Lecture 10: Optimization and Probability for Matrix Lie Groups

AER1513: State Estimation

Timothy D. Barfoot

University of Toronto

UNIVERSITY OF TORONTO

Outline

UNIVERSITY OF TORONTO

Motivation

- in the last lecture, we learned that the sets of rotation and transformation (pose) matrices are matrix Lie groups
- we now would like to revisit two key concepts,
  - optimization
  - probability

  for rotations/poses through the matrix Lie group lens
- both of these tools will be important for us to adjust our estimation tools to work with rotations/poses

UNIVERSITY OF TORONTO

Special orthogonal group

- the set of rotations is called the special orthogonal group:

$$SO(3) = \left\{ \mathbf{C} \in \mathbb{R}^{3\times 3} | \mathbf{C}\mathbf{C}^T = \mathbf{1}, \det \mathbf{C} = 1 \right\} \tag{1}$$

- the $\mathbf{C}\mathbf{C}^T = \mathbf{1}$ orthogonality condition is needed to impose 6 constraints on the 9-parameter rotation matrix, reducing the degrees of freedom to 3
- noticing that

$$(\det \mathbf{C})^2 = \det \left(\mathbf{C}\mathbf{C}^T \right) = \det \mathbf{1} = 1 \tag{2}$$

  we have that $\det \mathbf{C} = \pm 1$, allowing for two possibilities
- choosing $\det \mathbf{C} = 1$ ensures that we have a proper rotation
- the other case, $\det \mathbf{C} = -1$, corresponds to a rotary reflection

Special Euclidean group

– the set of transformation matrices representing poses is called the special Euclidean group:

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

UNIVERSITY OF TORONTO

Exponential rotations

- for rotations, we can relate elements of $SO(3)$ (Lie group) to elements of $\mathfrak{so}(3)$ (Lie algebra) through the exponential map:

$$\mathbf{C} = \exp\left(\boldsymbol{\phi}^{\wedge}\right) = \sum_{n=0}^{\infty} \frac{1}{n!}\left(\boldsymbol{\phi}^{\wedge}\right)^{n} \tag{4}$$

  where $\mathbf{C} \in SO(3)$ and $\boldsymbol{\phi} \in \mathbb{R}^3$ (and hence $\boldsymbol{\phi}^{\wedge} \in \mathfrak{so}(3)$)
- we can also go in the other direction (but not uniquely) using

$$\boldsymbol{\phi} = \ln\left(\mathbf{C}\right)^{\vee} \tag{5}$$

- the mapping is surjective (or onto), meaning every element of $SO(3)$ can be generated by at least one element of $\mathfrak{so}(3)$
- the non-unique inverse mapping is precisely the idea of singularities discussed earlier – in this case $\phi + 2\pi m$ with m any integer produces the same $\mathbf{C}$

UNIVERSITY OF TORONTO

Exponential poses

- for poses, we can relate elements of $SE(3)$ (Lie group) to elements of $\mathfrak{se}(3)$ (Lie algebra), again through the exponential map:

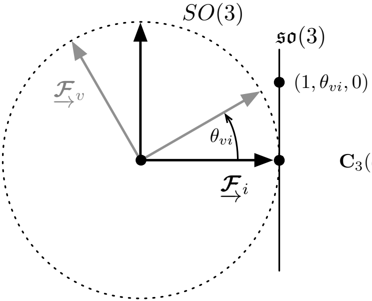
$$\mathbf{T} = \exp\left(\boldsymbol{\xi}^\wedge\right) = \sum_{n=0}^{\infty} \frac{1}{n!}\left(\boldsymbol{\xi}^\wedge\right)^n \tag{6}$$

  where $\mathbf{T} \in SE(3)$ and $\boldsymbol{\xi} \in \mathbb{R}^6$ (and hence $\boldsymbol{\xi}^\wedge \in \mathfrak{se}(3)$)
- we can also go in the other direction (again, not uniquely) using

$$\boldsymbol{\xi} = \ln\left(\mathbf{T}\right)^\vee \tag{7}$$

- the exponential map from $\mathfrak{se}(3)$ to $SE(3)$ is also surjective: every $\boldsymbol{\xi} \in \mathbb{R}^6$ maps to some $\mathbf{T} \in SE(3)$ (many-to-one) and every $\mathbf{T} \in SE(3)$ can be generated by at least one $\boldsymbol{\xi} \in \mathbb{R}^6$

Tangent space



$$\mathbf{C}_3(\theta_{vi})\,\mathbf{1}_1 = \exp\left(\begin{bmatrix} 0 \\ 0 \\ \theta_{vi} \end{bmatrix}^\wedge\right)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$
$$\approx \left(\mathbf{1} + \begin{bmatrix} 0 \\ 0 \\ \theta_{vi} \end{bmatrix}^\wedge\right)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ \theta_{vi} \\ 0 \end{bmatrix}$$

- the vectorspace of a Lie algebra is the tangent space of the associated Lie group at the identity element of the group, and it completely captures the local structure of the group

Perturbations

- we now introduce the idea of perturbations
- for vectors, we usually perturb like this:

$$\mathbf{x} = \underbrace{\bar{\mathbf{x}}}_{\text{'big'}} + \underbrace{\delta\mathbf{x}}_{\text{'small'}} \tag{8}$$

  but actually this is an arbitrary choice
- in an optimization setting, perturbations are used like this:

$$\mathbf{x} = \underbrace{\bar{\mathbf{x}}}_{\text{initial guess}} + \underbrace{\delta\mathbf{x}}_{\text{optimal update}} \tag{9}$$

- in a probability setting, perturbations are used like this:

$$\mathbf{x} = \underbrace{\bar{\mathbf{x}}}_{\text{deterministic}} + \underbrace{\delta\mathbf{x}}_{\text{random noise}} \tag{10}$$

UNIVERSITY OF TORONTO

Rotation perturbations

- for rotations, we will perturb like this:

$$\mathbf{C} = \underbrace{\delta\mathbf{C}}_{\text{'small'}} \; \underbrace{\bar{\mathbf{C}}}_{\text{'big'}} \tag{11}$$

- we pick the following perturbation

$$\delta\mathbf{C} = \exp\left(\boldsymbol{\psi}^\wedge\right) \tag{12}$$

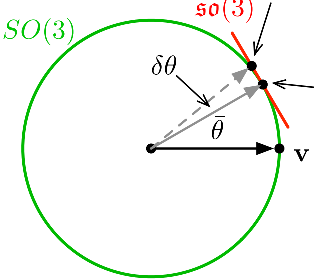
  which ensures $\mathbf{C}$ is still a valid rotation (by closure)

- this lets us linearize the product of a rotation and point, $\mathbf{v}$:

$$\mathbf{C}\mathbf{v} = \delta\mathbf{C}\,\bar{\mathbf{C}}\mathbf{v} = \exp\left(\boldsymbol{\psi}^\wedge\right)\bar{\mathbf{C}}\mathbf{v} \approx \left(\mathbf{1} + \boldsymbol{\psi}^\wedge\right)\bar{\mathbf{C}}\mathbf{v} = \bar{\mathbf{C}}\mathbf{v} - \left(\bar{\mathbf{C}}\mathbf{v}\right)^\wedge \boldsymbol{\psi} \tag{13}$$

