Navigation filter design towards a lunar lander

John O. “Juno” Woods, Ph.D.
Guidance, Navigation & Control
Open Lunar Foundation
San Francisco, California

December 16, 2019

1 Introduction

The Open Lunar Foundation has a sequence of three missions planned: a LEO 3U cubesat, a lunar orbiter, and a lunar lander. Each mission is designed to test concepts and hardware applicable for subsequent missions, with an emphasis on providing open source solutions that can be utilized by other teams.

1.1 Sensors

While a typical low-earth orbit cubesat includes a GPS receiver, our mission relies primarily on horizon recognition [Christian, 2017] for positioning, in order to build a rendering and simulation capability for future terrain-relative navigation technologies needed for landing, as well as to reduce reliance on potentially expensive two-way radiometric solutions. We use a star tracker for attitude, with an IMU used for the propagation.

1.2 Frames

Let us define some attitude frames:

- i for the inertial frame (centered on either the earth or the moon);
- e for the earth-centered, earth-fixed frame;

- l for the lunar-centered, lunar-fixed frame;
- p for a generic planet-centered, planet-fixed frame (which could be either e or l);
- b for the spacecraft's body frame; and
- c to indicate the sensor case frame for either the star tracker or the horizon camera, as needed;

1.3 Quaternions

Rotations may be represented as direction cosine matrices or as unit quaternions [Markley, 2003]. A unit quaternion is defined as

$$\mathrm{q} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} = \begin{bmatrix} \cos(\phi/2) \\ \mathrm{e}\sin(\phi/2) \end{bmatrix} , \tag{1}$$

where e is the axis of rotation and ϕ is the angle. If we write $\mathrm{q}_v = \begin{bmatrix} q_x & q_y & q_z \end{bmatrix}^\top$, then we can relate the quaternion to the direction cosine matrix by

$$\mathrm{T}(\mathrm{q}) = \left(q_w^2 - q_v^2\right) \mathrm{I}_3 - 2q_w \left[\mathrm{q}_v \times\right] + 2\mathrm{q}_v\mathrm{q}_v^\top . \tag{2}$$

Note that q and $-$q are degenerate; they represent the same rotation matrix. We arbitrarily restrict $q_w \geq 0$.

We write the quaternion product as

$$\mathrm{p} \otimes \mathrm{q} = \begin{bmatrix} p_w q_w - \mathrm{p}_v^\top \mathrm{q}_\mathrm{v} \\ p_w \mathrm{q}_v + q_w \mathrm{p}_v - \mathrm{p}_v \times \mathrm{q}_v \end{bmatrix} \tag{3}$$

as in Shuster [1993]. We can relate quaternion products to the matrix products of transformation matrices as

$$\mathrm{T}(\mathrm{p})\,\mathrm{T}(\mathrm{q}) = \mathrm{T}(\mathrm{p} \otimes \mathrm{q})\,. \tag{4}$$

We can also write the rate of change of a quaternion as

$$\dot{\mathrm{q}} \;=\; \frac{1}{2}\Omega(\boldsymbol{\omega})\mathrm{q} \tag{5}$$

$$\Omega(\boldsymbol{\omega}) \;=\; \begin{bmatrix} 0 & -\boldsymbol{\omega}^\top \\ \boldsymbol{\omega} & -\left[\boldsymbol{\omega}\times\right] \end{bmatrix} , \tag{6}$$

where $\boldsymbol{\omega}$ the angular rate ω multiplied by the unit vector defining the axis about which the rotation is occurring.

1.4 Attitude errors

We define $\mathrm{T}_a^b = \mathrm{q}_a^b$ as the 3×3 transformation from frame a to frame b.

