AIAA Atmospheric Flight Mechanics Conference and Exhibit
5-8 August 2002, Monterey, California

AIAA 2002-4501

MARS ENTRY NAVIGATION: ATMOSPHERIC INTERFACE THROUGH PARACHUTE DEPLOY

Dr. Timothy P. Crain
Member AIAA
Aerospace Engineer
NASA Johnson Space Center
Houston, TX 77058

Dr. Robert H. Bishop
Associate Fellow AIAA
Professor
The University of Texas at Austin
Austin, TX 77812

Abstract

The foundation of a linear error analysis associated with a Mars entry mission phase in which only inertial measurement unit measurements are available is presented. The entry mission phase considered begins at atmospheric entry interface and terminates at parachute deploy. The main question addressed is how to incorporate the accelerometer and gyro errors in the state estimation error covariance growth online during dead-reckoning navigation. The results show that an approximate error covariance can be constructed that accurately reflects the true navigation uncertainty. A *monte carlo* analysis confirms the approximate analytic error covariance formulation.

Introduction

On-board entry navigation at Mars is comprised of a sequence of navigation modes characterized by the availability of various classes of measurement data. The entry, descent, and landing (EDL) scenario is depicted in Figure 1[1]. Typical measurement classes include attitude and acceleration observations provided by a strapdown inertial measurement unit (IMU), conventional radar altimetry, LIDAR, and optical and radiometric observations providing measurements of relative ground velocity and hazard locations. The difficulty in achieving precision landing on Mars is due mainly to the fact that in the upper atmosphere, where there is sufficient time and available lift to actively guide out any delivery and navigation errors, there are no external measurements available for navigation, hence the guidance must depend only on IMU dead-reckoning. Once the heat shield is jettisoned and the altimeter, now exposed to the external environment, provide measurements to the navigation filter (assisted by the IMU), the spacecraft is on the parachute and cannot be actively guided using lift modulation. During this phase of EDL, the navigation uncertainty is significantly reduced, but guidance cannot compensate for any existing state errors. In the final phase of EDL, the parachutes are jettisoned and the altimeter and LIDAR (or other hazard avoidance sensor) are available. Guidance can now actively be utilized to maneuver the vehicle, however, there is not much time to make large excursions to hit a pinpoint landing. The challenge is (working within the constraints of nonavailability of sensors) to provide the most accurate navigated state and on-board computation of the uncertainty in that estimated state.

In this paper, we present the foundation of a linear error analysis associated with the entry mission phase in which only IMU measurements are available, that is, after atmospheric entry interface but before the first parachute deploy. The navigation solution methodology is conventional in the sense that the IMU provided observations are not incorporated as measurements in the navigation filter. Rather, the observations are numerically integrated in conjunction with a simplified gravity model—this is known as *dead-reckoning navigation*. The main question to be addressed is how to correctly account for the accelerometer and gyro errors in the state estimation error covariance growth on-line. Since active guidance acts on the navigated state, the knowledge of the uncertainty associated with the dead-reckoning navigation is important to overall mission success.

Reference Frames

Planet-Centered Reference Frames

The planet-centered inertial frame is depicted in Figure 2, where $\mathbf{e}_3^i$ lies along the planet spin axis, and $\mathbf{e}_1^i$ and $\mathbf{e}_2^i$ lie in the equatorial plane and form a right-handed coordinate system.

We assume the planet is rotating at a constant rate, denoted by Ω, that is

$$\mathbf{\Omega}^i = \begin{bmatrix} 0 \\ 0 \\ \Omega \end{bmatrix} .$$

As illustrated in Figure 2, the planet-centered rotating frame unit vector $\mathbf{e}_3^m$ is aligned with the planet-centered inertial frame unit vector $\mathbf{e}_3^i$. The planet-centered rotating frame rotates relative to the planet-centered inertial frame at a constant rate, $\Omega := \|\mathbf{\Omega}^i\|$, through the angle $\theta = \Omega t + \theta_0$.

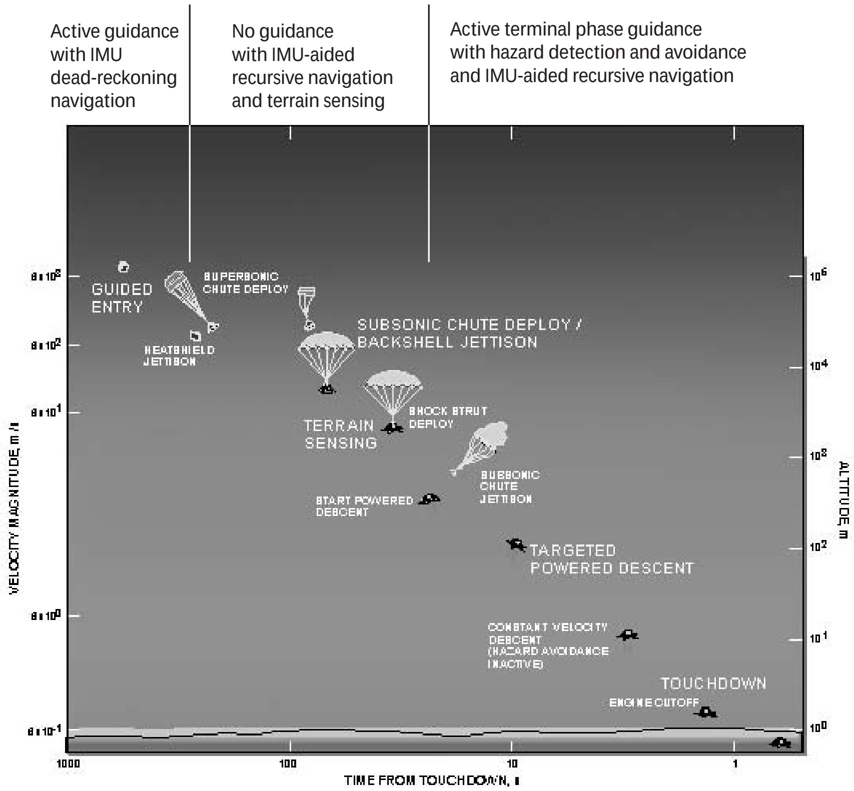

Figure 1: An entry, descent, and landing scenario at Mars.

Wind Frame

The wind frame, defined with respect to the atmosphere, is depicted in Figure 3. The spacecraft velocity relative to the atmosphere is given by

$$\mathbf{v}_{rel} = \dot{\mathbf{r}} - \mathbf{\Omega} * \mathbf{r} \ ,$$

where $\mathbf{r}$ is the position of the entry vehicle in the planet-centered inertial frame. The unit vector $\mathbf{e}_1^w$ is defined as

$$\mathbf{e}_1^w = \frac{\mathbf{v}_{rel}}{v_{rel}} \ , \tag{1}$$

where $\mathbf{v}_{rel}$ is the velocity of the entry vehicle relative to the local atmosphere in the planet-centered inertial frame, $v_{rel} := \|\mathbf{v}_{rel}\|$, and the superscript "w" denotes the wind reference frame. Since the vectors $\mathbf{e}_2^w$ and $\mathbf{e}_3^w$ only need to span the "lift space," we can use the definitions

$$\mathbf{e}_2^w = \frac{\mathbf{e}_1^w * \mathbf{r}}{\|\mathbf{e}_1^w * \mathbf{r}\|} \qquad \text{and} \qquad \mathbf{e}_3^w = \mathbf{e}_1^w * \mathbf{e}_2^w \ . \tag{2}$$

This means that $\mathbf{e}_2^w$ is defined with respect to the local horizontal. Also, the transformation from the wind frame to the inertial frame is given by

$$\mathbf{L}_{IW} = [\ \mathbf{e}_1^w \ \vdots \ \mathbf{e}_2^w \ \vdots \ \mathbf{e}_3^w \ ] \ .$$

The bank angle, denoted by φ and shown in Figure 3, is the main guidance variable. It is a rotation about the relative velocity vector, $\mathbf{e}_1^w$. When $\varphi = 0$, we have a "lift-up" condition. Similarly, when $\varphi = \pm 180°$, we have a "lift-down" condition.