UNIVERSITY OF TORONTO

Rotation perturbations



$$\mathbf{C}\mathbf{v} \approx \bar{\mathbf{C}}\mathbf{v} - \left(\bar{\mathbf{C}}\mathbf{v}\right)^\wedge \boldsymbol{\psi} = \begin{bmatrix}\cos\bar{\theta} \\ \sin\bar{\theta} \\ 0\end{bmatrix} + \begin{bmatrix}-\sin\bar{\theta} \\ \cos\bar{\theta} \\ 0\end{bmatrix}\delta\theta$$

$$\bar{\mathbf{C}}\mathbf{v} = \begin{bmatrix}\cos\bar{\theta} \\ \sin\bar{\theta} \\ 0\end{bmatrix}$$

$$\mathbf{v} = \begin{bmatrix}1 \\ 0 \\ 0\end{bmatrix} \qquad \boldsymbol{\psi} = \begin{bmatrix}0 \\ 0 \\ \delta\theta\end{bmatrix} \qquad \bar{\mathbf{C}} = \begin{bmatrix}\cos\bar{\theta} & -\sin\bar{\theta} & 0 \\ \sin\bar{\theta} & \cos\bar{\theta} & 0 \\ 0 & 0 & 1\end{bmatrix}$$

UNIVERSITY OF TORONTO

Pose perturbations

- for poses, we will perturb like this:

$$\mathbf{T} = \underbrace{\delta\mathbf{T}}_{\text{'small'}} \underbrace{\bar{\mathbf{T}}}_{\text{'big'}} \tag{14}$$

- we pick the following perturbation

$$\delta\mathbf{T} = \exp\left(\boldsymbol{\epsilon}^\wedge\right) \tag{15}$$

which ensures $\mathbf{T}$ is still a valid pose (by closure)

- this lets us linearize the product of a pose and homogeneous point, $\mathbf{p}$:

$$\mathbf{T}\mathbf{p} = \delta\mathbf{T}\,\bar{\mathbf{T}}\mathbf{p} = \exp\left(\boldsymbol{\epsilon}^\wedge\right)\bar{\mathbf{T}}\mathbf{p} \approx \left(\mathbf{1} + \boldsymbol{\epsilon}^\wedge\right)\bar{\mathbf{T}}\mathbf{p} = \bar{\mathbf{T}}\mathbf{p} + \left(\bar{\mathbf{T}}\mathbf{p}\right)^\odot \boldsymbol{\epsilon} \tag{16}$$

- we have used

$$\boldsymbol{\epsilon}^\wedge\mathbf{p} \equiv \mathbf{p}^\odot\boldsymbol{\epsilon}, \qquad \mathbf{p}^\odot = \begin{bmatrix} \boldsymbol{\rho} \\ \eta \end{bmatrix}^\odot = \begin{bmatrix} \eta\mathbf{1} & -\boldsymbol{\rho}^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \tag{17}$$

UNIVERSITY OF TORONTO

Rotation optimization

– choose a perturbation scheme,

$$\mathbf{C} = \exp\left(\boldsymbol{\psi}^\wedge\right)\mathbf{C}_{\text{op}} \tag{18}$$

where $\boldsymbol{\psi}$ is a small perturbation applied to an initial guess, $\mathbf{C}_{\text{op}}$

– insert this in the function, $u(\mathbf{x})$, to be optimized:

$$\begin{aligned} u\left(\mathbf{C}\mathbf{v}\right) &= u\left(\exp\left(\boldsymbol{\psi}^\wedge\right)\mathbf{C}_{\text{op}}\mathbf{v}\right) \approx u\left(\left(\mathbf{1} + \boldsymbol{\psi}^\wedge\right)\mathbf{C}_{\text{op}}\mathbf{v}\right) \\ &\approx u(\mathbf{C}_{\text{op}}\mathbf{v}) \underbrace{- \left.\frac{\partial u}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{C}_{\text{op}}\mathbf{v}} \left(\mathbf{C}_{\text{op}}\mathbf{v}\right)^\wedge}_{\boldsymbol{\delta}^T} \boldsymbol{\psi} = u(\mathbf{C}_{\text{op}}\mathbf{v}) + \boldsymbol{\delta}^T\boldsymbol{\psi} \end{aligned} \tag{19}$$

– then pick a perturbation, $\boldsymbol{\psi}$, to decrease the function

UNIVERSITY OF TORONTO

Rotation optimization: gradient descent

- suppose we would like to perform gradient descent
- in this case, we would pick the perturbation to be of the form

$$\boldsymbol{\psi} = -\alpha \boldsymbol{\delta} \tag{20}$$

  with $\alpha > 0$ a small step size
- we see the function is reduced by taking this step:

$$u\left(\mathbf{C}\mathbf{v}\right) - u\left(\mathbf{C}_{\mathrm{op}}\mathbf{v}\right) \approx -\underbrace{\alpha\, \boldsymbol{\delta}^T \boldsymbol{\delta}}_{\geq 0} \tag{21}$$

- then apply the perturbation to update the initial guess,

$$\mathbf{C}_{\mathrm{op}} \leftarrow \exp\left(-\alpha \boldsymbol{\delta}^\wedge\right) \mathbf{C}_{\mathrm{op}} \tag{22}$$

  so that $\mathbf{C}_{\mathrm{op}} \in SO(3)$ at each iteration; iterate to convergence

UNIVERSITY OF TORONTO

Rotation optimization: Gauss-Newton

- gradient descent can be quite slow
- let's look at Gauss-Newton optimization
- suppose we have a general nonlinear, quadratic cost function of a rotation of the form,

$$J(\mathbf{C}) = \frac{1}{2} \sum_m \left(u_m(\mathbf{C}\mathbf{v}_m)\right)^2 \tag{23}$$

  where $u_m(\cdot)$ are scalar nonlinear functions and $\mathbf{v}_m \in \mathbb{R}^3$ are three-dimensional points
- we begin with an initial guess for the optimal rotation, $\mathbf{C}_{\text{op}} \in SO(3)$, and then perturb this (on the left) according to

$$\mathbf{C} = \exp\left(\boldsymbol{\psi}^\wedge\right) \mathbf{C}_{\text{op}} \tag{24}$$

  where $\boldsymbol{\psi}$ is the perturbation

UNIVERSITY OF TORONTO

Rotation optimization: Gauss-Newton

– we then apply our perturbation scheme inside each $u_m(\cdot)$ so that

$$u_m\left(\mathbf{C}\mathbf{v}_m\right) = u_m\left(\exp(\boldsymbol{\psi}^\wedge)\mathbf{C}_{\mathrm{op}}\mathbf{v}_m\right) \approx u_m\left(\left(\mathbf{1} + \boldsymbol{\psi}^\wedge\right)\mathbf{C}_{\mathrm{op}}\mathbf{v}_m\right)$$

$$\approx \underbrace{u_m(\mathbf{C}_{\mathrm{op}}\mathbf{v}_m)}_{\beta_m} - \underbrace{\left.\frac{\partial u_m}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{C}_{\mathrm{op}}\mathbf{v}_m} \left(\mathbf{C}_{\mathrm{op}}\mathbf{v}_m\right)^\wedge \boldsymbol{\psi}}_{\boldsymbol{\delta}_m^T} \tag{25}$$

is a linearized version of $u_m(\cdot)$ in terms of our perturbation, $\boldsymbol{\psi}$

