Notes on Lie Theory (SO3, SE3)

Gyubeom Edward Im*

February 24, 2024

Contents

*blog: alida.tistory.com, email: criterion.im@gmail.com

1 Introduction

In this post, we will discuss Lie Theory, which is used in SLAM. When studying the optimization part of SLAM, optimization methods based on Lie Theory often appear, but without prior knowledge of the content, it is difficult to understand the optimization process. Therefore, this post briefly summarizes the essential content needed to understand the optimization part of SLAM. Most of the content is written referring to Joan Solà's *Lie theory for the roboticist* YouTube video.

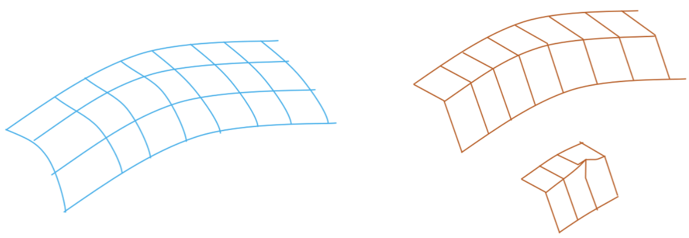
2 Group Theory

A group (Group) refers to an algebraic structure consisting of a set and a binary operation between two elements. For example, if a set is denoted as A and a binary operation as $*$, a group can be represented as $G = (A, *)$. Common sets of numbers such as integers, rational numbers, real numbers, and complex numbers, and operations such as addition and multiplication, belong to groups.

Groups generally have the following characteristics:

- Associativity: For any three elements $a, b, c \in G$ in the group, the associative law $(a*b)*c = a*(b*c)$ holds.
- Identity element: If there exists an element $e \in G$ such that $a * e = a = e * a$ for any element $a \in G$, then e is called the identity element.
- Inverse: If there exists an element $x \in G$ such that $a * x = e = x * a$ for any element $a \in G$, then x is called the inverse, and is sometimes denoted as a^{-1}.
- Composition: For any two elements $a, b \in G$, $a * b \in G$ holds. In most groups, the binary operation $*$ does not commute, meaning $a * b \neq b * a$.

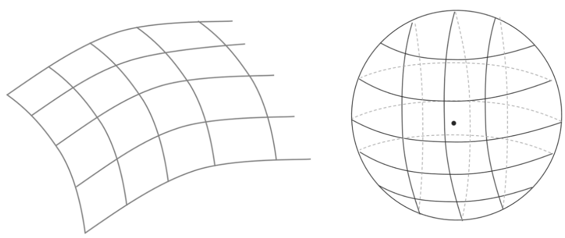
2.1 Lie Group



Smooth Manifold　　Non-Smooth Manifold (Edge, Spike)

Among various groups, a Lie Group is a group that has a Smooth Manifold. Here, a Smooth Manifold refers to a manifold where all elements of the group exist without edges or spikes, as shown in the figure. All elements existing on the Smooth Manifold have the characteristic of being differentiable.

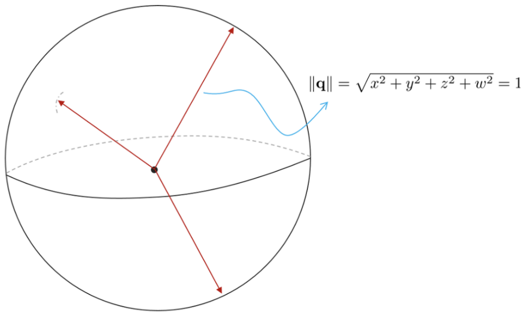
2.2 Manifold



Manifold $\mathcal{M}$

An N-dimensional Manifold $\mathcal{M}$ refers to a geometric space where any point $\mathbf{x} \in \mathcal{M}$ within $\mathcal{M}$ locally has a Euclidean structure. In other words, all points near $\mathbf{x}$ have the characteristic of being topologically homeomorphic to the $\mathbb{R}^N$ space. Intuitively, a Manifold represents the constraint space of the group.

For example, for any quaternion $\mathbf{q} = [x, y, z, w]$ to be used as a 3D rotation operator, it must satisfy the properties of a unit quaternion, where the constraint is $\|\mathbf{q}\| = 1$. This means that $\mathbf{q}$ must satisfy a point on a 4-dimensional Manifold, referred to as the unit quaternion manifold. Constraints of four dimensions or more cannot be visualized on a plane, so they are typically explained using a 3-dimensional sphere.



Unit Quaternion should meet the constraint $\|\mathbf{q}\| = 1$

In the case of the SE(3) Lie Group to be discussed later, 6-dimensional elements must satisfy specific constraints. That is, elements of the SE(3) Lie Group must exist on a 6-dimensional Manifold. Since this cannot be explained on a plane, a 3-dimensional sphere is used to illustrate it.

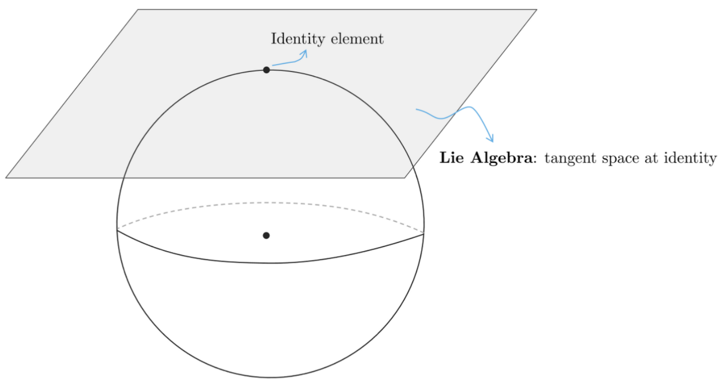
2.3 Group Action

One of the characteristics of a group is that it can transform (=act) another set or group. This means that the group can function as an operator transforming a specific set. This demonstrates that Lie Theory is a suitable tool for representing the movement of objects in 3-dimensional space.

For example, given a rotation matrix $\mathbf{R} \in SO(3)$, a 3-dimensional vector $\mathbf{x} \in \mathbb{R}^3$, and a binary operation $\cdot$: - $\mathbf{R}$ can rotate (=act on) a point in vector space $\rightarrow \mathbf{x}' = \mathbf{R} \cdot \mathbf{x}$

Given a transformation matrix $\mathbf{T} \in SE(3)$, a 4-dimensional vector $\mathbf{X} \in \mathbb{R}^4$, and a binary operation $\cdot$: - $\mathbf{T}$ can transform (=act on) a point in vector space $\rightarrow \mathbf{X}' = \mathbf{T} \cdot \mathbf{X}$

2.4 Topology of Lie Theory



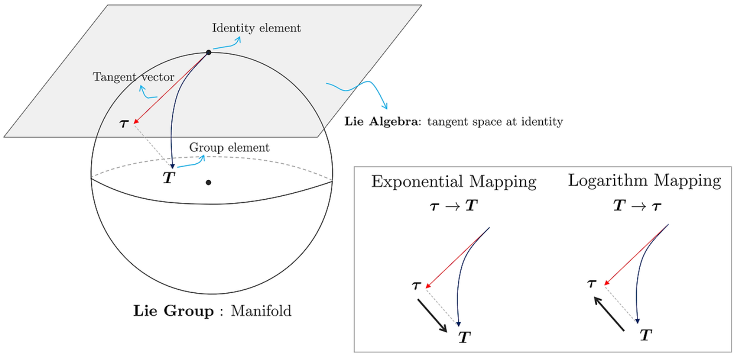
Lie Group : Manifold

The geometric structure of Lie Theory is as shown in the figure above.

- A Lie Group represents a collection of points on a manifold with constraints and has **non-linear characteristics**.

- The Lie Algebra represents the tangent space at the identity element on the manifold. This tangent space of the Lie Algebra is only valid at the identity element and has the characteristic of a **linear vector space**.

The reason why Lie Groups and Lie Algebras are important is because a 1:1 transformation is possible between the two spaces. **Therefore, instead of directly operating in the complex constraint-laden non-linear manifold space (Lie Group), operations are performed in the relatively simple linear vector space (Lie Algebra) and then transformed back to the manifold space.** The operations that make this possible are the Exponential Mapping and Logarithm Mapping.

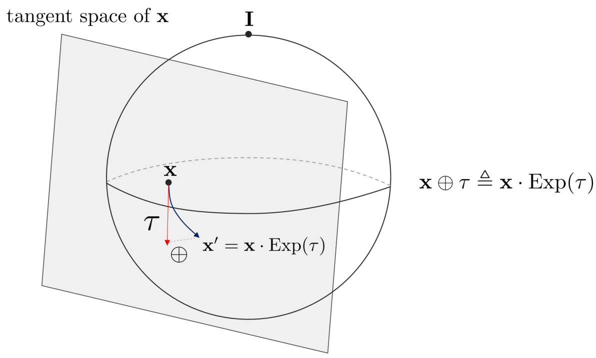


Exponential Mapping refers to the operation that transforms from Lie Algebra to Lie Group, while Logarithm Mapping is the opposite operation that transforms from Lie Group to Lie Algebra.

