M3228.000300

SLAM 101

Lecture 04 Uncertainty Propagation

Ayoung Kim

Uncertainty Propagation

- Vector space
  - Uncertainty as a Gaussian
  - Motion/measurement model as a system
  - Linearize if needed
  - Gaussian input/output via linear system

$$X = [x, y, z, r, p, h]$$

→ Linearized vector representation

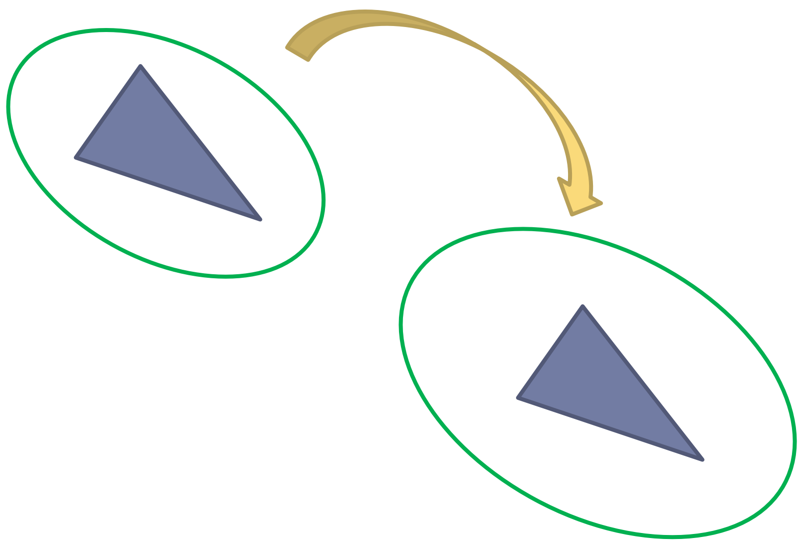
- Manifold
  - Project onto locally linear space
  - Rotation (*not* linear)

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

→ Nonlinear SE(3) representation

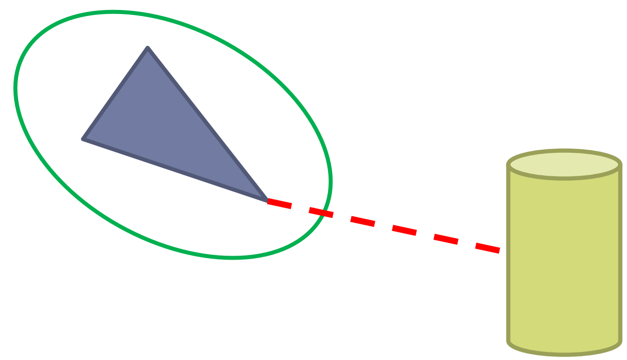
Uncertainty in Motion and Sensing

- Uncertainty after motion



  - Control error
  - Motion uncertainty

- Uncertainty after sensing

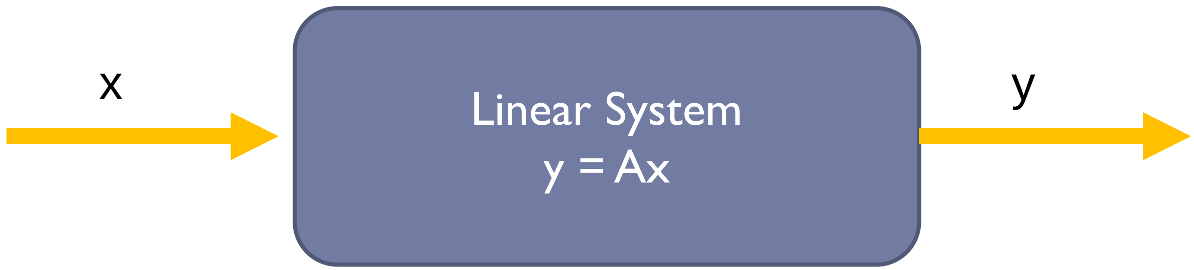


  - Sensing error
  - Measurement uncertainty

Motion & Sensing = Transformation = System = in/out

Gaussian Propagation via Linear System

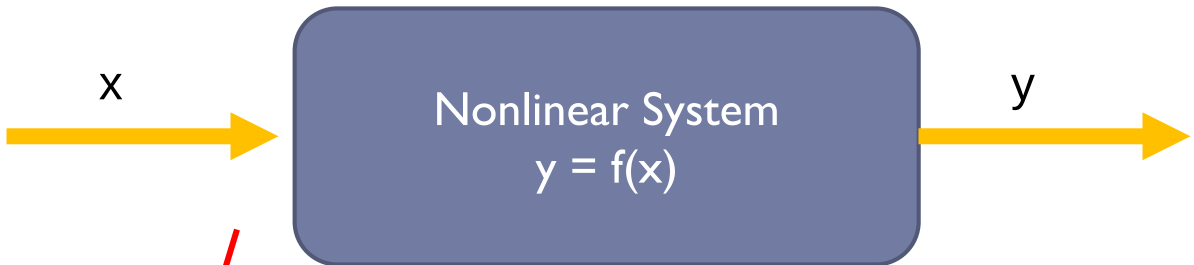
- Random variable x is Gaussian $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$
  - How about y ?

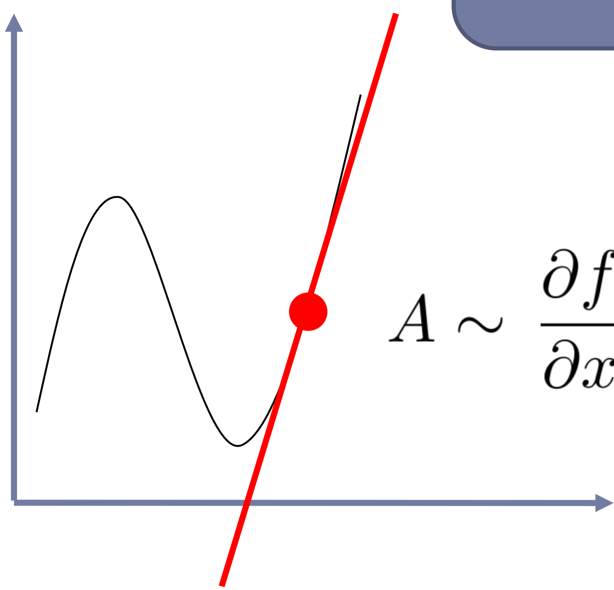


$$\begin{aligned} \mu_y &= A\mu_x \\ \sigma_y^2 &= A\sigma_x^2 A^\top \end{aligned}$$

Gaussian Propagation via Linear System

- Random variable x is Gaussian $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$
  - How about y for nonlinear system?





$$A \sim \left.\frac{\partial f}{\partial x}\right|_{x_0}$$

$$\begin{aligned} \mu_y &= f(\mu_x) \\ \sigma_y^2 &= \left.\frac{\partial f}{\partial x}\right|_{x_0} \sigma_x^2 \left.\frac{\partial f}{\partial x}\right|_{x_0}^\top \end{aligned}$$

Transformation as a System

(Oplus and ominus for Vector Space)

Uncertainty Propagation

- Motion/sensing transformation
  - Vector space examples
  - Oplus/ominus operation

Estimating Uncertain Spatial Relationships in Robotics[*]

Randall Smith[†] Matthew Self[‡] Peter Cheeseman[§]

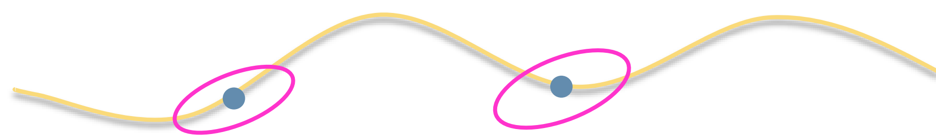
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

In this paper, we describe a representation for spatial information, called the *stochastic map*, and associated procedures for building it, reading information from it, and revising it incrementally as new information is obtained. The map contains the estimates of relationships among objects in the map, and their uncertainties, given all the available information. The procedures provide a general solution to the problem of estimating uncertain relative spatial relationships. The estimates are probabilistic in nature, an advance over the previous, very conservative, worst-case approaches to the problem. Finally, the procedures are developed in the context of state-estimation and filtering theory, which provides a solid basis for numerous extensions.

