Basic

SLAM Tutorials for Everyone
Lec. #9. Lie Group and Lie Algebra

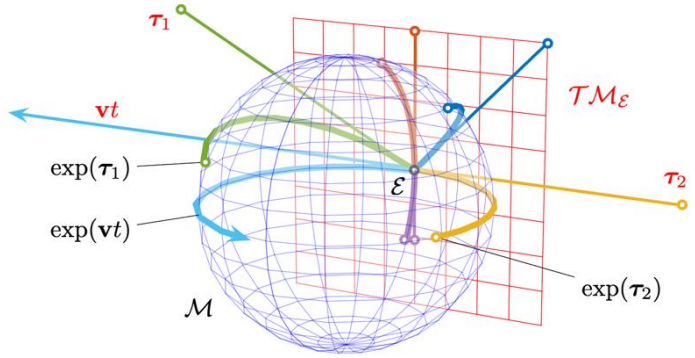
Hyungtae Lim, Ph.D.

<shapelim [at] mit [dot] edu> / <fudxo5143+slam [at] gmail [dot] com>

Today's Topic: Relationship Between Lie Group and Lie Algebra

Q1. How to express the Pose in 3D?

Q2. What is Lie Group & Lie Algebra?



Q3. How to optimize?

Q4. How to perform data association?

Problem: Addition Is Not a Valid Update on $SO(3)$

1D position (easy):

$$x_{k+1} = x_k + \Delta x \quad (\checkmark)$$

3D position (still easy):

$$\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \boldsymbol{v}\,\delta t \quad (\checkmark)$$

$\Rightarrow$ Simple vector addition. Works perfectly.

3D rotation (broken!):

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k + \Delta \boldsymbol{R} \quad (\times)$$

Even if $\boldsymbol{R}_k \in SO(3)$ and $\Delta \boldsymbol{R} \in SO(3)$:

$$(\boldsymbol{R}_k + \Delta \boldsymbol{R})^\top (\boldsymbol{R}_k + \Delta \boldsymbol{R}) \neq \boldsymbol{I}$$

$\Rightarrow$ The sum **leaves** $SO(3)$.

Goal: Find a way to do $\boldsymbol{p}_{k+1} = \boldsymbol{p}_k + \boldsymbol{v}\,\delta t$-style updates for rotation.
Answer: Lie algebra gives us exactly this — but on a curved space.

Solution: Introduce the Lie Algebra Representation

Instead of adding matrices directly, **work in the algebra** (a flat $\mathbb{R}^3$ space) and **map back** to $SO(3)$. This unlocks two operations that were previously impossible:

Example 1 — Incremental update

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k \cdot \exp([\boldsymbol{\varphi}_k]_\times)$$

Add a small rotation $\boldsymbol{\varphi}_k \in \mathbb{R}^3$ step by step.
$\rightarrow$ mirrors $x_{k+1} = x_k + \Delta x$

Example 2 — Scalar scaling

$$\boldsymbol{R}(t) = \exp(t\,[\boldsymbol{\varphi}]_\times), \quad t \in [0, 1]$$

Scale a rotation continuously by t.
$\rightarrow$ mirrors $s(t) = t\,\Delta x$

Key tool: the exponential map exp : $[\boldsymbol{\varphi}]_\times \mapsto SO(3)$ and its inverse, the logarithmic map log : $SO(3) \mapsto [\boldsymbol{\varphi}]_\times$.

We will build up to these — starting with why skew-symmetric matrices appear.

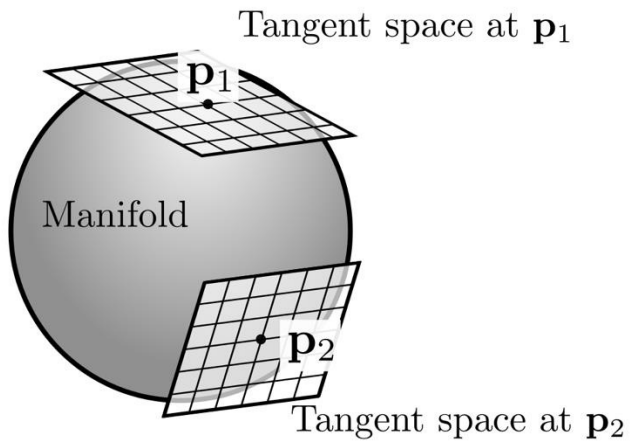
Focus on "Algebra", not "Lie"

- "Lie Group" and "Lie Algebra" sound intimidating. Let's break them down.

