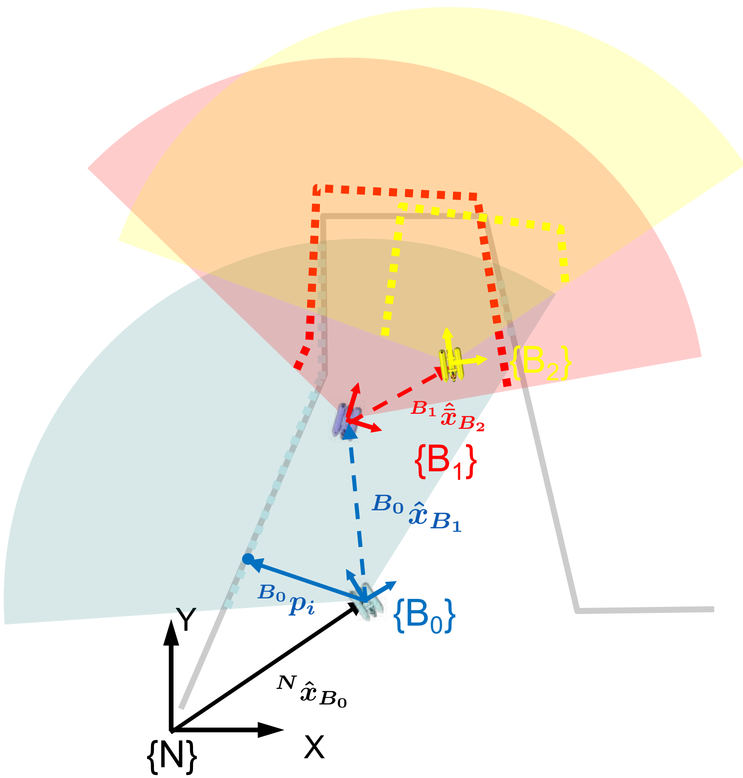
EKF Pose-Based SLAM

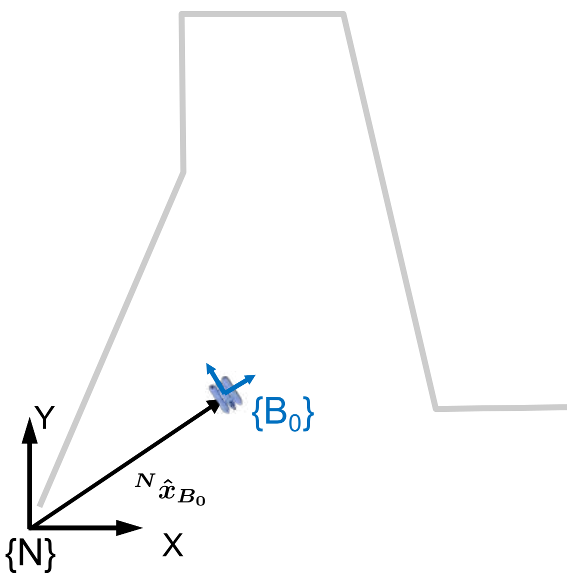


Spatial Relationships

$$^{N}\hat{x}_k = \begin{bmatrix} ^{N}\hat{x}_{B_0} \\ ^{N}\hat{x}_{B_1} \\ \vdots \\ ^{N}\hat{x}_{B_k} \end{bmatrix} \; ; \; ^{N}P_k = \begin{bmatrix} ^{N}P_{B_0} & ^{N}P_{B_0B_1} & \dots & ^{N}P_{B_0B_k} \\ ^{N}P_{B_1B_0} & ^{N}P_{B_1} & \dots & ^{N}P_{B_1B_k} \\ \vdots & \vdots & \ddots & \vdots \\ ^{N}P_{B_kB_0} & ^{N}P_{B_kB_1} & \dots & ^{N}P_{B_k} \end{bmatrix}$$

$$\mathcal{M} = \begin{bmatrix} ^{B_0}S_0 & ^{B_1}S_1 & \dots & \underbrace{^{B_k}S_k}_{[^{B_k}p_0 \; \dots \; ^{B_k}p_i \; \dots \; ^{B_k}p_{np}]} \end{bmatrix}$$

- A map is represented by:
  - The robot poses referenced to the N-Frame
  - The covariances of the robot poses
  - The scans gathered from the poses
- The map is built incrementally

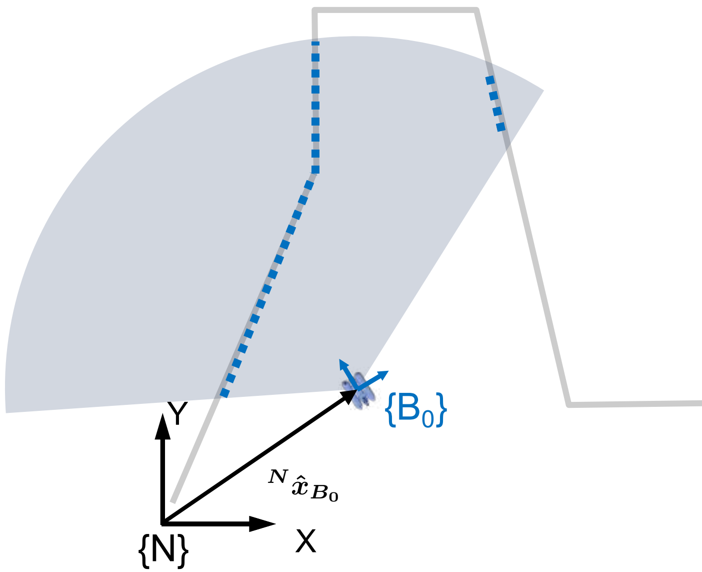


Method Localization$(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$

- $[^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}];$ // Initialize SLAM state vector
- $[^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan();$ // Read the Scan from the sensor
- $[^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = AddNewPose(^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0);$ // Grow the state vector
- $\mathcal{M} = [[^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]];$ // Store the scan in the map
- **for** $k = 1$ *to steps* **do**
  - $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput();$ // Get input to the motion model
  - $[^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction(^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k);$
  - $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement();$ // Read navigation sensors
  - $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,];$
  - **if** *ScanAvailable* **then**
    - $[^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan();$
    - $[^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose(^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k);$ // Grow the state vector
    - $\mathcal{M}[k] = [^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}];$ // Store the scan in the map
    - $\mathcal{H}_o = OverlappingScans(^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M});$ // Get pairs of overlapping scans
    - $c = 1;\ \mathcal{H}_p = [\,];$
    - **for** $i = 1$ *to length*$(\mathcal{H}_o)$ **do** // for all overlaps
      - $j = \mathcal{H}_o[i];$ // Get the pair: $^{B_j}S_j$ overlaps $^{B_k}S_k$
      - $[^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j];$ // Get the scans from the map
      - $^{B_j}\boldsymbol{x}_{B_k} = (\ominus {}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k}\ ;$ // Scan Displacement guess
      - $^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$
      - $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register(^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k});$ // Scan displacement mean & cov
      - **if** $IndividuallyCompatible(^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**
        - $\boldsymbol{z}_p[c] = \boldsymbol{z}_r;\ \boldsymbol{R}_p[c] = \boldsymbol{R}_r;$ // Accepted registration
        - $\mathcal{H}_p[c] = i;\ c = c + 1;$
  - $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p);$
  - $[^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update(^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p);$

$$^N\hat{\bar{\boldsymbol{x}}}_0 = \begin{bmatrix} ^N\hat{\boldsymbol{x}}_{B_0} \\ \\ \\ \end{bmatrix} ; \quad {}^N\bar{\boldsymbol{P}}_2 = \begin{bmatrix} ^N\boldsymbol{P}_{B_0} & & \\ & & \\ & & \end{bmatrix}$$

$$\mathcal{M}_0 = [\qquad\qquad\qquad\qquad]$$

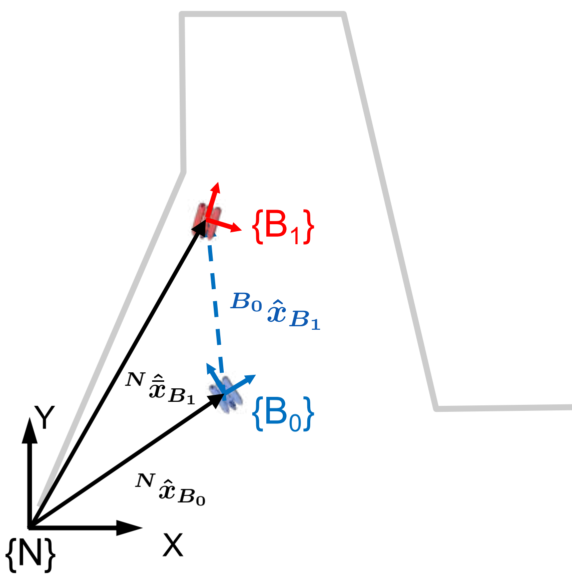


Method Localization$(\hat{x}_0, P_0)$

- $[{}^N\hat{x}_0, {}^N P_0] = [{}^N\hat{x}_{B_0}, {}^N P_{B_0}]$; // Initialize SLAM state vector
- $[{}^{B_0}S_0, {}^{B_0}R_{S_0}] = GetScan()$; // Read the Scan from the sensor
- $[{}^N\hat{\bar{x}}_0, {}^N P_0] = AddNewPose({}^N\hat{x}_0, {}^N\bar{P}_0)$; // Grow the state vector
- $\mathcal{M} = [[{}^{B_0}S_0, {}^{B_0}R_{S_0}]]$; // Store the scan in the map
- **for** $k = 1$ *to steps* **do**
  - $[u_k, Q_k] = GetInput()$; // Get input to the motion model
  - $[{}^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = Prediction({}^N\hat{x}_{k-1}, {}^N P_{k-1}, u_k, Q_k)$;
  - $[z_m, R_m] = GetMeasurement()$; // Read navigation sensors
  - $z_p = [\,]; z_p = [\,]; \mathcal{H}_p = [\,]$;
  - **if** *ScanAvailable* **then**
    - $[{}^{B_k}S_k, {}^{B_k}R_{S_k}] = GetScan()$;
    - $[{}^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = AddNewPose({}^N\hat{\bar{x}}_k, {}^N\bar{P}_k)$; // Grow the state vector
    - $\mathcal{M}[k] = [{}^{B_k}S_k, {}^{B_k}R_{S_k}]$; // Store the scan in the map
    - $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{x}}_k, \mathcal{M})$; // Get pairs of overlapping scans
    - $c = 1;\ \mathcal{H}_p = [\,]$;
    - **for** $i = 1$ *to length*$(\mathcal{H}_o)$ **do** // for all overlaps
      - $j = \mathcal{H}_o[i]$; // Get the pair: ${}^{B_j}S_j$ overlaps ${}^{B_k}S_k$
      - $[{}^{B_j}S_j, {}^{B_j}R_{S_j}] = \mathcal{M}[j]$; // Get the scans from the map
      - ${}^{B_j}x_{B_k} = (\ominus {}^N x_{B_j}) \oplus {}^N x_{B_k}$; // Scan Displacement guess
      - ${}^{B_j}P_{B_k} = J_{1\oplus} J_{\ominus} {}^N P_{B_j} J_{\ominus}^T J_{1\oplus}^T + J_{2\oplus} {}^N P_{B_k} J_{2\oplus}^T$
      - $[z_r, R_r] = Register({}^{B_j}S_j, {}^{B_k}S_k, {}^{B_j}x_{B_k})$; // Scan displacement mean & cov
      - **if** *IndividuallyCompatible*$({}^{B_j}x_{B_k}, {}^{B_j}P_{B_k}, z_r, R_r, \alpha)$ **then**
        - $z_p[c] = z_r;\ R_p[c] = R_r$; // Accepted registration
        - $\mathcal{H}_p[c] = i;\ c = c + 1$;
  - $[z_k, R_k, H_k, V_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{x}}_k, z_m, R_m, z_p, R_p)$;
  - $[{}^N\hat{x}_k, {}^N P_k] = Update({}^N\hat{\bar{x}}_k, {}^N\bar{P}_k, z_k, R_k, H_k, V_k, \mathcal{H}_p)$;

$$
{}^N\hat{\bar{x}}_0 = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ \boxed{{}^N\hat{x}_{B_0}} \\ \\ \end{bmatrix} \; ; \; {}^N\bar{P}_0 = \begin{bmatrix} {}^N P_{B_0} & \boxed{{}^N P_{B_0}} & & \\ \boxed{{}^N P_{B_0}} & \boxed{{}^N P_{B_0}} & & \\ & & & \end{bmatrix}
$$

$$
\mathcal{M}_0 = [\, \boxed{[{}^{B_0}S_0, {}^{B_0}R_{S_0}]} \quad]
$$

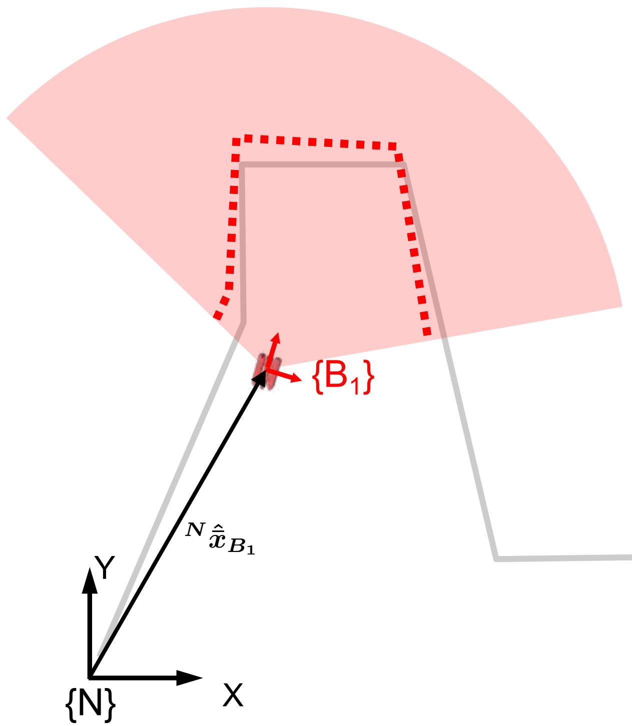


Method Localization($\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0$)