IMU Case Frame

The IMU case frame is attached to the IMU unit, as depicted in Figure 3. The accelerometer package measures nongravitational accelerations in the IMU case frame, denoted by $\mathbf{a}_m^c$. Dead-reckoning navigation requires that the IMU-measured accelerations be transformed to the inertial frame for numerical integration. The transformation matrix is a function of the spacecraft attitude, which we choose to represent via the classical Euler quaternion $\mathbf{Q} \in \mathcal{R}^4$, where

$$\mathbf{Q} := \begin{bmatrix} \mathbf{q} \\ q_4 \end{bmatrix} \in \mathcal{R}^4,$$

and $\mathbf{q} = [q_1 \ q_2 \ q_3]^T$ subject to the constraint that $\mathbf{q}^T\mathbf{q} + q_4^2 = 1$. In terms of the quaternion components, the transformation matrix from the IMU case

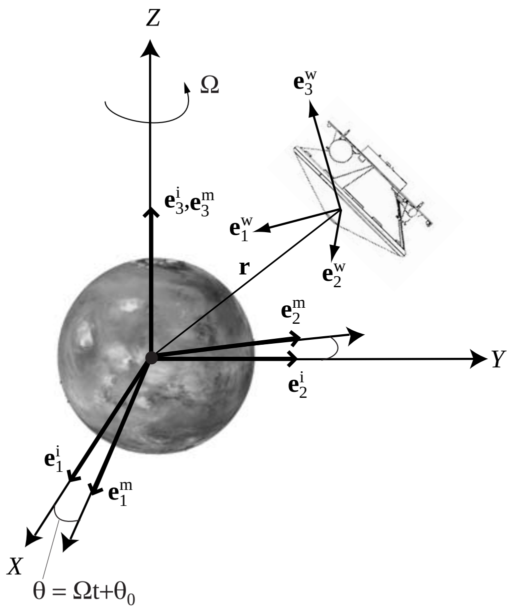

Figure 2: Planet-centered inertial and planet-centered rotating reference frames.

frame to the inertial frame is given by

$$\mathbf{T}_{IC} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_1q_2-q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_1q_4) \\ 2(q_1q_3+q_2q_4) & 2(q_2q_3-q_1q_4) & 1-2(q_1^2+q_2^2) \end{bmatrix}. \quad (3)$$

For purposes of computing the spacecraft attitude, the IMU unit contains a gyro package that provides measurements of the relative angular velocity of the IMU case reference frame relative to the inertial reference frame, denoted by $\boldsymbol{\omega}_m$. The measured angular velocity velocity vector is integrated to obtain the attitude, represented by $\mathbf{Q}$. Nominally, the IMU unit will be installed in the spacecraft at a position offset from the spacecraft center of mass. This position is not known perfectly, and will indeed vary as the spacecraft expends fuel. A potentially significant step change in the offset of the IMU from the center of mass is expected when the heat shield separates. If the IMU offset is not properly accounted for, attitude motion will inadvertently be measured as translated acceleration. The offset of the IMU from the center of mass is not considered here.

Dynamics Modeling

The system dynamics in the inertially-fixed frame are given in the general form

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= \mathbf{g}(\mathbf{r}) + \mathbf{T}_{IC}(\mathbf{Q})\mathbf{a}^c \\ \dot{\mathbf{Q}} &= \frac{1}{2}\mathbf{B}(\boldsymbol{\omega})\mathbf{Q} \end{aligned} \quad (4)$$

with the initial conditions

$$\mathbf{r}(t_0) = \mathbf{r}_0, \quad \mathbf{v}(t_0) = \mathbf{v}_0, \quad \mathbf{Q}(t_0) = \mathbf{Q}_0,$$

where $\mathbf{g}(\mathbf{r})$ is the acceleration due to gravity, $\mathbf{a}^c$ is the true nongravitational acceleration represented in the IMU case frame, $\mathbf{T}_{IC}$ is the transformation matrix from the IMU case frame to the inertial frame, and $\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]^T$ is the relative angular velocity vector of the IMU case frame with respect to the inertial frame. The matrix $\mathbf{B}(\boldsymbol{\omega})$ in Eq. (4) is defined to be

$$\mathbf{B}(\boldsymbol{\omega}) := \begin{bmatrix} 0 & \omega_z & -\omega_y & \omega_x \\ -\omega_z & 0 & \omega_x & \omega_y \\ \omega_y & -\omega_x & 0 & \omega_z \\ -\omega_x & -\omega_y & -\omega_z & 0 \end{bmatrix}. \quad (5)$$

Alternatively, we can represent the quaternion dynamics via

$$\dot{\mathbf{Q}} = \frac{1}{2}\mathbf{A}(\mathbf{Q})\boldsymbol{\omega} \quad (6)$$

where

$$\mathbf{A}(\mathbf{Q}) := \begin{bmatrix} q_4 & q_3 & q_2 \\ q_3 & q_4 & -q_1 \\ -q_2 & q_1 & q_4 \\ -q_1 & -q_2 & -q_3 \end{bmatrix}. \quad (7)$$

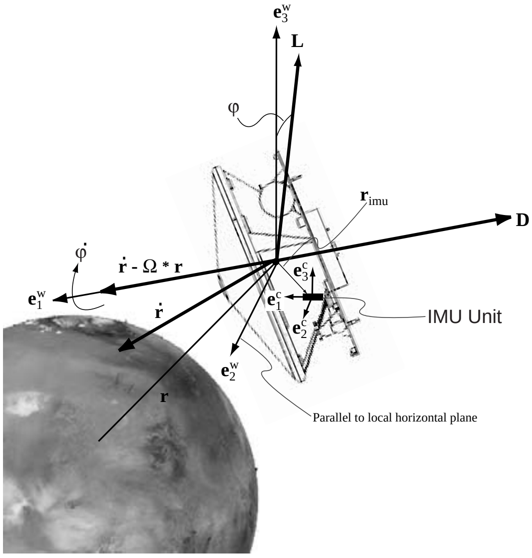

Figure 3: Wind and IMU case reference frames with lift/drag/bank angle definitions.

Sensor Error Models

The IMU unit contains both an accelerometer package and a gyro package. In this paper, we consider only the strapdown implementation of the IMU unit. The accelerometers and gyros produce measurements corrupted by random errors (noise and biases), systematic biases, and other errors. The effectiveness of the navigation system (esp. dead-reckoning navigation) to produce a precise navigated state depends almost entirely on the magnitude of the IMU errors. In dead-reckoning navigation, the other major contribution to uncertainty in the navigated state is the accuracy of the initial spacecraft state knowledge.