"Lie" $\rightarrow$ named after Sophus Lie. Ignore this for now.

"Algebra" $\rightarrow$ a set equipped with operations.

- You already know $\mathbb{R}^3$: a vector space with $+$ and $\times$.
- $\mathfrak{so}(3)$ (the Lie algebra of $SO(3)$) is also a vector space $\cong \mathbb{R}^3$.
- So increments $\Delta\boldsymbol{\phi} \in \mathfrak{so}(3) \cong \mathbb{R}^3$ can be **added normally**.



Strategy: Compute the increment $\Delta\boldsymbol{\phi}$ in the flat algebra $\mathbb{R}^3$, then map it back to the curved manifold $SO(3)$.

To make this concrete, we need two tools:

- **Skew-symmetric matrix** $[\cdot]_\times$ — encodes a rotation increment as a flat vector.
- **Exponential map** $\exp(\cdot)$ — lifts that flat vector back onto $SO(3)$.

Intuitive Understanding of Incremental Update

Start in 2D. Differentiate $\boldsymbol{R}(\theta)$:

$$\frac{\partial \boldsymbol{R}(\theta)}{\partial \theta} = \boldsymbol{R}(\theta)\,\hat{\Omega}, \quad \hat{\Omega} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$\hat{\Omega}$ is **skew-symmetric** — it is the 2D Lie algebra generator.

Small angle $\delta\theta \approx 0$ ($\cos\delta\theta \approx 1,\ \sin\delta\theta \approx \delta\theta$):

$$\boldsymbol{R}(\delta\theta) \approx \boldsymbol{I} + \hat{\Omega}\,\delta\theta$$

So a rotation update becomes additive:

$$\begin{aligned} \boldsymbol{R}(\theta)\,\boldsymbol{R}(\delta\theta) &\approx \boldsymbol{R}(\theta) + \frac{\partial \boldsymbol{R}}{\partial \theta}\,\delta\theta \\ &= \boldsymbol{R}(\theta) + \boldsymbol{R}(\theta)\hat{\Omega}\,\delta\theta \end{aligned}$$

Intuitive Understanding of Incremental Update (Cont'd)

Start in 2D. Differentiate $\boldsymbol{R}(\theta)$:

$$\frac{\partial \boldsymbol{R}(\theta)}{\partial \theta} = \boldsymbol{R}(\theta)\,\hat{\Omega}, \quad \hat{\Omega} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$\hat{\Omega}$ is **skew-symmetric** — it is the 2D Lie algebra generator.

Small angle $\delta\theta \approx 0$ ($\cos\delta\theta \approx 1,\ \sin\delta\theta \approx \delta\theta$):

$$\boldsymbol{R}(\delta\theta) \approx \boldsymbol{I} + \hat{\Omega}\,\delta\theta$$

So a rotation update becomes additive:

$$\begin{aligned} \boldsymbol{R}(\theta)\,\boldsymbol{R}(\delta\theta) &\approx \boldsymbol{R}(\theta) + \frac{\partial \boldsymbol{R}}{\partial \theta}\,\delta\theta \\ &= \boldsymbol{R}(\theta) + \boldsymbol{R}(\theta)\hat{\Omega}\,\delta\theta \end{aligned}$$

3D generalization. Replace $\hat{\Omega}\,\delta\theta$ with $[\boldsymbol{\omega}]_\times$ ($\boldsymbol{\omega} \in \mathbb{R}^3$, small):

$$\begin{aligned} \boldsymbol{R}_{\text{new}} &\approx \boldsymbol{R} \cdot (\boldsymbol{I} + [\boldsymbol{\omega}]_\times) \\ &= \boldsymbol{R} + \boldsymbol{R}\,[\boldsymbol{\omega}]_\times \end{aligned}$$

where $[\boldsymbol{\omega}]_\times$ is skew-symmetric with 3 free entries:

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Pattern

- 2D: $\hat{\Omega}$ — **1** free parameter (z-axis).
- 3D: $[\cdot]_\times$ — **3** free parameters.
- Both from differentiating $\boldsymbol{R}$ at the identity.

