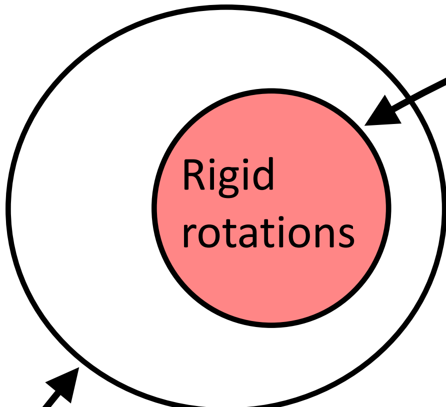
# Metrics on SO(3) and Inverse Kinematics

Luca Ballan

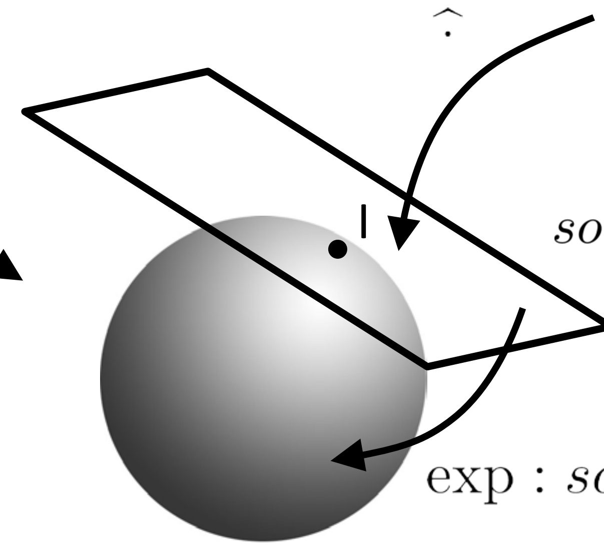
# Last Lecture

**Rigid transformations**
(maps which preserve distances and space orientation)

isomorphic




$\mathbb{R}^n$
- Angle/axis representation




$so\,(n)$ =
- Tangent space at the identity
- Lie algebra of the nxn skew-symmetric matrices

$\exp : so(n) \to SO(n)$
- smooth map
- coincides with the matrix exponential

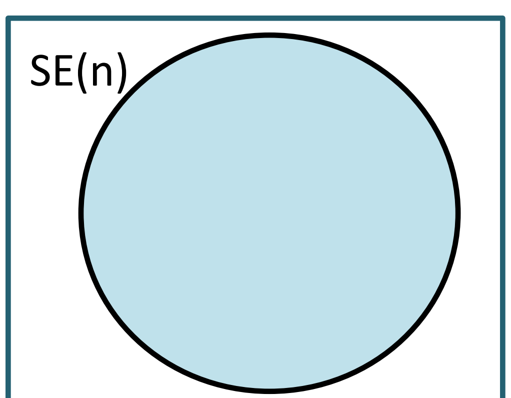
$$\exp(M) = \sum_{k=0}^{\infty} \frac{1}{k!} M^k$$

$SO\,(n)$ =
- k – manifold of the rotation matrices immerse in $\mathbb{R}^{n \times n}$
- Lie group

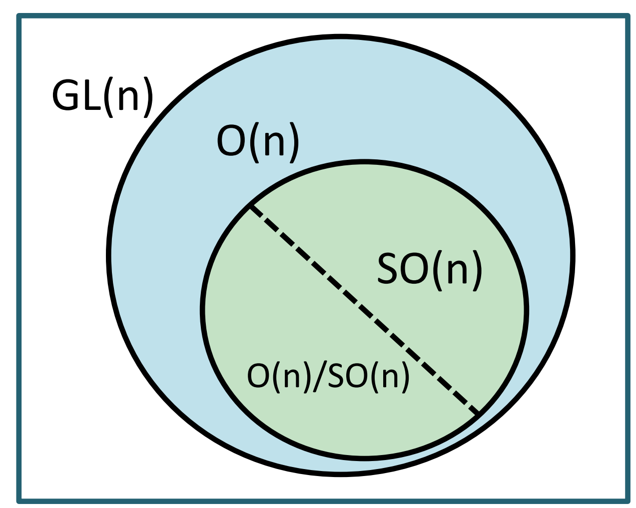
isomorphic

$SE\,(n)$ =
- k+n – Manifold defined as $SO\,(n) \times \mathbb{R}^n$
- Lie group

$\mathbb{R}^{nxn+n}$

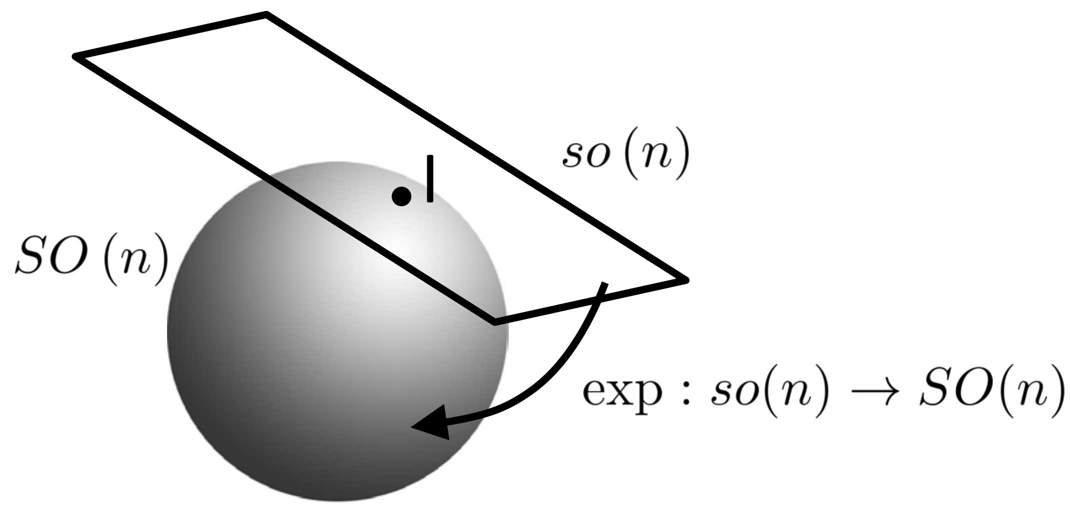



$\mathbb{R}^{nxn}$



# Exponential Map

- The exponential map is a function proper of a Lie Group




- For matrix groups

$$\exp(A) = \sum_{k=0}^{\infty} \frac{1}{k!} A^k$$

- For SO(3), Rodrigues' rotation formula:

$$\exp(\widehat{a}) = I + \frac{\sin(\|a\|)}{\|a\|}\widehat{a} + \frac{(1-\cos(\|a\|))}{\|a\|^2}\widehat{a}^2$$

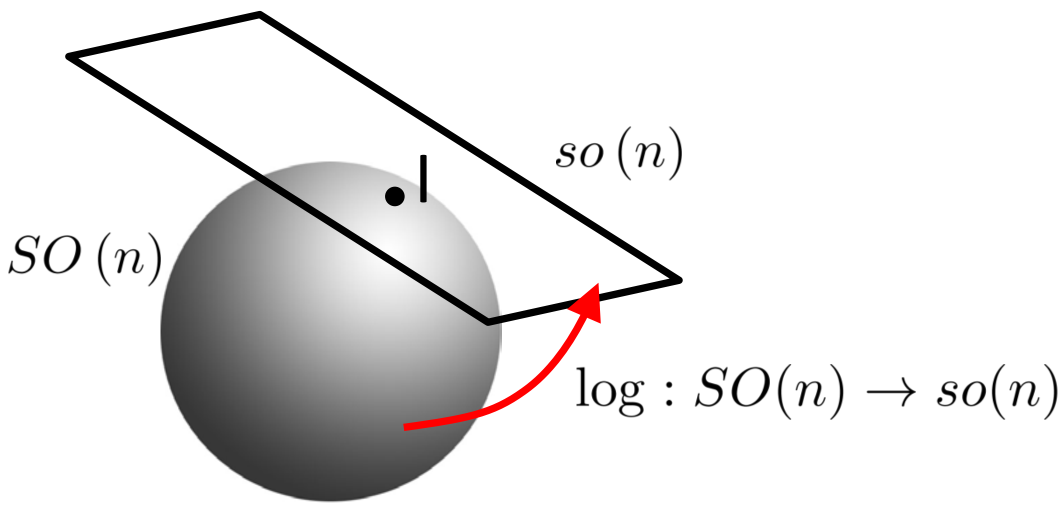
  - Smooth
  - Surjective
  - not Injective

  - not Linear $e^{X+Y} \neq e^X e^Y$ (not an isomorphism)