- $[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}]$; `// Initialize SLAM state vector`
- $[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan()$; `// Read the Scan from the sensor`
- $[{}^N\hat{\bar{\boldsymbol{x}}}_0, {}^N\boldsymbol{P}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0)$; `// Grow the state vector`
- $\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]]$; `// Store the scan in the map`
- **for** $k = 1$ *to steps* **do**
  - $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput()$; `// Get input to the motion model`
  - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k)$;
  - $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement()$; `// Read navigation sensors`
  - $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,]$;
  - **if** *ScanAvailable* **then**
    - $[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan()$;
    - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k)$; `// Grow the state vector`
    - $\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}]$; `// Store the scan in the map`
    - $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M})$; `// Get pairs of overlapping scans`
    - $c = 1$; $\mathcal{H}_p = [\,]$;
    - **for** $i = 1$ *to length*$(\mathcal{H}_o)$ **do** `// for all overlaps`
      - $j = \mathcal{H}_o[i]$; `// Get the pair:` ${}^{B_j}S_j$ `overlaps` ${}^{B_k}S_k$
      - $[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j]$; `// Get the scans from the map`
      - ${}^{B_j}\boldsymbol{x}_{B_k} = (\ominus {}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k}$; `// Scan Displacement guess`
      - ${}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$
      - $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k})$; `// Scan displacement mean & cov`
      - **if** $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**
        - $\boldsymbol{z}_p[c] = \boldsymbol{z}_r$; $\boldsymbol{R}_p[c] = \boldsymbol{R}_r$; `// Accepted registration`
        - $\mathcal{H}_p[c] = i$; $c = c + 1$;
  - $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p)$;
  - $[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p)$;

$$
{}^N\hat{\bar{\boldsymbol{x}}}_1 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ \boxed{{}^N\hat{\bar{\boldsymbol{x}}}_{B_1}} \\ \\ \end{bmatrix} \; ; \; {}^N\bar{\boldsymbol{P}}_1 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & \boxed{{}^N\bar{\boldsymbol{P}}_{B_0B_1}} & \\ \boxed{{}^N\bar{\boldsymbol{P}}_{B_1B_0}} & \boxed{{}^N\bar{\boldsymbol{P}}_{B_1}} & \\ & & \end{bmatrix}
$$

$$
\mathcal{M}_1 = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] \qquad] \qquad \boxed{\boldsymbol{z}_m \quad \boldsymbol{R}_m}
$$



Method `Localization`($\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0$)

$[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}]$; `// Initialize SLAM state vector`

$[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan()$; `// Read the Scan from the sensor`

$[{}^N\hat{\bar{\boldsymbol{x}}}_0, {}^N\bar{\boldsymbol{P}}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0)$; `// Grow the state vector`

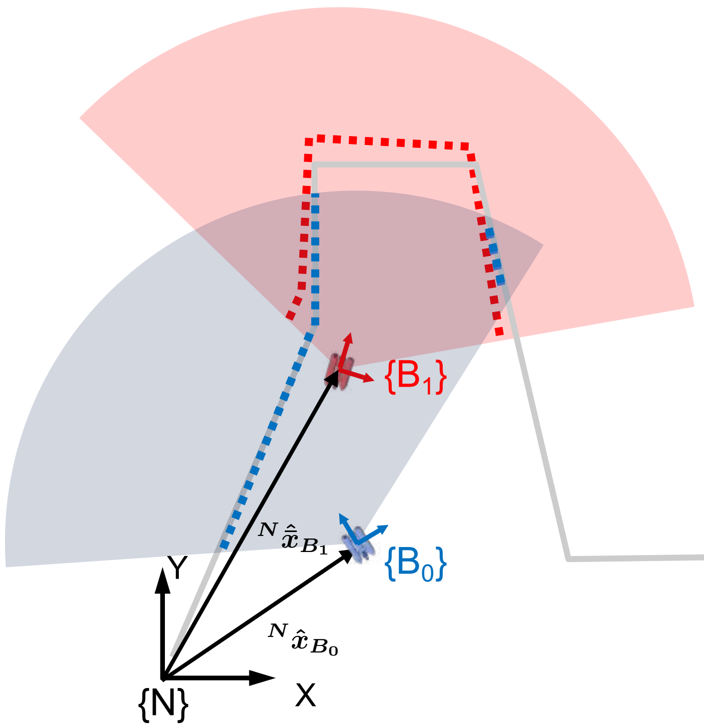
$\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]]$; `// Store the scan in the map`

for $k = 1$ *to steps* **do**

- $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput()$; `// Get input to the motion model`
- $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k)$;
- $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement()$; `// Read navigation sensors`
- $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,]$;
- **if** *ScanAvailable* **then**
  - $[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan()$;
  - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k)$; `// Grow the state vector`
  - $\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}]$; `// Store the scan in the map`
  - $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M})$; `// Get pairs of overlapping scans`
  - $c = 1$; $\mathcal{H}_p = [\,]$;
  - **for** $i = 1$ *to* $length(\mathcal{H}_o)$ **do** `// for all overlaps`
    - $j = \mathcal{H}_o[i]$; `// Get the pair:` ${}^{B_j}S_j$ `overlaps` ${}^{B_k}S_k$
    - $[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j]$; `// Get the scans from the map`
    - ${}^{B_j}\boldsymbol{x}_{B_k} = (\ominus {}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k}$; `// Scan Displacement guess`
    - ${}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$
    - $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k})$; `// Scan displacement mean & cov`
    - **if** $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**
      - $\boldsymbol{z}_p[c] = \boldsymbol{z}_r$; $\boldsymbol{R}_p[c] = \boldsymbol{R}_r$; `// Accepted registration`
      - $\mathcal{H}_p[c] = i$; $c = c + 1$;
- $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p)$;
- $[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p)$;

$$
{}^N\hat{\bar{\boldsymbol{x}}}_1 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_1} \\ \boxed{{}^N\hat{\bar{\boldsymbol{x}}}_{B_1}} \end{bmatrix} ; \quad {}^N\bar{\boldsymbol{P}}_1 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & {}^N\bar{\boldsymbol{P}}_{B_0B_1} & {}^N\bar{\boldsymbol{P}}_{B_0B_1} \\ {}^N\bar{\boldsymbol{P}}_{B_1B_0} & {}^N\bar{\boldsymbol{P}}_{B_1} & {}^N\bar{\boldsymbol{P}}_{B_1} \\ {}^N\bar{\boldsymbol{P}}_{B_1B_0} & {}^N\bar{\boldsymbol{P}}_{B_1} & {}^N\bar{\boldsymbol{P}}_{B_1} \end{bmatrix}
$$

$$
\mathcal{M}_1 = [[{}^{B_0}\boldsymbol{S}_0, {}^{B0}\boldsymbol{R}_{S_0}]\ \boxed{[{}^{B_1}\boldsymbol{S}_1, {}^{B1}\boldsymbol{R}_{S_1}]}\ ] \quad [\boldsymbol{z}_m \quad \boldsymbol{R}_m]
$$



Method Localization$(\hat{x}_0, P_0)$

$[^N\hat{x}_0, {}^N P_0] = [^N\hat{x}_{B_0}, {}^N P_{B_0}]$; // Initialize SLAM state vector

$[^{B_0}S_0, {}^{B_0}R_{S_0}] = GetScan()$; // Read the Scan from the sensor

$[^N\hat{\bar{x}}_0, {}^N P_0] = AddNewPose(^N\hat{x}_0, {}^N\bar{P}_0)$; // Grow the state vector

$\mathcal{M} = [[^{B_0}S_0, {}^{B_0}R_{S_0}]]$; // Store the scan in the map

for $k = 1$ *to steps* **do**

$[u_k, Q_k] = GetInput()$; // Get input to the motion model

$[^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = Prediction(^N\hat{x}_{k-1}, {}^N P_{k-1}, u_k, Q_k)$;

$[z_m, R_m] = GetMeasurement()$; // Read navigation sensors

$z_p = [\,]; z_p = [\,]; \mathcal{H}_p = [\,]$;

if *ScanAvailable* **then**

$[^{B_k}S_k, {}^{B_k}R_{S_k}] = GetScan()$;

$[^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = AddNewPose(^N\hat{\bar{x}}_k, {}^N\bar{P}_k)$; // Grow the state vector

$\mathcal{M}[k] = [^{B_k}S_k, {}^{B_k}R_{S_k}]$; // Store the scan in the map

$\mathcal{H}_o = OverlappingScans(^N\hat{\bar{x}}_k, \mathcal{M})$; // Get pairs of overlapping scans

$c = 1$; $\mathcal{H}_p = [\,]$;

for $i = 1$ *to* $length(\mathcal{H}_o)$ **do** // for all overlaps

$j = \mathcal{H}_o[i]$; // Get the pair: $^{B_j}S_j$ overlaps $^{B_k}S_k$

$[^{B_j}S_j, {}^{B_j}R_{S_j}] = \mathcal{M}[j]$; // Get the scans from the map

$^{B_j}x_{B_k} = (\ominus {}^N x_{B_j}) \oplus {}^N x_{B_k}$; // Scan Displacement guess

$^{B_j}P_{B_k} = J_{1\oplus} J_{\ominus} {}^N P_{B_j} J_{\ominus}^T J_{1\oplus}^T + J_{2\oplus} {}^N P_{B_k} J_{2\oplus}^T$

$[z_r, R_r] = Register(^{B_j}S_j, {}^{B_k}S_k, {}^{B_j}x_{B_k})$; // Scan displacement mean & cov

if $IndividuallyCompatible(^{B_j}x_{B_k}, {}^{B_j}P_{B_k}, z_r, R_r, \alpha)$ **then**

$z_p[c] = z_r$; $R_p[c] = R_r$; // Accepted registration

$\mathcal{H}_p[c] = i$; $c = c + 1$;

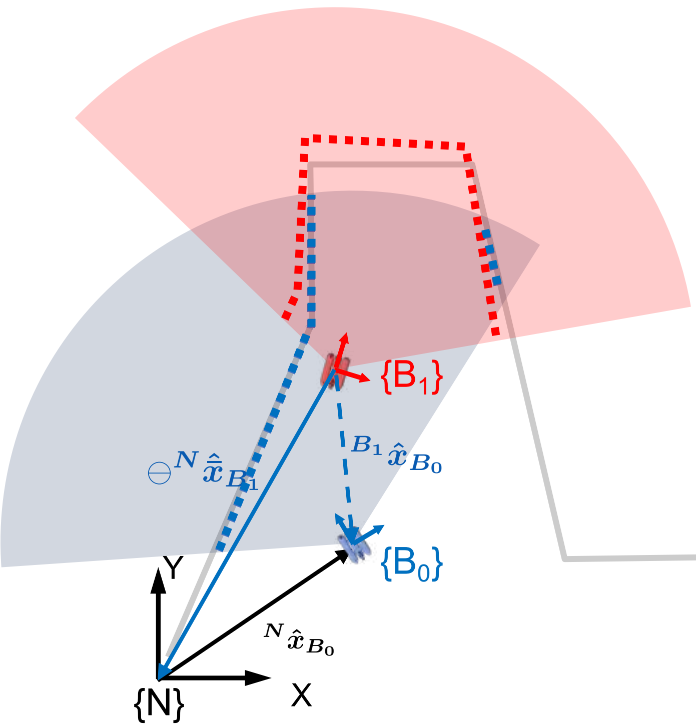
$[z_k, R_k, H_k, V_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{x}}_k, z_m, R_m, z_p, R_p)$;

