Error State Multiplicative EKF

Redefining notation

Body frame → rotates with drone, nose is x, right wing y, down is z.
Inertial fixed frame (Earth) with NED corresponding to x,y,z here.
For vec $\vec{v}$, $\vec{v}^i$ in intertial frame, $\vec{v}^b$ body frame.

For quaternion $\vec{q} = \begin{Bmatrix} q_1 \\ \vec{q}_{2:4} \end{Bmatrix}$. Quaternion product can be rewritten as

$$\vec{p} \otimes \vec{q} = \begin{bmatrix} p_1 q_1 - \vec{p}_{2:4} \overset{\text{dot product}}{\cdot} \vec{q}_{2:4} \\ p_1 q_{2:4} + q_1 p_{2:4} + p_{2:4} \times q_{2:4} \end{bmatrix}$$

Inverse $\vec{q}^{-1} = \begin{bmatrix} q_1 \\ -\vec{q}_{2:4} \end{bmatrix}$. Identity $= \begin{bmatrix} 1 \\ \vec{0} \end{bmatrix}$. So for vector $\vec{v}^b$ in body frame $\begin{bmatrix} 0 \\ \vec{v}^b \end{bmatrix} = \vec{q}^{-1} \otimes \begin{bmatrix} 0 \\ \vec{v}^i \end{bmatrix} \otimes \vec{q}$. Same can be done with $\vec{v}^b = C_i^b(\vec{q})\vec{v}^i$, C_i^b from i frame to b frame, same as R before (sub/superscripts may be inconsistent).

$$C_i^b(\vec{q}) = 2\vec{q}_{2:4} \vec{q}_{2:4}^{\,T} + I_3(q_1^2 - \vec{q}_{2:4}^{\,T} \vec{q}_{2:4}) - 2q_1 [q_{2:4} \times].$$

Also, for $\vec{v} \in V$, $\dim(V) = 3$, then

$$[\vec{v}\times] = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}$$

skew symmetric matrix so that $[\vec{v}\times]\vec{w} = \vec{v} \times \vec{w} \quad \forall \vec{v}, \vec{w}$.

Updated Kalman Filter Setup

Previously we had, gyro as control, accelerometer and magnetometer as measurements/correction. This only works for stationary objects as accelerometer only measures gravity, but in case of movement, can't construct measurement function because to estimate $\vec{q}$, we would need to know actual acceleration of drone but that is itself being measured from accelerometer.

As a result, can have both accelerometer and gyro as control and other sensors as measurement/correction. Can have all sensors as correction, but this way more common in literature, and makes sense since gryo + accelerometer have highest rate of data (IMU has most frequent measurements).

ES MEKF Setup

In the ES MEKF, rather than our EKF estimating our desired values (e.g. attitude, velocity, position) in it's state, in the EKF we estimate the errors in a naive estimation of our state from our dynamics equation. E.g. we would naively integrate $\vec{q}$ from our $\dot{q}$ from our gyro input, and have $\delta\vec{q}$ in our EKF.

System Dynamics

We want to estimate $\vec{r}^i, \vec{v}^i, \vec{q}$ displacement, velocity, and quaternion for attitude. We get $\vec{\omega}_{ib}^b$, body-fixed coordinates of angular velocity of body frame w.r.t. the inertial frame from gyro and $\vec{f}^b$ acceleration from accelerometer. We have the following system dynamics equations

$$\dot{\vec{q}} = \frac{1}{2}\vec{q} \otimes \begin{bmatrix} 0 \\ \omega_{ib}^b \end{bmatrix} \quad \dots \text{①}$$

$$\dot{\vec{v}}^b = C_b^i(\vec{q})\vec{f}^b + \vec{g}^i \quad \dots \text{②}$$

For ①,

$$\dot{\vec{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\vec{\omega}_{ib}^{bT} \\ \vec{\omega}_{ib}^b & -[\vec{\omega}_{ib}^b \times] \end{bmatrix}\vec{q} \quad \because \text{rewrite quaternion mult matrix}$$

$$:= \frac{1}{2}\Omega(\vec{\omega}_{ib}^b)\vec{q}$$

This is a linear equation hence recalling notes on the multivariate KF, to go from a dynamics system to state extrapolation, we get