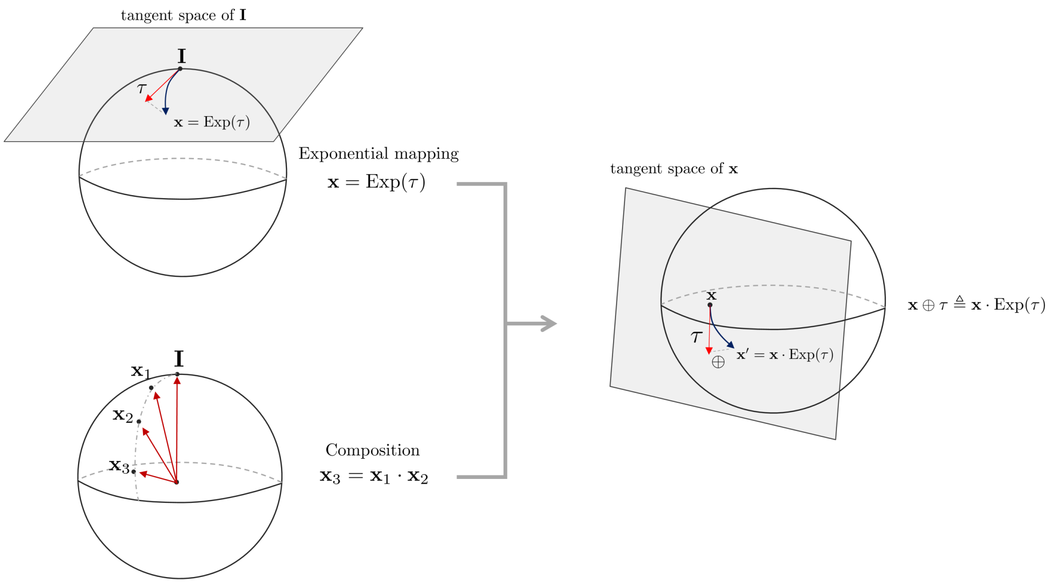
2.5 Plus and Minus Operators of Lie Group

Using the previously mentioned exponential mapping, any element τ of Lie Algebra can be used to transform an element $\mathbf{x}$ of Lie Group. Since the usual $+, -$ operators do not apply between elements of the Lie Group and Lie Algebra, new operators $\oplus, \ominus$ must be defined. The $\oplus$ operator applies an additional transformation to an arbitrary Lie Group element $\mathbf{x}$ by the amount τ.

$$\mathbf{x} \oplus \tau \triangleq \mathbf{x} \cdot \text{Exp}(\tau) \tag{1}$$

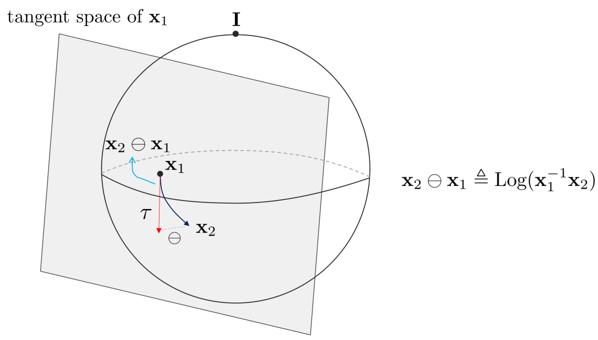


At this time, the τ vector on the tangent space of x is treated as identical to the τ vector on the tangent space of the identity element due to the composition properties of the lie group.

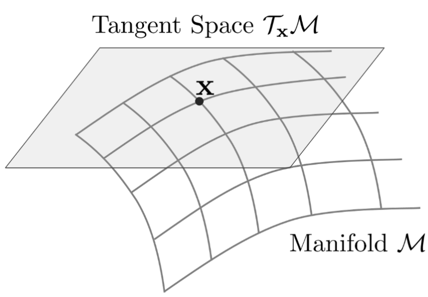


Conversely, the $\ominus$ operator is used in cases like $\mathbf{x}_2 \ominus \mathbf{x}_1$ when two Lie Group elements $\mathbf{x}_1, \mathbf{x}_2$ exist, indicating the relative change from $\mathbf{x}_1$ to $\mathbf{x}_2$.

$$\mathbf{x}_2 \ominus \mathbf{x}_1 \triangleq \mathrm{Log}(\mathbf{x}_1^{-1}\mathbf{x}_2) \tag{2}$$



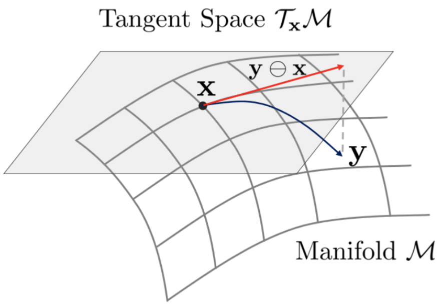
2.6 Tangent Space and Lie Algebra



When there exists a point $\mathbf{x} \in \mathcal{M}$ on an arbitrary manifold $\mathcal{M}$, its tangent space is denoted as $\mathcal{T}_{\mathbf{x}}\mathcal{M}$.

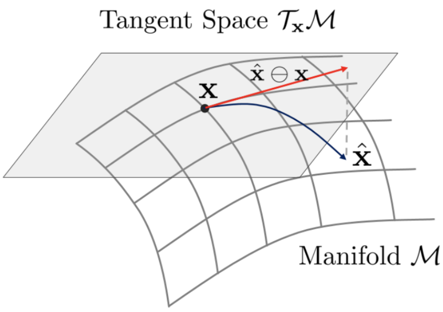
The tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ is uniquely determined for each point and has the characteristic of being a vector space, enabling calculus operations. The dimension of the tangent space is determined by the number of degrees of freedom of the manifold. For instance, SO(3) has three degrees of freedom for rotation, so so(3) has a three-dimensional tangent space, while SE(3), which accounts for pose, has six degrees of freedom, making se(3) a six-dimensional tangent space. The tangent space at the identity element is specifically referred to as the Lie Algebra.

2.7 Calculus on Lie Group



Let us assume that there exist two elements $\mathbf{x}, \mathbf{y}$ on a Lie Group. In this case, the difference between the two elements can be represented on the tangent space using the $\ominus$ operator as $\mathbf{y} \ominus \mathbf{x}$. As mentioned earlier, the tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ is a linear vector space, which makes it relatively easy to calculate the jacobian and covariance. This characteristic is a key reason Lie Theory can be used for optimization calculations. For example, an error function model can be used when the two elements are relatively close.

$$\mathbf{e} = \hat{\mathbf{x}} \ominus \mathbf{x} \tag{3}$$



2.8 Jacobians on Lie Group

Given a vector function $f(\mathbf{x})$ existing in a linear vector space, suppose that $f(\mathbf{x})$ satisfies $\mathbf{x} \in \mathbb{R}^n, f(\mathbf{x}) \in \mathbb{R}^m$.

$$f : \mathbb{R}^n \mapsto \mathbb{R}^m \tag{4}$$

In this case, the first partial derivative of the vector function becomes a matrix, which is specifically referred to as the jacobian.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{5}$$

When represented using a small change $\mathbf{h}$, it can be expressed as follows:

$$\mathbf{J} = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \triangleq \lim_{\mathbf{h} \to \mathbf{0}} \frac{f(\mathbf{x} + \mathbf{h}) - f(\mathbf{x})}{\mathbf{h}} \in \mathbb{R}^{m \times n} \tag{6}$$

In a linear vector space, $+, -$ operators can be used to calculate the jacobian. However, as mentioned earlier, since the elements of a Lie Group are not closed under $+, -$ operations, the jacobian cannot be expressed using traditional methods.

