3D Sensing and Sensor Fusion
http://cg.elte.hu/~sensing

Iván Eichhardt
eiiraai@inf.elte.hu

Eötvös Loránd University
Faculty of Informatics

2023 fall (2023-2024-1)

Introduction

Introduction

Introduction

Introduction

Introduction

Motivation

The study of the **representation of motion** is relevant:

- 3D rotation using $\mathbb{R}^{3\times 3} \longleftrightarrow$ has only 3 DoF. Why?
- What is the (continuous) *manifold* of motion?
- Articulated robots.
- Autonomous vehicles.
- Sensors, uncertainity propagation, Kalman filtering.
- Optimisation.
- *etc.*

Reminders

Reminders:

- Vector spaces
- Linear independence, Basis, Inner product, Dot product, Properties
- Linear transformations, Matrices
- Range, Span, Null space/Kernel, Rank
- Eigenvalues and eigenvectors, properties
- Symmetric matrices, positive (semi-)definite
- Skew-symmetric matrices $\mathbf{A}^\mathsf{T} = -\mathbf{A}$

Groups

A **group** is a set G with and operation $\circ : G \times G \rightarrow G$ for which the following properties hold.:

- $\forall g_1, g_2 \in G : g_1 \circ g_2 \in G$ (**closure**)
- $\forall g_1, g_2, g_3 \in G : (g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$ (**associativity**)
- $\exists! e \in G \; \forall g \in G$: $e \circ g = g \circ e = g$ (**identity element**)
- $\forall g \in G \; \exists g^{-1} \in G$: $g^{-1} \circ g = g \circ g^{-1} = e$ (**inverse element**)

General and Special Linear groups

- general linear group $GL(n) = \{\mathbf{A} \in \mathbb{R}^{n \times n} : \det(\mathbf{A}) \neq 0\}$
  $GL(n)$ is a group *w.r.t.* matrix multiplication
- special linear group $SL(n) = \{\mathbf{A} \in \mathbb{R}^{n \times n} : \det(\mathbf{A}) = 1\}$
  Note: if $\mathbf{A} \in SL(n)$, then $\mathbf{A}^{-1} \in SL(n)$

Matrix representation of groups

Think...

- How to represent complex numbers $\mathbb{C}$ using real matrices?
- ... and dual numbers $\mathbb{D}$?

Group homomorphism is an injective map, preserving composition:

- $R : G \to GL(n)$ is a group homomorphism, if
- if $R(e) = I_{n \times n}$ and $\forall f, g \in G$: $R(f \circ g) = R(f)R(g)$.

The Affine group $A(n)$

- Reminder: affine transformations, homogeneous coordinates
- for $\mathbf{A} \in GL(n)$, $b \in \mathbb{R}^n$, then $\begin{bmatrix} A & b \\ 0 & 1 \end{bmatrix} \in GL(n+1)$ is an affine matrix. Affine matrices form a subgroup in $GL(n+1)$

The Orthogonal group

The **Orthogonal group**:

$$O(n) = \left\{ \mathbf{R} \in GL(n) : \mathbf{R}^\top \mathbf{R} = \mathbf{I}_{n \times n} \right\}$$

Special Orthogonal group

Removing mirroring, by adding the constraint $\det(\mathbf{R}) = 1$:

$$SO(n) = O(n) \cap SL(n)$$

Note: $SO(3)$ is the group of all 3D *rotation matrices*.

The Euclidean group

The **Euclidean group**:

$$E(n) = \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} : \mathbf{R} \in O(n), \mathbf{t} \in \mathbb{R}^n \right\} \subset GL(n+1)$$

The Special Euclidean group SE(n)

$$SE(n) = \left\{ \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} : \mathbf{R} \in SO(n), \mathbf{t} \in \mathbb{R}^n \right\} \subset GL(n+1)$$

Note: $SE(3)$ is the group of rigid body motions in $\mathbb{R}^3$.