$$\vec{q}_k = \exp\left(\frac{1}{2}\Omega(\bar{\vec{w}})\Delta t\right)\vec{q}_{k-1}$$

where

$$\bar{\vec{w}} = \frac{\omega_k + \omega_{k-1}}{2} \qquad \ldots (4)$$

Incremental rotation vec can be approximated as $\vec{\sigma} = \frac{1}{2}\bar{\vec{w}}\,\Delta t$. The general solution to

$$\exp(\Omega(\vec{\sigma})) = \cos(\|\vec{\sigma}\|)I_4 + \frac{\sin(\|\vec{\sigma}\|)}{\|\vec{\sigma}\|}\Omega(\vec{\sigma}) \qquad \text{(proven in appendix)}$$

Hence,

$$\begin{aligned}\vec{q}_k &= \exp\left(\frac{1}{2}\Omega(\bar{\vec{w}})\Delta t\right)\vec{q}_{k-1} \\ &= \exp\left(\Omega\left(\frac{1}{2}\Delta t\bar{\vec{w}}\right)\right)\vec{q}_{k-1} \\ &= \left(\cos\left(\frac{\Delta t}{2}\|\bar{\vec{w}}\|\right)I_4 + \frac{\sin\left(\frac{1}{2}\Delta t\|\bar{\vec{w}}\|\right)}{\cancel{\frac{1}{2}\Delta t}\|\bar{\vec{w}}\|}\cancel{\frac{1}{2}\Delta t}\,\Omega(\bar{\vec{w}})\right)\vec{q}_{k-1}\end{aligned}$$

$$\vec{q}_k = \left(\cos\left(\frac{\Delta t}{2}\|\bar{\vec{w}}\|\right)I_4 + \frac{\sin\left(\frac{\Delta t}{2}\|\bar{\vec{w}}\|\right)}{\|\bar{\vec{w}}\|}\Omega(\bar{\vec{w}})\right)\vec{q}_{k-1} \qquad \ldots (5)$$

Now for velocity, we can integrate acceleration (trapezoidal scheme chosen here).

$$\vec{v}_k = \frac{\Delta t}{2}\left(C_b^i(\vec{q}_k)\vec{f}_k^{\,b} + C_b^i(\vec{q}_{k-1})\vec{f}_{k-1}^{\,b}\right) + \vec{v}_{k-1} \qquad \ldots (6)$$

and position can similarly be integrated.

$$r_k^i = \frac{\Delta t}{2}\left(v_k^i + v_{k-1}^i\right) + r_{k-1}^i \qquad \ldots (7)$$

So now on each predict step, we can compute (4), (5), (6), (7) from w_k and f_k and now in the EKF, we can have the errors for each. Then on each measurement we can compute changes in errors and then update our nominal state. Note for trapezoidal integration, we have to compute $\vec{q}_k$ first.

Error State Formulation

IMU outputs have errors $\omega = \hat{\omega} + \delta\omega$, $f = \hat{f} + \delta f$ noise (dropping subscripts and super scripts from now on). $\delta\omega$, δf random noise here. For quaternion q, we have small error quaternion δq for estimate $\hat{q}$ so

$$q = \hat{q} \otimes \delta q \Leftrightarrow \delta q = \hat{q}^{-1} \otimes q$$

Also,

$$C_i^b(q) = C_i^b(\hat{q} \otimes \delta q) = C_i^b(\delta q)\, C_i^b(\hat{q})$$

Now by differentiating $\delta q = \hat{q}^{-1} \otimes q$, we get

$$\delta\dot{q} = \hat{q}^{-1} \otimes \dot{q} + \dot{\hat{q}}^{-1} \otimes q$$

and we can get an identity for $\dot{\hat{q}}^{-1}$ by differentiating $\begin{bmatrix} 1 \\ \vec{0} \end{bmatrix} = \hat{q}^{-1} \otimes \hat{q}$

$$\frac{d}{dt}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{d}{dt}\left[\hat{q}^{-1} \otimes \hat{q}\right]$$

$$\Rightarrow \vec{0} = \dot{\hat{q}}^{-1} \otimes \hat{q} + \hat{q}^{-1} \otimes \dot{\hat{q}}$$