Suppose that some sensor's true mounting (after spacecraft assembly) has the frame transformation T_b^c. Then we define

$$\mathrm{T}_b^c = \mathrm{T}_{\hat{c}}^c \mathrm{T}_b^{\hat{c}} \tag{7}$$

as the matrix product of body-to-nominal-mount $\mathrm{T}_b^{\hat{c}}$ and the nominal-mount-to-actual-mount $\mathrm{T}_{\hat{c}}^c$. Moreover, we can use the small angle approximation to define, without loss of generality,

$$\begin{align} \mathrm{T}_{\hat{A}}^A &\approx \mathrm{I} - [\mathrm{a}\times] \tag{8} \\ \mathrm{a} &= \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^\top \tag{9} \end{align}$$

which is defined by the unit-vector axis $\frac{\mathrm{a}}{a}$, with a being the angular rotation about that axis.[1] The skew-symmetric matrix is defined as

$$[a\times] = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} . \tag{10}$$

We can also approximate the inverse as

$$\mathrm{T}_A^{\hat{A}} \approx \mathrm{I} + [\mathrm{a}\times] \; . \tag{11}$$

2 State

Note that we write the state as xand the estimated state as $\hat{\mathrm{x}}$. For simplicity, we usually drop the hat, but sometimes retain it for clarity.

We define our state as

$$\mathrm{x} = \begin{bmatrix} \mathrm{r} & \mathrm{v} & \delta\boldsymbol{\phi} & \mathrm{b}_a & \mathrm{b}_g \end{bmatrix}^\top \tag{12}$$

where each component is a length-3 sub-vector. ris the position of the spacecraft IMU in the inertial frame, vis its velocity, $\boldsymbol{\phi}$ the attitude error, and b_a and b_g the accelerometer and gyroscope bias terms. In the future, we may also include 'constant' misalignment terms for the sensors.[2]

[1] This definition follows from Equations 1 and 25 of Markley [2003].

[2] We call these terms constant because they are not propagated, but they may be estimated by measurements.

2.1 State propagation

The position and velocity states are propagated according to

$$\begin{aligned}\dot{\mathrm{r}} &= \mathrm{v} \\ \dot{\mathrm{v}} &= \mathrm{a}_g + \mathrm{a}_{ng} ,\end{aligned} \tag{13}$$

where the two acceleration terms are the gravitational and non-gravitational components. The former of these is not observable by the IMU (since the IMU measures zero acceleration in free fall), but the latter is observable in the IMU case frame as a_c. We can rewrite it in terms of the accelerometer measurement as

$$\dot{\mathrm{v}} = \mathrm{a}_g + \mathrm{T}(\mathrm{q}_b^i)\, \mathrm{T}_c^b\, \mathrm{a}_c \, . \tag{14}$$

Likewise, we can propagate the attitude estimate — which is not part of the state vector — using Eq. 6 in terms of the case-frame gyroscope measurement, which is the same as the body-frame gyroscope measurement, e.g. ${}_c\omega_{c/i} = {}_b\omega_{b/i}$; the attitude propagation is given by

$$\dot{\mathrm{q}}_i^b = \Omega({}_b\omega_{b/i})\mathrm{q}_i^b \, . \tag{15}$$

The bias terms all have a rate of zero; they are not propagated.

2.2 Error propagation

Although we propagate the quaternion as part of the state, our attitude covariance reflects the attitude error component of the state vector. Similarly, the bias terms are not propagated, but since they are estimated, we need to be able to propagate our certainty about them. We must therefore come up with time derivatives for these terms.

We follow Bishop et al. [2016] in writing

$$\mathrm{q}_i^b = \mathrm{q}_{\hat{b}}^b \otimes \mathrm{q}_i^{\hat{b}} \tag{16}$$

and defining the attitude error as the rotation from the estimated body frame to the true body frame,

$$\delta\mathrm{q}_b = \mathrm{q}_{\hat{b}}^b \, . \tag{17}$$

Given the definition of the quaternion from Eq. 1, a small angle assumption allows us to write

$$\delta\mathrm{q}_b \approx \begin{bmatrix} 0 \\ \frac{1}{2}\delta\phi \end{bmatrix} \, . \tag{18}$$