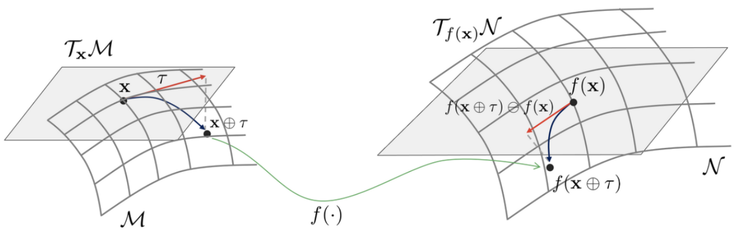
Given a vector function $f(\mathbf{x})$ existing on a Lie Group as follows:

$$f : \mathcal{M} \mapsto \mathcal{N}; \mathbf{x} \mapsto \mathbf{y} = f(\mathbf{x}) \tag{7}$$

This function maps an element $\mathbf{x}$ on the manifold $\mathcal{M}$ to another element $\mathbf{y}$ on the manifold $\mathcal{N}.Thejacobianforthiscanbe$

$$\mathbf{J} = \frac{Df(\mathbf{x})}{D\mathbf{x}} = \lim_{\tau \to 0} \frac{f(\mathbf{x} \oplus \tau) \ominus f(\mathbf{x})}{\tau} \in \mathbb{R}^{m \times n} \tag{8}$$

The jacobian $\mathbf{J}$ can be thought of as a function mapping an element on $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ to an element on $\mathcal{T}_{f(\mathbf{x})}\mathcal{N}$. This can be illustrated as follows:



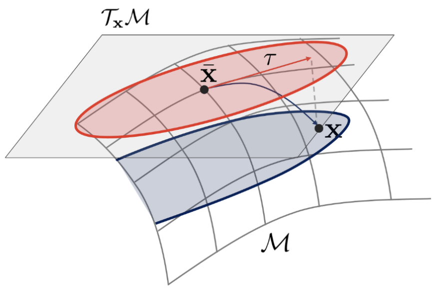
2.9 Perturbations on Lie Group

Using the property that the tangent space is a linear vector space, elements of the Lie Group can be modeled as perturbations of a random variable.

$$\mathbf{x} = \bar{\mathbf{x}} \oplus \tau \quad \text{where, } \tau = \mathbf{x} \ominus \bar{\mathbf{x}} \tag{9}$$

$$\mathbf{P}_{\mathbf{x}} = \mathbb{E}[\tau \cdot \tau^{\intercal}] \tag{10}$$

$$\mathbf{P}_{\mathbf{x}} = \mathbb{E}[(\mathbf{x} \ominus \bar{\mathbf{x}}) \cdot (\mathbf{x} \ominus \bar{\mathbf{x}})^{\intercal}] \tag{11}$$



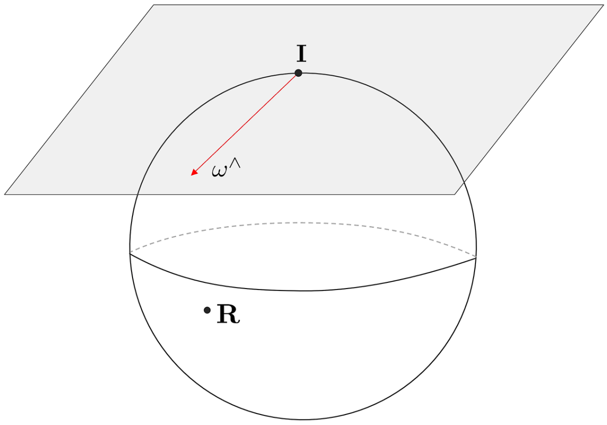
If a function $\mathbf{y} = f(\mathbf{x})$ is given, the jacobian becomes $\mathbf{J} = \frac{D\mathbf{y}}{D\mathbf{x}}$ and the covariance $\mathbf{P}_{\mathbf{y}}$ can be calculated as follows:

$$\mathbf{P}_{\mathbf{y}} = \mathbf{J} \cdot \mathbf{P}_{\mathbf{x}} \cdot \mathbf{J}^{\intercal} \tag{12}$$

This follows the same formula for covariance propagation in vector space.

3 SO(3) Group

tangent space of so(3)



SO(3) Manifold

3.1 Lie Group SO(3)

One of the Lie groups, the Special Orthogonal 3 (SO(3)) group, consists of 3-dimensional rotation matrices and operations that are closed under these matrices. It is used to represent the rotation of 3D objects.

$$SO(3) = \{\mathbf{R} \in \mathbb{R}^{3\times3} \mid \mathbf{R}\mathbf{R}^T = \mathbf{I}, \det(\mathbf{R}) = 1\} \tag{13}$$

3.1.1 SO(3) group properties

- Associativity: Associativity holds as $(\mathbf{R}_1 \cdot \mathbf{R}_2) \cdot \mathbf{R}_3 = \mathbf{R}_1 \cdot (\mathbf{R}_2 \cdot \mathbf{R}_3)$
- Identity element: An identity matrix $\mathbf{I}$ exists such that $\mathbf{R} \cdot \mathbf{I} = \mathbf{I} \cdot \mathbf{R} = \mathbf{R}$
- Inverse: An inverse matrix exists such that $\mathbf{R}^{-1} \cdot \mathbf{R} = \mathbf{R} \cdot \mathbf{R}^{-1} = \mathbf{I}$. Due to the properties of the SO(3) group, $\mathbf{R}^{-1} = \mathbf{R}^\intercal$. Thus, $\mathbf{R} \cdot \mathbf{R}^\intercal = \mathbf{I}$.
- Composition: The composition in the SO(3) group is performed by matrix multiplication as follows

$$\mathbf{R}_1 \cdot \mathbf{R}_2 = \mathbf{R}_3 \in SO(3) \tag{14}$$

- Non-commutative: The commutative property does not hold as $\mathbf{R}_1 \cdot \mathbf{R}_2 \neq \mathbf{R}_2 \cdot \mathbf{R}_1$
- Determinant: The determinant of $\mathbf{R}$ satisfies $\det(\mathbf{R}) = 1$. (Only pure rotation without reflection or inversion is represented)
- Rotation: A point or vector $\mathbf{x} = \begin{bmatrix} x & y & z \end{bmatrix}^\intercal \in \mathbb{P}^2$ space can be rotated to another point or vector $\mathbf{x}'$.

$$\mathbf{x}' = \mathbf{R} \cdot \mathbf{x} \tag{15}$$

- Adjoint: Given any rotation matrix $\mathbf{R}$ and an angular velocity ω that exists on the tangent plane of $\mathbf{R}$, the properties of the adjoint matrix yield the following useful formulas.

$$\exp((\mathbf{R}\omega)^\wedge) = \mathbf{R}\exp(\omega^\wedge)\mathbf{R}^\intercal = \exp(\mathbf{R}\omega^\wedge\mathbf{R}^\intercal) \tag{16}$$

3.2 Lie Algebra so(3)

The Lie Algebra so(3) of the SO(3) group consists of the following $\mathbb{R}^{3\times3}$ size skew-symmetric matrices:

$$\text{so}(3) = \left\{ \omega^\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \in \mathbb{R}^{3\times3} \;\middle|\; \omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \in \mathbb{R}^3 \right\} \tag{17}$$

The generators of so(3) are derived from rotations about each axis from the origin and signify orthogonal basis matrices.

$$G_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad G_3 = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{18}$$

Each element of so(3) can be expressed as a linear combination of the generators.

$$\begin{aligned} &\omega \in \mathbb{R}^3 \\ &\omega_1 G_1 + \omega_2 G_2 + \omega_3 G_3 \in so(3) \end{aligned} \tag{19}$$

Here, ω is a 3-dimensional vector representing angular velocity about an arbitrary axis. By generating a skew-symmetric matrix through ω, it becomes so(3).

$$\omega^\wedge \in so(3) \tag{20}$$

3.3 Exponential Mapping and Logarithm Mapping

The Lie Group SO(3) and the Lie Algebra so(3) are uniquely matched through exponential mapping and logarithm mapping.

$$\begin{aligned} \exp(\omega^\wedge) &= \mathbf{R} \in SO(3) \\ \log(\mathbf{R}) &= \omega^\wedge \in so(3) \end{aligned} \tag{21}$$

Expanding $\exp(\omega^\wedge)$ according to the definition of Exponential Mapping results in:

$$\begin{aligned} \exp(\omega^\wedge) &= \exp\left(\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \right) \\ &= \mathbf{I} + \omega^\wedge + \frac{1}{2!}(\omega^\wedge)^2 + \frac{1}{3!}(\omega^\wedge)^3 + \cdots \end{aligned} \tag{22}$$

In three-dimensional space, an arbitrary angular velocity ω can be separated into magnitude $|\omega|$ and unit vector $\mathbf{u}$.

$$\begin{aligned} \omega &= |\omega|\mathbf{u} \\ &= \theta\mathbf{u} \quad (\theta = |\omega|) \end{aligned} \tag{23}$$

Applying the characteristic of the skew-symmetric matrix $(\omega^\wedge)^3 = -(\omega^T\omega) \cdot \omega^\wedge = -\theta^2\omega^\wedge$ results in:

$$\begin{aligned} &\theta^2 = \omega^T\omega \\ &(\omega^\wedge)^{2i+1} = (-1)^i\theta^{2i}\omega^\wedge \\ &(\omega^\wedge)^{2i+2} = (-1)^i\theta^{2i}(\omega^\wedge)^2 \end{aligned} \tag{24}$$

Reorganizing the formula through Taylor expansion results in:

