Lecture 11: Pose Estimation Problems

AER1513: State Estimation

Timothy D. Barfoot

University of Toronto

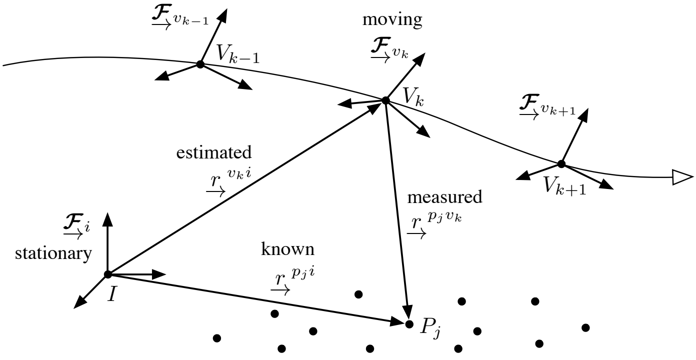
Outline

UNIVERSITY OF TORONTO

Motivation

- in the last lecture, we learned how to perturb rotations and poses using ideas from matrix Lie groups
- this lead to practical methods to perform optimization and represent uncertainty for rotations and poses
- we now want to use these ideas to adapt our state estimation algorithms to work with rotations and poses

General Setup



UNIVERSITY OF TORONTO

Problems to Consider

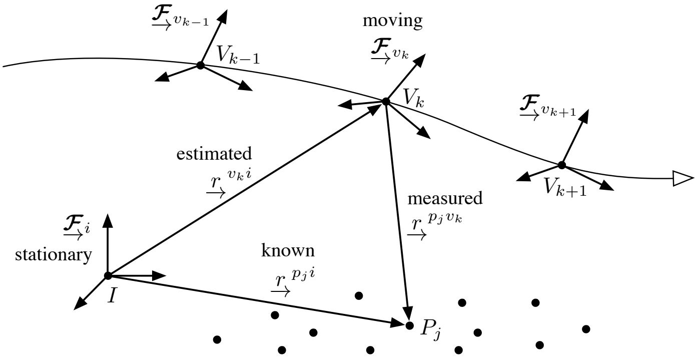
– there are two different problems we could consider with our general setup

point-cloud alignment : in this problem, we have two point-clouds, one expressed in the stationary frame and one in the moving frame (at a single time) and we want to know the pose change between the two frames by aligning the point-clouds

point-cloud tracking : in this problem, we want to estimate the pose of the moving frame with respect to the stationary frame over a longer period of time using one of the estimators from the first part of the course

UNIVERSITY OF TORONTO

Point-cloud alignment



– we will consider just a single time

– match a point-cloud in the moving frame to one in the stationary frame to get the pose

UNIVERSITY OF TORONTO

Point-cloud alignment

– we have two point-clouds, one expressed in the stationary frame and one in the moving frame (at a single time)
– we will use some simplified notation to avoid repeating sub- and super-scripts:

$$\mathbf{y}_j = \mathbf{r}_{v_k}^{p_j v_k}, \quad \mathbf{p}_j = \mathbf{r}_i^{p_j i}, \quad \mathbf{r} = \mathbf{r}_i^{v_k i}, \quad \mathbf{C} = \mathbf{C}_{v_k i} \tag{1}$$

– also, we define

$$\mathbf{y} = \frac{1}{w} \sum_{j=1}^{M} w_j \mathbf{y}_j, \quad \mathbf{p} = \frac{1}{w} \sum_{j=1}^{M} w_j \mathbf{p}_j, \quad w = \sum_{j=1}^{M} w_j \tag{2}$$

where the w_j are scalar weights for each point

UNIVERSITY OF TORONTO

Optimization problem

– we define an error term for each point:

$$\mathbf{e}_j = \mathbf{y}_j - \mathbf{C}(\mathbf{p}_j - \mathbf{r}) \tag{3}$$

– our estimation problem is then to globally minimize the cost function,

$$J(\mathbf{C}, \mathbf{r}) = \frac{1}{2}\sum_{j=1}^{M} w_j \mathbf{e}_j^T \mathbf{e}_j = \frac{1}{2}\sum_{j=1}^{M} w_j \left(\mathbf{y}_j - \mathbf{C}(\mathbf{p}_j - \mathbf{r})\right)^T \left(\mathbf{y}_j - \mathbf{C}(\mathbf{p}_j - \mathbf{r})\right) \tag{4}$$

subject to $\mathbf{C} \in SO(3)$ (i.e., $\mathbf{C}\mathbf{C}^T = \mathbf{1}$ and $\det \mathbf{C} = 1$)

– it turns out that it is possible to carry out this optimization in a one-shot (non-iterative) manner

UNIVERSITY OF TORONTO

Change of variables

- we will make a change of variables for the translation parameter:

$$\mathbf{d} = \mathbf{r} + \mathbf{C}^T\mathbf{y} - \mathbf{p} \tag{5}$$

  which is easy to isolate for $\mathbf{r}$ if all the other quantities are known
- in this case, we can rewrite our cost function as

$$J(\mathbf{C}, \mathbf{d}) = \underbrace{\frac{1}{2}\sum_{j=1}^{M} w_j \left((\mathbf{y}_j - \mathbf{y}) - \mathbf{C}(\mathbf{p}_j - \mathbf{p})\right)^T \left((\mathbf{y}_j - \mathbf{y}) - \mathbf{C}(\mathbf{p}_j - \mathbf{p})\right)}_{\text{depends only on } \mathbf{C}} + \underbrace{\frac{1}{2}\mathbf{d}^T\mathbf{d}}_{\text{depends only on } \mathbf{d}} \tag{6}$$

  which is the sum of two positive-definite terms, the first depending only on $\mathbf{C}$ and the second only on $\mathbf{d}$