  - $\partial e^X = \partial X e^X = e^X \partial X$

# Logarithm Map

- Since $\exp(\cdot)$ is surjective… it exists at least an inverse




- The inverse of $\exp(\cdot)$ is

$$\log(X) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}(X - I)^k$$

- For SO(3), Rodrigues' rotation formula:

$$\log(X) = \tfrac{1}{2\sin(\theta)}\left(X - X^T\right) \qquad\qquad R \neq I$$

$$\theta = arccos\left(\tfrac{trace(X)-1}{2}\right)$$

# Properties

$\log(I) = 0 = \widehat{0}$ **Identity**

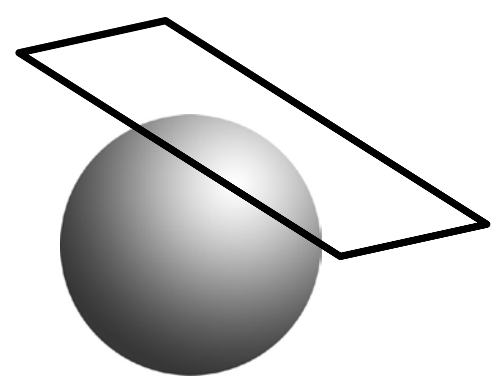
$\log(X^{-1}) = -\log(X)$ **Inverse**

$\log(XY) \neq \log(X) + \log(Y)$ **in general not "Linear"** **(different from the standard log in $\mathbb{R}$)**

$$e^X e^Y \neq e^Y e^X$$

$$e^{\log(X)} = X$$

$$\log(e^A) = \ ?$$

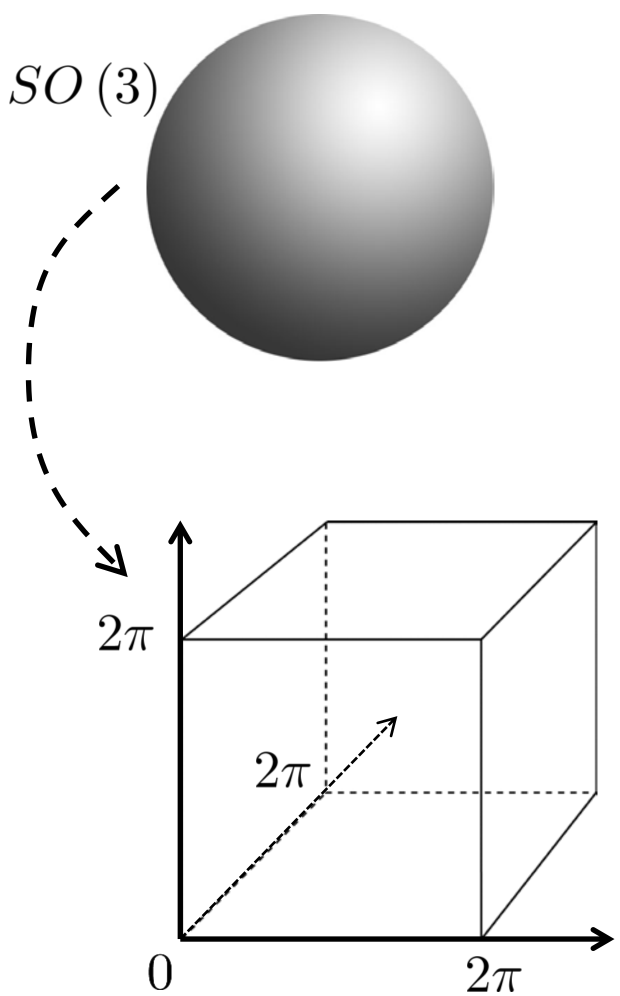


$\partial \log(X) = X^{-1} \partial X$ **Derivative**

$\boxed{*}$

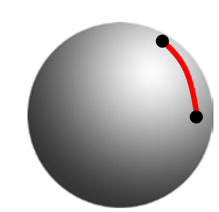
# Last Lecture

- There exists a famous "local charts" for $SO\,(3)$




- Euler-Angle representation (cube)

- any rotation matrix in SO(3) can be describe as a non-unique combination of 3 rotations
  (e.g. one along the x-axis, one on the y-axis, and one on the z-axis)
  - Although it is widely used, this representation has some problems
    - Topology is not conserved $(\,0\,,2\pi\,)$
    - Metric is distorted



    - Derivative is complex
      (although people use it)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & -sin(\alpha) \\ 0 & sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} cos(\beta) & 0 & sin(\beta) \\ 0 & 1 & 0 \\ -sin(\beta) & 0 & cos(\beta) \end{bmatrix} \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 \\ sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

    - Gimbal Lock

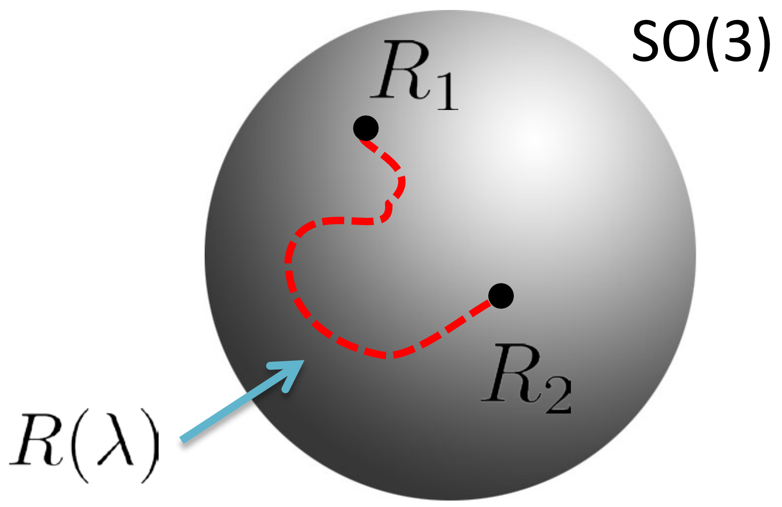
- Intuitive (easy to visualize)
- Easy to set constraints

# Content

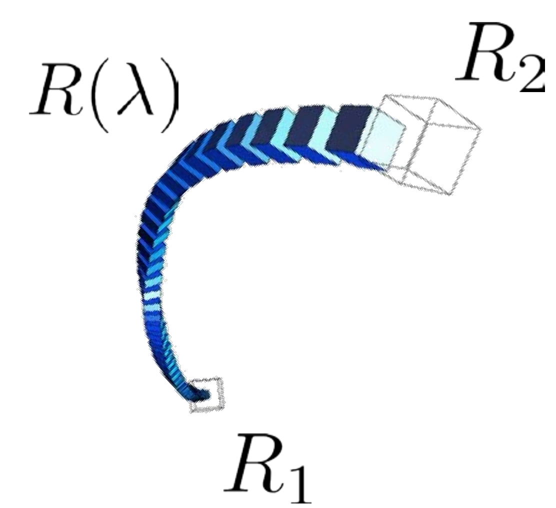
- **Interpolation in SO(3)**
- Metric in SO(3)
- Kinematic chains

# Interpolation in SO(3)

- Given two rotation matrices $R_1, R_2 \in SO(3)$, one would like to find a **smooth path** in SO(3) connecting these two matrices.



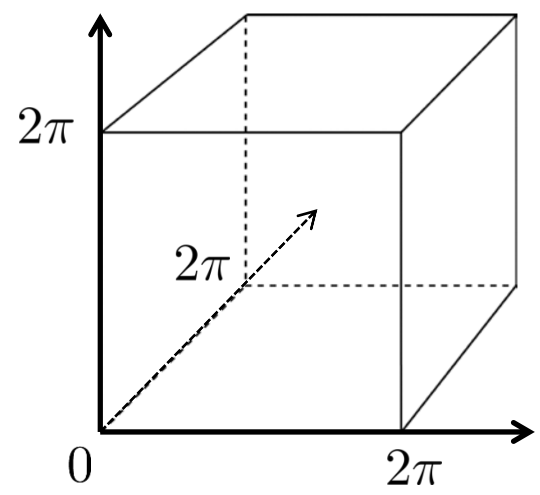

$$\begin{cases} R(\lambda) \in SO(3) \qquad \lambda \in [0,1] \\ R(\lambda) \quad \text{smooth} \\ R(0) = R_1 \\ R(1) = R_2 \end{cases}$$



# Interpolation in SO(3)

**Approach 1**: Linearly interpolate R1 and R2 in one of their representation

- **Euler angles**:



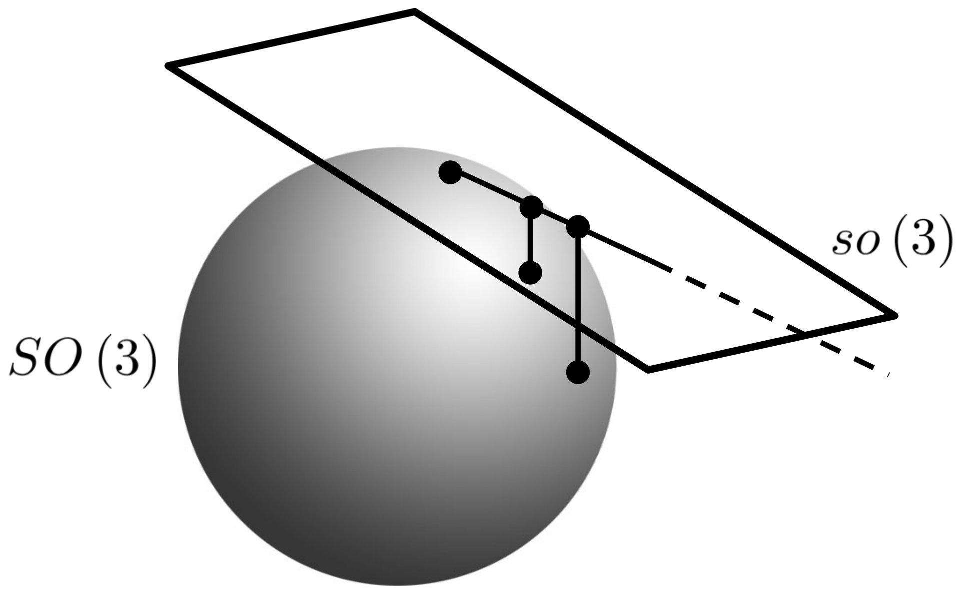

- R1, R2 too far -> not intuitive motion
- Topology is not conserved

# Interpolation in SO(3)

**Approach 1**: Linearly interpolate R1 and R2 in one of their representation

- **Angle-Axis**:

$$\omega(\lambda) = (\lambda\omega_1 + (1-\lambda)\omega_2)$$




- Interpolate on a plane and then project on a sphere
- The movement is not linear with a constant speed. It gets faster the more away it is from the Identity