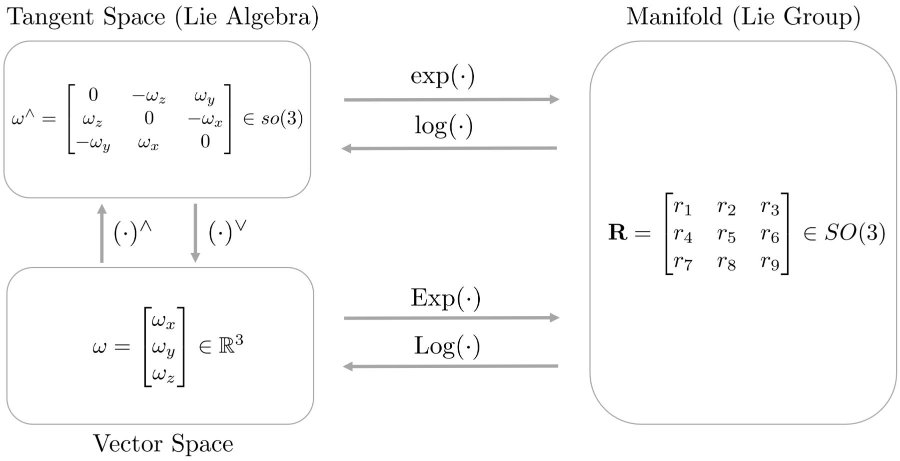
$$\begin{aligned} \exp(\omega^\wedge) &= \mathbf{I} + \omega^\wedge + \frac{1}{2!}(\omega^\wedge)^2 + \frac{1}{3!}(\omega^\wedge)^3 + \cdots \\ &= \mathbf{I} + \left(\sum_{i=0}^{\infty} \frac{(-1)^i\theta^{2i}}{(2i+1)!} \right)\omega^\wedge + \left(\sum_{i=0}^{\infty} \frac{(-1)^i\theta^{2i}}{(2i+2)!} \right)(\omega^\wedge)^2 \\ &= \mathbf{I} + \left(\frac{\sin\theta}{\theta} \right)\omega^\wedge + \left(\frac{1-\cos\theta}{\theta^2} \right)(\omega^\wedge)^2 \end{aligned} \tag{25}$$

This formula is called the Rodrigues Formula. It represents the relationship between angle-axis notation and the rotation matrix $\mathbf{R}$, acting as an axis of rotation and rotating by an angle θ. The Rodrigues Formula allows for exponential mapping from $\omega^\wedge \in so(3)$ to $\mathbf{R} \in SO(3)$.

For convenience in operation, the mapping process generally uses the $\exp(\cdot)$ operator for mapping $\omega^\wedge \to \mathbf{R}$ and the $\mathrm{Exp}(\cdot)$ operator for mapping $\omega \to \mathbf{R}$. The logarithm mapping follows similarly.

$$\begin{aligned} \exp(\cdot) &: \omega^\wedge \mapsto \mathbf{R} \\ \mathrm{Exp}(\cdot) &: \omega \mapsto \mathbf{R} \end{aligned} \tag{26}$$

The diagram summarizes this as follows.



3.4 Derivation of Exponential Mapping

An arbitrary rotation matrix satisfies the following property:

$$\mathbf{R}^\intercal\mathbf{R} = \mathbf{I} \tag{27}$$

If we consider $\mathbf{R}$ as a continuously changing camera rotation, it can be denoted as a function of time $\mathbf{R}(t)$.

$$\mathbf{R}^\intercal(t)\mathbf{R}(t) = \mathbf{I} \tag{28}$$

Differentiating both sides of the equation with respect to time, we obtain:

$$\begin{aligned} \dot{\mathbf{R}}^\intercal(t)\mathbf{R}(t) + \mathbf{R}^\intercal(t)\dot{\mathbf{R}}(t) = 0 \\ \mathbf{R}^\intercal(t)\dot{\mathbf{R}}(t) = -\Big(\mathbf{R}^\intercal(t)\dot{\mathbf{R}}(t)\Big)^\intercal \end{aligned} \tag{29}$$

From this, we can see that $\mathbf{R}^\intercal(t)\dot{\mathbf{R}}(t)$ satisfies the properties of a skew-symmetric matrix.

Any skew-symmetric matrix $\mathbf{A}$ is defined as follows. An operator that converts a 3-dimensional vector $\mathbf{a}$ into $\mathbf{A}$ is defined as $(\cdot)^\wedge$, and the reverse operation as $(\cdot)^\vee$.

$$\mathbf{a}^\wedge = \mathbf{A} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}, \quad \mathbf{A}^\vee = \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{30}$$

From any skew-symmetric matrix, a corresponding vector can be found. Thus, $\mathbf{R}^\intercal(t)\dot{\mathbf{R}}(t)$ can be associated with a 3-dimensional vector $\omega(t) \in \mathbb{R}^3$.

$$\mathbf{R}^\intercal(t)\dot{\mathbf{R}}(t) = \omega(t)^\wedge. \tag{31}$$

Multiplying both sides of the equation by $\mathbf{R}(t)$, and since $\mathbf{R}(t)$ is an orthogonal matrix, we obtain the following expression:

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t)\omega(t)^\wedge = \mathbf{R}(t)\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{32}$$

Assuming that the rotation matrix $\mathbf{R}(0) = \mathbf{I}$ at time $t_0 = 0$, $\dot{\mathbf{R}}(0)$ can be represented as follows:

$$\dot{\mathbf{R}}(0) = \omega(0)^\wedge \tag{33}$$

Here, ω represents the tangent plane at the origin of SO(3). Assuming ω is constant near $t_0 = 0$, we have $\omega(t_0) = \omega$.

$$\dot{\mathbf{R}}(t) = \omega(0)^\wedge = \omega^\wedge \tag{34}$$

The equation being a differential equation, its solution is as follows:

$$\mathbf{R}(t) = \mathbf{R}_0 \exp(\omega^\wedge t). \tag{35}$$

Setting $\mathbf{R}_0 = \mathbf{R}(0) = \mathbf{I}$ and omitting the time function representation as $\omega^\wedge t \to \omega^\wedge$, we have:

$$\mathbf{R} = \exp(\omega^\wedge). \tag{36}$$

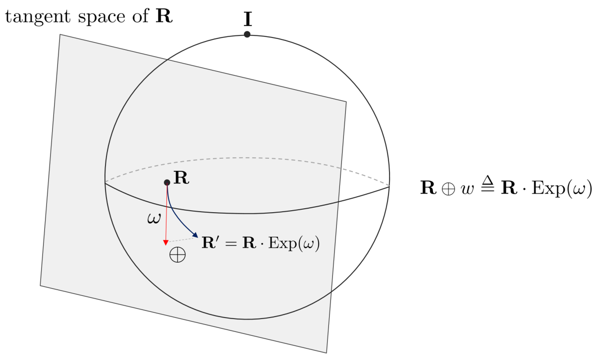
This expression means that the rotation matrix can be calculated through $\exp(\omega^\wedge)$.

3.5 Plus and Minus Operator of SO(3)

Using the previously described exponential mapping, we can transform $\mathbf{R}$ using the angular velocity ω. As the usual $+$ and $-$ operators do not apply between SO(3) and so(3), new $\oplus$ and $\ominus$ operators must be defined. The $\oplus$ operator applies an additional rotation of ω to an arbitrary rotation matrix $\mathbf{R}$.

$$\oplus : SO(3) \times \mathbb{R}^3 \mapsto SO(3) \tag{37}$$

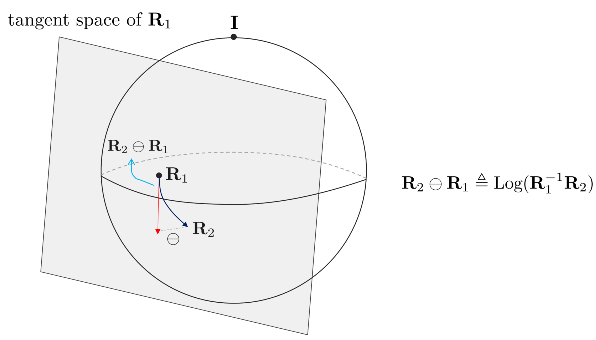
$$\mathbf{R} \oplus \omega \triangleq \mathbf{R} \cdot \mathrm{Exp}(\omega) \tag{38}$$



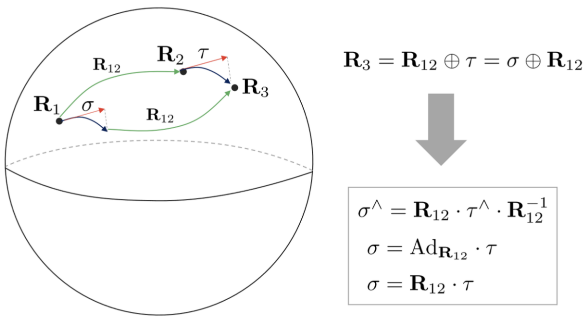
Conversely, the $\ominus$ operator is used when two rotation matrices $\mathbf{R}_1, \mathbf{R}_2$ exist, and it is used as $\mathbf{R}_2 \ominus \mathbf{R}_1$, representing the difference in rotation from $\mathbf{R}_1$ to $\mathbf{R}_2$.

$$\ominus : SO(3) \times SO(3) \mapsto \mathbb{R}^3 \tag{39}$$

$$\mathbf{R}_2 \ominus \mathbf{R}_1 \triangleq \mathrm{Log}(\mathbf{R}_1^{-1}\mathbf{R}_2) \tag{40}$$



3.6 Adjoint Matrix of SO(3)



SO(3) Manifold

The Adjoint matrix of the SO(3) group transforms an arbitrary angular velocity $\tau \in \mathbb{R}^3$ on the tangent plane of $\mathbf{R}_2 \in SO(3)$ to another angular velocity σ on the tangent plane of $\mathbf{R}_1$. If we define the Adjoint matrix with respect to $\mathbf{R}_{12} \in SO(3)$ as $\mathrm{Ad}_{\mathbf{R}_{12}}$, the following holds:

$$\sigma = \mathrm{Ad}_{\mathbf{R}_{12}}\tau \tag{41}$$

Since it transforms one angular velocity $\tau \in \mathbb{R}^3$ to another angular velocity σ, $\mathrm{Ad}_{\mathbf{R}_{12}}$ has dimensions $\mathbb{R}^{3\times 3}$.