– inserting this back into our cost function we have

$$J(\mathbf{C}) \approx \frac{1}{2}\sum_m \left(\boldsymbol{\delta}_m^T\boldsymbol{\psi} + \beta_m\right)^2 \tag{26}$$

which is exactly quadratic in $\boldsymbol{\psi}$

UNIVERSITY OF TORONTO

Rotation optimization: Gauss-Newton

- taking the derivative of J with respect to $\boldsymbol{\psi}$ we have

$$\frac{\partial J}{\partial \boldsymbol{\psi}^T} = \sum_m \boldsymbol{\delta}_m \left(\boldsymbol{\delta}_m^T \boldsymbol{\psi} + \beta_m \right) \tag{27}$$

- set the derivative to zero to find the optimal perturbation, $\boldsymbol{\psi}^\star$:

$$\left(\sum_m \boldsymbol{\delta}_m \boldsymbol{\delta}_m^T \right) \boldsymbol{\psi}^\star = -\sum_m \beta_m \boldsymbol{\delta}_m \tag{28}$$

- this is a linear system of equations, which we can solve for $\boldsymbol{\psi}^\star$
- apply this optimal perturbation to our initial guess,

$$\mathbf{C}_{\mathrm{op}} \leftarrow \exp\left(\boldsymbol{\psi}^{\star^\wedge} \right) \mathbf{C}_{\mathrm{op}}, \tag{29}$$

  so that $\mathbf{C}_{\mathrm{op}} \in SO(3)$ at each iteration; iterate to convergence

UNIVERSITY OF TORONTO

Rotation optimization commentary

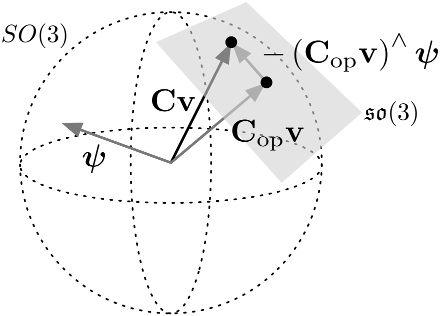
- we have adapted classic optimization algorithms to work with the matrix Lie group, $SO(3)$, by exploiting the surjective property of the exponential map to define an appropriate perturbation scheme

$$\mathbf{C} = \exp\left(\boldsymbol{\psi}^\wedge\right)\mathbf{C}_{\text{op}} \tag{30}$$

- we are essentially assuming that at each iteration the update, $\boldsymbol{\psi}$, will be small and so have mapped the optimization problem from the Lie group up into the Lie algebra, $\mathfrak{so}(3)$
- this approach has three major advantages:
  - we are storing our rotation in a singularity-free format, $\mathbf{C}_{\text{op}}$
  - at each iteration we are performing unconstrained optimization
  - our manipulations occur at the matrix level
- we get away with this because the perturbation always becomes very small as we converge to the optimum

UNIVERSITY OF TORONTO

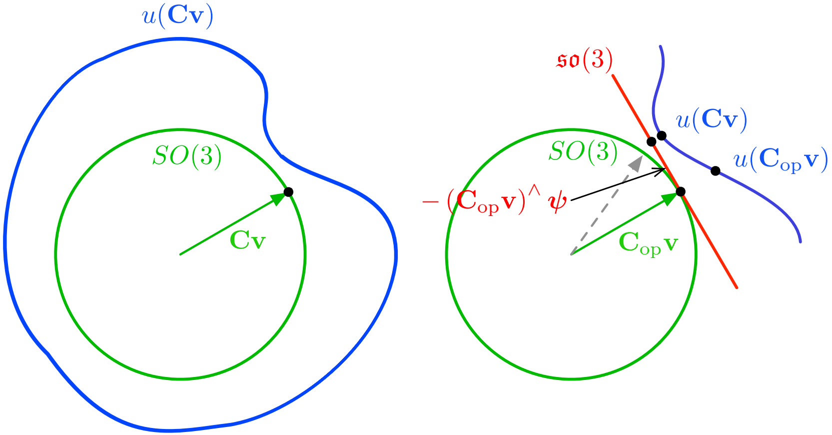
Rotation perturbations



$$\mathbf{C}\mathbf{v} = \exp\left(\boldsymbol{\psi}^{\wedge}\right)\mathbf{C}_{\mathrm{op}}\mathbf{v} \approx \mathbf{C}_{\mathrm{op}}\mathbf{v} - \left(\mathbf{C}_{\mathrm{op}}\mathbf{v}\right)^{\wedge}\boldsymbol{\psi}$$

UNIVERSITY OF TORONTO

Rotation optimization



UNIVERSITY OF TORONTO

Pose optimization

- the same concepts can also be applied to poses
- suppose we have a general nonlinear, quadratic cost function of a transformation of the form

$$J(\mathbf{T}) = \frac{1}{2} \sum_m \left(u_m(\mathbf{T}\mathbf{p}_m)\right)^2 \tag{31}$$

where $u_m(\cdot)$ are nonlinear functions and $\mathbf{p}_m \in \mathbb{R}^4$ are three-dimensional points expressed in homogeneous coordinates

- we begin with an initial guess for the optimal transformation, $\mathbf{T}_{\text{op}} \in SE(3)$, and then perturb this (on the left) according to

$$\mathbf{T} = \exp\left(\boldsymbol{\epsilon}^\wedge\right)\mathbf{T}_{\text{op}} \tag{32}$$

where $\boldsymbol{\epsilon}$ is the perturbation

UNIVERSITY OF TORONTO

Pose optimization

– we then apply our perturbation scheme inside each $u_m(\cdot)$ so that

$$u_m\left(\mathbf{T}\mathbf{p}_m\right) = u_m\left(\exp(\boldsymbol{\epsilon}^\wedge)\mathbf{T}_{\text{op}}\mathbf{p}_m\right) \approx u_m\left(\left(\mathbf{1} + \boldsymbol{\epsilon}^\wedge\right)\mathbf{T}_{\text{op}}\mathbf{p}_m\right)$$
$$\approx \underbrace{u_m(\mathbf{T}_{\text{op}}\mathbf{p}_m)}_{\beta_m} + \underbrace{\left.\frac{\partial u_m}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{T}_{\text{op}}\mathbf{p}_m}\left(\mathbf{T}_{\text{op}}\mathbf{p}_m\right)^{\odot}}_{\boldsymbol{\delta}_m^T}\boldsymbol{\epsilon} \quad (33)$$

is a linearized version of $u_m(\cdot)$ in terms of our perturbation, $\boldsymbol{\epsilon}$

– inserting this back into our cost function we have

$$J(\mathbf{T}) = \frac{1}{2}\sum_m \left(\boldsymbol{\delta}_m^T\boldsymbol{\epsilon} + \beta_m\right)^2 \quad (34)$$

which is exactly quadratic in $\boldsymbol{\epsilon}$

UNIVERSITY OF TORONTO

Pose optimization

- taking the derivative of J with respect to $\boldsymbol{\epsilon}$ we have

$$\frac{\partial J}{\partial \boldsymbol{\epsilon}^T} = \sum_m \boldsymbol{\delta}_m \left(\boldsymbol{\delta}_m^T \boldsymbol{\epsilon} + \beta_m \right) \tag{35}$$

- set the derivative to zero to find the optimal perturbation, $\boldsymbol{\epsilon}^\star$:

$$\left(\sum_m \boldsymbol{\delta}_m \boldsymbol{\delta}_m^T \right) \boldsymbol{\epsilon}^\star = - \sum_m \beta_m \boldsymbol{\delta}_m \tag{36}$$