# Interpolation in SO(3)

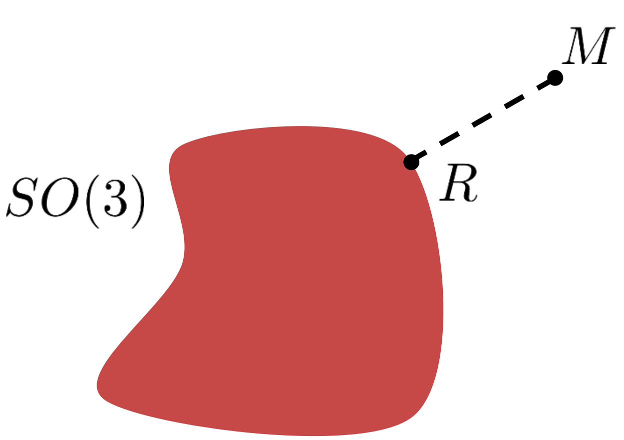
**Approach 2**: Linearly interpolate R1 and R2 as matrices

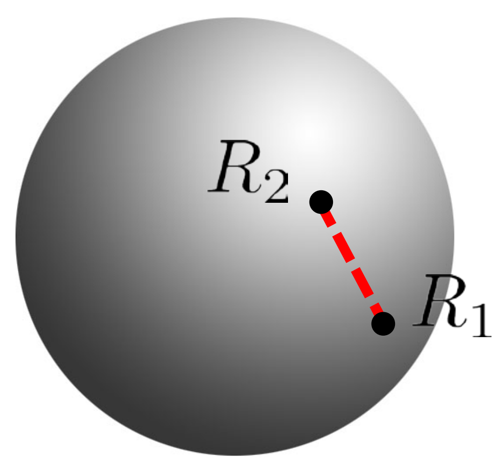
$$R(\lambda) = \underbrace{(\lambda R_1 + (1-\lambda)R_2)}_{\notin SO(3)}$$

Not an element of SO(3) because it is a multiplicative group not an additive one

- it needs to be projected back on the sphere

$$\pi_{SO(3)}(M) = \underset{R \in SO(3)}{\arg\min} \|M - R\|_F^2$$



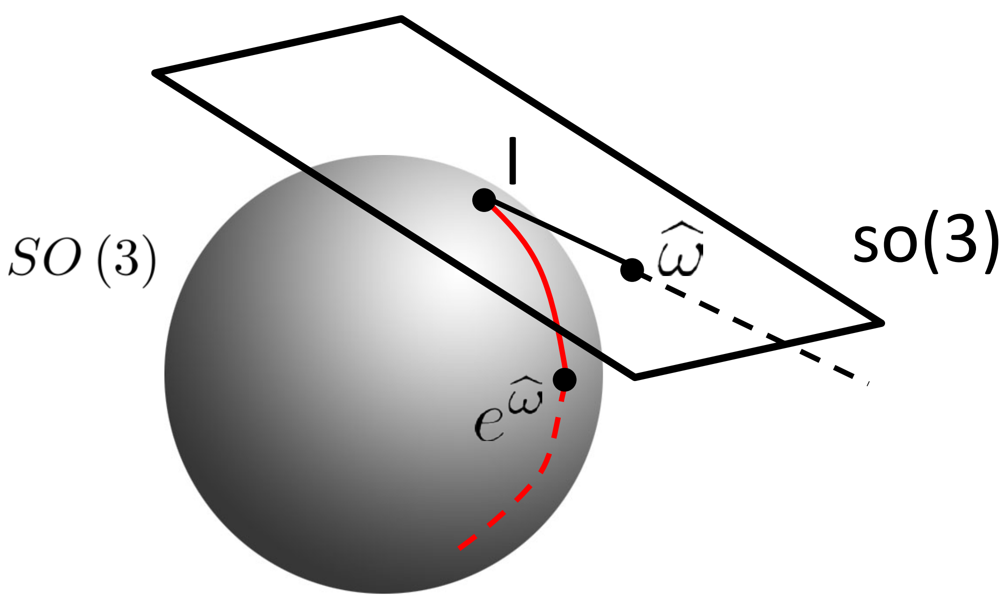




- if R1, R2 are far away from each other, the speed is not linear at all

# Interpolation in SO(3)

**Approach 3**: use the geodesics of SO(3)

- Lie Groups: a line passing through 0 in the Lie algebra **maps** to a geodesic of the Lie group through the identity




⇨ consequently the curve

$$R(\lambda) = e^{\lambda\widehat{\omega}}$$

is a geodesic of SO(3) passing through I

This holds only for any line passing through 0 and consequently for any geodesic passing through the identity

# Interpolation in SO(3)

- To find the geodesic passing through $R_1$ and $R_2$ we need to rotate the ball SO(3) by $R_1^{-1}$

$$R_1^{-1}R_1 \longrightarrow R_1^{-1}R_2$$

$$I \longrightarrow e^{\log(R_1^{-1}R_2)}$$

$$I \longrightarrow e^{\lambda \log(R_1^{-1}R_2)}$$ geodesic between I and $R_1^{-1}R_2$

SLERP (spherical linear interpolation) $$R_1 \longrightarrow R_1 e^{\lambda \log(R_1^{-1}R_2)}$$ geodesic between $R_1$ and $R_2$

- The resulting motion is very intuitive and it is performed at uniform angular speed in $SO(3)$

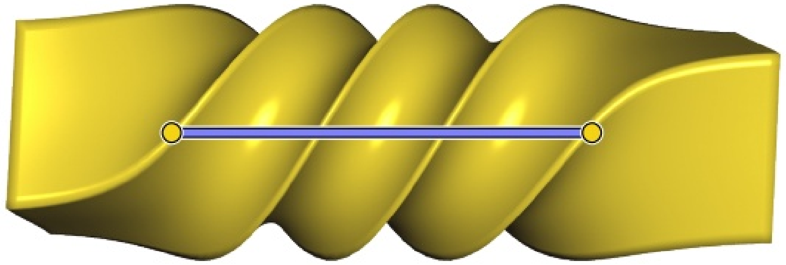
# Interpolation in SO(3)

- On a vector space with Euclidean metric, the geodesic connecting $R_1$ and $R_2$ would have corresponded to the straight line

$$R(\lambda) = R_1 + \lambda(R_2 - R_1)$$

# Questions?

- Given two rotations R1 and R2, interpolate along the geodesic starting from R1 passing n times through R2 and R1 and ending in R2.

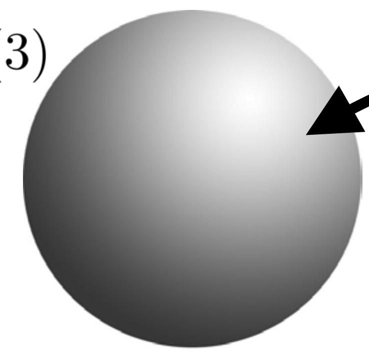


from [jacobson 2011]

something like this but not limited to a single axis

# A word about quaternions...

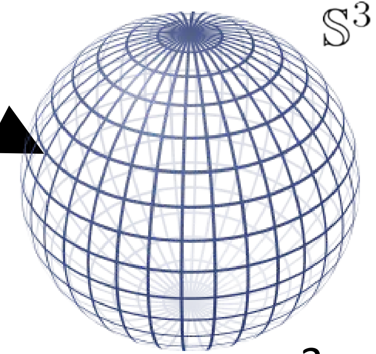
$SO(3)$



3-manifold

isomorphic

isometry



3-manifold

$\mathbb{S}^3/\{-I, I\}$ = the hypersphere in $\mathbb{R}^4$ quotient the antipodal points.
(the hemisphere)

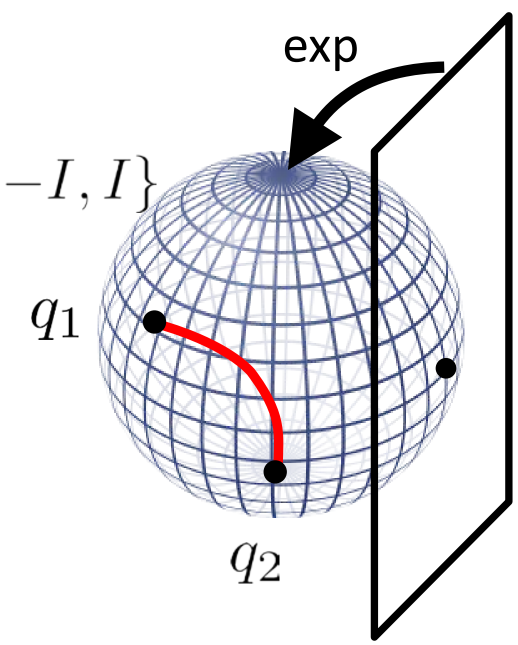
- quaternion multiplication

$(\mathbb{S}^3/\{-I, I\}, \cdot) =$ **Group of the unit quaternions**

- 3-manifold
- Lie group

exp

$\mathbb{S}^3/\{-I, I\}$




I = $(\vec{1},0)$

$\mathrm{Im} = \{(0, \vec{v}) \mid \forall \vec{v} \in \mathbb{R}^3\}$

Tangent space at the identity
- Lie Algebra
- Imaginary numbers

Exponential map
- quaternion exponential

Geodesics

$$q_1 \longrightarrow q_1 e^{\lambda \log(q_1^{-1} q_2)}$$

$$q_1 \longrightarrow q_1 (q_1^{-1} q_2)^{\lambda}$$

- they are the same as the geodesics in $SO(3)$

- PRO: easy to compute SLERP
- CON: difficult to perform derivatives in this space $\quad q \cdot s \cdot q^{-1}$

# Content

- Interpolation in SO(3)
- **Metric in SO(3)**
- Kinematic chains

# Metric in SO(3)

- We talk about geodesics, but what was the used metric?