Mobile Robot – Uncertainty Propagation

- Uncertainty propagation
  - Head-to-tail operation
  - Tail-to-tail operation
  - "oplus" or "ominus" operation

  - pose = [x,y,z,r,p,h] (roll pitch heading)

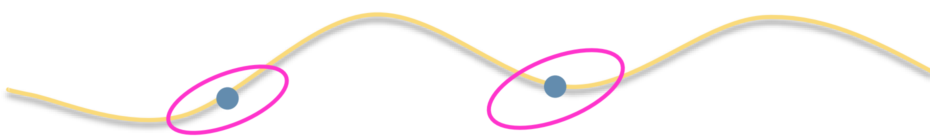


$$\mathrm{x}_{02} = \mathrm{x}_{01} \oplus u_{12}$$

Operator – Oplus & Ominus

- Propagation
  - oplus

$$\mathbf{x}_{ik} = \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} = \begin{bmatrix} x_{jk}\cos\phi_{ij} - y_{jk}\sin\phi_{ij} + x_{ij} \\ x_{jk}\sin\phi_{ij} + y_{jk}\cos\phi_{ij} + y_{ij} \\ \phi_{ij} + \phi_{jk} \end{bmatrix}$$



  - Matrix multiplication in SE(2) and SE(3)

$$T_{ik} = T_{ij}T_{jk} \in SE(3)$$

Operator – Oplus & Ominus

- Inverse
  - ominus

$$\mathbf{x}_{ji} \quad = \quad \ominus \mathbf{x}_{ij} = \begin{bmatrix} -x_{ij} \cos \phi_{ij} - y_{ij} \sin \phi_{ij} \\ x_{ij} \sin \phi_{ij} - y_{ij} \cos \phi_{ij} \\ -\phi_{ij} \end{bmatrix}$$

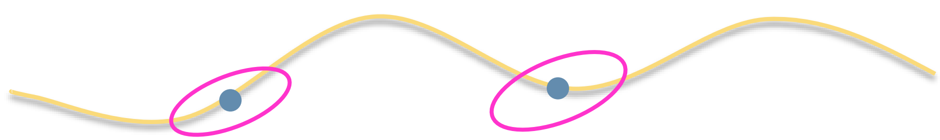
  - Matrix inversion in SE(2) and SE(3)

$$T_{ji} = T_{ij}^{-1} \in SE(3)$$

Mobile Robot – Head to Tail

- 2D case: head to tail operation

$$\mathbf{x}_{ik} \quad = \quad \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} = \begin{bmatrix} x_{jk}\cos\phi_{ij} - y_{jk}\sin\phi_{ij} + x_{ij} \\ x_{jk}\sin\phi_{ij} + y_{jk}\cos\phi_{ij} + y_{ij} \\ \phi_{ij} + \phi_{jk} \end{bmatrix}$$



$$\begin{bmatrix} x_{ik} \\ y_{ik} \\ \theta_{ik} \end{bmatrix} = fcns(\begin{bmatrix} x_{ij} \\ y_{ij} \\ \theta_{ij} \\ x_{jk} \\ y_{jk} \\ \theta_{jk} \end{bmatrix})$$

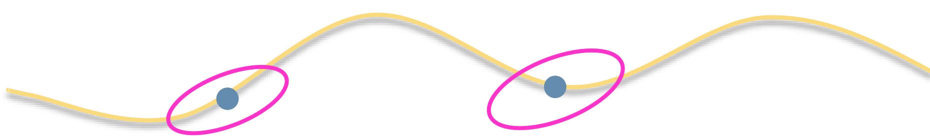
$$\begin{aligned} \mu_y &= A\mu_x + b \\ \Sigma_y &= A\Sigma A^\top \end{aligned}$$

$$\begin{bmatrix} x_{ik} \\ y_{ik} \\ \theta_{ik} \end{bmatrix} = A \begin{bmatrix} x_{ij} \\ y_{ij} \\ \theta_{ij} \\ x_{jk} \\ y_{jk} \\ \theta_{jk} \end{bmatrix}$$

Mobile Robot – Propagation

- 2D case: head to tail operation

$$\mathbf{x}_{ik} = \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} = \begin{bmatrix} x_{jk}\cos\phi_{ij} - y_{jk}\sin\phi_{ij} + x_{ij} \\ x_{jk}\sin\phi_{ij} + y_{jk}\cos\phi_{ij} + y_{ij} \\ \phi_{ij} + \phi_{jk} \end{bmatrix}$$



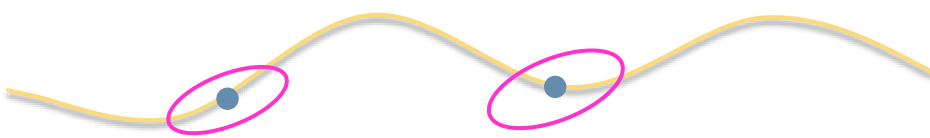
$$J_{\oplus} \qquad \begin{aligned} \mu_y &= A\mu_x + b \\ \Sigma_y &= A\Sigma A^{\top} \end{aligned}$$

$$\begin{bmatrix} x_{ik} \\ y_{ik} \\ \theta_{ik} \end{bmatrix} = fcns(\begin{bmatrix} x_{ij} \\ y_{ij} \\ \theta_{ij} \\ x_{jk} \\ y_{jk} \\ \theta_{jk} \end{bmatrix}) \qquad \begin{bmatrix} x_{ik} \\ y_{ik} \\ \theta_{ik} \end{bmatrix} = A \begin{bmatrix} x_{ij} \\ y_{ij} \\ \theta_{ij} \\ x_{jk} \\ y_{jk} \\ \theta_{jk} \end{bmatrix} \qquad J_{\oplus}$$

Mobile Robot – Propagation

- 2D case: head to tail operation

$$\mathbf{x}_{ik} = \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} = \begin{bmatrix} x_{jk}\cos\phi_{ij} - y_{jk}\sin\phi_{ij} + x_{ij} \\ x_{jk}\sin\phi_{ij} + y_{jk}\cos\phi_{ij} + y_{ij} \\ \phi_{ij} + \phi_{jk} \end{bmatrix}$$



$$J_{\oplus} \qquad \begin{aligned} \mu_y &= A\mu_x + b \\ \Sigma_y &= A\Sigma A^{\top} \end{aligned}$$

$$\mathbf{C}(\mathbf{x}_{ik}) \approx \mathbf{J}_{\oplus} \begin{bmatrix} \mathbf{C}(\mathbf{x}_{ij}) & \mathbf{C}(\mathbf{x}_{ij}, \mathbf{x}_{jk}) \\ \mathbf{C}(\mathbf{x}_{jk}, \mathbf{x}_{ij}) & \mathbf{C}(\mathbf{x}_{jk}) \end{bmatrix} \mathbf{J}_{\oplus}^{T}$$

