

# The new IMU Factor

Frank Dellaert & Varun Agrawal

September 21, 2021

## Navigation States

Let us assume a setup where frames with image and/or laser measurements are processed at some fairly low rate, e.g., 10 Hz.

We define the state of the vehicle at those times as attitude, position, and velocity. These three quantities are jointly referred to as a NavState  $X_b^n \triangleq \{R_b^n, P_b^n, V_b^n\}$ , where the superscript  $n$  denotes the *navigation frame*, and  $b$  the *body frame*. For simplicity, we drop these indices below where clear from context.

## Vector Fields and Differential Equations

We need a way to describe the evolution of a NavState over time. The NavState lives in a 9-dimensional manifold  $M$ , defined by the orthonormality constraints on  $\mathbb{R}$ . For a NavState  $X$  evolving over time we can write down a differential equation

$$\dot{X}(t) = F(t, X) \quad (1)$$

where  $F$  is a time-varying **vector field** on  $M$ , defined as a mapping from  $\mathbb{R} \times M$  to tangent vectors at  $X$ . A **tangent vector** at  $X$  is defined as the derivative of a trajectory at  $X$ , and for the NavState manifold this will be a triplet

$$\left[ \dot{R}(t, X), \dot{P}(t, X), \dot{V}(t, X) \right] \in \mathfrak{so}(3) \times \mathbb{R}^3 \times \mathbb{R}^3$$

where we use square brackets to indicate a tangent vector. The space of all tangent vectors at  $X$  is denoted by  $T_X M$ , and hence  $F(t, X) \in T_X M$ . For example, if the state evolves along a constant velocity trajectory

$$X(t) = \{R_0, P_0 + V_0 t, V_0\}$$

then the differential equation describing the trajectory is

$$\dot{X}(t) = [0_{3 \times 3}, V_0, 0_{3 \times 1}], \quad X(0) = \{R_0, P_0, V_0\}$$

Valid vector fields on a NavState manifold are special, in that the attitude and velocity derivatives can be arbitrary functions of  $X$  and  $t$ , but the derivative of position is constrained to be equal to the current velocity  $V(t)$ :

$$\dot{X}(t) = \left[ \dot{R}(X, t), V(t), \dot{V}(X, t) \right] \quad (2)$$

Suppose we are given the **body angular velocity**  $\omega^b(t)$  and non-gravity **acceleration**  $a^b(t)$  in the body frame. We know (from Murray84book) that the derivative of  $R$  can be written as

$$\dot{R}(X, t) = R(t)[\omega^b(t)]_\times$$

where  $[\theta]_\times \in \mathfrak{so}(3)$  is the skew-symmetric matrix corresponding to  $\theta$ , and hence the resulting exact vector field is

$$\dot{X}(t) = \left[ \dot{R}(X, t), V(t), \dot{V}(X, t) \right] = \left[ R(t)[\omega^b(t)]_\times, V(t), g + R(t)a^b(t) \right] \quad (3)$$

## Local Coordinates

Optimization on manifolds relies crucially on the concept of **local coordinates**. For example, when optimizing over the rotations  $SO(3)$  starting from an initial estimate  $R_0$ , we define a local map  $\Phi_{R_0}$  from  $\theta \in \mathbb{R}^3$  to a neighborhood of  $SO(3)$  centered around  $R_0$ ,

$$\Phi_{R_0}(\theta) = R_0 \exp([\theta]_{\times})$$

where  $\exp$  is the matrix exponential, given by

$$\exp([\theta]_{\times}) = \sum_{k=0}^{\infty} \frac{1}{k!} [\theta]_{\times}^k \quad (4)$$

which for  $SO(3)$  can be efficiently computed in closed form.

The local coordinates  $\theta$  are isomorphic to tangent vectors at  $R_0$ . To see this, define  $\theta = \omega t$  and note that

$$\left. \frac{d\Phi_{R_0}(\omega t)}{dt} \right|_{t=0} = \left. \frac{dR_0 \exp([\omega t]_{\times})}{dt} \right|_{t=0} = R_0 [\omega]_{\times}$$

Hence, the 3-vector  $\omega$  defines a direction of travel on the  $SO(3)$  manifold, but does so in the local coordinate frame defined by  $R_0$ .