$$\Rightarrow -\dot{\hat{q}}^{-1} \otimes \hat{q} = \hat{q}^{-1} \otimes \dot{\hat{q}}$$

$$\Rightarrow \dot{\hat{q}}^{-1} = -\hat{q}^{-1} \otimes \dot{\hat{q}} \otimes \hat{q}^{-1}$$

$$\Rightarrow \dot{\hat{q}}^{-1} = -\hat{q}^{-1} \otimes \frac{1}{2}\hat{q} \otimes \begin{bmatrix} 0 \\ \hat{\omega}_{ib}^{b} \end{bmatrix} \otimes \hat{q}^{-1} \qquad \text{from } (1)$$

$$\Rightarrow \dot{\hat{q}}^{-1} = \frac{1}{2}\begin{bmatrix} 0 \\ \hat{\omega} \end{bmatrix} \otimes \hat{q}^{-1} \qquad \ldots (8)$$

So,

$$\delta\dot{q} = \hat{q}^{-1} \otimes \dot{q} + \dot{\hat{q}}^{-1} \otimes q$$

$$= \hat{q}^{-1} \otimes \frac{1}{2} q \begin{bmatrix} 0 \\ w \end{bmatrix} + \dot{\hat{q}}^{-1} \otimes q \qquad \text{from (1)}$$

$$= \hat{q}^{-1} \otimes \frac{1}{2} q \begin{bmatrix} 0 \\ w \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 0 \\ \hat{w} \end{bmatrix} \otimes \hat{q}^{-1} \otimes q \qquad \text{from (8)}$$

$$= \frac{1}{2} \delta q \otimes \begin{bmatrix} 0 \\ w \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 0 \\ \hat{w} \end{bmatrix} \otimes \delta q \qquad \because \delta q = \hat{q}^{-1} \otimes q$$

$$= \frac{1}{2} \delta q \otimes \begin{bmatrix} 0 \\ \hat{w} + \delta w \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 0 \\ \hat{w} \end{bmatrix} \otimes \delta q \qquad \because w = \hat{w} + \delta w$$

$$= \frac{1}{2} \delta q \otimes \begin{bmatrix} 0 \\ \hat{w} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 0 \\ \hat{w} \end{bmatrix} \otimes \delta q + \frac{1}{2} \delta q \otimes \begin{bmatrix} 0 \\ \delta w \end{bmatrix}$$

$\because$ quaternion mult not commutative

Now we make an assumption that δq has small magnitude so $\delta q_1 \approx 1$, hence we don't estimate it.

$$\Rightarrow \delta\dot{q} = \frac{1}{2} \begin{Bmatrix} -\delta q_{2:4} \cdot \hat{w} \\ \hat{w} + \delta q_{2:4} \times \hat{w} \end{Bmatrix} + \frac{1}{2} \begin{Bmatrix} \hat{w} \cdot \delta q_{2:4} \\ -\hat{w} - \hat{w} \times \delta q_{2:4} \end{Bmatrix} + \frac{1}{2} \begin{Bmatrix} -\delta q_{2:4} \cdot \delta\hat{w} \\ \delta\hat{w} + \delta q_{2:4} \times \delta\hat{w} \end{Bmatrix} \qquad \because \text{quat mult definition}$$

$$\Rightarrow \begin{Bmatrix} 0 \\ \delta\dot{q}_{2:4} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -\hat{w} \times \delta q_{2:4} \end{Bmatrix} + \frac{1}{2} \begin{Bmatrix} -\delta q_{2:4} \cdot \delta\hat{w} \\ \delta\hat{w} + \underbrace{\delta q_{2:4} \times \delta\hat{w}}_{\text{2nd order term}} \end{Bmatrix}$$

assuming magnitude not changing since $\delta q_1 \approx 1$

We further approximate by assuming $\delta q_{2:4}, \delta\hat{w}$ small, hence

$$\delta\dot{q}_{2:4} \approx -\hat{w} \times \delta q_{2:4} + \frac{1}{2} \delta\hat{w}$$

We now right $\alpha = 2\delta q_{2:4}$ ... (9), and get

$$\delta\dot{\alpha} \approx -\hat{w} \times \alpha + \delta w \qquad \dots (10)$$