Accelerometers

The accelerometer package produces a measure of the spacecraft nongravitational accelerations in the IMU case frame, denoted by $\mathbf{a}_m^c$. The measurement of the nongravitational accelerations is corrupted by errors due to nonorthogonality and misalignment of the axes, errors due to scale factor uncertainties, random biases, and noise[2–4]. The strapdown accelerometer error model can be formulated as

$$\mathbf{a}_m^c = (\mathbf{I} + \boldsymbol{\Upsilon}_a)(\mathbf{I} + \boldsymbol{\Xi}_a)(\mathbf{a}^c + \mathbf{b}_a + \boldsymbol{\epsilon}_a), \qquad (8)$$

where

$$\boldsymbol{\Upsilon}_a := \begin{bmatrix} 0 & \upsilon_{a_{xz}} & -\upsilon_{a_{xy}} \\ -\upsilon_{a_{yz}} & 0 & \upsilon_{a_{yx}} \\ \upsilon_{a_{zy}} & -\upsilon_{a_{zx}} & 0 \end{bmatrix}$$

$$\boldsymbol{\Xi}_a := \begin{bmatrix} \xi_{a_x} & 0 & 0 \\ 0 & \xi_{a_y} & 0 \\ 0 & 0 & \xi_{a_z} \end{bmatrix} \qquad \mathbf{b}_a := \begin{bmatrix} b_{a_x} \\ b_{a_y} \\ b_{a_z} \end{bmatrix}$$

and $(\upsilon_{a_{yz}}, \upsilon_{a_{zy}}, \upsilon_{a_{zx}}, \upsilon_{a_{xz}}, \upsilon_{a_{xy}}, \upsilon_{a_{yx}})$ are nonorthogonality and axes misalignment errors, $\mathbf{b}_a$ is the bias in the accelerometer, $(\xi_{a_x}, \xi_{a_y}, \xi_{a_z})$ are scale factor errors, and $\boldsymbol{\epsilon}_a$ is a white noise stochastic process. The nonorthogonality and axes misalignment errors, scale factor errors, and bias parameters are modeled as zero-mean, Gaussian-distributed random constants with appropriate covariances. The noise $\boldsymbol{\epsilon}_a$ is modeled as a zero-mean, Gaussian-distributed, timewise uncorrelated random process, with

$$E(\boldsymbol{\epsilon}_a(t)\boldsymbol{\epsilon}_a^T(\tau)) = \mathbf{V}_a(t)\delta(t-\tau),$$

where $\delta(t)$ is the Dirac delta function. If we assume that the various errors are "small," then to first-order we have

$$(\mathbf{I} + \boldsymbol{\Upsilon}_a)(\mathbf{I} + \boldsymbol{\Xi}_a) \approx \mathbf{I} + \boldsymbol{\Upsilon}_a + \boldsymbol{\Xi}_a.$$

Defining

$$\boldsymbol{\Delta}_a := \boldsymbol{\Upsilon}_a + \boldsymbol{\Xi}_a \qquad (9)$$

yields the accelerometer measurement model

$$\mathbf{a}_m^c = (\mathbf{I} + \boldsymbol{\Delta}_a)(\mathbf{a}^c + \mathbf{b}_a + \boldsymbol{\epsilon}_a). \qquad (10)$$

Gyros

The gyro package produces a measure of the spacecraft angular velocity vector, that is, the angular velocity of the IMU case frame relative to the inertial reference frame, denoted by $\boldsymbol{\omega}_m$. The measurement of the angular velocity vector is corrupted by random biases, errors due to scale factor uncertainties, errors due to nonorthogonality and axes misalignments, and random noise. The gyro error model can be formulated as

$$\boldsymbol{\omega}_m = (\mathbf{I} + \mathbf{S}_g)(\mathbf{I} + \boldsymbol{\Gamma}_g)(\boldsymbol{\omega} + \mathbf{b}_g + \boldsymbol{\epsilon}_g) \ , \tag{11}$$

where $\mathbf{b}_g$ is the gyro bias, $\mathbf{S}_g$ is the gyro scale factor matrix, $\boldsymbol{\Gamma}_g$ is the gyro nonorthogonality and axes misalignment matrix, $\boldsymbol{\epsilon}_g$ is a white noise stochastic process, and where

$$\begin{aligned} \mathbf{S}_g &:= \begin{bmatrix} S_{g_x} & 0 & \\ 0 & S_{g_y} & 0 \\ 0 & 0 & S_{g_z} \end{bmatrix} \\ \boldsymbol{\Gamma}_g &:= \begin{bmatrix} 0 & \gamma_{g_{xz}} & -\gamma_{g_{xy}} \\ -\gamma_{g_{yz}} & 0 & \gamma_{g_{yx}} \\ \gamma_{g_{zy}} & -\gamma_{g_{zx}} & 0 \end{bmatrix} \end{aligned} \tag{12}$$

and $(\gamma_{g_{yz}}, \gamma_{g_{zy}}, \gamma_{g_{zx}}, \gamma_{g_{xz}}, \gamma_{g_{xy}}, \gamma_{g_{yx}})$ are nonorthogonality and axes misalignment errors, $\mathbf{b}_g$ is the bias in the gyro, $(S_{g_x}, S_{g_y}, S_{g_z})$ are scale factor errors, and $\boldsymbol{\epsilon}_g$ is a white noise stochastic process. The nonorthogonality and axes misalignment errors, scale factor errors, and bias parameters are all modeled as zero-mean, Gaussian-distributed random constants with appropriate covariances. The noise $\boldsymbol{\epsilon}_g$ is modeled as a zero-mean, Gaussian-distributed, time-wise uncorrelated random process, with

$$E(\boldsymbol{\epsilon}_g(t)\boldsymbol{\epsilon}_g^T(\tau)) = \mathbf{V}_g(t)\delta(t-\tau).$$

To first-order, we have

$$(I + \mathbf{S}_g)(\mathbf{I} + \boldsymbol{\Gamma}_g) \approx \mathbf{I} + \mathbf{S}_g + \boldsymbol{\Gamma}_g \ .$$

Hence, Eq. (11) can be written in the form

$$\boldsymbol{\omega}_m = (\mathbf{I} + \boldsymbol{\Delta}_g)(\boldsymbol{\omega} + \mathbf{b}_g + \boldsymbol{\epsilon}_g), \tag{13}$$

where $\boldsymbol{\Delta}_g := \mathbf{S}_g + \boldsymbol{\Gamma}_g$.

State Estimation Errors

Suppose that we have available IMU measurements of nongravitational acceleration and of angular velocity. Then the estimated vehicle state at time t is obtained by numerically integrating over the interval $[t_0, t]$ the following equations:

$$\begin{aligned} \dot{\hat{\mathbf{r}}} &= \hat{\mathbf{v}} \\ \dot{\hat{\mathbf{v}}} &= \mathbf{g}(\hat{\mathbf{r}}) + \mathbf{T}_{IC}(\hat{\mathbf{Q}})\mathbf{a}_m^c \\ \dot{\hat{\mathbf{Q}}} &= \frac{1}{2}\mathbf{B}(\boldsymbol{\omega}_m)\hat{\mathbf{Q}} \end{aligned} \tag{14}$$

with the initial conditions

$$\hat{\mathbf{r}}(t_0) = \hat{\mathbf{r}}_0, \quad \hat{\mathbf{v}}(t_0) = \hat{\mathbf{v}}_0, \quad \hat{\mathbf{Q}}(t_0) = \hat{\mathbf{Q}}_0.$$

Integration of the navigation equations given in Eq. (14) yields the navigated spacecraft position, velocity, and attitude. Since dead-reckoning is essentially an *open-loop* estimation process, the accuracy of the navigated state depends strongly on knowledge of the initial spacecraft state. Also, any measurement errors present in $\boldsymbol{\omega}_m$ and $\mathbf{a}_m^c$ will corrupt the navigation solution.

The estimation errors associated with the dead-reckoning navigation solution is comprised of the attitude estimation errors and the position and velocity estimation errors. As can be seen in Eq. (14), integration of the position and velocity equations requires a transformation of the IMU accelerations from the case frame to the inertial frame, which depends, in turn, on the spacecraft attitude estimate. Therefore, any estimation error in the attitude estimate naturally couples into the position and velocity navigation. The attitude estimation does not rely on the position and velocity estimation, hence can be addressed independently.