Start to optimize

- we can minimize the term depending on $\mathbf{d}$ by taking $\mathbf{d} = \mathbf{0}$, which in turn implies that

$$\mathbf{r} = \mathbf{p} - \mathbf{C}^T \mathbf{y} \tag{7}$$

- if we multiply out each smaller term within the term that depends on $\mathbf{C}$, only one part actually depends on $\mathbf{C}$

$$\begin{aligned} &\left((\mathbf{y}_j - \mathbf{y}) - \mathbf{C}(\mathbf{p}_j - \mathbf{p})\right)^T \left((\mathbf{y}_j - \mathbf{y}) - \mathbf{C}(\mathbf{p}_j - \mathbf{p})\right) \\ = &\underbrace{(\mathbf{y}_j - \mathbf{y})^T(\mathbf{y}_j - \mathbf{y})}_{\text{independent of } \mathbf{C}} - 2\underbrace{\left((\mathbf{y}_j - \mathbf{y})^T \mathbf{C}(\mathbf{p}_j - \mathbf{p})\right)}_{\mathrm{tr}\left(\mathbf{C}(\mathbf{p}_j - \mathbf{p})(\mathbf{y}_j - \mathbf{y})^T\right)} + \underbrace{(\mathbf{p}_j - \mathbf{p})^T(\mathbf{p}_j - \mathbf{p})}_{\text{independent of } \mathbf{C}} \end{aligned} \tag{8}$$

- we can therefore replace the $\mathbf{C}$ term with

$$-\mathrm{tr}\left(\mathbf{C}\mathbf{W}^T\right), \quad \mathbf{W} = \frac{1}{w}\sum_{j=1}^{M} w_j(\mathbf{y}_j - \mathbf{y})(\mathbf{p}_j - \mathbf{p})^T \tag{9}$$

UNIVERSITY OF TORONTO

Introduce constraints

- we can define a new cost function that we seek to minimize with respect to $\mathbf{C}$ as

$$J(\mathbf{C}, \mathbf{\Lambda}, \gamma) = -\mathrm{tr}(\mathbf{C}\mathbf{W}^T) + \underbrace{\mathrm{tr}\left(\mathbf{\Lambda}(\mathbf{C}\mathbf{C}^T - \mathbf{1})\right) + \gamma(\det \mathbf{C} - 1)}_{\text{Lagrange multiplier terms}} \tag{10}$$

  where $\mathbf{\Lambda}$ and γ are Lagrange multipliers associated with the two terms on the right; these are used to ensure that the resulting $\mathbf{C} \in SO(3)$
- note, when $\mathbf{C}\mathbf{C}^T = \mathbf{1}$ and $\det \mathbf{C} = 1$, these terms have no effect on the resulting cost
- it is also worth noting that $\mathbf{\Lambda}$ is symmetric since we only need to enforce six orthogonality constraints
- this new cost function will be minimized by the same $\mathbf{C}$ as our original one

UNIVERSITY OF TORONTO

Optimize

– taking the derivative of $J(\mathbf{C}, \mathbf{\Lambda}, \gamma)$ with respect to $\mathbf{C}$, $\mathbf{\Lambda}$, and γ, we have

$$\begin{aligned}
\frac{\partial J}{\partial \mathbf{C}} &= -\mathbf{W} + 2\mathbf{\Lambda}\mathbf{C} + \gamma \underbrace{\det \mathbf{C}}_{1} \underbrace{\mathbf{C}^{-T}}_{\mathbf{C}} = -\mathbf{W} + \mathbf{L}\mathbf{C} && \text{(11a)} \\
\frac{\partial J}{\partial \mathbf{\Lambda}} &= \mathbf{C}\mathbf{C}^T - \mathbf{1} && \text{(11b)} \\
\frac{\partial J}{\partial \gamma} &= \det \mathbf{C} - 1 && \text{(11c)}
\end{aligned}$$

where we have lumped together the Lagrange multipliers as

$$\mathbf{L} = 2\mathbf{\Lambda} + \gamma \mathbf{1} \tag{12}$$

– setting the first equation to zero, we find that

$$\mathbf{L}\mathbf{C} = \mathbf{W} \tag{13}$$

Easy case

- if we could assume that $\mathbf{W} > 0$, we could postmultiply (13) by itself transposed to find

$$\mathbf{L}\underbrace{\mathbf{C}\mathbf{C}^T}_{\mathbf{1}}\mathbf{L}^T = \mathbf{W}\mathbf{W}^T \tag{14}$$

- since $\mathbf{L}$ is symmetric, we have that

$$\mathbf{L} = \left(\mathbf{W}\mathbf{W}^T\right)^{\frac{1}{2}} \tag{15}$$

  which we see involves a matrix square-root
- substituting this back into (13), the optimal rotation is

$$\mathbf{C} = \left(\mathbf{W}\mathbf{W}^T\right)^{-\frac{1}{2}}\mathbf{W} \tag{16}$$

- we are essentially projecting $\mathbf{W}$ onto $SO(3)$

UNIVERSITY OF TORONTO

Hard case

- if we cannot assume that $\mathbf{W} > 0$, things get complicated quickly
- the details are a bit messy, but it is still possible to work out the optimal rotation
- start by doing a singular value decomposition (SVD) on $\mathbf{W}$

$$\mathbf{W} = \mathbf{U}\mathbf{D}\mathbf{V}^T \tag{17}$$

- the optimal rotation is then

$$\mathbf{C} = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det \mathbf{U} \det \mathbf{V} \end{bmatrix} \mathbf{V}^T \tag{18}$$

- the main reason this approach is necessary is to ensure that $\det \mathbf{C} = 1$ instead of -1

UNIVERSITY OF TORONTO

Another approach using transformation matrices

– we can also use an iterative scheme to accomplish the same point-cloud alignment objective

– this is more inline with our general method of carrying out state estimation with rotations and poses

– we will again use some simplified notation to avoid repeating sub- and super-scripts:

$$\boldsymbol{y}_j = \begin{bmatrix} \mathbf{y}_j \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{v_k}^{p_j v_k} \\ 1 \end{bmatrix}, \quad \boldsymbol{p}_j = \begin{bmatrix} \mathbf{p}_j \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{r}_i^{p_j i} \\ 1 \end{bmatrix},$$

$$\mathbf{T} = \mathbf{T}_{v_k i} = \begin{bmatrix} \mathbf{C}_{v_k i} & -\mathbf{C}_{v_k i}\mathbf{r}_i^{v_k i} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{19}$$