  - a metric tells how close two rotations are

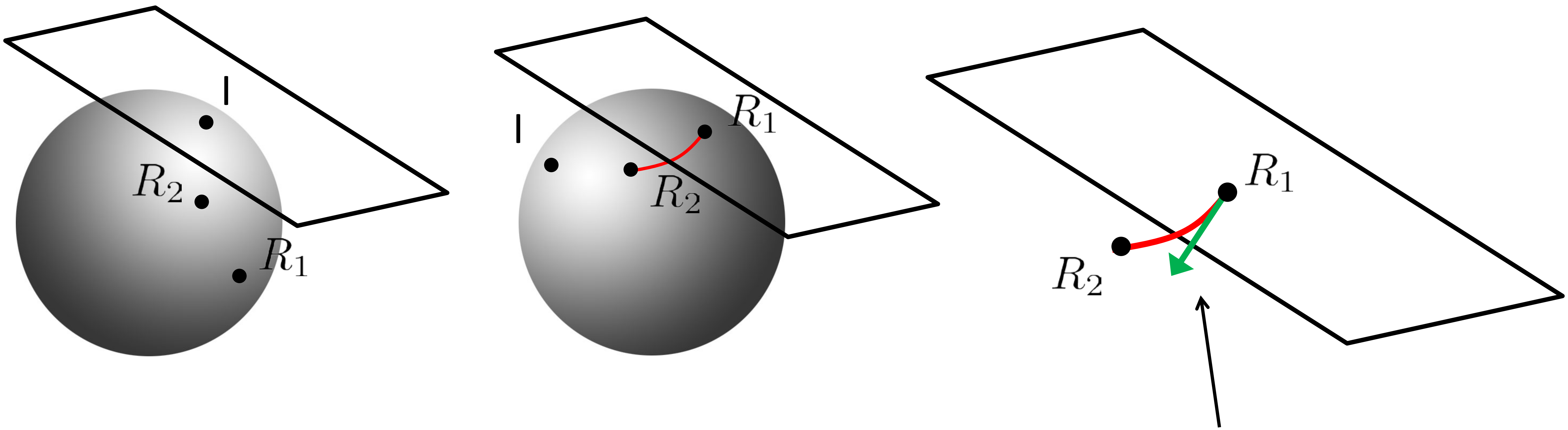
  - it is necessary to evaluate an estimator w.r.t. a ground truth

# Metric in SO(3)

- We talk about geodesics, but what was the used metric?

$$d_R(R_1, R_2) = \frac{1}{\sqrt{2}} \| \log(R_1^{-1} R_2) \|_F$$

Riemannian/Geodesic/Angle metric
(= to the length of the geodesic
connecting R1 and R2)




$\| \cdot \|_F$ = $\sqrt{2} * \text{lenght}$

# Metric in SO(3)

- We talk about geodesics, but what was the used metric?

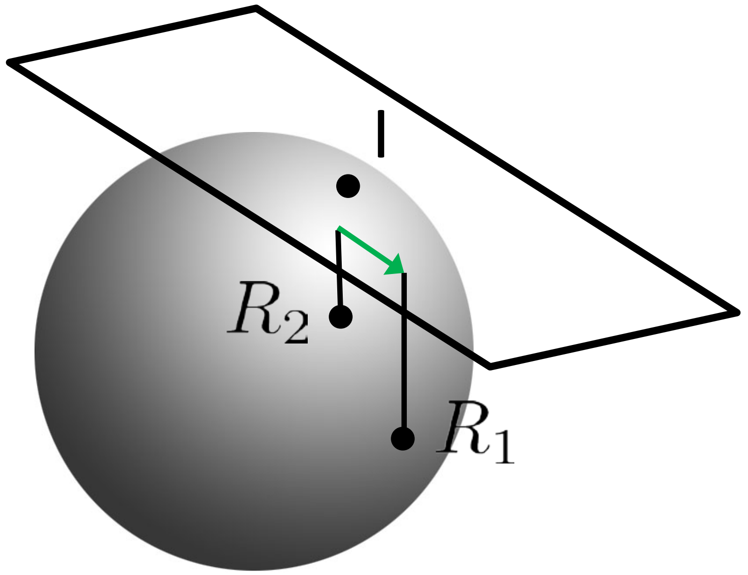
$$d_R(R_1, R_2) = \frac{1}{\sqrt{2}} \| \log(R_1^{-1} R_2) \|_F$$

Riemannian/Geodesic/Angle metric
(= to the length of the geodesic
connecting R1 and R2)

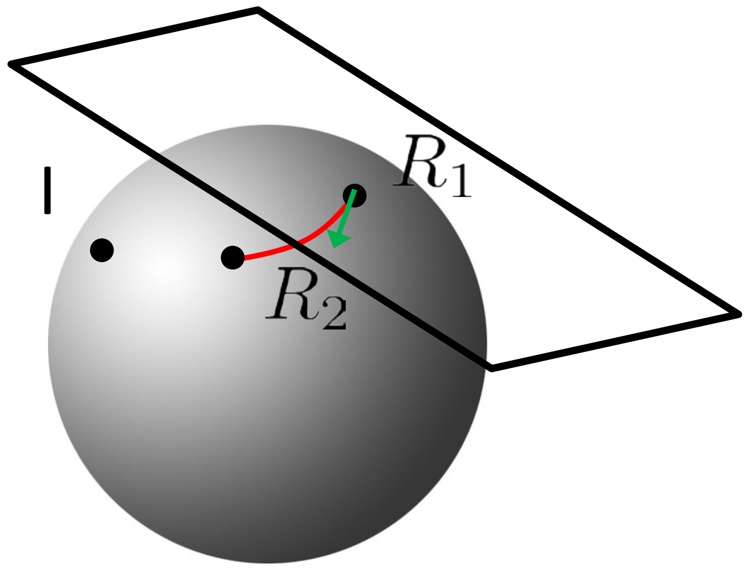
$$d_H(R_1, R_2) = \| \log(R2) - \log(R1) \|_F$$

Hyperbolic metric

- similar to the Riemannian if R1=I




Hyperbolic metric




Riemannian metric

# Metric in SO(3)

- We talk about geodesics, but what was the used metric?

$$d_R(R_1, R_2) = \frac{1}{\sqrt{2}} \| \log(R_1^{-1} R_2) \|_F$$

Riemannian/Geodesic/Angle metric
(= to the length of the geodesic
connecting R1 and R2)

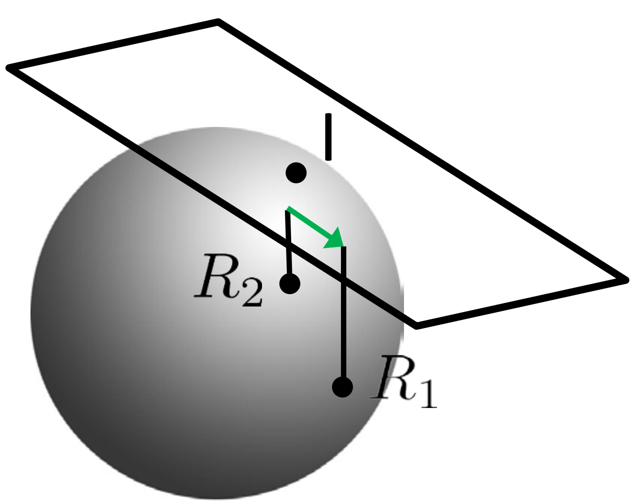
$$d_H(R_1, R_2) = \| \log(R2) - \log(R1) \|_F$$

Hyperbolic metric

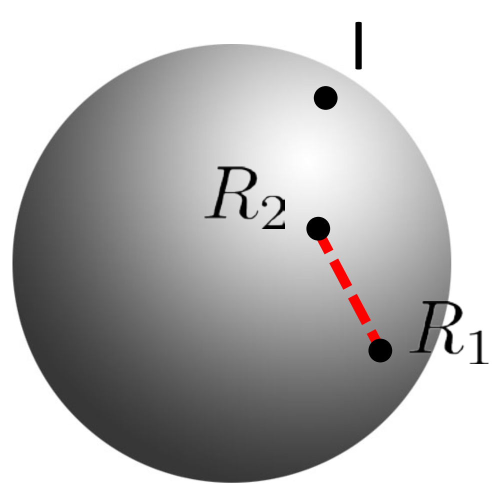
$$d_F(R_1, R_2) = \| R1 - R2 \|_F$$

Frobenius/Chordal metric

- not similar to Hyperbolic
- similar to the Riemannian if R1 and R2 are close to each other




Hyperbolic metric




Frobenius metric

# Metric in SO(3)

- We talk about geodesics, but what was the used metric?

