COLUMBIA UNIVERSITY EEME E6911 FALL '25

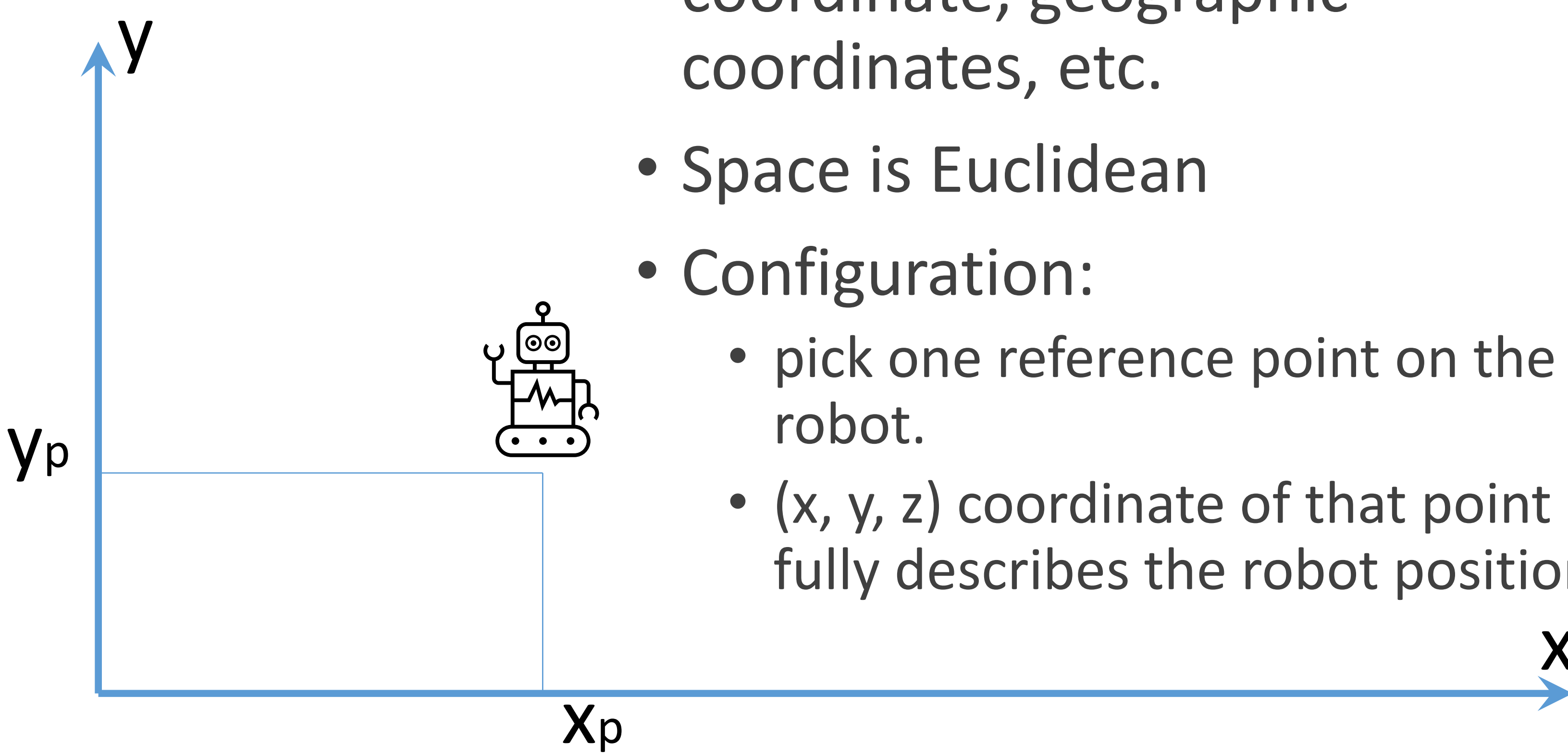
TOPICS IN CONTROL : PROBABILISTIC ROBOTICS

POSE AND UNCERTAINTY
TRANSFORMATIONS AND UNCERTAINTY

Instructor: Ilija Hadzic

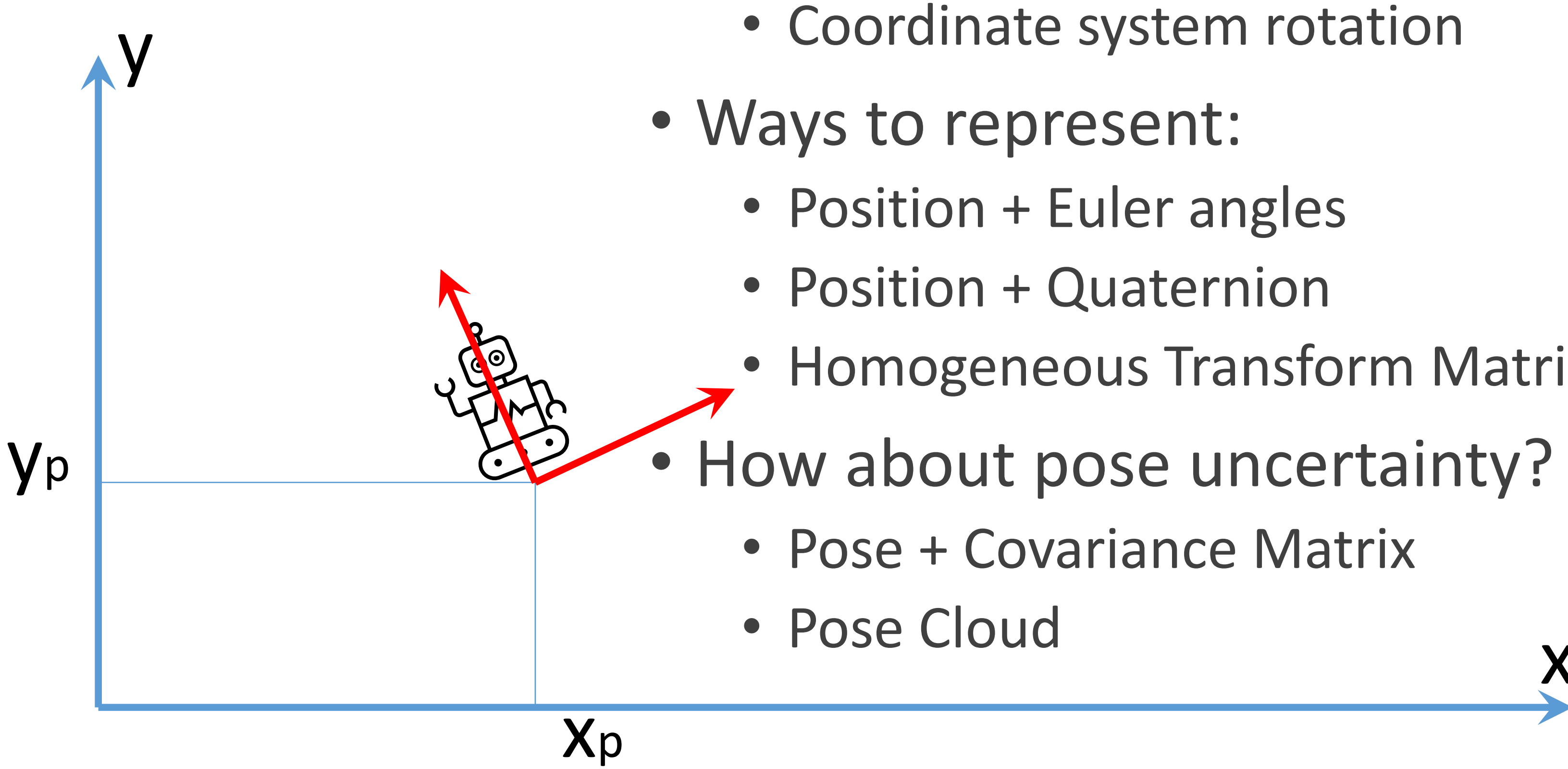
Position

- Displacement vector relative to some frame of reference
- Example: building map coordinate, geographic coordinates, etc.
- Space is Euclidean
- Configuration:
  - pick one reference point on the robot.
  - (x, y, z) coordinate of that point fully describes the robot position.



Pose

- Position and orientation
- Place a coordinate system on the robot:
  - Origin translation
  - Coordinate system rotation
- Ways to represent:
  - Position + Euler angles
  - Position + Quaternion
  - Homogeneous Transform Matrix
- How about pose uncertainty?
  - Pose + Covariance Matrix
  - Pose Cloud



Transformations

- Transformation from local (robot) frame of reference to the global (map) frame of reference is the robot pose.
- Transformation is the displacement needed to bring the global frame into alignment with the local frame.
- Transformation is the conversion of point (or vector) coordinates in the local frame to global frame.

