# GPS-aided INS Solution for OpenPilot

**Dale E. Schinstock**
Kansas State University

## NOMENCLATURE

SYSTEM MODEL:

$\vec{f}(\vec{x},\vec{u},\vec{w})$ - nonlinear state equations

$\vec{h}(\vec{x})$ - nonlinear measurement equations

$\mathbf{F}$ - system matrix from linearization

$\mathbf{G}$ - input matrix from linearization

$\mathbf{H}$ - measurement matrix from linearization

$\vec{x}$ - state vector

$\vec{P} = \{P_x \quad P_y \quad P_z\}^T$ - position vector in the NED (North-East-Down) Earth fixed frame

$\vec{V} = \{V_x \quad V_y \quad V_z\}^T$ - velocity vector in the NED earth fixed frame

$\vec{q} = \{q_0 \quad q_1 \quad q_2 \quad q_3\}^T$ - attitude vector as a unit quaternion

$\vec{b}_\omega = \{b_{\omega x} \quad b_{\omega y} \quad b_{\omega z}\}^T$ - rate gyro bias vector

$g$ - Earth's gravitational acceleration

$\vec{u} = \{\vec{\omega}_m \quad \vec{a}_m\}$ - input vector

$\vec{a} = \{a_x \quad a_y \quad a_z\}^T$ - true acceleration vector in the body fixed frame

$\vec{\omega} = \{\omega_x \quad \omega_y \quad \omega_z\}^T$ - true rotational rates vector in the body fixed frame

$\vec{w}_a$ - acceleration sensor noise vector

$\vec{w}_\omega$ - angular rate sensor noise vector

$\vec{w}_b$ - noise vector for bias random walks

$\vec{w} = \{\vec{w}_\omega{}^T \quad \vec{w}_a{}^T \quad \vec{w}_b{}^T\}^T$ - process noise vector

$\vec{v}^T$ - measurement noise vector

$\vec{a}_m = \{a_{mx} \quad a_{my} \quad a_{mz}\}^T$ - measured acceleration vector in the body fixed frame

$\vec{\omega}_m = \{\omega_{mx} \quad \omega_{my} \quad \omega_{mz}\}^T$ - measured rotational rate vector in the body fixed frame

$\mathbf{R}_{eb}$ - rotation matrix rotating vectors in the body fixed frame to the earth fixed frame

$\mathbf{\Omega}$ - matrix responsible for converting angular rates in the body fixed frame to quaternion rates

$A_b$ - barometric altitude measurement

$B_e$ - magnetic vector in Earth frame

$B_b$ - magnetic vector measurement in body frame

$\vec{z}$ - measurement vector

$\vec{y}$ - prediction of measurements from the state vector

EXTENDED KALMAN FILTER IMPLEMENTATION:

$\vec{\chi}_k$ - discrete time version of a vector $\vec{\chi}$

$\mathbf{\Phi}$ - state transition matrix

$\mathbf{\Gamma}$ - discrete time input matrix

$T$ - period of prediction step

$\mathbf{Q} = diag(\sigma^2_{\omega_x}, \sigma^2_{\omega_y}, \sigma^2_{\omega_z}, \sigma^2_{a_x}, \sigma^2_{a_y}, \sigma^2_{a_z}, \sigma^2_{b_x}, \sigma^2_{b_y}, \sigma^2_{b_z})$ - plant/disturbance noise covariance matrix

$\mathbf{P}_k$ - state estimate error covariance matrix

$\mathbf{K}$ - Kalman gain matrix

$\mathbf{R} = diag(\sigma^2_{P_x}, \sigma^2_{P_y}, \sigma^2_{P_z}, \sigma^2_{V_x}, \sigma^2_{V_y}, \sigma^2_{V_z}, \sigma^2_{B_x}, \sigma^2_{B_y}, \sigma^2_{B_z}, \sigma^2_{Alt})$ - measurement noise covariance matrix

## SYSTEM MODEL

The dynamic system model developed here is a kinematic model for a six DOF rigid body with position and velocity represented in an inertial coordinate frame (Earth fixed) and angular velocity and acceleration in a body fixed frame. Because it is a kinematic model, it is applicable to any vehicle, independent of the specific dynamics of that vehicle. In addition to the dynamic state variables, the state vector also includes bias states for sensors, which are modeled dynamically as simple random walks.