Mobile Robot – Propagation

- 2D case: head to tail operation

$$\mathbf{x}_{ik} = \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} = \begin{bmatrix} x_{jk}\cos\phi_{ij} - y_{jk}\sin\phi_{ij} + x_{ij} \\ x_{jk}\sin\phi_{ij} + y_{jk}\cos\phi_{ij} + y_{ij} \\ \phi_{ij} + \phi_{jk} \end{bmatrix}$$

$$\mathbf{C}(\mathbf{x}_{ik}) \approx \mathbf{J}_{\oplus} \begin{bmatrix} \mathbf{C}(\mathbf{x}_{ij}) & \mathbf{C}(\mathbf{x}_{ij}, \mathbf{x}_{jk}) \\ \mathbf{C}(\mathbf{x}_{jk}, \mathbf{x}_{ij}) & \mathbf{C}(\mathbf{x}_{jk}) \end{bmatrix} \mathbf{J}_{\oplus}^{T},$$

  - Jacobian = first derivative

$$\mathbf{J}_{\oplus} \triangleq \frac{\partial(\mathbf{x}_{ij} \oplus \mathbf{x}_{jk})}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{jk})} = \frac{\partial \mathbf{x}_{ik}}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{jk})}$$

$$= \begin{bmatrix} 1 & 0 & -(y_{ik} - y_{ij}) & \cos\phi_{ij} & -\sin\phi_{ij} & 0 \\ 0 & 1 & (x_{ik} - x_{ij}) & \sin\phi_{ij} & \cos\phi_{ij} & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Mobile Robot – Relative Pose

- Relative Pose

  - We know
    - x_01: Pose {1} in global frame {0}
    - x_02: Pose {2} in global frame {0}
  - Relative pose between 1 and 2?

$$x_{12} = \ominus x_{01} \oplus x_{02}$$

    - Head-to-tail operation

Mobile Robot – Relative Pose

- Tail-to-tail operation

$$\hat{\mathbf{x}}_{jk} = \hat{\mathbf{x}}_{ji} \oplus \hat{\mathbf{x}}_{ik} = \ominus\hat{\mathbf{x}}_{ij} \oplus \hat{\mathbf{x}}_{ik}.$$

$$\begin{aligned} \mathbf{C}(\mathbf{x}_{jk}) &\approx \mathbf{J}_{\oplus} \begin{bmatrix} \mathbf{C}(\mathbf{x}_{ji}) & \mathbf{C}(\mathbf{x}_{ji}, \mathbf{x}_{ik}) \\ \mathbf{C}(\mathbf{x}_{ik}, \mathbf{x}_{ji}) & \mathbf{C}(\mathbf{x}_{ik}) \end{bmatrix} \mathbf{J}_{\oplus}^{T} \\ &\approx \mathbf{J}_{\oplus} \begin{bmatrix} \mathbf{J}_{\ominus}\mathbf{C}(\mathbf{x}_{ij})\mathbf{J}_{\ominus}^{T} & \mathbf{J}_{\ominus}\mathbf{C}(\mathbf{x}_{ij}, \mathbf{x}_{ik}) \\ \mathbf{C}(\mathbf{x}_{ik}, \mathbf{x}_{ij})\mathbf{J}_{\ominus}^{T} & \mathbf{C}(\mathbf{x}_{ik}) \end{bmatrix} \mathbf{J}_{\oplus}^{T}. \end{aligned}$$

  - Jacobian built with chain rule

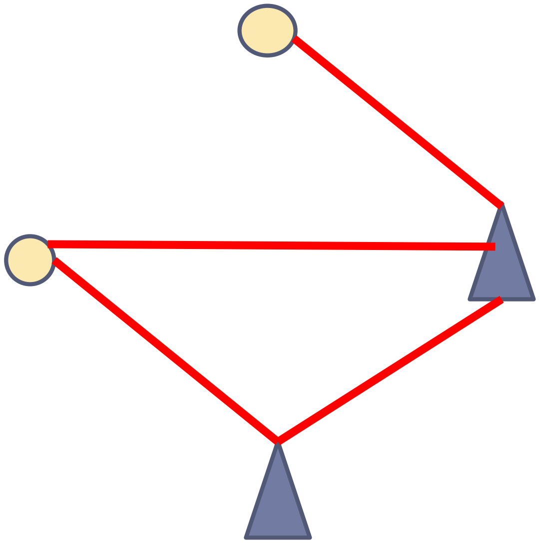
$${}_{\ominus}\mathbf{J}_{\oplus} \triangleq \frac{\partial \mathbf{x}_{jk}}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{ik})} = \frac{\partial \mathbf{x}_{jk}}{\partial(\mathbf{x}_{ji}, \mathbf{x}_{ik})} \frac{\partial(\mathbf{x}_{ji}, \mathbf{x}_{ik})}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{ik})}$$

$$= \mathbf{J}_{\oplus} \begin{bmatrix} \mathbf{J}_{\ominus} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{1\oplus}\mathbf{J}_{\ominus} & \mathbf{J}_{2\oplus} \end{bmatrix}$$

Example

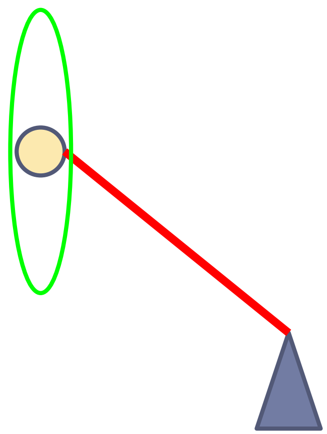
Example

- 1) Robot senses the object #1
- 2) Robot moves
- 3) Robot senses another object #2
- 4) Robot senses the object #1 again