$[^N\hat{x}_k, {}^N P_k] = Update(^N\hat{\bar{x}}_k, {}^N\bar{P}_k, z_k, R_k, H_k, V_k, \mathcal{H}_p)$;

$$^N\hat{x}_1 = \begin{bmatrix} ^N\hat{x}_{B_0} \\ ^N\hat{x}_{B_1} \\ ^N\hat{x}_{B_1} \\ \end{bmatrix} \; ; \; ^N P_1 = \begin{bmatrix} ^N P_{B_0} & ^N P_{B_0 B_1} & ^N P_{B_0 B_1} & \\ ^N P_{B_1 B_0} & ^N P_{B_1} & ^N P_{B_1} & \\ ^N P_{B_1 B_0} & ^N P_{B_1} & ^N P_{B_1} & \\ & & & \end{bmatrix}$$

$$\mathcal{M}_1 = [[^{B_0}S_0, {}^{B0}R_{S_0}] \; [^{B_1}S_1, {}^{B1}R_{S_1}] \qquad] \quad [z_m \quad R_m]$$

$$\mathcal{H}_p = [\boxed{0} \quad]$$



Method Localization($\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0$)

- $[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}]$; // Initialize SLAM state vector
- $[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan()$; // Read the Scan from the sensor
- $[{}^N\hat{\bar{\boldsymbol{x}}}_0, {}^N\bar{\boldsymbol{P}}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0)$; // Grow the state vector
- $\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]]$; // Store the scan in the map
- **for** $k = 1$ *to steps* **do**
  - $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput()$; // Get input to the motion model
  - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k)$;
  - $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement()$; // Read navigation sensors
  - $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,]$;
  - **if** *ScanAvailable* **then**
    - $[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan()$;
    - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k)$; // Grow the state vector
    - $\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}]$; // Store the scan in the map
    - $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M})$; // Get pairs of overlapping scans
    - $c = 1$; $\mathcal{H}_p = [\,]$;
    - **for** $i = 1$ *to* $length(\mathcal{H}_o)$ **do** // for all overlaps
      - $j = \mathcal{H}_o[i]$; // Get the pair: ${}^{B_j}S_j$ overlaps ${}^{B_k}S_k$
      - $[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j]$; // Get the scans from the map
      - ${}^{B_j}\boldsymbol{x}_{B_k} = (\ominus{}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k}$; // Scan Displacement guess
      - ${}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$
      - $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k})$; // Scan displacement mean & cov
      - **if** $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**
        - $\boldsymbol{z}_p[c] = \boldsymbol{z}_r$; $\boldsymbol{R}_p[c] = \boldsymbol{R}_r$; // Accepted registration
        - $\mathcal{H}_p[c] = i$; $c = c + 1$;
  - $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p)$;
  - $[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p)$;

$$
{}^N\hat{\boldsymbol{x}}_1 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ {}^N\hat{\boldsymbol{x}}_{B_1} \\ {}^N\hat{\boldsymbol{x}}_{B_1} \end{bmatrix} \; ; \; {}^N\boldsymbol{P}_1 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & {}^N\boldsymbol{P}_{B_0B_1} & {}^N\boldsymbol{P}_{B_0B_1} & \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\boldsymbol{P}_{B_1} & \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\boldsymbol{P}_{B_1} & \end{bmatrix}
$$

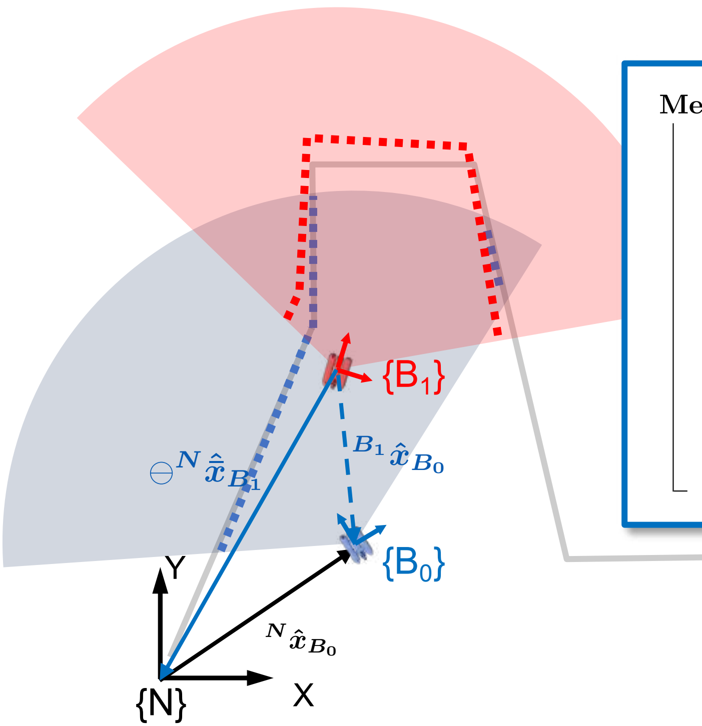
$$
\mathcal{M}_1 = \left[[{}^{B_0}\boldsymbol{S}_0, {}^{B0}\boldsymbol{R}_{S_0}] \; [{}^{B_1}\boldsymbol{S}_1, {}^{B1}\boldsymbol{R}_{S_1}] \qquad \right] \quad [\boldsymbol{z}_m \quad \boldsymbol{R}_m]
$$

$$
\mathcal{H}_p = [0 \quad]
$$

$$
{}^{B_1}\boldsymbol{x}_{B_0} = (\ominus{}^N\boldsymbol{x}_{B_2}) \oplus {}^N\boldsymbol{x}_{B_0})
$$

$$
[\boldsymbol{z}_{p_1}, \boldsymbol{R}_{p_1}] = Register({}^{B_1}\boldsymbol{S}_1, {}^{B_0}\boldsymbol{S}_0, {}^{B_2}\boldsymbol{x}_{B_0})
$$

$$
\boldsymbol{z}_p = [\boldsymbol{z}_{p_1}] \; ; \; \boldsymbol{R}_p = [\boldsymbol{R}_{p_1}]
$$



Method `Localization`$(\hat{x}_0, P_0)$

$[{}^N\hat{x}_0, {}^N P_0] = [{}^N\hat{x}_{B_0}, {}^N P_{B_0}];$ `// Initialize SLAM state vector`

$[{}^{B_0}S_0, {}^{B_0}R_{S_0}] = GetScan();$ `// Read the Scan from the sensor`

$[{}^N\hat{\bar{x}}_0, {}^N P_0] = AddNewPose({}^N\hat{x}_0, {}^N\bar{P}_0);$ `// Grow the state vector`

$\mathcal{M} = [[{}^{B_0}S_0, {}^{B_0}R_{S_0}]];$ `// Store the scan in the map`

for $k = 1$ *to steps* **do**

$[u_k, Q_k] = GetInput();$ `// Get input to the motion model`

$[{}^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = Prediction({}^N\hat{x}_{k-1}, {}^N P_{k-1}, u_k, Q_k);$

$[z_m, R_m] = GetMeasurement();$ `// Read navigation sensors`

Method `ObservationMatrix`$(\mathcal{H}_p, {}^N\hat{\bar{x}}_k, z_m, R_m, z_p, R_p)$

$H_p = 0;$

for $i = 1$ *to length*(z_p) **do**

$H_p[i,1] = J_{1\oplus}(\ominus^N\hat{\bar{x}}_{B_k}, {}^N\hat{x}_k[\mathcal{H}_p[i]])J_{\ominus}({}^N\hat{\bar{x}}_{B_k});$

$H_p[i, \mathcal{H}_p[i]] = J_{2\oplus}({}^N\hat{\bar{x}}_{B_k});$

$z_k = \begin{bmatrix} z_m \\ z_p \end{bmatrix}; R_k = \begin{bmatrix} R_m & 0 \\ 0 & R_p \end{bmatrix};$ `// Indep. measure&displacement noise`

$H_k = \begin{bmatrix} J_{h_x}(\hat{\bar{x}}_k) \\ H_p \end{bmatrix};$ `// Stack measurements and Displ. observ.`

$V_k = \begin{bmatrix} J_{h_v}(\hat{\bar{x}}_k) \\ V_p \end{bmatrix};$

return $[z_k, R_k, H_k, V_k];$

$\mathcal{H}_p[c] = i; c = c + 1;$

$[z_k, R_k, H_k, V_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{x}}_k, z_m, R_m, z_p, R_p);$

$[{}^N\hat{x}_k, {}^N P_k] = Update({}^N\hat{\bar{x}}_k, {}^N\bar{P}_k, z_k, R_k, H_k, V_k, \mathcal{H}_p);$

$${}^N\hat{x}_1 = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ {}^N\hat{x}_{B_1} \\ {}^N\hat{x}_{B_1} \end{bmatrix} ; {}^N P_1 = \begin{bmatrix} {}^N P_{B_0} & {}^N P_{B_0 B_1} & {}^N P_{B_0 B_1} \\ {}^N P_{B_1 B_0} & {}^N P_{B_1} & {}^N P_{B_1} \\ {}^N P_{B_1 B_0} & {}^N P_{B_1} & {}^N P_{B_1} \end{bmatrix}$$

$$\mathcal{M}_1 = [[{}^{B_0}S_0, {}^{B0}R_{S_0}]\ [{}^{B_1}S_1, {}^{B1}R_{S_1}]\ ] \quad [z_m \quad R_m]$$

$$\mathcal{H}_p = [0 \quad]$$

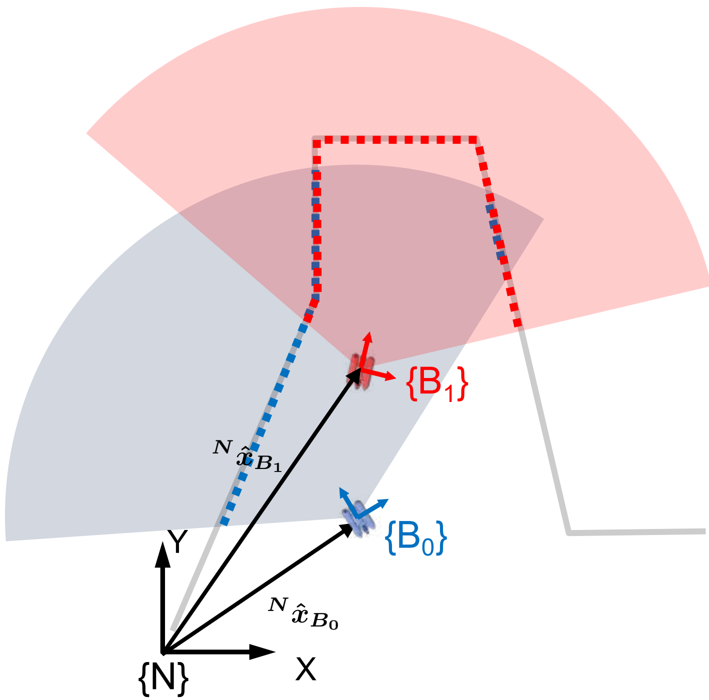
$${}^{B_1}x_{B_0} = (\ominus^N x_{B_2}) \oplus {}^N x_{B_0})$$

$$[z_{p_1}, R_{p_1}] = Register({}^{B_1}S_1, {}^{B_0}S_0, {}^{B_2}x_{B_0})$$

$$z_p = [z_{p_1}] \; ; \; R_p = [R_{p_1}]$$

$$z_k = \begin{bmatrix} z_m \\ z_p \end{bmatrix} ; R_k = \begin{bmatrix} R_m & 0 \\ 0 & R_p \end{bmatrix}$$

$$H_k = \begin{bmatrix} H_m & 0 \\ 0 & H_p \end{bmatrix} ; V_k = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$