Attitude Estimation Errors

Define the attitude error, $\mathbf{e}_g \in R^4$, as

$$\mathbf{e}_g := \mathbf{Q} - \hat{\mathbf{Q}} \ . \tag{15}$$

Computing $\dot{\mathbf{e}}_g$ yields

$$\dot{\mathbf{e}}_g = \dot{\mathbf{Q}} - \dot{\hat{\mathbf{Q}}} = \frac{1}{2}\mathbf{A}(\mathbf{Q})\boldsymbol{\omega} - \frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}})\boldsymbol{\omega}_m,$$

where the matrix $\mathbf{A}$ is given in Eq. (7). This can also be written as

$$\dot{\mathbf{e}}_g = \frac{1}{2}\mathbf{B}(\boldsymbol{\omega}_m)\mathbf{e}_g + \frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}})(\boldsymbol{\omega} - \boldsymbol{\omega}_m) + \cdots,$$

where the matrix $\mathbf{B}$ is given in Eq. (5). It then follows from Eq. (13) that

$$\dot{\mathbf{e}}_g = \frac{1}{2}\mathbf{B}(\boldsymbol{\omega}_m)\mathbf{e}_g + \frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}})[(I + \boldsymbol{\Delta}_g)^{-1}\boldsymbol{\omega}_m - \mathbf{b}_g - \boldsymbol{\epsilon}_g - \boldsymbol{\omega}_m] \ .$$

From the matrix inversion lemma, we have the general result that

$$(\mathbf{I} + \boldsymbol{\Delta})^{-1} = \mathbf{I} - \boldsymbol{\Delta}(\mathbf{I} + \boldsymbol{\Delta})^{-1}.$$

Hence, assuming "small" $\boldsymbol{\Delta}_g$ yields

$$(\mathbf{I} + \boldsymbol{\Delta}_g)^{-1} \approx \mathbf{I} - \boldsymbol{\Delta}_g.$$

To first-order, it then follows that

$$\dot{\mathbf{e}}_g = \frac{1}{2}\mathbf{B}(\boldsymbol{\omega}_m)\mathbf{e}_g - \frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}})\left[\boldsymbol{\Delta}_g\boldsymbol{\omega}_m + \mathbf{b}_g + \boldsymbol{\epsilon}_g\right] .$$

With the given definition of $\mathbf{S}_g$ in Eq. (12), we compute

$$\begin{aligned}\mathbf{S}_g\boldsymbol{\omega}_m &= \begin{bmatrix} S_{gx}\omega_{m_x} \\ S_{gy}\omega_{m_y} \\ S_{gz}\omega_{m_z} \end{bmatrix} \\ &= \begin{bmatrix} \omega_{m_x} & 0 & 0 \\ 0 & \omega_{m_y} & 0 \\ 0 & 0 & \omega_{m_z} \end{bmatrix} \begin{bmatrix} S_{gx} \\ S_{gy} \\ S_{gz} \end{bmatrix} .\end{aligned}$$

Therefore, we can write

$$\mathbf{S}_g\boldsymbol{\omega}_m = \mathbf{N}_1(\omega_m)\mathbf{s}_g,$$

where

$$\mathbf{s}_g := \begin{bmatrix} S_{gx} \\ S_{gy} \\ S_{gz} \end{bmatrix}$$

and

$$\mathbf{N}_1(\omega_m) := \begin{bmatrix} \omega_{m_x} & 0 & 0 \\ 0 & \omega_{m_y} & 0 \\ 0 & 0 & \omega_{m_z} \end{bmatrix} .$$

Similarly, we can write

$$\boldsymbol{\Gamma}_g\boldsymbol{\omega}_m = \mathbf{N}_2(\omega_m)\boldsymbol{\gamma}_g \ ,$$

where

$$\boldsymbol{\gamma}_g := \begin{bmatrix} \gamma_{g_{xy}} \\ \gamma_{g_{xz}} \\ \gamma_{g_{yx}} \\ \gamma_{g_{yz}} \\ \gamma_{g_{zx}} \\ \gamma_{g_{zy}} \end{bmatrix}$$

and

$$\mathbf{N}_2(\omega_m) := \left[\begin{array}{cccccc} -\omega_{m_z} & \omega_{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{m_z} & -\omega_{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_{m_y} & \omega_{m_x} \end{array} \right] .$$

Therefore, it follows that

$$\begin{aligned}\dot{\mathbf{e}}_g &= \frac{1}{2}\mathbf{B}(\boldsymbol{\omega}_m)\mathbf{e}_g - \frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}})\left[\mathbf{N}_1(\omega_m)\mathbf{s}_g\right. \\ &+ \left.\mathbf{N}_2(\omega_m)\boldsymbol{\gamma}_g + \mathbf{b}_g + \boldsymbol{\epsilon}_g\right] . \qquad (16)\end{aligned}$$

Position and Velocity Estimation Errors

The position and velocity estimation error are defined to be

$$\mathbf{e}_r := \mathbf{r} - \hat{\mathbf{r}} \qquad \text{and} \qquad \mathbf{e}_v := \mathbf{v} - \hat{\mathbf{v}}. \qquad (17)$$

Computing the time-derivative of $\mathbf{e}_r$ and $\mathbf{e}_v$ yields, respectively,

$$\dot{\mathbf{e}}_r = \mathbf{e}_v \qquad (18)$$

and

$$\dot{\mathbf{e}}_v = \mathbf{g}(\mathbf{r}) - \mathbf{g}(\hat{\mathbf{r}}) + \mathbf{T}_{IC}(\boldsymbol{Q})\mathbf{a}^c - \mathbf{T}_{IC}(\hat{\boldsymbol{Q}})\mathbf{a}_m^c. \qquad (19)$$

Expanding gravity, utilizing a Taylor series and neglecting higher order terms, it follows that

$$\mathbf{g}(\mathbf{r}) - \mathbf{g}(\hat{\mathbf{r}}) = \left.\frac{\partial\mathbf{g}}{\partial\mathbf{r}}\right|_{\mathbf{r}=\hat{\mathbf{r}}}(\mathbf{r} - \hat{\mathbf{r}}) = \mathbf{G}(\hat{\mathbf{r}})\mathbf{e}_r,$$

where

$$\mathbf{G}(\hat{\mathbf{r}}) := \left.\frac{\partial\mathbf{g}}{\partial\mathbf{r}}\right|_{\mathbf{r}=\hat{\mathbf{r}}} .$$

Therefore, using Eq. (19) and neglecting higher-order terms, we obtain the relationship

$$\dot{\mathbf{e}}_v = \mathbf{G}(\hat{\mathbf{r}})\mathbf{e}_r + \mathbf{T}_{IC}(\boldsymbol{Q})\mathbf{a}^c - \mathbf{T}_{IC}(\hat{\boldsymbol{Q}})\mathbf{a}_m^c. \qquad (20)$$

Expanding the transformation matrix, $\mathbf{T}_{IC}$, in a Taylor series yields

$$\begin{aligned}\mathbf{T}_{IC}(\boldsymbol{Q}) &= \mathbf{T}_{IC}(\hat{\boldsymbol{Q}}) + \sum_{i=1}^{4}\left.\frac{\partial\mathbf{T}_{IC}}{\partial q_i}\right|_{\mathbf{Q}=\hat{\mathbf{Q}}}(q_i - \hat{q}_i) \\ &+ \cdots = \mathbf{T}_{IC}(\hat{\boldsymbol{Q}}) + \boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q) + \cdots ,\end{aligned}$$

where

$$\boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q) := \sum_{i=1}^{4}\left.\frac{\partial\mathbf{T}_{IC}}{\partial q_i}\right|_{\mathbf{Q}=\hat{\mathbf{Q}}}(q_i - \hat{q}_i). \qquad (21)$$

Using the transformation matrix $\mathbf{T}_{IC}$ given in Eq. (3), we compute

$$\begin{aligned}\frac{\partial\mathbf{T}_{IC}}{\partial q_1} &= \begin{bmatrix} 0 & 2q_2 & 2q_3 \\ 2q_2 & -4q_1 & 2q_4 \\ 2q_3 & -2q_4 & -4q_1 \end{bmatrix} \\ \frac{\partial\mathbf{T}_{IC}}{\partial q_2} &= \begin{bmatrix} -4q_2 & 2q_1 & -2q_4 \\ 2q_1 & 0 & 2q_3 \\ 2q_4 & 2q_3 & -4q_2 \end{bmatrix} \\ \frac{\partial\mathbf{T}_{IC}}{\partial q_3} &= \begin{bmatrix} -4q_3 & 2q_4 & 2q_1 \\ -2q_4 & -4q_3 & 2q_2 \\ 2q_1 & 2q_2 & 0 \end{bmatrix} \\ \frac{\partial\mathbf{T}_{IC}}{\partial q_4} &= \begin{bmatrix} 0 & 2q_3 & -2q_2 \\ -2q_3 & 0 & 2q_1 \\ 2q_2 & -2q_1 & 0 \end{bmatrix} .\end{aligned}$$