A similar story holds in  $SE(3)$ : we define local coordinates  $\xi = [\omega t, vt] \in \mathbb{R}^6$  and a mapping

$$\Phi_{T_0}(\xi) = T_0 \exp \hat{\xi}$$

where  $\hat{\xi} \in \mathfrak{se}(3)$  is defined as

$$\hat{\xi} = \begin{bmatrix} [\omega]_{\times} & v \\ 0 & 0 \end{bmatrix} t$$

and the 6-vectors  $\xi$  are mapped to tangent vectors  $T_0 \hat{\xi}$  at  $T_0$ .

## Derivative of The Local Coordinate Mapping

For the local coordinate mapping  $\Phi_{R_0}(\theta)$  in  $SO(3)$  we can define a  $3 \times 3$  Jacobian  $H(\theta)$  that models the effect of an incremental change  $\delta$  to the local coordinates:

$$\Phi_{R_0}(\theta + \delta) \approx \Phi_{R_0}(\theta) \exp([H(\theta)\delta]_{\times}) = \Phi_{\Phi_{R_0}(\theta)}(H(\theta)\delta) \quad (5)$$

This Jacobian depends only on  $\theta$  and, for the case of  $SO(3)$ , is given by a formula similar to the matrix exponential map,

$$H(\theta) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)!} [\theta]_{\times}^k$$

which can also be computed in closed form. In particular,  $H(0) = I_{3 \times 3}$  at the base  $R_0$ .

## Numerical Integration in Local Coordinates

Inspired by the paper “Lie Group Methods” by Iserles et al. [1], when we have a differential equation on  $SO(3)$ ,

$$\dot{R}(t) = F(R, t), \quad R(0) = R_0 \quad (6)$$

we can transfer it to a differential equation in the 3-dimensional local coordinate space. To do so, we model the solution to (6) as

$$R(t) = \Phi_{R_0}(\theta(t))$$

To find an expression for  $\dot{\theta}(t)$ , create a trajectory  $\gamma(\delta)$  that passes through  $R(t)$  for  $\delta = 0$ , and moves  $\theta(t)$  along the direction  $\dot{\theta}(t)$ :

$$\gamma(\delta) = R(t + \delta) = \Phi_{R_0}(\theta(t) + \dot{\theta}(t)\delta) \approx \Phi_{R(t)}(H(\theta)\dot{\theta}(t)\delta)$$

Taking the derivative for  $\delta = 0$  we obtain

$$\dot{R}(t) = \left. \frac{d\gamma(\delta)}{d\delta} \right|_{\delta=0} = \left. \frac{d\Phi_{R(t)}(H(\theta)\dot{\theta}(t)\delta)}{d\delta} \right|_{\delta=0} = R(t)[H(\theta)\dot{\theta}(t)]_{\times}$$

Comparing this to (6) we obtain a differential equation for  $\theta(t)$ :

$$\dot{\theta}(t) = H(\theta)^{-1} \{R(t)^T F(R, t)\}^{\sim}, \quad \theta(0) = 0_{3 \times 1}$$

In other words, the vector field  $F(R, t)$  is rotated to the local frame, the inverse hat operator is applied to get a 3-vector, which is then corrected by  $H(\theta)^{-1}$  away from  $\theta = 0$ .

## Retractions

Note that the use of the exponential map in local coordinate mappings is not obligatory, even in the context of Lie groups. Often it is computationally expedient to use mappings that are easier to compute, but yet induce the same tangent vector at  $T_0$ . Mappings that satisfy this constraint are collectively known as **retractions**. For example, for  $SE(3)$  one could use the retraction  $\mathcal{R}_{T_0} : \mathbb{R}^6 \rightarrow SE(3)$

$$\mathcal{R}_{T_0}(\xi) = T_0 \{ \exp([\omega t]_{\times}), vt \} = \{ \Phi_{R_0}(\omega t), P_0 + R_0 vt \}$$

This trajectory describes a linear path in position while the frame rotates, as opposed to the helical path traced out by the exponential map. The tangent vector at  $T_0$  can be computed as

$$\left. \frac{d\mathcal{R}_{T_0}(\xi)}{dt} \right|_{t=0} = [R_0[\omega]_{\times}, R_0 v]$$

which is identical to the one induced by  $\Phi_{T_0}(\xi) = T_0 \exp \hat{\xi}$ .