Rotation

$$\begin{bmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{t}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

Translation

Notation

Lynch:

$$\boldsymbol{T}_{RA}$$

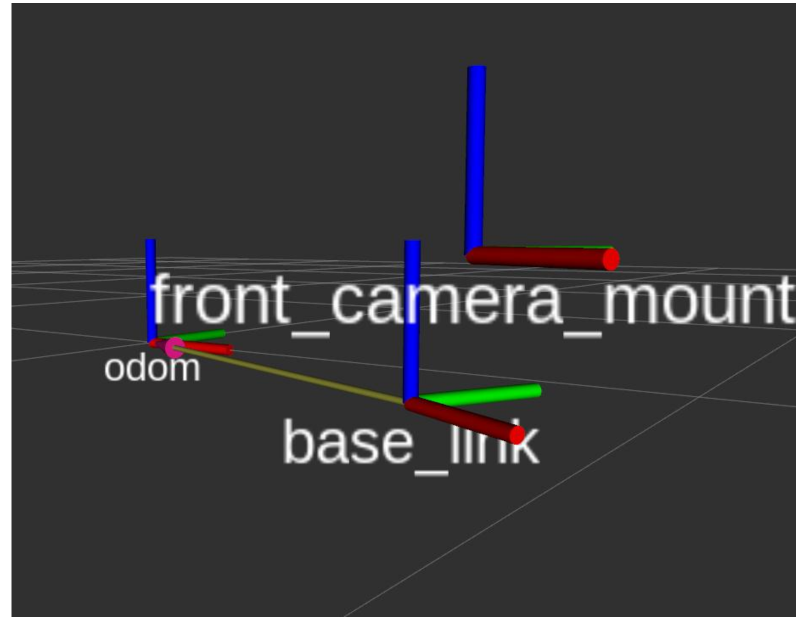
Reference frame

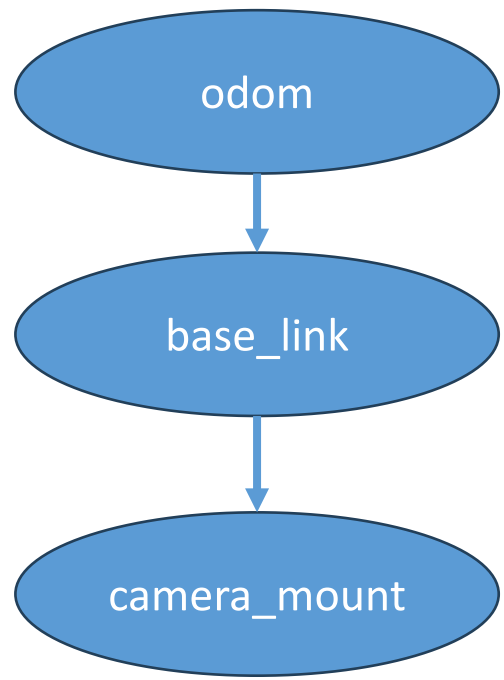
Local frame

Other literature:

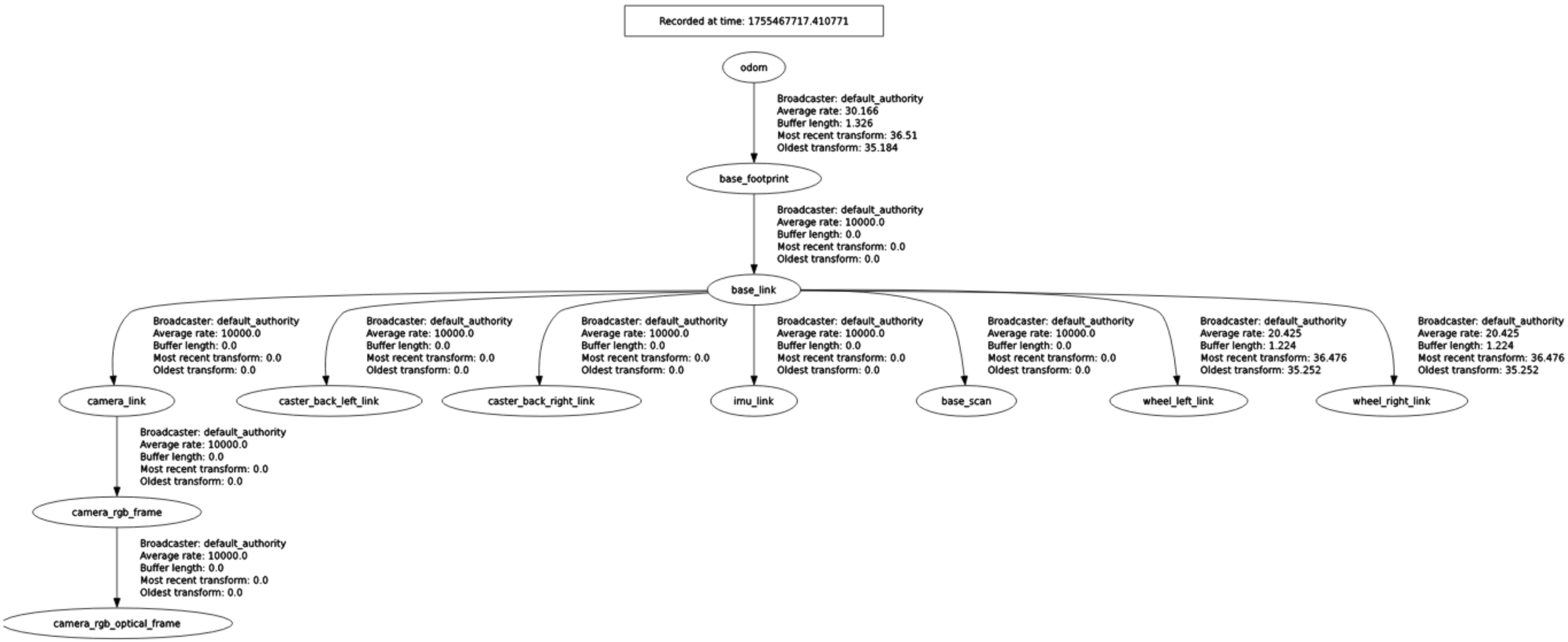
$$^{R}_{A}\boldsymbol{T}$$

$$\boldsymbol{T}^{R}_{A}$$

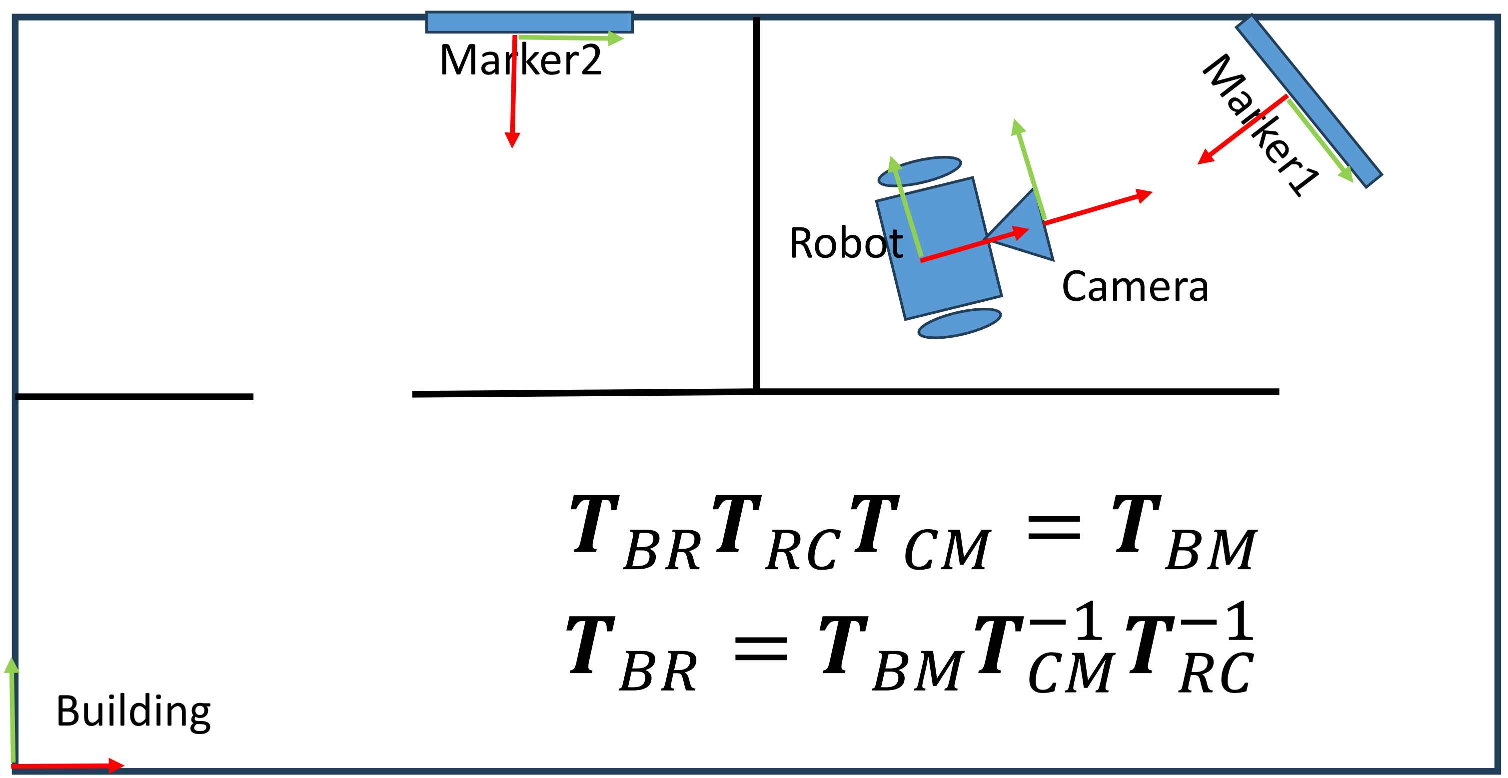




Example: Turtlebot 3 frames



Why do we care about transforms?