Substituting Eq. (21) into Eq. (20) and neglecting higher-order terms in the Taylor series expansion of $\mathbf{T}_{IC}$ yields

$$\begin{aligned} \dot{\mathbf{e}}_v &= \mathbf{G}(\hat{\mathbf{r}})\mathbf{e}_r + \mathbf{T}_{IC}(\hat{\boldsymbol{Q}})(\mathbf{a}^c - \mathbf{a}^c_m) \\ &+ \boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q)\mathbf{a}^c. \end{aligned} \tag{22}$$

Using the accelerometer model given in Eq. (10), we find that the third term on the right hand-side of Eq. (22) is

$$\boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q)\mathbf{a}^c = \boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q)\mathbf{a}^c_m - \boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q)\left[\mathbf{b}_a + \boldsymbol{\epsilon}_a + \boldsymbol{\Delta}_a\left(\mathbf{a}^c + \mathbf{b}_a + \boldsymbol{\epsilon}_a\right)\right].$$

Hence, to first-order, we have

$$\boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q)\mathbf{a}^c \approx \boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q)\mathbf{a}^c_m,$$

and Eq. (22) reduces to

$$\begin{aligned} \dot{\mathbf{e}}_v &= \mathbf{G}(\hat{\mathbf{r}})\mathbf{e}_r + \mathbf{T}_{IC}(\hat{\boldsymbol{Q}})(\mathbf{a}^c - \mathbf{a}^c_m) \\ &+ \boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q)\mathbf{a}^c_m. \end{aligned} \tag{23}$$

Careful analysis of the third term on the right hand-side of Eq. (23) yields the following convenient representation:

$$\boldsymbol{\Delta}_e(\hat{\mathbf{Q}}, \mathbf{e}_q)\mathbf{a}^c_m = \mathbf{R}(\hat{\mathbf{Q}}, \mathbf{a}^c_m)\mathbf{e}_q,$$

where $\mathbf{R}(\hat{\mathbf{Q}}, \mathbf{a}^c_m) \in \mathcal{R}^{3\times 4}$ is given by

$$\mathbf{R}(\hat{\mathbf{Q}}, \mathbf{a}^c_m) := \Big[-4\left[\hat{\mathbf{q}} * \mathbf{a}^c_m\right]^\times - 2\left[\hat{\mathbf{q}} \otimes \mathbf{a}^c_m\right] + 2\hat{q}_4\left[\mathbf{a}^c_m\right]^\times + 2\left[\hat{\mathbf{q}} \odot \mathbf{a}^c_m\right]\mathbf{I} \quad \Big| -2\hat{\mathbf{q}} * \mathbf{a}^c_m \Big], \tag{24}$$

and where $\odot$ is the inner product, $\otimes$ is the outer product, and $[\cdot]^\times$ denotes the cross-product matrix.

Rearranging the terms in the acceleration model given in Eq. (10) yields

$$\mathbf{a}^c = (\mathbf{I} + \boldsymbol{\Delta}_a)^{-1}\mathbf{a}^c_m - \mathbf{b}_a - \boldsymbol{\epsilon}_a$$

where, after some manipulation and using the fact that for "small" $\boldsymbol{\Delta}_a$ we have $(\mathbf{I} + \boldsymbol{\Delta}_a)^{-1} \approx \mathbf{I} - \boldsymbol{\Delta}_a$, we obtain

$$\mathbf{a}^c - \mathbf{a}^c_m = -\boldsymbol{\Delta}_a\mathbf{a}^c_m - \mathbf{b}_a - \boldsymbol{\epsilon}_a. \tag{25}$$

Eq. (23) can thus be rewritten as

$$\begin{aligned} \dot{\mathbf{e}}_v &= \mathbf{G}(\hat{\mathbf{r}})\mathbf{e}_r - \mathbf{T}_{IC}(\hat{\boldsymbol{Q}})(\boldsymbol{\Delta}_a\mathbf{a}^c_m + \mathbf{b}_a + \boldsymbol{\epsilon}_a) \\ &+ \mathbf{R}(\hat{\mathbf{Q}}, \mathbf{a}^c_m)\mathbf{e}_q. \end{aligned} \tag{26}$$

Keep in mind that in Eq. (26), the matrix $\boldsymbol{\Delta}_a$ is comprised of random constants, $\mathbf{b}_a$ is a random constant vector, $\boldsymbol{\epsilon}_a$ is a random process, and $\mathbf{e}_q$ is the attitude estimation error that contributes directly to the uncertainty in the position (through integration) and velocity estimation errors.

Consider the term $\boldsymbol{\Delta}_a\mathbf{a}^c_m$ more closely. From the definition of $\boldsymbol{\Delta}_a$ given in Eq. (9) we have

$$\boldsymbol{\Delta}_a\mathbf{a}^c_m = \left(\boldsymbol{\Upsilon}_a + \boldsymbol{\Xi}_a\right)\mathbf{a}^c_m.$$

With the definitions of $\boldsymbol{\Upsilon}_a$ and $\boldsymbol{\Xi}_a$ given in Eq. (8), we find that we can also write $\boldsymbol{\Delta}_a\mathbf{a}^c_m$ as

$$\boldsymbol{\Delta}_a\mathbf{a}^c_m = \mathbf{M}_1(\mathbf{a}^c_m)\boldsymbol{\xi}_a + \mathbf{M}_2(\mathbf{a}^c_m)\boldsymbol{\upsilon}_a, \tag{27}$$

where

$$\mathbf{M}_1(\mathbf{a}^c_m) := \begin{bmatrix} a^c_{m_x} & 0 & 0 \\ 0 & a^c_{m_y} & 0 \\ 0 & 0 & a^c_{m_z} \end{bmatrix}$$

$$\mathbf{M}_2(\mathbf{a}^c_m) := \begin{bmatrix} -a^c_{m_z} & a^c_{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & a^c_{m_z} & -a^c_{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -a^c_{m_y} & a^c_{m_x} \end{bmatrix}$$

$$\boldsymbol{\upsilon}_a := \begin{bmatrix} \upsilon_{a_{xy}} \\ \upsilon_{a_{xz}} \\ \upsilon_{a_{yx}} \\ \upsilon_{a_{yz}} \\ \upsilon_{a_{zx}} \\ \upsilon_{a_{zy}} \end{bmatrix}, \text{ and } \boldsymbol{\xi}_a := \begin{bmatrix} \xi_{a_x} \\ \xi_{a_y} \\ \xi_{a_z} \end{bmatrix}.$$

Substituting Eq. (27) into Eq. (26) yields

$$\dot{\mathbf{e}}_v = \mathbf{G}(\hat{\mathbf{r}})\mathbf{e}_r - \mathbf{T}_{IC}(\hat{\boldsymbol{Q}})\left[\mathbf{M}_1(\mathbf{a}^c_m)\boldsymbol{\xi}_a + \mathbf{M}_2(\mathbf{a}^c_m)\boldsymbol{\upsilon}_a + \mathbf{b}_a + \boldsymbol{\epsilon}_a\right] + \mathbf{R}(\hat{\mathbf{Q}}, \mathbf{a}^c_m)\mathbf{e}_q. \tag{28}$$