The NavState manifold is not a Lie group like  $SE(3)$ , but we can easily define a retraction that behaves similarly to the one for  $SE(3)$ , while treating velocities the same way as positions:

$$\mathcal{R}_{X_0}(\zeta) = \{ \Phi_{R_0}(\omega t), P_0 + R_0 vt, V_0 + R_0 at \}$$

Here  $\zeta = [\omega t, vt, at]$  is a 9-vector, with respectively angular, position, and velocity components. The tangent vector at  $X_0$  is

$$\left. \frac{d\mathcal{R}_{X_0}(\zeta)}{dt} \right|_{t=0} = [R_0[\omega]_{\times}, R_0 v, R_0 a]$$

and the isomorphism between  $\mathbb{R}^9$  and  $T_{X_0}M$  is  $\zeta \rightarrow [R_0[\omega t]_{\times}, R_0 vt, R_0 at]$ .

## Integration in Local Coordinates

We now proceed exactly as before to describe the evolution of the NavState in local coordinates. Let us model the solution of the differential equation (1) as a trajectory  $\zeta(t) = [\theta(t), p(t), v(t)]$ , with  $\zeta(0) = 0$ , in the local coordinate frame anchored at  $X_0$ . Note that this trajectory evolves away from  $X_0$ , and we use the symbols  $\theta$ ,  $p$ , and  $v$  to indicate that these are integrated rather than differential quantities. With that, we have

$$X(t) = \mathcal{R}_{X_0}(\zeta(t)) = \{\Phi_{R_0}(\theta(t)), P_0 + R_0 p(t), V_0 + R_0 v(t)\} \quad (7)$$

We can create a trajectory  $\gamma(\delta)$  that passes through  $X(t)$  for  $\delta = 0$

$$\gamma(\delta) = X(t + \delta) = \left\{ \Phi_{R_0}(\theta(t) + \dot{\theta}(t)\delta), P_0 + R_0 \{p(t) + \dot{p}(t)\delta\}, V_0 + R_0 \{v(t) + \dot{v}(t)\delta\} \right\}$$

and taking the derivative for  $\delta = 0$  we obtain

$$\dot{X}(t) = \left. \frac{d\gamma(\delta)}{d\delta} \right|_{\delta=0} = \left[ R(t)[H(\theta)\dot{\theta}(t)]_{\times}, R_0 \dot{p}(t), R_0 \dot{v}(t) \right]$$

Comparing that with the vector field (3), we have exact integration iff

$$\left[ R(t)[H(\theta)\dot{\theta}(t)]_{\times}, R_0 \dot{p}(t), R_0 \dot{v}(t) \right] = \left[ R(t)[\omega^b(t)]_{\times}, V(t), g + R(t)a^b(t) \right]$$

Or, as another way to state this, if we solve the differential equations for  $\theta(t)$ ,  $p(t)$ , and  $v(t)$  such that

$$\begin{aligned} \dot{\theta}(t) &= H(\theta)^{-1} \omega^b(t) \\ \dot{p}(t) &= R_0^T V_0 + v(t) \\ \dot{v}(t) &= R_0^T g + R_b^0(t) a^b(t) \end{aligned}$$

where  $R_b^0(t) = R_0^T R(t)$  is the rotation of the body frame with respect to  $R_0$ , and we have used  $V(t) = V_0 + R_0 v(t)$ .

## Application: The New IMU Factor

In the IMU factor, we need to predict the NavState  $X_j$  from the current NavState  $X_i$  and the IMU measurements in-between. The above scheme suffers from a problem, which is that  $X_i$  needs to be known in order to compensate properly for the initial velocity and rotated gravity vector. Hence, the idea of Lupton was to split up  $v(t)$  into a gravity-induced part and an accelerometer part

$$v(t) = v_g(t) + v_a(t)$$

evolving as

$$\begin{aligned} \dot{v}_g(t) &= R_i^T g \\ \dot{v}_a(t) &= R_b^i(t) a^b(t) \end{aligned}$$