Additionally, the Adjoint matrix satisfies the following equations:

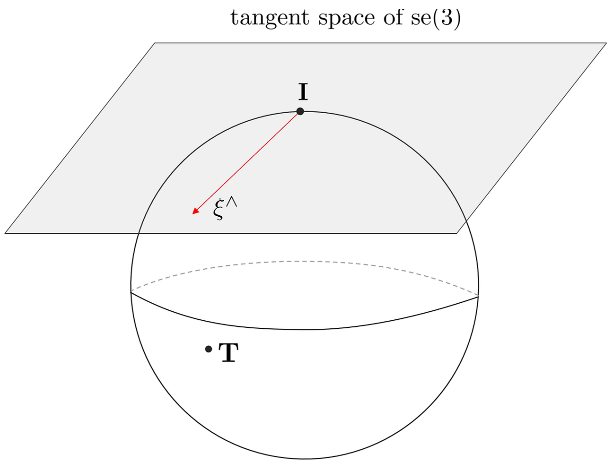
$$\begin{aligned} \mathbf{R}_{12} \cdot \exp(\tau^\wedge) &= \exp((\mathrm{Ad}_{\mathbf{R}_{12}} \cdot \tau)^\wedge) \cdot \mathbf{R}_{12} \\ \exp((\mathrm{Ad}_{\mathbf{R}_{12}} \cdot \tau)^\wedge) &= \mathbf{R}_{12} \cdot \exp(\tau^\wedge) \cdot \mathbf{R}_{12}^{-1} \end{aligned} \tag{42}$$

The derivation of the Adjoint matrix for the so(3) algebra is as follows:

$$\begin{aligned} (\mathrm{Ad}_{\mathbf{R}_{12}} \cdot \tau)^\wedge &= \mathbf{R}_{12} \cdot \left(\sum_{i=1}^{3} \tau_i G_i \right) \cdot \mathbf{R}_{12}^{-1} \\ &= \mathbf{R}_{12} \cdot \tau^\wedge \cdot \mathbf{R}_{12}^{-1} \\ &= (\mathbf{R}_{12}\tau)^\wedge \end{aligned} \tag{43}$$

$$\mathrm{Ad}_{\mathbf{R}_{12}} = \mathbf{R}_{12} \in \mathbb{R}^{3\times 3} \tag{44}$$

4 SE(3) Group



SE(3) Manifold

4.1 Lie Group SE(3)

Lie 군 중 하나인 Special Euclidean 3 (SE(3))군은 3차원 공간 상에서 강체의 변환과 관련된 행렬과 이에 닫혀있는 연산들로 구성된 군을 의미한다.

$$SE(3) = \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} \mid \mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3 \right\} \tag{45}$$

4.1.1 SE(3) group properties

- Associativity : $(\mathbf{T}_1 \cdot \mathbf{T}_2) \cdot \mathbf{T}_3 = \mathbf{T}_1 \cdot (\mathbf{T}_2 \cdot \mathbf{T}_3)$와 같이 결합 법칙이 성립한다
- Identity element : $\mathbf{T} \cdot \mathbf{I} = \mathbf{I} \cdot \mathbf{T} = \mathbf{T}$를 만족하는 4x4 항등 행렬 $\mathbf{I}$이 존재한다
- Inverse : $\mathbf{T}^{-1} \cdot \mathbf{T} = \mathbf{T} \cdot \mathbf{T}^{-1} = \mathbf{I}$을 만족하는 역행렬이 존재한다.

$$\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{R}^T & -\mathbf{R}^T\mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tag{46}$$

- Composition : SE(3)군의 합성은 아래와 같이 행렬의 곱셈 연산으로 수행한다

$$\begin{aligned} \mathbf{T}_1 \cdot \mathbf{T}_2 &= \begin{bmatrix} \mathbf{R}_1 & \mathbf{t}_1 \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_2 & \mathbf{t}_2 \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_1\mathbf{R}_2 & \mathbf{R}_1\mathbf{t}_2 + \mathbf{t}_1 \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3) \end{aligned} \tag{47}$$

- Non-commutative : $\mathbf{T}_1 \cdot \mathbf{T}_2 \neq \mathbf{T}_2 \cdot \mathbf{T}_1$ 교환 법칙이 성립하지 않는다
- Transformation : $\mathbb{P}^3$ 공간 상의 점 또는 벡터 $\mathbf{X} = \begin{bmatrix} X & Y & Z & W \end{bmatrix}^\intercal \in \mathbb{P}^3$를 다른 방향과 위치를 가지는 점 또는 벡터 $\mathbf{X}'$로 변환할 수 있다.

$$\begin{aligned} \mathbf{X}' = \mathbf{T} \cdot \mathbf{X} &= \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \mathbf{X} \\ &= \begin{bmatrix} \mathbf{R}(X\ Y\ Z)^T + W \cdot \mathbf{t} \\ W \end{bmatrix} \end{aligned} \tag{48}$$

4.2 Lie Algebra se(3)

SE(3)군의 Lie algebra se(3)는 다음과 같은 $\mathbb{R}^{4\times 4}$ 크기의 행렬로 정의된다.

$$\text{se}(3) = \left\{ \xi^\wedge = \begin{bmatrix} \omega^\wedge & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \;\middle|\; \xi = \begin{bmatrix} \omega \\ \mathbf{v} \end{bmatrix} \in \mathbb{R}^6 \right\} \tag{49}$$

이 때, ξ 는 3차원 공간 상에서 물체의 속도를 의미하는 twist 이며 $\omega = \begin{pmatrix} w_x & w_y & w_z \end{pmatrix}^T \in \mathbb{R}^3$ 는 각속도를 의미하고 $\mathbf{v} = \begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T \in \mathbb{R}^3$ 는 속도를 의미한다. ξ에서 ω와 $\mathbf{v}$의 순서는 종종 바뀌어 사용하기도 한다.

se(3)는 다음과 같이 회전과 이동에 관련된 6개의 생성자(Generator)들이 존재한다.

$$\begin{gathered} G_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad G_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad G_3 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ G_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad G_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad G_6 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{gathered} \tag{50}$$

se(3)의 각 원소들은 생성자들의 선형결합(Linear Combination)으로 표현할 수 있다.

$$\begin{aligned} &\xi = (\omega,\ \mathbf{v})^T \in \mathbb{R}^6 \\ &\omega_1 G_1 + \omega_2 G_2 + \omega_3 G_3 + v_1 G_4 + v_2 G_5 + v_3 G_6 \in se(3) \end{aligned} \tag{51}$$

4.3 Exponential Mapping and Logarithm Mapping

Lie Group SE(3)와 Lie Algebra se(3)는 지수 매핑(exponential mapping)과 로그 매핑(logarithm mapping)으로 서로 일대일 매칭되는 특징이 있다. 우선, twist $\xi \in \mathbb{R}^6$ 를 se(3) lie algebra로 변환하는 연산 $\xi^\wedge$ 은 다음과 같이 정의된다.

$$\xi^\wedge = \begin{bmatrix} \omega^\wedge & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4} \tag{52}$$

se(3)의 지수 매핑은 다음과 같이 정의된다.

$$\begin{aligned} \exp(\xi^\wedge) &= \exp\left(\begin{bmatrix} \omega^\wedge & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix}\right) \\ &= \mathbf{I} + \begin{bmatrix} \omega^\wedge & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} + \frac{1}{2!}\begin{bmatrix} (\omega^\wedge)^2 & \omega^\wedge\mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} + \frac{1}{3!}\begin{bmatrix} (\omega^\wedge)^3 & (\omega^\wedge)^2\mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix} + \cdots \end{aligned} \tag{53}$$

이 때, 회전과 관련된 부분은 SO(3)군과 동일하지만 이동과 관련된 부분은 별도의 급수 형태를 가진다.

$$\begin{aligned} \exp\left(\begin{bmatrix} \omega^\wedge & \mathbf{v} \\ \mathbf{0} & 0 \end{bmatrix}\right) &= \begin{bmatrix} \sum_0^\infty \frac{1}{n!}(\omega^\wedge)^n & \sum_0^\infty \frac{1}{(n+1)!}(\omega^\wedge)^n\mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \exp(\omega^\wedge) & \mathbf{Q}\mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix} \end{aligned} \tag{54}$$

$$\mathbf{Q} = \mathbf{I} + \frac{1}{2!}\omega^\wedge + \frac{1}{3!}(\omega^\wedge)^2 + \frac{1}{4!}(\omega^\wedge)^3 + \frac{1}{5!}(\omega^\wedge)^4 + \cdots \tag{55}$$