Collecting the position, velocity, and attitude estimation error equations from Eqs. (16), (18), and (28), and writing in matrix form yields the stochastic linear matrix differential equation

$$\dot{\mathbf{e}} = \mathbf{F}\mathbf{e} + \mathbf{H}_1\mathbf{w} + \mathbf{H}_2\mathbf{v}, \tag{29}$$

where

$$\mathbf{e} := \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_v \\ \mathbf{e}_g \end{bmatrix} \in R^{10}, \qquad \mathbf{w} := \begin{bmatrix} \boldsymbol{\xi}_a \\ \boldsymbol{\upsilon}_a \\ \mathbf{b}_a \\ \mathbf{s}_g \\ \boldsymbol{\gamma}_g \\ \mathbf{b}_g \end{bmatrix} \in R^{24}$$

$$\mathbf{v} := \begin{bmatrix} \boldsymbol{\epsilon}_a \\ \boldsymbol{\epsilon}_g \end{bmatrix} \in R^6, \ \mathbf{e} := \begin{bmatrix} \mathbf{e}_r \\ \mathbf{e}_v \\ \mathbf{e}_g \end{bmatrix} \in R^{10}$$

$$\mathbf{w} := \begin{bmatrix} \boldsymbol{\xi}_a \\ \boldsymbol{\upsilon}_a \\ \mathbf{b}_a \\ \mathbf{s}_g \\ \boldsymbol{\gamma}_g \\ \mathbf{b}_g \end{bmatrix} \in R^{24}, \ \mathbf{v} := \begin{bmatrix} \boldsymbol{\epsilon}_a \\ \boldsymbol{\epsilon}_g \end{bmatrix} \in R^{6}.$$

The error state matrix $\mathbf{F} \in R^{10\times 10}$ is

$$\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{G}(\hat{\mathbf{r}}) & \mathbf{0} & \mathbf{R}(\hat{\mathbf{Q}}, \mathbf{a}_m^c) \\ \mathbf{0} & \mathbf{0} & \frac{1}{2}\mathbf{B}(\boldsymbol{\omega}_m) \end{bmatrix},$$

and the input mapping matrices $\mathbf{H}_1 \in R^{10\times 24}$ and $\mathbf{H}_2 \in R^{10\times 6}$ are

$$\mathbf{H}_1 := \left[\begin{array}{c|c} \multicolumn{2}{c}{\cdots \ \mathbf{0} \ \cdots} \\ \mathbf{H}_{1a} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{1b} \end{array} \right].$$

where

$$\mathbf{H}_{1a} = -\mathbf{T}_{IC}(\hat{\mathbf{Q}}) \begin{bmatrix} \mathbf{M}_1(\mathbf{a}_m^c) & \mathbf{M}_2(\mathbf{a}_m^c) & \mathbf{I} \end{bmatrix}$$

$$\mathbf{H}_{1b} = -\frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}}) \begin{bmatrix} \mathbf{N}_1(\boldsymbol{\omega}_m) & \mathbf{N}_2(\boldsymbol{\omega}_m) & \mathbf{I} \end{bmatrix}$$

and

$$\mathbf{H}_2 := \left[\begin{array}{c|c} \multicolumn{2}{c}{\cdots\cdots \ \mathbf{0} \ \cdots\cdots} \\ -\mathbf{T}_{IC}(\hat{\mathbf{Q}}) & \mathbf{0} \\ \mathbf{0} & -\frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}}) \end{array} \right].$$

The components of $\mathbf{w}$ in Eq. (29) are the various random constant errors associated with the IMU, where it assumed that

$$E[\mathbf{w}] = \mathbf{0}$$

and the block diagonal covariance matrix, $\mathbf{W} \in R^{24\times 24}$, has given weights

$$\begin{aligned} &\mathbf{W} := E\left[\mathbf{w}\mathbf{w}^T\right] \\ &= \operatorname{diag}\left(\mathbf{W}_{\xi_a}, \mathbf{W}_{\upsilon_a}, \mathbf{W}_{b_a}, \mathbf{W}_{s_g}, \mathbf{W}_{\gamma_g}, \mathbf{W}_{b_g}\right). \end{aligned}$$

The components of $\mathbf{v}(t)$ in Eq. (29) are the random components of the IMU errors, where it is assumed that

$$E[\mathbf{v}(t)] = \mathbf{0}$$

and

$$\begin{aligned} E\left[\mathbf{v}(t)\mathbf{v}(\tau)^T\right] &= \begin{bmatrix} \mathbf{V}_a & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_g \end{bmatrix} \delta(t-\tau) \\ &= \mathbf{V}(t)\delta(t-\tau). \end{aligned}$$

Estimation Error Covariance

Define the state transition matrix $\boldsymbol{\Phi}(t, t_0) \in R^{10\times 10}$ associated with $\mathbf{F}$ as the solution to the matrix differential equation

$$\dot{\boldsymbol{\Phi}}(t, t_0) = \mathbf{F}\boldsymbol{\Phi}(t, t_0), \tag{30}$$

where

$$\boldsymbol{\Phi}(t_0, t_0) = \mathbf{I}.$$

The solution to Eq. (29) is

$$\begin{aligned} \mathbf{e}(t) &= \boldsymbol{\Phi}(t, t_0)\mathbf{e}(t_0) + \left[\int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\mathbf{H}_1(\tau)d\tau\right]\mathbf{w} \\ &+ \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\mathbf{H}_2(\tau)\mathbf{v}(\tau)d\tau, \end{aligned} \tag{31}$$

and with the estimation error covariance, $\mathbf{P}(t)$, defined as

$$\mathbf{P}(t) = E\left(\mathbf{e}(t)\mathbf{e}^T(t)\right), \tag{32}$$

we can compute $\mathbf{P}(t)$ using Eqs. (30)–(32) yielding

$$\begin{aligned} \mathbf{P}(t) &= \boldsymbol{\Phi}(t, t_0)\mathbf{P}(t_0)\boldsymbol{\Phi}^T(t, t_0) \\ &+ \mathbf{V}_t + \mathbf{W}_t\mathbf{W}\mathbf{W}_t^T \end{aligned} \tag{33}$$

where

$$\begin{aligned} \mathbf{V}_t &:= \int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\mathbf{H}_2(\tau)\mathbf{V}(\tau)\mathbf{H}_2^T(\tau)\boldsymbol{\Phi}^T(t,\tau)d\tau \\ \mathbf{W}_t &:= \left[\int_{t_0}^{t} \boldsymbol{\Phi}(t,\tau)\mathbf{H}_1(\tau)d\tau\right]. \end{aligned} \tag{34}$$

Then, taking the time-derivative of $\mathbf{V}_t$ and $\mathbf{W}_t$ yields

$$\begin{aligned} \dot{\mathbf{V}}_t = \mathbf{F}(t)\mathbf{V}_t &+ \mathbf{V}_t\mathbf{F}^T(t) \\ &+ \mathbf{H}_2(t)\mathbf{V}(t)\mathbf{H}_2^T(t), \end{aligned} \tag{35}$$

and

$$\dot{\mathbf{W}}_t = \mathbf{F}(t)\mathbf{W}_t + \mathbf{H}_1(t), \tag{36}$$

respectively. The appropriate initial conditions are $\mathbf{V}_t(t_0) = \mathbf{0}$ and $\mathbf{W}_t(t_0) = \mathbf{0}$. Once the values of $\boldsymbol{\Phi}(t, t_0)$, $\mathbf{V}_t$ and $\mathbf{W}_t$ have been computed via integration of Eqs. (30), (35) and (36) from t_0 to t, the error covariance $\mathbf{P}(t_0)$ at time t_0 is mapped forward to time t via Eq. (33).