- this is a linear system of equations, which we can solve for $\boldsymbol{\epsilon}^\star$
- apply this optimal perturbation to our initial guess:

$$\mathbf{T}_{\text{op}} \leftarrow \exp\left({\boldsymbol{\epsilon}^\star}^\wedge \right) \mathbf{T}_{\text{op}} \tag{37}$$

so that $\mathbf{T}_{\text{op}} \in SE(3)$ at each iteration; iterate to convergence

UNIVERSITY OF TORONTO

Point-cloud alignment

- consider the problem of aligning two point-clouds, $\boldsymbol{y}_j$ and $\boldsymbol{p}_j$, which are in homogeneous-point form and $j = 1 \ldots J$
- we define our error term for each point pair as

$$\mathbf{e}_j = \boldsymbol{y}_j - \mathbf{T}\boldsymbol{p}_j \tag{38}$$

- we define our objective function as

$$J(\mathbf{T}) = \frac{1}{2}\sum_{j=1}^{M} w_j \mathbf{e}_j^T \mathbf{e}_j = \frac{1}{2}\sum_{j=1}^{M} w_j \left(\boldsymbol{y}_j - \mathbf{T}\boldsymbol{p}_j\right)^T \left(\boldsymbol{y}_j - \mathbf{T}\boldsymbol{p}_j\right) \tag{39}$$

  where $w_j > 0$ are scalar weights
- we seek to minimize J with respect to $\mathbf{T} \in SE(3)$; we want to know the pose between the two point-clouds

Point-cloud alignment

– we use our $SE(3)$-sensitive perturbation scheme

$$\mathbf{T} = \exp\left(\boldsymbol{\epsilon}^\wedge\right)\mathbf{T}_{\text{op}} \approx \left(\mathbf{1} + \boldsymbol{\epsilon}^\wedge\right)\mathbf{T}_{\text{op}} \tag{40}$$

where $\mathbf{T}_{\text{op}}$ is some initial guess and $\boldsymbol{\epsilon}$ is a small perturbation

– inserting this into the objective function we then have

$$J(\mathbf{T}) \approx \frac{1}{2}\sum_{j=1}^{M} w_j \left((\boldsymbol{y}_j - \boldsymbol{z}_j) - \boldsymbol{z}_j^{\odot}\,\boldsymbol{\epsilon}\right)^T \left((\boldsymbol{y}_j - \boldsymbol{z}_j) - \boldsymbol{z}_j^{\odot}\,\boldsymbol{\epsilon}\right) \tag{41}$$

where $\boldsymbol{z}_j = \mathbf{T}_{\text{op}}\boldsymbol{p}_j$ and we have used that

$$\boldsymbol{\epsilon}^\wedge\,\boldsymbol{z}_j = \boldsymbol{z}_j^{\odot}\,\boldsymbol{\epsilon} \tag{42}$$

– the objective function is now exactly quadratic in $\boldsymbol{\epsilon}$

UNIVERSITY OF TORONTO

Point-cloud alignment

- we can carry out a simple, unconstrained optimization for $\boldsymbol{\epsilon}$
- taking the derivative we find

$$\frac{\partial J}{\partial \boldsymbol{\epsilon}^T} = -\sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^T} \left((\boldsymbol{y}_j - \boldsymbol{z}_j) - \boldsymbol{z}_j^{\odot} \boldsymbol{\epsilon} \right) \tag{43}$$

- setting this to zero, we have the following system of equations for the optimal $\boldsymbol{\epsilon}^\star$:

$$\left(\frac{1}{w} \sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^T} \boldsymbol{z}_j^{\odot} \right) \boldsymbol{\epsilon}^\star = \frac{1}{w} \sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^T} (\boldsymbol{y}_j - \boldsymbol{z}_j) \tag{44}$$

- we update our operating point and iterate to convergence:

$$\mathbf{T}_{\text{op}} \leftarrow \exp\left(\boldsymbol{\epsilon}^{\star^\wedge} \right) \mathbf{T}_{\text{op}} \tag{45}$$

UNIVERSITY OF TORONTO

Pose optimization commentary

- we have adapted classic optimization algorithms to work with the matrix Lie group, $SE(3)$, by exploiting the surjective property of the exponential map to define an appropriate perturbation scheme

$$\mathbf{T} = \exp\left(\boldsymbol{\epsilon}^\wedge\right)\mathbf{T}_{\text{op}} \tag{46}$$

- we are essentially assuming that at each iteration the update, $\boldsymbol{\epsilon}$, will be small and so have mapped the optimization problem from the Lie group up into the Lie algebra, $\mathfrak{se}(3)$
- this approach has three major advantages:
  - we are storing our pose in a singularity-free format, $\mathbf{T}_{\text{op}}$
  - at each iteration we are performing unconstrained optimization
  - our manipulations occur at the matrix level
- we get away with this because the perturbation always becomes very small as we converge to the optimum

UNIVERSITY OF TORONTO

Probabilistic matrix Lie groups

- optimization is not the only use for our perturbations; we can also use them to extend the idea of probability to matrix Lie groups
- with normal vector random variables we write

$$\mathbf{x} \sim \mathcal{N}\left(\bar{\mathbf{x}}, \mathbf{\Sigma}\right) \tag{47}$$

  but we could have equivalently written

$$\mathbf{x} = \bar{\mathbf{x}} + \delta\mathbf{x}, \quad \delta\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}\right) \tag{48}$$

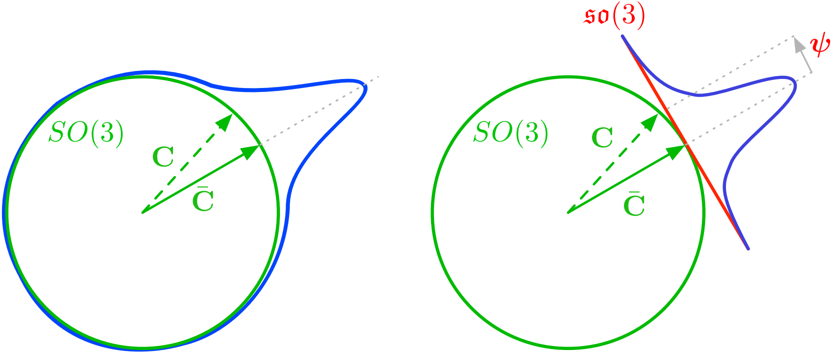
- for Lie groups, we will define Gaussian random variables like this:

$$\begin{aligned} \mathbf{C} &= \exp\left(\boldsymbol{\psi}^\wedge\right)\bar{\mathbf{C}}, \quad \boldsymbol{\psi} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}\right), \quad \mathbf{\Sigma} \in \mathbb{R}^{3\times 3} && \text{(49a)} \\ \mathbf{T} &= \exp\left(\boldsymbol{\epsilon}^\wedge\right)\bar{\mathbf{T}}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Xi}\right), \quad \mathbf{\Xi} \in \mathbb{R}^{6\times 6} && \text{(49b)} \end{aligned}$$

  where $\boldsymbol{\psi} \in \mathbb{R}^3$ and $\boldsymbol{\epsilon} \in \mathbb{R}^6$ are just vector random variables

UNIVERSITY OF TORONTO

Probabilistic rotations



– we are defining the ‘big’ mean in the Lie group and the ‘small’ covariance in the Lie algebra

