# Attitude Control by Lie Algebra and Anisotropic Sensor Fusion

Nobuaki Tanaka

November 16, 2025

## 1 Rotation Optimization on $SO(3)$ via Lie Algebra and Anisotropic Sensor Fusion

### 1.1 Lie group, Lie algebra, and tangent space

We model the rover attitude by a rotation matrix

$$R \in SO(3) = \{\, R \in \mathbb{R}^{3\times 3} \,\big|\, R^\top R = I,\ \det R = 1 \}.$$

The associated Lie algebra is

$$\mathfrak{so}(3) = \{\, \Omega \in \mathbb{R}^{3\times 3} \,\big|\, \Omega^\top = -\Omega \},$$

which is a three-dimensional vector space. Using the standard "hat" operator

$$\widehat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3), \qquad \boldsymbol{\omega} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \mapsto \widehat{\boldsymbol{\omega}} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix},$$

any element of $\mathfrak{so}(3)$ can be written as $\widehat{\boldsymbol{\omega}}$ for some $\boldsymbol{\omega} \in \mathbb{R}^3$.

A smooth curve on $SO(3)$ through a point $R$ can be written as

$$R(t) = R\exp\big(t\,\widehat{\boldsymbol{\omega}}\big), \qquad t \in \mathbb{R},$$

where $\exp : \mathfrak{so}(3) \to SO(3)$ is the matrix exponential. By differentiating at $t = 0$ we obtain the tangent vector

$$\dot{R}(0) = \frac{d}{dt} R\exp\big(t\,\widehat{\boldsymbol{\omega}}\big)\bigg|_{t=0} = R\,\widehat{\boldsymbol{\omega}}.$$

Thus the tangent space at $R$ is

$$T_R SO(3) = \{\, R\Omega \,\big|\, \Omega \in \mathfrak{so}(3) \},$$

which is isomorphic to $\mathfrak{so}(3)$ itself. In other words, infinitesimal rotations $\widehat{\boldsymbol{\omega}} \in \mathfrak{so}(3)$ can be viewed as directions in the tangent space $T_R SO(3)$ at any $R \in SO(3)$.

In our rover, the wheel encoders generate high-frequency pulse signals. As a consequence, the rotational increment between successive control cycles is very small. Therefore it is natural to treat each update as an infinitesimal rotation on the Lie algebra $\mathfrak{so}(3)$ rather than as an arbitrary finite rotation. Let $R_k \in SO(3)$ be the attitude at time step $k$, and let $\boldsymbol{\omega}_k \in \mathbb{R}^3$ be the angular velocity estimated from encoder pulses over the sampling interval $\Delta t$. Then the predicted attitude is

$$R_{k+1}^{\text{pred}} = R_k \exp\big(\widehat{\boldsymbol{\omega}_k \Delta t}\big), \tag{1}$$

which is precisely an integration on the tangent space $T_{R_k}SO(3)$ via the exponential map.

## 1.2 Sensor models and residuals

To correct the encoder prediction and obtain a statistically consistent estimate of the pose, we fuse three heterogeneous sensors:

- ARCore pose $(R^{\text{ar}}, \boldsymbol{t}^{\text{ar}})$: low-rate but globally consistent 6-DoF pose.
- Gravity vector $\boldsymbol{g}^{\text{meas}}$ in the body frame: constrains roll and pitch.
- Magnetic field $\boldsymbol{m}^{\text{meas}}$ in the body frame: constrains yaw.

We assume known reference directions $\boldsymbol{g}_0, \boldsymbol{m}_0$ in the world frame. Given a candidate pose $(R, \boldsymbol{t}) \in SE(3)$, these references appear in the body frame as

$$\boldsymbol{g}^{\text{pred}} = R^\top \boldsymbol{g}_0, \qquad \boldsymbol{m}^{\text{pred}} = R^\top \boldsymbol{m}_0.$$

### 1.2.1 ARCore pose residual

The ARCore pose provides a full $SE(3)$ measurement. The rotational part is compared on $SO(3)$ via the logarithm map

$$\text{Log} : SO(3) \to \mathfrak{so}(3) \cong \mathbb{R}^3,$$

which yields the minimal three-dimensional representation of the relative rotation. The residual is

$$r_{\text{ar}}(R, \boldsymbol{t}) = \begin{pmatrix} \text{Log}\big((R^{\text{ar}})^\top R\big) \\ \boldsymbol{t} - \boldsymbol{t}^{\text{ar}} \end{pmatrix} \in \mathbb{R}^6. \tag{2}$$