We also get

$$C_i^b(\delta q) = 2\delta\vec{q}_{2:4}\delta\vec{q}_{2:4}^{\,T} + I_3(\delta q_1^2 - \delta\vec{q}_{2:4}^{\,T}\delta\vec{q}_{2:4}) - 2\delta q_1[\delta q_{2:4}\times].$$

Assuming 2nd order terms 0, $\delta q_1 \approx 1$, we get

$$C_i^b(q) = (I - [\alpha\times])C_i^b(\hat{q}) \quad \ldots (11)$$

$$C_b^i(q) = C_b^i(\hat{q})[I + [\alpha\times]] \quad \ldots (12)$$

Now for velocity and position error dynamics, we have

$$\delta v = v - \hat{v}$$
$$\delta r = r - \hat{r}$$
$$\delta\dot{v} = \dot{v} - \dot{\hat{v}}$$

$$\Rightarrow \delta\dot{v} = C_b^i(q)f^b + g - C_b^i(\hat{q})\hat{f}^b - \hat{g} \qquad \because (2)$$

$$\Rightarrow \delta\dot{v} = C_b^i(q)(\hat{f} + \delta f) - C_b^i(\hat{q})\hat{f} + \delta g \qquad \because f = \hat{f} + \delta f$$

$$\Rightarrow \delta\dot{v} = C_b^i(\hat{q})[I + [\alpha\times]](\hat{f} + \delta f) - C_b^i(\hat{q})\hat{f} + \delta g \qquad \because (12)$$

$$\Rightarrow \delta\dot{v} = C_b^i(\hat{q})[\alpha\times]\hat{f} + C_b^i(\hat{q})[I + [\alpha\times]](\delta f) + \delta g$$

$$\Rightarrow \delta\dot{v} = -C_b^i(\hat{q})[\hat{f}\times]\alpha + C_b^i(\hat{q})[I + [\alpha\times]](\delta f) + \delta g \qquad \because [\alpha\times]\hat{f} = -[\hat{f}\times]\alpha$$

cross product identity

$$\Rightarrow \delta\dot{v} = -C_b^i(\hat{q})[\hat{f}\times]\alpha + C_b^i(\hat{q})\delta f \quad \ldots (13) \qquad \because \text{assuming } \delta g \approx 0,\ C_b^i(\hat{q})[I + [\alpha\times]]\delta f \approx C_b^i(\hat{q})\delta f$$

with 2nd degree small term removed

Also finally $\delta\dot{r} = \dot{r} - \dot{\hat{r}} = v - \hat{v} = \delta v \quad \ldots (14)$.

Bias State Estimation

We assume bias only in gyro, accelerometer, magnetometer. All other sensors should be calibrated to remove biases and only have random white noise error.

Assume $\omega = \hat{\omega} - \beta_\omega - \eta_\omega$, $\dot{\beta}_\omega = \nu_\omega$, n_ω, ν_ω random noise with cov matrices

$$E[n_\omega(t) n_\omega^T(\tau)] = \begin{bmatrix} \sigma_\omega^2 & 0 & 0 \\ 0 & \sigma_\omega^2 & 0 \\ 0 & 0 & \sigma_\omega^2 \end{bmatrix} \delta(t-\tau)$$

Can be seperate values if var different for x, y, z axises

$$E[\nu_\omega(t) \nu_\omega^T(\tau)] = \begin{bmatrix} \sigma_{B\omega}^2 & 0 & 0 \\ 0 & \sigma_{B\omega}^2 & 0 \\ 0 & 0 & \sigma_{B\omega}^2 \end{bmatrix} \delta(t-\tau)$$

And $\delta\omega = -\beta_\omega - n_\omega$, $\therefore \delta\omega = \omega - \hat{\omega}$, $\delta\dot{\omega} = -\nu_\omega$.

Similarly, $f = \hat{f} - \beta_f - n_f$, $\dot{\beta}_f = \nu_f$, $\delta f = -\beta_f - n_f$, with n_f, ν_f random white noise, $E[n_f(t) n_f^T(\tau)] = \mathrm{diag}(\sigma_f^2)\delta(t-\tau)$, $E[\nu_f(t)\nu_f^T(\tau)] = \mathrm{diag}(\sigma_{B_f}^2)\delta(t-\tau)$.