UNIVERSITY OF TORONTO

Optimization problem

- we define our error term for each point as

$$\mathbf{e}_j = \boldsymbol{y}_j - \mathbf{T}\boldsymbol{p}_j \tag{20}$$

and our objective function as

$$J(\mathbf{T}) = \frac{1}{2}\sum_{j=1}^{M} w_j \mathbf{e}_j^T \mathbf{e}_j = \frac{1}{2}\sum_{j=1}^{M} w_j \left(\boldsymbol{y}_j - \mathbf{T}\boldsymbol{p}_j\right)^T \left(\boldsymbol{y}_j - \mathbf{T}\boldsymbol{p}_j\right) \tag{21}$$

where $w_j > 0$ are the usual scalar weights

- we seek to minimize J with respect to $\mathbf{T} \in SE(3)$

Avoiding constraints

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

$$\mathbf{T} = \exp\left(\boldsymbol{\epsilon}^\wedge\right)\mathbf{T}_{\text{op}} \approx \left(\mathbf{1} + \boldsymbol{\epsilon}^\wedge\right)\mathbf{T}_{\text{op}} \tag{22}$$

  where $\mathbf{T}_{\text{op}}$ is some initial guess (i.e., operating point of our linearization) and $\boldsymbol{\epsilon}$ is a small perturbation to that guess
- inserting this into the objective function we then have

$$J(\mathbf{T}) \approx \frac{1}{2}\sum_{j=1}^{M} w_j \left((\boldsymbol{y}_j - \boldsymbol{z}_j) - \boldsymbol{z}_j^\odot\,\boldsymbol{\epsilon}\right)^T \left((\boldsymbol{y}_j - \boldsymbol{z}_j) - \boldsymbol{z}_j^\odot\,\boldsymbol{\epsilon}\right) \tag{23}$$

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

$$\boldsymbol{\epsilon}^\wedge\,\boldsymbol{z}_j = \boldsymbol{z}_j^\odot\,\boldsymbol{\epsilon} \tag{24}$$

UNIVERSITY OF TORONTO

Optimize

- our objective function is now exactly quadratic in $\boldsymbol{\epsilon}$ and therefore we can carry out a simple, unconstrained optimization for $\boldsymbol{\epsilon}$
- taking the derivative we find

$$\frac{\partial J}{\partial \boldsymbol{\epsilon}^T} = -\sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^T} \left((\boldsymbol{y}_j - \boldsymbol{z}_j) - \boldsymbol{z}_j^{\odot} \boldsymbol{\epsilon} \right) \tag{25}$$

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

$$\left(\frac{1}{w} \sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^T} \boldsymbol{z}_j^{\odot} \right) \boldsymbol{\epsilon}^\star = \frac{1}{w} \sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^T} (\boldsymbol{y}_j - \boldsymbol{z}_j) \tag{26}$$

UNIVERSITY OF TORONTO

Improving efficiency

- to improve efficiency, we can write the left-hand side as

$$\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^T} \boldsymbol{z}_j^{\odot} = \underbrace{\boldsymbol{\mathcal{T}}_{\text{op}}^{-T}}_{>0} \underbrace{\left(\frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{p}_j^{\odot^T} \boldsymbol{p}_j^{\odot}\right)}_{\boldsymbol{\mathcal{M}}} \underbrace{\boldsymbol{\mathcal{T}}_{\text{op}}^{-1}}_{>0} \tag{27}$$

where

$$\boldsymbol{\mathcal{T}}_{\text{op}} = \text{Ad}(\mathbf{T}_{\text{op}}), \quad \boldsymbol{\mathcal{M}} = \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ -\mathbf{p}^\wedge & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{p}^\wedge \\ \mathbf{0} & \mathbf{1} \end{bmatrix},$$

$$w = \sum_{j=1}^{M} w_j, \quad \mathbf{p} = \frac{1}{w}\sum_{j=1}^{M} w_j \mathbf{p}_j, \quad \mathbf{I} = -\frac{1}{w}\sum_{j=1}^{M} w_j (\mathbf{p}_j - \mathbf{p})^\wedge (\mathbf{p}_j - \mathbf{p})^\wedge \tag{28}$$

- the 6×6 matrix, $\boldsymbol{\mathcal{M}}$, has the form of a generalized mass matrix with the weights as surrogates for masses; it is only a function of the points in the stationary frame and is therefore a constant

UNIVERSITY OF TORONTO

Improving efficiency

– looking to the right-hand side we can show

$$\mathbf{a} = \frac{1}{w}\sum_{j=1}^{M} w_j \boldsymbol{z}_j^{\odot^T}(\boldsymbol{y}_j - \boldsymbol{z}_j) = \begin{bmatrix} \mathbf{y} - \mathbf{C}_{\mathrm{op}}(\mathbf{p} - \mathbf{r}_{\mathrm{op}}) \\ \mathbf{b} - \mathbf{y}^{\wedge}\mathbf{C}_{\mathrm{op}}(\mathbf{p} - \mathbf{r}_{\mathrm{op}}) \end{bmatrix} \tag{29}$$

where

$$\mathbf{b} = \left[\mathrm{tr}\left(\mathbf{1}_i^{\wedge}\mathbf{C}_{\mathrm{op}}\mathbf{W}^T\right)\right]_i, \quad \mathbf{T}_{\mathrm{op}} = \begin{bmatrix} \mathbf{C}_{\mathrm{op}} & -\mathbf{C}_{\mathrm{op}}\mathbf{r}_{\mathrm{op}} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{30}$$

$$\mathbf{W} = \frac{1}{w}\sum_{j=1}^{M} w_j(\mathbf{y}_j - \mathbf{y})(\mathbf{p}_j - \mathbf{p})^T, \quad \mathbf{y} = \frac{1}{w}\sum_{j-1}^{M} w_j \mathbf{y}_j \tag{31}$$

– both $\mathbf{W}$ and $\mathbf{y}$ we have seen before and can be computed in advance from the points and then used at each iteration

UNIVERSITY OF TORONTO

Improving efficiency

- using these efficient forms, we can write the solution for the optimal update down in closed form:

$$\boldsymbol{\epsilon}^\star = \boldsymbol{\mathcal{T}}_{\mathrm{op}} \boldsymbol{\mathcal{M}}^{-1} \boldsymbol{\mathcal{T}}_{\mathrm{op}}^T \, \mathbf{a} \tag{32}$$