### 1.2.2 Unit-vector residuals for gravity and magnetometer

For a world-frame reference unit vector $\boldsymbol{v}_0$ (gravity or magnetic north), the predicted body-frame direction is $\boldsymbol{v}^{\text{pred}} = R^\top \boldsymbol{v}_0$ and the measured body-frame direction is $\boldsymbol{v}^{\text{meas}}$. We normalize both vectors and define

$$\boldsymbol{a}^{\text{pred}} = \frac{\boldsymbol{v}^{\text{pred}}}{\|\boldsymbol{v}^{\text{pred}}\|}, \qquad \boldsymbol{a}^{\text{meas}} = \frac{\boldsymbol{v}^{\text{meas}}}{\|\boldsymbol{v}^{\text{meas}}\|}.$$

A small rotation that aligns $\boldsymbol{a}^{\text{pred}}$ with $\boldsymbol{a}^{\text{meas}}$ is well approximated by the cross product

$$\boldsymbol{r} = \boldsymbol{a}^{\text{pred}} \times \boldsymbol{a}^{\text{meas}} \in \mathbb{R}^3.$$

Geometrically, $\boldsymbol{r}$ is an infinitesimal rotation vector in $\mathfrak{so}(3) \cong \mathbb{R}^3$ that rotates the predicted direction toward the measured one. We use the same form for both gravity and magnetometer:

$$r_g(R) = \boldsymbol{a}_g^{\text{pred}} \times \boldsymbol{a}_g^{\text{meas}} \in \mathbb{R}^3, \tag{3}$$

$$r_m(R) = \boldsymbol{a}_m^{\text{pred}} \times \boldsymbol{a}_m^{\text{meas}} \in \mathbb{R}^3. \tag{4}$$

The Jacobian of these residuals with respect to a small rotation $\delta\boldsymbol{\theta} \in \mathbb{R}^3$ is a $3 \times 3$ matrix

$$J_w = \frac{\partial \boldsymbol{r}}{\partial(\delta\boldsymbol{\theta})} \in \mathbb{R}^{3\times 3}, \tag{5}$$

which is evaluated analytically in the implementation and used to construct the normal equations.

## 1.3 Anisotropic Mahalanobis distance and maximum likelihood

Each sensor has anisotropic noise characteristics: some directions in $\mathbb{R}^3$ (or $\mathbb{R}^6$) are more reliable than others. We represent this by a positive-definite covariance matrix for each residual:

$$\Sigma_{\text{ar}} \in \mathbb{R}^{6\times 6}, \quad \Sigma_g \in \mathbb{R}^{3\times 3}, \quad \Sigma_m \in \mathbb{R}^{3\times 3}.$$

Their inverses

$$S_{\text{ar}} = \Sigma_{\text{ar}}^{-1}, \quad S_g = \Sigma_g^{-1}, \quad S_m = \Sigma_m^{-1}$$

encode the anisotropic weights. For example, a gravity covariance of the form

$$\Sigma_g = R_g \begin{pmatrix} \sigma_\perp^2 & 0 & 0 \\ 0 & \sigma_\parallel^2 & 0 \\ 0 & 0 & \sigma_\parallel^2 \end{pmatrix} R_g^\top$$

means that the component orthogonal to gravity (tilt) is more strictly penalized than components parallel to gravity (yaw), if $\sigma_\perp^2 \ll \sigma_\parallel^2$.

Assuming zero-mean Gaussian noise, the negative log-likelihood of a state $(R, \boldsymbol{t})$ given the measurements is, up to an additive constant, the sum of Mahalanobis distances:

$$J(R, \boldsymbol{t}) = \underbrace{r_{\text{ar}}(R, \boldsymbol{t})^\top S_{\text{ar}}\, r_{\text{ar}}(R, \boldsymbol{t})}_{\text{ARCore pose}} + \underbrace{r_g(R)^\top S_g\, r_g(R)}_{\text{gravity}} + \underbrace{r_m(R)^\top S_m\, r_m(R)}_{\text{magnetometer}}. \tag{6}$$

This functional explicitly incorporates the anisotropy of each sensor.

The maximum-likelihood (and, under a Gaussian prior, MAP) estimate is then

$$(R^*, \boldsymbol{t}^*) = \arg \min_{R \in SO(3),\, \boldsymbol{t} \in \mathbb{R}^3} J(R, \boldsymbol{t}). \tag{7}$$

Thus the "decision rule" for the rover orientation is *minimization of the sum of Mahalanobis distances* over $SE(3)$.