$$\boldsymbol{T}_{BR}\boldsymbol{T}_{RC}\boldsymbol{T}_{CM} = \boldsymbol{T}_{BM}$$

$$\boldsymbol{T}_{BR} = \boldsymbol{T}_{BM}\boldsymbol{T}_{CM}^{-1}\boldsymbol{T}_{RC}^{-1}$$

Uncertainty Propagation (2D case)

- Have 2D pose with uncertainty
- Pose is transformed deterministically
  - Left multiplication
  - Right multiplication
  - Both sides
- Characterize the uncertainty of the result
- Example:
  - State: robot body pose (robot local frame)
  - Measurement: landmark vision (camera local frame)
  - Problem: need the prior for camera local frame, not robot!

Definitions

- Robot body position: $\boldsymbol{p} + \boldsymbol{\Delta p}$

$$\boldsymbol{p} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \boldsymbol{\Delta p} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \qquad \boldsymbol{\Delta p}: \mathcal{N}(\mathbf{0}, \Sigma_{xy})$$

- Robot body yaw: $\theta + \Delta\theta$

$$\Delta\theta: \mathcal{N}(0, \sigma_\theta^2)$$

- Pose and pose covariance:

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{p} \\ \theta \end{bmatrix} \qquad \Sigma_P = \begin{bmatrix} \Sigma_{xy} & & 0 \\ & & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix}$$

Pose as transformation

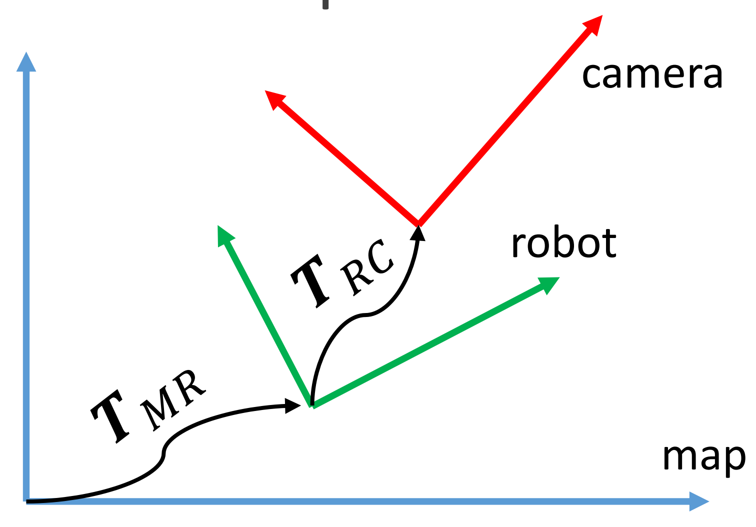
- Uncertain robot pose in map reference frame

$$\boldsymbol{T}_{MR} = \begin{bmatrix} \boldsymbol{R}(\theta + \Delta\theta) & \boldsymbol{p} + \boldsymbol{\Delta p} \\ \boldsymbol{0}_{1\times 2} & 1 \end{bmatrix}_{3\times 3}$$

- Uncertain *camera* pose in map reference frame:

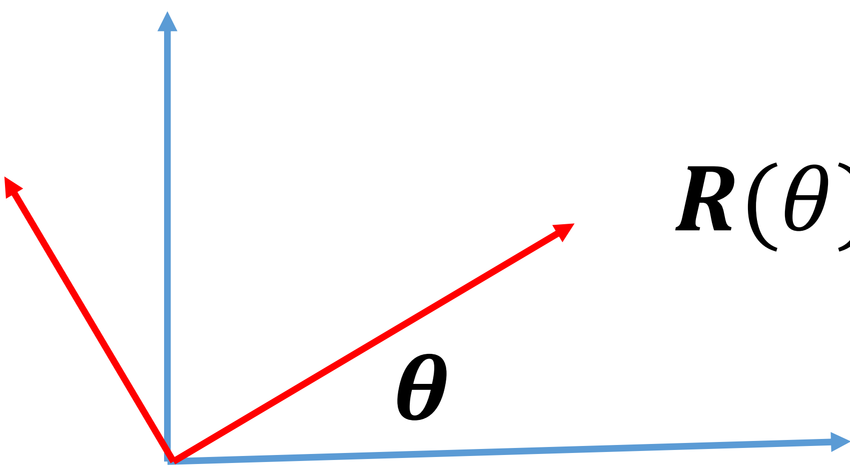
$$\boldsymbol{T}_{MC} = \boldsymbol{T}_{MR}\boldsymbol{T}_{RC}$$

$$\boldsymbol{T}_{RC} = \begin{bmatrix} \boldsymbol{R}(\varphi) & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix}$$



Basic 2D Rotation

Projection of red y-axis to blue axes

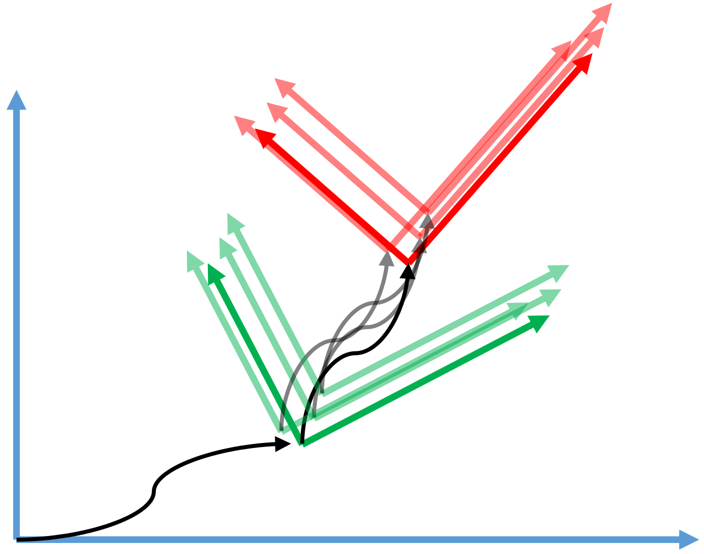


$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

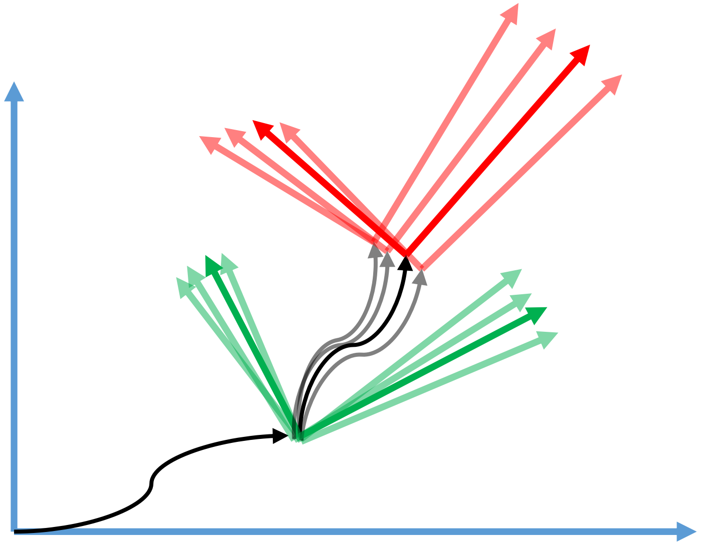
Projection of red x-axis to blue axes

Visualizing uncertainty

- Translation ($\boldsymbol{\Delta p}$)



- Rotation ($\Delta\theta$)



Expand

$$\boldsymbol{T}_{MC} = \boldsymbol{T}_{MR}\boldsymbol{T}_{RC}$$

$$\boldsymbol{T}_{MC} = \begin{bmatrix} \boldsymbol{R}(\theta + \Delta\theta) & \boldsymbol{p} + \boldsymbol{\Delta p} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}(\varphi) & \boldsymbol{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\boldsymbol{T}_{MC} = \begin{bmatrix} \boldsymbol{R}(\theta + \Delta\theta)\boldsymbol{R}(\varphi) & \boldsymbol{R}(\theta + \Delta\theta)\boldsymbol{t} + \boldsymbol{p} + \mathbf{\Delta p} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\boldsymbol{T}_{MC} = \begin{bmatrix} \boldsymbol{R}(\theta + \varphi + \Delta\theta) & \boldsymbol{R}(\theta + \Delta\theta)\boldsymbol{t} + \boldsymbol{p} + \mathbf{\Delta p} \\ \mathbf{0} & 1 \end{bmatrix}$$