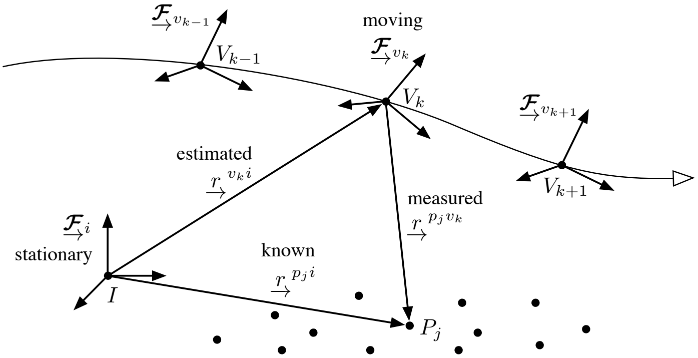
- once computed, we simply update our operating point,

$$\mathbf{T}_{\mathrm{op}} \leftarrow \exp\left(\boldsymbol{\epsilon}^{\star\wedge}\right) \mathbf{T}_{\mathrm{op}} \tag{33}$$

  and iterate the procedure to convergence
- note, applying the optimal perturbation through the exponential map ensures that $\mathbf{T}_{\mathrm{op}}$ remains in $SE(3)$ at each iteration
- we can see that our iterative optimization of $\mathbf{T}$ is exactly in the form of a Gauss-Newton style estimator, but adapted to work with $SE(3)$

UNIVERSITY OF TORONTO

Point-cloud tracking



- now we want to consider a longer interval of time
- this is essentially a localization problem; the stationary point-cloud is our map

UNIVERSITY OF TORONTO

Problem setup

- the state we want to estimate is the entire trajectory of poses:

$$\mathbf{x} = \{\mathbf{T}_0, \mathbf{T}_1, \ldots, \mathbf{T}_K\}, \quad \mathbf{T}_k = \mathbf{T}_{v_k i} = \begin{bmatrix} \mathbf{C}_{v_k i} & -\mathbf{C}_{v_k i}\mathbf{r}_i^{v_k i} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{34}$$

- the inputs (including the initial state) are

$$\mathbf{v} = \left\{\check{\mathbf{T}}_0, \boldsymbol{\varpi}_1, \boldsymbol{\varpi}_2, \ldots, \boldsymbol{\varpi}_K\right\} \tag{35}$$

  where $\boldsymbol{\varpi}_k$ is a body-fixed six-degree-of-freedom velocity
- the measurements are

$$\mathbf{y} = \{\mathbf{y}_{11}, \ldots, \mathbf{y}_{M1}, \ldots, \mathbf{y}_{1K}, \ldots \mathbf{y}_{MK}\} \tag{36}$$

  where $\mathbf{y}_{jk} = \mathbf{r}_{v_k}^{p_j v_k}$ is the observation of point P_j at time k

UNIVERSITY OF TORONTO

Motion model

- in continuous time our motion model is

$$\dot{\mathbf{T}} = \boldsymbol{\varpi}^\wedge \mathbf{T} \tag{37}$$

where the quantities involved are perturbed by process noise according to

$$\mathbf{T} = \exp\left(\delta\boldsymbol{\xi}^\wedge\right)\bar{\mathbf{T}} \tag{38a}$$

$$\boldsymbol{\varpi} = \bar{\boldsymbol{\varpi}} + \delta\boldsymbol{\varpi} \tag{38b}$$

- we can separate these into nominal and perturbation kinematics:

$$\text{nominal kinematics:} \quad \dot{\bar{\mathbf{T}}} = \bar{\boldsymbol{\varpi}}^\wedge \bar{\mathbf{T}} \tag{39a}$$

$$\text{perturbation kinematics:} \quad \delta\dot{\boldsymbol{\xi}} = \bar{\boldsymbol{\varpi}}^\curlywedge \delta\boldsymbol{\xi} + \delta\boldsymbol{\varpi} \tag{39b}$$

UNIVERSITY OF TORONTO

Motion model

– if we assume quantities remain constant between discrete times, then we can write

$$\text{nominal kinematics:} \quad \bar{\mathbf{T}}_k = \underbrace{\exp\left(\Delta t_k \bar{\boldsymbol{\varpi}}_k^{\wedge}\right)}_{\boldsymbol{\Xi}_k} \bar{\mathbf{T}}_{k-1} \tag{40a}$$

$$\text{perturbation kinematics:} \quad \delta\boldsymbol{\xi}_k = \underbrace{\exp\left(\Delta t_k \bar{\boldsymbol{\varpi}}_k^{\curlywedge}\right)}_{\mathrm{Ad}(\boldsymbol{\Xi}_k)} \delta\boldsymbol{\xi}_{k-1} + \mathbf{w}_k \tag{40b}$$

with $\Delta t_k = t_k - t_{k-1}$ for the nominal and perturbation kinematics in discrete time

– the process noise is now $\mathbf{w}_k = \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$

UNIVERSITY OF TORONTO

Measurement model

– our 3×1 measurement model can be compactly written as

$$\mathbf{y}_{jk} = \mathbf{D}^T \, \mathbf{T}_k \mathbf{p}_j + \mathbf{n}_{jk} \tag{41}$$

where the position of the known points on the moving vehicle are expressed in 4×1 homogeneous coordinates,

$$\mathbf{p}_j = \begin{bmatrix} \mathbf{r}_i^{p_j i} \\ 1 \end{bmatrix}, \quad \mathbf{D}^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{42}$$

where $\mathbf{D}^T$ is a projection matrix used to ensure the measurements are indeed 3×1 by removing the 1 on the bottom row

– we have also now included, $\mathbf{n}_{jk} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{jk})$, which is Gaussian measurement noise

UNIVERSITY OF TORONTO

Measurement model

- we linearize the measurement model using our perturbations:

$$\mathbf{T}_k = \exp\left(\delta\boldsymbol{\xi}_k^\wedge\right)\bar{\mathbf{T}}_k \quad (43a)$$

$$\mathbf{y}_{jk} = \bar{\mathbf{y}}_{jk} + \delta\mathbf{y}_{jk} \quad (43b)$$

- substituting these in we have

$$\bar{\mathbf{y}}_{jk} + \delta\mathbf{y}_{jk} = \mathbf{D}^T\left(\exp\left(\delta\boldsymbol{\xi}_k^\wedge\right)\bar{\mathbf{T}}_k\right)\mathbf{p}_j + \mathbf{n}_{jk} \quad (44)$$

- subtracting off the nominal solution (i.e., the operating point in our linearization),

$$\bar{\mathbf{y}}_{jk} = \mathbf{D}^T\,\bar{\mathbf{T}}_k\mathbf{p}_j \quad (45)$$

we are left with

$$\delta\mathbf{y}_{jk} \approx \mathbf{D}^T\left(\bar{\mathbf{T}}_k\mathbf{p}_j\right)^\odot\delta\boldsymbol{\xi}_k + \mathbf{n}_{jk} \quad (46)$$