$$d_R(R_1, R_2) = \frac{1}{\sqrt{2}} \| \log(R_1^{-1} R_2) \|_F$$

Riemannian/Geodesic/Angle metric
(= to the length of the geodesic
connecting R1 and R2)

$$d_H(R_1, R_2) = \| \log(R2) - \log(R1) \|_F$$

Hyperbolic metric

$$d_F(R_1, R_2) = \| R1 - R2 \|_F$$

Frobenius/Chordal metric

$$d_{\mathbb{S}^3}(q_1, q_2) = \| q_1 - q_2 \|_2$$

Quaternion metric
(related to the space of quaternions, not specifically to the sphere of unit quaternions)

- Similar to the Hyperbolic one

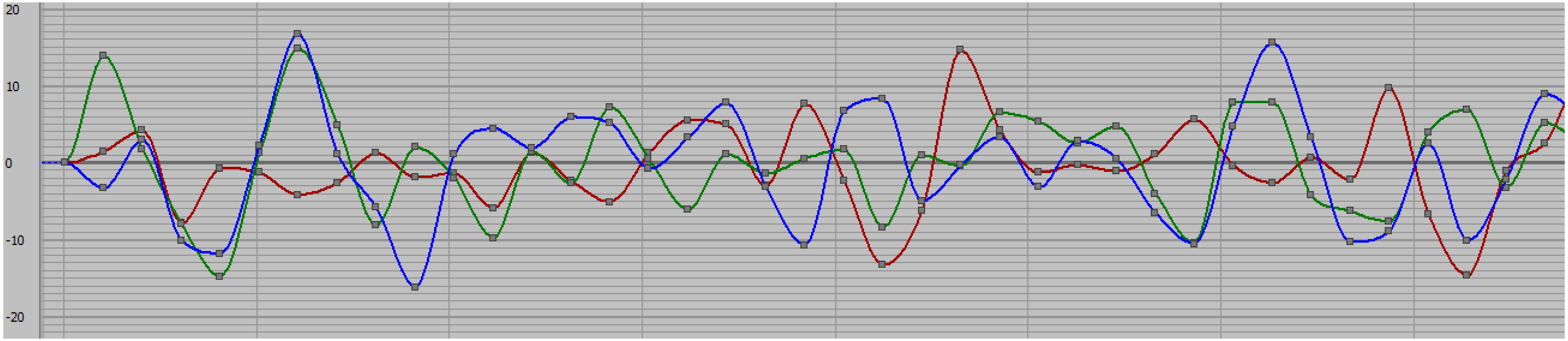
# Filtering in SO(3)

- Given n different estimation for the rotation of an object

$$R_1, \ldots, R_n$$

- how can I get a better estimate of $R$ ?

Object at unknown rotation $R$



# Filtering in SO(3)

- Given n different estimation for the rotation of an object

$$R_1, \ldots, R_n$$

- how can I get a better estimate of $R$ ?

Object at unknown rotation $R$

- **Solution:** which of these is the best?
  - Average the rotation matrices $R_i$ ? $\quad \frac{1}{n}\sum_{i=1}^{n} R_i \quad$ (not rotation)
  - Average the Euler angles of each $R_i$? $\quad \left(\frac{1}{n}\sum_{i=1}^{n}\alpha_i, \frac{1}{n}\sum_{i=1}^{n}\beta_i, \frac{1}{n}\sum_{i=1}^{n}\gamma_i\right)$
  - Average the angle-axes of each $R_i$ ? $\quad \frac{1}{n}\sum_{i=1}^{n}\omega_i$
  - Average the quaternions related to each $R_i$? $\quad \frac{1}{n}\sum_{i=1}^{n} q_i$

(rotation matrices)

- Why average?

# Filtering in SO(3)

- Why average?
  - By saying average, I'm implicitly assuming that the error in the measurements is Gaussian with zero mean

$$\operatorname*{argmin}_x \sum_{i=1}^{n} \|x - x_i\|^2$$

Average $= \frac{1}{n}\sum_{i=1}^{n} x_i$

$\ell_2$

$$\operatorname*{argmin}_x \sum_{i=1}^{n} \|x - x_i\|$$

Median $= \mathrm{sort}(\{x_i\})[n/2]$

$\ell_1$

This can be generalized using metrics instead of norms

# Filtering in SO(3)

- Why average?
  - By saying average, I’m implicitly assuming that the error in the measurements is Gaussian with zero mean

$$\operatorname*{argmin}_x \sum_{i=1}^n d(x, x_i)^2$$

Average $\ell_2$

$$= \frac{1}{n} \sum_{i=1}^n x_i$$

$$\operatorname*{argmin}_x \sum_{i=1}^n d(x, x_i)$$

Median $\ell_1$

$$= \mathrm{sort}(\{x_i\})[n/2]$$

This formulas can be applicable only to $\mathbb{R}$ neither to $\mathbb{R}^n$

in case of SO(3),
which metric do we use here?

# Filtering in SO(3)

$$d_H(R_1, R_2) = \| \log(R2) - \log(R1) \|_F$$

$$\Downarrow$$

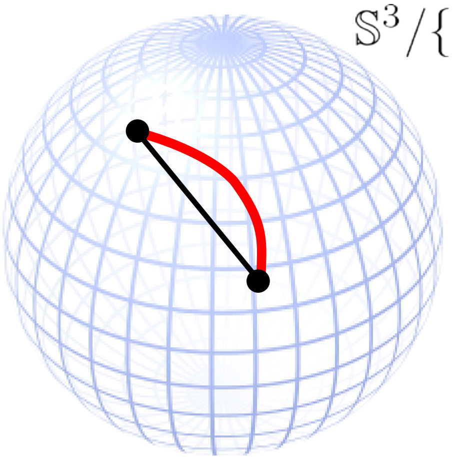
$$\underset{R \in SO(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d_H(R, R_i)^2$$

Geometric mean

- $= \frac{1}{n} \sum_{i=1}^{n} \log(R_i) = \frac{1}{n} \sum_{i=1}^{n} \omega_i$ Average of the angle-axes of each $R_i$

$\boxed{*}$

- Similar to the **projection** of $\frac{1}{n} \sum_{i=1}^{n} q_i$



# Filtering in SO(3)

$$d_F(R_1, R_2) = \|R1 - R2\|_F$$

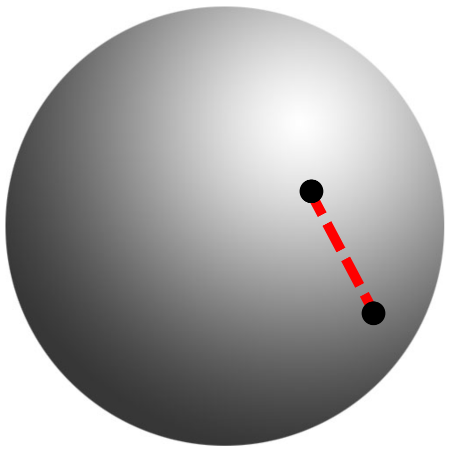
$$\Downarrow$$

$$\underset{R \in SO(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d_F(R, R_i)^2$$

Matrix mean

- Similar to the **projection** of $\frac{1}{n}\sum_{i=1}^{n} R_i$

Average of the each matrix element



# Filtering in SO(3)

$$d_R(R_1, R_2) = \frac{1}{\sqrt{2}} \| \log(R_1^{-1} R_2) \|_F$$

$\Downarrow$

$$\underset{R \in SO(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d_R(R, R_i)^2$$

Fréchet/Karcker mean

- No close form solution
- Solve a minimization problem

- when the solution R is close to I $\Rightarrow$ = Geometric mean

- when the $R_i$ are all close together $\Rightarrow$ = Matrix mean

# Filtering in SO(3)

$$d_R(R_1, R_2) = \frac{1}{\sqrt{2}} \| \log(R_1^{-1} R_2) \|_F$$

$$\Downarrow$$

$$\underset{R \in SO(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d_R(R, R_i)^2$$

Fréchet/Karcker mean

- Why is so different?
  - we need to find the rotation R such that the squared sum of the lengths of all the geodesics connecting R to each $R_i$ is minimized
  - The geodesics should start from R and not from the identity (like in the geometric mean)
  - we need to find the tangent space such that the squared sum of the lengths of all the geodesics of each $R_i$ is minimized

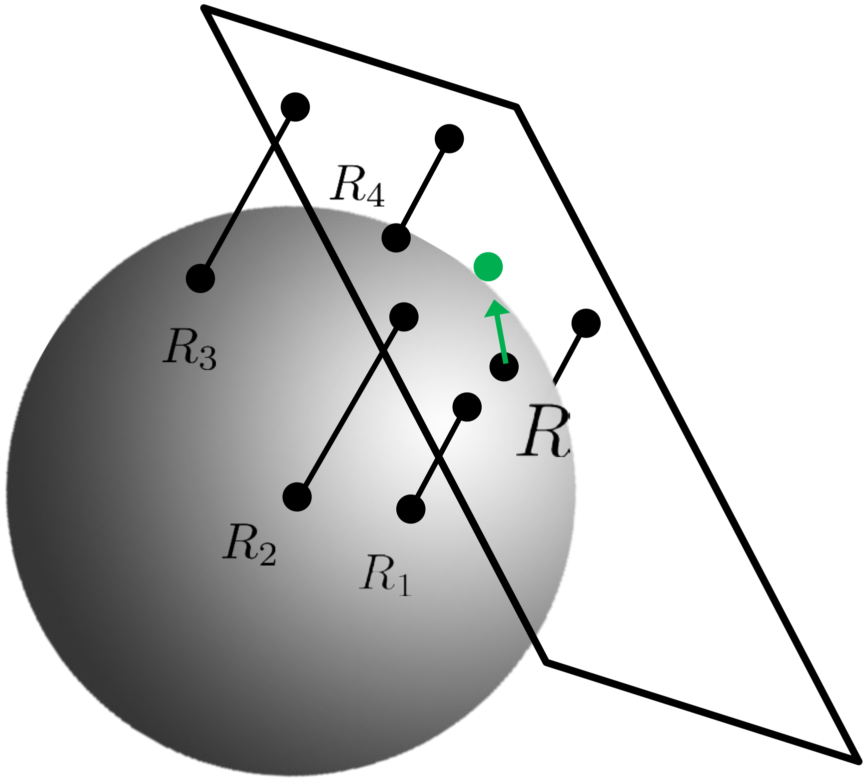
$\boxed{*}$

# Fréchet mean

$$\underset{R \in SO(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d_R(R, R_i)^2$$