Finally for magnetometer, bias term still used since some correlation expected between KF update equations. For β_m, model assumes it as a diverging process due to white noise, $\dot{\beta}_m = \nu_m$, ν_m white noise, $E[\nu_m(t)\nu_m^T(\tau)] = \mathrm{diag}(\sigma_{B_m}^2)\delta(t-\tau)$.

Predict Step Summary

1) Trapezoidal integration of v, r, and update q with ω.
2) EKF state $\delta x = \{\alpha^T \ \delta v^T \ \delta r^T \ \beta_\omega^T \ \beta_f^T \ \beta_m^T\}^T$ with

$$\begin{bmatrix} \dot{\alpha} \\ \delta\dot{v} \\ \delta\dot{r} \\ \dot{\beta}_\omega \\ \dot{\beta}_f \\ \dot{\beta}_m \end{bmatrix} = \underbrace{\begin{bmatrix} -[\hat{\omega}\times] & 0_3 & 0_3 & I_3 & 0_3 & 0_3 \\ -C_b^i(\hat{q})[\hat{f}\times] & 0_3 & 0_3 & 0_3 & -C_b^i(\hat{q}) & 0_3 \\ 0_3 & I_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ & & 0_{9\times18} & & & \end{bmatrix}}_{F(\hat{q},\hat{\omega},\hat{f})} \delta x + \begin{bmatrix} -n_\omega \\ -C_b^i(\hat{q})n_f \\ 0_{3\times1} \\ \nu_\omega \\ \nu_f \\ \nu_m \end{bmatrix}$$

3) Process noise which can be generated assuming F linearized at $\vec{\omega} = \vec{f} = 0$ and $C_i^b(\hat{q}) = I$. Works well in practice.

$$Q_d = \int_0^{\Delta t} e^{F(t-\tau)} Q_c \, e^{F^T(t-\tau)} \, d\tau$$

where for $\delta\dot{x} = F\,\delta x + Gu$, $E[(Gu)(Gu)^T] = Q_c\,\delta(t-\tau)$, hence for white noise, Q_c diagonal with

$$Q_c = \begin{bmatrix} \mathrm{diag}(\sigma_w^2) & & & & & \\ & \mathrm{diag}(\sigma_f^2) & & 0 & & \\ & & & & & \\ & & & \mathrm{diag}(\sigma_{Bw}^2) & & \\ & & & & \mathrm{diag}(\sigma_{Bf}^2) & \\ & & & & & \mathrm{diag}(\sigma_{Bm}^2) \end{bmatrix}$$

Some assumptions here for noise can be fixed but this should work well in practice. So,

$$Q_d = \begin{bmatrix} \Lambda(\sigma_w^2)\Delta t + \Lambda(\sigma_{Bw}^2)\frac{\Delta t^3}{3} & 0 & 0 & -\Lambda(\sigma_{Bw}^2)\frac{\Delta t^2}{2} & 0 & 0 \\ 0 & \Lambda(\sigma_f^2)\Delta t + \Lambda(\sigma_{Bf}^2)\frac{\Delta t^3}{3} & \Lambda(\sigma_{Bf}^2)\frac{\Delta t^4}{8} + \Lambda(\sigma_f^2)\frac{\Delta t^2}{2} & 0 & -\Lambda(\sigma_{Bf}^2)\frac{\Delta t^2}{2} & 0 \\ 0 & \Lambda(\sigma_f^2)\frac{\Delta t^2}{2} + \Lambda(\sigma_{Bf}^2)\frac{\Delta t^4}{8} & \Lambda(\sigma_f^2)\frac{\Delta t^3}{3} + \Lambda(\sigma_{Bf}^2)\frac{\Delta t^5}{20} & 0 & -\Lambda(\sigma_{Bf}^2)\frac{\Delta t^3}{6} & 0 \\ -\Lambda(\sigma_{Bw}^2)\frac{\Delta t^2}{2} & 0 & 0 & \Lambda(\sigma_{Bw}^2)\frac{\Delta t^2}{2} & 0 & 0 \\ 0 & -\Lambda(\sigma_{Bf}^2)\frac{\Delta t^2}{2} & -\Lambda(\sigma_{Bf}^2)\frac{\Delta t^3}{6} & 0 & \Lambda(\sigma_{Bf}^2)\Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & \Lambda(\sigma_{Bm}^2)\Delta t \end{bmatrix}$$

with tuning params σ_w^2, σ_f^2, σ_{Bw}^2, σ_{Bf}^2, σ_{Bm}^2

4) $\Phi = I + F\Delta t + \frac{1}{2}F^2\Delta t^2$, state transition function to calculate $P_{n+1|n} = \Phi P_{n|n} \Phi^T + Q_d$.