The development of the system model will make use of two matrices that are stated here without derivation. The first of these matrices is the rotation matrix as a function of the unit quaternion.

$$\mathbf{R}_{be}(\vec{q}) = \begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix} \quad (1)$$

where

$$\begin{aligned}
R_{1,1} &= q_0{}^2 + q_1{}^2 - q_2{}^2 - q_3{}^2 \\
R_{1,2} &= 2(q_1 q_2 + q_0 q_3) \\
R_{1,3} &= 2(q_1 q_3 - q_0 q_2) \\
R_{2,1} &= 2(q_1 q_2 - q_0 q_3) \\
R_{2,2} &= q_0{}^2 - q_1{}^2 + q_2{}^2 - q_3{}^2 \\
R_{2,3} &= 2(q_2 q_3 + q_0 q_1) \\
R_{3,1} &= 2(q_1 q_3 + q_0 q_2) \\
R_{3,2} &= 2(q_2 q_3 - q_0 q_1) \\
R_{3,3} &= q_0{}^2 - q_1{}^2 - q_2{}^2 + q_3{}^2
\end{aligned}$$

It is worthwhile to note that the inverse of the rotation matrix is its transpose, $\mathbf{R}_{be}{}^{-1} = \mathbf{R}_{eb} = \mathbf{R}_{be}{}^T$. The second of these matrices is used in the "strapdown" equation, $\dot{\vec{q}} = \mathbf{\Omega}(\vec{q})\vec{\omega}/2$, to relate the body axis angular velocity to the unit quaternion rates.

$$\mathbf{\Omega}(q)=\begin{bmatrix}-q_1 & -q_2 & -q_3\\ q_0 & -q_3 & q_2\\ q_3 & q_0 & -q_1\\ -q_2 & q_1 & q_0\end{bmatrix} \tag{2}$$

STATE VARIABLES

The state variables estimated by the INS/GPS system are the position, velocity, attitude (unit quaternion), and rate gyro biases.

$$\vec{x}=\begin{Bmatrix}\vec{P}\\ \vec{V}\\ \vec{q}\\ \vec{b}_\omega\end{Bmatrix} \tag{3}$$

INPUTS

The true inputs to the dynamic system, i.e. the kinematic system, are the true angular velocity and acceleration vectors. However, we are modeling the INS as dynamic system, which uses measurements of these vectors that include noise, biases, and gravity.

$$\vec{u}=\begin{Bmatrix}\vec{\omega}_m\\ \vec{a}_m\end{Bmatrix}=\begin{Bmatrix}\vec{\omega}-\vec{w}_\omega+\vec{b}_\omega\\ \vec{a}-\vec{w}_a-\mathbf{R}_{be}\{0 \quad 0 \quad g\}^T\end{Bmatrix} \tag{4}$$

Here, the accelerometer measurement includes earth's gravitational acceleration rotated into the body fixed frame with $\mathbf{R}_{be}$. Also, without loss of generality, the sensor noise is shown here as subtractive rather than additive. This is done purely for convenience in the state equations developed below, where it will become additive.

STATE EQUATIONS

The state equations are the derivatives of the state variables. In general, these equations are nonlinear functions of the state variables and inputs.

$$\dot{\vec{x}}=\begin{Bmatrix}\dot{\vec{P}}\\ \dot{\vec{V}}\\ \dot{\vec{q}}\\ \dot{\vec{b}}_\omega\end{Bmatrix}=\begin{Bmatrix}\vec{V}\\ \mathbf{R}_{eb}\vec{a}\\ \frac{1}{2}\mathbf{\Omega}\vec{\omega}\\ \vec{w}_b\end{Bmatrix} \tag{5}$$

The first three vector equations in (5) are kinematic equations for a six DOF rigid body. The last vector equation, for the bias states, is a random walk used to model in a simple manner the dynamics of states that vary slowly in a random way .