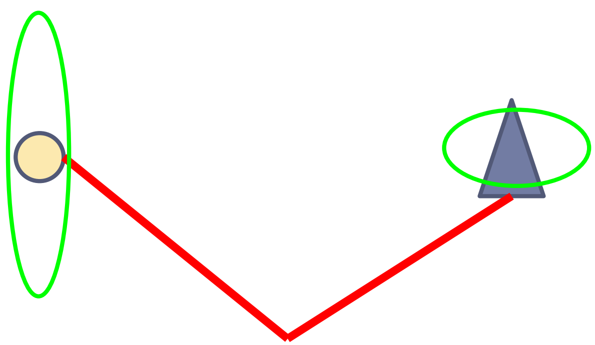
Example

- Robot senses the object #1

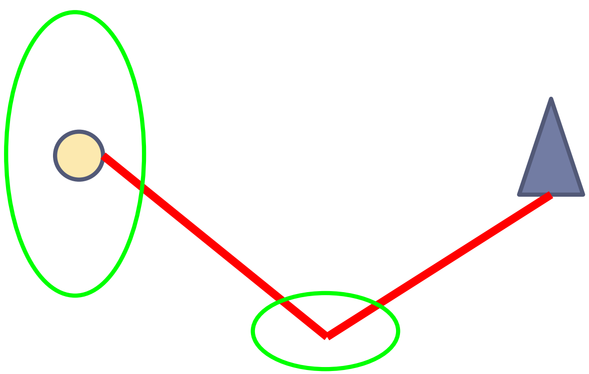


Example

- 2) Robot moves



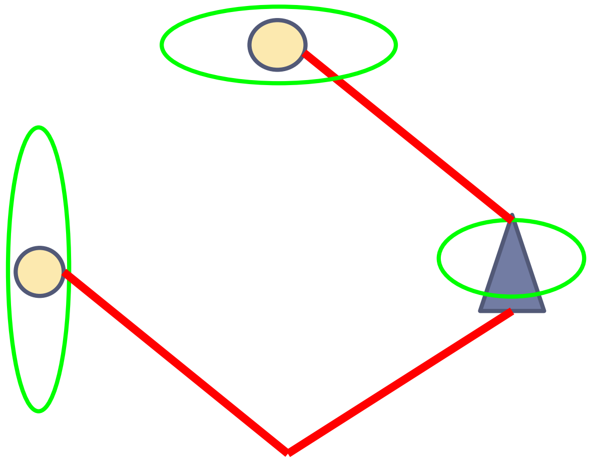
World point of view



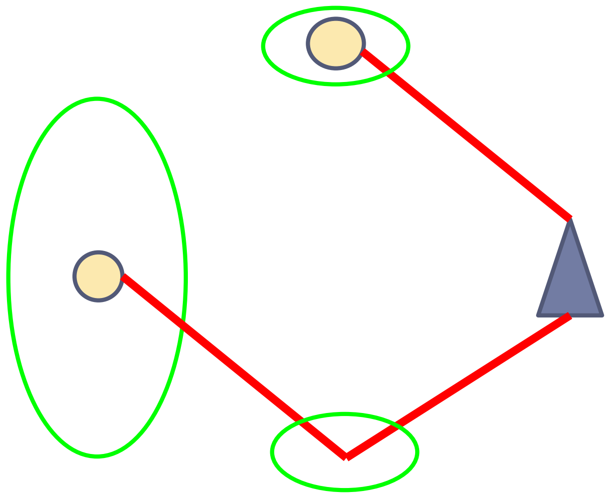
Robot point of view

Example

- 3) Robot senses another object #2



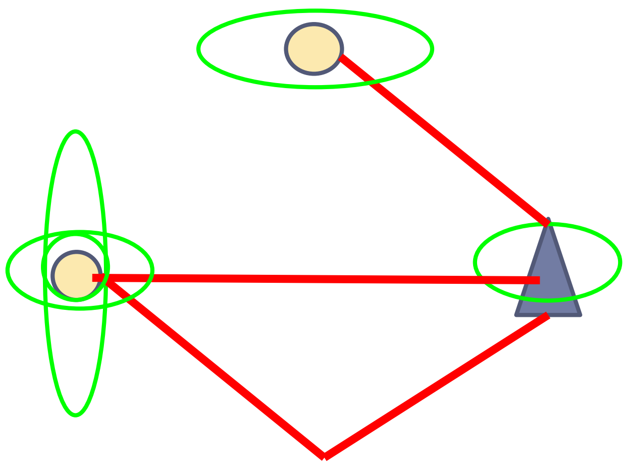
World point of view



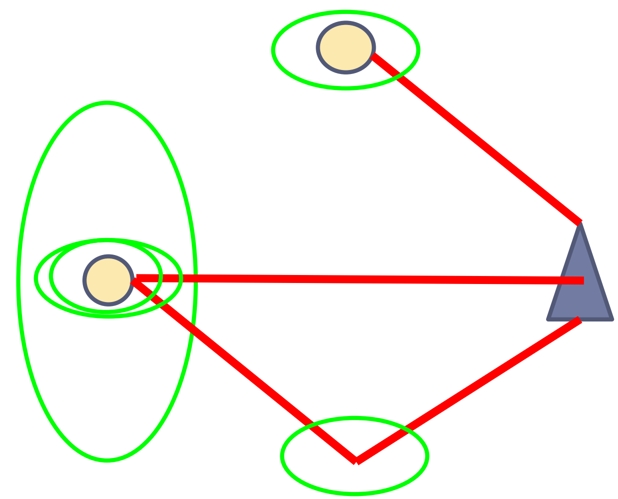
Robot point of view

Example

- 4) Robot senses the object #1 again



World point of view



Robot point of view

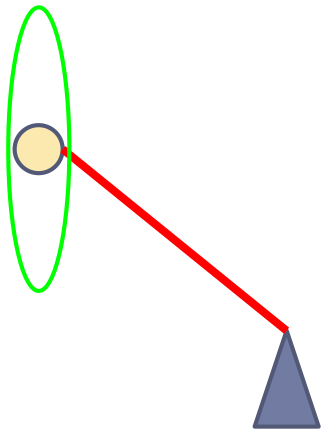
Think About this Example

- Clear?

- Let's implement this in Matlab
  - Q. How many state do we need?
    - Object 1
    - Object 2
    - Robot
      - Only the last pose
      - Entire trajectory?

Example – World Point of View

- Robot senses the object #1



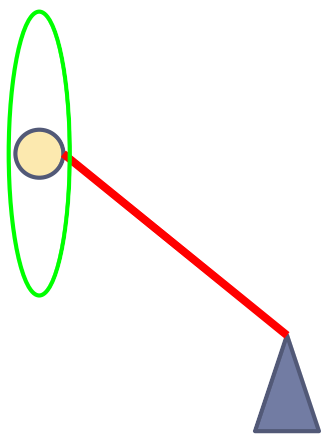
$$\hat{\mathbf{x}} = [\hat{\mathbf{x}}_R] = [\mathbf{0}]$$

$$\mathbf{C}(\mathbf{x}) = [\mathbf{C}(\mathbf{x}_R)] = [\mathbf{0}]$$

```
% robot starts at the origin = world frame
xr = [0 0 0]'; Sr = zeros(3,3);
```

Example – World Point of View

- Robot senses the object #1

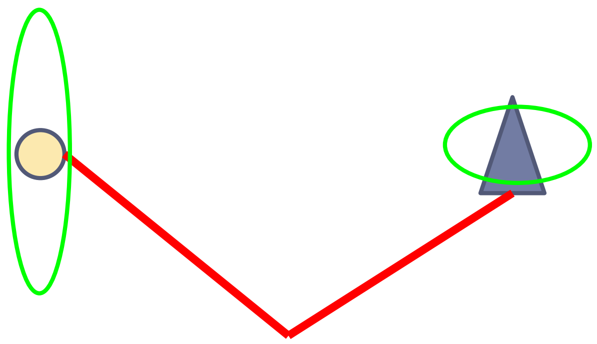


$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_R \\ \hat{\mathbf{x}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \hat{\mathbf{z}}_1 \end{bmatrix}$$

$$\mathbf{C}(\mathbf{x}) = \begin{bmatrix} \mathbf{C}(\mathbf{x}_R) & \mathbf{C}(\mathbf{x}_R, \mathbf{x}_1) \\ \mathbf{C}(\mathbf{x}_1, \mathbf{x}_R) & \mathbf{C}(\mathbf{x}_1) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(\mathbf{z}_1) \end{bmatrix}$$