- Gradient descent on the manifold
- J. H. Manton, A globally convergent numerical algorithm for computing the centrer of mass on compact Lie groups, ICARCV 2004

- Set $R = \overline{R}$ Matrix or Geometric mean
- Compute the average on the tangent space of $R$ $$r = \sum_{i=1}^{n} \log(R^{-1} R_i)$$
- Move towards $r$ $$R = Re^r$$



# Content

- Interpolation in SO(3)
- Metric in SO(3)
- **Kinematic chains**

# Special Euclidean group SE(3)

$SE(3) = (SO(3) \times \mathbb{R}^3, \times)$

**Special Euclidean group of order 3**

- for simplicity of notation, from now on, we will use homogenous coordinates

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE(3)$$

- A way of parameterize SE(3) is the following

$$\xi = \quad (\omega, t) \rightarrow \begin{bmatrix} e^{\widehat{\omega}} & t \\ 0 & 1 \end{bmatrix} \quad = e^{\widehat{\xi}}$$

This is not the real exponential map in SE(3)
(but it is more intuitive)

Translation $t \in \mathbb{R}^3$

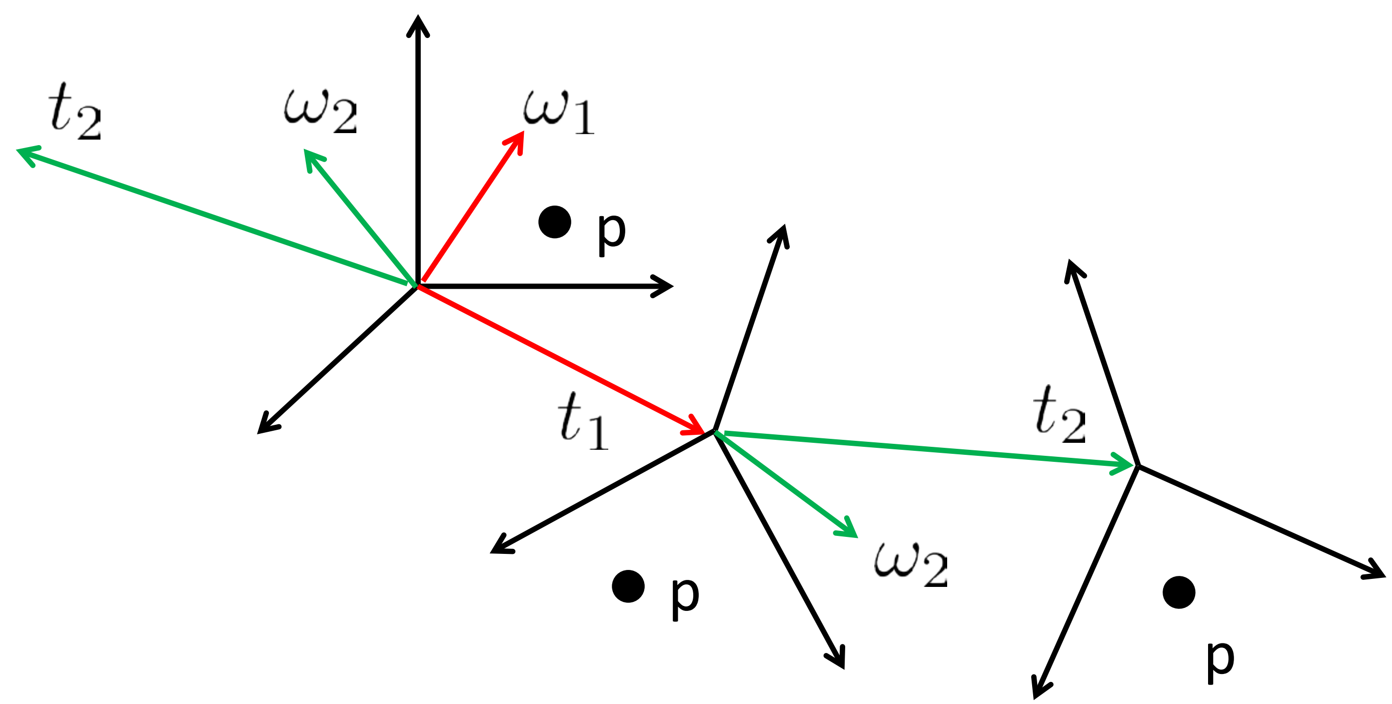
Angle/axis representation of the rotation $\omega \in so(3)$

- $(\omega, t)$ is called **twist**, and usually indicated with the symbol $\xi$

# Composition of Rigid Motions

$$\underbrace{e^{\hat{\xi}_1} \underbrace{e^{\hat{\xi}_2} p}_{\text{Transform p}}}_{\text{Transform the transformation of p}}$$

$$\xi_1 = (\omega_1, t_1)$$
$$\xi_2 = (\omega_2, t_2)$$



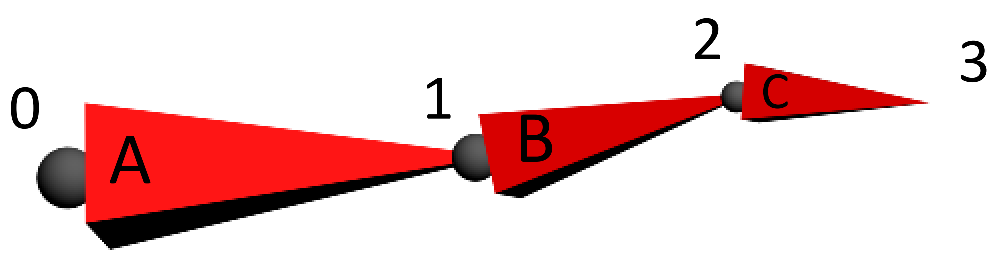

$\xi_2$ is expressed in local coordinates relative to the framework induced by $\xi_1$

The second transformation is actually performed on the twist

$$e^{\hat{\xi}_1} \xi_2$$

# Kinematic Chain

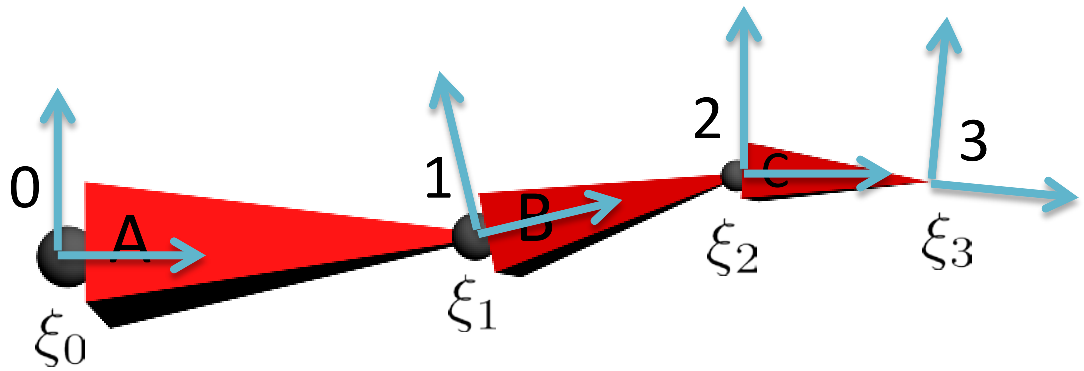
- A kinematic chain is an ordered set of rigid transformations

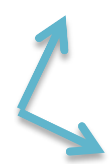



- Each  is called **bone** (A,B,C)
- Each  is called **joint** (0,1,2,3)
- joint 0 is called **base/root** (and assumed to be fixed)
- joint 3 is called **end effector**

# Kinematic Chain

- A kinematic chain is an ordered set of rigid transformations




- Each bone has its own coordinate system  determining its position in the space and the orientation of its local axes

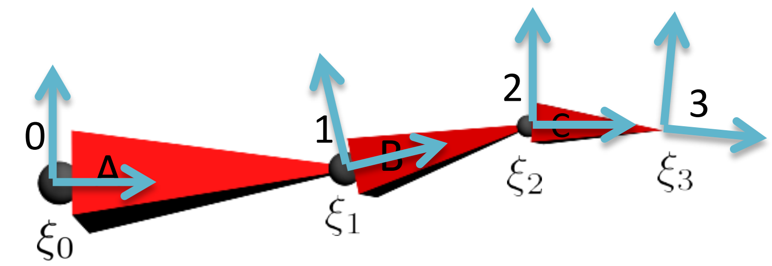
- **the bones A, B, C are oriented accordingly to the x-axis of the reference system**

- The base of each bone corresponds to a joint

- Each reference system is an element of SE(3) determined by a **twists** $(\xi_0, \xi_1, \xi_2, \xi_3)$

- the twists $\xi_0, \xi_1, \xi_2$, and $\xi_3$ all together determine completely the configuration of the kinematic chain