Outline

Motivation (1/2): Skew-symmetric matrices & cross product

The **cross product** can be defined between two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$: $\mathbf{u} \times \mathbf{v} \in \mathbb{R}^3$, furthermore

$$\mathbf{u} \times \mathbf{v} = \widehat{\mathbf{u}}\mathbf{v},$$

where $\widehat{\mathbf{u}}$ is a skew-symmetric matrix

$$\widehat{\mathbf{u}} = [\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

The unary operator $\widehat{\cdot}$ is an isomorphism, between $\mathbb{R}^3$ and $so(3) \subset \mathbb{R}^{3\times 3}$, the *set of all skew-symmetric matrices*.

Note that $\mathbf{A} \in so(n)$ *iff* $\mathbf{A} = -\mathbf{A}^T$.

Motivation (2/2): Infinitesimal rotation

Remark

Skew-symmetric matrices $so(n) = \{\widehat{\mathbf{w}} : \mathbf{w} \in \mathbb{R}^n\} \subset \mathbb{R}^{n\times n}$ form the tangent space to the orthogonal group $O(n)$, at $\mathbf{I}_{n\times n}$. In that sense, $so(n)$ can be thought of as *infinitesimal rotations*.

Let $R(t) \in \mathbb{R} \to SO(3)$, $R(0) = \mathbf{I}_{3\times 3}$ be a continuously differentiable family of rotation matrices. Let $\dot{R}$ denote $\frac{d}{dt}R(t)$. As $R(t)R(t)^{\mathsf{T}} = \mathbf{I}_{3\times 3}$ for all t, differentiating *w.r.t.* t gives:

$$\dot{R}R^{\mathsf{T}} + R\dot{R}^{\mathsf{T}} = 0$$

This implies that $\dot{R}R^{\mathsf{T}}$ is *skew-symmetric*, and that $\exists w \in \mathbb{R} \to so(3)$, for which $\widehat{w}(t) = \dot{R}(t)R(t)^{\mathsf{T}}$. Therefore, the first-order approximation of R at $t = 0$ is $\widehat{w}(0) \in so(3)$:

$$R(0+\delta) \approx \mathbf{I}_{3\times 3} + \widehat{w}(0)\delta.$$

Lie group, Lie algebra

Remark

The group $SO(3)$ is a *Lie group*, while the space $so(3)$ is its corresponding *Lie algebra*. The latter is the tangent space at the identity of $SO(3)$.

A **Lie group** is simultaneously a group and a smooth differentiable manifold, with smooth product (and inverse) operation.

A **Lie algebra** V is a vector space over a field K, with the operation $[.,.] : V \times V \to V$ (the so-called commutator- or Lie-bracket).

The Lie group is a complicated nonlinear object, while its Lie algebra is just a vector space: it is usually simpler to work with.

Maps for a Lie group

Assume a Lie group (*manifold*) G and the corresponding Lie algebra (*local tangent space*) g.

Exponential map

exp: A map from the tangent space g to the manifold G.

$$\exp : g \to G$$

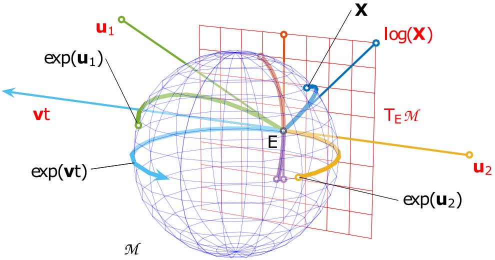
Logarithmic map

log: Inverse map, from the manifold to the tangent space.

$$\log : G \to g$$

We'll further investigate these concepts for specific Lie groups.

The relation of Lie group and Lie algebra



The Lie algebra $T_{\mathbf{E}}\mathcal{M}$ (red plane) to the Lie group's manifold $\mathcal{M}$ (blue sphere) at the identity (here denoted as $\mathbf{E}$)[1].