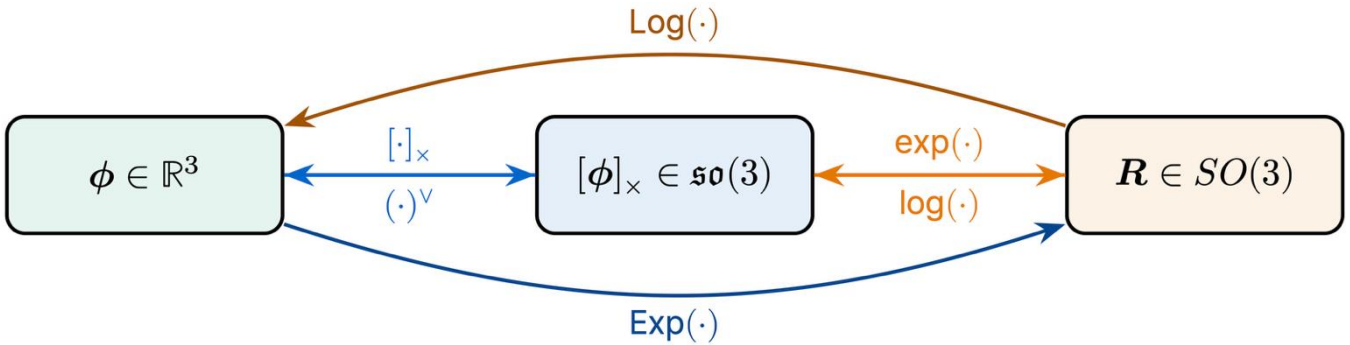
Lie Algebra $\mathfrak{so}(3)$: The Tangent Space of $SO(3)$

The Lie algebra is the **tangent space at the identity** of the Lie group.

For $SO(3)$, the Lie algebra $\mathfrak{so}(3)$ consists of 3×3 skew-symmetric matrices:

$$\boldsymbol{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} \in \mathbb{R}^3 \quad \overset{[\cdot]_\times}{\longleftrightarrow} \quad [\boldsymbol{\phi}]_\times = \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix} \in \mathfrak{so}(3)$$

- $\boldsymbol{\phi} = \theta\,\hat{\boldsymbol{a}}$, where $\hat{\boldsymbol{a}} \in \mathbb{R}^3$ is the rotation **axis** ($\|\hat{\boldsymbol{a}}\| = 1$) and $\theta = \|\boldsymbol{\phi}\|$ is the rotation **angle**.
- $\mathfrak{so}(3) \cong \mathbb{R}^3$: it is a vector space. Addition and scalar multiplication are valid.
- This is where we can safely perform optimization increments.



What Does $\exp(\cdot)$ of Matrix Mean?

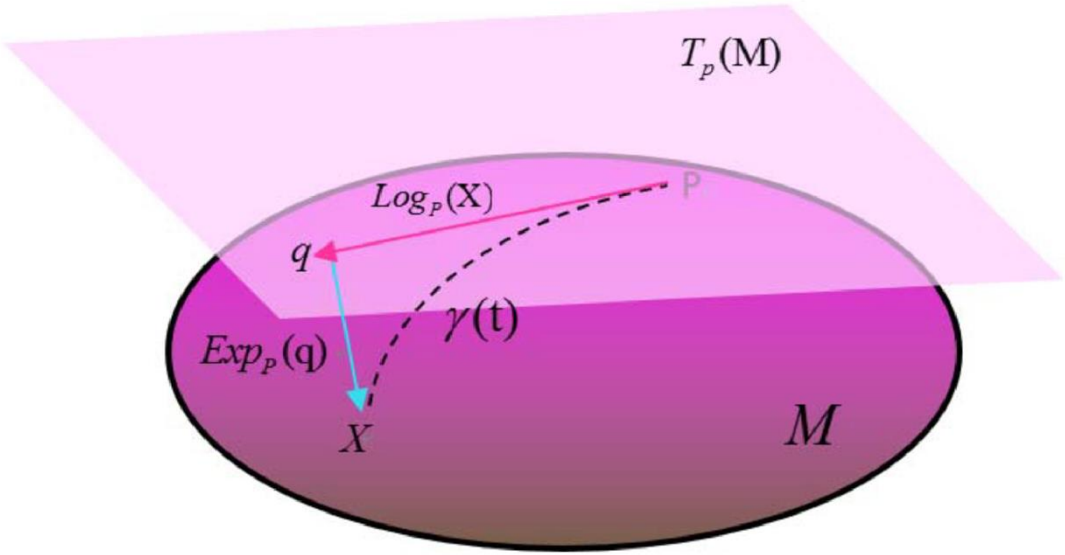
Via Taylor Series,
Scalar case (where $x \in \mathbb{R}^1$):

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

Matrix case:

$$\exp(\boldsymbol{A}) = \boldsymbol{I} + \boldsymbol{A} + \frac{\boldsymbol{A}^2}{2!} + \frac{\boldsymbol{A}^3}{3!} + \cdots$$

Properties: $\exp(\mathbf{0}) = \boldsymbol{I}$;
$\exp(\boldsymbol{A})\exp(-\boldsymbol{A}) = \boldsymbol{I}$.