– this really only works well when the covariance is not very big so that there is very little probability mass on the far side

UNIVERSITY OF TORONTO

Uncertainty on a rotated vector

- consider the simple mapping from rotation to position given by

$$\mathbf{y} = \mathbf{C}\mathbf{x} \tag{50}$$

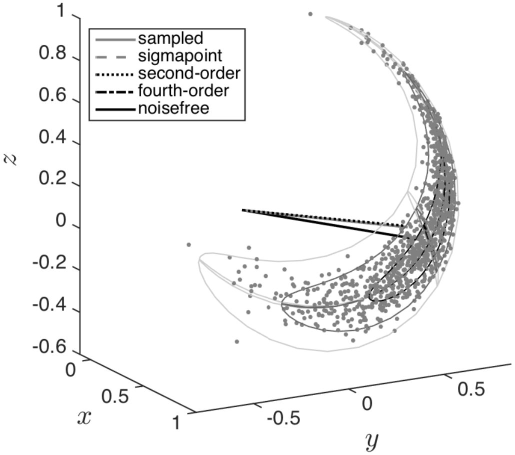
  where $\mathbf{x} \in \mathbb{R}^3$ is a constant and

$$\mathbf{C} = \exp\left(\boldsymbol{\epsilon}^\wedge\right)\bar{\mathbf{C}}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}) \tag{51}$$

- now imagine drawing a bunch of samples for $\mathbf{y}$
  - draw $\boldsymbol{\epsilon}$ from $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$
  - build $\mathbf{y} = \exp\left(\boldsymbol{\epsilon}^\wedge\right)\bar{\mathbf{C}}\mathbf{x}$
- we expect the samples live to live on sphere whose radius is $|\mathbf{x}|$ since rotations preserve length

UNIVERSITY OF TORONTO

Uncertainty on a rotated vector



Two key operations

- now that we have a way to define Gaussian PDFs for matrix Lie groups, let's look at two key things we can do with uncertain poses:
  - compounding: we may want to compound two uncertain poses, which comes up in, for example, the prediction step of the EKF
  - fusing: we may want to fuse two uncertain poses to produce a combined estimate, which comes up in the correction step of the EKF

UNIVERSITY OF TORONTO

Compounding uncertain poses

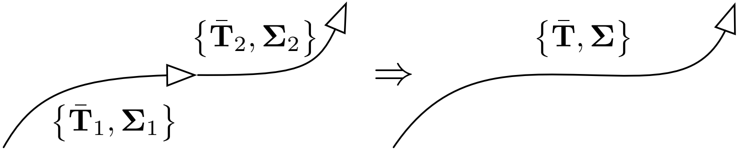
- let's look at the problem of compounding two poses, each with associated uncertainty:

$$\left\{ \bar{\mathbf{T}}_1, \mathbf{\Sigma}_1 \right\}, \quad \left\{ \bar{\mathbf{T}}_2, \mathbf{\Sigma}_2 \right\} \tag{52}$$

- suppose now we let

$$\mathbf{T} = \mathbf{T}_1 \mathbf{T}_2 \tag{53}$$

- what is $\left\{ \bar{\mathbf{T}}, \mathbf{\Sigma} \right\}$?



UNIVERSITY OF TORONTO

Compounding uncertain poses

- under our perturbation scheme we have

$$\exp\left(\boldsymbol{\epsilon}^\wedge\right)\bar{\mathbf{T}} = \exp\left(\boldsymbol{\epsilon}_1^\wedge\right)\bar{\mathbf{T}}_1\exp\left(\boldsymbol{\epsilon}_2^\wedge\right)\bar{\mathbf{T}}_2 \tag{54}$$

- moving all the uncertain factors to the left side, we have

$$\exp\left(\boldsymbol{\epsilon}^\wedge\right)\bar{\mathbf{T}} = \exp\left(\boldsymbol{\epsilon}_1^\wedge\right)\exp\left(\left(\bar{\boldsymbol{\mathcal{T}}}_1\boldsymbol{\epsilon}_2\right)^\wedge\right)\bar{\mathbf{T}}_1\bar{\mathbf{T}}_2 \tag{55}$$

  where $\bar{\boldsymbol{\mathcal{T}}}_1 = \mathrm{Ad}\left(\bar{\mathbf{T}}_1\right)$

- if we let

$$\bar{\mathbf{T}} = \bar{\mathbf{T}}_1\bar{\mathbf{T}}_2 \tag{56}$$

  we are left with

$$\exp\left(\boldsymbol{\epsilon}^\wedge\right) = \exp\left(\boldsymbol{\epsilon}_1^\wedge\right)\exp\left(\left(\bar{\boldsymbol{\mathcal{T}}}_1\boldsymbol{\epsilon}_2\right)^\wedge\right) \tag{57}$$

UNIVERSITY OF TORONTO

Combining matrix exponentials

– we can combine two scalar exponential functions as follows:

$$\exp(a)\exp(b) = \exp(a+b) \tag{58}$$

where $a, b \in \mathbb{R}$

– unfortunately, this is not so easy for the matrix case

$$\exp(\mathbf{A})\exp(\mathbf{B}) \neq \exp(\mathbf{A}+\mathbf{B}) \tag{59}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times N}$

– there are some special circumstances where it can be true or a good approximation (e.g., when both $\mathbf{A}$ and $\mathbf{B}$ are very small), but not in general

– to do the combination, we need to use the Baker-Campbell-Hausdorff (BCH) formula

UNIVERSITY OF TORONTO

BCH

- to combine matrix exponentials, we can use the Baker-Campbell-Hausdorff (BCH) formula
- the first several terms are

$$\begin{aligned}
\ln\left(\exp(\mathbf{A})\exp(\mathbf{B})\right) &= \mathbf{A} + \mathbf{B} + \frac{1}{2}\left[\mathbf{A}, \mathbf{B}\right] \\
&+ \frac{1}{12}\left[\mathbf{A}, \left[\mathbf{A}, \mathbf{B}\right]\right] - \frac{1}{12}\left[\mathbf{B}, \left[\mathbf{A}, \mathbf{B}\right]\right] - \frac{1}{24}\left[\mathbf{B}, \left[\mathbf{A}, \left[\mathbf{A}, \mathbf{B}\right]\right]\right] \\
&- \frac{1}{720}\left(\left[\left[\left[\left[\mathbf{A}, \mathbf{B}\right], \mathbf{B}\right], \mathbf{B}\right], \mathbf{B}\right] + \left[\left[\left[\left[\mathbf{B}, \mathbf{A}\right], \mathbf{A}\right], \mathbf{A}\right], \mathbf{A}\right]\right) \\
&+ \frac{1}{360}\left(\left[\left[\left[\left[\mathbf{A}, \mathbf{B}\right], \mathbf{B}\right], \mathbf{B}\right], \mathbf{A}\right] + \left[\left[\left[\left[\mathbf{B}, \mathbf{A}\right], \mathbf{A}\right], \mathbf{A}\right], \mathbf{B}\right]\right) \\
&+ \frac{1}{120}\left(\left[\left[\left[\left[\mathbf{A}, \mathbf{B}\right], \mathbf{A}\right], \mathbf{B}\right], \mathbf{A}\right] + \left[\left[\left[\left[\mathbf{B}, \mathbf{A}\right], \mathbf{B}\right], \mathbf{A}\right], \mathbf{B}\right]\right) + \cdots .
\end{aligned} \tag{60}$$

where the Lie bracket is

$$\left[\mathbf{A}, \mathbf{B}\right] = \mathbf{A}\mathbf{B} - \mathbf{B}\mathbf{A} \tag{61}$$