UNIVERSITY OF TORONTO

EKF point-cloud tracking

– we now work out the details of carrying out point-cloud tracking using the extended Kalman filter, starting with the prediction step

– predicting the mean forwards in time is not difficult in the case of the EKF; we simply pass our prior estimate and latest input through the nominal kinematics model:

$$\check{\mathbf{T}}_k = \underbrace{\exp\left(\Delta t_k \, \boldsymbol{\varpi}_k^\wedge\right)}_{\boldsymbol{\Xi}_k} \hat{\mathbf{T}}_{k-1} \tag{47}$$

UNIVERSITY OF TORONTO

EKF prediction step

– to predict the covariance of the estimate,

$$\check{\mathbf{P}}_k = E\left[\delta\check{\boldsymbol{\xi}}_k \delta\check{\boldsymbol{\xi}}_k^T\right] \tag{48}$$

we require the perturbation kinematics model,

$$\delta\check{\boldsymbol{\xi}}_k = \underbrace{\exp\left(\Delta t_k \, \boldsymbol{\varpi}_k^{\curlywedge}\right)}_{\mathbf{F}_{k-1} = \mathrm{Ad}(\boldsymbol{\Xi}_k)} \delta\hat{\boldsymbol{\xi}}_{k-1} + \mathbf{w}_k \tag{49}$$

– thus, in this case the coefficient matrix of the linearized motion model is

$$\mathbf{F}_{k-1} = \exp\left(\Delta t_k \, \boldsymbol{\varpi}_k^{\curlywedge}\right) \tag{50}$$

which depends only on the input due to our convenient choice of representing uncertainty via the exponential map

– the covariance prediction proceeds in the usual EKF manner as

$$\check{\mathbf{P}}_k = \mathbf{F}_{k-1} \hat{\mathbf{P}}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_k \tag{51}$$

UNIVERSITY OF TORONTO

EKF correction step

– looking back to the perturbation measurement model,

$$\delta \mathbf{y}_{jk} = \underbrace{\mathbf{D}^T \left(\check{\mathbf{T}}_k \mathbf{p}_j \right)^{\odot}}_{\mathbf{G}_{jk}} \delta \check{\boldsymbol{\xi}}_k + \mathbf{n}_{jk} \qquad (52)$$

we see that the coefficient matrix of the linearized measurement model is

$$\mathbf{G}_{jk} = \mathbf{D}^T \left(\check{\mathbf{T}}_k \mathbf{p}_j \right)^{\odot} \qquad (53)$$

which is evaluated at the predicted mean pose, $\check{\mathbf{T}}_k$

– to handle the case in which there are M observations of points on the vehicle, we can stack the quantities as follows:

$$\mathbf{y}_k = \begin{bmatrix} \mathbf{y}_{1k} \\ \vdots \\ \mathbf{y}_{Mk} \end{bmatrix}, \quad \mathbf{G}_k = \begin{bmatrix} \mathbf{G}_{1k} \\ \vdots \\ \mathbf{G}_{Mk} \end{bmatrix}, \quad \mathbf{R}_k = \mathrm{diag}\left(\mathbf{R}_{1k}, \ldots, \mathbf{R}_{Mk} \right) \qquad (54)$$

UNIVERSITY OF TORONTO

EKF correction step

- the Kalman gain and covariance update equations are then unchanged from the generic case:

$$\mathbf{K}_k = \check{\mathbf{P}}_k \mathbf{G}_k^T \left(\mathbf{G}_k \check{\mathbf{P}}_k \mathbf{G}_k^T + \mathbf{R}_k \right)^{-1} \tag{55a}$$

$$\hat{\mathbf{P}}_k = \left(\mathbf{1} - \mathbf{K}_k \mathbf{G}_k \right) \check{\mathbf{P}}_k \tag{55b}$$

- note, we must be careful to interpret the EKF corrective equations properly since

$$\hat{\mathbf{P}}_k = E \left[\delta \hat{\boldsymbol{\xi}}_k \delta \hat{\boldsymbol{\xi}}_k^T \right] \tag{56}$$

UNIVERSITY OF TORONTO

EKF correction step

- for the mean update we rearrange the equation as follows:

$$\boldsymbol{\epsilon}_k = \underbrace{\ln\left(\hat{\mathbf{T}}_k \check{\mathbf{T}}_k^{-1}\right)^\vee}_{\text{update}} = \mathbf{K}_k \underbrace{(\mathbf{y}_k - \check{\mathbf{y}}_k)}_{\text{innovation}} \tag{57}$$

  where $\boldsymbol{\epsilon}_k$ is the difference of the corrected and predicted means and $\check{\mathbf{y}}_k$ is the measurement model evaluated at the predicted mean:

$$\check{\mathbf{y}}_k = \begin{bmatrix} \check{\mathbf{y}}_{1k} \\ \vdots \\ \check{\mathbf{y}}_{Mk} \end{bmatrix}, \qquad \check{\mathbf{y}}_{jk} = \mathbf{D}^T \check{\mathbf{T}}_k \mathbf{p}_j \tag{58}$$

- we apply the mean correction, $\boldsymbol{\epsilon}_k$, according to

$$\hat{\mathbf{T}}_k = \exp\left(\boldsymbol{\epsilon}_k^\wedge\right) \check{\mathbf{T}}_k \tag{59}$$

  which ensures the mean stays in $SE(3)$

EKF summary

– putting all the pieces together, the EKF equations are

$$
\begin{array}{rrcll}
\text{predictor:} & \check{\mathbf{P}}_k & = & \mathbf{F}_{k-1}\hat{\mathbf{P}}_{k-1}\mathbf{F}_{k-1}^T + \mathbf{Q}_k & (60a) \\
 & \check{\mathbf{T}}_k & = & \mathbf{\Xi}_k \, \hat{\mathbf{T}}_{k-1} & (60b) \\
\text{Kalman gain:} & \mathbf{K}_k & = & \check{\mathbf{P}}_k\mathbf{G}_k^T \left(\mathbf{G}_k\check{\mathbf{P}}_k\mathbf{G}_k^T + \mathbf{R}_k\right)^{-1} & (60c) \\
\text{corrector:} & \hat{\mathbf{P}}_k & = & \left(\mathbf{1} - \mathbf{K}_k\mathbf{G}_k\right)\check{\mathbf{P}}_k & (60d) \\
 & \hat{\mathbf{T}}_k & = & \exp\left(\left(\mathbf{K}_k\left(\mathbf{y}_k - \check{\mathbf{y}}_k\right)\right)^\wedge\right)\check{\mathbf{T}}_k & (60e)
\end{array}
$$