Example – World Point of View

- 2) Robot moves



World point of view

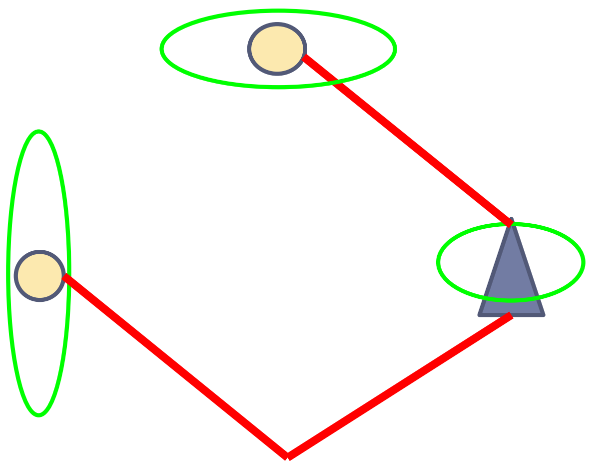
$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_R \\ \hat{\mathbf{x}}_1 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{y}}_R \\ \hat{\mathbf{z}}_1 \end{bmatrix}$$

$$\mathbf{C}(\mathbf{x}) = \begin{bmatrix} \mathbf{C}(\mathbf{x}_R) & \mathbf{C}(\mathbf{x}_R, \mathbf{x}_1) \\ \mathbf{C}(\mathbf{x}_1, \mathbf{x}_R) & \mathbf{C}(\mathbf{x}_1) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{C}(\mathbf{y}_R) & \mathbf{0} \\ \mathbf{0} & \mathbf{C}(\mathbf{z}_1) \end{bmatrix}$$

Example – World Point of View

- 3) Robot senses another object #2



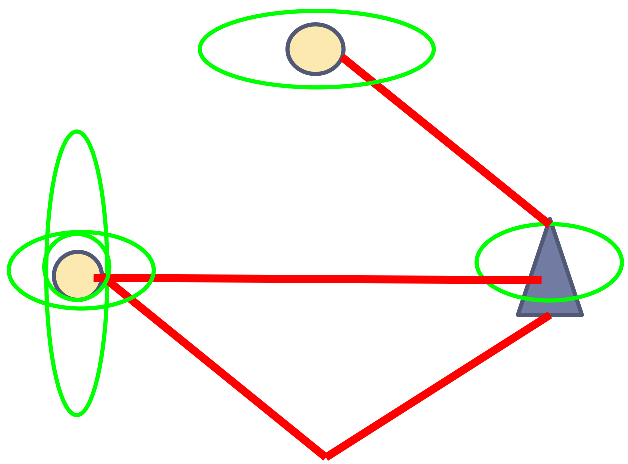
World point of view

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_R \\ \hat{\mathbf{x}}_1 \\ \hat{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{y}}_R \\ \hat{\mathbf{z}}_1 \\ \hat{\mathbf{y}}_R \oplus \hat{\mathbf{z}}_2 \end{bmatrix}$$

$$\begin{aligned} \mathbf{C}(\mathbf{x}) &= \begin{bmatrix} \mathbf{C}(\mathbf{x}_R) & \mathbf{C}(\mathbf{x}_R, \mathbf{x}_1) & \mathbf{C}(\mathbf{x}_R, \mathbf{x}_2) \\ \mathbf{C}(\mathbf{x}_1, \mathbf{x}_R) & \mathbf{C}(\mathbf{x}_1) & \mathbf{C}(\mathbf{x}_1, \mathbf{x}_2) \\ \mathbf{C}(\mathbf{x}_2, \mathbf{x}_R) & \mathbf{C}(\mathbf{x}_2, \mathbf{x}_1) & \mathbf{C}(\mathbf{x}_2) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{C}(\mathbf{y}_R) & \mathbf{0} & \mathbf{C}(\mathbf{y}_R)\mathbf{J}_{1\oplus}^T \\ \mathbf{0} & \mathbf{C}(\mathbf{z}_1) & \mathbf{0} \\ \mathbf{J}_{1\oplus}\mathbf{C}(\mathbf{y}_R) & \mathbf{0} & \mathbf{C}(\mathbf{x}_2) \end{bmatrix}. \end{aligned}$$

Example – World Point of View

- 4) Robot senses the object #1 again



World point of view

Cannot do this right now…
But in 2~3 lectures !!

- What about 3D?
  - [x,y,z,r,p,h]

- What about nonlinearity?
  - y = Ax no longer available just f(x)

3D Transformation

- Propagation and relative pose
  - What about 3D? (See the rsmith-1990a appendix)

$$
\begin{aligned}
\mathbf{x}_{ik} &= \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} \\
&= [x_{ik}, y_{ik}, z_{ik}, \phi_{ik}, \theta_{ik}, \psi_{ik}]^\top \\
&= \begin{bmatrix}
{}^{i}_{j}\mathrm{R} \begin{bmatrix} x_{jk} \\ y_{jk} \\ z_{jk} \end{bmatrix} + \begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix} \\
\operatorname{atan2}\left({}^{i}_{k}\mathrm{R}_{1,3} \sin\psi_{ik} - {}^{i}_{k}\mathrm{R}_{2,3} \cos\psi_{ik}, \quad -{}^{i}_{k}\mathrm{R}_{1,2} \sin\psi_{ik} + {}^{i}_{k}\mathrm{R}_{2,2} \cos\psi_{ik}\right) \\
\operatorname{atan2}\left(-{}^{i}_{k}\mathrm{R}_{3,1}, \quad {}^{i}_{k}\mathrm{R}_{1,1} \cos\psi_{ik} + {}^{i}_{k}\mathrm{R}_{2,1} \sin\psi_{ik}\right) \\
\operatorname{atan2}\left({}^{i}_{k}\mathrm{R}_{2,1}, \quad {}^{i}_{k}\mathrm{R}_{1,1}\right)
\end{bmatrix}
\end{aligned}
$$

(rsmith-199a) Estimating Uncertain Spatial Relationships in Robotics

3D Transformation

- Propagation and relative pose
  - What about 3D?
    (See the rsmith-1990a appendix)

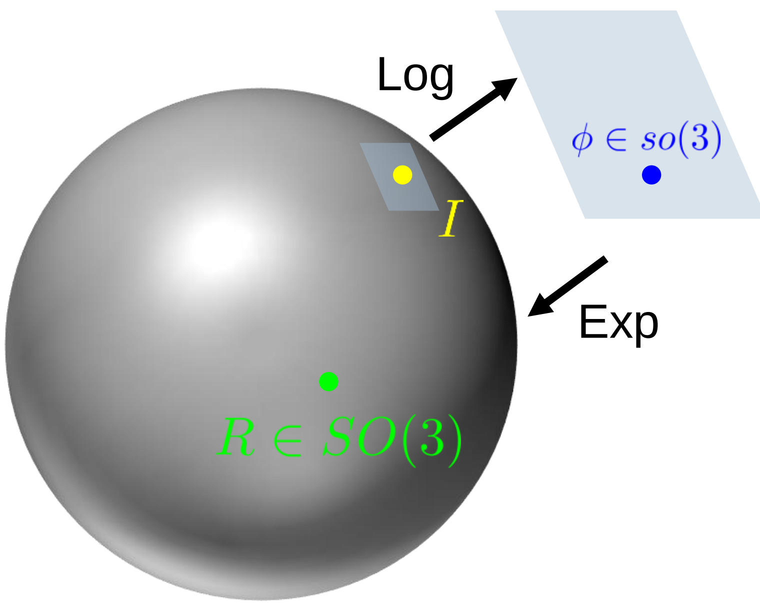
$$
\begin{aligned}
J_{\oplus} &= \frac{\partial \mathbf{x}_{ik}}{\partial(\mathbf{x}_{ij}, \mathbf{x}_{jk})} \\
&= \begin{bmatrix} J_{\oplus 1} & J_{\oplus 2} \end{bmatrix} \\
&= \left[\begin{array}{cc|cc} I_{3\times 3} & M & {}^i_j R & 0_{3\times 3} \\ 0_{3\times 3} & K_1 & 0_{3\times 3} & K_2 \end{array}\right]
\end{aligned}
$$