Method Localization$(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$

$[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}]$; // Initialize SLAM state vector

$[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan()$; // Read the Scan from the sensor

$[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0)$; // Grow the state vector

$\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]]$; // Store the scan in the map

for $k = 1$ *to steps* **do**

  $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput()$; // Get input to the motion model

  $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k)$;

  $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement()$; // Read navigation sensors

  $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,]$;

  if *ScanAvailable* **then**

    $[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan()$;

    $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k)$; // Grow the state vector

    $\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}]$; // Store the scan in the map

    $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M})$; // Get pairs of overlapping scans

    $c = 1$; $\mathcal{H}_p = [\,]$;

    for $i = 1$ *to length*$(\mathcal{H}_o)$ **do** // for all overlaps

      $j = \mathcal{H}_o[i]$; // Get the pair: ${}^{B_j}S_j$ overlaps ${}^{B_k}S_k$

      $[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j]$; // Get the scans from the map

      ${}^{B_j}\boldsymbol{x}_{B_k} = (\ominus{}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k}$; // Scan Displacement guess

      ${}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$

      $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k})$; // Scan displacement mean & cov

      if $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**

        $\boldsymbol{z}_p[c] = \boldsymbol{z}_r$; $\boldsymbol{R}_p[c] = \boldsymbol{R}_r$; // Accepted registration

        $\mathcal{H}_p[c] = i$; $c = c + 1$;

  $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p)$;

  $[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p)$;

$$
{}^N\hat{\boldsymbol{x}}_1 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ {}^N\hat{\boldsymbol{x}}_{B_1} \\ {}^N\hat{\boldsymbol{x}}_{B_1} \end{bmatrix} \; ; \; {}^N\boldsymbol{P}_1 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & {}^N\boldsymbol{P}_{B_0B_1} & {}^N\boldsymbol{P}_{B_0B_1} \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\boldsymbol{P}_{B_1} \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\boldsymbol{P}_{B_1} \end{bmatrix}
$$

$$
\mathcal{M}_1 = [[{}^{B_0}\boldsymbol{S}_0, {}^{B0}\boldsymbol{R}_{S_0}] \; [{}^{B_1}\boldsymbol{S}_1, {}^{B1}\boldsymbol{R}_{S_1}] \quad] \quad [\boldsymbol{z}_m \quad \boldsymbol{R}_m]
$$

$$
\mathcal{H}_p = [0 \quad]
$$

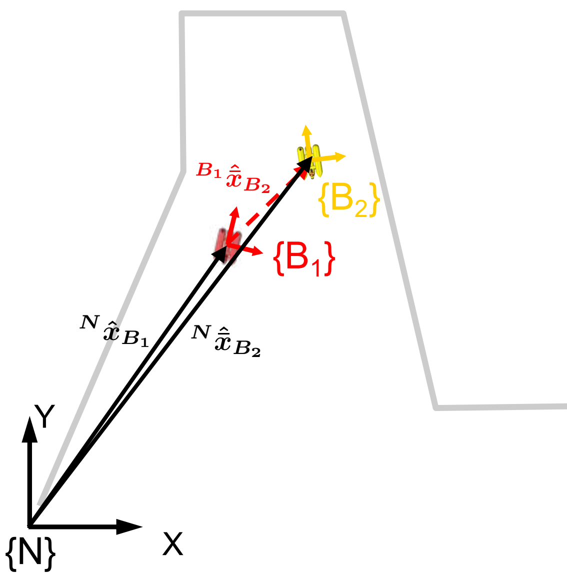
$$
{}^{B_1}\boldsymbol{x}_{B_0} = (\ominus{}^N\boldsymbol{x}_{B_2}) \oplus {}^N\boldsymbol{x}_{B_0})
$$

$$
[\boldsymbol{z}_{p_1}, \boldsymbol{R}_{p_1}] = Register({}^{B_1}\boldsymbol{S}_1, {}^{B_0}\boldsymbol{S}_0, {}^{B_2}\boldsymbol{x}_{B_0})
$$

$$
\boldsymbol{z}_p = [\boldsymbol{z}_{p_1}] \; ; \; \boldsymbol{R}_p = [\boldsymbol{R}_{p_1}]
$$

$$
\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{z}_m \\ \boldsymbol{z}_p \end{bmatrix} \; ; \; \boldsymbol{R}_k = \begin{bmatrix} \boldsymbol{R}_m & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_p \end{bmatrix}
$$

$$
\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{H}_m & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_p \end{bmatrix} \; ; \; \boldsymbol{V}_k = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}
$$



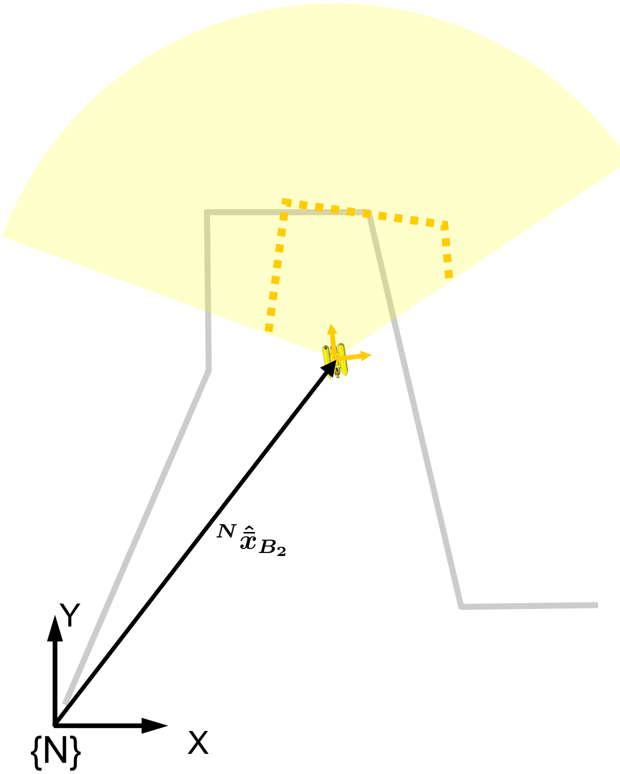
Method `Localization`$(\hat{x}_0, P_0)$

- $[{}^N\hat{x}_0, {}^N P_0] = [{}^N\hat{x}_{B_0}, {}^N P_{B_0}]$; `// Initialize SLAM state vector`
- $[{}^{B_0}S_0, {}^{B_0}R_{S_0}] = GetScan()$; `// Read the Scan from the sensor`
- $[{}^N\hat{\bar{x}}_0, {}^N P_0] = AddNewPose({}^N\hat{x}_0, {}^N\bar{P}_0)$; `// Grow the state vector`
- $\mathcal{M} = [[{}^{B_0}S_0, {}^{B_0}R_{S_0}]]$; `// Store the scan in the map`
- **for** $k = 1$ *to steps* **do**
  - $[u_k, Q_k] = GetInput()$; `// Get input to the motion model`
  - $[{}^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = Prediction({}^N\hat{x}_{k-1}, {}^N P_{k-1}, u_k, Q_k)$;
  - $[z_m, R_m] = GetMeasurement()$; `// Read navigation sensors`
  - $z_p = [\,]; z_p = [\,]; \mathcal{H}_p = [\,]$;
  - **if** *ScanAvailable* **then**
    - $[{}^{B_k}S_k, {}^{B_k}R_{S_k}] = GetScan()$;
    - $[{}^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = AddNewPose({}^N\hat{\bar{x}}_k, {}^N\bar{P}_k)$; `// Grow the state vector`
    - $\mathcal{M}[k] = [{}^{B_k}S_k, {}^{B_k}R_{S_k}]$; `// Store the scan in the map`
    - $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{x}}_k, \mathcal{M})$; `// Get pairs of overlapping scans`
    - $c = 1$; $\mathcal{H}_p = [\,]$;
    - **for** $i = 1$ *to* $length(\mathcal{H}_o)$ **do** `// for all overlaps`
      - $j = \mathcal{H}_o[i]$; `// Get the pair:` ${}^{B_j}S_j$ `overlaps` ${}^{B_k}S_k$
      - $[{}^{B_j}S_j, {}^{B_j}R_{S_j}] = \mathcal{M}[j]$; `// Get the scans from the map`
      - ${}^{B_j}x_{B_k} = (\ominus {}^N x_{B_j}) \oplus {}^N x_{B_k}$; `// Scan Displacement guess`
      - ${}^{B_j}P_{B_k} = J_{1\oplus} J_{\ominus} {}^N P_{B_j} J_{\ominus}^T J_{1\oplus}^T + J_{2\oplus} {}^N P_{B_k} J_{2\oplus}^T$
      - $[z_r, R_r] = Register({}^{B_j}S_j, {}^{B_k}S_k, {}^{B_j}x_{B_k})$; `// Scan displacement mean & cov`
      - **if** $IndividuallyCompatible({}^{B_j}x_{B_k}, {}^{B_j}P_{B_k}, z_r, R_r, \alpha)$ **then**
        - $z_p[c] = z_r$; $R_p[c] = R_r$; `// Accepted registration`
        - $\mathcal{H}_p[c] = i$; $c = c + 1$;
  - $[z_k, R_k, H_k, V_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{x}}_k, z_m, R_m, z_p, R_p)$;
  - $[{}^N\hat{x}_k, {}^N P_k] = Update({}^N\hat{\bar{x}}_k, {}^N\bar{P}_k, z_k, R_k, H_k, V_k, \mathcal{H}_p)$;

$$
{}^N\hat{\bar{x}}_2 = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ {}^N\hat{x}_{B_1} \\ \boxed{{}^N\hat{\bar{x}}_{B_2}} \end{bmatrix} \; ; \; {}^N\bar{P}_2 = \begin{bmatrix} {}^N P_{B_0} & {}^N P_{B_0 B_1} & {}^N\bar{P}_{B_0 B_2} & \\ {}^N P_{B_1 B_0} & {}^N P_{B_1} & {}^N\bar{P}_{B_1, B_2} & \\ {}^N\bar{P}_{B_2 B_0} & {}^N\bar{P}_{B_2 B_1} & {}^N\bar{P}_{B_2} & \\ & & & \end{bmatrix}
$$

$$
\mathcal{M}_2 = [[{}^{B_0}S_0, {}^{B0}R_{S_0}] \; [{}^{B_1}S_1, {}^{B1}R_{S_1}] \qquad] \qquad \boxed{z_m \quad R_m}
$$

$$
\mathcal{H}_p = [0 \qquad]
$$



Method Localization$(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$

$[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}];$ // Initialize SLAM state vector

$[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan();$ // Read the Scan from the sensor

$[{}^N\hat{\bar{\boldsymbol{x}}}_0, {}^N\bar{\boldsymbol{P}}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0);$ // Grow the state vector

$\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]];$ // Store the scan in the map

for $k = 1$ *to steps* **do**

$\quad[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput();$ // Get input to the motion model

$\quad[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k);$

$\quad[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement();$ // Read navigation sensors

$\quad\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,];$

$\quad$**if** *ScanAvailable* **then**

$\quad\quad[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan();$

$\quad\quad[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k);$ // Grow the state vector

$\quad\quad\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}];$ // Store the scan in the map

$\quad\quad\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M});$ // Get pairs of overlapping scans

$\quad\quad c = 1; \mathcal{H}_p = [\,];$

$\quad\quad$**for** $i = 1$ *to length*$(\mathcal{H}_o)$ **do** // for all overlaps

$\quad\quad\quad j = \mathcal{H}_o[i];$ // Get the pair: ${}^{B_j}S_j$ overlaps ${}^{B_k}S_k$

$\quad\quad\quad[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j];$ // Get the scans from the map

$\quad\quad\quad {}^{B_j}\boldsymbol{x}_{B_k} = (\ominus {}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k};$ // Scan Displacement guess

$\quad\quad\quad {}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$

$\quad\quad\quad[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k});$ // Scan displacement mean & cov

$\quad\quad\quad$**if** $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**

$\quad\quad\quad\quad\boldsymbol{z}_p[c] = \boldsymbol{z}_r;\ \boldsymbol{R}_p[c] = \boldsymbol{R}_r;$ // Accepted registration