– we have essentially modified the EKF so that all the mean calculations occur in $SE(3)$, the Lie group, and all of the covariance calculations occur in $\mathfrak{se}(3)$, the Lie algebra

– as usual, we must initialize the filter at the first timestep using $\check{\mathbf{T}}_0$

UNIVERSITY OF TORONTO

Batch MAP

- we can also set up point-cloud tracking as a batch MAP estimation problem for state $\mathbf{x} = \{\mathbf{T}_0, \ldots, \mathbf{T}_K\}$
- as usual, we begin by defining an error term for each of our inputs and measurements
- for the inputs, $\check{\mathbf{T}}_0$ and $\boldsymbol{\varpi}_k$, we have

$$\mathbf{e}_{v,k}(\mathbf{x}) = \begin{cases} \ln\left(\check{\mathbf{T}}_0 \mathbf{T}_0^{-1}\right)^\vee & k = 0 \\ \ln\left(\boldsymbol{\Xi}_k \mathbf{T}_{k-1} \mathbf{T}_k^{-1}\right)^\vee & k = 1 \ldots K \end{cases} \tag{61}$$

  where $\boldsymbol{\Xi}_k = \exp\left(\Delta t_k \, \boldsymbol{\varpi}_k^\wedge\right)$
- for the measurements, $\mathbf{y}_{jk}$, we have

$$\mathbf{e}_{y,jk}(\mathbf{x}) = \mathbf{y}_{jk} - \mathbf{D}^T \mathbf{T}_k \mathbf{p}_j \tag{62}$$

UNIVERSITY OF TORONTO

Error analysis

- next, we examine the noise properties of these errors so that we know how much to weight them by in our objective function
- taking the Bayesian point of view, we consider that the true pose variables are drawn from the prior so that

$$\mathbf{T}_k = \exp\left(\delta\boldsymbol{\xi}_k^\wedge\right)\check{\mathbf{T}}_k \tag{63}$$

where $\delta\boldsymbol{\xi}_k \sim \mathcal{N}\left(\mathbf{0}, \check{\mathbf{P}}_k\right)$

- for the first input (initial state) error, we have

$$\mathbf{e}_{v,0}(\mathbf{x}) = \ln\left(\check{\mathbf{T}}_0\mathbf{T}_0^{-1}\right)^\vee = \ln\left(\check{\mathbf{T}}_0\check{\mathbf{T}}_0^{-1}\exp\left(-\delta\boldsymbol{\xi}_0^\wedge\right)\right)^\vee = -\delta\boldsymbol{\xi}_0 \tag{64}$$

so that

$$\mathbf{e}_{v,0}(\mathbf{x}) \sim \mathcal{N}\left(\mathbf{0}, \check{\mathbf{P}}_0\right) \tag{65}$$

Error analysis

- for the later input errors, we have

$$
\begin{aligned}
\mathbf{e}_{v,k}(\mathbf{x}) &= \ln\left(\boldsymbol{\Xi}_k \mathbf{T}_{k-1}\mathbf{T}_k^{-1}\right)^{\vee} \\
&= \ln\left(\boldsymbol{\Xi}_k \exp\left(\delta\boldsymbol{\xi}_{k-1}^{\wedge}\right)\check{\mathbf{T}}_{k-1}\check{\mathbf{T}}_k^{-1}\exp\left(-\delta\boldsymbol{\xi}_k^{\wedge}\right)\right)^{\vee} \\
&= \ln\Bigg(\underbrace{\boldsymbol{\Xi}_k \check{\mathbf{T}}_{k-1}\check{\mathbf{T}}_k^{-1}}_{\mathbf{1}}\exp\left(\left(\mathrm{Ad}(\boldsymbol{\Xi}_k)\,\delta\boldsymbol{\xi}_{k-1}\right)^{\wedge}\right)\exp\left(-\delta\boldsymbol{\xi}_k^{\wedge}\right)\Bigg)^{\vee} \\
&\approx \mathrm{Ad}(\boldsymbol{\Xi}_k)\,\delta\boldsymbol{\xi}_{k-1} - \delta\boldsymbol{\xi}_k \\
&= -\mathbf{w}_k
\end{aligned} \tag{66}
$$

so that

$$
\mathbf{e}_{v,k}(\mathbf{x}) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{Q}_k\right) \tag{67}
$$

UNIVERSITY OF TORONTO

Error analysis

– for the measurement model, we consider that the measurements are generated by evaluating the noise-free versions (based on the true pose variables) and then corrupted by noise so that

$$\mathbf{e}_{y,jk}(\mathbf{x}) = \mathbf{y}_{jk} - \mathbf{D}^T \mathbf{T}_k \mathbf{p}_j = \mathbf{n}_{jk} \tag{68}$$

so that

$$\mathbf{e}_{y,jk}(\mathbf{x}) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{jk}\right) \tag{69}$$

UNIVERSITY OF TORONTO

Objective function

- we can now construct the squared-error terms:

$$J_{v,k}(\mathbf{x}) = \begin{cases} \frac{1}{2}\mathbf{e}_{v,0}(\mathbf{x})^T \check{\mathbf{P}}_0^{-1} \mathbf{e}_{v,0}(\mathbf{x}) & k = 0 \\ \frac{1}{2}\mathbf{e}_{v,k}(\mathbf{x})^T \mathbf{Q}_k^{-1} \mathbf{e}_{v,k}(\mathbf{x}) & k = 1 \ldots K \end{cases} \tag{70a}$$

$$J_{y,k}(\mathbf{x}) = \frac{1}{2}\mathbf{e}_{y,k}(\mathbf{x})^T \mathbf{R}_k^{-1} \mathbf{e}_{y,k}(\mathbf{x}) \tag{70b}$$

where we have stacked the M point quantities together:

$$\mathbf{e}_{y,k}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_{y,1k}(\mathbf{x}) \\ \vdots \\ \mathbf{e}_{y,Mk}(\mathbf{x}) \end{bmatrix}, \qquad \mathbf{R}_k = \mathrm{diag}\left(\mathbf{R}_{1k}, \ldots, \mathbf{R}_{Mk}\right) \tag{71}$$