# Kinematic Chain




- The **base twist** $\xi_0$ has the form

$$\xi_0 = (\omega_0, T_0)$$

represents the coordinates of the joint 0

determine the orientation of the reference system of bone A

- All the **internal twists** ($\xi_1$ and $\xi_2$ ) are defined as

$$\xi_1 = (\omega_1, (l_1, 0, 0))$$

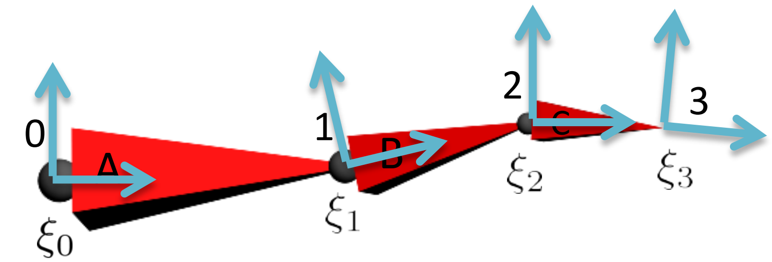
the translation is applied only along the x-axis with amount $l_1$

$$\xi_2 = (\omega_2, (l_2, 0, 0))$$

$l_1$ and $l_2$ denote the length of the bone A and B, respectively

# Kinematic Chain

- The **end effector twist** $\xi_3$ has the form

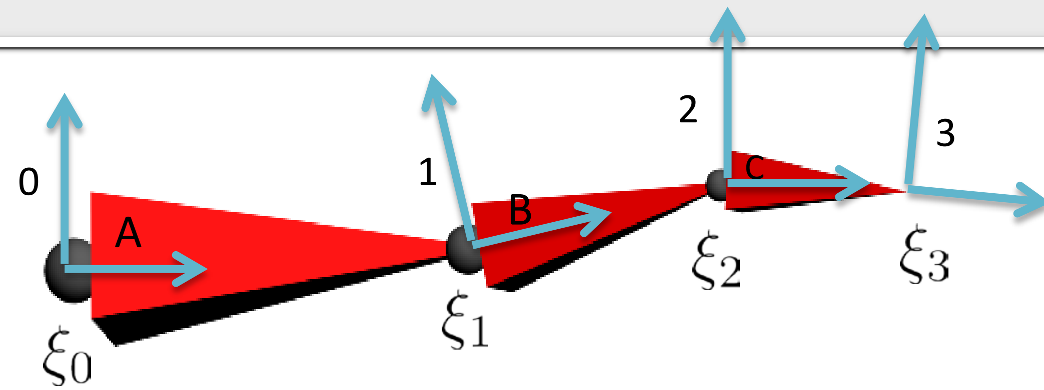



$$\xi_3 = (0, (l_3, 0, 0))$$

$l_3$ denote the length of the bone C

The orientation of the end effector is the same as the bone C

# Kinematic Chain: Summary



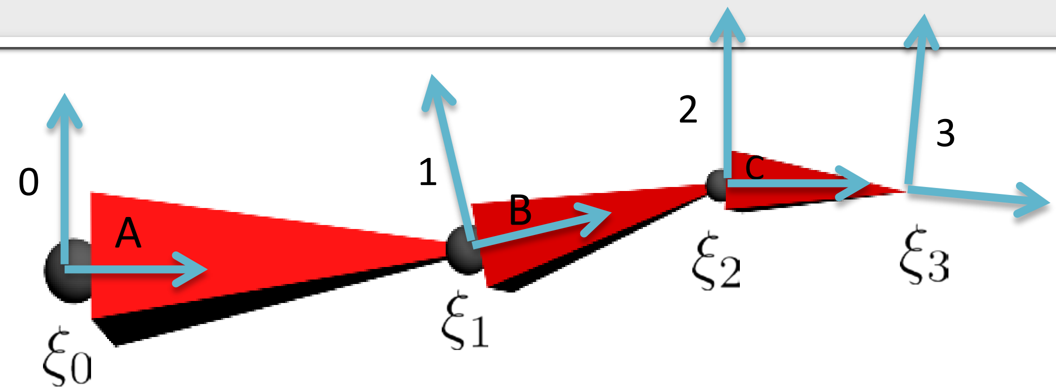

- $\xi_0$ determines the position of joint 0 and the orientation of bone A $\qquad \xi_0 = (\omega_0, T_0)$

- $\xi_1$ determine the position of joint 1, the length of bone A, and the orientation of bone B w.r.t. the reference system of joint 0 $\qquad \xi_1 = (\omega_1, (l_1, 0, 0))$

- $\xi_2$ determine the position of joint 2, the length of bone B, and the orientation of bone C w.r.t. the reference system of joint 1 $\qquad \xi_2 = (\omega_2, (l_2, 0, 0))$

- $\xi_3$ determine the position of joint 3 and the length of bone C $\qquad \xi_3 = (0, (l_3, 0, 0))$

# Kinematic Chain: DOF




- Given the constraints

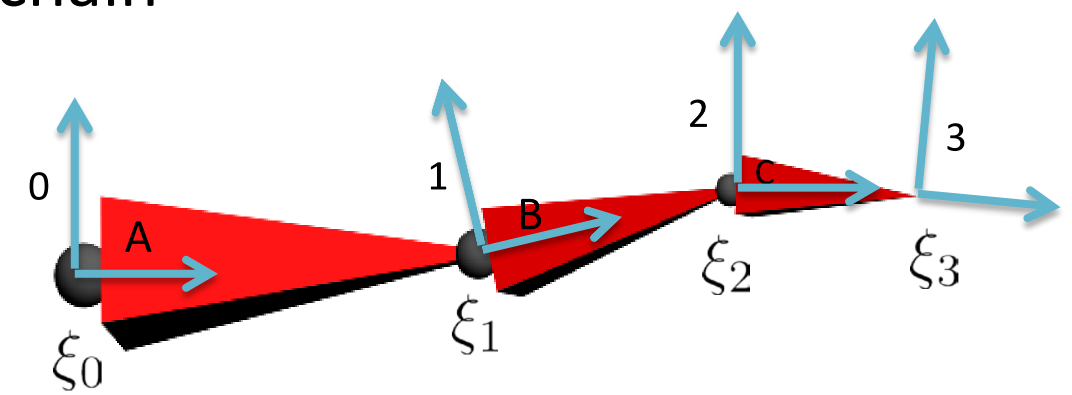
$$\begin{aligned}\xi_0 &= (\omega_0, T_0)\\ \xi_1 &= (\omega_1, (l_1, 0, 0))\\ \xi_2 &= (\omega_2, (l_2, 0, 0))\\ \xi_3 &= (0, (l_3, 0, 0))\end{aligned}$$

- the actual DOFs of this particular kinematic chain are

| | | |
|---|---|---|
| $\omega_0, \omega_1, \omega_2$ | 3x3 DOF | (**ball joints**) |
| $T_0$ | +3 DOF if the base can move | |
| $l_1, l_2, l_3$ | +3x1 DOF if the bone is extendible | (**prismatic joints**) |

# Kinematic Chain Problems

Given a kinematic chain



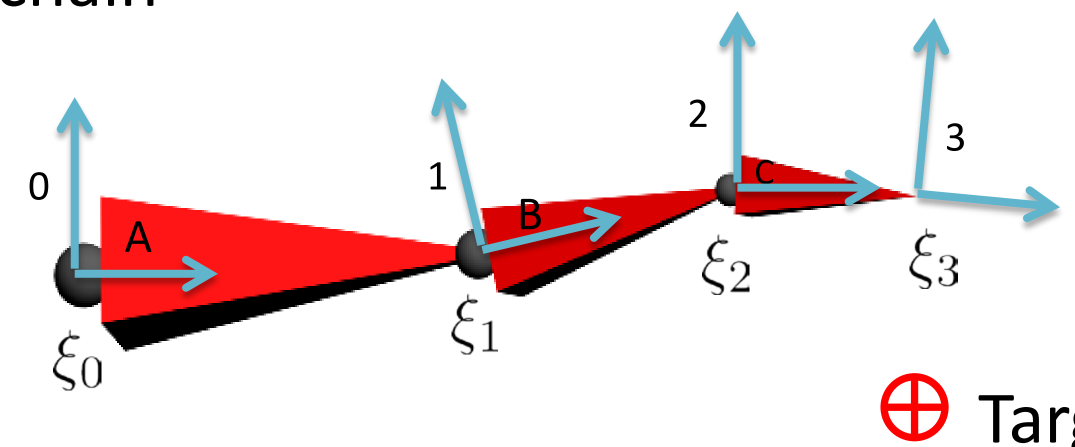

- A **Forward Kinematics Problem** consists in finding the coordinates of the end effector given a specific kinematic chain configuration $(\xi_0, \xi_1, \xi_2, \xi_3)$

$$p(\xi_0, \xi_1, \xi_2, \xi_3) = e^{\widehat{\xi_0}} e^{\widehat{\xi_1}} e^{\widehat{\xi_2}} e^{\widehat{\xi_3}} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Forward Kinematics of the end effector

# Kinematic Chain Problems

Given a kinematic chain




Target **q**

- An **Inverse Kinematics Problem** consists in finding the configuration of the kinematic chain for which the distance between the end effector and a pre-defined target point **q** is minimized