Liu et al., "Visualization of the Geometric Transformation Group Based on the Riemannian Metric," ICSAI, 2016

Why does it give a rotation? For skew-symmetric $[\boldsymbol{\phi}]_\times$, the series simplifies (via $[\boldsymbol{\phi}]_\times^3 = -\theta^2[\boldsymbol{\phi}]_\times$) into the **Rodrigues formula** — always a valid $SO(3)$ matrix.

Exponential and Logarithmic Maps

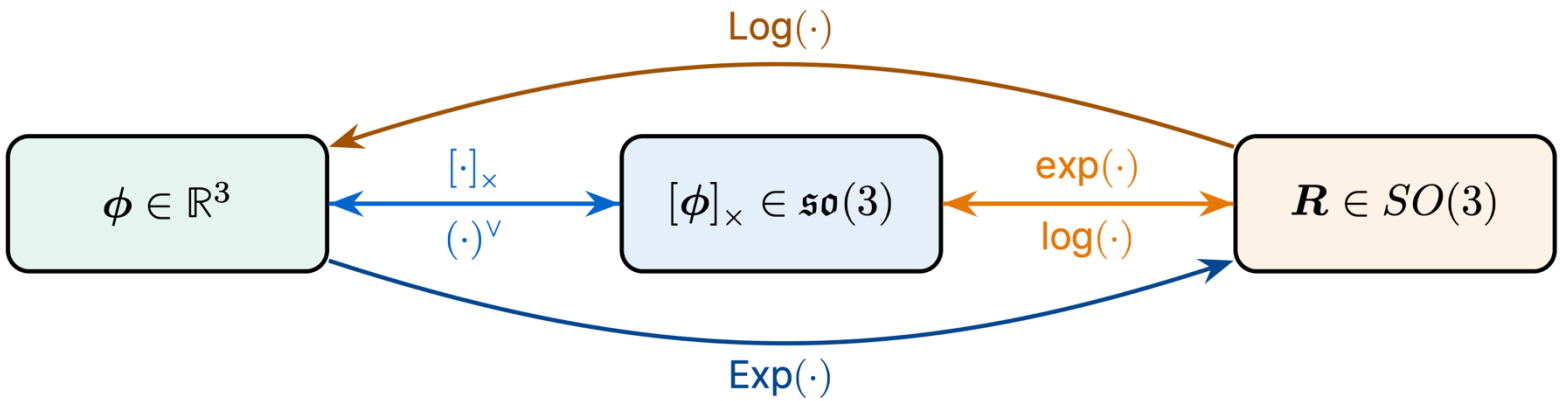
Exponential map $\exp : \mathfrak{so}(3) \to SO(3)$: maps a tangent vector to a rotation.

$$\boldsymbol{R} = \exp([\boldsymbol{\phi}]_\times) = \boldsymbol{I} + \frac{\sin\theta}{\theta}[\boldsymbol{\phi}]_\times + \frac{1-\cos\theta}{\theta^2}[\boldsymbol{\phi}]_\times^2, \qquad \theta = \|\boldsymbol{\phi}\|$$

This is the **Rodrigues' formula**. It converts axis-angle $\boldsymbol{\phi}$ to a rotation matrix.

Logarithmic map $\ln : SO(3) \to \mathfrak{so}(3)$: the inverse operation.

$$\theta = \arccos\left(\frac{\mathrm{tr}(\boldsymbol{R})-1}{2}\right), \qquad [\boldsymbol{\phi}]_\times = \frac{\theta}{2\sin\theta}(\boldsymbol{R}-\boldsymbol{R}^\top)$$



Small-Angle Approximation: $SO(3)$

As $\theta = \|\boldsymbol{\phi}\| \to 0$, the Rodrigues formula collapses:

$$\frac{\sin\theta}{\theta} \to 1, \qquad \frac{1-\cos\theta}{\theta^2} \to 0$$

so the **exact** exp becomes a **first-order** approximation:

$$\boxed{\exp([\boldsymbol{\phi}]_\times) \approx \boldsymbol{I} + [\boldsymbol{\phi}]_\times}$$

Two immediate uses

IMU integration (δt tiny $\Rightarrow \boldsymbol{\omega}\delta t$ small):

$$\begin{aligned}\boldsymbol{R}_{k+1} &= \boldsymbol{R}_k \exp([\boldsymbol{\omega}_k \delta t]_\times) \\ &\approx \boldsymbol{R}_k(\boldsymbol{I} + [\boldsymbol{\omega}_k \delta t]_\times)\end{aligned}$$

The rotation update becomes a single matrix multiply.