- the overall objective function that we will seek to minimize is then

$$J(\mathbf{x}) = \sum_{k=0}^{K} \left(J_{v,k}(\mathbf{x}) + J_{y,k}(\mathbf{x}) \right) \tag{72}$$

UNIVERSITY OF TORONTO

Linearized error terms

- it is fairly straightforward to linearize our error terms (in order to carry out Gauss-Newton optimization) just as we earlier linearized our motion and observation models
- we will linearize about an operating point for each pose, $\mathbf{T}_{\text{op},k}$, which we can think of as our current trajectory guess that will be iteratively improved
- thus, we will take

$$\mathbf{T}_k = \exp\left(\boldsymbol{\epsilon}_k^\wedge\right)\mathbf{T}_{\text{op},k} \tag{73}$$

where $\boldsymbol{\epsilon}_k$ will be the perturbation to the current guess that we seek to optimize at each iteration

- we will use the shorthand

$$\mathbf{x}_{\text{op}} = \left\{\mathbf{T}_{\text{op},1}, \mathbf{T}_{\text{op},2}, \ldots, \mathbf{T}_{\text{op},K}\right\} \tag{74}$$

for the operating point of the entire trajectory

UNIVERSITY OF TORONTO

Linearized error terms

- for the first input error, we have

$$\mathbf{e}_{v,0}(\mathbf{x}) = \ln\left(\check{\mathbf{T}}_0 \mathbf{T}_0^{-1}\right)^{\vee} = \ln\Big(\underbrace{\check{\mathbf{T}}_0 \mathbf{T}_{\text{op},0}^{-1}}_{\exp(\mathbf{e}_{v,0}(\mathbf{x}_{\text{op}})^{\wedge})} \exp\left(-\boldsymbol{\epsilon}_0^{\wedge}\right)\Big)^{\vee}$$
$$\approx \mathbf{e}_{v,0}(\mathbf{x}_{\text{op}}) - \underbrace{\boldsymbol{\mathcal{J}}\left(-\mathbf{e}_{v,0}(\mathbf{x}_{\text{op}})\right)^{-1}}_{\mathbf{E}_0} \boldsymbol{\epsilon}_0 \quad (75)$$

  where $\mathbf{e}_{v,0}(\mathbf{x}_{\text{op}}) = \ln\left(\check{\mathbf{T}}_0 \mathbf{T}_{\text{op},0}^{-1}\right)^{\vee}$ is the error evaluated at the operating point

- note, we have used a version of the BCH formula to arrive at the approximation on the right (i.e., assumes the perturbation is small), but this approximation will get better as $\boldsymbol{\epsilon}_0$ goes to zero, which will happen as the Gauss-Newton algorithm converges

UNIVERSITY OF TORONTO

Linearized error terms

- for the later input errors, we have

$$
\begin{aligned}
\mathbf{e}_{v,k}(\mathbf{x}) &= \ln\left(\boldsymbol{\Xi}_k \mathbf{T}_{k-1}\mathbf{T}_k^{-1}\right)^\vee \\
&= \ln\left(\boldsymbol{\Xi}_k \exp\left(\boldsymbol{\epsilon}_{k-1}^\wedge\right)\mathbf{T}_{\mathrm{op},k-1}\mathbf{T}_{\mathrm{op},k}^{-1}\exp\left(-\boldsymbol{\epsilon}_k^\wedge\right)\right)^\vee \\
&= \ln\Bigg(\underbrace{\boldsymbol{\Xi}_k \mathbf{T}_{\mathrm{op},k-1}\mathbf{T}_{\mathrm{op},k}^{-1}}_{\exp(\mathbf{e}_{v,k}(\mathbf{x}_{\mathrm{op}})^\wedge)}\exp\left(\left(\mathrm{Ad}\left(\mathbf{T}_{\mathrm{op},k}\mathbf{T}_{\mathrm{op},k-1}^{-1}\right)\boldsymbol{\epsilon}_{k-1}\right)^\wedge\right) \\
&\qquad\qquad \times \exp\left(-\boldsymbol{\epsilon}_k^\wedge\right)\Bigg)^\vee \\
&\approx \mathbf{e}_{v,k}(\mathbf{x}_{\mathrm{op}}) + \underbrace{\boldsymbol{\mathcal{J}}\left(-\mathbf{e}_{v,k}(\mathbf{x}_{\mathrm{op}})\right)^{-1}\mathrm{Ad}\left(\mathbf{T}_{\mathrm{op},k}\mathbf{T}_{\mathrm{op},k-1}^{-1}\right)}_{\mathbf{F}_{k-1}}\boldsymbol{\epsilon}_{k-1} \\
&\qquad\qquad - \underbrace{\boldsymbol{\mathcal{J}}\left(-\mathbf{e}_{v,k}(\mathbf{x}_{\mathrm{op}})\right)^{-1}}_{\mathbf{E}_k}\boldsymbol{\epsilon}_k \qquad (76)
\end{aligned}
$$

where $\mathbf{e}_{v,k}(\mathbf{x}_{\mathrm{op}}) = \ln\left(\boldsymbol{\Xi}_k \mathbf{T}_{\mathrm{op},k-1}\mathbf{T}_{\mathrm{op},k}^{-1}\right)^\vee$ is the error evaluated at the operating point

UNIVERSITY OF TORONTO

Linearized error terms

– for the measurement errors, we have

$$\begin{aligned} \mathbf{e}_{y,jk}(\mathbf{x}) &= \mathbf{y}_{jk} - \mathbf{D}^T \mathbf{T}_k \mathbf{p}_j \\ &= \mathbf{y}_{jk} - \mathbf{D}^T \exp\left(\boldsymbol{\epsilon}_k^\wedge\right) \mathbf{T}_{\mathrm{op},k} \mathbf{p}_j \\ &\approx \mathbf{y}_{jk} - \mathbf{D}^T \left(\mathbf{1} + \boldsymbol{\epsilon}_k^\wedge\right) \mathbf{T}_{\mathrm{op},k} \mathbf{p}_j \\ &= \underbrace{\mathbf{y}_{jk} - \mathbf{D}^T \mathbf{T}_{\mathrm{op},k} \mathbf{p}_j}_{\mathbf{e}_{y,jk}(\mathbf{x}_{\mathrm{op}})} - \underbrace{\left(\mathbf{D}^T \left(\mathbf{T}_{\mathrm{op},k} \mathbf{p}_j\right)^\odot\right)}_{\mathbf{G}_{jk}} \boldsymbol{\epsilon}_k \end{aligned} \tag{77}$$