Dead Reckoning Navigation

Suppose that the time history of the IMU observations, that is, $\mathbf{a}_m^c(t)$ and $\boldsymbol{\omega}_m(t)$ are continuously available. Then, dead reckoning navigation, including computing the associated state estimation error covariance, is the process of integrating over the interval $[t_0, t]$ the following equations:

$$\begin{aligned}
\dot{\hat{\mathbf{r}}} &= \hat{\mathbf{v}} \\
\dot{\hat{\mathbf{v}}} &= \hat{\mathbf{g}} + \mathbf{T}_{IC}\mathbf{a}_m^c \\
\dot{\hat{\mathbf{Q}}} &= \frac{1}{2}\mathbf{B}\hat{\mathbf{Q}} \\
\dot{\mathbf{\Phi}} &= \mathbf{F}\mathbf{\Phi} \\
\dot{\mathbf{W}}_t &= \mathbf{F}\mathbf{W}_t + \mathbf{H}_1 \\
\dot{\mathbf{V}}_t &= \mathbf{F}\mathbf{V}_t + \mathbf{V}_t\mathbf{F}^T + \mathbf{H}_2\mathbf{V}\mathbf{H}_2^T,
\end{aligned} \qquad (37)$$

and the estimation error covariance at time t_0 is mapped forward to time t via

$$\mathbf{P}(t) = \mathbf{\Phi}\mathbf{P}_0\mathbf{\Phi}^T + \mathbf{V}_t + \mathbf{W}_t\mathbf{W}\mathbf{W}_t^T,$$

where $\hat{\mathbf{g}} := \mathbf{g}(\hat{\mathbf{r}})$ is the modeled gravity, $\hat{\mathbf{Q}} = \left[\hat{\mathbf{q}}^T \mid \hat{q}_4\right]^T = [\hat{q}_1\ \hat{q}_2\ \hat{q}_3\ \hat{q}_4]^T$ and

$$\mathbf{A}(\hat{\mathbf{Q}}) := \begin{bmatrix} \hat{q}_4 & \hat{q}_3 & \hat{q}_2 \\ \hat{q}_3 & \hat{q}_4 & -\hat{q}_1 \\ -\hat{q}_2 & \hat{q}_1 & \hat{q}_4 \\ -\hat{q}_1 & -\hat{q}_2 & -\hat{q}_3 \end{bmatrix}$$

$$\mathbf{B}(\boldsymbol{\omega}_m) := \begin{bmatrix} 0 & \omega_{m_z} & -\omega_{m_y} & \omega_{m_x} \\ -\omega_{m_z} & 0 & \omega_{m_x} & \omega_{m_y} \\ \omega_{m_y} & -\omega_{m_x} & 0 & \omega_{m_z} \\ -\omega_{m_x} & -\omega_{m_y} & -\omega_{m_z} & 0 \end{bmatrix},$$

$$\mathbf{T}_{IC}(\hat{\mathbf{Q}}) := \begin{bmatrix} 1-2(\hat{q}_2^2+\hat{q}_3^2) & 2(\hat{q}_1\hat{q}_2+\hat{q}_3\hat{q}_4) & 2(\hat{q}_1\hat{q}_3-\hat{q}_2\hat{q}_4) \\ 2(\hat{q}_1\hat{q}_2-\hat{q}_3\hat{q}_4) & 1-2(\hat{q}_1^2+\hat{q}_3^2) & 2(\hat{q}_2\hat{q}_3+\hat{q}_1\hat{q}_4) \\ 2(\hat{q}_1\hat{q}_3+\hat{q}_2\hat{q}_4) & 2(\hat{q}_2\hat{q}_3-\hat{q}_1\hat{q}_4) & 1-2(\hat{q}_1^2+\hat{q}_2^2) \end{bmatrix},$$

$$\mathbf{F}(\hat{\mathbf{r}}, \hat{\mathbf{Q}}, \boldsymbol{\omega}_m, \mathbf{a}_m^c(t)) := \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{G}(\hat{\mathbf{r}}) & \mathbf{0} & \mathbf{R}(\hat{\mathbf{Q}}, \mathbf{a}_m^c) \\ \mathbf{0} & \mathbf{0} & \frac{1}{2}\mathbf{B}(\boldsymbol{\omega}_m) \end{bmatrix},$$

$$\mathbf{R}(\hat{\mathbf{Q}}, \mathbf{a}_m^c) = \Big[-4\left[\hat{\mathbf{q}} * \mathbf{a}_m^c\right]^{\times} - 2\left[\hat{\mathbf{q}} \otimes \mathbf{a}_m^c\right] + 2\hat{q}_4\left[\mathbf{a}_m^c\right]^{\times} + 2\left[\hat{\mathbf{q}} \odot \mathbf{a}_m^c\right]\mathbf{I} \;\Big|\; -2\hat{\mathbf{q}} * \mathbf{a}_m^c\Big],$$

$$\mathbf{H}_1(\mathbf{a}_m^c, \boldsymbol{\omega}_m, \hat{\mathbf{Q}}) = \begin{bmatrix} \cdots & \mathbf{0} & \cdots \\ \mathbf{H}_{1a} & \vert & \mathbf{0} \\ \mathbf{0} & \vert & \mathbf{H}_{1b} \end{bmatrix}.$$

$$\mathbf{H}_{1a}(\mathbf{a}_m^c, \hat{\mathbf{Q}}) = -\mathbf{T}_{IC}(\hat{\mathbf{Q}}) \begin{bmatrix} \mathbf{M}_1(\mathbf{a}_m^c) & \mathbf{M}_2(\mathbf{a}_m^c) & \mathbf{I} \end{bmatrix}$$

$$\mathbf{H}_{1b}(\boldsymbol{\omega}_m) = -\frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}}) \begin{bmatrix} \mathbf{N}_1(\boldsymbol{\omega}_m) & \mathbf{N}_2(\boldsymbol{\omega}_m) & \mathbf{I} \end{bmatrix}$$

$$\mathbf{H}_2(\hat{\mathbf{Q}}) = \begin{bmatrix} \cdots\cdots & \mathbf{0} & \cdots\cdots \\ -\mathbf{T}_{IC}(\hat{\mathbf{Q}}) & \vert & \mathbf{0} \\ \mathbf{0} & \vert & -\frac{1}{2}\mathbf{A}(\hat{\mathbf{Q}}) \end{bmatrix},$$

$$\mathbf{M}_1(\mathbf{a}_m^c) = \begin{bmatrix} a_{m_x}^c & 0 & 0 \\ 0 & a_{m_y}^c & 0 \\ 0 & 0 & a_{m_z}^c \end{bmatrix}$$

$$\mathbf{M}_2(\mathbf{a}_m^c) = \begin{bmatrix} -a_{m_z}^c & a_{m_y}^c & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{m_z}^c & -a_{m_x}^c & 0 & 0 \\ 0 & 0 & 0 & 0 & -a_{m_y}^c & a_{m_x}^c \end{bmatrix},$$

$$\mathbf{N}_1(\omega_m) = \begin{bmatrix} \omega_{m_x} & 0 & 0 \\ 0 & \omega_{m_y} & 0 \\ 0 & 0 & \omega_{m_z} \end{bmatrix}$$

$$\mathbf{N}_2(\omega_m) = \begin{bmatrix} -\omega_{m_z} & \omega_{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & \omega_{m_z} & -\omega_{m_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\omega_{m_y} & \omega_{m_x} \end{bmatrix},$$

with the initial conditions

$$\hat{\mathbf{r}}(t_0) = \hat{\mathbf{r}}_0,\ \hat{\mathbf{v}}(t_0) = \hat{\mathbf{v}}_0,\ \hat{\mathbf{Q}}(t_0) = \hat{\mathbf{Q}}_0,\ \mathbf{W}_t(t_0) = \mathbf{0}$$

$$\mathbf{\Phi}(t_0, t_0) = \mathbf{I},\ \mathbf{P}(t_0) = \mathbf{P}_0,\ \mathbf{V}_t(t_0) = \mathbf{0}.$$

The sensor models are assumed known *apriori* and represented by the matrices $\mathbf{V}(t)$ and $\mathbf{W}$.