The solution for the first equation is simply  $v_g(t) = R_i^T g t$ . Similarly, we split the position  $p(t)$  up in three parts

$$p(t) = p_i(t) + p_g(t) + p_v(t)$$

evolving as

$$\begin{aligned}\dot{p}_i(t) &= R_i^T V_i \\ \dot{p}_g(t) &= v_g(t) = R_i^T g t \\ \dot{p}_v(t) &= v_a(t)\end{aligned}$$

Here the solutions for the two first equations are simply

$$\begin{aligned}p_i(t) &= R_i^T V_i t \\ p_g(t) &= R_i^T \frac{g t^2}{2}\end{aligned}$$

The recipe for the IMU factor is then, in summary. Solve the ordinary differential equations

$$\begin{aligned}\dot{\theta}(t) &= H(\theta(t))^{-1} \omega^b(t) \\ \dot{p}_v(t) &= v_a(t) \\ \dot{v}_a(t) &= R_b^i(t) a^b(t)\end{aligned}$$

starting from zero, up to time  $t_{ij}$ , where  $R_b^i(t) = \exp[\theta(t)]_\times$  at all times. Form the local coordinate vector as

$$\zeta(t_{ij}) = [\theta(t_{ij}), p(t_{ij}), v(t_{ij})] = \left[ \theta(t_{ij}), R_i^T V_i t_{ij} + R_i^T \frac{g t_{ij}^2}{2} + p_v(t_{ij}), R_i^T g t_{ij} + v_a(t_{ij}) \right]$$

Predict the NavState  $X_j$  at time  $t_j$  from

$$X_j = \mathcal{R}_{X_i}(\zeta(t_{ij})) = \left\{ \Phi_{R_0}(\theta(t_{ij})), P_i + V_i t_{ij} + \frac{g t_{ij}^2}{2} + R_i p_v(t_{ij}), V_i + g t_{ij} + R_i v_a(t_{ij}) \right\}$$

Note that the predicted NavState  $X_j$  depends on  $X_i$ , but the integrated quantities  $\theta(t), p_v(t)$ , and  $v_a(t)$  do not.

## A Simple Euler Scheme

To solve the differential equation we can use a simple Euler scheme:

$$\theta_{k+1} = \theta_k + \dot{\theta}(t_k) \Delta t = \theta_k + H(\theta_k)^{-1} \omega_k^b \Delta t \quad (8)$$

$$p_{k+1} = p_k + \dot{p}_v(t_k) \Delta t = p_k + v_k \Delta t \quad (9)$$

$$v_{k+1} = v_k + \dot{v}_a(t_k) \Delta t = v_k + \exp([\theta_k]_\times) a_k^b \Delta t \quad (10)$$

where  $\theta_k \triangleq \theta(t_k)$ ,  $p_k \triangleq p_v(t_k)$ , and  $v_k \triangleq v_a(t_k)$ . However, the position propagation can be done more accurately, by using exact integration of the zero-order hold acceleration  $a_k^b$ :

$$\theta_{k+1} = \theta_k + H(\theta_k)^{-1} \omega_k^b \Delta t \quad (11)$$

$$p_{k+1} = p_k + v_k \Delta t + R_k a_k^b \frac{\Delta t^2}{2} \quad (12)$$

$$v_{k+1} = v_k + R_k a_k^b \Delta t \quad (13)$$

where we defined the rotation matrix  $R_k = \exp([\theta_k]_\times)$ .

## Noise Propagation

Even when we assume uncorrelated noise on  $\omega^b$  and  $a^b$ , the noise on the final computed quantities will have a non-trivial covariance structure, because the intermediate quantities  $\theta_k$  and  $v_k$  appear in multiple places. To model the noise propagation, let us define  $\zeta_k = [\theta_k, p_k, v_k]$  and rewrite Eqns. (11-13) as the non-linear function  $f$

$$\zeta_{k+1} = f(\zeta_k, a_k^b, \omega_k^b)$$

Then the noise on  $\zeta_{k+1}$  propagates as

$$\Sigma_{k+1} = A_k \Sigma_k A_k^T + B_k \Sigma_\eta^{ad} B_k + C_k \Sigma_\eta^{gd} C_k \quad (14)$$

where  $A_k$  is the  $9 \times 9$  partial derivative of  $f$  wrpt  $\zeta$ , and  $B_k$  and  $C_k$  the respective  $9 \times 3$  partial derivatives with respect to the measured quantities  $a^b$  and  $\omega^b$ .