We can then write the time derivative of the attitude error's vector component as

$$\delta \dot{\mathrm{q}}_{b_v} = -\left[{}_b\boldsymbol{\omega}_{b/i}\times\right] \delta \mathrm{q}_{b_v} - \frac{1}{2}\delta \mathrm{b}_g \,, \tag{19}$$

the first term of which is recognizable as the quaternion time-rate-of-change equation rewritten with a small angle assumption, and the second term of which is the gyroscope bias. If we write it instead as the time-rate-of-change of the attitude error rotation vector, we get

$$\delta \dot{\boldsymbol{\phi}} = -\left[{}_b\boldsymbol{\omega}_{b/i}\times\right] \delta \boldsymbol{\phi} - \delta \mathrm{b}_g \,. \tag{20}$$

We can also write the attitude error in matrix notation using

$$\begin{aligned} \mathrm{T}(\mathrm{q}_i^b) &= \mathrm{T}_{\hat{b}}^b\, \mathrm{T}(\mathrm{q}_i^{\hat{b}}) = \delta \mathrm{T}_b\, \mathrm{T}(\mathrm{q}_i^{\hat{b}}) \\ &= (\mathrm{I} - [\delta \boldsymbol{\phi}\times])\, \mathrm{T}(\mathrm{q}_i^{\hat{b}}) \,. \end{aligned} \tag{21}$$

Finally, the errors on the bias terms are propagated as exponentially correlated random variables,

$$\delta \dot{\mathrm{b}} = -\frac{1}{\tau}\delta \mathrm{b} + \boldsymbol{\nu} \,, \tag{22}$$

where $\boldsymbol{\nu}$ is zero-mean white noise, τ is the time constant, and $\delta \mathrm{b} = \mathrm{b} - \hat{\mathrm{b}}$ for each bias term.

2.3 Dynamics matrix

If our states x are propagated by $\dot{\mathrm{x}}$, we can linearize at some prior state $\hat{\mathrm{x}}$ and define a dynamics matrix as

$$\mathrm{F} = \left.\frac{\partial \dot{\mathrm{x}}}{\partial \mathrm{x}}\right|_{\mathrm{x}=\hat{\mathrm{x}}} \,. \tag{23}$$

The nonzero components of this matrix are

$$\begin{aligned}
\frac{\partial \dot{\mathrm{r}}}{\partial \mathrm{v}} &= \mathrm{I} && (24)\\
\frac{\partial \dot{\mathrm{v}}}{\partial \mathrm{r}} &= \mathrm{G}(\mathrm{r} + \mathrm{r}_{\mathrm{cg/imu}}) && (25)\\
\frac{\partial \delta \dot{\phi}}{\partial \phi} &= -\left[{}_b\tilde{\omega}_{b/i}\times\right] && (26)\\
\frac{\partial \delta \dot{\phi}}{\partial \mathrm{b}_g} &= -\mathrm{I} && (27)\\
\frac{\partial \delta \dot{\mathrm{b}}}{\partial \mathrm{b}} &= -\frac{1}{\tau}\mathrm{I}_{6\times 6}\,. && (28)
\end{aligned}$$

Note that the gravity gradient is given by

$$\mathrm{G}(\mathrm{r}) = \frac{3\mu}{r^5}\mathrm{r}\mathrm{r}^\top - \frac{\mu}{r^3}\mathrm{I}\,. \qquad (29)$$

2.4 State transition matrix

We use a first-order approximation for the state transition matrix,

$$\Phi(t_k + \Delta t, t_k) \approx \mathrm{I} + \mathrm{F}\Delta t\,. \qquad (30)$$

This can be used to propagate the covariance forward in time:

$$\mathrm{P}_k^- = \Phi_{k,k-1}\mathrm{P}_{k-1}^+\Phi_{k,k-1}^\top + \mathrm{Q}_k \qquad (31)$$

However, we may use a more carefully optimized formulation, from D'Souza and Hanak [2011], if processor cycles are in doubt.

$$\begin{aligned}
\Phi(t_{k+1}, t_k) &= \mathbf{I} + \mathbf{F}(t_k)\,(t_{k+1} - t_k) = \mathbf{I} + \mathbf{F}_k\,\Delta t \;\text{ and}\\
\Phi(t_{k+1}, t_{k-1}) &= \Phi(t_{k+1}, t_k)\,\Phi(t_k, t_{k-1})\,,\text{ so}\\
\Phi(t_{k+1}, t_{k-1}) &= \Phi(t_k, t_{k-1}) + \mathbf{F}(t_k)\,\Phi(t_k, t_{k-1})\,\Delta t\,.
\end{aligned}$$

2.5 State update

We have a choice between state update maths which dramatically changes the floating point requirements on the flight hardware.