앞서 SO(3) 파트에서 설명한 Rodrigues 공식과 반대칭 행렬의 원리를 적용하여 다시 정리하면 다음과 같다.

$$\begin{aligned} \mathbf{Q} &= \mathbf{I} + \frac{1}{2!}\omega^\wedge + \frac{1}{3!}(\omega^\wedge)^2 + \frac{1}{4!}(\omega^\wedge)^3 + \frac{1}{5!}(\omega^\wedge)^4 + \cdots \\ &= \mathbf{I} + \left(\frac{1}{2!} - \frac{\theta^2}{4!} + \frac{\theta^4}{6!} + \cdots\right)\omega^\wedge + \left(\frac{1}{3!} - \frac{\theta^2}{5!} + \frac{\theta^4}{7!} + \cdots\right)(\omega^\wedge)^2 \\ &= \mathbf{I} + \sum_{i=0}^\infty \left[\tfrac{(\omega^\wedge)^{2i+1}}{(2i+2)!} + \tfrac{(\omega^\wedge)^{2i+2}}{(2i+3)!}\right] \\ &= \mathbf{I} + \left(\sum_{i=0}^\infty \frac{(-1)^i\theta^{2i}}{(2i+2)!}\right)\omega^\wedge + \left(\sum_{i=0}^\infty \frac{(-1)^i\theta^{2i}}{(2i+3)!}\right)(\omega^\wedge)^2 \\ &= \mathbf{I} + \left(\frac{1-\cos\theta}{\theta^2}\right)\omega^\wedge + \left(\frac{\theta - \sin\theta}{\theta^3}\right)(\omega^\wedge)^2 \end{aligned} \tag{56}$$

지금까지 유도과정을 정리하여 se(3)의 지수 매핑을 표현하면 다음과 같다.

$$\begin{aligned} &\xi = (\omega, \mathbf{v}) \in \mathbb{R}^6 \\ &\theta = |\omega| \\ &\theta^2 = \omega^\intercal\omega \\ &A = \frac{\sin\theta}{\theta} \\ &B = \frac{1-\cos\theta}{\theta^2} \\ &C = \frac{1-A}{\theta^2} = \frac{\theta-\sin\theta}{\theta^3} \\ &\mathbf{R} = \mathbf{I} + A\omega^\wedge + B(\omega^\wedge)^2 \\ &\mathbf{Q} = \mathbf{I} + B\omega^\wedge + C(\omega^\wedge)^2 \end{aligned} \tag{57}$$

$$\exp(\xi^\wedge) = \begin{bmatrix} \mathbf{R} & \mathbf{Q}\mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix} \tag{58}$$

SE(3)군의 로그 매핑은 회전 부분은 SO(3)군의 로그 매핑을 사용하고 이동 부분은 $\mathbf{Q}^{-1}$ 을 사용한다.

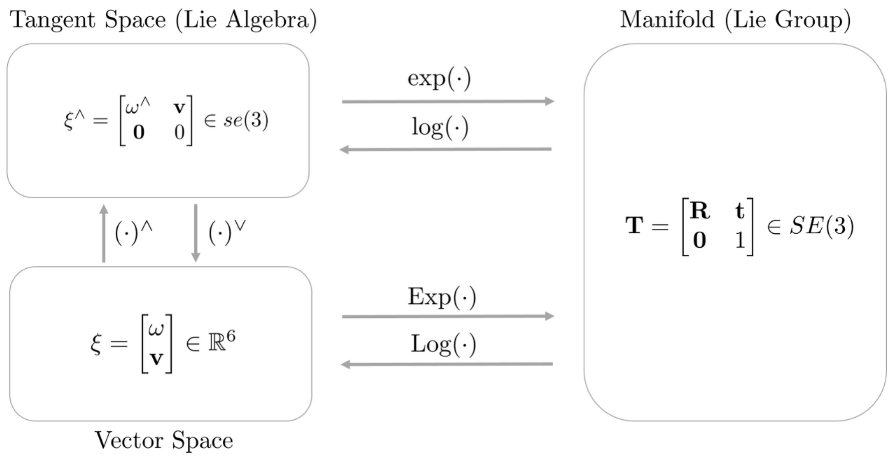
$$\mathbf{Q}^{-1} = \mathbf{I} - \frac{1}{2!}\omega^\wedge + \frac{1}{\theta^2}\left(1 - \frac{A}{2B}\right)(\omega^\wedge)^2 \tag{59}$$

$$
\begin{aligned}
\omega^{\wedge} &= \log(\mathbf{R}) \\
\mathbf{v} &= \mathbf{Q}^{-1}\mathbf{t}
\end{aligned} \tag{60}
$$

SO(3)와 동일하게 연산의 편의를 위해 일반적으로 $\xi^{\wedge} \rightarrow \mathbf{T}$로 매핑하는 과정은 $\exp(\cdot)$ 연산자를 활용하며 $\xi \rightarrow \mathbf{T}$로 매핑하는 과정은 $\mathrm{Exp}(\cdot)$ 연산자를 활용한다. 로그 매핑 또한 동일하다.

$$
\begin{aligned}
\exp(\cdot) &: \xi^{\wedge} \mapsto \mathbf{T} \\
\mathrm{Exp}(\cdot) &: \xi \mapsto \mathbf{T}
\end{aligned} \tag{61}
$$

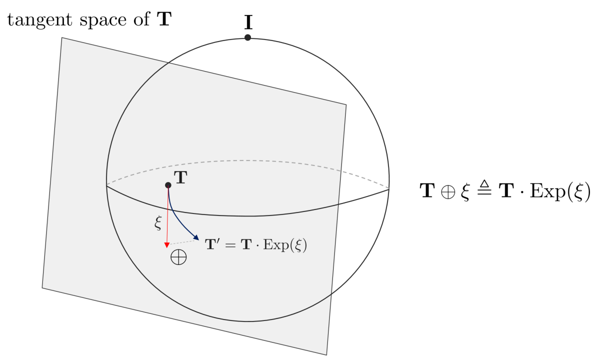
이를 그림으로 정리하면 다음과 같다.



4.4 Plus and Minus Operator of SE(3)

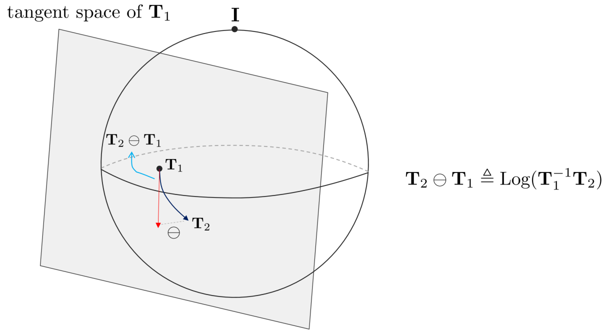
지금까지 설명한 exponential mapping을 사용하면 twist ξ를 사용하여 $\mathbf{T}$을 변환할 수 있다. SE(3)와 se(3) 사이에는 일반적인 $+, -$ 연산자가 적용되지 않으므로 새로운 $\oplus, \ominus$ 연산자를 정의해야 한다. 우선 $\oplus$ 연산자는 임의의 포즈 $\mathbf{T}$에 ξ 만큼 추가적인 변환을 적용하는 연산자이다.

$$
\mathbf{T} \oplus \xi \triangleq \mathbf{T} \cdot \mathrm{Exp}(\xi) \tag{62}
$$

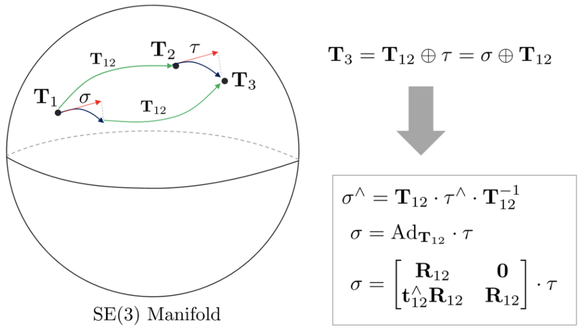


반대로 $\ominus$ 연산자는 두 포즈 $\mathbf{T}_1, \mathbf{T}_2$가 존재할 때 $\mathbf{T}_2 \ominus \mathbf{T}_1$과 같이 사용되며 $\mathbf{T}_1$에서 $\mathbf{T}_2$의 상대 포즈를 의미한다.

$$
\mathbf{T}_2 \ominus \mathbf{T}_1 \triangleq \mathrm{Log}(\mathbf{T}_1^{-1}\mathbf{T}_2) \tag{63}
$$