Expand

$$\boldsymbol{T}_{MC} = \boldsymbol{T}_{MR}\boldsymbol{T}_{RC}$$

$$\boldsymbol{T}_{MC} = \begin{bmatrix} \boldsymbol{R}(\theta + \Delta\theta) & \boldsymbol{p} + \boldsymbol{\Delta p} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}(\varphi) & \boldsymbol{t} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\boldsymbol{T}_{MC} = \begin{bmatrix} \boldsymbol{R}(\theta + \Delta\theta)\boldsymbol{R}(\varphi) & \boldsymbol{R}(\theta + \Delta\theta)\boldsymbol{t} + \boldsymbol{p} + \mathbf{\Delta p} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\boldsymbol{T}_{MC} = \begin{bmatrix} \boldsymbol{R}(\theta + \varphi + \Delta\theta) & \boldsymbol{R}(\theta + \Delta\theta)\boldsymbol{t} + \boldsymbol{p} + \mathbf{\Delta p} \\ \mathbf{0} & 1 \end{bmatrix}$$

Rotation uncertainty does not change

New term. Forms the “banana” shape

Original position uncertainty

Linearization

$$\boldsymbol{n} = \boldsymbol{R}(\theta + \Delta\theta)\boldsymbol{t}$$

$$\boldsymbol{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} t_x \cos(\theta + \Delta\theta) - t_y \sin(\theta + \Delta\theta) \\ t_x \sin(\theta + \Delta\theta) + t_y \cos(\theta + \Delta\theta) \end{bmatrix}$$

High school trigonometry with approximation $\sin \Delta\theta \approx \Delta\theta$ and $\cos \Delta\theta \approx 1$ for small $\Delta\theta$

$$\boldsymbol{n} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix} - \begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix} \Delta\theta$$

Linearization

$$\boldsymbol{n} = \boldsymbol{R}(\theta + \Delta\theta)\boldsymbol{t}$$

$$\boldsymbol{n} = \begin{bmatrix} n_x \\ n_y \end{bmatrix} = \begin{bmatrix} t_x \cos(\theta + \Delta\theta) - t_y \sin(\theta + \Delta\theta) \\ t_x \sin(\theta + \Delta\theta) + t_y \cos(\theta + \Delta\theta) \end{bmatrix}$$

High school trigonometry with approximation $\sin \Delta\theta \approx \Delta\theta$ and $\cos \Delta\theta \approx 1$ for small $\Delta\theta$

$\boldsymbol{R}(\theta)$

$$\boldsymbol{n} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix} \Delta\theta$$

$\boldsymbol{R}(\theta - \frac{\pi}{2})$

Plug back in

$$\boldsymbol{T}_{MC} = \begin{bmatrix} \boldsymbol{R}(\theta + \varphi + \Delta\theta) & \boldsymbol{R}(\theta)\boldsymbol{t} + \boldsymbol{p} - \boldsymbol{R}\left(\theta - \frac{\pi}{2}\right)\Delta\theta + \mathbf{\Delta p} \\ \mathbf{0} & 1 \end{bmatrix}$$

- Position uncertainty has two components:
  - Original uncertainty characterized by Σ_{xy}
  - Orientation uncertainty characterized by σ_θ^2, projected to x/y axes
- Result:
  - Banana-shaped point-cloud approximated by an ellipse
  - Stretches the original ellipse in tangent direction

Covariance

- Assumption:
  - Independent position and orientation estimates
- Corollary:
  - Covariance terms are additive
- Camera pose and covariance:

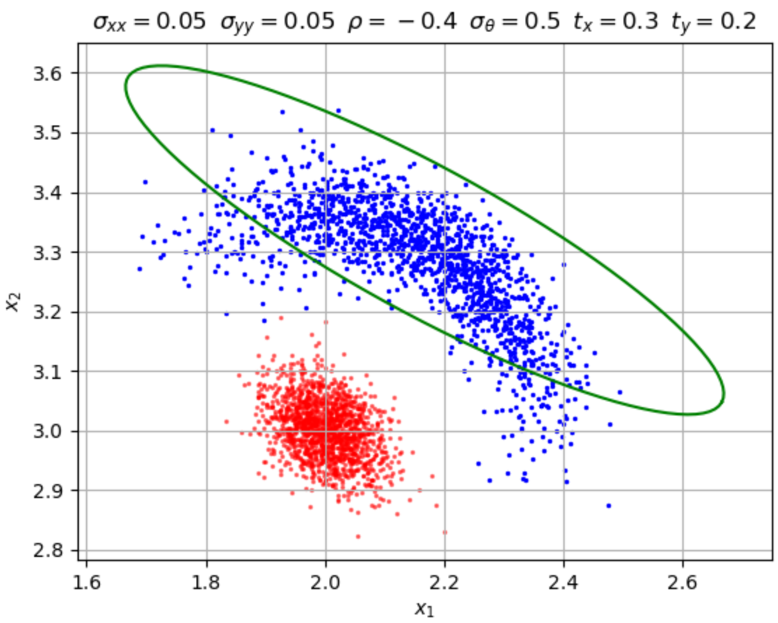
$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{R}(\theta)\boldsymbol{t} + \boldsymbol{p} \\ \theta + \varphi \end{bmatrix}$$

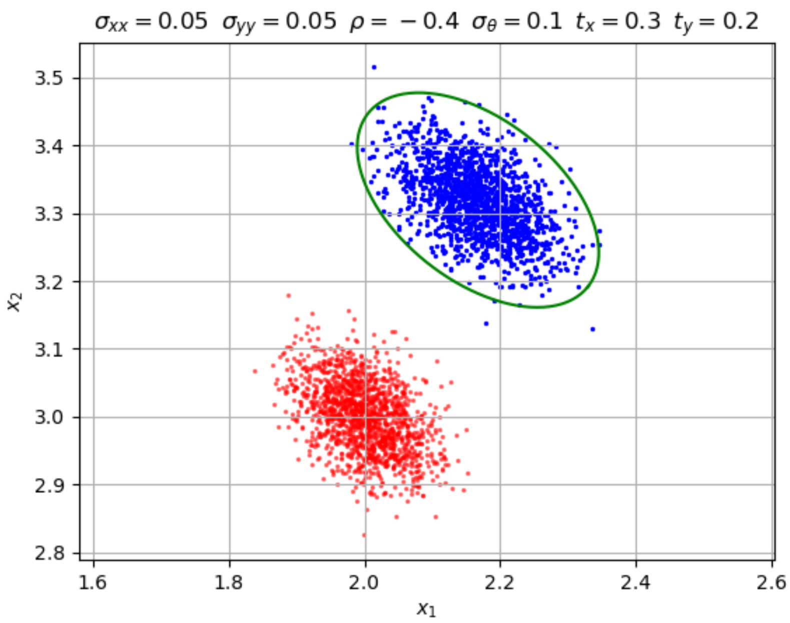
$$\Sigma_Q = \begin{bmatrix} \Sigma_{xy} + \Sigma_n & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}$$

Covariance

$$\Sigma_n = \boldsymbol{R}\left(\theta - \frac{\pi}{2}\right)\boldsymbol{t}\sigma_\theta^2\left(\boldsymbol{R}\left(\theta - \frac{\pi}{2}\right)\boldsymbol{t}\right)^{\mathrm{T}}$$

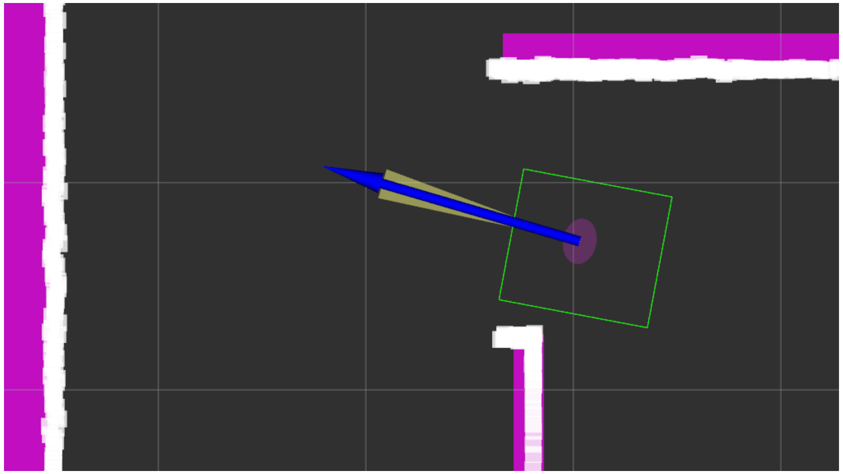
$$\Sigma_n = \boldsymbol{R}\left(\theta - \frac{\pi}{2}\right)\boldsymbol{t}\boldsymbol{t}^T\boldsymbol{R}\left(\frac{\pi}{2} - \theta\right)\sigma_\theta^2$$





Orientation Uncertainty

- One-dimensional in 2D plane (ground robots)
- Yaw, θ, and yaw variance, σ_θ^2
- Gaussian approximation valid for small errors.
- Careful with wraparound.