Measurement/Correction Step

1) Magnetometer: Measures $\tilde{m}^b = C_i^b(q) m^i + \beta_m + n_m$ (noise), m^i pulled from WMM. Can calculate this with magnetic dip θ_{dip} so $m^i = \begin{bmatrix} \cos\theta_{dip} \\ 0 \\ \sin\theta_{dip} \end{bmatrix}$

Our prediction for measurement is $\hat{m}^b = C_i^b(\hat{q}) m^i$. Hence

$$\tilde{m}^b - \hat{m}^b = C_i^b(q) m^i + \beta_m + n_m - C_i^b(\hat{q}) m^i$$

$$\Rightarrow \delta\tilde{m}^b = [I - [\alpha \times]] C_i^b(\hat{q}) m^i + \beta_m + n_m - C_i^b(\hat{q}) m^i$$

$$\Rightarrow \delta\tilde{m}^b = \left[[(C_i^b(\hat{q}) m^i)\times] \;\; I_3 \right] \begin{Bmatrix} \alpha \\ \beta_m \end{Bmatrix} + n_m$$

Rewrite $\delta\tilde{m}^b$ as innovation, $\begin{Bmatrix} \alpha \\ \beta_m \end{Bmatrix}$ as state and as h, we get

$$\text{innovation} = \delta\tilde{m}^b = \underbrace{\left[[(C_i^b(\hat{q}) m^i)\times] \;\; 0_3 \;\; 0_3 \;\; 0_3 \;\; 0_3 \;\; I_3 \right]}_{h} \delta x + \underbrace{\mathrm{diag}(\sigma_m^2)}_{R}$$

Note that ES MEKF uses measurements to innovation of nominal state, cause can't measure error.

2) GPS: measure $\tilde{r}^i$, with $\tilde{r} - \hat{r} = I\delta r + n_r$. Hence

$$\delta\tilde{r}^i = \left[0_3 \; 0_3 \; I_3 \;\; 0_3 \; 0_3 \; 0_3 \right] \delta x + \underbrace{\mathrm{diag}(\sigma_{GPS}^2)}_{R}$$

(assuming diag)

Note that we can do the same with $\tilde{v} - \hat{v} = I\delta v + n_v$ if GPS gives velocity with measurement model better than just differentiating position measurements (i.e. it adds new data).

Can add wind as param in measurement noise (not discussed here).

Now we get $K_n = P_{n,n} H^T (H P_{n,n} H^T + R)^{-1}$, $\delta x_{n,n} = \delta x_{n,n-1} + K_n(\text{innovation})$.
Only add to error correction if doing multiple measurements at once.
Finally, $P_{n,n} = (I - K_n H) P_{n,n-1}$
give expanded/non-simplified equation here?

Now update nominal state

$$\hat{q}_{n,n} = q_{n,n-1} \otimes \begin{bmatrix} 1 \\ \alpha/2 \end{bmatrix}$$
$$\hat{r}_{n,n} = \hat{r}_{n,n-1} + \delta r$$
$$\hat{v}_{n,n} = \hat{v}_{n,n-1} + \delta v$$

Note that after each measurement we should set $\delta x = 0$. But if not directly updating/calibrating sensors with biases, then we should **NOT** reset bias terms in δx to 0, but save them and subtract biases from any measurement (gyro, acc, mag) in EKF code.

Next Steps

1) We assume Earth is inertial frame but should account for Coriolis acceleration, Schuler tuning, and plumb-bob gravity.

2) Pull m^i from WMM for long distance flights with GPS.

3) Accounting for wind by updating GPS measurement cov matrix.

4) Look into UKF, IEKF maybe?

5) GPS velocity correction (combine with altimeter?)

