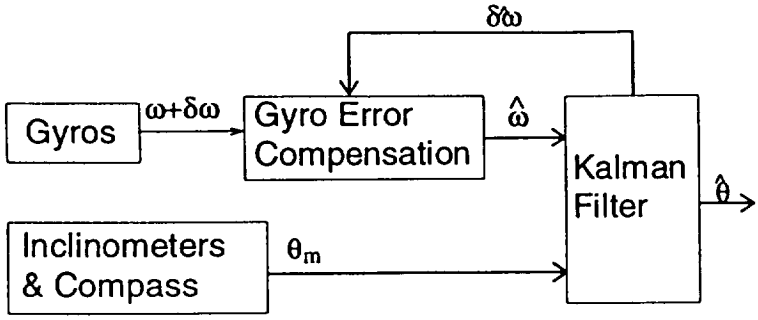


Figure 2: Direct Kalman filter for orientation

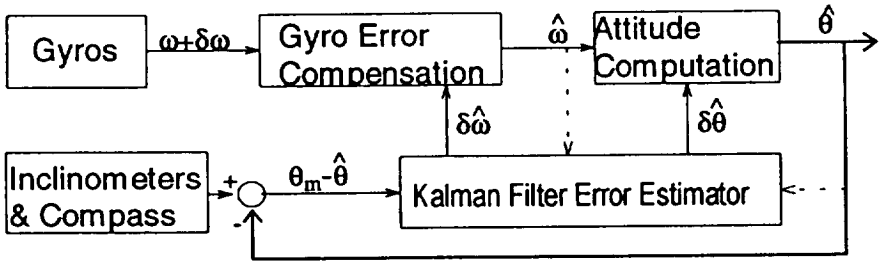


Figure 3: Complimentary Kalman filter for orientation

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 ω in the state vector. Since the head dynamics are not being modeled in this implementation, ω is excess baggage, and by removing it from **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

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

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 δθ represents the error in the output of the attitude computer, and δω 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 θ(t) over a short time interval by its Taylor series expansion

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

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

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

Defining (with time indices suppressed for brevity)

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

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

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

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

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 **A** and **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 **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 $\mathbf{x}$, since $\mathbf{b}$ 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 $\mathbf{x}$ and not $\mathbf{b}$, so $\mathbf{C} = \mathbf{0}$ will be used throughout, resulting in a great simplification from Friedland's derivation.

Applying Kalman filtering to this model, the optimal estimate of $\mathbf{z}$ is

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

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

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:

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

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

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

These separate gains are used in two essentially separate Kalman filters, one for estimating $\mathbf{x}$ and one for $\mathbf{b}$. 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.

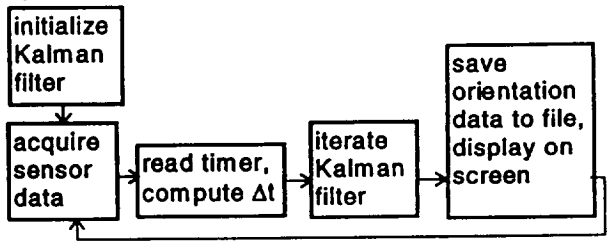


Figure 5: Inertial orientation tracker main software loop.

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 $\delta\omega$ 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: $\mathbf{P}_x(0) = \mathbf{I}$.

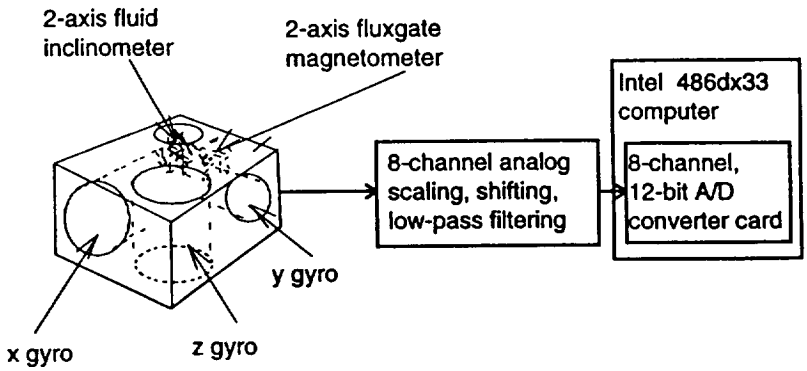


Figure 4: Orientation tracker hardware configuration.

$\mathbf{P}_b(0)$: 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 $y(t+\Delta t)$. 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 $\delta\hat{\omega}$

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

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

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 [illegible] down to 0.01, corresponding to [illegible]

[redacted]

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 $\mathbf{Q}_k = \mathbf{W}_B \tilde{\mathbf{Q}}_k \mathbf{W}_B^T$

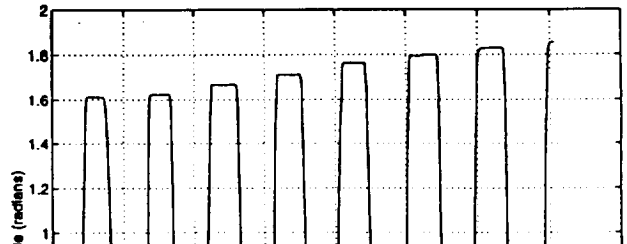
This algorithm is very crude and likely to overestimate $\mathbf{Q}_k$ because it uses ω_{max} to find the variance for all three diagonal elements of $\mathbf{Q}_k$.

$\mathbf{R}_k$ 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

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_m 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.



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

[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

Proceedings of the

IEEE 1996 Virtual Reality Annual International Symposium

March 30 – April 3, 1996 Santa Clara, California

Sponsored by

The IEEE Computer Society Technical Committee on Computer Graphics
The IEEE Neural Networks Council Virtual Reality Technical Committe

IEEE Computer Society Press
Los Alamitos, California

Washington • Brussels • Tokyo

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264

Copyright © 1996 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved.

Copyright and Reprint Permissions: Abstracting is permitted with credit to the source. Libraries may photocopy beyond the limits of US copyright law, for private use of patrons, those articles in this volume that carry a code at the bottom of the first page, provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.

Other copying, reprint, or republication requests should be addressed to: IEEE Copyrights Manager, IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331.

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society Press, or the Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Press Order Number PR07295
ISBN 0-8186-7295-1
ISSN 1087-8270

IEEE Order Plan Catalog Number 96CB35922
IEEE Order Plan ISBN 0-8186-7296-X
Microfiche ISBN 0-8186-7297-8

Additional copies may be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264
Tel: +1-714-821-8380
Fax: +1-714-821-4641
Email: cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: +1-908-981-1393
Fax: +1-908-981-9667
mis.custserv@computer.org

IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM
Tel: +32-2-770-2198
Fax: +32-2-770-8505
euro.ofc@computer.org

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN
Tel: +81-3-3408-3118
Fax: +81-3-3408-3553
tokyo.ofc@computer.org

Editorial and cover production by Mary E. Kavanaugh
Printed in the United States of America by KNI Inc.

The Institute of Electrical and Electronics Engineers, Inc.