Each element in $T_{\mathbf{E}}\mathcal{M}$ has an equivalent on $\mathcal{M}$: e.g., $\mathbf{v}t$ produces path $\exp(\mathbf{v}t)$, and $\log(\mathbf{X})$ corresponds to $\mathbf{X}$. Notice the geodesics.

[1] Solà *et al.*– A micro Lie theory for state estimation in robotics

Remark: The use of Lie algebras

Sophus Lie (1841 - 1899) originally formulated the related concepts while creating the theory of continuous symmetry and applied it to the geometric problems and differential equations.

Today, Lie algebras have numerous applications in the fields of mathematics, physics, and among else, even computer/robot vision and control. A few applications in vision:

- interpolation,
- (on-manifold) optimisation,
- tracking,
- statistics,
- etc.

Outline

Group action

Lie groups have the power to *transform* elements of other sets (*e.g.*, rotation, translation, scaling of vectors *etc.*).

Let G be a Lie group, and $\mathcal{V}$ some set. The **group action** is a mapping

$$\cdot\,:\,G \times \mathcal{V} \rightarrow \mathcal{V}.$$

The group action must satisfy the following *axioms*:

$$\text{Identity:}\quad \mathbf{I} \cdot \mathbf{v} = \mathbf{v}$$

$$\text{Compatibility:}\quad (\mathbf{X} \circ \mathbf{Y}) \cdot \mathbf{v} = \mathbf{X} \cdot (\mathbf{Y} \cdot \mathbf{v})$$

Group action: Examples

On $SO(n)$ rotation of a vector. Let $\mathbf{R} \in SO(n)$, $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{R} \cdot \mathbf{x} \doteq \mathbf{R}\mathbf{x}.$$

Rigid motion of a point. Let $\mathbf{H} \in SE(n)$, $\mathbf{x} \in \mathbb{R}^n$:

$$\mathbf{H} \cdot \mathbf{x} \doteq \mathbf{R}\mathbf{x} + \mathbf{t}.$$

On S^3 rotation of a vector. Let $\mathbf{q}$ be a unit quaternion, $\mathbf{x} \in \mathbb{R}^3$:

$$\mathbf{q} \cdot \mathbf{x} \doteq \mathbf{q}\mathbf{x}\mathbf{q}^*.$$

Notation: Capital Exp and Log maps

The parameters of the exp map and the result of the log are in the Lie algebra. However, there's usually a compact representation, *e.g.*, for skew-symmetric matrices.

Assume a Lie group G and the corresponding Lie algebra g. The compact representation of elements of g is in $\mathbb{R}^m$:

$$\text{if } \widehat{\mathbf{u}} \in g \text{ then } \mathbf{u} \in \mathbb{R}^m.$$

Capital Exp **and** Log **maps** consider $\mathbb{R}^m$:

$$\text{Exp} : \mathbb{R}^m \to G, \text{ so that } \text{Exp}(\mathbf{u}) \doteq \exp(\widehat{\mathbf{u}}),$$

$$\text{Log} : G \to \mathbb{R}^m, \text{ so that } \widehat{\text{Log}(\mathbf{X})} = \text{Log}(\mathbf{X})^{\wedge} \doteq \log(\mathbf{X}).$$

Plus and minus operators

Nonlinear mapping operators, **boxplus** and **boxminus** can express addition and subtraction for $\mathbf{X}, \mathbf{Y} \in \mathcal{M}$ and $\widehat{\mathbf{u}} \in \mathsf{T}_{\mathbf{E}}\mathcal{M}$:

$$\mathbf{X} \boxplus \mathbf{u} \doteq \mathrm{Exp}(\mathbf{u}) \circ \mathbf{X}$$
$$\mathbf{X} \boxminus \mathbf{Y} \doteq \mathrm{Log}(\mathbf{X} \circ \mathbf{Y}^{-1})$$

Note that $(\mathbf{X} \boxminus \mathbf{Y})^{\wedge} \in \mathsf{T}_{\mathbf{E}}\mathcal{M}$, *i.e.*, in the tangential space at the identity $\mathbf{E}$ – in the *global frame*.