6) Test with dynamics model with less assumptions/keeping 2nd degree terms.

Appendix

1) Prove $\exp\left(\Omega\left(\vec{\sigma}\right)\right) = \cos\left(\|\vec{\sigma}\|\right) I_4 + \frac{\sin(\|\vec{\sigma}\|)}{\|\vec{\sigma}\|}\Omega\left(\vec{\sigma}\right)$ where $\vec{\sigma} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$, $\|\vec{\sigma}\| = \sqrt{x^2+y^2+z^2}$, $\exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!}$ $\forall$ matrices A, and

$$\Omega\left(\vec{\sigma}\right) = \begin{bmatrix} 0 & -\vec{\sigma}^T \\ \vec{\sigma} & -[\vec{\sigma}\times] \end{bmatrix} = \begin{bmatrix} 0 & -x & -y & -z \\ x & 0 & z & -y \\ y & -z & 0 & x \\ z & y & -x & 0 \end{bmatrix}$$

Now notice,
$$\left(\Omega\left(\vec{\sigma}\right)\right)^2 = \begin{bmatrix} 0 & -x & -y & -z \\ x & 0 & z & -y \\ y & -z & 0 & x \\ z & y & -x & 0 \end{bmatrix}\begin{bmatrix} 0 & -x & -y & -z \\ x & 0 & z & -y \\ y & -z & 0 & x \\ z & y & -x & 0 \end{bmatrix}$$

$$= \left(-x^2 - y^2 - z^2\right) I_4$$

$$\left(\Omega\left(\vec{\sigma}\right)\right)^2 = -\|\vec{\sigma}\|^2 I_4 \quad \ldots \; (*)$$

Now
$$\exp\left(\Omega\left(\vec{\sigma}\right)\right) = \sum_{k=0}^{\infty} \frac{\left(\Omega\left(\vec{\sigma}\right)\right)^k}{k!}$$

$$= \sum_{k=0}^{\infty} \frac{\left(\Omega\left(\vec{\sigma}\right)\right)^{2k}}{(2k)!} + \sum_{k=0}^{\infty} \frac{\left(\Omega\left(\vec{\sigma}\right)\right)^{2k+1}}{(2k+1)!} \qquad \because \text{splitting odd and even terms}$$

$$= \sum_{k=0}^{\infty} \frac{\left(\left(\Omega\left(\vec{\sigma}\right)\right)^2\right)^k}{(2k)!} + \sum_{k=0}^{\infty} \frac{\left(\left(\Omega\left(\vec{\sigma}\right)\right)^2\right)^k}{(2k+1)!}\Omega\left(\vec{\sigma}\right)$$

$$= \sum_{k=0}^{\infty} \frac{\left(-\|\vec{\sigma}\|^2 I_4\right)^k}{(2k)!} + \sum_{k=0}^{\infty} \frac{\left(-\|\vec{\sigma}\|^2 I_4\right)^k}{(2k+1)!} \qquad \because (*)$$

$$= \sum_{k=0}^{\infty} \frac{(-1)^k \|\vec{\sigma}\|^{2k}}{(2k)!} I_4 + \sum_{k=0}^{\infty} \frac{(-1)^k \|\vec{\sigma}\|^{2k}}{(2k+1)!}\Omega\left(\vec{\sigma}\right)$$

$$= \cos\left(\|\vec{\sigma}\|\right) I_4 + \sum_{k=0}^{\infty} \frac{(-1)^k \|\vec{\sigma}\|^{2k}}{(2k+1)!}\Omega\left(\vec{\sigma}\right) \qquad \because \text{by taylor series } \cos(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k}}{2k!}$$

$$= \cos\left(\|\vec{\sigma}\|\right) I_4 + \sum_{k=0}^{\infty} \frac{(-1)^k \|\vec{\sigma}\|^{2k+1}}{(2k+1)!}\left(\frac{1}{\|\vec{\sigma}\|}\right)\Omega\left(\vec{\sigma}\right)$$

$$= \cos\left(\|\vec{\sigma}\|\right) I_4 + \frac{\sin(\|\vec{\sigma}\|)}{\|\vec{\sigma}\|}\Omega\left(\vec{\sigma}\right) \qquad \because \text{taylor series for } \sin(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$$