4.5 Adjoint Matrix of SE(3)



SE(3)군의 Adjoint 행렬은 임의의 $\mathbf{T}_2 \in SE(3)$의 접평면에 존재하는 임의의 twist $\tau \in \mathbb{R}^6$을 다른 $\mathbf{T}_1$의 접평면에 대응하는 twist σ 로 변환하는 행렬이다. $\mathbf{T}_{12} \in SE(3)$ 에 대하여 Adjoint 행렬을 $\text{Ad}_{\mathbf{T}_{12}}$ 라고 하면 다음이 성립한다.

$$\xi_1 = \text{Ad}_{\mathbf{T}_{12}} \xi_2 \tag{64}$$

Twist를 다른 twist로 변환하는 행렬이므로 $\text{Ad}_{\mathbf{T}_{12}} \in \mathbb{R}^{6\times6}$ 의 크기를 가진다. 또한 Adjoint 행렬은 다음과 같은 수식을 만족한다.

$$\begin{aligned} \mathbf{T}_{12} \cdot \exp(\tau) &= \exp((\text{Ad}_{\mathbf{T}_{12}} \cdot \tau)^\wedge) \cdot \mathbf{T}_{12} \\ \exp((\text{Ad}_{\mathbf{T}_{12}} \cdot \tau)^\wedge) &= \mathbf{T}_{12} \cdot \exp(\tau) \cdot \mathbf{T}_{12}^{-1} \end{aligned} \tag{65}$$

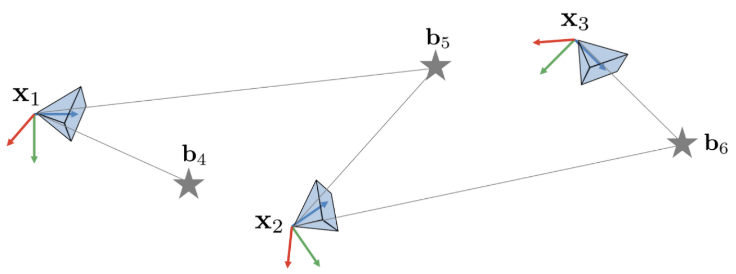
se(3) 대수에 대한 Adjoint 행렬의 유도 과정은 다음과 같다.

$$\begin{aligned} (\text{Ad}_{\mathbf{T}_{12}} \cdot \tau)^\wedge &= \mathbf{T}_{12} \cdot \left(\sum_{i=1}^{6} \xi_i G_i \right) \cdot \mathbf{T}_{12}^{-1} \\ &= \begin{pmatrix} \mathbf{R}_{12}\omega_{12} \\ \mathbf{t}_{12}^\wedge \mathbf{R}_{12}\omega_{12} + \mathbf{R}_{12}\mathbf{v}_{12} \end{pmatrix}^\wedge \\ &= \left(\begin{pmatrix} \mathbf{R}_{12} & \mathbf{0} \\ \mathbf{t}_{12}^\wedge \mathbf{R}_{12} & \mathbf{R}_{12} \end{pmatrix} \begin{pmatrix} \omega_{12} \\ \mathbf{v}_{12} \end{pmatrix} \right)^\wedge \end{aligned} \tag{66}$$

$$\text{Ad}_{\mathbf{T}_{12}} = \begin{pmatrix} \mathbf{R}_{12} & \mathbf{0} \\ \mathbf{t}_{12}^\wedge \mathbf{R}_{12} & \mathbf{R}_{12} \end{pmatrix} \in \mathbb{R}^{6\times6} \tag{67}$$

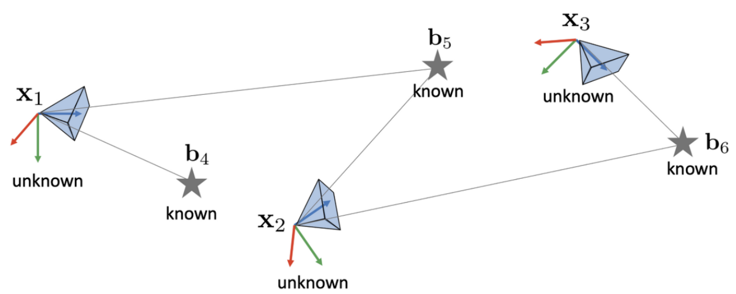
5 Applications for estimation

다음으로 Lie Group을 활용한 실제 상태추정 예시를 알아본다. 아래와 같이 3차원 공간 상에 카메라와 랜드마크들이 존재한다고 가정하자.



이 때, $\mathbf{x}_i, i = 1, 2, 3$은 카메라의 3차원 공간 상의 포즈를 의미하며 $\mathbf{b}_i, i = 4, 5, 6$는 랜드마크들의 좌표를 의미한다.

5.1 EKF map-based localzation



만약 $\mathbf{x}_i$은 모르는 상태에서 $\mathbf{b}_i$의 값만 주어졌다고 했을 때, 카메라는 랜드마크를 사용하여 자신의 포즈 $\mathbf{x}_i$를 EKF를 통해 추정할 수 있다.

$$\begin{aligned} \mathbf{x} \sim \mathcal{N}(\bar{\mathbf{x}}, \mathbf{P}) \in SE(3) &: \text{Unknown} \\ \mathbf{P} = \mathbb{E}[(\mathbf{x} \ominus \bar{\mathbf{x}})(\mathbf{x} \ominus \bar{\mathbf{x}})^\intercal] &: \text{Unknown} \\ \mathbf{b} \in \mathbb{R}^3 &: \text{Known} \end{aligned} \tag{68}$$

카메라의 motion model과 measurement model은 다음과 같다.

$$\begin{aligned} &\text{motion model:} && \mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{u}_i) = \mathbf{x}_{i-1} \oplus (\mathbf{u_i} dt + \omega) \\ &\text{measurement model:} && \mathbf{y}_k = h(\mathbf{x}) = \mathbf{x}^{-1}\mathbf{b}_k + v, \quad \text{where, } v \sim \mathcal{N}(0, \mathbf{R}) \end{aligned} \tag{69}$$

- $\omega \sim \mathcal{N}(0, \mathbf{Q})$: perturbation
- $v \sim \mathcal{N}(0, \mathbf{R})$: noise

이 때, 카메라의 포즈는 SE(3) 군에 속하므로 $\oplus$ 연산을 사용하여 포즈를 업데이트 한다. 위 두 모델을 활용한 EKF의 prediction, correction step은 다음과 같다

5.1.1 Prediction Step

$$\begin{aligned} \hat{\mathbf{x}} &\leftarrow \hat{\mathbf{x}} \oplus \mathbf{u}_i dt \\ \mathbf{P} &\leftarrow \mathbf{F}\mathbf{P}\mathbf{F}^\intercal + \mathbf{G}\mathbf{Q}\mathbf{G}^\intercal \end{aligned} \tag{70}$$

- $\mathbf{F} = \frac{Df}{D\mathbf{x}}$
- $\mathbf{G} = \frac{Df}{D\omega}$

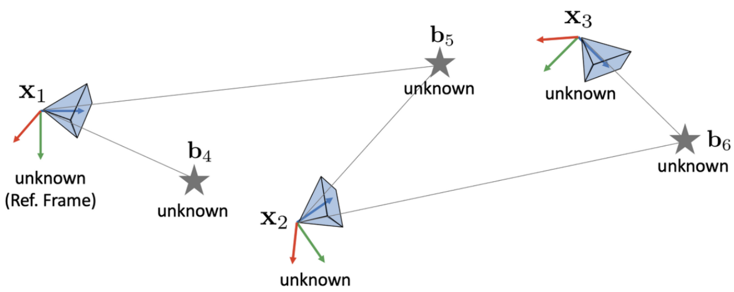
5.1.2 Correction Step

$$\begin{aligned} \mathbf{z}_k &= \mathbf{y}_k - \hat{\mathbf{x}}^{-1}\mathbf{b}_k \\ \mathbf{Z}_k &= \mathbf{H}\mathbf{P}\mathbf{H}^\intercal + \mathbf{R} \\ \mathbf{K} &= \mathbf{P}\mathbf{H}^\intercal\mathbf{Z}_k^{-1} \\ \hat{\mathbf{x}} &\leftarrow \hat{\mathbf{x}} \oplus \mathbf{K}\mathbf{z}_k \\ \mathbf{P} &\leftarrow \mathbf{P} - \mathbf{K}\mathbf{Z}_k\mathbf{K}^\intercal \end{aligned} \tag{71}$$

- $\mathbf{H} = \frac{Dh}{D\mathbf{x}}$

위 공식은 일반적인 EKF 공식과 동일하다. 따라서 Lie Group 연산자 $\oplus, \ominus, \frac{D*}{D*}, \mathbf{x}^{-1}$ 등을 사용하면 nonlinear한 $SE(3)$ 연산을 linear vector space에서 사용하는 연산과 동일하게 수행할 수 있다.