We always start by computing the measurement residual

$$\delta \mathrm{y} = \tilde{\mathrm{y}} - \mathrm{h}(\hat{\mathrm{x}})\,, \tag{32}$$

where $\tilde{\mathrm{y}}$ is the vector measurement, and $\mathrm{h}(\hat{\mathrm{x}})$ is the measurement model evaluated at some estimated state $\hat{\mathrm{x}}$. We also need the measurement sensitivity matrix,

$$\mathrm{H} = \left.\frac{\partial \mathrm{h}(x)}{\partial \mathrm{x}}\right|_{\mathrm{x}=\hat{\mathrm{x}}}\,, \tag{33}$$

and the measurement residual covariance R.

2.5.1 Vector state update

We first compute the innovation covariance and underweighted innovation covariance,

$$\begin{aligned} \mathrm{W} &= \mathrm{H}\mathrm{P}^{-}\mathrm{H}^{\top} + \mathrm{R} & (34) \\ \mathrm{W}_u &= \mathrm{W}k_u\,, & (35) \end{aligned}$$

where k_u is the underweighting scalar (some value between 0 and 1, defaulting to 1).

The measurement residual is now tested to see if it is an outlier; this process is occasionally referred to as a *residual edit check*. For a vector measurement, the square root of

$$z^2 = \delta \mathrm{y}^{\top} \mathrm{W}^{-1} \delta \mathrm{y} \tag{36}$$

gives the number of standard deviations between the measurement and the measurement model (telling us how much of an outlier it might be relative to the current state estimate). If $z^2 > z^2_{\mathrm{max}}$, we reject the measurement (and do not update the state or covariance with it).

Since W ought to be positive definite, we can use a Cholesky decomposition. Many measurements are 3×3, for which the Cholesky factorization algorithm is fairly straightforward. We write

$$\begin{aligned} z^2 &= \delta \mathrm{y}^{\top} \left(\mathrm{L}\mathrm{L}^{\top}\right)^{-1} \delta \mathrm{y} \\ &= \delta \mathrm{y}^{\top} \left(\mathrm{L}^{\top}\right)^{-1} \mathrm{L}^{-1} \delta \mathrm{y} \\ &= \mathrm{z}^{\top}\mathrm{z}\,, \end{aligned}$$

where

$$z = L^{-1}\delta y\,. \tag{37}$$

If the Cholesky factorization fails, we can employ an $LDL^\top$ decomposition instead; however, it's dramatically easier from a verification and algorithmic standpoint to mark measurements with Cholesky decomposition errors as having failed the edit check, and not allow the failed measurement to contaminate the state or covariance.

The Kalman gain is given by

$$K = P^- H^\top W_u^{-1}\,, \tag{38}$$

and the measurement state update by

$$\begin{aligned} \delta x &= K\,\delta y \qquad (39)\\ \hat{x}^+ &= \hat{x}^- + \delta x\,. \qquad (40) \end{aligned}$$

To complete the covariance update, we use the Joseph form,

$$P^+ = (I - KH)\,P^-\,(I - KH)^\top + KRK^\top\,, \tag{41}$$

which unfortunately requires several large matrix multiplications.

2.5.2 Scalar state update

We now examine an update strategy which requires no Cholesky decomposition and much less in the way of matrix multiplication math. In this strategy, we treat each scalar component of a measurement vector independently.

The innovation and underweighted innovation are given by

$$\begin{aligned} w_i &= H_i P^- H_i^\top + r_i \qquad (42)\\ w_{i,u} &= w_i k_u, \qquad (43) \end{aligned}$$

respectively; r_i is the measurement variance for component i and k_u is the underweighting scalar. The residual edit check calculation is

$$z^2 = \frac{\delta y_i^2}{w_i}\,. \tag{44}$$

It is also possible to check the entire measurement as a unit instead of component-by-component; for this, we use Eq. 36.