UNIVERSITY OF TORONTO

BCH special cases

- when $[\mathbf{A}, \mathbf{B}] = \mathbf{AB} - \mathbf{BA} = \mathbf{0}$, we have

$$\ln\left(\exp(\mathbf{A})\exp(\mathbf{B})\right) = \mathbf{A} + \mathbf{B} \tag{62}$$

- if we keep only terms linear in $\mathbf{A}$, BCH becomes

$$\ln\left(\exp(\mathbf{A})\exp(\mathbf{B})\right) \approx \mathbf{B} + \sum_{n=0}^{\infty} \frac{B_n}{n!} \underbrace{[\mathbf{B}, [\mathbf{B}, \ldots [\mathbf{B}}_{n}, \mathbf{A}] \ldots]] \tag{63}$$

- if we keep only terms linear in $\mathbf{B}$, BCH becomes

$$\ln\left(\exp(\mathbf{A})\exp(\mathbf{B})\right) \approx \mathbf{A} + \sum_{n=0}^{\infty} (-1)^n \frac{B_n}{n!} \underbrace{[\mathbf{A}, [\mathbf{A}, \ldots [\mathbf{A}}_{n}, \mathbf{B}] \ldots]] \tag{64}$$

- the B_n are the Bernoulli numbers

$$B_0 = 1, B_1 = -\frac{1}{2}, B_2 = \frac{1}{6}, B_3 = 0, B_4 = -\frac{1}{30}, B_5 = 0, B_6 = \frac{1}{42},$$
$$B_7 = 0, B_8 = -\frac{1}{30}, B_9 = 0, B_{10} = \frac{5}{66}, B_{11} = 0, B_{12} = -\frac{691}{2730}, \ldots \tag{65}$$

UNIVERSITY OF TORONTO

BCH rotations

- in the particular case of $SO(3)$, we can show that

$$\begin{aligned}\ln\left(\mathbf{C}_1\mathbf{C}_2\right)^\vee &= \ln\left(\exp(\boldsymbol{\phi}_1^\wedge)\exp(\boldsymbol{\phi}_2^\wedge)\right)^\vee \\ &= \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2 + \frac{1}{2}\boldsymbol{\phi}_1^\wedge\boldsymbol{\phi}_2 + \frac{1}{12}\boldsymbol{\phi}_1^\wedge\boldsymbol{\phi}_1^\wedge\boldsymbol{\phi}_2 + \frac{1}{12}\boldsymbol{\phi}_2^\wedge\boldsymbol{\phi}_2^\wedge\boldsymbol{\phi}_1 + \cdots \end{aligned} \tag{66}$$

  where $\mathbf{C}_1 = \exp(\boldsymbol{\phi}_1^\wedge), \mathbf{C}_2 = \exp(\boldsymbol{\phi}_2^\wedge) \in SO(3)$

- when the axes of rotation are parallel, $\boldsymbol{\phi}_1 \parallel \boldsymbol{\phi}_2$, we have

$$\ln\left(\mathbf{C}_1\mathbf{C}_2\right)^\vee = \ln\left(\exp(\boldsymbol{\phi}_1^\wedge)\exp(\boldsymbol{\phi}_2^\wedge)\right)^\vee = \boldsymbol{\phi}_1 + \boldsymbol{\phi}_2 \tag{67}$$

- when one of the rotations can be considered small, we have

$$\begin{aligned}\ln\left(\mathbf{C}_1\mathbf{C}_2\right)^\vee &= \ln\left(\exp(\boldsymbol{\phi}_1^\wedge)\exp(\boldsymbol{\phi}_2^\wedge)\right)^\vee \\ &\approx \begin{cases} \mathbf{J}(\boldsymbol{\phi}_2)^{-1}\boldsymbol{\phi}_1 + \boldsymbol{\phi}_2 & \text{if } \boldsymbol{\phi}_1 \text{ small} \\ \boldsymbol{\phi}_1 + \mathbf{J}(-\boldsymbol{\phi}_1)^{-1}\boldsymbol{\phi}_2 & \text{if } \boldsymbol{\phi}_2 \text{ small} \end{cases}\end{aligned} \tag{68}$$

BCH poses

- in the particular case of $SE(3)$, we can show that

$$\begin{aligned}\ln\left(\mathbf{T}_1\mathbf{T}_2\right)^\vee &= \ln\left(\exp(\boldsymbol{\xi}_1^\wedge)\exp(\boldsymbol{\xi}_2^\wedge)\right)^\vee \\ &= \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 + \frac{1}{2}\boldsymbol{\xi}_1^\curlywedge\boldsymbol{\xi}_2 + \frac{1}{12}\boldsymbol{\xi}_1^\curlywedge\boldsymbol{\xi}_1^\curlywedge\boldsymbol{\xi}_2 + \frac{1}{12}\boldsymbol{\xi}_2^\curlywedge\boldsymbol{\xi}_2^\curlywedge\boldsymbol{\xi}_1 + \cdots \end{aligned} \tag{69}$$

  where $\mathbf{T}_1 = \exp(\boldsymbol{\xi}_1^\wedge), \mathbf{T}_2 = \exp(\boldsymbol{\xi}_2^\wedge) \in SE(3)$
- when the poses are parallel, $\boldsymbol{\xi}_1 \parallel \boldsymbol{\xi}_2$, we have

$$\ln\left(\mathbf{T}_1\mathbf{T}_2\right)^\vee = \ln\left(\exp(\boldsymbol{\xi}_1^\wedge)\exp(\boldsymbol{\xi}_2^\wedge)\right)^\vee = \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 \tag{70}$$

- when one of the poses can be considered small, we have

$$\begin{aligned}\ln\left(\mathbf{T}_1\mathbf{T}_2\right)^\vee &= \ln\left(\exp(\boldsymbol{\xi}_1^\wedge)\exp(\boldsymbol{\xi}_2^\wedge)\right)^\vee \\ &\approx \begin{cases} \boldsymbol{\mathcal{J}}(\boldsymbol{\xi}_2)^{-1}\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 & \text{if } \boldsymbol{\xi}_1 \text{ small} \\ \boldsymbol{\xi}_1 + \boldsymbol{\mathcal{J}}(-\boldsymbol{\xi}_1)^{-1}\boldsymbol{\xi}_2 & \text{if } \boldsymbol{\xi}_2 \text{ small} \end{cases}\end{aligned} \tag{71}$$