Also note that some works[2] use *local frames*, *i.e.*, defining the (right) minus operator as $\mathrm{Log}(\mathbf{X}^{-1} \circ \mathbf{Y})^{\wedge} \in \mathsf{T}_{\mathbf{X}}\mathcal{M}$.

[2] *E.g.*, J. Sola *et al.*– A micro Lie theory for state estimation in robotics

Further topics to explore

- The adjoint matrix
- Derivatives
- Uncertainity / covariance propagation
- Velocity

Outline

The Exponential map

Assume the following differential equation system, where $\widehat{\mathbf{w}}$ is constant in time:

$$\dot{R}(t) = \widehat{\mathbf{w}} R(t),$$
$$R(0) = \mathbf{I}_{3\times 3}.$$

Its solution is

$$R(t) = e^{\widehat{\mathbf{w}} t} = \sum_{n=0}^{\infty} \frac{1}{n!} \left(\widehat{\mathbf{w}} t\right)^n = \mathbf{I}_{3\times 3} + \widehat{\mathbf{w}} t + \frac{1}{2} \left(\widehat{\mathbf{w}} t\right)^2 + \ldots,$$

that is, a rotation around axis $\mathbf{w} \in \mathbb{R}^3$ by an angle t, given $\|\mathbf{w}\| = 1$. Alternatively, embed t into $\mathbf{w}$ by setting $\|\mathbf{w}\| = t$.

The *matrix exponential* defines a Lie algebra to Lie group mapping:

$$\exp : so(3) \to SO(3), \exp(\widehat{\mathbf{w}}) = e^{\widehat{\mathbf{w}}}.$$

Rodrigues' formula

Analogous to the Euler equation $e^{i\phi} = \cos\phi + i\sin(\phi), \forall\phi \in \mathbb{R}$, we can use the **Rodrigues' formula** for the elements of $so(3)$:

$$e^{\widehat{\mathbf{w}}} = \mathbf{I}_{3\times 3} + \frac{\widehat{\mathbf{w}}}{|\mathbf{w}|}\sin(|\mathbf{w}|) + \frac{\widehat{\mathbf{w}}^2}{|\mathbf{w}|^2}(1 - \cos(|\mathbf{w}|))\,.$$

The Logarithmic map

An inverse function to the exponential map can also be defined, that is, the logarithm.

For all $\mathbf{R} \in SO(3)$: $\exists \mathbf{w} \in \mathbb{R}^3$ such that $\mathbf{R} = \exp(\widehat{\mathbf{w}})$. Let us denote this element by $\widehat{\mathbf{w}} = \log \mathbf{R}$. If $\mathbf{R} \neq \mathbf{I}_{3\times 3}$, then

$$|\mathbf{w}| = \cos^{-1}\left(\frac{\mathrm{tr}(\mathbf{R}) - 1}{2}\right),$$

$$\frac{\mathbf{w}}{|\mathbf{w}|} = \frac{1}{2\sin(|\mathbf{w}|)} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}.$$

Note that for $\mathbf{R} = \mathbf{I}_{3\times 3}$, $|\mathbf{w}| = 0$. Also note that this representation is periodic *w.r.t.* the angle, by multiples of 2π, *i.e.*, not unique.