We start with the noise propagation on  $\theta$ , which is independent of the other quantities. Taking the derivative, we have

$$\frac{\partial \theta_{k+1}}{\partial \theta_k} = I_{3 \times 3} + \frac{\partial H(\theta_k)^{-1} \omega_k^b}{\partial \theta_k} \Delta t$$

It can be shown that for small  $\theta_k$  we have

$$\frac{\partial H(\theta_k)^{-1} \omega_k^b}{\partial \theta_k} \approx -\frac{1}{2} [\omega_k^b]_\times \text{ and hence } \frac{\partial \theta_{k+1}}{\partial \theta_k} = I_{3 \times 3} - \frac{\Delta t}{2} [\omega_k^b]_\times$$

For the derivatives of  $p_{k+1}$  and  $v_{k+1}$  we need the derivative

$$\frac{\partial R_k a_k^b}{\partial \theta_k} = R_k [-a_k^b]_\times \frac{\partial R_k}{\partial \theta_k} = R_k [-a_k^b]_\times H(\theta_k)$$

where we used

$$\frac{\partial (Ra)}{\partial R} \approx R[-a]_\times$$

and the fact that the dependence of the rotation  $R_k$  on  $\theta_k$  is the already computed  $H(\theta_k)$ .

Putting all this together, we finally obtain

$$A_k \approx \begin{bmatrix} I_{3 \times 3} - \frac{\Delta t}{2} [\omega_k^b]_\times & & \\ R_k [-a_k^b]_\times H(\theta_k) \frac{\Delta t}{2} & I_{3 \times 3} & I_{3 \times 3} \Delta t \\ R_k [-a_k^b]_\times H(\theta_k) \Delta t & & I_{3 \times 3} \end{bmatrix}$$

The other partial derivatives are simply

$$B_k = \begin{bmatrix} 0_{3 \times 3} \\ R_k \frac{\Delta t}{2} \\ R_k \Delta t \end{bmatrix}, \quad C_k = \begin{bmatrix} H(\theta_k)^{-1} \Delta t \\ 0_{3 \times 3} \\ 0_{3 \times 3} \end{bmatrix}$$

## Combined IMU Factor

We can similarly account for bias drift over time, as is commonly seen in commercial grade IMUs. This is accomplished via the *CombinedImuFactor* which is a 6-way factor between the previous *pose/velocity/bias* and the *pose/velocity/bias* at the next timestep.

We expand the state vector as  $\zeta_k = [\theta_k, p_k, v_k, b_k^a, b_k^\omega]$  to include the bias terms. This gives the noise propagation equation as

$$\Sigma_{k+1} = F_k \Sigma_k F_k^T + G_k \Sigma_k G_k \quad (15)$$

where  $F_k$  is the  $15 \times 15$  derivative of  $f$  wrpt this new  $\zeta$ , and  $G_k$  is the  $15 \times 15$  matrix for first order uncertainty propagation. The top-left  $9 \times 9$  of  $F_k$  is the same as  $A_k$ , thus we only have the jacobians wrpt the biases left to account for.

Conveniently, the jacobians of the pose and velocity wrpt the biases are already computed in the *ImuFactor* derivation as matrices  $B_k$  and  $C_k$ , while they are identity matrices wrpt the biases themselves. Thus, we can easily plug-in the values from the previous section to give us the final result

$$F_k \approx \begin{bmatrix} I_{3 \times 3} - \frac{\Delta_t}{2} [\omega_k^b]_{\times} & & & & H(\theta_k)^{-1} \Delta_t \\ R_k [-a_k^b]_{\times} H(\theta_k) \frac{\Delta_t^2}{2} & I_{3 \times 3} & I_{3 \times 3} \Delta_t & R_k \frac{\Delta_t^2}{2} & \\ R_k [-a_k^b]_{\times} H(\theta_k) \Delta_t & & I_{3 \times 3} & R_k \Delta_t & \\ & & & I_{3 \times 3} & \\ & & & & I_{3 \times 3} \end{bmatrix}$$

## References

- [1] Arieh Iserles, Hans Z Munthe-Kaas, Syvert P Nørsett, and Antonella Zanna. Lie-group methods. *Acta Numerica 2000*, 9:215–365, 2000.