Optimizer step ($\Delta\boldsymbol{\phi}$ small near convergence):

$$\boldsymbol{R} \leftarrow \boldsymbol{R} \cdot (\boldsymbol{I} + [\Delta\boldsymbol{\phi}]_\times)$$

keeps $\boldsymbol{R}$ valid in $SO(3)$ to first order.

Lie Algebra $\mathfrak{se}(3)$: The Tangent Space of $SE(3)$

For full rigid body motion, the Lie algebra is $\mathfrak{se}(3) \cong \mathbb{R}^6$. The 6-vector $\boldsymbol{\xi}$ is called a **twist** (GTSAM convention: angle first, position second):

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\phi} \\ \boldsymbol{\rho} \end{bmatrix} \in \mathbb{R}^6 \quad \overset{[\cdot]_\wedge}{\longleftrightarrow} \quad [\boldsymbol{\xi}]_\wedge = \begin{bmatrix} [\boldsymbol{\phi}]_\times & \boldsymbol{\rho} \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathfrak{se}(3) \subset \mathbb{R}^{4\times 4}$$

where $\boldsymbol{\phi} \in \mathbb{R}^3$ is the rotational component (angle) and $\boldsymbol{\rho} \in \mathbb{R}^3$ is the translational component (position).

Exponential map for $SE(3)$:

$$\boldsymbol{T} = \exp([\boldsymbol{\xi}]_\wedge) = \begin{bmatrix} \exp([\boldsymbol{\phi}]_\times) & \boldsymbol{J}\,\boldsymbol{\rho} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

where $\boldsymbol{J}$ is the **left Jacobian** of $SO(3)$:

$$\boldsymbol{J} = \boldsymbol{I} + \frac{1-\cos\theta}{\theta^2}[\boldsymbol{\phi}]_\times + \frac{\theta - \sin\theta}{\theta^3}[\boldsymbol{\phi}]_\times^2$$

Note: translation is coupled with rotation through $\boldsymbol{J}$, not simply $\boldsymbol{\rho}$.

Small-Angle Approximation: $SE(3)$

Recall: the exact exponential map gives $\boldsymbol{T} = \exp([\boldsymbol{\xi}]_\wedge)$. For a **small** perturbation $\Delta\boldsymbol{\xi} = [\Delta\boldsymbol{\phi}, \Delta\boldsymbol{\rho}]^\top$, $\boldsymbol{J} \to \boldsymbol{I}$ and:

$$\exp([\Delta\boldsymbol{\xi}]_\wedge) \approx \boldsymbol{I}_4 + [\Delta\boldsymbol{\xi}]_\wedge = \begin{bmatrix} \boldsymbol{I} + [\Delta\boldsymbol{\phi}]_\times & \Delta\boldsymbol{\rho} \\ \boldsymbol{0}^\top & 1 \end{bmatrix}$$

This is a first-order perturbation around the identity $\boldsymbol{I}_4$, not a full pose.

Pose update step (current pose $\boldsymbol{T}$, correction $\Delta\boldsymbol{\xi}$):

$$\begin{aligned} \boldsymbol{T}_{\text{new}} &= \boldsymbol{T} \cdot \exp([\Delta\boldsymbol{\xi}]_\wedge) \\ &\approx \boldsymbol{T} \cdot (\boldsymbol{I}_4 + [\Delta\boldsymbol{\xi}]_\wedge) \\ &= \boldsymbol{T} + \boldsymbol{T}\,[\Delta\boldsymbol{\xi}]_\wedge \end{aligned}$$

$\boldsymbol{T}_{\text{new}}$ is a new valid $SE(3)$ pose shifted by $\Delta\boldsymbol{\xi}$ from $\boldsymbol{T}$.

When is this valid?

	Valid?
Initial pose guess	× (large $\boldsymbol{\xi}$)
Optimizer step	✓ (small $\Delta\boldsymbol{\xi}$)
IMU step ($\delta t \to 0$)	✓ (tiny)

Retraction Operator: $\boldsymbol{x} \leftarrow \boldsymbol{x} \boxplus \Delta\boldsymbol{x}$

Core idea: Perform optimization in the tangent space (flat, Euclidean), then map the update back to the manifold.