While (5) captures the kinematics of a rigid body, it is not in the proper form for state equations. State equations should be written as a function of the states, inputs, and process/disturbance noise.

$$\dot{\vec{x}}=\vec{f}(\vec{x},\vec{u},\vec{w})$$

This can be accomplished by solving (4) for the true angular velocity and acceleration.

$$\vec{\omega}=\vec{\omega}_m+\vec{w}_\omega-\vec{b}_\omega$$
$$\vec{a}=\vec{a}_m+\vec{w}_a+\mathbf{R}_{be}\{0 \quad 0 \quad g\}^T$$

Then plugging these into (5) gives the state equations as a function of the states, the inputs $\vec{u}=\{\vec{\omega}_m{}^T \quad \vec{a}_m{}^T\}^T$, and the process noise $\vec{w}=\{\vec{w}_m{}^T \quad \vec{w}_a{}^T \quad \vec{w}_b{}^T\}^T$.

$$\vec{f}(\vec{x},\vec{u},\vec{w})=\begin{Bmatrix}\dot{\vec{P}}\\ \dot{\vec{V}}\\ \dot{\vec{q}}\\ \dot{\vec{b}}_\omega\end{Bmatrix}=\begin{Bmatrix}\vec{V}\\ \mathbf{R}_{eb}(\vec{q})\cdot(\vec{a}_m+\vec{w}_a)+\{0 \quad 0 \quad g\}^T\\ \frac{1}{2}\mathbf{\Omega}(\vec{q})\cdot(\vec{\omega}_m+\vec{w}_\omega-\vec{b}_\omega)\\ \vec{w}_b\end{Bmatrix} \tag{6}$$

OUTPUTS

The outputs of the dynamic system model are variables that are measured by sensors to be used in the correction steps. They must be formed as a function of the state variables.

$$\vec{y}=\vec{h}(\vec{x})=\begin{Bmatrix}\vec{P}\\ \vec{V}\\ \vec{B}_b\\ A_b\end{Bmatrix}=\begin{Bmatrix}\vec{P}\\ \vec{V}\\ \mathbf{R}_{be}(\vec{q})\vec{B}_e\\ -P_z\end{Bmatrix} \tag{7}$$

Here, the magnetic field in the body frame is expressed as function of the constant magnetic field in the Earth frame and a rotation matrix that is a function of the quaternion. The barometric altimeter output is simply the negative of the down component from the position vector. There is a measurement vector from the sensors corresponding to the output vector with the addition of sensor noise.

$$\vec{z}=\vec{y}+\vec{v}$$

The difference between these two vectors, $\vec{z}-\vec{y}$, is used in the feedback of the correction steps to correct the states predicted in the prediction steps through numerical integration.

LINEARIZATION

To implement the EKF it is necessary to linearize the state equations at each calculation of the prediction step and to linearize the output equations at each calculation of the so correction step so that the KF equations can be used. This linearization results in equations for the following Jacobian matrices,

$$\mathbf{F} = \frac{\partial \vec{f}}{\partial \vec{x}}$$
$$\mathbf{G} = \frac{\partial \vec{f}}{\partial \vec{w}}$$
$$\mathbf{H} = \frac{\partial \vec{h}}{\partial \vec{x}}$$

Calculating the partial derivatives in the elements of $\mathbf{F}$ gives

$$\mathbf{F} = \left[\begin{array}{c|c} \begin{array}{c|c} \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \end{array} & \mathbf{0}_{3x7} \\ \hline \mathbf{0}_{10x6} & \begin{array}{c|c} \mathbf{F}_{Vq} & \mathbf{0}_{3x3} \\ \hline \mathbf{F}_{qq} & \mathbf{F}_{qb} \\ \hline \multicolumn{2}{c}{\mathbf{0}_{3x7}} \end{array} \end{array}\right] \quad (8)$$

where

$$\mathbf{F}_{Vq} = \begin{bmatrix} F_{Vq0} & F_{Vq1} & F_{Vq2} & F_{Vq3} \\ -F_{Vq3} & -F_{Vq2} & F_{Vq1} & F_{Vq0} \\ F_{Vq2} & -F_{Vq3} & -F_{Vq0} & F_{Vq1} \end{bmatrix}$$