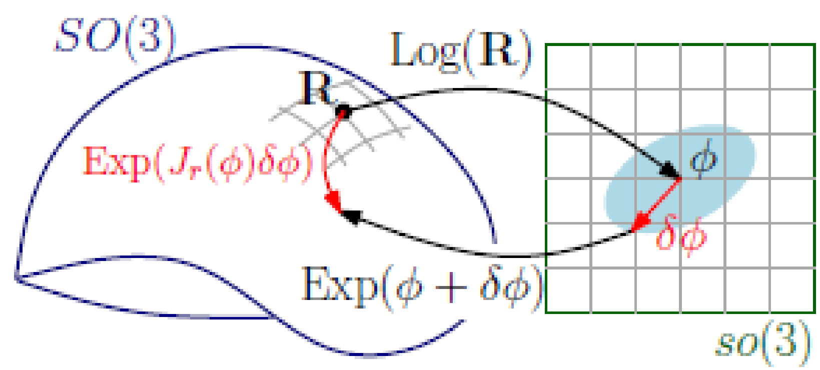
$$
\begin{aligned}
M &= \begin{bmatrix}
{}^i_j R_{1,3}\, y_{jk} - {}^i_j R_{1,2}\, z_{jk} & (z_{ik} - z_{ij}) \cos\psi_{ij} & -(y_{ik} - y_{ij}) \\
{}^i_j R_{2,3}\, y_{jk} - {}^i_j R_{2,2}\, z_{jk} & (z_{ik} - z_{ij}) \sin\psi_{ij} & (x_{ik} - x_{ij}) \\
{}^i_j R_{3,3}\, y_{jk} - {}^i_j R_{3,2}\, z_{jk} & -x_{jk} \cos\theta_{ij} - \left(y_{jk} \sin\phi_{ij} + z_{jk} \cos\phi_{ij}\right) \sin\theta_{ij} & 0
\end{bmatrix} \\
K_1 &= \begin{bmatrix}
\cos\theta_{ij} \cos(\psi_{ik} - \psi_{ij}) \sec\theta_{ik} & \sin(\psi_{ik} - \psi_{ij}) \sec\theta_{ik} & 0 \\
-\cos\theta_{ij} \sin(\psi_{ik} - \psi_{ij}) & \cos(\psi_{ik} - \psi_{ij}) & 0 \\
{}^j_k R_{1,2} \sin\phi_{ik} + {}^j_k R_{1,3} \cos\phi_{ik} \sec\theta_{ik} & \sin(\psi_{ik} - \psi_{ij}) \tan\theta_{ik} & 1
\end{bmatrix} \\
K_2 &= \begin{bmatrix}
1 & \sin(\phi_{ik} - \phi_{jk}) \tan\theta_{ik} & \left({}^i_j R_{1,3} \cos\psi_{ik} + {}^i_j R_{2,3} \sin\psi_{ik}\right) \sec\theta_{ik} \\
0 & \cos(\phi_{ik} - \phi_{jk}) & -\cos\theta_{jk} \sin(\phi_{ik} - \phi_{jk}) \\
0 & \sin(\phi_{ik} - \phi_{jk}) \sec\theta_{ik} & \cos\theta_{jk} \cos(\phi_{ik} - \phi_{jk}) \sec\theta_{ik}
\end{bmatrix}.
\end{aligned}
$$

Propagation in a Lie Group

Uncertainty on a Manifold

- Log and Exp mapping between SO(3) and so(3)
  - Lie algebra lives in a locally linear space
  - Consider perturbation in Lie algebra





$$\boxed{\tilde{\mathrm{R}} = \mathrm{R}\,\mathrm{Exp}(\epsilon), \qquad \epsilon \sim \mathcal{N}(0, \Sigma)}$$

Noise Propagation on Lie Group (1/4)

- For vector space

$$x_{k+1} = f(x_k, u_k) + w, \quad w \sim \mathcal{N}(0, \Sigma_w)$$

- For Lie group

$$X_{k+1} = X_k U_k \mathrm{Exp}(w^\wedge), \quad w \in \Re^n \sim \mathcal{N}(0, \Sigma_w)$$

Matrix multiplication

  - State noise and control noise are in Lie algebra
  - How to propagate to k+1 step?

Noise Propagation on Lie Group (2/4)

- For true state X and current state noise is

$$\bar{X} = X\mathrm{Exp}(\xi^\wedge)$$

  - Reversely, the true state

$$X = \bar{X}\mathrm{Exp}(-\xi^\wedge)$$

using $\mathrm{Exp}(A^{-1}) = \mathrm{Exp}(-A)$ for Lie algebra A

- Let's consider the propagation

$$X_{k+1} = X_k U_k \mathrm{Exp}(w^\wedge), \quad w \in \Re^n \sim \mathcal{N}(0, \Sigma_w)$$

$$\bar{X}_{k+1}\underline{\mathrm{Exp}(-\xi_{k+1}^\wedge)} = \bar{X}_k\underline{\mathrm{Exp}(-\xi_k^\wedge)}\, U_k\underline{\mathrm{Exp}(w^\wedge)}$$

How to move Exp to one side?
== How to switch the multiplication order?

Adjoint!

Adjoint (1/2)

- For Lie group element X and Lie algebra ξ

$$\boxed{Ad_X(\xi) = X\xi^\wedge X^{-1}} \qquad Ad_X(\cdot)$$

$$(Ad_X\xi)^\wedge = X\xi^\wedge X^{-1} \qquad Ad_X\text{: Adjoint matrix}$$

- Adjoint matrix
  - For rotation, SO(3) $(Ad_R w)^\wedge = Rw^\wedge R^{-1} = (Rw)^\wedge \rightarrow Ad_R = R$
  - For SE(3)

$$(Ad_X\xi)^\wedge = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w^\wedge & v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R^\top & -R^\top p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} RwR^\top & -Rw^\wedge R^\top p + Rv \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} (Rw)^\wedge & p^\wedge (Rw)^\wedge + Rv \\ 0 & 0 \end{bmatrix} = \left(\begin{bmatrix} R & 0 \\ p^\wedge R & R \end{bmatrix} \begin{bmatrix} w \\ v \end{bmatrix} \right)^\wedge \rightarrow Ad_X = \begin{bmatrix} R & 0 \\ p^\wedge R & R \end{bmatrix}$$

$$\boxed{Ad_R = R \quad \& \quad Ad_X = \begin{bmatrix} R & 0 \\ p^\wedge R & R \end{bmatrix}}$$