$\quad\quad\quad\quad\mathcal{H}_p[c] = i;\ c = c + 1;$

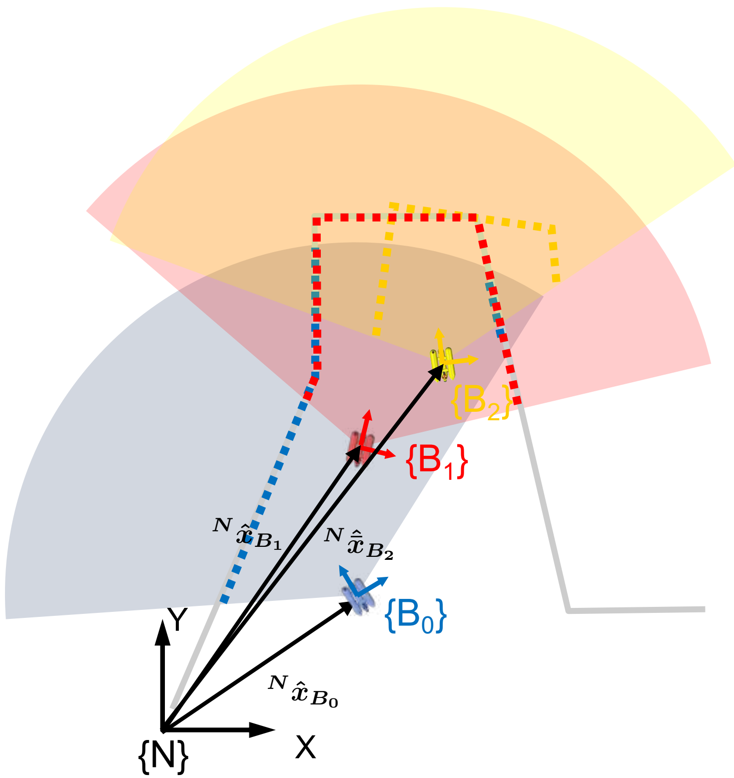
$\quad[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p);$

$\quad[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p);$

$$
{}^N\hat{\bar{\boldsymbol{x}}}_2 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ {}^N\hat{\boldsymbol{x}}_{B_1} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \\ \boxed{{}^N\hat{\bar{\boldsymbol{x}}}_{B_2}} \end{bmatrix} \; ; \; {}^N\bar{\boldsymbol{P}}_2 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & {}^N\boldsymbol{P}_{B_0B_1} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\bar{\boldsymbol{P}}_{B_1,B_2} & {}^N\bar{\boldsymbol{P}}_{B_1B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \end{bmatrix}
$$

$$
\mathcal{M}_2 = [[{}^{B_0}\boldsymbol{S}_0, {}^{B0}\boldsymbol{R}_{S_0}]\ [{}^{B_1}\boldsymbol{S}_1, {}^{B1}\boldsymbol{R}_{S_1}]\ \boxed{[{}^{B_2}\boldsymbol{S}_2, {}^{B2}\boldsymbol{R}_{S_2}]}] \quad [\boldsymbol{z}_m \quad \boldsymbol{R}_m]
$$

$$
\mathcal{H}_p = [0 \quad]
$$



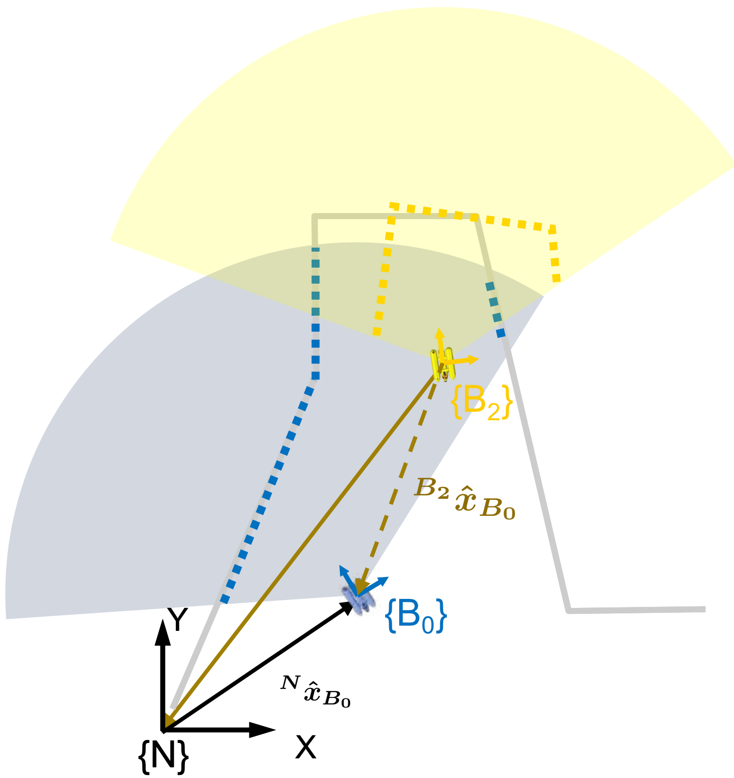
Method Localization$(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$

- $[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}]$; // Initialize SLAM state vector
- $[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan()$; // Read the Scan from the sensor
- $[{}^N\hat{\bar{\boldsymbol{x}}}_0, {}^N\boldsymbol{P}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0)$; // Grow the state vector
- $\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]]$; // Store the scan in the map
- **for** $k = 1$ *to steps* **do**
  - $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput()$; // Get input to the motion model
  - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k)$;
  - $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement()$; // Read navigation sensors
  - $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,]$;
  - **if** *ScanAvailable* **then**
    - $[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan()$;
    - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k)$; // Grow the state vector
    - $\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}]$; // Store the scan in the map
    - $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M})$; // Get pairs of overlapping scans
    - $c = 1;\ \mathcal{H}_p = [\,]$;
    - **for** $i = 1$ *to* $length(\mathcal{H}_o)$ **do** // for all overlaps
      - $j = \mathcal{H}_o[i]$; // Get the pair: ${}^{B_j}S_j$ overlaps ${}^{B_k}S_k$
      - $[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j]$; // Get the scans from the map
      - ${}^{B_j}\boldsymbol{x}_{B_k} = (\ominus {}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k}$; // Scan Displacement guess
      - ${}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$
      - $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k})$; // Scan displacement mean & cov
      - **if** $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**
        - $\boldsymbol{z}_p[c] = \boldsymbol{z}_r;\ \boldsymbol{R}_p[c] = \boldsymbol{R}_r$; // Accepted registration
        - $\mathcal{H}_p[c] = i;\ c = c + 1$;
  - $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p)$;
  - $[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p)$;

$$
{}^N\hat{\bar{\boldsymbol{x}}}_2 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_1} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \end{bmatrix} \; ; \; {}^N\bar{\boldsymbol{P}}_2 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & {}^N\boldsymbol{P}_{B_0B_1} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\bar{\boldsymbol{P}}_{B_1,B_2} & {}^N\bar{\boldsymbol{P}}_{B_1B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \end{bmatrix}
$$

$$
\mathcal{M}_2 = [[{}^{B_0}\boldsymbol{S}_0, {}^{B0}\boldsymbol{R}_{S_0}]\ [{}^{B_1}\boldsymbol{S}_1, {}^{B1}\boldsymbol{R}_{S_1}]\ [{}^{B_2}\boldsymbol{S}_2, {}^{B2}\boldsymbol{R}_{S_2}]] \quad [\boldsymbol{z}_m \quad \boldsymbol{R}_m]
$$

$$
\mathcal{H}_p = [0 \quad \boxed{1}]
$$



Method `Localization`$(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$

$[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}]$; `// Initialize SLAM state vector`

$[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan()$; `// Read the Scan from the sensor`

$[{}^N\hat{\bar{\boldsymbol{x}}}_0, {}^N\bar{\boldsymbol{P}}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0)$; `// Grow the state vector`

$\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]]$; `// Store the scan in the map`

for $k = 1$ *to steps* **do**

  $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput()$; `// Get input to the motion model`

  $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k)$;

  $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement()$; `// Read navigation sensors`

  $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,]$;

  if *ScanAvailable* **then**

    $[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan()$;

    $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k)$; `// Grow the state vector`

    $\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}]$; `// Store the scan in the map`

    $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M})$; `// Get pairs of overlapping scans`

    $c = 1$; $\mathcal{H}_p = [\,]$;

    for $i = 1$ *to* $length(\mathcal{H}_o)$ **do** `// for all overlaps`

      $j = \mathcal{H}_o[i]$; `// Get the pair:` ${}^{B_j}S_j$ `overlaps` ${}^{B_k}S_k$

      $[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j]$; `// Get the scans from the map`

      ${}^{B_j}\boldsymbol{x}_{B_k} = (\ominus {}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k}$; `// Scan Displacement guess`

      ${}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$

      $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k})$; `// Scan displacement mean & cov`

      if $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**

        $\boldsymbol{z}_p[c] = \boldsymbol{z}_r$; $\boldsymbol{R}_p[c] = \boldsymbol{R}_r$; `// Accepted registration`

        $\mathcal{H}_p[c] = i$; $c = c + 1$;

  $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p)$;

  $[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p)$;

$$
{}^N\hat{\bar{\boldsymbol{x}}}_2 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ {}^N\hat{\boldsymbol{x}}_{B_1} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \end{bmatrix} \; ; \; {}^N\bar{\boldsymbol{P}}_2 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & {}^N\boldsymbol{P}_{B_0B_1} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\bar{\boldsymbol{P}}_{B_1,B_2} & {}^N\bar{\boldsymbol{P}}_{B_1B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \end{bmatrix}
$$

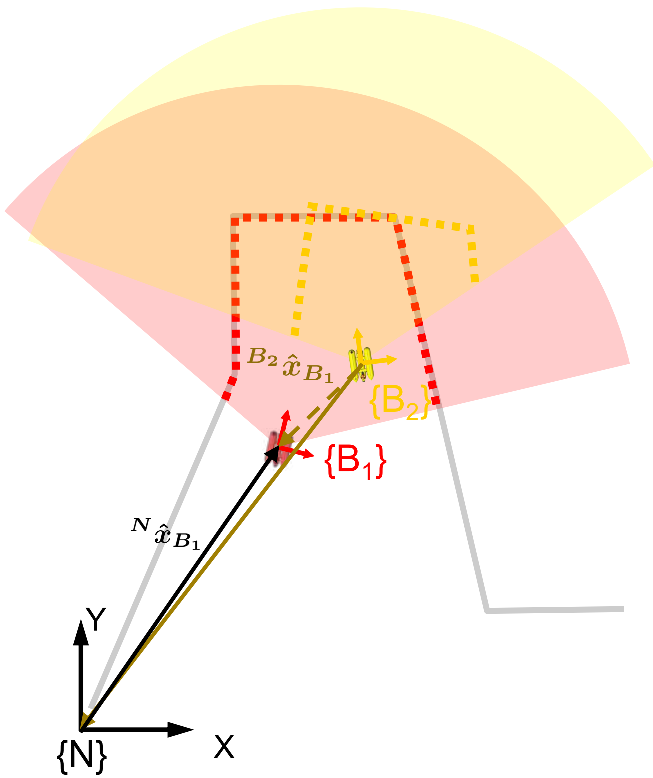
$$
\mathcal{M}_2 = \left[\boxed{{}^{B_0}\boldsymbol{S}_0, {}^{B0}\boldsymbol{R}_{S_0}} \;\; [{}^{B_1}\boldsymbol{S}_1, {}^{B1}\boldsymbol{R}_{S_1}] \;\; \boxed{{}^{B_2}\boldsymbol{S}_2, {}^{B2}\boldsymbol{R}_{S_2}}\right] \quad [\boldsymbol{z}_m \quad \boldsymbol{R}_m]
$$

$$
\mathcal{H}_p = [\boxed{0} \quad 1]
$$

$$
\boxed{{}^{B_2}\boldsymbol{x}_{B_0} = (\ominus {}^N\boldsymbol{x}_{B_2}) \oplus {}^N\boldsymbol{x}_{B_0})};
$$

$$
\boxed{[\boldsymbol{z}_{p_1}, \boldsymbol{R}_{p_1}] = Register({}^{B_2}\boldsymbol{S}_2, {}^{B_0}\boldsymbol{S}_0, {}^{B_2}\boldsymbol{x}_{B_0})};
$$

$$
\boxed{\boldsymbol{z}_p = [\boldsymbol{z}_{p_1} \quad \boldsymbol{z}_{p_2}]};
$$