$$F_{Vq0} = 2(q_0 a_{mx} - q_3 a_{my} + q_2 a_{mz})$$
$$F_{Vq1} = 2(q_1 a_{mx} + q_2 a_{my} + q_3 a_{mz})$$
$$F_{Vq2} = 2(-q_2 a_{mx} + q_1 a_{my} + q_0 a_{mz})$$
$$F_{Vq3} = 2(-q_3 a_{mx} - q_0 a_{my} + q_1 a_{mz})$$

$$\mathbf{F}_{qq} = \frac{1}{2}\begin{bmatrix} 0 & b_{\omega x} - \omega_{mx} & b_{\omega y} - \omega_{my} & b_{\omega z} - \omega_{mz} \\ \omega_x - b_{\omega x} & 0 & \omega_z - b_{\omega z} & b_{\omega y} - \omega_{my} \\ \omega_y - b_{\omega y} & b_{\omega z} - \omega_{mz} & 0 & \omega_x - b_{\omega x} \\ \omega_z - b_{\omega z} & \omega_y - b_{\omega y} & b_{\omega x} - \omega_{mx} & 0 \end{bmatrix}$$

and

$$\mathbf{F}_{qb} = -\frac{1}{2}\mathbf{\Omega}(\vec{q})$$

Calculating the partial derivatives in the elements of $\mathbf{G}$ gives

$$\mathbf{G} = \left[\begin{array}{c|c} \multicolumn{2}{c}{\mathbf{0}_{3x9}} \\ \hline \begin{array}{c|c} \mathbf{0}_{3x3} & \mathbf{R}_{eb}(\vec{q}) \\ \hline \mathbf{\Omega}(\vec{q})/2 & \mathbf{0}_{4x3} \end{array} & \mathbf{0}_{7x3} \\ \hline \mathbf{0}_{3x6} & \mathbf{I}_{3x3} \end{array}\right] \quad (9)$$

Calculating the partial derivatives in the elements of $\mathbf{H}$ gives

$$\mathbf{H} = \left[\begin{array}{c|c} \mathbf{I}_{6x6} & \mathbf{0}_{6x7} \\ \hline \mathbf{0}_{3x6} & \begin{array}{c|c} \mathbf{H}_{Bq} & \mathbf{0}_{3x3} \end{array} \\ \hline \begin{array}{ccc|c} 0 & 0 & -1 & \mathbf{0}_{1x3} \end{array} & \mathbf{0}_{1x7} \end{array}\right] \quad (10)$$

where

$$\mathbf{H}_{Bq} = \begin{bmatrix} H_{Bq0} & H_{Bq1} & H_{Bq2} & H_{Bq3} \\ H_{Bq3} & -H_{Bq2} & H_{Bq1} & -H_{Bq0} \\ -H_{Bq2} & -H_{Bq3} & H_{Bq0} & H_{Bq1} \end{bmatrix}$$

$$H_{Bq0} = 2(q_0 B_{ex} + q_3 B_{ey} - q_2 B_{ez})$$
$$H_{Bq1} = 2(q_1 B_{ex} + q_2 B_{ey} + q_3 B_{ez})$$
$$H_{Bq2} = 2(-q_2 B_{ex} + q_1 B_{ey} - q_0 B_{ez})$$
$$H_{Bq3} = 2(-q_3 B_{ex} + q_0 B_{ey} + q_1 B_{ez})$$

## EXTENDED KALMAN FILTER IMPLEMENTATION

A general description of the EKF and KF are beyond the scope of this discussion. However, a couple of details about the specific implementation here might be important in relating the algorithms to more general EKF equations found in many sources. The KF algorithm is ultimately a discrete time algorithm. At its root it is based on a very simple discrete model of the system.

$$\begin{aligned} \vec{x}_k &= \mathbf{\Phi}\vec{x}_{k-1} + \mathbf{\Gamma}\vec{w}_{k-1} \\ \vec{z}_k &= \mathbf{H}\vec{x}_k + \vec{v}_k \end{aligned} \quad (31)$$

The discrete time disturbance/process noise vector, $\vec{w}_k$, is assumed to be white and to have a noise covariance matrix, $\mathbf{Q}$. The measurement noise, $\vec{v}_k$, is assumed to be white and have a noise covariance matrix, $\mathbf{R}$. The fact that these are discrete time noise processes does not present a problem because the sensors are ultimately sampled in discrete time. In fact this is a benefit because it is possible to directly estimate the noise variance from data samples. The implemented computer algorithm does make the simplifying assumption that these matrices are diagonal, i.e. the noise on each of the sensors is independent.