A Kalman gain matrix column is computed for each scalar measurement component δy_i as

$$\mathrm{K}_i = \frac{1}{w_{i,u}} \mathrm{P}^- \mathrm{H}_i^\top \,. \tag{45}$$

The scalar state update corresponding to each measurement component is

$$\delta \mathrm{x}_i = \mathrm{K}_i \delta y_i \,, \tag{46}$$

We now proceed to adjust the prior covariance with the new evidence, yielding a posterior covariance. The measurement residual covariance used in the Joseph form, Eq. 41, is

$$\mathrm{R} = \mathrm{diag}(r_1, r_2, \cdots, r_k) \,. \tag{47}$$

By multiplying out the covariance update equation, we get

$$\begin{aligned} \mathrm{P}^+ &= \mathrm{P}^- - \mathrm{KHP}^- - \mathrm{P}^-\mathrm{H}^\top\mathrm{K}^\top + \mathrm{K}\left(k_u(\mathrm{HP}^-\mathrm{H}^\top + \mathrm{R})\right)\mathrm{K}^\top \\ &= \mathrm{P}^- - \mathrm{KHP}^- - \mathrm{P}^-\mathrm{H}^\top\mathrm{K}^\top + \mathrm{KW}_u\mathrm{K}^\top \,, \end{aligned}$$

where

$$\mathrm{W}_u = \mathrm{diag}(w_{1,u}, w_{2,u}, \cdots, w_{k,u}) \,. \tag{48}$$

This form avoids large matrix multiplications by treating each measurement component as independent of the others. Notably, the quantity $\mathrm{P}^-\mathrm{H}^\top$ and its transpose have already been calculated in Eq. 45.

3 Measurement models

3.1 Horizon

We use the horizon navigation model described by Christian [2017].

3.1.1 Summary

Given a camera image of a lit horizon arc, we define a principal axis frame p and a camera frame c whose positive z axis points away from the camera (in the viewing direction).

The position of the spacecraft camera with respect to the visible planet is given by

$$\mathrm{r}_{c/p} = -(\mathrm{n}^\top\mathrm{n} - 1)^{-1/2}\, \mathrm{T}_p^c\, \mathrm{Q}^{-1}\mathrm{n} \,, \tag{49}$$

where the semi axes of the planet ellipsoid model are given by a, b, and c in

$$\mathrm{Q}^{-1} = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} . \tag{50}$$

n is the solution to a linear least squares problem,

$$\begin{bmatrix} \bar{\mathrm{s}}_1'^{\top} \\ \vdots \\ \bar{\mathrm{s}}_m'^{\top} \end{bmatrix} \mathrm{n} = 1_{m \times 1} \tag{51}$$

where

$$\begin{aligned} \bar{\mathrm{s}}_i' &= \frac{\bar{\mathrm{s}}_i}{\bar{s}_i} \\ \bar{\mathrm{s}}_i &= \mathrm{Q}\,\mathrm{T}_c^{pa}\,\mathrm{s}_i \\ \mathrm{s}_i &= \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{d_x} & -\frac{\alpha}{d_x d_y} & \frac{\alpha v_p - d_y u_p}{d_x d_y} \\ 0 & \frac{1}{d_y} & -\frac{v_p}{d_y} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} . \end{aligned}$$

The last of these equations gives the pinhole camera model mapping the pixel coordinates of m points on a body's lit limb u_i, v_i to image plane coordinates x_i, y_i using the camera parameters α (detector array skewness), d_x and d_y (ratio of focal length to pixel pitch), and principal point (u_p, v_p) (the boresight intersection with the image plane). It is not straightforward to define $\bar{\mathrm{s}}_i'$ in a concise way, but we may think of them as unit vectors from the camera to the observed points in a world where the planet is a sphere of radius 1. In that same world, then, n is a vector pointing from the camera to the center of the sphere. [Christian and Robinson, 2016]

3.1.2 Model

Having now briefly described the model, we can start to put it in the correct terms for our navigation filter. Eq. 49 is given in terms of the camera position with respect to the planet, but we may choose to navigate relative to a planet that is not at the origin of our inertial frame. For example, we may wish to navigate with respect to the moon while still well outside of the moon's sphere-of-influence.