Method Localization$(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$

- $[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}];$ // Initialize SLAM state vector
- $[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan();$ // Read the Scan from the sensor
- $[{}^N\hat{\bar{\boldsymbol{x}}}_0, {}^N\bar{\boldsymbol{P}}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0);$ // Grow the state vector
- $\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]];$ // Store the scan in the map
- **for** $k = 1$ *to steps* **do**
  - $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput();$ // Get input to the motion model
  - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k);$
  - $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement();$ // Read navigation sensors
  - $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,];$
  - **if** *ScanAvailable* **then**
    - $[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan();$
    - $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k);$ // Grow the state vector
    - $\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}];$ // Store the scan in the map
    - $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M});$ // Get pairs of overlapping scans
    - $c = 1; \mathcal{H}_p = [\,];$
    - **for** $i = 1$ *to* $length(\mathcal{H}_o)$ **do** // for all overlaps
      - $j = \mathcal{H}_o[i];$ // Get the pair: ${}^{B_j}S_j$ overlaps ${}^{B_k}S_k$
      - $[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j];$ // Get the scans from the map
      - ${}^{B_j}\boldsymbol{x}_{B_k} = (\ominus {}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k};$ // Scan Displacement guess
      - ${}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$
      - $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k});$ // Scan displacement mean & cov
      - **if** $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**
        - $\boldsymbol{z}_p[c] = \boldsymbol{z}_r;\ \boldsymbol{R}_p[c] = \boldsymbol{R}_r;$ // Accepted registration
        - $\mathcal{H}_p[c] = i;\ c = c + 1;$
  - $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p);$
  - $[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p);$

$$
{}^N\hat{\bar{\boldsymbol{x}}}_2 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ {}^N\hat{\boldsymbol{x}}_{B_1} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \end{bmatrix} \; ; \; {}^N\bar{\boldsymbol{P}}_2 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & {}^N\boldsymbol{P}_{B_0B_1} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\bar{\boldsymbol{P}}_{B_1,B_2} & {}^N\bar{\boldsymbol{P}}_{B_1B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \end{bmatrix}
$$

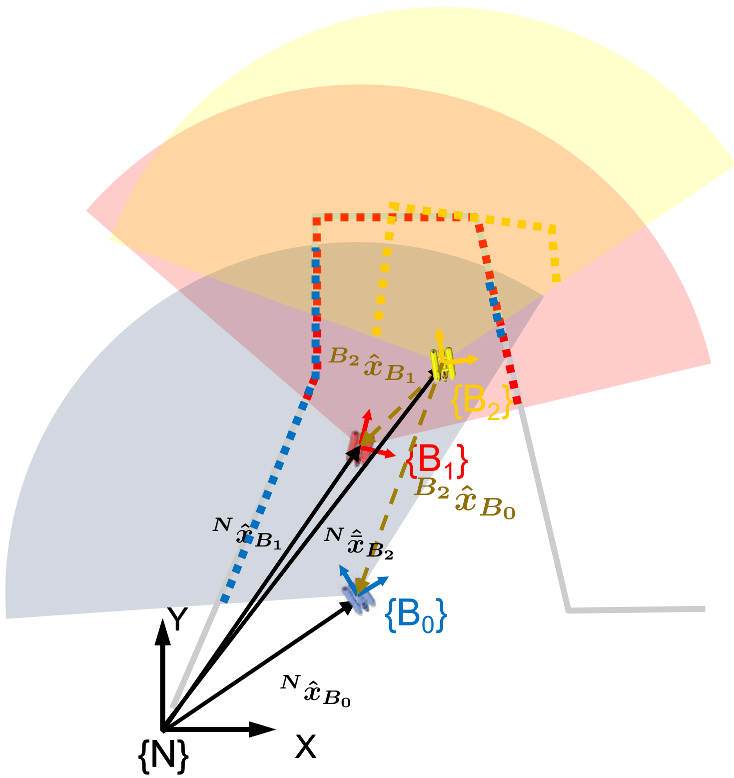
$$
\mathcal{M}_2 = [[{}^{B_0}\boldsymbol{S}_0, {}^{B0}\boldsymbol{R}_{S_0}]\ \boxed{[{}^{B_1}\boldsymbol{S}_1, {}^{B1}\boldsymbol{R}_{S_1}]}\ \boxed{[{}^{B_2}\boldsymbol{S}_2, {}^{B2}\boldsymbol{R}_{S_2}]}] \quad [\boldsymbol{z}_m \quad \boldsymbol{R}_m]
$$

$$
\mathcal{H}_p = [0 \ \boxed{1}]
$$

$$
{}^{B_2}\boldsymbol{x}_{B_0} = (\ominus {}^N\boldsymbol{x}_{B_2}) \oplus {}^N\boldsymbol{x}_{B_0}) \; ; \; \boxed{{}^{B_2}\boldsymbol{x}_{B_1} = (\ominus {}^N\boldsymbol{x}_{B_2}) \oplus {}^N\boldsymbol{x}_{B_1})}
$$

$$
[\boldsymbol{z}_{p_1}, \boldsymbol{R}_{p_1}] = Register({}^{B_2}\boldsymbol{S}_2, {}^{B_0}\boldsymbol{S}_0, {}^{B_2}\boldsymbol{x}_{B_0}) \; ; \; \boxed{[\boldsymbol{z}_{p_2}, \boldsymbol{R}_{p_2}] = Register({}^{B_2}\boldsymbol{S}_2, {}^{B_1}\boldsymbol{S}_1, {}^{B_2}\boldsymbol{x}_{B_1})}
$$

$$
\boldsymbol{z}_p = [\boldsymbol{z}_{p_1} \quad \boldsymbol{z}_{p_2}] \; ; \; \boxed{\boldsymbol{R}_p = [\boldsymbol{R}_{p_1} \quad \boldsymbol{R}_{p_2}]}
$$



Method Localization$(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0)$

$[{}^N\hat{\boldsymbol{x}}_0, {}^N\boldsymbol{P}_0] = [{}^N\hat{\boldsymbol{x}}_{B_0}, {}^N\boldsymbol{P}_{B_0}]$; // Initialize SLAM state vector

$[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}] = GetScan()$; // Read the Scan from the sensor

$[{}^N\hat{\hat{\boldsymbol{x}}}_0, {}^N\boldsymbol{P}_0] = AddNewPose({}^N\hat{\boldsymbol{x}}_0, {}^N\bar{\boldsymbol{P}}_0)$; // Grow the state vector

$\mathcal{M} = [[{}^{B_0}\boldsymbol{S}_0, {}^{B_0}\boldsymbol{R}_{S_0}]]$; // Store the scan in the map

for $k = 1$ *to steps* **do**

  $[\boldsymbol{u}_k, \boldsymbol{Q}_k] = GetInput()$; // Get input to the motion model

  $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = Prediction({}^N\hat{\boldsymbol{x}}_{k-1}, {}^N\boldsymbol{P}_{k-1}, \boldsymbol{u}_k, \boldsymbol{Q}_k)$;

  $[\boldsymbol{z}_m, \boldsymbol{R}_m] = GetMeasurement()$; // Read navigation sensors

  $\boldsymbol{z}_p = [\,]; \boldsymbol{z}_p = [\,]; \mathcal{H}_p = [\,]$;

  if *ScanAvailable* **then**

    $[{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}] = GetScan()$;

    $[{}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k] = AddNewPose({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k)$; // Grow the state vector

    $\mathcal{M}[k] = [{}^{B_k}\boldsymbol{S}_k, {}^{B_k}\boldsymbol{R}_{S_k}]$; // Store the scan in the map

    $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_k, \mathcal{M})$; // Get pairs of overlapping scans

    $c = 1$; $\mathcal{H}_p = [\,]$;

    for $i = 1$ *to* $length(\mathcal{H}_o)$ **do** // for all overlaps

      $j = \mathcal{H}_o[i]$; // Get the pair: ${}^{B_j}S_j$ overlaps ${}^{B_k}S_k$

      $[{}^{B_j}\boldsymbol{S}_j, {}^{B_j}\boldsymbol{R}_{S_j}] = \mathcal{M}[j]$; // Get the scans from the map

      ${}^{B_j}\boldsymbol{x}_{B_k} = (\ominus{}^N\boldsymbol{x}_{B_j}) \oplus {}^N\boldsymbol{x}_{B_k}$; // Scan Displacement guess

      ${}^{B_j}\boldsymbol{P}_{B_k} = \boldsymbol{J}_{1\oplus}\boldsymbol{J}_{\ominus}{}^N\boldsymbol{P}_{B_j}\boldsymbol{J}_{\ominus}^T\boldsymbol{J}_{1\oplus}^T + \boldsymbol{J}_{2\oplus}{}^N\boldsymbol{P}_{B_k}\boldsymbol{J}_{2\oplus}^T$

      $[\boldsymbol{z}_r, \boldsymbol{R}_r] = Register({}^{B_j}\boldsymbol{S}_j, {}^{B_k}\boldsymbol{S}_k, {}^{B_j}\boldsymbol{x}_{B_k})$; // Scan displacement mean & cov

      if $IndividuallyCompatible({}^{B_j}\boldsymbol{x}_{B_k}, {}^{B_j}\boldsymbol{P}_{B_k}, \boldsymbol{z}_r, \boldsymbol{R}_r, \alpha)$ **then**

        $\boldsymbol{z}_p[c] = \boldsymbol{z}_r$; $\boldsymbol{R}_p[c] = \boldsymbol{R}_r$; // Accepted registration

        $\mathcal{H}_p[c] = i$; $c = c + 1$;

  $[\boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z}_m, \boldsymbol{R}_m, \boldsymbol{z}_p, \boldsymbol{R}_p)$;

  $[{}^N\hat{\boldsymbol{x}}_k, {}^N\boldsymbol{P}_k] = Update({}^N\hat{\bar{\boldsymbol{x}}}_k, {}^N\bar{\boldsymbol{P}}_k, \boldsymbol{z}_k, \boldsymbol{R}_k, \boldsymbol{H}_k, \boldsymbol{V}_k, \mathcal{H}_p)$;

$$
{}^N\hat{\bar{\boldsymbol{x}}}_2 = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ {}^N\hat{\boldsymbol{x}}_{B_1} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_2} \end{bmatrix} \;;\; {}^N\bar{\boldsymbol{P}}_2 = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & {}^N\boldsymbol{P}_{B_0B_1} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} & {}^N\bar{\boldsymbol{P}}_{B_0B_2} \\ {}^N\boldsymbol{P}_{B_1B_0} & {}^N\boldsymbol{P}_{B_1} & {}^N\bar{\boldsymbol{P}}_{B_1,B_2} & {}^N\bar{\boldsymbol{P}}_{B_1B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \\ {}^N\bar{\boldsymbol{P}}_{B_2B_0} & {}^N\bar{\boldsymbol{P}}_{B_2B_1} & {}^N\bar{\boldsymbol{P}}_{B_2} & {}^N\bar{\boldsymbol{P}}_{B_2} \end{bmatrix}
$$

$$
\mathcal{M}_2 = \left[[{}^{B_0}\boldsymbol{S}_0, {}^{B0}\boldsymbol{R}_{S_0}]\; [{}^{B_1}\boldsymbol{S}_1, {}^{B1}\boldsymbol{R}_{S_1}]\; [{}^{B_2}\boldsymbol{S}_2, {}^{B2}\boldsymbol{R}_{S_2}]\right] \quad [\boldsymbol{z}_m \quad \boldsymbol{R}_m]
$$

$$
\mathcal{H}_p = [0 \quad 1]
$$

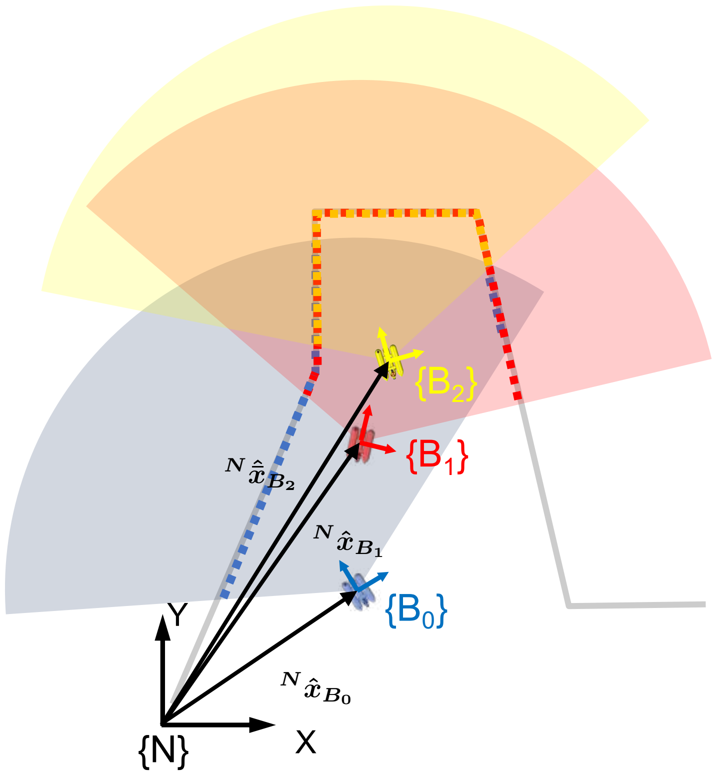
$$
{}^{B_2}\boldsymbol{x}_{B_0} = (\ominus{}^N\boldsymbol{x}_{B_2}) \oplus {}^N\boldsymbol{x}_{B_0}) \;;\; {}^{B_2}\boldsymbol{x}_{B_1} = (\ominus{}^N\boldsymbol{x}_{B_2}) \oplus {}^N\boldsymbol{x}_{B_1})
$$