Adjoint (2/2)

- Adjoint for matrix multiplication order change

$$(Ad_X \xi)^\wedge = X \xi^\wedge X^{-1}$$

$$\mathrm{Exp}\left((Ad_X \xi)^\wedge\right) = X \mathrm{Exp}(\xi^\wedge) X^{-1}$$

$$X^{-1}\mathrm{Exp}\left((Ad_X \xi)^\wedge\right) = \mathrm{Exp}(\xi^\wedge) X^{-1}$$

$$\mathrm{Exp}\left((Ad_X \xi)^\wedge\right) X = X \mathrm{Exp}(\xi^\wedge)$$

$$\boxed{\mathrm{Exp}(-\xi_k^\wedge)\, U_k} = U_k \,\mathrm{Exp}\left((-Ad_{U_k^{-1}} \xi_k)^\wedge\right)$$

$$\bar{X}_{k+1}\mathrm{Exp}(-\xi_{k+1}^\wedge) = \bar{X}_k \boxed{\mathrm{Exp}(-\xi_k^\wedge)\, U_k} \mathrm{Exp}(w^\wedge)$$

Noise Propagation on Lie Group (3/4)

- Back to the equation

$$\bar{X}_{k+1}\mathrm{Exp}(-\xi_{k+1}^{\wedge}) = \bar{X}_k \underline{\mathrm{Exp}(-\xi_k^{\wedge})}\ \underline{U_k}\mathrm{Exp}(w^{\wedge})$$

Switch using Adjoint

$$= U_k\ \mathrm{Exp}\left((-Ad_{U_k^{-1}}\xi_k)^{\wedge}\right)$$

$$\bar{X}_{k+1}\underline{\mathrm{Exp}(-\xi_{k+1}^{\wedge})} = \bar{X}_k U_k\ \underline{\mathrm{Exp}\left((-Ad_{U_k^{-1}}\xi_k)^{\wedge}\right)\mathrm{Exp}(w^{\wedge})}$$

Only look at the Exp parts

  - Only examining the Exp parts, we have

$$\mathrm{Exp}(-\xi_{k+1}^{\wedge}) = \mathrm{Exp}\left((-Ad_{U_k^{-1}}\xi_k)^{\wedge}\right)\mathrm{Exp}(w^{\wedge})$$

$$\mathrm{Exp}(A)\mathrm{Exp}(B) \neq \mathrm{Exp}(A+B)$$

  - Q. How to express ξ_{k+1} (propagated uncertainty) in terms of ξ_k (current state uncertainty) and the control noise w ?

Noise Propagation on Lie Group (4/4)

- BCH Series (Baker–Campbell–Hausdorff formula)
  - Approximate the multiplication of two Exp
  - For $\mathrm{Exp}(\mathrm{X})\mathrm{Exp}(\mathrm{Y}) = \mathrm{Exp}(\mathrm{Z})$

$$Z = X + Y + \frac{1}{2}[X, Y] + \frac{1}{12}[X, [X, Y]] - \frac{1}{12}[Y, [X, Y]] + \cdots$$

Lie bracket: $[X, Y] = XY - YX$

  - If BOTH ξ_1 and ξ_2 are very small, we can approximate

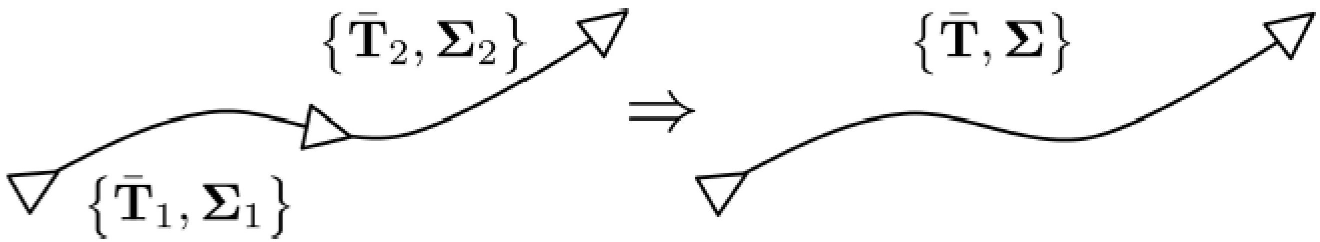
$$\mathrm{Exp}(\xi_1^\wedge)\mathrm{Exp}(\xi_2^\wedge) \approx \mathrm{Exp}(\xi_1^\wedge + \xi_2^\wedge) + H.O.T$$

  - Then we can simplify $\mathrm{Exp}(-\xi_{k+1}^\wedge) = \mathrm{Exp}\left((-Ad_{U_k^{-1}}\xi_k)^\wedge\right)\mathrm{Exp}(w^\wedge)$

$$\boxed{\xi_{k+1}^\wedge = Ad_{U_k^{-1}}\xi_k - w \rightarrow \Sigma_{k+1} = Ad_{U_k^{-1}}\Sigma_k Ad_{U_k^{-1}}^\top + \Sigma_w}$$

Example: Pose Compounding

- Pose compound



$\{\bar{\mathbf{T}}_1, \boldsymbol{\Sigma}_1\}, \quad \{\bar{\mathbf{T}}_2, \boldsymbol{\Sigma}_2\}$ What is $\{\bar{\mathbf{T}}, \boldsymbol{\Sigma}\}$?

$$\mathbf{T} := \exp\left(\boldsymbol{\xi}^\wedge\right) \bar{\mathbf{T}}$$

Left/right multiplication both valid

$$\exp\left(\boldsymbol{\xi}^\wedge\right) \bar{\mathbf{T}} = \exp\left(\boldsymbol{\xi}_1^\wedge\right) \bar{\mathbf{T}}_1 \exp\left(\boldsymbol{\xi}_2^\wedge\right) \bar{\mathbf{T}}_2.$$

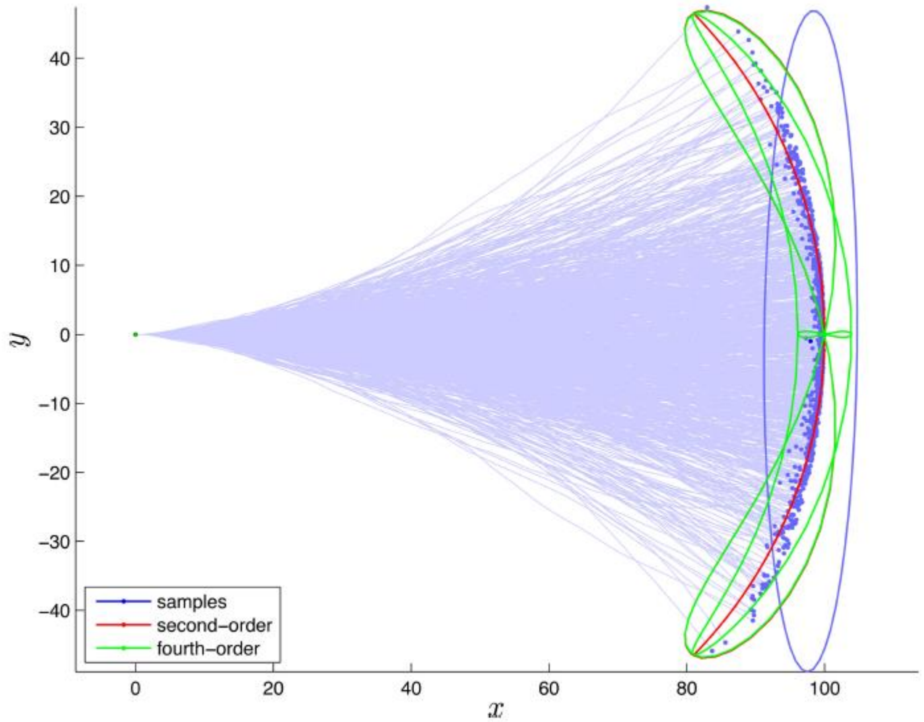
using $\bar{\mathbf{T}} = \bar{\mathbf{T}}_1 \bar{\mathbf{T}}_2$