State space	Update rule	Increment
$\mathbb{R}^n$ (position, velocity)	$\boldsymbol{t} \leftarrow \boldsymbol{t} + \Delta\boldsymbol{t}$	$\Delta\boldsymbol{t} \in \mathbb{R}^3$
$SO(3)$ (rotation)	$\boldsymbol{R} \leftarrow \boldsymbol{R} \cdot \exp([\Delta\boldsymbol{\phi}]_\times)$	$\Delta\boldsymbol{\phi} \in \mathbb{R}^3$
$SE(3)$ (pose)	$\boldsymbol{T} \leftarrow \boldsymbol{T} \cdot \exp([\Delta\boldsymbol{\xi}]_\wedge)$	$\Delta\boldsymbol{\xi} \in \mathbb{R}^6$

General notation:

$$\boldsymbol{x} \leftarrow \boldsymbol{x} \boxplus \Delta\boldsymbol{x}$$

- The $\boxplus$ operator applies a tangent-space increment $\Delta\boldsymbol{x} \in \mathbb{R}^n$ to the current state on the manifold, yielding a new valid element.
- For $\mathbb{R}^n$, $\boxplus$ reduces to standard addition.
- For $SO(3)$/$SE(3)$, $\boxplus$ uses the exponential map.
- This enables standard solvers (Gauss-Newton, LM) to work on manifold-valued states.

So, Now We Understand Box-minus ⊟

$$\chi^{\mathrm{MAP}} = \underset{\mathbf{X}\in\mathcal{X}}{\operatorname{argmin}} \sum_i \left\| h\left(\mathbf{X}_i, \mathbf{X}_{i+1}\right) \boxminus \mathbf{z}_i^{\mathrm{odom}} \right\|^2_{\Sigma_{\mathrm{odom}_i}} + \sum_{(j,k)\in\mathcal{I}} \left\| s_{jk} \left(h\left(\mathbf{X}_j, \mathbf{X}_k\right) \boxminus \mathbf{z}_{jk}^{\mathrm{loop}} \right) \right\|^2_{\Sigma_{\mathrm{loop}_{jk}}}$$

Example 1. IMU Pose Integration Using Lie Algebra

Raw IMU measurements (body frame b):

$$\boldsymbol{a}_m^b = \boldsymbol{R}_{wb}^\top(\boldsymbol{a}^w - \boldsymbol{g}^w) + \boldsymbol{a}_{bt} + \boldsymbol{a}_n$$
$$\boldsymbol{\omega}_m^b = \boldsymbol{\omega}^b + \boldsymbol{\omega}_{bt} + \boldsymbol{\omega}_n$$

where $\boldsymbol{g}^w = [0, 0, -9.81]^\top$, $(\cdot)_{bt}$ = bias, $(\cdot)_n$ = noise, $(\cdot)_m^b$ = measured w.r.t. body frame.

After bias correction $\Rightarrow$ world-frame quantities:

$$\boldsymbol{a}^w = \boldsymbol{R}_{wb}(\boldsymbol{a}_m^b - \boldsymbol{a}_{bt}) + \boldsymbol{g}^w$$
$$\boldsymbol{\omega}^b = \boldsymbol{\omega}_m^b - \boldsymbol{\omega}_{bt}$$

State: $(\boldsymbol{R}_k, \boldsymbol{v}_k^w, \boldsymbol{p}_k^w) \in SE(3) \times \mathbb{R}^6$

Discrete integration (Δt = IMU period):

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k \exp([\boldsymbol{\omega}^b \Delta t]_\times)$$
$$\boldsymbol{v}_{k+1}^w = \boldsymbol{v}_k^w + \boldsymbol{a}^w \Delta t$$
$$\boldsymbol{p}_{k+1}^w = \boldsymbol{p}_k^w + \boldsymbol{v}_k^w \Delta t + \tfrac{1}{2}\boldsymbol{a}^w \Delta t^2$$

Rotation uses $\exp(\cdot)$ on $SO(3)$; position/velocity are plain $\mathbb{R}^3$ additions.