$$
[\boldsymbol{z}_{p_1}, \boldsymbol{R}_{p_1}] = Register({}^{B_2}\boldsymbol{S}_2, {}^{B_0}\boldsymbol{S}_0, {}^{B_2}\boldsymbol{x}_{B_0}) \;;\; [\boldsymbol{z}_{p_2}, \boldsymbol{R}_{p_2}] = Register({}^{B_2}\boldsymbol{S}_2, {}^{B_1}\boldsymbol{S}_1, {}^{B_2}\boldsymbol{x}_{B_1})
$$

$$
\boldsymbol{z}_p = [\boldsymbol{z}_{p_1} \quad \boldsymbol{z}_{p_2}] \;;\; \boldsymbol{R}_p = [\boldsymbol{R}_{p_1} \quad \boldsymbol{R}_{p_2}]
$$

$$
\boldsymbol{z}_k = \begin{bmatrix} \boldsymbol{z}_m \\ \boldsymbol{z}_p \end{bmatrix} \;;\; \boldsymbol{R}_k = \begin{bmatrix} \boldsymbol{R}_m & 0 \\ 0 & \boldsymbol{R}_p \end{bmatrix}
$$

$$
\boldsymbol{H}_k = \begin{bmatrix} \boldsymbol{H}_m & 0 \\ 0 & \boldsymbol{H}_p \end{bmatrix} \;;\; \boldsymbol{V}_k = \begin{bmatrix} \boldsymbol{I} & 0 \\ 0 & \boldsymbol{I} \end{bmatrix}
$$



Method Localization($\hat{x}_0, P_0$)

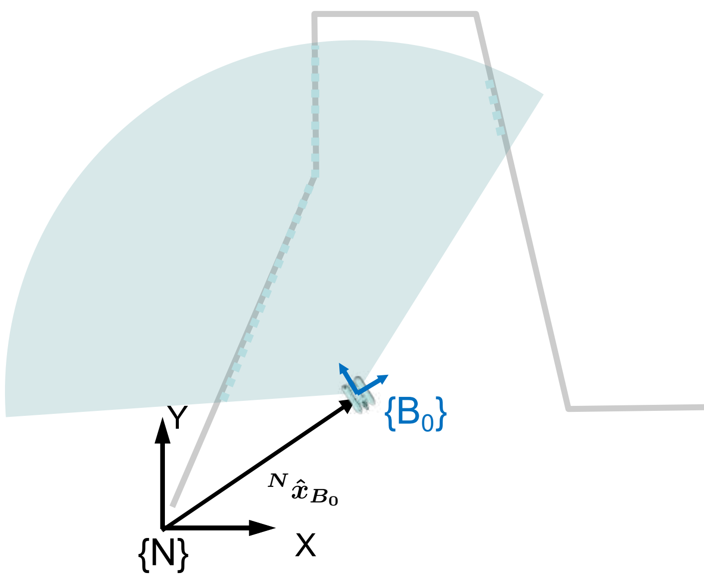
$[{}^N\hat{x}_0, {}^N P_0] = [{}^N\hat{x}_{B_0}, {}^N P_{B_0}]$; // Initialize SLAM state vector
$[{}^{B_0}S_0, {}^{B_0}R_{S_0}] = GetScan()$; // Read the Scan from the sensor
$[{}^N\hat{\bar{x}}_0, {}^N P_0] = AddNewPose({}^N\hat{x}_0, {}^N\bar{P}_0)$; // Grow the state vector
$\mathcal{M} = [[{}^{B_0}S_0, {}^{B_0}R_{S_0}]]$; // Store the scan in the map
for $k = 1$ *to steps* **do**
  $[u_k, Q_k] = GetInput()$; // Get input to the motion model
  $[{}^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = Prediction({}^N\hat{x}_{k-1}, {}^N P_{k-1}, u_k, Q_k)$;
  $[z_m, R_m] = GetMeasurement()$; // Read navigation sensors
  $z_p = [\,]; z_p = [\,]; \mathcal{H}_p = [\,]$;
  if *ScanAvailable* **then**
    $[{}^{B_k}S_k, {}^{B_k}R_{S_k}] = GetScan()$;
    $[{}^N\hat{\bar{x}}_k, {}^N\bar{P}_k] = AddNewPose({}^N\hat{\bar{x}}_k, {}^N\bar{P}_k)$; // Grow the state vector
    $\mathcal{M}[k] = [{}^{B_k}S_k, {}^{B_k}R_{S_k}]$; // Store the scan in the map
    $\mathcal{H}_o = OverlappingScans({}^N\hat{\bar{x}}_k, \mathcal{M})$; // Get pairs of overlapping scans
    $c = 1$; $\mathcal{H}_p = [\,]$;
    for $i = 1$ *to length*$(\mathcal{H}_o)$ **do** // for all overlaps
      $j = \mathcal{H}_o[i]$; // Get the pair: ${}^{B_j}S_j$ overlaps ${}^{B_k}S_k$
      $[{}^{B_j}S_j, {}^{B_j}R_{S_j}] = \mathcal{M}[j]$; // Get the scans from the map
      ${}^{B_j}x_{B_k} = (\ominus {}^N x_{B_j}) \oplus {}^N x_{B_k}$; // Scan Displacement guess
      ${}^{B_j}P_{B_k} = J_{1\oplus} J_{\ominus} {}^N P_{B_j} J_{\ominus}^T J_{1\oplus}^T + J_{2\oplus} {}^N P_{B_k} J_{2\oplus}^T$
      $[z_r, R_r] = Register({}^{B_j}S_j, {}^{B_k}S_k, {}^{B_j}x_{B_k})$; // Scan displacement mean & cov
      if $IndividuallyCompatible({}^{B_j}x_{B_k}, {}^{B_j}P_{B_k}, z_r, R_r, \alpha)$ **then** // Accepted registration
        $z_p[c] = z_r$; $R_p[c] = R_r$;
        $\mathcal{H}_p[c] = i$; $c = c + 1$;
  $[z_k, R_k, H_k, V_k] = ObservationMatrix(\mathcal{H}_p, {}^N\hat{\bar{x}}_k, z_m, R_m, z_p, R_p)$;
  $[{}^N\hat{x}_k, {}^N P_k] = Update({}^N\hat{\bar{x}}_k, {}^N\bar{P}_k, z_k, R_k, H_k, V_k, \mathcal{H}_p)$;

$$
{}^N\hat{x}_2 = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ {}^N\hat{x}_{B_1} \\ {}^N\hat{x}_{B_2} \\ {}^N\hat{x}_{B_2} \end{bmatrix} \; ; \; {}^N P_2 = \begin{bmatrix} {}^N P_{B_0} & {}^N P_{B_0 B_1} & {}^N P_{B_0 B_2} & {}^N P_{B_0 B_2} \\ {}^N P_{B_1 B_0} & {}^N P_{B_1} & {}^N P_{B_1, B_2} & {}^N P_{B_1 B_2} \\ {}^N P_{B_2 B_0} & {}^N P_{B_2 B_1} & {}^N P_{B_2} & {}^N P_{B_2} \\ {}^N P_{B_2 B_0} & {}^N P_{B_2 B_1} & {}^N P_{B_2} & {}^N P_{B_2} \end{bmatrix}
$$

$$
\mathcal{M}_2 = \left[[{}^{B_0}S_0, {}^{B0}R_{S_0}] \; [{}^{B_1}S_1, {}^{B1}R_{S_1}] \; [{}^{B_2}S_2, {}^{B2}R_{S_2}] \right]
$$

EKF Pose-Based SLAM

Add a New Viewpoint Pose



> Previous State Vector

$$
{}^N\hat{\bar{\boldsymbol{x}}}_k = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ \vdots \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_k} \end{bmatrix} \; ; \; {}^N\bar{\boldsymbol{P}}_k = \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & \dots & {}^N\bar{\boldsymbol{P}}_{B_{0,k}} \\ \vdots & \ddots & \vdots \\ {}^N\bar{\boldsymbol{P}}_{B_{k,0}} & \dots & {}^N\bar{\boldsymbol{P}}_{B_k} \end{bmatrix}
$$

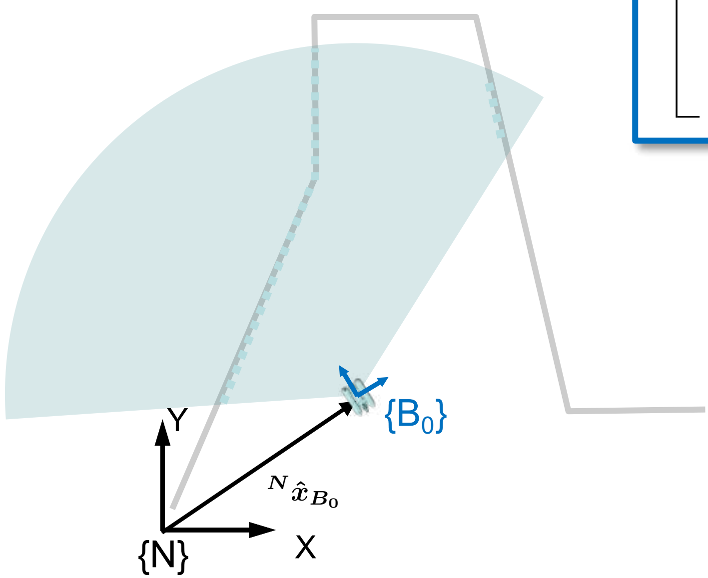
> Grow the state vector cloning the robot pose

$$
{}^N\hat{\bar{\boldsymbol{x}}}_k^{\,+} = \left[\begin{array}{cccc} \boldsymbol{I} & 0 & \dots & 0 \\ 0 & \boldsymbol{I} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{I} \\ \hline 0 & 0 & \dots & \boldsymbol{I} \end{array}\right] \cdot \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{B_0} \\ \vdots \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_k} \end{bmatrix} = \left[\begin{array}{c} {}^N\hat{\boldsymbol{x}}_{B_0} \\ \vdots \\ {}^N\hat{\bar{\boldsymbol{x}}}_{B_k} \\ \hline {}^N\hat{\bar{\boldsymbol{x}}}_{B_k} \end{array}\right]
$$

$$
{}^N\boldsymbol{P}_k^{+} = \left[\begin{array}{cccc} \boldsymbol{I} & 0 & \dots & 0 \\ 0 & \boldsymbol{I} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \boldsymbol{I} \\ \hline 0 & 0 & \dots & \boldsymbol{I} \end{array}\right] \cdot \begin{bmatrix} {}^N\boldsymbol{P}_{B_0} & \dots & {}^N\bar{\boldsymbol{P}}_{B_{0,k}} \\ \vdots & \ddots & \vdots \\ {}^N\bar{\boldsymbol{P}}_{B_{k,0}} & \dots & {}^N\bar{\boldsymbol{P}}_{B_k} \end{bmatrix} \cdot \left[\begin{array}{cccc|c} \boldsymbol{I} & 0 & \dots & 0 & 0 \\ 0 & \boldsymbol{I} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \boldsymbol{I} & \boldsymbol{I} \end{array}\right]
$$

$$
= \left[\begin{array}{ccc|c} {}^N\boldsymbol{P}_{B_0} & \dots & {}^N\bar{\boldsymbol{P}}_{B_{0,k}} & {}^N\bar{\boldsymbol{P}}_{B_{0,k}} \\ \vdots & \ddots & \vdots & \vdots \\ {}^N\bar{\boldsymbol{P}}_{B_{k,0}} & \dots & {}^N\bar{\boldsymbol{P}}_{B_k} & {}^N\bar{\boldsymbol{P}}_{B_k} \\ \hline {}^N\bar{\boldsymbol{P}}_{B_{k,0}} & \dots & {}^N\bar{\boldsymbol{P}}_{B_k} & {}^N\bar{\boldsymbol{P}}_{B_k} \end{array}\right]
$$