Simulation Results

The linear covariance formulation thus described has been incorporated into the NASA JSC SORT simulation program[5]. Verification of the formulation is made through judicious use of *monte carlo* analysis with SORT. In the section, we compare the sampled estimation error covariance obtained through *m*onte carlo analysis with the linear covariance formulation. A set of 1000 initial estimation errors were utilized in the analysis to compute the sample estimation error covariance for comparison with the approximate analytic error covariance developed in the previous sections.

The entry scenario is consistent with a Mars 2005 entry. The IMU is modeled as a Honeywell MIMU with the following parameters: **Gyro Errors**: Bias stability $\mathbf{b}_g$=0.05 deg/hr, random walk $\mathbf{V}_g$=0.01 deg/$\sqrt{\text{hr}}$, scale factor $\mathbf{S}_g$=5 ppm, nonorthogonality $\mathbf{\Gamma}_g$=5 arcsec; and **Accelerometer Errors**: Bias $\mathbf{b}_a$=0.1 mg, bias Noise $\mathbf{V}_a$=10 μg, scale factor $\mathbf{\Xi}_a$=175 ppm, nonorthogonality $\mathbf{\Upsilon}_a$=5 ppm.

Figure 4 shows the estimation error uncertainties for position. It can be seen that the approximate error covariance computed via the dead-reckoning equations presented in the previous section matches quite well the estimation error uncertainty computed by navigating with dead-reckoning using the 1000 initial state estimation errors. Similar results are obtained for the velocity estimation accuracies. The excellent match depicted in Fig. 4 gives us confidence that if the dead-reckoning navigation system were in fact supplemented with the estimation error covariance computations proposed in this paper, then at parachute deploy the Kalman filter would have an initial error covariance that reflects reality. Hence, we would expect that the Kalman filter would process the altimetry in a more precise fashion than would be achieved with an i-loaded initial covariance.

Summary and Future Directions

In the paper, the algorithms for dead reckoning navigation were derived to include the state estimation error covariance computation. It was assumed that a strapdown IMU provides a continuous stream of measured accelerations (in the IMU case frame) and relative angular velocity of the IMU case frame to the navigation frame, taken to be a planet-centered inertial reference frame. The underlying error equations were linearized and utilized to develop a formulation of the approximate state estimation error covariance. The correlation of attitude errors with position and velocity errors was explicitly derived. The resulting set of dead reckoning relationships can be used as an independent verification of *monte carlo* analysis during the verification of the entry filter.

The key issues yet to be addressed are as follows:

1. The effects of the IMU offset from the spacecraft center of mass needs to be included in the formulation of the state estimation error covariance.

2. The IMU measurements are not provided continuously as assumed here. Instead, a sampled version of the measurements are delivered to the navigation process. The effects of sampling and round-off need to be considered and directly incorporated into the dead reckoning formulation.

3. Navigation during the next two phases of the EDL (i.e., parachute phase and terminal guidance phase) need to be integrated with the dead-reckoning phase to insure smooth transition and filter stability upon initial incorporation of altimetry data.

Acknowledgements

The work of the second author is supported under NASA Contract NAG9-1225. Special thanks to Dr. Dilmurat Azimov for performing the SORT *monte carlo* simulations.

References

1. P. Daniel Burkhart , Robert H. Bishop and Timothy P. Crain, "Integrated Navigation and Guidance for Precision Landing at Mars," 16th International Symposium on Space Flight Dynamics, Pasadena, CA, 2001.

2. Britting, K., *Inertial Navigation Systems Analysis*, John Wiley & Sons, 1971.

3. Savage, P. G., *Strapdown Analytics*, Strapdown Associates, Inc., Maple Plain,Minnesota, 2000.

4. Carpenter, J. Russell, "Delivery of Inertial Measurement Unit Model to Mars Surveyor '01 Atmospheric Flight Team," Report No. EG-97-103, NASA Johnson Space Center, 1997.

5. *Programmer's Guide for the Simulation and Optimization of Rocket Trajectories (SORT) Program*, Report No. LESC-3042, Systems Definition Branch, Systems Engineering Division, NASA Johnson Space Center, 1992.

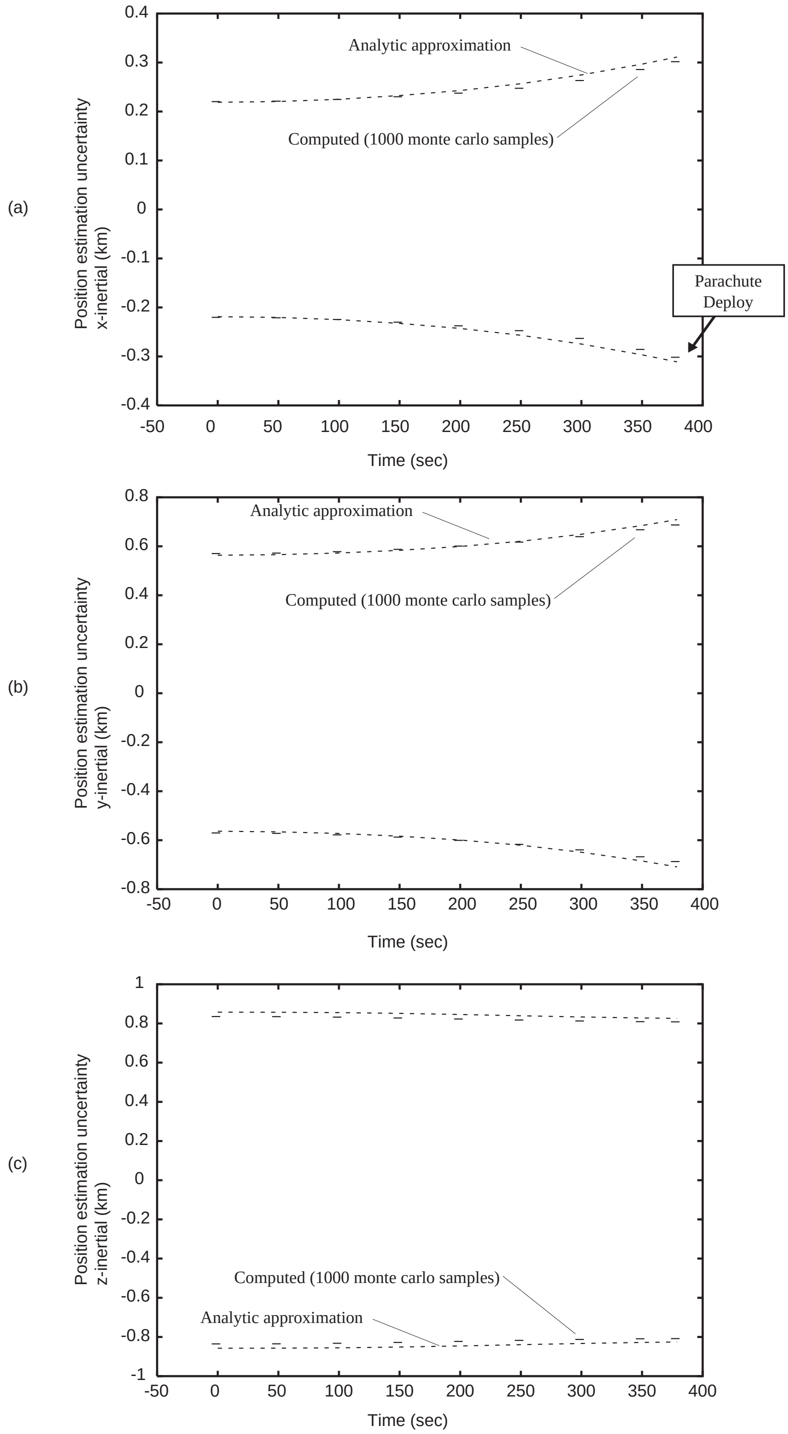

Figure 4: Comparison of the position estimation error for a sampled 1000 run simulation and the approximate analytic error covariance.