Rotations in 2D: $SO(2)$

The Lie algebra $so(2)$ corresponding to $SO(2)$ is generated by a single skew-symmetric matrix:

$$\exp\left(\phi \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\right) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}$$

Outline

Other representations of rotation: Lie-Cartan coordinates

Lie-Cartan coordinates of the first kind

Given a basis $\widehat{\mathbf{w}}_1, \widehat{\mathbf{w}}_2, \widehat{\mathbf{w}}_3 \in so(3)$ we can define the mapping

$$\alpha : (\alpha_1, \alpha_2, \alpha_3) \to \exp\left(\alpha_1 \widehat{\mathbf{w}}_1 + \alpha_2 \widehat{\mathbf{w}}_2 + \alpha_3 \widehat{\mathbf{w}}_3\right).$$

$(\alpha_1, \alpha_2, \alpha_3)$ are the **Lie-Cartan coordinates of the first kind** relative to the above basis.

Lie-Cartan coordinates of the second kind

$$\beta : (\beta_1, \beta_2, \beta_3) \to \exp\left(\beta_1 \widehat{\mathbf{w}}_1\right) \exp\left(\beta_2 \widehat{\mathbf{w}}_2\right) \exp\left(\beta_3 \widehat{\mathbf{w}}_3\right),$$

where $\mathbf{w}_1 = (0,0,1)^\mathsf{T}$, $\mathbf{w}_2 = (0,1,0)^\mathsf{T}$, and $\mathbf{w}_3 = (1,0,0)^\mathsf{T}$.
$(\beta_1, \beta_2, \beta_3)$ are **Euler angles**, rotations around the x, y, z axes.

The parameterizations are only correct for a portion of $SO(3)$!

Other representations of rotation: Unit quaternions

Compared to rotation matrices, they are more compact, numerically more stable, and more efficient.

Unit quaternions

Given the angle ϕ of rotation around the unit axis (x, y, z) can be represented as:

$$\mathbf{q} = \left[\cos(\phi/2), \quad \sin(\phi/2)x, \quad \sin(\phi/2)y, \quad \sin(\phi/2)z\right] \in \mathbb{Q}$$

Comparing some representations of rotation

Method	mul	add/sub	total ops.
Matrices	27	18	45
Quaternions	16	12	28

Performance of rotation chaining.

Method	mul	add/sub	sin/cos	total ops.
Matrices	9	6	0	15
Quaternions	15	15	0	30
Euler angles	18	12	2	30+2

Performance of vector rotation.

Note that one may convert to matrix representation to leverage the cost of vector rotation.

Outline

Estimating transformation between point sets (1/2)

Given $x_i, y_i \in \mathbb{R}^3$ $(i \in \{1 \dots n\})$, find $\mathbf{R} \in SO(3)$, $\mathbf{t} \in \mathbb{R}^3$, $c \in \mathbb{R}$:

$$\min_{c,\mathbf{R},\mathbf{t}} \frac{1}{n} \sum_{i=1}^{n} ||y_i - (c\mathbf{R}x_i + \mathbf{t})||_2^2$$

Multiple approaches exist:
SVD, Dual Quaternion, Unit Quaternion, Orthogonal matrices, ...

SVD:

- Umeyama's LSq algorithm[3]
- *E.g.*: Eigen library (C++): `Eigen::umeyama()`

[3] Umeyama, S. *Least-squares estimation of transformation parameters between two point patterns*. (1991) IEEE TPAMI, (4), 376-380. umeyama.pdf

Estimating transformation between point sets (2/2)

Refined estimate can be achieved, if needed:

1. First, estimate rotation using *e.g.* Umeyama's method.
2. Then, perform non-linear refinement.

Notes on non-linear refinement:

- Possible parameterizations:
  Unit quaternions, Euler angles, $SO(3)+\mathbb{R}^3$, $SE(3)$ or $Sim(3)$.
- Approach:
  1. Perform refinement using corresp. Lie algebra.
  2. Update transformation using the boxplus operator.
- More robust cost functions can also be applied.

