1 Comparison of Kalman Filters for fusing altitude and acceleration data

This document describes two Kalman filters for fusing altitude data and acceleration data to estimate altitude and climb/sinkrate in an altimeter/variometer used by glider pilots.

Altitude measurements are obtained from a barometric pressure sensor. Gravity-compensated vertical acceleration measurements are obtained from an IMU sensor and AHRS software.

The first algorithm (KF3) is less computation-intensive, with 3 state variables. It uses acceleration data in the prediction phase even though acceleration is not included in the state vector, state transition or measurement model.

The second algorithm (KF4) has 4 state variables including acceleration. Acceleration data is used in the update phase as part of the measurement model.

Conclusion : The KF4 model provides perceptible improvements in performance at the cost of increased code complexity, size and nearly double the execution time. Our application is implemented on an ESP32 microcontroller with 4MBytes of flash and clocked at 80MHz. There is no issue with the increased code size, and execution time is still acceptable given our real-time data sampling constraints.

2 KalmanFilter3 Algorithm

2.1 Process Model : physical equations

k is the sample time index

z_k is the altitude, in cm

v_k is the vertical velocity, in cm/s

a_k is the vertical acceleration, in cm/s^2

b_k is the vertical acceleration bias, in cm/s^2. This is the residual bias after accelerometer calibration and can drift unpredictably.

The true vertical acceleration is the measured acceleration minus the acceleration bias.

These variables at time index k are related to the variables at time $k-1$ by the equations

$$\begin{aligned} z_k &= z_{k-1} + dt * v_{k-1} + \frac{dt^2}{2} * (a_{k-1} - b_{k-1}) \\ v_k &= v_{k-1} + dt * (a_{k-1} - b_{k-1}) \\ b_k &= b_{k-1} + \eta_b \end{aligned}$$

where dt is the time interval between samples and η_b is a random variable representing acceleration bias noise.

2.2 Process Model : state transition Representation

In vector/matrix notation, these equations can be written as

$$\begin{bmatrix} z_k \\ v_k \\ b_k \end{bmatrix} = \begin{bmatrix} 1 & dt & \frac{-dt^2}{2} \\ 0 & 1 & -dt \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_{k-1} \\ v_{k-1} \\ b_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{dt^2}{2} & 0 \\ dt & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_{k-1} \\ \eta_b \end{bmatrix}$$

which is a state-transition equation in the form

$$x_k = F x_{k-1} + L w_k$$

x_k is our state vector

$$x_k = \begin{bmatrix} z_k \\ v_k \\ b_k \end{bmatrix}$$

F is the state transition matrix

$$F = \begin{bmatrix} 1 & dt & \frac{-dt^2}{2} \\ 0 & 1 & -dt \\ 0 & 0 & 1 \end{bmatrix}$$

w_k is the process noise (environmental perturbation), in which a_k represents the random acceleration input, and η_b is a random variable representing acceleration bias noise

$$w_k = \begin{bmatrix} a_k \\ \eta_b \end{bmatrix}$$

L is the process noise transformation matrix

$$L = \begin{bmatrix} \frac{dt^2}{2} & 0 \\ dt & 0 \\ 0 & 1 \end{bmatrix}$$

The process state covariance matrix $P_k = E(x_k * x_k^T)$

$$P_k = \begin{bmatrix} P_{zz} & P_{zv} & P_{zb} \\ P_{vz} & P_{vv} & P_{vb} \\ P_{bz} & P_{bv} & P_{bb} \end{bmatrix}$$

Note the state covariance matrix is symmetric, i.e. $P_{zv} = P_{vz}$, $P_{zb} = P_{bz}$, $P_{vb} = P_{bv}$.

2.3 Sensor Measurements

In this model, a barometric sensor provides periodic measurements for altitude zm. These measurements are assumed to be corrupted by sensor noise ν_z which is described by a zero-mean Gaussian probability distribution.