– we can stack all of the point measurement errors at time k together so that

$$\mathbf{e}_{y,k}(\mathbf{x}) \approx \mathbf{e}_{y,k}(\mathbf{x}_{\mathrm{op}}) - \mathbf{G}_k \boldsymbol{\epsilon}_k \tag{78}$$

where

$$\mathbf{e}_{y,k}(\mathbf{x}) = \begin{bmatrix} \mathbf{e}_{y,1k}(\mathbf{x}) \\ \vdots \\ \mathbf{e}_{y,Mk}(\mathbf{x}) \end{bmatrix}, \quad \mathbf{e}_{y,k}(\mathbf{x}_{\mathrm{op}}) = \begin{bmatrix} \mathbf{e}_{y,1k}(\mathbf{x}_{\mathrm{op}}) \\ \vdots \\ \mathbf{e}_{y,Mk}(\mathbf{x}_{\mathrm{op}}) \end{bmatrix}, \quad \mathbf{G}_k = \begin{bmatrix} \mathbf{G}_{1k} \\ \vdots \\ \mathbf{G}_{Mk} \end{bmatrix} \tag{79}$$

Gauss-Newton update

- to set up the Gauss-Newton update, we define the following stacked quantities:

$$\delta\mathbf{x} = \begin{bmatrix} \boldsymbol{\epsilon}_0 \\ \boldsymbol{\epsilon}_1 \\ \boldsymbol{\epsilon}_2 \\ \vdots \\ \boldsymbol{\epsilon}_K \end{bmatrix}, \quad \mathbf{H} = \left[\begin{array}{ccccc} \mathbf{E}_0 & & & & \\ -\mathbf{F}_0 & \mathbf{E}_1 & & & \\ & -\mathbf{F}_1 & \ddots & & \\ & & \ddots & \mathbf{E}_{K-1} & \\ & & & -\mathbf{F}_{K-1} & \mathbf{E}_K \\ \hline \mathbf{G}_0 & & & & \\ & \mathbf{G}_1 & & & \\ & & \mathbf{G}_2 & & \\ & & & \ddots & \\ & & & & \mathbf{G}_K \end{array}\right], \quad \mathbf{e}(\mathbf{x}_{\text{op}}) = \left[\begin{array}{c} \mathbf{e}_{v,0}(\mathbf{x}_{\text{op}}) \\ \mathbf{e}_{v,1}(\mathbf{x}_{\text{op}}) \\ \vdots \\ \mathbf{e}_{v,K-1}(\mathbf{x}_{\text{op}}) \\ \mathbf{e}_{v,K}(\mathbf{x}_{\text{op}}) \\ \hline \mathbf{e}_{y,0}(\mathbf{x}_{\text{op}}) \\ \mathbf{e}_{y,1}(\mathbf{x}_{\text{op}}) \\ \vdots \\ \mathbf{e}_{y,K}(\mathbf{x}_{\text{op}}) \end{array}\right] \tag{80}$$

and

$$\mathbf{W} = \text{diag}\left(\check{\mathbf{P}}_0, \mathbf{Q}_1, \ldots, \mathbf{Q}_K, \mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_K\right) \tag{81}$$

which are identical to the matrices in the nonlinear version

UNIVERSITY OF TORONTO

Gauss-Newton update

- the quadratic (in terms of the perturbation, $\delta\mathbf{x}$) approximation to the objective function is then

$$J(\mathbf{x}) \approx J(\mathbf{x}_{\mathrm{op}}) - \mathbf{b}^T \, \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T \mathbf{A} \, \delta\mathbf{x} \tag{82}$$

where

$$\mathbf{A} = \underbrace{\mathbf{H}^T\mathbf{W}^{-1}\mathbf{H}}_{\text{block-tridiagonal}}, \quad \mathbf{b} = \mathbf{H}^T\mathbf{W}^{-1}\mathbf{e}(\mathbf{x}_{\mathrm{op}}) \tag{83}$$

- minimizing with respect to $\delta\mathbf{x}$, we have

$$\mathbf{A} \, \delta\mathbf{x}^\star = \mathbf{b} \tag{84}$$

for the optimal perturbation,

$$\delta\mathbf{x}^\star = \begin{bmatrix} \boldsymbol{\epsilon}_0^\star \\ \boldsymbol{\epsilon}_1^\star \\ \vdots \\ \boldsymbol{\epsilon}_K^\star \end{bmatrix} \tag{85}$$

UNIVERSITY OF TORONTO

Gauss-Newton update

- once we have the optimal perturbation, we update our operating point through the original perturbation scheme,

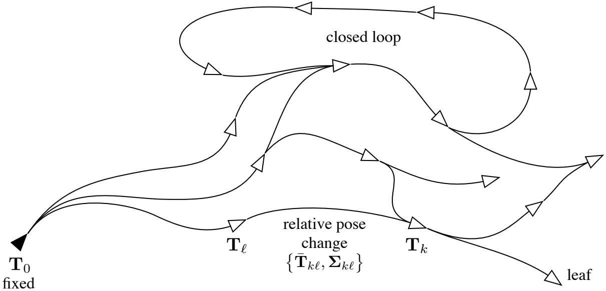
$$\mathbf{T}_{\text{op},k} \leftarrow \exp\left(\boldsymbol{\epsilon}_k^{\star\wedge}\right)\mathbf{T}_{\text{op},k} \tag{86}$$

  which ensures that $\mathbf{T}_{\text{op},k}$ stays in $SE(3)$
- we then iterate the entire scheme to convergence

- once again, the main concept that we have used to derive this Gauss-Newton optimization problem involving pose variables is to compute the update in the Lie algebra, $\mathfrak{se}(3)$, but store the mean in the Lie group, $SE(3)$

UNIVERSITY OF TORONTO

Pose-graph relaxation



- another interesting problem involving the estimation of pose variables is pose-graph relaxation
- pseudomeasurements are provided in terms of relative pose changes that are not necessarily consistent around loops
- the goal is to express the pose-graph consistently in the initial frame (see the book for details)

UNIVERSITY OF TORONTO