- In 3D:
  - 3x3 (9D) rotation matrix
  - 4D quaternion
  - 4D axis-angle
  - 3D Euler angles + order convention
- How to represent uncertainty in 3D?

Basic 3D Rotations

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & cos\theta \end{bmatrix}$$

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

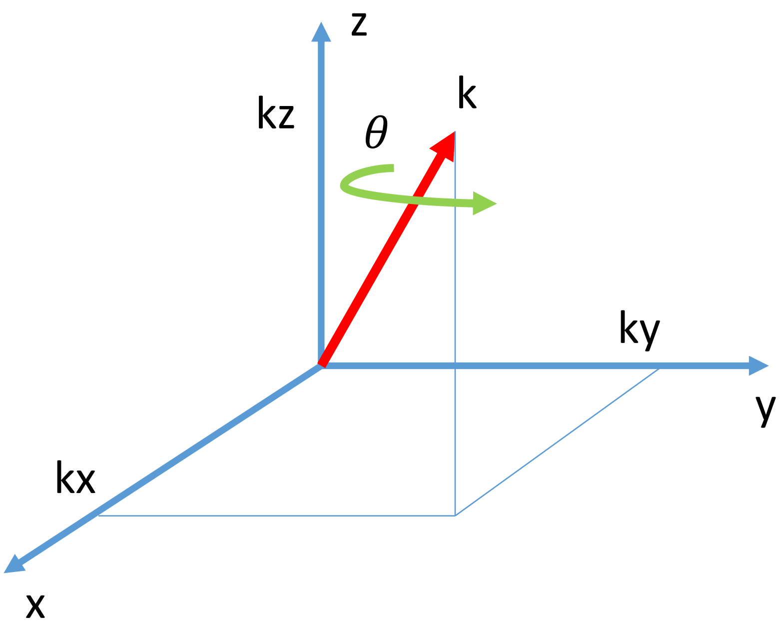
General 3D Rotation

- Straightforward approach:
  - Compose basic rotations: Euler angles, Yaw-Pitch-Roll
  - Order matters
  - Choice of rotating or stationary frame
- A set of rotation matrices and matrix multiplication form a group, that we call **SO(3) group**.

Axis-Angle Representation

- Based on Euler Rotation Theorem

$$\boldsymbol{R} = \begin{bmatrix} k_x^2(1-\cos\theta)+\cos\theta & k_xk_y(1-\cos\theta)-k_z\sin\theta & k_xk_z(1-\cos\theta)+k_y\sin\theta \\ k_xk_y(1-\cos\theta)+k_z\sin\theta & k_y^2(1-\cos\theta)+\cos\theta & k_yk_z(1-\cos\theta)-k_x\sin\theta \\ k_xk_z(1-\cos\theta)-k_y\sin\theta & k_yk_z(1-\cos\theta)+k_x\sin\theta & k_z^2(1-\cos\theta)+\cos\theta \end{bmatrix}$$



$$\theta = \cos^{-1}\left(\frac{r_{11}+r_{22}+r_{33}-1}{2}\right)$$

$$\boldsymbol{k} = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32}-r_{23} \\ r_{13}-r_{31} \\ r_{21}-r_{12} \end{bmatrix}$$

Unit Quaternions

- Algebra on axis-angle representation
- Extension of complex numbers
- Quaternion product is a composition of rotations
- Also forms an SO(3) group

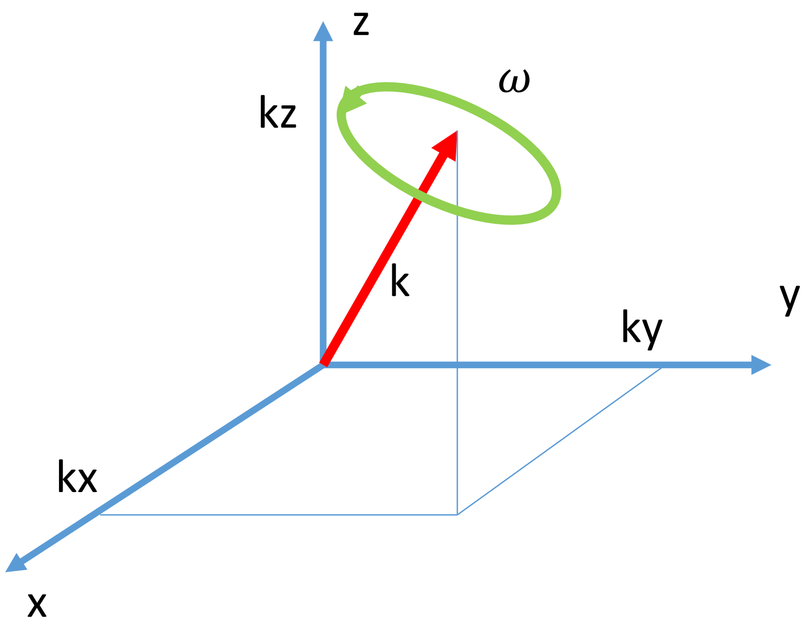
$$\boldsymbol{q} = \cos\frac{\theta}{2} + \sin\frac{\theta}{2}\left(k_x\boldsymbol{i} + k_y\boldsymbol{j} + k_z\boldsymbol{k}\right)$$

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1$$

Properties of Rotation Matrices

- Multiplication is associative, but not commutative
- It is commutative in 2D, SO(2) group
- $\det(\boldsymbol{R}) = 1$ (for right-hand coordinate systems)
- Columns are orthogonal unit vectors
- $\boldsymbol{R}^{-1} = \boldsymbol{R}^{T}$ (follows from orthogonality)

Angular Velocity



First derivative of rotation matrix expressed in terms of angular velocity.

- Which way are the x, y, and z vectors spinning?

$$\dot{\boldsymbol{x}} = \boldsymbol{k}\omega \times \boldsymbol{x}$$

$$\dot{\boldsymbol{y}} = \boldsymbol{k}\omega \times \boldsymbol{y}$$

$$\dot{\boldsymbol{z}} = \boldsymbol{k}\omega \times \boldsymbol{z}$$

- What about projections of the axes of the rotating frame?

$$\boldsymbol{R} = [\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3]$$

$$\dot{\boldsymbol{R}} = [\dot{\boldsymbol{r}}_1 \quad \dot{\boldsymbol{r}}_2 \quad \dot{\boldsymbol{r}}_3]$$

$$\boldsymbol{\omega}_k = \boldsymbol{k}\omega$$

$$\dot{\boldsymbol{R}} = \boldsymbol{\omega}_k \times \boldsymbol{R}$$

Exponential Coordinates

- Get rid of the cross product

$$\dot{\boldsymbol{R}} = \boldsymbol{\omega}_k \times \boldsymbol{R}$$

$$\boxed{\dot{\boldsymbol{R}} = \lfloor \boldsymbol{\omega}_k \rfloor \boldsymbol{R}}$$

$$\lfloor \boldsymbol{\omega}_k \rfloor = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

- Formulate the problem:
  - Frame is rotating at angular velocity ω around $\boldsymbol{k}$.
  - Its initial orientation is described by $\boldsymbol{R}(0)$.
  - What will its orientation $\boldsymbol{R}(t)$ be after time t?
- Answer:
  - Solve the above differential equation!

Exponential Mapping

- Solution to $\dot{\boldsymbol{R}} = \lfloor \boldsymbol{\omega}_k \rfloor \boldsymbol{R}$ is

$$\boldsymbol{R}(t) = e^{\lfloor \boldsymbol{\omega}_k \rfloor t} \boldsymbol{R}(0) = e^{\lfloor \boldsymbol{\theta}_k(t) \rfloor} \boldsymbol{R}(0)$$

  - What is $e^{\lfloor \boldsymbol{\omega}_k \rfloor t}$?
  - Use Taylor expansion
  - Use convenient property $\lfloor \boldsymbol{k} \rfloor^3 = -\lfloor \boldsymbol{k} \rfloor$ ($\boldsymbol{k}$ is unit vector)
  - Get Rodrigues formula:

$$e^{\lfloor \boldsymbol{\theta}_k \rfloor} = I + \lfloor \boldsymbol{k} \rfloor \sin\theta + \lfloor \boldsymbol{k} \rfloor^2 (1 - \cos\theta)$$

Quick Summary

- Frame orientation at time 0 is described by $\boldsymbol{R}(0)$.
- Frame is rotating around axis $\boldsymbol{k}$ in **global** frame at rate ω.
- Vector $\boldsymbol{\omega}_k = \omega\boldsymbol{k}$ is called **exponential coordinates**.
- Why? Because it is used in **exponential mapping** $e^{\lfloor\boldsymbol{\omega}_k\rfloor t}$.
- The operator $\lfloor\boldsymbol{x}\rfloor$ is a **skew-symmetric matrix** of vector x.
- Angular velocity integrates in exponential coordinate space, $\lfloor\boldsymbol{\theta}_k(t)\rfloor = \lfloor\boldsymbol{\omega}_k\rfloor t$, **before** mapping!
- We can use **Rodrigues formula** to calculate the mapping.
- So, at time t, we have $\boldsymbol{R}(t) = e^{\lfloor\boldsymbol{\theta}_k(t)\rfloor}\boldsymbol{R}(0)$.
- Rotation is around $\boldsymbol{k}$ in **fixed frame** so multiplication is on the **left**!