5.2 Pose Graph SLAM



만약 $\mathbf{x}_i$과 $\mathbf{b}_i$ 모두 모른다고 하면, pose graph SLAM을 통해 두 상태변수를 추정할 수 있다.

$$\begin{aligned} \mathbf{x} \in SE(3) &: \text{Unknown} \\ \mathbf{b} \in \mathbb{R}^3 &: \text{Unknown} \end{aligned} \tag{72}$$

이 때, 상태변수는 다음과 같이 벡터 형태로 한 번에 묶어서 표현할 수 있다.

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{b}_4, \mathbf{b}_5, \mathbf{b}_6) \tag{73}$$

이 때 비선형 최적화 문제는 다음과 같이 정의할 수 있다.

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_p \|\mathbf{r}_p(\mathbf{x})\|^2 \tag{74}$$

이 때, residual $\mathbf{r}$은 다음과 같이 정의할 수 있다.

$$\begin{aligned} \text{prior: } \mathbf{r}_1 &= \Omega_1^{\intercal/2}(\mathbf{x}_1 \ominus \mathbf{x}_1^{ref}) \\ \text{motion: } \mathbf{r}_{ij} &= \Omega_{ij}^{\intercal/2}(\mathbf{u}_j dt - (\mathbf{x}_j \ominus \mathbf{x}_i)) \\ \text{measurement: } \mathbf{r}_{ik} &= \Omega_{ik}^{\intercal/2}(\mathbf{y}_{ik} - \mathbf{x}_i^{-1}\mathbf{b}_k) \end{aligned} \tag{75}$$

그리고 모든 카메라 포즈와 랜드마크에 대한 residual과 자코비안을 다음과 같이 정의할 수 있다.

$$\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_{12} & \mathbf{r}_{23} & \mathbf{r}_{14} & \mathbf{r}_{15} & \mathbf{r}_{25} & \mathbf{r}_{26} & \mathbf{r}_{36} \end{bmatrix}^\intercal \tag{76}$$

$$\mathbf{J} = \begin{array}{cc} \begin{array}{cccccc} \quad\mathbf{x}_1 & \quad\mathbf{x}_2 & \quad\mathbf{x}_3 & \quad\mathbf{b}_4 & \quad\mathbf{b}_5 & \quad\mathbf{b}_6 \end{array} & \\ \begin{bmatrix} \mathbf{J}^{\mathbf{r}_1}_{\mathbf{x}_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}^{\mathbf{r}_{12}}_{\mathbf{x}_1} & \mathbf{J}^{\mathbf{r}_{12}}_{\mathbf{x}_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{\mathbf{r}_{23}}_{\mathbf{x}_2} & \mathbf{J}^{\mathbf{r}_{23}}_{\mathbf{x}_3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}^{\mathbf{r}_{14}}_{\mathbf{x}_1} & \mathbf{0} & \mathbf{0} & \mathbf{J}^{\mathbf{r}_{14}}_{\mathbf{b}_4} & \mathbf{0} & \mathbf{0} \\ \mathbf{J}^{\mathbf{r}_{15}}_{\mathbf{x}_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}^{\mathbf{r}_{15}}_{\mathbf{b}_5} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{\mathbf{r}_{25}}_{\mathbf{x}_2} & \mathbf{0} & \mathbf{0} & \mathbf{J}^{\mathbf{r}_{25}}_{\mathbf{b}_5} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{\mathbf{r}_{26}}_{\mathbf{x}_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{J}^{\mathbf{r}_{26}}_{\mathbf{b}_6} \\ \mathbf{0} & \mathbf{0} & \mathbf{J}^{\mathbf{r}_{36}}_{\mathbf{x}_3} & \mathbf{0} & \mathbf{0} & \mathbf{J}^{\mathbf{r}_{36}}_{\mathbf{b}_6} \end{bmatrix} & \begin{array}{c} \mathbf{r}_1 \\ \mathbf{r}_{12} \\ \mathbf{r}_{23} \\ \mathbf{r}_{14} \\ \mathbf{r}_{15} \\ \mathbf{r}_{25} \\ \mathbf{r}_{26} \\ \mathbf{r}_{36} \end{array} \end{array}$$

위 자코비안을 활용하여 앞서 정의한 비선형 최적화 문제를 Newton step을 통해 iterative하게 풀 수 있다.

$$\begin{aligned} \Delta\mathbf{x} &= -(\mathbf{J}^\intercal\mathbf{J})^{-1}\mathbf{J}^\intercal\mathbf{r} \\ \mathbf{x} &\leftarrow \mathbf{x} \oplus \Delta\mathbf{x} \end{aligned} \tag{77}$$

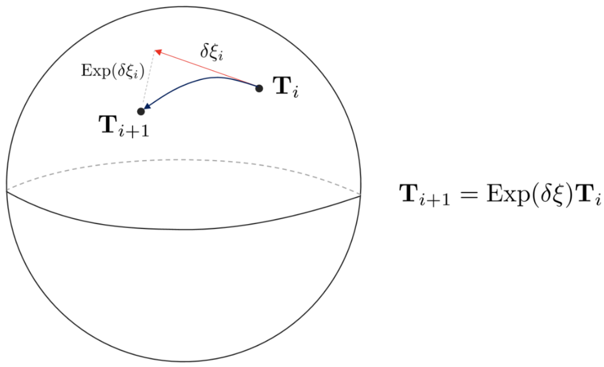
이 때 사용하는 최적화 공식 또한 Lie Group 연산자 $\oplus, \ominus, \frac{D*}{D*}, \mathbf{x}^{-1}$ 등을 통해 nonlinear한 $SE(3)$ 연산을 linear vector space에서 사용하는 연산과 동일하게 수행할 수 있다.

6 Lie theory-based optimization on SLAM

3차원 공간에서 카메라 포즈를 최적화 할 때 lie group(SO(3), SE(3)) 표현법을 그대로 사용하게 되면 회전 표현법이 over-parameterized 되었기 때문에 다양한 문제점이 발생한다. over-parameterized 표현법의 단점은 다음과 같다.

- 중복되는 파라미터를 계산해야 하기 때문에 최적화 수행 시 연산량이 증가한다.
- 추가적인 자유도로 인해 수치적인 불안정성(numerical instability) 문제가 야기될 수 있다.
- 파라미터가 업데이트될 때마다 항상 제약조건을 만족하는 지 체크해줘야 한다.

반면에, lie algebra(so(3), se(3))을 사용하면 비선형 최적화(e.g., GN, LM) 방법을 통해 lie algebra의 증분량을 구하고 이를 지수 매핑 하여 lie group(SO(3), SE(3)) 공간으로 변환하면 제약조건 없는 최적화가 가능하다. 따라서 lie theory를 사용하면 기존의 constrained optimization 문제를 unconstrained opotimization 문제로 변환 하여 푸는 것과 동일한 이점이 존재한다. 또한 lie algebra는 선형 벡터공간(linear vector space)이므로 자코비안과 섭동(perturbation)을 계산하기 비교적 용이하여 기존의 최적화 기법을 적용하기 위한 모델링이 그대로 사용이 가능한 이점이 있다.



SE(3) Manifold

se(3) 기반 최적화를 예로 들면, SLAM에서 최적화 변수를 $\delta\xi = [\delta\omega, \delta\mathbf{v}]^\intercal$로 설정하여 비선형 최적화(e.g., GN, LM)을 수행하면 매 iteration마다 계산되는 증분량을 $\mathrm{Exp}(\delta\xi)$를 통해 SE(3) 군으로 변환할 수 있다. 이를 기존의 카메라 포즈 $\mathbf{T}$에 곱하여 포즈를 업데이트하면 별도의 제약조건 없이 업데이트를 진행할 수 있다.

$$\mathbf{T} \leftarrow \mathrm{Exp}(\delta\xi) \cdot \mathbf{T} \tag{78}$$

Tip

기존 상태 $\mathbf{T}$의 오른쪽에 곱하느냐 왼쪽에 곱하느냐에 따라서 각각 로컬 좌표계에서 본 포즈를 업데이트할 것 인지(오른쪽) 전역 좌표계에서 본 포즈를 업데이트할 것 인지(왼쪽) 달라지게 된다.

$$\begin{aligned} \mathbf{T} &\leftarrow \mathrm{Exp}(\delta\xi) \cdot \mathbf{T} \quad \cdots \text{ globally updated (left mult)} \\ \mathbf{T} &\leftarrow \mathbf{T} \cdot \mathrm{Exp}(\delta\xi) \quad \cdots \text{ locally updated (right mult)} \end{aligned} \tag{79}$$

7 Reference

1. (Paper) Lie Groups for 2D and 3D Transformations
2. (Paper) A tutorial on SE3 transformation parameterizations and on-manifold optimization
3. (Book) 입문 Visual SLAM
4. (Youtube) Modern Robotics Lecture
5. (Youtube) Lie theory for the roboticist (by Joan Sola)

8 Revision log

- 1st: 2022-01-04
- 2nd: 2022-08-14
- 3rd: 2022-09-07
- 4th: 2022-11-26
- 5th: 2023-01-21
- 6th: 2023-01-23
- 7th: 2023-01-25
- 8th: 2023-01-28
- 9th: 2023-11-14
- 10th: 2024-02-24