Outline

Lie algebra $se(3)$ and the twist

As we did for rotations, we can define a continuous family of rigid body motions $g(t) : \mathbb{R} \to SE(3)$.

$$g(t) = \begin{bmatrix} R(t) & T(t) \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4}$$

Considering $\widehat{\xi}(t) = \dot{g}(t)g^{-1}(t)$, we have

$$\widehat{\xi} = \begin{bmatrix} \dot{R}R^{\mathsf{T}} & \dot{T} - \dot{R}R^{\mathsf{T}}T \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{w}} & \mathbf{v} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{4\times 4},$$

where $\mathbf{v} = \dot{T} - \widehat{\mathbf{w}}T$.
Thus, $\dot{g} = \dot{g}g^{-1}g = \widehat{\xi}g$: the matrix $\widehat{\xi}$ can be viewed as a tangent vector to curve g, a so-called **twist**.

Lie algebra $se(3)$ and the twist vector

The set of all twists forms a tangent space to the Lie group $SE(3)$, the Lie algebra $se(3)$ is defined as follows:

$$se(3) = \left\{ \begin{bmatrix} \widehat{\mathbf{w}} & \mathbf{v} \\ 0 & 0 \end{bmatrix} : \widehat{\mathbf{w}} \in so(3), \mathbf{v} \in \mathbb{R}^3 \right\}$$

The **twist vector** $\xi \in \mathbb{R}^6$ corresponds to the twist $\widehat{\xi} \in se(3)$:

$$\xi = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \longleftrightarrow \begin{bmatrix} \widehat{\mathbf{w}} & \mathbf{v} \\ 0 & 0 \end{bmatrix} = \widehat{\xi}$$

Exponential and Logarithmic maps for $SE(3)$

Let $\xi = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \in \mathbb{R}^6$ be the vector tangent space element corresponding to $\mathbf{M} \in SE(3)$:

$$\mathbf{M} = \mathrm{Exp}(\xi) \doteq \begin{bmatrix} \mathrm{Exp}(\mathbf{w}) & \mathbf{V}(\mathbf{w})\mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix},$$

$$\xi = \mathrm{Log}(\mathbf{M}) \doteq \begin{bmatrix} \mathbf{V}^{-1}(\mathbf{w})\mathbf{T} \\ \mathrm{Log}(\mathbf{R}) \end{bmatrix},$$

where

$$\mathbf{V}(\mathbf{w}) = \mathbf{V}(\theta\mathbf{u}) = \mathbf{I}_{3\times 3} + \frac{1-\cos\theta}{\theta}\widehat{\mathbf{u}} + \frac{\theta - \sin\theta}{\theta}\widehat{\mathbf{u}}^2.$$

Outline

Representation of camera motion

Let's consider and element of $\mathbf{P} \in SE(3)$ to represent camera motion.

- Often called the **camera pose**.
- By convention, a *world* frame to *local* frame transformation.

Assume a camera projection function $p : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ – a mapping from *local* frame to 2D *image space*.

To map *world* frame point $\mathbf{X} \in \mathbb{R}^3$ to *image space*:

$$\mathbf{x} = p(\mathbf{P} \cdot \mathbf{X}) \in \mathbb{R}^2.$$

Outline

Interpolation on the Manifold

Let G be a Lie group[4], with respective log and exp maps to the respective Lie algebra and back. Given two elements $\mathbf{X}, \mathbf{Y} \in G$ (e.g. elements $SO(3)$) and a coefficient $t \in [0, 1]$, we can define **interpolation** as follows:

$$\exp\left(t \cdot \log\left(\mathbf{Y} \cdot \mathbf{X}^{-1}\right)\right) \cdot \mathbf{X} = X \boxplus [t \cdot (\mathbf{Y} \boxminus \mathbf{X})] .$$

Note that the interpolation always moves along the '*shortest*' transformation in the Lie group (i.e., it is moving along a *geodesic* of the manifold).

[4]Remember, a Lie group is also a smooth manifold.