So, the sensor measurement zm_k is related to the true altitude z_k by the equation

$$zm_k = z_k + \nu_z$$

In vector/matrix notation

$$[zm_k] = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_k \\ v_k \\ b_k \end{bmatrix} + [v_z]$$

or

$$m_k = H_k * x_k + \nu_k$$

where m_k is the measurement vector, H_k is the measurement model matrix transforming the process state vector to measurement vector space, and ν_k is the measurement noise vector. In this case, both m_k and ν_k are scalars.

Note that we are also getting periodic gravity-compensated acceleration measurements a_k, but we cheat and use the acceleration measurements as input in the prediction step.

2.4 Prediction

Assuming the process noise w components can be described by zero-mean gaussian probability distributions, the process noise covariance matrix Q_k is

$$Q_k = L * E(w_k * w_k^T) * L^T$$

Let

$$N_k = E(w_k * w_k^T)$$

$$N_k = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix}$$

where σ_a^2 is the statistical variance of acceleration perturbations from the environment, and σ_b^2 is the statistical variance of acceleration bias noise.

$$Q_k = L * N_k * L^T$$

$$Q_k = \begin{bmatrix} (\frac{dt^4}{4}\sigma_a^2) & (\frac{dt^3}{2}\sigma_a^2) & 0 \\ (\frac{dt^3}{2}\sigma_a^2) & (dt^2\sigma_a^2) & 0 \\ 0 & 0 & \sigma_b^2 \end{bmatrix}$$

Assuming process model noise is additive and zero-mean, the predicted (a priori) state estimate $\hat{x_k^-}$ is

$$\hat{x_k^-} = F_{k-1} * \hat{x_{k-1}^+}$$

where $\hat{x_{k-1}^+}$ is the updated (a posteriori) best state estimate for index $k-1$.

Here we cheat and use the measured gravity-compensated acceleration am_k to predict z_k^- and v_k^- using

$$z_k^- = z_{k-1}^+ + (v_{k-1}^+ * dt)$$

$$v_k^- = v_{k-1}^+ + (am_{k-1} - b_{k-1}) * dt)$$

The predicted (a priori) state covariance estimate P_k^- is

$$P_k^- = F_{k-1} * P_{k-1}^+ * F_{k-1}^T + Q_{k-1}$$

2.5 Update

We update the predicted state estimate $\hat{x_k^-}$ using the new measurement m_k to generate the best estimate $\hat{x_k^+}$ at time sample k.

The predicted measurement$\hat{m_k}$ is

$$\hat{m_k} = H_k * \hat{x_k^-}$$

The innovation (or error) $\tilde{y_k}$ is the difference between the new measurement m_k and the predicted measurement $\hat{m_k}$:

$$\tilde{y_k} = m_k - \hat{m_k}$$

The measurement noise covariance R_k is

$$R_k = E(\nu_k * \nu_k^T)$$

In this case R_k is a scalar $= \sigma_{zm}^2$ where σ_{zm}^2 is the statistical variance of the altitude sensor noise distribution.

The innovation covariance S_k is

$$S_k = H_k * P_k^- * H_k^T + R_k$$

$$S_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} P_k^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \sigma_{zm}^2$$

$$S_k = P_{zz} + \sigma_{zm}^2$$

In this model, S_k is also a scalar variable.

The updated (a posteriori) best state estimate $\hat{x_k^+}$ is

$$\hat{x_k^+} = \hat{x_k^-} + K_k * \tilde{y_k}$$

where K_k is the [3x1] Kalman gain matrix calculated as a ratio of the current state uncertainty mapped to measurement space, and measurement uncertainty S_k.

$$K_k = (P_k^- * H_k^T) * S_k^{-1}$$

$$K_k = P_k^- \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \frac{1}{(P_{zz} + \sigma_{zm}^2)}$$

$$K_k = \frac{1}{(P_{zz} + \sigma_{zm}^2)} \begin{bmatrix} P_{zz} \\ P_{vz} \\ P_{bz} \end{bmatrix}$$

So the updated (a posteriori) best state estimate is a blend between the predicted state estimate and the new measurement depending on the relative uncertainty in predicted state estimate and uncertainty in the new measurement.

The updated (a posteriori) state covariance estimate P_k^+ is

$$P_k^+ = (I - K_k * H_k) * P_k^-$$

When the filter is working as expected, the process covariance $trace(P_k^+)$ should reduce as new measurements arrive.