Notes

- Rotation matrices with matrix multiplication form a group called **SO(3) group**.
- Exponential coordinates form an algebra called **Lie so(3) algebra**.
- Exponential mapping relates so(3) with SO(3).
- We can also define matrix logarithm as reverse mapping.
- Special-case commutativity: $\boldsymbol{A}e^{\boldsymbol{A}t} = e^{\boldsymbol{A}t}\boldsymbol{A}$.
- Inversion:$(e^{\lfloor \boldsymbol{\theta}_k \rfloor})^{-1} = e^{\lfloor -\boldsymbol{\theta}_k \rfloor}$.
- Exponential coordinates rotation: $\boldsymbol{R}e^{\lfloor \boldsymbol{\theta}_k \rfloor} = e^{\lfloor \boldsymbol{R}\boldsymbol{\theta}_k \rfloor}\boldsymbol{R}$.

Covariance of Rotations

- What do the rows and columns in covariance matrix represent?

$$\mathbf{\Sigma}_R = \begin{bmatrix} \sigma_x^2 & c_{xy} & c_{xz} \\ c_{xy} & \sigma_y^2 & c_{yz} \\ c_{xz} & c_{yz} & \sigma_z^2 \end{bmatrix}$$

Covariance of Rotations

- What do the rows and columns in covariance matrix represent?

$$\mathbf{\Sigma}_R = \begin{bmatrix} \sigma_x^2 & c_{xy} & c_{xz} \\ c_{xy} & \sigma_y^2 & c_{yz} \\ c_{xz} & c_{yz} & \sigma_z^2 \end{bmatrix}$$

- Variance and covariance of **exponential coordinates** tangential to the rotation.
- We are looking at the point in SO(3) space but expressing its variations in so(3) space!

Drawing Samples in SO(3) Space

- Set the mean to $\boldsymbol{R}$.
- Generate a zero-mean sample $\boldsymbol{\epsilon}$ in so(3) space.
- Exponentially-map to SO(3) space: $e^{\lfloor \boldsymbol{\epsilon} \rfloor}$
- Apply the rotation: $e^{\lfloor \boldsymbol{\epsilon} \rfloor}\boldsymbol{R}$ (if using global frame)
- Exercise:
  - Write a program to draw samples of rotated frame with uncertainty for a given mean rotation and covariance.
  - Convert the matrix to ROS pose at the origin of the global frame, publish the pose, and visualize it in RVIZ.
  - If you prefer not to use ROS, then visualize it in Matlab or using Python Matplotlib

Covariance After Rotation

$$\widetilde{\boldsymbol{R}}_{mb} = \boldsymbol{D}_m \boldsymbol{R}_{mb}$$

Uncertain rotation.

Disturbance (small random rotation).

Ground-truth rotation.

$$\widetilde{\boldsymbol{R}}_{mb} = e^{\lfloor \boldsymbol{\delta} \rfloor} \boldsymbol{R}_{mb}$$

Covariance expresses uncertainty of this!

Covariance After Rotation

$$\widetilde{\boldsymbol{R}}_{mb} = \boldsymbol{D}_m \boldsymbol{R}_{mb}$$

$$\widetilde{\boldsymbol{R}}_{mb} = e^{\lfloor \boldsymbol{\delta} \rfloor} \boldsymbol{R}_{mb}$$

$$\boldsymbol{R}_{wm} \widetilde{\boldsymbol{R}}_{mb} = \boldsymbol{R}_{wm} e^{\lfloor \boldsymbol{\delta} \rfloor} \boldsymbol{R}_{mb}$$

$$\widetilde{\boldsymbol{R}}_{wb} = \boldsymbol{R}_{wm} e^{\lfloor \boldsymbol{\delta} \rfloor} \boldsymbol{R}_{wm}^{-1} \boldsymbol{R}_{wm} \boldsymbol{R}_{mb}$$

$$\widetilde{\boldsymbol{R}}_{wb} = \boldsymbol{R}_{wm} e^{\lfloor \boldsymbol{\delta} \rfloor} \boldsymbol{R}_{wm}^{-1} \boldsymbol{R}_{wb}$$

$$\widetilde{\boldsymbol{R}}_{wb} = e^{\lfloor \boldsymbol{R}_{wm} \boldsymbol{\delta} \rfloor} \boldsymbol{R}_{wb}$$

- Exponential coordinates of the disturbance have been rotated!

Reminder: Covariance Propagation

- Linear Transformation

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{x}$$

$$\boldsymbol{\Sigma}_y = \mathbf{A}\boldsymbol{\Sigma}_x\mathbf{A}^T$$

- General Transformation

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$$

$$\boldsymbol{\Sigma}_y = \mathbf{J}\boldsymbol{\Sigma}_x\mathbf{J}^T$$

$$\mathrm{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_m} \end{bmatrix}$$

Rotation Inversion

$$\widetilde{\boldsymbol{R}}_{mb} = e^{\lfloor \boldsymbol{\delta} \rfloor} \boldsymbol{R}_{mb}$$

$$\widetilde{\boldsymbol{R}}_{mb}^{-1} = \boldsymbol{R}_{mb}^{-1} e^{\lfloor -\boldsymbol{\delta} \rfloor}$$

$$\widetilde{\boldsymbol{R}}_{mb}^{-1} = \boldsymbol{R}_{mb}^{-1} e^{\lfloor -\boldsymbol{\delta} \rfloor} \boldsymbol{R}_{mb} \boldsymbol{R}_{mb}^{-1}$$

$$\widetilde{\boldsymbol{R}}_{mb}^{-1} = e^{\lfloor -\boldsymbol{R}_{mb}^{\boldsymbol{T}} \boldsymbol{\delta} \rfloor} \boldsymbol{R}_{mb}^{-1}$$

- Covariance is quadratic, so negative sign does not change it.
- Hence, $\boldsymbol{\Sigma}_{inv} = \boldsymbol{R}_{mb}^{T} \boldsymbol{\Sigma} \boldsymbol{R}_{mb}$

Summary

- Have uncertain rotation $\boldsymbol{R}_1$ with covariance $\boldsymbol{\Sigma}_1$.
- Apply *deterministic* rotation $\boldsymbol{R}$ in global frame.
- Composite Rotation is $\boldsymbol{R}\boldsymbol{R}_1$ with covariance $\boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T$.
- If rotation is in *local* frame, covariance is unchanged.
- If *both* rotations are uncertain, $(\boldsymbol{R}_1, \boldsymbol{\Sigma}_1)$ and $(\boldsymbol{R}, \boldsymbol{\Sigma})$ and we rotate $\boldsymbol{R}_1$ by $\boldsymbol{R}$ in global frame, the resulting covariance is $\boldsymbol{\Sigma} + \boldsymbol{R}\boldsymbol{\Sigma}_1\boldsymbol{R}^T$.
- Inversion expression looks similar but be aware of subtle differences.

Exercises

- Prove the expression for covariance propagation when both the transformation and rotated frame orientations are uncertain.
- The Attitude Heading Reference System (AHRS) reports the orientation with covariance of the drone in Earth’s East-North-Up magnetic frame. The X-axis of the building map frame forms a 60-degree angle with East-axis of the Earth’s magnetic frame. The AHRS sensor is mounted on the drone such that its Z-axis points down and X-axis aligns with the frame of the vehicle. Derive the covariance transformation for orientation uncertainty of the vehicle frame in the building map frame..

Appendix – 3D Homogeneous Transformations

Rigid Body Motion in 3D Space

- Rotation and translation combined – SE(3) group:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{t}_{3\times1} \\ \boldsymbol{0}_{1\times3} & 1 \end{bmatrix}$$

- Exponential coordinates (twist) – se(3) algebra:

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{v} \end{bmatrix}$$

- Extension of skew-symmetric matrix:

$$\lceil \boldsymbol{v} \rceil = \left\lceil \begin{matrix} \lfloor \boldsymbol{\omega} \rfloor & \boldsymbol{v} \\ 0 & 0 \end{matrix} \right\rceil$$

Adjoint Transformation

- We have the twist in body frame.
- We know the transform between body frame and stationary frame.
- What is the twist in stationary frame?

$$\boldsymbol{T}_{sb} = \begin{bmatrix} \boldsymbol{R}_{sb} & \boldsymbol{t}_{sb} \\ \boldsymbol{0} & 1 \end{bmatrix}$$