Averaging on Manifolds

Averaging in Euclidean spaces works fine, using the usual definition

$$\bar{\mathbf{q}} = \arg\min_{\mathbf{p}} \sum_{i=1}^{N} \|\mathbf{p} - \mathbf{q}_i\|_2^2 = \frac{1}{N} \sum_{i=1}^{N} \mathbf{q}_i,$$

however, not in non-linear manifolds.
Given a *metric* $d(\mathbf{x}, \mathbf{y})$, the average can be defined as

$$\bar{\mathbf{p}} = \arg\min_{\mathbf{p}} \sum_{i=1}^{N} d(\mathbf{p}, \mathbf{q}_i)^2$$

E.g. the length of the shortest geodesic:

$$d_R(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} \boxminus \mathbf{B}\|_2 = \frac{1}{\sqrt{2}} \left\|\log\left(\mathbf{A}^{-1}\mathbf{B}\right)\right\|_F$$

Outline

Uncertain transformations: Sampling 3D rotations

To encode Gaussian distribution, choose a mean $\mathbf{R} \in SO(3)$ and a covariance matrix $\mathbf{\Sigma} \in so(3)$.

Now *draw a sample* $\mathbf{S}$ using the mean-covariance pair $(\mathbf{R}, \mathbf{\Sigma})$:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}),$$
$$\mathbf{S} = \mathbf{R} \boxplus \mathbf{w} = \text{Exp}(\mathbf{w}) \cdot \mathbf{R}.$$

Uncertain transformations: Composition

Given two mean-covariance pairs $(\mathbf{R}_0, \mathbf{\Sigma}_0)$ and $(\mathbf{R}_1, \mathbf{\Sigma}_1)$, the *composition*, *i.e.* distribution of rotations by first transforming by $\mathbf{R}_0$ and then by $\mathbf{R}_1$ is given by:

$$(\mathbf{R}_1, \mathbf{\Sigma}_1) \circ (\mathbf{R}_0, \mathbf{\Sigma}_0) = (\mathbf{R}_1 \cdot \mathbf{R}_0, \mathbf{\Sigma}_1 + \mathbf{R}_1 \cdot \mathbf{\Sigma}_0 \cdot \mathbf{R}_1^\top).$$

Uncertain transformations: Bayesian combination

The *Bayesian combination* of $(\mathbf{R}_0, \mathbf{\Sigma}_0)$ and $(\mathbf{R}_1, \mathbf{\Sigma}_1)$ is $(\mathbf{R}_c, \mathbf{\Sigma}_c)$:

1. Find the deviation between the two means in the tangent space.
2. Weight by the information of the two estimates.

$$\mathbf{\Sigma}_c = \left(\mathbf{\Sigma}_0^{-1} + \mathbf{\Sigma}_1^{-1}\right)^{-1},$$

$$\mathbf{R}_c = \mathbf{R}_0 \boxplus \left(\mathbf{\Sigma}_c \cdot \mathbf{\Sigma}_1^{-1} \cdot (\mathbf{R}_1 \boxminus \mathbf{R}_0)\right).$$

Extended Kalman filtering (EKF) in SO(3)

Let $\mathbf{R}_0$ and $\mathbf{\Sigma}_0$ be the prior state and state covariance. Assuming a trivial measurement Jacobian (identity matrix), a tangent vector $\mathbf{v}$ is the *innovation*.

$$\begin{aligned}\text{Kalman gain: } & \mathbf{K} \doteq \mathbf{\Sigma}_0(\mathbf{\Sigma}_0 + \mathbf{\Sigma}_1)^{-1}, \\ \text{Kalman update (cov.): } & \mathbf{\Sigma}_c = (\mathbf{I}_{3\times 3} - \mathbf{K}) \cdot \mathbf{\Sigma}_0, \\ \text{Kalman update (mean): } & \mathbf{R}_c = \mathbf{R}_0 \boxplus (\mathbf{K} \cdot \mathbf{v}).\end{aligned}$$

Note that mathematical identity to Bayesian combination can be proven, considering $\mathbf{v}{=}\mathbf{R}_1 \boxminus \mathbf{R}_0$ is the *innovation*, *i.e.*, the *measurement update*.