3 KalmanFilter4 algorithm

The KalmanFilter3 algorithm uses sensor acceleration measurements in the prediction phase without including acceleration in the state transition matrix, or in the measurement model. The KalmanFilter4 algorithm explicitly includes acceleration in the process state vector, state transition and measurement model. This results in a 4x4 process covariance matrix and more complex computation.

3.1 Process Model : physical equations

k is the sample index

z_k is the altitude sample, in cm

v_k is the vertical velocity, in cm/s

a_k is the vertical acceleration, in cm/s^2

b_k is the vertical acceleration bias, in cm/s^2. This is the residual acceleration bias after accelerometer calibration and is assumed to drift unpredictably. The true vertical acceleration is the measured acceleration minus the acceleration bias.

These variables at time index k are related to the state variables at time $k-1$ by the equations

$$\begin{aligned} z_k &= z_{k-1} + dt * v_{k-1} + (dt^2/2) * (a_{k-1} - b_{k-1}) \\ v_k &= v_{k-1} + dt * (a_{k-1} - b_{k-1}) \\ a_k &= a_{k-1} + \eta_a \\ b_k &= b_{k-1} + \eta_b \end{aligned}$$

where η_a represents the random acceleration perturbation from the environment, and η_b represents the random acceleration bias noise input.

This is our physical process model.

3.2 Process Model : state transition representation

In vector/matrix notation, the process model can be written as

$$\begin{bmatrix} z_k \\ v_k \\ a_k \\ b_k \end{bmatrix} = \begin{bmatrix} 1 & dt & (dt^2/2) & (-dt^2/2) \\ 0 & 1 & dt & -dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} z_{k-1} \\ v_{k-1} \\ a_{k-1} \\ b_{k-1} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_a \\ \eta_b \end{bmatrix}$$

or

$$x_k = F_k * x_{k-1} + L_k * w_k$$

where x is the process model state vector, F is the state transition matrix, w is the process model noise vector, and L is the process model noise sensitivity matrix.

The process state covariance matrix $P_k = E(x_k * x_k^T)$

$$P_k = \begin{bmatrix} P_{zz} & P_{zv} & P_{za} & P_{zb} \\ P_{vz} & P_{vv} & P_{va} & P_{vb} \\ P_{az} & P_{av} & P_{aa} & P_{ab} \\ P_{bz} & P_{bv} & P_{ba} & P_{bb} \end{bmatrix}$$

Note that P_k is a symmetric matrix.

3.3 Sensor Measurements

In this model, sensors provide periodic measurements for altitude zm and vertical acceleration am. These measurements are assumed to be corrupted by altitude sensor noise ν_z and acceleration sensor noise ν_a which have zero-mean Gaussian probability distributions.

The sensor measurements are related to the process model state variables by the equations

$$zm_k = z_k + \nu_z$$
$$am_k = a_k + \nu_a$$

In vector/matrix notation

$$\begin{bmatrix} zm_k \\ am_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} z_k \\ v_k \\ a_k \\ b_k \end{bmatrix} + \begin{bmatrix} \nu_z \\ \nu_a \end{bmatrix}$$

or

$$m_k = H_k * x_k + \nu_k$$

where m_k is the measurement vector, H_k is the measurement model matrix transforming the process state vector to the measurement vector space, and ν_k is the measurement noise vector.

3.4 Prediction

Assuming the process noise w components can be described by zero-mean gaussian probability distributions, the process noise covariance matrix Q_k is

$$Q_k = L * E(w_k * w_k^T) * L^T$$

or

$$Q_k = L * N_k * L^T$$

where

$$N_k = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix}$$

where σ_a^2 is the statistical variance of the random acceleration perturbations, and σ_b^2 is the statistical variance of the random acceleration bias noise.

$$Q_k = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_a^2 & 0 \\ 0 & 0 & 0 & \sigma_b^2 \end{bmatrix}$$

The predicted (a priori) state estimate $\hat{x_k^-}$ (assuming process model noise is additive and zero-mean) is

$$\hat{x_k^-} = F_{k-1} * \hat{x_{k-1}^+}$$

The predicted (a priori) state covariance estimate P_k^- is

$$P_k^- = F_{k-1} * P_{k-1}^+ * F_{k-1}^T + Q_{k-1}$$

3.5 Update

We update the predicted state estimate $\hat{x_k^-}$ using the new measurement m_k to generate the best estimate $\hat{x_k^+}$ at time sample k.