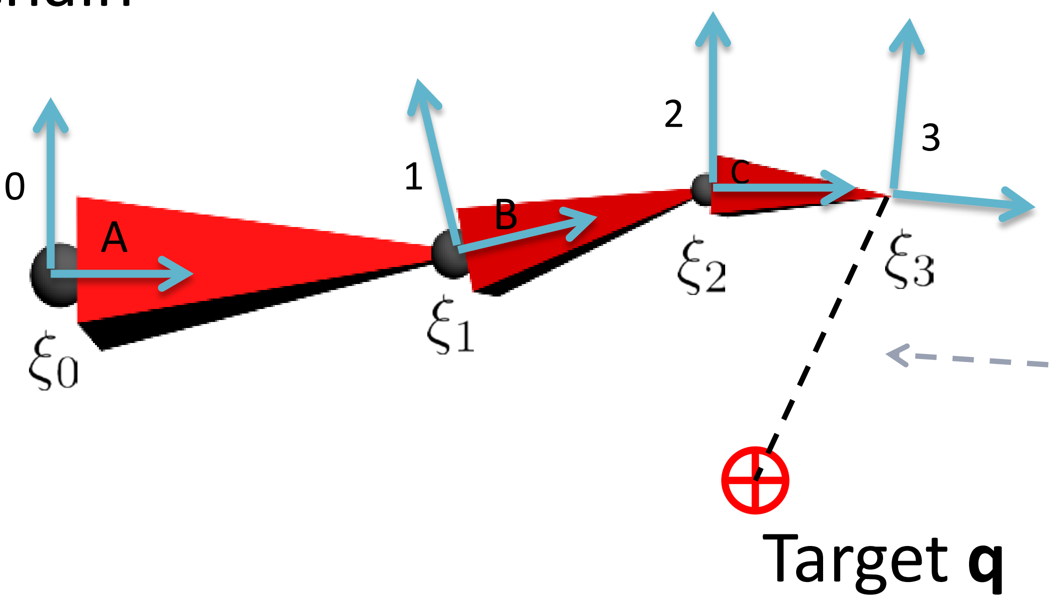
$$
\left\{
\begin{array}{ll}
\arg\min \|p(\xi_0, \xi_1, \xi_2, \xi_3) - q\| & \\
& \\
\text{subject to} & \begin{aligned} \xi_0 &= (\omega_0, T_0) \\ \xi_1 &= (\omega_1, (l_1, 0, 0)) \\ \xi_2 &= (\omega_2, (l_2, 0, 0)) \\ \xi_3 &= (0, (l_3, 0, 0)) \end{aligned}
\end{array}
\right.
$$

$l_1, l_2, l_3$ fixed/or not

$T_0$ fixed/or not

# Inverse Kinematics Problem

Given a kinematic chain



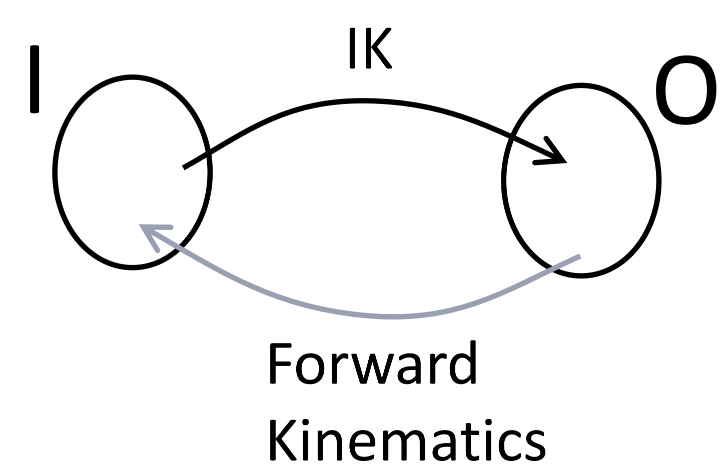

Minimize the distance between where the end effector is and where it should be

$$\arg\min \|\underbrace{p(\xi_0, \xi_1, \xi_2, \xi_3)}_{\text{Generative model for p}} - q\|$$

**Generative model for p**

= Forward Kinematics

**Generative approach** to IK



# Inverse Kinematics Problem

$$\begin{cases} \arg\min \|p(\xi_0, \xi_1, \xi_2, \xi_3) - q\| \\ \text{subject to} \quad \begin{aligned} \xi_0 &= (\omega_0, T_0) \\ \xi_1 &= (\omega_1, (l_1, 0, 0)) \\ \xi_2 &= (\omega_2, (l_2, 0, 0)) \\ \xi_3 &= (0, (l_3, 0, 0)) \end{aligned} \end{cases}$$

$l_1, l_2, l_3$ fixed/or not

$T_0$ fixed/or not

- it is equivalent to a **non-linear least square optimization problem**

(it is equivalent to the squared norm and this is $\|\cdot\|^2 = x^2 + y^2 + z^2$)

(note: here it does not matter if the norm is squared or not, later it will)

$\boxed{*}$

- The problem is **under-constrained**, 3 equations and (at least) 9 unknowns
  - If q is reachable by the kinematic chain, there are **infinite solutions** to the problem
  - If q is not reachable, the solution is unique up to rotations along the bones axes

# A Possible Solution

**Newton's method**

- let denote with $x$ our unknowns $x = (\xi_0, \xi_1, \xi_2, \xi_3)$

$$\arg\min \|p(x) - q\|$$

- let $\overline{x}$ be the current estimate for the solution
- compute the Taylor expansion of $p(x)$ around $\overline{x}$

$$p(x + \Delta x) = p(\overline{x}) + Jp(\overline{x})\Delta x + \ldots$$

$$\arg\min \|\overbrace{p(\overline{x}) + Jp(\overline{x})\Delta x} - q\|$$

$$\Updownarrow$$

$$p(\overline{x}) + Jp(\overline{x})\Delta x - q = 0$$

$$\Updownarrow$$

$$\Delta x = Jp(\overline{x})^{\dagger}(q - p(\overline{x}))$$

$Jp(\overline{x})^{\dagger}$ can be computed using SVD, or approximated as $\cong Jp(\overline{x})^T$ if speed is critical

# The Jacobian of the Forward Kinematics

- Given the forward kinematic $p(\xi_0, \xi_1, \xi_2, \xi_3) = e^{\widehat{\xi_0}} e^{\widehat{\xi_1}} e^{\widehat{\xi_2}} e^{\widehat{\xi_3}} p$

- assuming
$$\begin{aligned} \xi_0 &= (\omega_0, T_0) \\ \xi_1 &= (\omega_1, (l_1, 0, 0)) \\ \xi_2 &= (\omega_2, (l_2, 0, 0)) \\ \xi_3 &= (0, (l_3, 0, 0)) \end{aligned}$$
$l_1, l_2, l_3$ fixed
$T_0$ fixed

- and $\omega_i = (\theta_i^x, \theta_i^y, \theta_i^z)$

- the Jacobian of the forward kinematic is

1x3 column vector

$$Jp = \left[ \frac{\partial p}{\partial \theta_0^x} \quad \frac{\partial p}{\partial \theta_0^y} \quad \frac{\partial p}{\partial \theta_0^z} \quad \frac{\partial p}{\partial \theta_1^x} \quad \frac{\partial p}{\partial \theta_1^y} \quad \frac{\partial p}{\partial \theta_1^z} \quad \frac{\partial p}{\partial \theta_2^x} \quad \frac{\partial p}{\partial \theta_2^y} \quad \frac{\partial p}{\partial \theta_2^z} \right]$$

only one term depends on $\theta_2^y$

$$\frac{\partial p}{\partial \theta_2^y}(\xi_0, \xi_1, \xi_2, \xi_3) = e^{\widehat{\xi_0}} e^{\widehat{\xi_1}} \frac{\partial e^{\widehat{\xi_2}}}{\partial \theta_2^y} e^{\widehat{\xi_3}} p$$

# The Jacobian of the Forward Kinematics

$$\frac{\partial p}{\partial \theta_2^y}(\xi_0, \xi_1, \xi_2, \xi_3) = e^{\widehat{\xi_0}} e^{\widehat{\xi_1}} \frac{\partial e^{\widehat{\xi_2}}}{\partial \theta_2^y} e^{\widehat{\xi_3}} p$$

$$= e^{\widehat{\xi_0}} e^{\widehat{\xi_1}} \begin{bmatrix} \frac{\partial e^{\widehat{\omega_2}}}{\partial \theta_2^y} & 0 \\ 0 & 0 \end{bmatrix} e^{\widehat{\xi_3}} p$$

$$= e^{\widehat{\xi_0}} e^{\widehat{\xi_1}} \begin{bmatrix} \frac{\partial \widehat{\omega_2}}{\partial \theta_2^y} e^{\widehat{\omega_2}} & 0 \\ 0 & 0 \end{bmatrix} e^{\widehat{\xi_3}} p$$

$$= e^{\widehat{\xi_0}} e^{\widehat{\xi_1}} \begin{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} e^{\widehat{\omega_2}} & 0 \\ 0 & 0 \end{bmatrix} e^{\widehat{\xi_3}} p$$

$$\omega_2 = (\theta_2^x, \theta_2^y, \theta_2^z)$$

$$\Downarrow$$

$$\widehat{\omega_2} = \begin{bmatrix} 0 & -\theta_2^z & \theta_2^y \\ \theta_2^z & 0 & -\theta_2^x \\ -\theta_2^y & \theta_2^x & 0 \end{bmatrix}$$

$$\Downarrow$$

$$\frac{\partial \widehat{\omega_2}}{\partial \theta_2^y} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

# The Jacobian of the Forward Kinematics

- and so on... (all the other derivatives are computed in a similar way)

- The Jacobian of forward kinematic is very easy to compute if the angle/axis representation is used. On the contrary, if quaternions are used instead, the Jacobian is not as trivial

$$q_1 \cdot q_2 \cdot s \cdot q_2^{-1} \cdot q_1^{-1}$$