Let us use p to indicate the planet used as the inertial frame's center and q to indicate the other. Then we can give the measurement, in the camera frame, as

$$\mathrm{y} = {}_c\mathrm{r}_{c/q} = \mathrm{T}^{c}_{\hat{c}}\mathrm{T}^{\hat{c}}_{b}\left(\mathrm{T}^{b}_{\hat{b}}\mathrm{T}^{\hat{b}}_{i}\left({}_i\mathrm{r}_{\mathrm{imu}/p} - {}_i\mathrm{r}_{q/p}\right) + {}_b\mathrm{r}_{c/\mathrm{imu}}\right) \,. \tag{52}$$

Thus $\mathrm{r}_{q/p}$ is the distance of the observed planet from the inertial frame center. If the observed planet is also the inertial frame center, this quantity is a 0 vector. $\mathrm{T}^{c}_{\hat{c}}$ represents the camera mismount (which is not a filter state, but might be in the future). In this model, ${}_i\mathrm{r}_{\mathrm{imu}/p}$ is our filter's position state r.

The measurement partials are given by

$$\begin{aligned}
\frac{\partial \mathrm{y}}{\partial \mathrm{r}} &= \mathrm{T}^{c}_{b}\mathrm{T}^{b}_{i} && (53)\\
\frac{\partial \mathrm{y}}{\partial \delta \phi} &= \mathrm{T}^{c}_{b}\left[\mathrm{T}^{\hat{b}}_{i}(\mathrm{r} - \mathrm{r}_{q/p})\times\right] && (54)\\
\frac{\partial \mathrm{y}}{\partial \delta \theta} &= \left[\mathrm{T}^{\hat{c}}_{b}\left(\mathrm{T}^{b}_{i}(\mathrm{r} - \mathrm{r}_{q/p}) + {}_b\mathrm{r}_{c/\mathrm{imu}}\right)\times\right] \,, && (55)
\end{aligned}$$

where $\delta\theta$ is used to indicate the mismount error state vector.

This model likely needs a bias term as well, but we'll come back to that in a later iteration.

3.2 Global positioning system

Gps may be used in a separate filter in order to evaluate the results of the horizon filter. The gps provides measurements of position and velocity, both in an earth-centered, earth-fixed frame e.

References

John A. Christian. Accurate planetary limb localization for image-based spacecraft navigation. *Journal of Spacecraft and Rockets*, pages 1–23, 2017. ISSN 0022-4650. doi: 10.2514/1.A33692. URL `https://arc.aiaa.org/doi/10.2514/1.A33692`.

F Landis Markley. Attitude error representations for Kalman filtering. *Journal of Guidance, Control, and Dynamics*, 26(2):311–317, 2003.

Malcolm D. Shuster. A survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517, 1993.

Robert H Bishop, Tim Crain, Kyle J. DeMars, Chad Hanak, John M. Carson, Nikolas Trawny, and John A. Christian. An inertial dual-state state estimator for precision planetary landing with hazard detection and avoidance. In *AIAA Guidance, Navigation, and Control Conference*, number January, pages 1–21, 2016. ISBN 978-1-62410-389-6. doi: 10.2514/6.2016-0098. URL `http://arc.aiaa.org/doi/10.2514/6.2016-0098`.

Christopher D'Souza and Chad Hanak. The state transition and process noise matrices in the Orion FILTNAV and EKF. Technical report, NASA JSC Engineering Directorate, Houston, TX, US, 2011.

John A. Christian and Shane B. Robinson. Noniterative horizon-based optical navigation by Cholesky factorization. *Journal of Guidance, Control, and Dynamics*, 39(12):2755–2763, 2016. ISSN 07315090. doi: 10.2514/1.G000539.