Differentiating rotation (in SO(3))

1) Consider $\widehat{\mathbf{w}} \in so(3)$ skew-symmetric matrix. (Remember, $\mathbf{w} \in \mathbb{R}^3$.)

$$\frac{\partial \widehat{\mathbf{w}}}{\partial \mathbf{w}} = \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right).$$

2) Since $\text{Exp}(\mathbf{w}) = \mathbf{I}_{3\times 3} + \widehat{\mathbf{w}} + \mathcal{O}(\widehat{\mathbf{w}}^2)$,

$$\frac{\partial}{\partial \mathbf{w}} \text{Exp}(\mathbf{w}) = \frac{\partial \widehat{\mathbf{w}}}{\partial \mathbf{w}}.$$

3) Let $\mathbf{R} \in SO(3)$. Analogous to random variables, perturbations of group elements are expressed in the local tangential space.

$$\frac{\partial \mathbf{R}}{\partial \mathbf{R}} = \left.\frac{\partial}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{0}} (\mathbf{R} \boxplus \mathbf{w}) = \left.\frac{\partial}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{0}} \text{Exp}(\mathbf{w})\mathbf{R}.$$

Differentiating the group action (in SO(3))

Let $\mathbf{y} = \mathbf{R} \cdot \mathbf{x}$, where $\mathbf{R} \in SO(3)$ and $\cdot : SO(3) \times \mathbb{R}^3 \to \mathbb{R}^3$ is the group action (*i.e.*, matrix-vector multiplication).
Differentiating by the vector to be rotated $\mathbf{x}$ yields:

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{R}.$$

To differentiate by $\mathbf{R}$, first perturb $\mathbf{R}$ locally by $\widehat{\mathbf{w}} \in so(3)$, the diff. by $\mathbf{w}$ around $\mathbf{w} = \mathbf{0}$ (around the zero perturbation):

$$\frac{\partial \mathbf{y}}{\partial \mathbf{R}} = \left.\frac{\partial}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{0}} (\mathbf{R} \boxplus \mathbf{w}) \cdot \mathbf{x} = \left.\frac{\partial}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{0}} \mathrm{Exp}(\mathbf{w}) \cdot \mathbf{y} = [-\mathbf{y}]_\times .$$

Outline

On-manifold optimisation

T.B.A. – TODO

- Objective: maximize the likelihood of observations
- Approximate residuals by first-order Taylor expansion
- Locally optimize for the parameter update
- Iterate until convergence
- Compare: Gauss-Newton vs Levenberg-Marquardt
- Also: Robust Cost functions

Software

g2o – A General Framework for Graph Optimisation (C++)

- Rainer Kuemmerle *et al.*– github.com/RainerKuemmerle/g2o
- optimizing graph-based nonlinear error functions
- *E.g.*, SLAM, Bundle Adjustment, *etc.*

MRPT – Mobile Robot Programming Toolkit

- www.mrpt.org

Libraries for on-manifold operations (template expressions, automatic differentiation):

- Sophus – github.com/strasdat/Sophus
- Wave geometry – github.com/wavelab/wave_geometry
- Kindr – Kinematics and Dynamics for Robotics
  github.com/ANYbotics/kindr [docs]
- *etc.*

Readings

- J. Solà *et al.*– **A micro Lie theory for state estimation in robotics** (2018) [pdf]
- J. L. Blanco Claraco –
  - **Non-Euclidean manifolds in robotics and computer vision: why should we care?** (2013) [pdf]
  - **A tutorial on SE(3) transformation parameterizations and on-manifold optimisation** (2019) [pdf]
- Various documents found on Ethan Eade's webpage [www]
- L. Koppel and S. L. Waslander – **Manifold Geometry with Fast Automatic Derivatives and Coordinate Frame Semantics Checking in C++** (2013) [pdf]