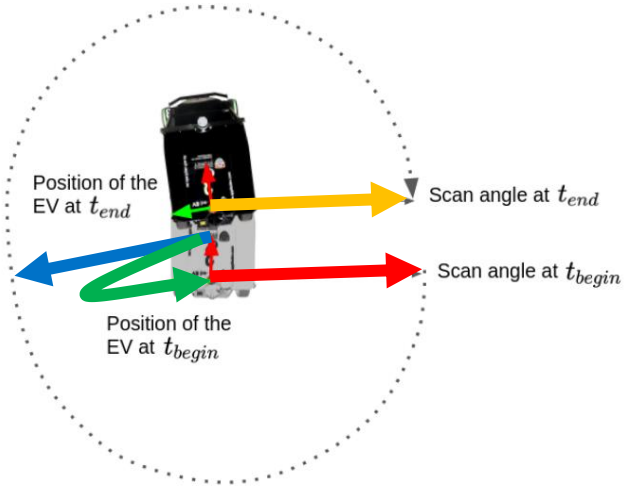
Without Lie algebra: $\boldsymbol{R}_k + \Delta\boldsymbol{R}$ breaks $SO(3)$ every step. With $\exp(\cdot)$, rotation stays on the manifold automatically.

Example 2: LiDAR Point Cloud Deskewing

Problem: A spinning LiDAR captures ~100 ms per sweep. The robot moves during the sweep, so each point $\boldsymbol{p}_i$ is captured at a different pose — causing "motion blur" in 3D.

Step 1 — Estimate inter-scan velocity (constant velocity model).
Using the two previous scan poses $\boldsymbol{T}_{t-2}, \boldsymbol{T}_{t-1} \in SE(3)$, compute the relative pose and extract the twist:

$$\boldsymbol{\xi} = \log\left(\boldsymbol{T}_{t-2}^{-1}\, \boldsymbol{T}_{t-1}\right) \in \mathbb{R}^6 \qquad \text{(assumed constant from } t{-}1 \text{ to } t\text{)}$$



Step 2 — Interpolate pose within the current sweep $[t,\, t{+}T]$.
For point $\boldsymbol{p}_i$ captured at fractional time $\alpha_i = (t_i - t)/T \in [0, 1]$:

$$\boldsymbol{T}(\alpha_i) = \exp\left(\alpha_i\, [\boldsymbol{\xi}]_\wedge\right) \in SE(3)$$

Step 3 — Compensate motion (bring each point to sweep-start frame):

$$\boldsymbol{p}_i' = \boldsymbol{T}(\alpha_i)^{-1}\, \boldsymbol{p}_i$$

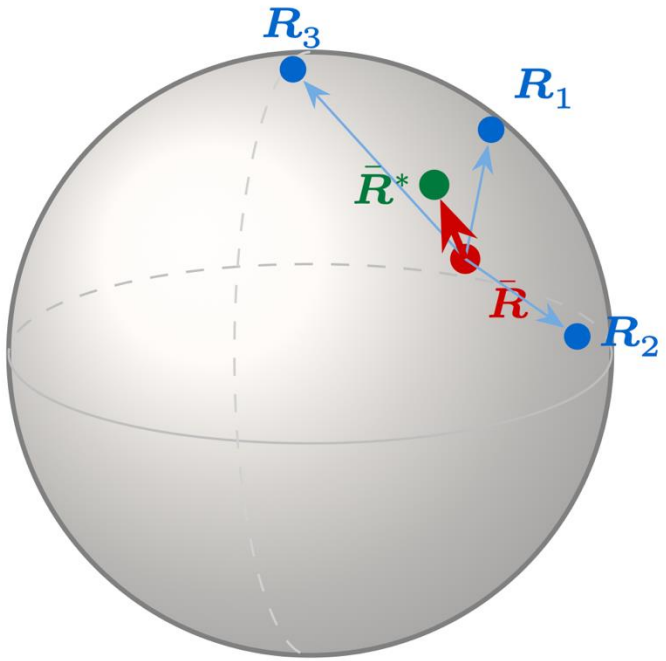
Used in: Every LiDAR-(inertial) odometry approaches uses deskewing to get better point cloud measurement. Empirically, sometimes rotation compensation is enough.

Example 3: Rotation Averaging (Kärcher Mean)

Problem. Given $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n$, find their mean $\bar{\boldsymbol{R}}$.
Naive averaging fails: $\frac{1}{n}\sum \boldsymbol{R}_i \notin SO(3)$ (×).

Solution: Kärcher mean

1. Initialize $\bar{\boldsymbol{R}} \leftarrow \boldsymbol{I}$.
2. Residuals in algebra: $\hat{\boldsymbol{\omega}}_i = \log(\boldsymbol{R}_i \bar{\boldsymbol{R}}^{-1})$
3. Average in $\mathbb{R}^3$: $\hat{\boldsymbol{\omega}} = \frac{1}{n}\sum_i \hat{\boldsymbol{\omega}}_i$
4. Update: $\bar{\boldsymbol{R}} \leftarrow \exp(\tau\hat{\boldsymbol{\omega}})\,\bar{\boldsymbol{R}}$
5. Repeat until $\|\hat{\boldsymbol{\omega}}\| < \varepsilon$.