UNIVERSITY OF TORONTO

$SE(3)$ Jacobian

– the (left) Jacobian for $SE(3)$ is related to the one for $SO(3)$, $\mathbf{J}$:

$$\boldsymbol{\mathcal{J}}(\boldsymbol{\xi}) = \sum_{n=0}^{\infty} \frac{1}{(n+1)!} \left(\boldsymbol{\xi}^{\curlywedge}\right)^n = \begin{bmatrix} \mathbf{J} & \mathbf{Q} \\ \mathbf{0} & \mathbf{J} \end{bmatrix} \tag{72}$$

where $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}$ and

$$\mathbf{Q}(\boldsymbol{\xi}) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{1}{(n+m+2)!} \left(\boldsymbol{\phi}^\wedge\right)^n \boldsymbol{\rho}^\wedge \left(\boldsymbol{\phi}^\wedge\right)^m \tag{73a}$$

$$\begin{aligned} &= \frac{1}{2}\boldsymbol{\rho}^\wedge + \frac{\phi - \sin\phi}{\phi^3}\left(\boldsymbol{\phi}^\wedge\boldsymbol{\rho}^\wedge + \boldsymbol{\rho}^\wedge\boldsymbol{\phi}^\wedge + \boldsymbol{\phi}^\wedge\boldsymbol{\rho}^\wedge\boldsymbol{\phi}^\wedge\right) \\ &\quad - \frac{1 - \frac{\phi^2}{2} - \cos\phi}{\phi^4}\left(\boldsymbol{\phi}^\wedge\boldsymbol{\phi}^\wedge\boldsymbol{\rho}^\wedge + \boldsymbol{\rho}^\wedge\boldsymbol{\phi}^\wedge\boldsymbol{\phi}^\wedge - 3\boldsymbol{\phi}^\wedge\boldsymbol{\rho}^\wedge\boldsymbol{\phi}^\wedge\right) \\ &\quad - \frac{1}{2}\left(\frac{1 - \frac{\phi^2}{2} - \cos\phi}{\phi^4} - 3\,\frac{\phi - \sin\phi - \frac{\phi^3}{6}}{\phi^5}\right) \\ &\quad \times \left(\boldsymbol{\phi}^\wedge\boldsymbol{\rho}^\wedge\boldsymbol{\phi}^\wedge\boldsymbol{\phi}^\wedge + \boldsymbol{\phi}^\wedge\boldsymbol{\phi}^\wedge\boldsymbol{\rho}^\wedge\boldsymbol{\phi}^\wedge\right) \end{aligned} \tag{73b}$$

Compounding uncertain poses

– we were left with dealing with

$$\exp\left(\boldsymbol{\epsilon}^{\wedge}\right) = \exp\left(\boldsymbol{\epsilon}_1^{\wedge}\right)\exp\left(\left(\bar{\boldsymbol{\mathcal{T}}}_1\boldsymbol{\epsilon}_2\right)^{\wedge}\right) \tag{74}$$

– since both $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$ are small, we can severely approximate the BCH formula by choosing

$$\boldsymbol{\epsilon} \approx \boldsymbol{\epsilon}_1 + \bar{\boldsymbol{\mathcal{T}}}_1\boldsymbol{\epsilon}_2 \tag{75}$$

– then for the mean we confirm

$$E[\boldsymbol{\epsilon}] \approx \underbrace{E[\boldsymbol{\epsilon}_1]}_{\mathbf{0}} + \bar{\boldsymbol{\mathcal{T}}}_1 \underbrace{E[\boldsymbol{\epsilon}_2]}_{\mathbf{0}} = \mathbf{0} \tag{76}$$

which we already assumed by letting $\bar{\mathbf{T}} = \bar{\mathbf{T}}_1\bar{\mathbf{T}}_2$

UNIVERSITY OF TORONTO

Compounding uncertain poses

- for the covariance we have

$$\underbrace{E[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T]}_{\boldsymbol{\Sigma}} \approx E\left[\left(\boldsymbol{\epsilon}_1 + \bar{\boldsymbol{\mathcal{T}}}_1\boldsymbol{\epsilon}_2\right)\left(\boldsymbol{\epsilon}_1 + \bar{\boldsymbol{\mathcal{T}}}_1\boldsymbol{\epsilon}_2\right)^T\right]$$
$$= \underbrace{E[\boldsymbol{\epsilon}_1\boldsymbol{\epsilon}_1^T]}_{\boldsymbol{\Sigma}_1} + \underbrace{E[\boldsymbol{\epsilon}_1\boldsymbol{\epsilon}_2^T]}_{\mathbf{0}}\bar{\boldsymbol{\mathcal{T}}}_1^T + \bar{\boldsymbol{\mathcal{T}}}_1\underbrace{E[\boldsymbol{\epsilon}_2\boldsymbol{\epsilon}_1^T]}_{\mathbf{0}} + \bar{\boldsymbol{\mathcal{T}}}_1\underbrace{E[\boldsymbol{\epsilon}_2\boldsymbol{\epsilon}_2^T]}_{\boldsymbol{\Sigma}_2}\bar{\boldsymbol{\mathcal{T}}}_1^T \quad (77)$$

- finally, for the overall pose compounding, $\mathbf{T} = \mathbf{T}_1\mathbf{T}_2$ we have

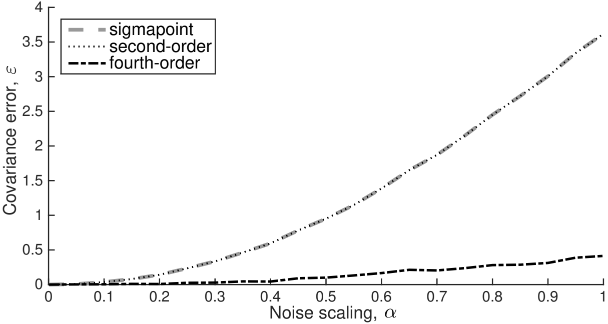
$$\left\{\bar{\mathbf{T}}, \boldsymbol{\Sigma}\right\} \approx \left\{\bar{\mathbf{T}}_1\bar{\mathbf{T}}_2, \boldsymbol{\Sigma}_1 + \bar{\boldsymbol{\mathcal{T}}}_1\boldsymbol{\Sigma}_2\bar{\boldsymbol{\mathcal{T}}}_1^T\right\} \quad (78)$$

where $\bar{\boldsymbol{\mathcal{T}}}_1 = \mathrm{Ad}\left(\bar{\mathbf{T}}_1\right)$

- more accurate compounding formulas can be worked out by choosing less severe approximations to the BCH formula

UNIVERSITY OF TORONTO

Compounding uncertain poses



- this shows how the error in our compounding formula increases with increasing uncertainty, α, on the input covariances
- our method is the 'second-order' one
- the 'fourth-order' one uses a better approximation to BCH

UNIVERSITY OF TORONTO

Compounding example

– consider the case of compounding transformations many times in a row:

$$\exp\left(\boldsymbol{\epsilon}_K^\wedge\right)\bar{\mathbf{T}}_K = \left(\prod_{k=1}^{K}\exp\left(\boldsymbol{\epsilon}^\wedge\right)\bar{\mathbf{T}}\right)\exp\left(\boldsymbol{\epsilon}_0^\wedge\right)\bar{\mathbf{T}}_0 \tag{79}$$

– make the following assumptions:

$$\bar{\mathbf{T}}_0 = \mathbf{1}, \quad \boldsymbol{\epsilon}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{0}) \tag{80a}$$

$$\bar{\mathbf{T}} = \begin{bmatrix} \bar{\mathbf{C}} & \bar{\mathbf{r}} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right) \tag{80b}$$

$$\bar{\mathbf{C}} = \mathbf{1}, \quad \bar{\mathbf{r}} = \begin{bmatrix} r \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{\Sigma} = \mathrm{diag}\left(0, 0, 0, 0, 0, \sigma^2\right) \tag{80c}$$