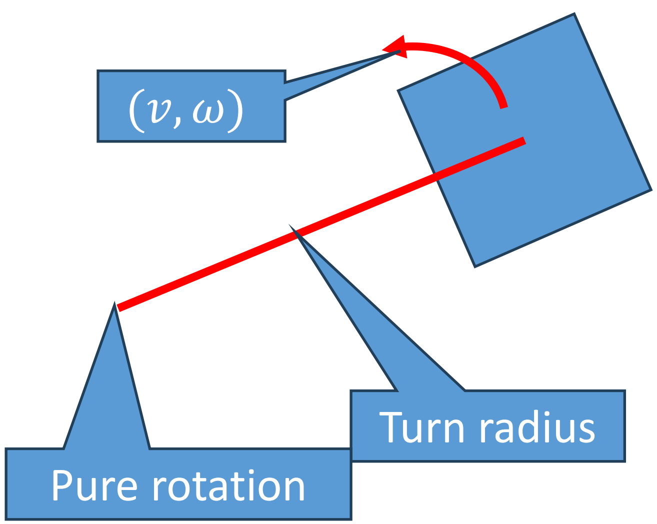
Adjoint: $Adj\boldsymbol{T}_{sb}$

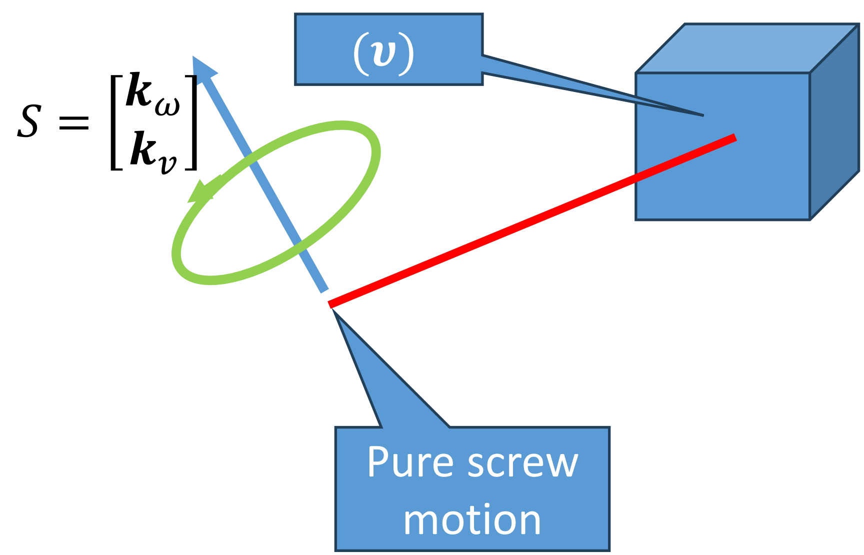
$$\begin{bmatrix} \boldsymbol{\omega}_s \\ \boldsymbol{v}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{sb} & \boldsymbol{0} \\ \lfloor \boldsymbol{t}_{sb} \rfloor \boldsymbol{R}_{sb} & \boldsymbol{R}_{sb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_b \\ \boldsymbol{v}_b \end{bmatrix}$$

Screw Motion Representation

- Chasles-Mozzi Theorem: For every motion in space there exists a frame in which that motion is pure screw-motion

Think 2D first → Think 3D equivalent





Screw Axis – General Case

- Both linear and angular velocity vectors point into the same direction.
- Angular-motion vector is a unit vector.
- Linear-motion vector is not!
- Express screw-displacement in terms of angular displacement θ.

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{k}_\omega \\ \boldsymbol{k}_v \end{bmatrix} = \begin{bmatrix} \dfrac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \\ \dfrac{\boldsymbol{v}}{\|\boldsymbol{\omega}\|} \end{bmatrix}$$

- Normalized $\boldsymbol{k}_v$ is conceptually what turn-radius is in planar motion.

Screw Axis – Exception

- For pure linear motion, angular velocity vector does not exist.
- Normalize linear velocity and call it the screw-axis.
- Displacement along the axis is now linear, but we still call it θ.

$$S = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{k}_v \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \dfrac{\boldsymbol{v}}{||\boldsymbol{v}||} \end{bmatrix}$$

Exponential Mapping

- General case:

$$e^{[\boldsymbol{S}]\theta} = \begin{bmatrix} e^{\lfloor \boldsymbol{k}_\omega \rfloor \theta} & \boldsymbol{G}(\theta)\boldsymbol{k}_v \\ \boldsymbol{0} & 1 \end{bmatrix}$$

$$\boldsymbol{G}(\theta) = \boldsymbol{I}\theta + (1 - \cos\theta)\lfloor \boldsymbol{k}_\omega \rfloor + (\theta - \sin\theta)\lfloor \boldsymbol{k}_\omega \rfloor^2$$

- Exception – Linear Motion:

$$e^{[\boldsymbol{S}]\theta} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{k}_v\theta \\ \boldsymbol{0} & 1 \end{bmatrix}$$

Notes

- Homogeneous transformation matrices with matrix multiplication form a group called **SE(3) group**.
- Exponential coordinates form an algebra called **Lie se(3) algebra**.
- Exponential coordinates are called twist.
- Twist is a 6x1 vector, adjoint is a 6x6 matrix, transformation is 4x4.
- Inversion:$(e^{[\boldsymbol{\theta}_k]})^{-1} = e^{[-\boldsymbol{\theta}_k]}$.
- Composition: $e^{[\boldsymbol{\theta}_1]}e^{[\boldsymbol{\theta}_2]} = e^{[\boldsymbol{\theta}_1+\boldsymbol{\theta}_2]}$
- Twist transformation: $\boldsymbol{T}e^{[\boldsymbol{\theta}_k]} = e^{[Adj\boldsymbol{T}\boldsymbol{\theta}_k]}\boldsymbol{T}$.

Covariance Propagation

$$\widetilde{\boldsymbol{T}}_{mb} = \boldsymbol{D}_m \boldsymbol{T}_{mb}$$

$$\widetilde{\boldsymbol{T}}_{mb} = e^{\lfloor \boldsymbol{\delta} \rfloor} \boldsymbol{T}_{mb}$$

$$\boldsymbol{T}_{wm} \widetilde{\boldsymbol{T}}_{mb} = \boldsymbol{T}_{wm} e^{\lceil \boldsymbol{\delta} \rceil} \boldsymbol{T}_{mb}$$

$$\widetilde{\boldsymbol{T}}_{wb} = \boldsymbol{T}_{wm} e^{\lceil \boldsymbol{\delta} \rceil} \boldsymbol{T}_{wm}^{-1} \boldsymbol{T}_{wm} \boldsymbol{T}_{mb}$$

$$\widetilde{\boldsymbol{T}}_{wb} = \boldsymbol{T}_{wm} e^{\lceil \boldsymbol{\delta} \rceil} \boldsymbol{T}_{wm}^{-1} \boldsymbol{T}_{wb}$$

$$\widetilde{\boldsymbol{T}}_{wb} = e^{\lceil \boldsymbol{Adj} \boldsymbol{T}_{wm} \boldsymbol{\delta} \rceil} \boldsymbol{T}_{wb}$$

- The 6x6 covariance matrix transformed by adjoint!

$$\boldsymbol{\Sigma}_{wb} = Adj\boldsymbol{T}_{wm} \boldsymbol{\Sigma}_{mb} Adj\boldsymbol{T}_{wm}^{T}$$

Covariance of Inversion

$$\widetilde{\boldsymbol{T}}_{mb} = e^{\lceil \boldsymbol{\delta} \rceil} \boldsymbol{T}_{mb}$$

$$\widetilde{\boldsymbol{T}}_{mb}^{-1} = \boldsymbol{T}_{mb}^{-1} e^{\lceil -\boldsymbol{\delta} \rceil}$$

$$\widetilde{\boldsymbol{T}}_{mb}^{-1} = \boldsymbol{T}_{mb}^{-1} e^{\lceil -\boldsymbol{\delta} \rceil} \boldsymbol{T}_{mb} \boldsymbol{T}_{mb}^{-1}$$

$$\widetilde{\boldsymbol{T}}_{mb}^{-1} = e^{\lfloor -\boldsymbol{AdjT}_{mb}^{-1} \boldsymbol{\delta} \rfloor} \boldsymbol{T}_{mb}^{-1}$$

- Hence, $\boldsymbol{\Sigma}_{inv} = \boldsymbol{AdjT}_{mb}^{-1} \boldsymbol{\Sigma} \boldsymbol{AdjT}_{mb}^{-1}{}^{T}$
- Note: Adjoint of an inverse equals inverse of adjoint.