## 1.4 Gauss–Newton on $SE(3)$

We now describe the concrete optimization procedure on the Lie group $SE(3)$. Let the current estimate be $(R, \boldsymbol{t})$, and parametrize a small increment in the Lie algebra by

$$\delta\xi = \begin{pmatrix} \delta\boldsymbol{\theta} \\ \delta\boldsymbol{\rho} \end{pmatrix} \in \mathbb{R}^6,$$

where $\delta\boldsymbol{\theta} \in \mathbb{R}^3$ is a small rotation vector and $\delta\boldsymbol{\rho} \in \mathbb{R}^3$ is a small translation. Using the right-invariant update

$$R \leftarrow R \exp\big(\widehat{\delta\boldsymbol{\theta}}\big), \qquad \boldsymbol{t} \leftarrow \boldsymbol{t} + \delta\boldsymbol{\rho}, \tag{8}$$

we ensure that $R$ stays on $SO(3)$ at every iteration.

Let $r_i(R, \boldsymbol{t})$ denote any of the residuals $r_{\mathrm{ar}}, r_g, r_m$. For a small perturbation $\delta\xi$, we linearize

$$r_i(R \oplus \delta\xi) \approx r_i(R) + J_i \delta\xi,$$

where $J_i = \partial r_i / \partial\delta\xi$ is the Jacobian. Substituting this into the cost (6) and ignoring higher-order terms, we obtain a quadratic approximation

$$J(R \oplus \delta\xi, \boldsymbol{t} \oplus \delta\xi) \approx J(R, \boldsymbol{t}) + \boldsymbol{b}^\top \delta\xi + \tfrac{1}{2} \delta\xi^\top H \, \delta\xi,$$

with the normal-equation matrix $H$ and vector $\boldsymbol{b}$ given by

$$H = \sum_i J_i^\top S_i J_i, \qquad \boldsymbol{b} = \sum_i J_i^\top S_i \, r_i(R, \boldsymbol{t}),$$

where $S_i$ is the corresponding inverse covariance.

The Gauss–Newton step $\delta\xi$ is obtained by solving

$$H \, \delta\xi = -\boldsymbol{b}. \tag{9}$$

We then apply the Lie-algebra update (8) and repeat until the increment is sufficiently small:

$$\|\delta\boldsymbol{\theta}\| < \varepsilon_{\mathrm{rot}}, \qquad \|\delta\boldsymbol{\rho}\| < \varepsilon_{\mathrm{trans}}.$$

In the actual implementation, $H$ is a $6 \times 6$ symmetric positive-definite matrix. We factor it by Cholesky decomposition $H = LL^\top$ and solve (9) via forward and backward substitution to obtain $\delta\xi$. The inverse $H^{-1}$ can be computed from $L$ if an approximate pose covariance is required.

## 1.5 Levenberg–Marquardt damping

Pure Gauss–Newton may become unstable when the linearization is poor or when the problem is ill-conditioned. Levenberg–Marquardt (LM) improves robustness by adding a damping term to the normal equations:

$$(H + \lambda I) \, \delta\xi = -\boldsymbol{b}, \qquad \lambda > 0. \tag{10}$$

For large $\lambda$, the step is small and close to gradient descent; for $\lambda \to 0$, the method reduces to Gauss–Newton. A typical LM algorithm proceeds as follows:

1. Initialize $\lambda$ (e.g. based on the diagonal of $H$).

2. Solve (10) for $\delta\xi$ and compute a tentative update.
3. Evaluate the actual reduction in $J$ and compare it with the predicted reduction from the quadratic model.
4. If the step is successful (sufficient reduction), accept the update and decrease $\lambda$. Otherwise reject the update and increase $\lambda$.

This adaptive damping realizes a compromise between ($i$) fast convergence near the optimum and ($ii$) robustness when far from the optimum.

In our rover, the LM idea is used in a lightweight form: a small positive value is added to the diagonal of $H$ when the Cholesky decomposition fails, which corresponds to (10) with a fixed $\lambda$. This stabilizes the optimization while keeping the computation cost suitable for real-time mobile execution.

### 1.6 Summary

- The encoder pulses are integrated on the Lie algebra $\mathfrak{so}(3)$ using the exponential map (1), which is justified by the small rotational increments and the interpretation of $\mathfrak{so}(3)$ as the tangent space $T_R SO(3)$ at each attitude $R$.
- ARCore, gravity, and magnetometer measurements are converted into residuals $r_{\mathrm{ar}}, r_g, r_m$ defined on $SE(3)$ and $\mathfrak{so}(3)$.
- Their anisotropic uncertainties are encoded by covariance matrices $\Sigma_{\mathrm{ar}}, \Sigma_g, \Sigma_m$, and the total cost is the sum of Mahalanobis distances (6).
- The maximum-likelihood estimate (7) is obtained by solving the nonlinear least-squares problem on $SE(3)$ via Gauss–Newton / Levenberg–Marquardt iterations on the Lie algebra, using the right-invariant update (8).

This completes the description of how infinitesimal rotations on the Lie algebra, anisotropic Mahalanobis distances, and a Lie-algebra based maximum-likelihood estimator (with LM damping) are combined for orientation control of the rover.