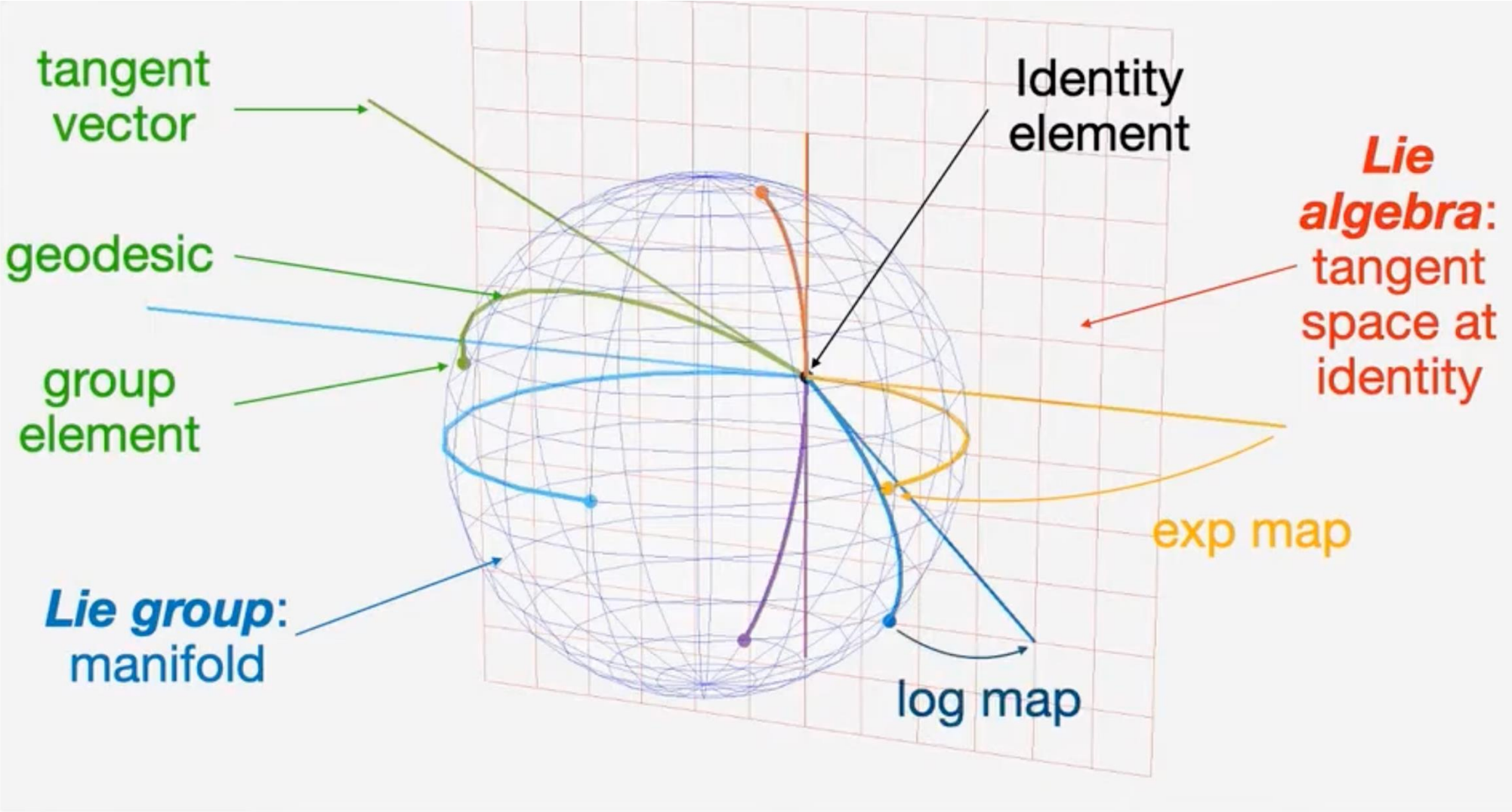


$SO(3)$ manifold

Key insight

Averaging in flat $\mathbb{R}^3$; exp maps back to $SO(3)$.

Conclusion



https://www.youtube.com/watch?v=QR1p0Rabuww&t=903s