The predicted measurement $\hat{m_k}$ is

$$\hat{m_k} = H_k * \hat{x_k^-}$$

The innovation (or error) $\tilde{y_k}$ is the difference between the new measurement m_k and the predicted measurement $\hat{m_k}$:

$$\tilde{y_k} = m_k - \hat{m_k}$$

The measurement noise covariance R_k is

$$R_k = E(\nu_k * \nu_k^T)$$

$$R_k = \begin{bmatrix} \sigma_{zm}^2 & 0 \\ 0 & \sigma_{am}^2 \end{bmatrix}$$

where $\sigma_{zm}^2$is the statistical variance of the altitude sensor noise, and σ_{am}^2 is the statistical variance of the acceleration sensor noise.

The innovation covariance S_k is

$$S_k = H_k * P_k^- * H_k^T + R_k$$

The updated (a posteriori) state estimate $\hat{x_k^+}$ is

$$\hat{x_k^+} = \hat{x_k^-} + K_k * \tilde{y_k}$$

where K_k is the [4x2] Kalman gain matrix. K_k is the ratio of the current state uncertainty mapped to measurement space, and measurement uncertainty

$$K_k = (P_k^- * H_k^T) * S_k^{-1}$$

So the updated (a posteriori) best estimate x_k^+ is a blend between the predicted state estimate x_k^- and the new measurement m_k, based on the relative uncertainty in state estimate and uncertainty in the new measurement.

The updated (a posteriori) state covariance estimate P_k^+ is

$$P_k^+ = (I - K_k * H_k) * P_k^-$$

When the filter is working as expected, the process covariance $trace(P_k^+)$ should reduce as new measurements arrive.

4 KF3 versus KF4

4.1 Computation time

On an ESP32 micro clocked at 80MHz, the computation times are :

KalmanFilter3 Predict +Update	~45uS
KalmanFilter4 Predict + Update	~85uS

4.1.1 Real-time constraint

We are sampling raw data from the MPU9250 IMU sensor at 500Hz. All additional tasks between reading IMU samples need to be completed well within 2mS. In the worst-case, this is all of the following :

1. Barometric pressure sensor sampling, compensation and pressure-to-altitude conversion (BMP388 = 145uS, MS5611 takes less time)

2. AHRS algorithm to compute orientation and gravity-compensated acceleration (19uS)

3. Kalman filter prediction and update steps (KF4 = 85uS)

4. Data logging to SPI flash. Worst case is IMU+Baro+GPS 80byte record, crossing page boundary (210 uS)

The worst-case total execution time works out to 459uS. Note that ESP32 is a dual-core microcontroller and other application tasks (UI, GPS, Bluetooth) are executing on a different core. So using the KF4 algorithm is not an issue in this execution environment.

4.2 Convergence

A high-speed IBG (IMU+Baro+GPS) data log (using an MS5611 barometric sensor) was captured and then downloaded from the vario. This has IMU sensor data at 500 samples/sec and Baro sensor derived altitude data at 50 samples/sec.

KF3 and KF4 algorithms were configured with identical common parameters (Acceleration Bias Variance = 0.005, Acceleration Variance = 90000.0f, Initial Bias = 0.0, Altitude Sensor Variance = 200.0, Initial Pzz = 400.0, Initial Pvv = 400.0).

After the first 512 filter output samples, KF3 and KF4 state uncertainties Pzz for altitude and Pvv for velocity had converged to :

	Pzz	Pvv
KF3	16.7 cm^2	477.3 $(cm/s)^2$
KF4	14.3 cm^2	299.7 $(cm/s)^2$

KF4 demonstrates a significant reduction in climb/sink rate uncertainty compared to KF3. This is perceptible in the variometer audio feedback as well.