Compounding example

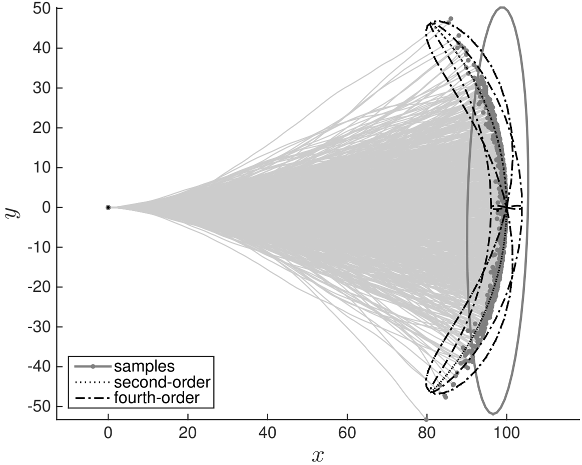
- using our compounding scheme we have

$$\bar{\mathbf{T}}_K = \begin{bmatrix} 1 & 0 & 0 & Kr \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{81a}$$

$$\boldsymbol{\Sigma}_K = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{K(K-1)(2K-1)}{6} r^2\sigma^2 & 0 & 0 & 0 & -\frac{K(K-1)}{2} r\sigma^2 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{K(K-1)}{2} r\sigma^2 & 0 & 0 & 0 & K\sigma^2 \end{bmatrix} \tag{81b}$$

UNIVERSITY OF TORONTO

Compounding example



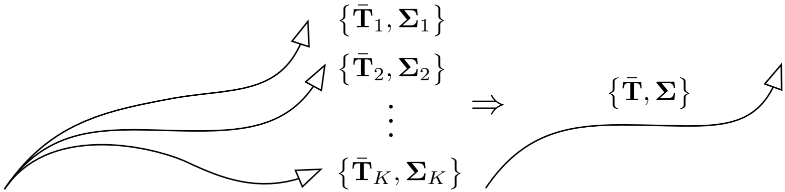
UNIVERSITY OF TORONTO

Fusing uncertain poses

- suppose that we have K estimates/measurements of a pose and associated uncertainties:

$$\left\{\bar{\mathbf{T}}_1, \boldsymbol{\Sigma}_1\right\}, \left\{\bar{\mathbf{T}}_2, \boldsymbol{\Sigma}_2\right\}, \ldots, \left\{\bar{\mathbf{T}}_K, \boldsymbol{\Sigma}_K\right\} \tag{82}$$

- how can we optimally fuse these into a single estimate, $\left\{\bar{\mathbf{T}}, \boldsymbol{\Sigma}\right\}$?



Fusing uncertain poses

- the vectorspace solution to fusion is straightforward and can be found exactly in closed form:

$$\boldsymbol{\Sigma}^{-1}\bar{\mathbf{x}} = \sum_{k=1}^{K} \boldsymbol{\Sigma}_k^{-1}\bar{\mathbf{x}}_k, \quad \boldsymbol{\Sigma}^{-1} = \sum_{k=1}^{K} \boldsymbol{\Sigma}_k^{-1} \tag{83}$$

- the situation is somewhat more complicated when dealing with $SE(3)$, and we shall resort to an approximate iterative scheme

- caution: here we will be using our Lie-group perturbations in both senses that we've learned about:
  - as updates within an optimization problem
  - to represent uncertainty in a probability sense

UNIVERSITY OF TORONTO

Fusing uncertain poses

- we define the error between the individual measurement and the optimal estimate, $\mathbf{T}$, as

$$\mathbf{e}_k(\mathbf{T}) = \ln\left(\bar{\mathbf{T}}_k \mathbf{T}^{-1}\right)^\vee \tag{84}$$

- we start with an initial guess, $\mathbf{T}_{\text{op}}$, and perturb this (on the left) by a small amount, $\boldsymbol{\epsilon}$ so that

$$\mathbf{T} = \exp\left(\boldsymbol{\epsilon}^\wedge\right)\mathbf{T}_{\text{op}} \tag{85}$$

- inserting this into the error expression we have

$$\begin{aligned}\mathbf{e}_k(\mathbf{T}) &= \ln\left(\bar{\mathbf{T}}_k \mathbf{T}^{-1}\right)^\vee = \ln\Big(\underbrace{\bar{\mathbf{T}}_k\,\mathbf{T}_{\text{op}}^{-1}}_{\text{small}}\exp\left(-\boldsymbol{\epsilon}^\wedge\right)\Big)^\vee \\ &= \ln\left(\exp\left(\mathbf{e}_k(\mathbf{T}_{\text{op}})^\wedge\right)\exp\left(-\boldsymbol{\epsilon}^\wedge\right)\right)^\vee \approx \mathbf{e}_k(\mathbf{T}_{\text{op}}) - \boldsymbol{\epsilon}\end{aligned} \tag{86}$$

where $\mathbf{e}_k(\mathbf{T}_{\text{op}}) = \ln\left(\bar{\mathbf{T}}_k \mathbf{T}_{\text{op}}^{-1}\right)^\vee$ and we used BCH to approximate

Fusing uncertain poses

- we define the cost function that we want to minimize as

$$J(\mathbf{T}) = \frac{1}{2}\sum_{k=1}^{K} \mathbf{e}_k(\mathbf{T})^T \,\boldsymbol{\Sigma}_k^{-1}\, \mathbf{e}_k(\mathbf{T})$$

$$\approx \frac{1}{2}\sum_{k=1}^{K} \left(\mathbf{e}_k(\mathbf{T}_{\mathrm{op}}) - \boldsymbol{\epsilon}\right)^T \boldsymbol{\Sigma}_k^{-1} \left(\mathbf{e}_k(\mathbf{T}_{\mathrm{op}}) - \boldsymbol{\epsilon}\right) \quad (87)$$

- take the derivative with respect to $\boldsymbol{\epsilon}$ and set to zero:

$$\left(\sum_{k=1}^{K} \boldsymbol{\Sigma}_k^{-1}\right) \boldsymbol{\epsilon}^\star = \sum_{k=1}^{K} \boldsymbol{\Sigma}_k^{-1}\, \mathbf{e}_k(\mathbf{T}_{\mathrm{op}}) \quad (88)$$

- we then apply this optimal perturbation to our current guess,

$$\mathbf{T}_{\mathrm{op}} \leftarrow \exp\left(\boldsymbol{\epsilon}^{\star\wedge}\right)\, \mathbf{T}_{\mathrm{op}} \quad (89)$$

which ensures $\mathbf{T}_{\mathrm{op}}$ remains in $SE(3)$, and iterate to convergence

UNIVERSITY OF TORONTO

Summary

- we learned about perturbations to matrix Lie groups, which happen in the Lie algebra, which is a vectorspace
- we used these perturbations to define iterative optimization schemes for rotations and poses; these schemes are advantageous because
  - we store our rotation/pose in a singularity-free format, $\mathbf{C}_{\text{op}}$ or $\mathbf{T}_{\text{op}}$
  - at each iteration we perform unconstrained optimization
  - our manipulations occur at the matrix level
- we also used perturbations to allow us to define Gaussian probability density functions for matrix Lie groups
  - the mean, $\bar{\mathbf{C}}$ or $\bar{\mathbf{T}}$, is defined the Lie group, which is singularity-free
  - the covariance, $\mathbf{\Sigma}$, is defined in the Lie algebra, which is a vectorspace
- using different approximations to the BCH formula, we can devise more/less accurate methods of manipulating matrix Lie group quantities

UNIVERSITY OF TORONTO