$$\exp\left(\boldsymbol{\xi}^\wedge\right) = \exp\left(\boldsymbol{\xi}_1^\wedge\right) \exp\left(\left(\bar{\boldsymbol{\mathcal{T}}}_1 \boldsymbol{\xi}_2\right)^\wedge\right) \quad \bar{\boldsymbol{\mathcal{T}}}_1 = \mathrm{Ad}\left(\bar{\mathbf{T}}_1\right)$$

Example: Uncertainty Propagation Plot

- We can solve for the compound uncertainty

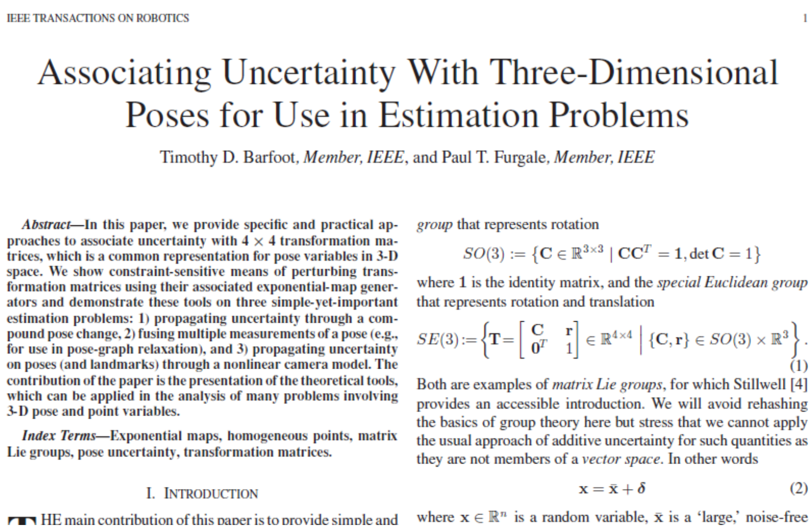
$$\xi = \xi_1 + \xi_2' + \frac{1}{2}\xi_1^{\curlywedge}\xi_2' + \frac{1}{12}\xi_1^{\curlywedge}\xi_1^{\curlywedge}\xi_2' + \frac{1}{12}\xi_2'^{\curlywedge}\xi_2'^{\curlywedge}\xi_1$$
$$- \frac{1}{24}\xi_2'^{\curlywedge}\xi_1^{\curlywedge}\xi_1^{\curlywedge}\xi_2' + \cdots .$$



Advanced topic !

Further Reference

- Uncertainty propagation on Lie Group



IEEE TRANSACTIONS ON ROBOTICS 1

Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems

Timothy D. Barfoot, *Member, IEEE*, and Paul T. Furgale, *Member, IEEE*

***Abstract*—In this paper, we provide specific and practical approaches to associate uncertainty with 4 × 4 transformation matrices, which is a common representation for pose variables in 3-D space. We show constraint-sensitive means of perturbing transformation matrices using their associated exponential-map generators and demonstrate these tools on three simple-yet-important estimation problems: 1) propagating uncertainty through a compound pose change, 2) fusing multiple measurements of a pose (e.g., for use in pose-graph relaxation), and 3) propagating uncertainty on poses (and landmarks) through a nonlinear camera model. The contribution of the paper is the presentation of the theoretical tools, which can be applied in the analysis of many problems involving 3-D pose and point variables.**

***Index Terms*—Exponential maps, homogeneous points, matrix Lie groups, pose uncertainty, transformation matrices.**

I. INTRODUCTION

THE main contribution of this paper is to provide simple and

group that represents rotation

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

where $\mathbf{1}$ is the identity matrix, and the *special Euclidean group* that represents rotation and translation

$$SE(3) := \left\{ \mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \;\middle|\; \{\mathbf{C}, \mathbf{r}\} \in SO(3) \times \mathbb{R}^3 \right\}. \quad (1)$$

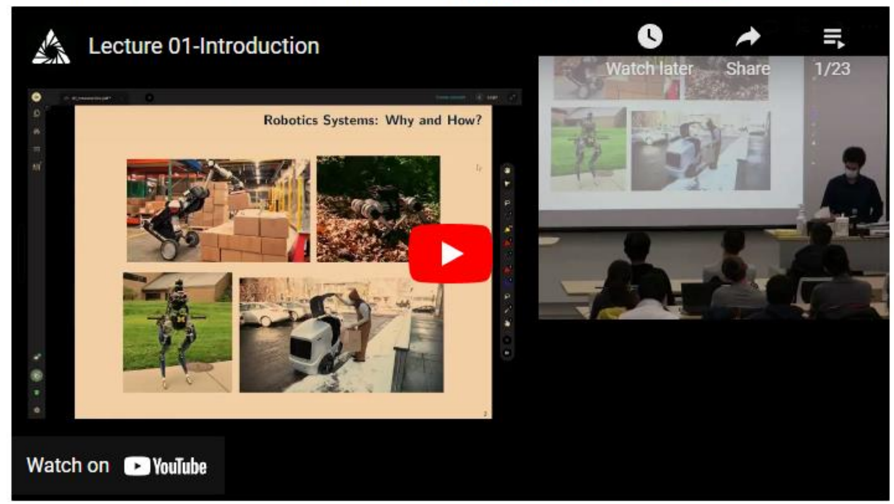
Both are examples of *matrix Lie groups*, for which Stillwell [4] provides an accessible introduction. We will avoid rehashing the basics of group theory here but stress that we cannot apply the usual approach of additive uncertainty for such quantities as they are not members of a *vector space*. In other words

$$\mathbf{x} = \bar{\mathbf{x}} + \boldsymbol{\delta} \quad (2)$$

where $\mathbf{x} \in \mathbb{R}^n$ is a random variable, $\bar{\mathbf{x}}$ is a 'large,' noise-free

T. Barfoot and P. Furgale, "Associating Uncertainty With Three-Dimensional Poses for Use in Estimation Problems", TRO 2014.

- University of Michigan SLAM lecture (영) (Prof. Maani Ghaffari) & 수업 Github



Lecture by Prof. Maani Ghaffari

Summary

- Vector space
  - Uncertainty as a Gaussian
  - Motion/measurement model as a system
  - Linearize if needed
  - Gaussian input/output via linear system

$$X = [x, y, z, r, p, h]$$

→ Linearized vector representation

- Manifold
  - Project onto locally linear space
  - Rotation (*not* linear)

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

→ Nonlinear SE(3) representation

Thank you very much !!