Composition of Uncertain Poses – Global Frame

$$\widetilde{\mathbf{T}} = \widetilde{\boldsymbol{T}}_1 \widetilde{\boldsymbol{T}}_{\mathbf{2}} = e^{\lceil \boldsymbol{\tau}_1 \rceil} \boldsymbol{T}_1 e^{\lceil \boldsymbol{\tau}_2 \rceil} \boldsymbol{T}_2$$

$$\widetilde{\mathbf{T}} = e^{\lceil \boldsymbol{\tau}_1 \rceil} \boldsymbol{T}_1 e^{\lceil \boldsymbol{\tau}_2 \rceil} \boldsymbol{T}_1^{-1} \boldsymbol{T}_1 \boldsymbol{T}_2$$

$$\widetilde{\mathbf{T}} = e^{\lceil \boldsymbol{\tau}_1 \rceil} \boldsymbol{T}_1 e^{\lceil \boldsymbol{\tau}_2 \rceil} \boldsymbol{T}_1^{-1} \boldsymbol{T}$$

$$\widetilde{\mathbf{T}} = e^{\lceil \boldsymbol{\tau}_1 \rceil} e^{\lceil Adj \boldsymbol{T}_1 \boldsymbol{\tau}_2 \rceil} \boldsymbol{T}$$

$$\widetilde{\mathbf{T}} = e^{\lceil \boldsymbol{\tau}_1 + Adj \boldsymbol{T}_1 \boldsymbol{\tau}_2 \rceil} \boldsymbol{T}$$

- Quadratic term with adjoint:

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 + Adj \boldsymbol{T}_1 \boldsymbol{\Sigma}_2 (Adj \boldsymbol{T}_1)^T$$

Composition of Uncertain Poses – Local Frame

$$\tilde{\mathbf{T}} = \tilde{\boldsymbol{T}}_1 \tilde{\boldsymbol{T}}_{\mathbf{2}} = \boldsymbol{T}_1 e^{\lceil \boldsymbol{\tau}_1 \rceil} \boldsymbol{T}_2 e^{\lceil \boldsymbol{\tau}_2 \rceil}$$

$$\tilde{\mathbf{T}} = \boldsymbol{T}_1 \boldsymbol{T}_2 \boldsymbol{T}_2^{-1} e^{\lceil \boldsymbol{\tau}_1 \rceil} \boldsymbol{T}_2 e^{\lceil \boldsymbol{\tau}_2 \rceil}$$

$$\tilde{\mathbf{T}} = \boldsymbol{T} \boldsymbol{T}_2^{-1} e^{\lceil \boldsymbol{\tau}_1 \rceil} \boldsymbol{T}_2 e^{\lceil \boldsymbol{\tau}_2 \rceil}$$

$$\tilde{\mathbf{T}} = \boldsymbol{T} e^{\lceil Adj \boldsymbol{T}_2^{-1} \boldsymbol{\tau}_1 \rceil} e^{\lceil \boldsymbol{\tau}_2 \rceil}$$

$$\tilde{\mathbf{T}} = \boldsymbol{T} e^{\lceil Adj \boldsymbol{T}_2^{-1} \boldsymbol{\tau}_1 + \boldsymbol{\tau}_2 \rceil}$$

- Quadratic term with adjoint:

$$\boldsymbol{\Sigma} = Adj \boldsymbol{T}_2^{-1} \boldsymbol{\Sigma}_1 (Adj \boldsymbol{T}_2^{-1})^T + \boldsymbol{\Sigma}_2$$

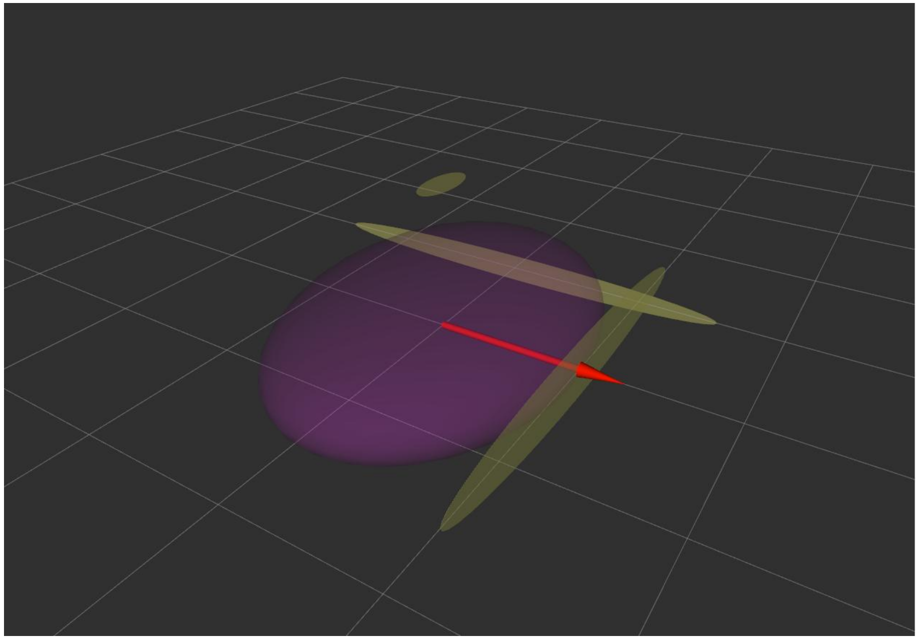
In Practice: ROS Pose With Covariance

- ROS PoseWithCovarianceStamped message type:
  - Timestamp.
  - Frame of reference name.
  - Position (Euclidean vector).
  - Orientation (Quaternion).
  - Covariance (6x6 matrix represented by 36-element list, in row-major order).
- Convention for covariance:
  - Origin of disturbances is always the local frame.
  - Rotational disturbance is in local frame.
  - Positional disturbance principal axes are parallel to the reference frame axes.

Visualizing Covariance

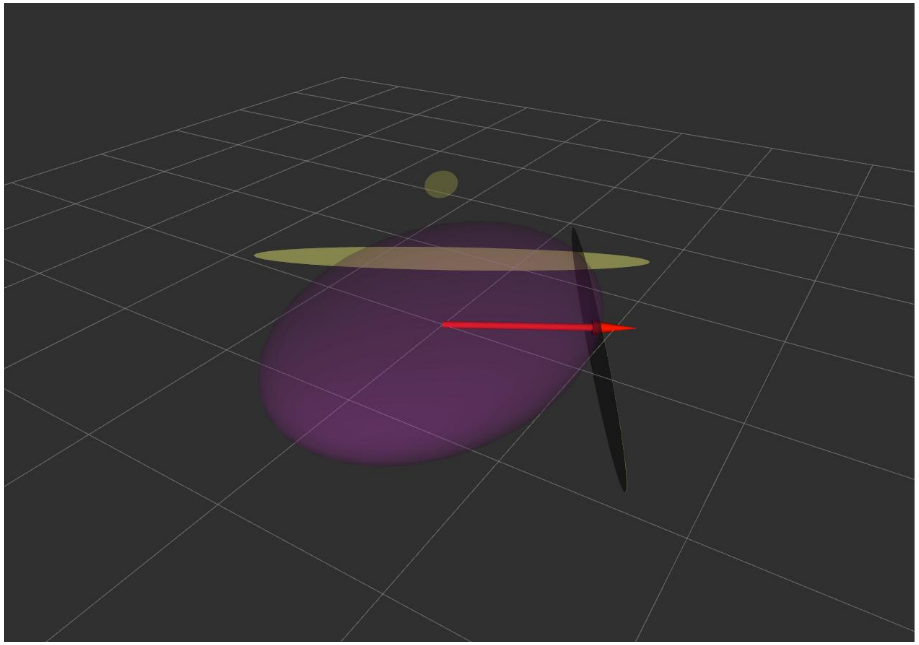
$p = (0,0,0)$
$q = 0i + 0j + 0k + 1$

$$\Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Visualizing Covariance

$$p = (0, 0, 0)$$
$$q = 0i + 0j + 0.259k + 0.966$$
$$\Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Pose Composition, Revisited

- Rotate position covariance to local frame.
- Compose covariance in local frame.
- Rotate position covariance back to the reference frame.

Pose Composition, Revisited

- Covariance used by ROS:

$$\Sigma^g = \begin{bmatrix} \boldsymbol{\Sigma}_p^g & 0 \\ 0 & \boldsymbol{\Sigma}_q^l \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}\Sigma_p^l\boldsymbol{R}^T & 0 \\ 0 & \boldsymbol{\Sigma}_q^l \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Sigma}_p^l & 0 \\ 0 & \boldsymbol{\Sigma}_q^l \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}^T$$

- Covariance all in local frame:

$$\Sigma^l = \begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Sigma}_p^g & 0 \\ 0 & \boldsymbol{\Sigma}_q^l \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}$$

Pose Composition, Revisited

- Start from the result for local frame:

$$\boldsymbol{\Sigma}^l = Adj\boldsymbol{T}_2^{-1}\boldsymbol{\Sigma}_1^l(Adj\boldsymbol{T}_2^{-1})^T + \boldsymbol{\Sigma}_2^l$$

- Insert transforms:

$$\boldsymbol{\Sigma}^g = \begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix} \left(Adj\boldsymbol{T}_2^{-1}\boldsymbol{\Sigma}_1^l(Adj\boldsymbol{T}_2^{-1})^T + \boldsymbol{\Sigma}_2^l\right) \begin{bmatrix} \boldsymbol{R} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}^T$$

$$\boldsymbol{\Sigma}_1^l = \begin{bmatrix} \boldsymbol{R}_1 & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}^T \boldsymbol{\Sigma}_1^g \begin{bmatrix} \boldsymbol{R}_1 & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}$$

$$\boldsymbol{\Sigma}_2^l = \begin{bmatrix} \boldsymbol{R}_2 & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}^T \boldsymbol{\Sigma}_2^g \begin{bmatrix} \boldsymbol{R}_2 & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}$$