Our model is a continuous time model. Therefore, we must therefore use some discrete time approximations for implementation. We use the following first order approximations.

$$\mathbf{\Phi} \cong \mathbf{I} + \mathbf{F}T$$
$$\mathbf{\Gamma} \cong \mathbf{G}T$$

The INS numerically integrates the inputs, i.e. the acceleration and angular rate measurements, to obtain estimates of position, velocity and orientation. The prediction step of the EKF uses the output of the INS, and also predicts the growth of covariance of the state estimate error. This covariance is a running approximation of the our confidence in the estimated state. The size of this covariance, and hopefully the true error in the state estimate, is reduced in the correction steps where we incorporate other sensors in a feedback loop.

## INS/GPS ALGORITHM

PREDICTION STEP

$$\vec{x}_k = \vec{x}_{k-1} + \int_{t_{k-1}}^{t_k} \vec{f}(\vec{x}_{k-1}, \vec{u}_{k-1})dt \quad \text{(p1)}$$

$$\textit{normalize quaternion} \quad \text{(p2)}$$

$$\textit{calculate } \mathbf{F} \textit{ and } \mathbf{G} \quad \text{(p3)}$$

$$\mathbf{P}_k = (\mathbf{I} + \mathbf{F}\mathrm{T})\mathbf{P}_{k-1}(\mathbf{I} + \mathbf{F}\mathrm{T})^{\mathrm{T}} + \mathrm{T}^2\mathbf{GQG}^T \quad \text{(p4)}$$

CORRECTION/UPDATE STEP

$$\textit{calculate } \mathbf{H} \quad \text{(c1)}$$

$$\textit{calculate } \vec{y}_k \quad \text{(c2)}$$

$$\textit{serial update} \quad \text{(c3)}$$

$$\mathbf{K} = \mathbf{P}_k\mathbf{H}^T(\mathbf{HP}_k\mathbf{H}^T + \mathbf{R})^{-1}$$

$$\vec{x}_k = \vec{x}_k + \mathbf{K}(\vec{z}_k - \vec{y}_k)$$

$$\mathbf{P}_k = \mathbf{P}_k - \mathbf{KHP}_k$$

$$\textit{normalize quaternion} \quad \text{(c4)}$$

Step (p1) is completed in `RungeKutta(X,U,dT).` It implements a numerical integration with a fourth order Runga Kutta algorithm through function calls to `StateEq(X,U,Xdot),` which implements (6).

Step (p3) is completed in `LinearizeFG(X,U,F,G).` It implements (8) and (9)

Step (p4) is completed in `CovariancePrediction(F,G,Q,dT,P).` It estimates the growth in the covariance of the state estimate error due to the process noise.

Step (c1) is completed in `LinearizeH(X,Be,H).` It implements (10) .

Step (c2) is completed in `MeasurementEq(X,Be,Y).` It implements (7) .

Step (c3) is completed in `SerialUpdate(H,R,Z,Y,P,X,SensorsUsed).` While it implements the equivalent of the equations shown in this step, it does so with a very different algorithm. The equations are implemented in a serial update algorithm treating each scalar measurement separately [1, ch 4.2][2, ch 4.5]. This avoids finding the matrix inverse by replacing it with scalar divisions. It is computationally efficient and numerically stable. Furthermore, it allows any chosen subset of sensors to be used in a single correction step. The use of this serial update algorithm is possible because the noise for each of the measurements is assumed to be uncorrelated, i.e. the covariance matrix, $\mathbf{R}$, is diagonal.

## REFERENCES


[1] Grewal, M.S., A.P. Andrews, *Kalman Filtering, Theory and Practice Using MATLAB,* John Wiley and Sons, Inc., 2001

[2] Farrell, J.A., M. Barth, *The Global Positioning System & Inertial Navigation,* McGraw Hill, 1999