EKF Pose-Based SLAM



Add a New Viewpoint Pose

Method `AddNewPose`($^N\hat{x}_k, {}^NP_k$)

$$^N\hat{x}_k{}^+ = \left[\begin{array}{c} ^N\hat{x}_k \\ \hline ^N\hat{x}_{B_k} \end{array}\right]; \quad ^NP_k^+ = \left[\begin{array}{c|c} ^NP_k & ^NP_{0\dots k,B_k} \\ \hline ^NP_{B_k,0\dots k} & ^NP_{B_k} \end{array}\right];$$

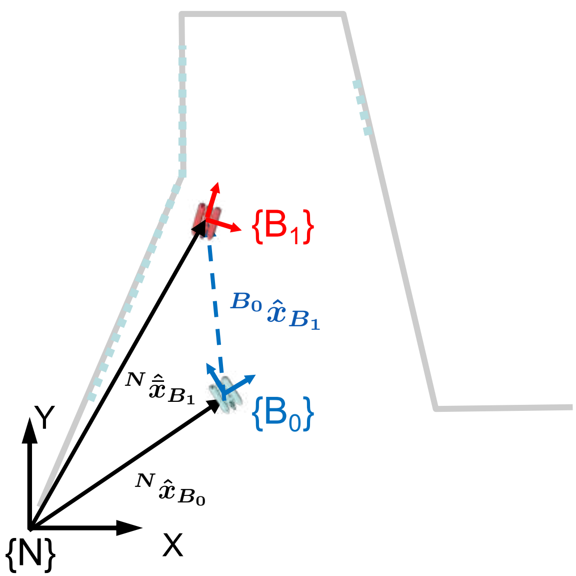
return $[^N\hat{x}_k^+, {}^N\bar{P}_k^+]$

$$^N\hat{x}_k{}^+ = \left[\begin{array}{cccc} I & 0 & \dots & 0 \\ 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \\ \hline 0 & 0 & \dots & I \end{array}\right] \cdot \left[\begin{array}{c} ^N\hat{x}_{B_0} \\ \vdots \\ ^N\hat{x}_{B_k} \end{array}\right] = \left[\begin{array}{c} ^N\hat{x}_{B_0} \\ \vdots \\ ^N\hat{x}_{B_k} \\ \hline ^N\hat{x}_{B_k} \end{array}\right]$$

$$^NP_k^+ = \left[\begin{array}{cccc} I & 0 & \dots & 0 \\ 0 & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & I \\ \hline 0 & 0 & \dots & I \end{array}\right] \cdot \left[\begin{array}{ccc} ^NP_{B_0} & \dots & ^N\bar{P}_{B_{0,k}} \\ \vdots & \ddots & \vdots \\ ^N\bar{P}_{B_{k,0}} & \dots & ^N\bar{P}_{B_k} \end{array}\right] \cdot \left[\begin{array}{cccc|c} I & 0 & \dots & 0 & 0 \\ 0 & I & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & I & I \end{array}\right]$$

$$= \left[\begin{array}{ccc|c} ^NP_{B_0} & \dots & ^N\bar{P}_{B_{0,k}} & ^N\bar{P}_{B_{0,k}} \\ \vdots & \ddots & \vdots & \vdots \\ ^N\bar{P}_{B_{k,0}} & \dots & ^N\bar{P}_{B_k} & ^N\bar{P}_{B_k} \\ \hline ^N\bar{P}_{B_{k,0}} & \dots & ^N\bar{P}_{B_k} & ^N\bar{P}_{B_k} \end{array}\right]$$

EKF Pose-Based SLAM



Prediction

> Previous State Vector

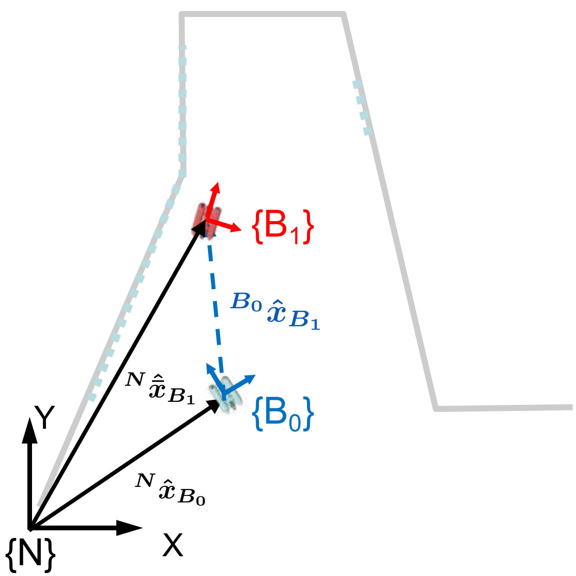
$$
{}^N\hat{x}_{k-1} = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ \vdots \\ {}^N\hat{x}_{B_{k-2}} \\ \boxed{{}^N\hat{x}_{B_{k-1}}} \end{bmatrix} \; ; \; {}^NP_{k-1} = \begin{bmatrix} {}^NP_{B_0} & \cdots & {}^NP_{B_{0,k-1}} \\ \vdots & \ddots & \vdots \\ {}^NP_{B_{k-1,0}} & \cdots & {}^NP_{B_{k-1}} \end{bmatrix}
$$

> Predicted State Vector

$$
{}^N\hat{\bar{x}}_k = f({}^N\hat{x}_{k-1}, u_k, \hat{w}_k) = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ \vdots \\ {}^N\hat{x}_{B_{k-2}} \\ \boxed{f({}^N\hat{x}_{B_{k-1}}, u_k, \hat{w}_k)} \end{bmatrix}
$$

$$
{}^N\bar{P}_k = F_{1_k} \, {}^NP_{k-1} F_{1_k}^T + F_{2_k} Q_k F_{2_k}^T
$$

EKF Pose-Based SLAM



Prediction

> Previous State Vector

$$ {}^N\hat{x}_{k-1} = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ \vdots \\ {}^N\hat{x}_{B_{k-2}} \\ {}^N\hat{x}_{B_{k-1}} \end{bmatrix} \; ; \; {}^N P_{k-1} = \begin{bmatrix} {}^N P_{B_0} & \dots & {}^N P_{B_{0,k-1}} \\ \vdots & \ddots & \vdots \\ {}^N P_{B_{k-1,0}} & \dots & {}^N P_{B_{k-1}} \end{bmatrix} $$

> Predicted State Vector

$$ {}^N\hat{\bar{x}}_k = \mathfrak{f}({}^N\hat{x}_{k-1}, u_k, \hat{w}_k) = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ \vdots \\ {}^N\hat{x}_{B_{k-2}} \\ f({}^N\hat{x}_{B_{k-1}}, u_k, \hat{w}_k) \end{bmatrix} $$

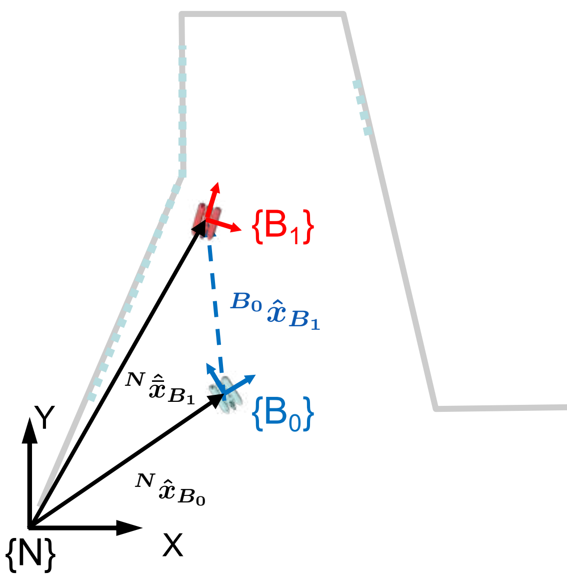
$$ {}^N\bar{P}_k = \boxed{F_{1_k}}\,{}^N P_{k-1}\boxed{F_{1_k}^T} + F_{2_k} Q_k F_{2_k}^T $$

$$ F_{1_k} = \frac{\partial \mathfrak{f}({}^N x_{k-1}, u_k, w_k)}{\partial {}^N x_{k-1}} $$

$$ = \begin{bmatrix} \frac{\partial {}^N x_{B_0}}{\partial {}^N x_{B_0}} & \frac{\partial {}^N x_{B_0}}{\partial {}^N x_{B_1}} & \dots & \frac{\partial {}^N x_{B_0}}{\partial {}^N x_{B_{k-2}}} & \frac{\partial {}^N x_{B_0}}{\partial {}^N x_{B_{k-1}}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial {}^N x_{B_{k-2}}}{\partial {}^N x_{B_0}} & \frac{\partial {}^N x_{B_{k-2}}}{\partial {}^N x_{B_1}} & \dots & \frac{\partial {}^N x_{B_{k-2}}}{\partial {}^N x_{B_{k-2}}} & \frac{\partial {}^N x_{B_{k-2}}}{\partial {}^N x_{B_{k-1}}} \\ \frac{\partial f({}^N x_{B_{k-1}}, u_k, w_k)}{\partial {}^N x_{B_0}} & \frac{\partial f({}^N x_{B_{k-1}}, u_k, w_k)}{\partial {}^N x_{B_1}} & \dots & \frac{\partial f({}^N x_{B_{k-1}}^T, u_k^T, w_k)}{\partial {}^N x_{B_{k-2}}} & \frac{\partial f({}^N x_{B_{k-1}}^T, u_k^T, w_k)}{\partial {}^N x_{B_{k-1}}} \end{bmatrix} $$

$$ = \begin{bmatrix} I & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & I & 0 \\ 0 & \dots & 0 & J_{f_x} \end{bmatrix} $$

EKF Pose-Based SLAM



Prediction

> Previous State Vector

$$ {}^N\hat{x}_{k-1} = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ \vdots \\ {}^N\hat{x}_{B_{k-2}} \\ {}^N\hat{x}_{B_{k-1}} \end{bmatrix} \; ; \; {}^N P_{k-1} = \begin{bmatrix} {}^N P_{B_0} & \dots & {}^N P_{B_{0,k-1}} \\ \vdots & \ddots & \vdots \\ {}^N P_{B_{k-1,0}} & \dots & {}^N P_{B_{k-1}} \end{bmatrix} $$

> Predicted State Vector

$$ {}^N\hat{\bar{x}}_k = \mathfrak{f}({}^N\hat{x}_{k-1}, u_k, \hat{w}_k) = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ \vdots \\ {}^N\hat{x}_{B_{k-2}} \\ f({}^N\hat{x}_{B_{k-1}}, u_k, \hat{w}_k) \end{bmatrix} $$

$$ {}^N\bar{P}_k = F_{1_k} {}^N P_{k-1} F_{1_k}^T + \boxed{F_{2_k}} Q_k \boxed{F_{2_k}^T} $$

$$ \boxed{F_{2_k} = \frac{\partial \mathfrak{f}({}^N x_{k-1}, u_k, w_k)}{\partial w_k} = \begin{bmatrix} \frac{\partial {}^N x_{B_0}}{\partial w_k} \\ \vdots \\ \frac{\partial {}^N x_{B_{k-2}}}{\partial w_k} \\ \frac{\partial f({}^N x_{B_{k-1}}, u_k, w_k)}{\partial w_k} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ J_{f_w} \end{bmatrix}} $$

$$ F_{1_k} = \frac{\partial \mathfrak{f}({}^N x_{k-1}, u_k, w_k)}{\partial {}^N x_{k-1}} $$

$$ = \begin{bmatrix} \frac{\partial {}^N x_{B_0}}{\partial {}^N x_{B_0}} & \frac{\partial {}^N x_{B_0}}{\partial {}^N x_{B_1}} & \dots & \frac{\partial {}^N x_{B_0}}{\partial {}^N x_{B_{k-2}}} & \frac{\partial {}^N x_{B_0}}{\partial {}^N x_{B_{k-1}}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial {}^N x_{B_{k-2}}}{\partial {}^N x_{B_0}} & \frac{\partial {}^N x_{B_{k-2}}}{\partial {}^N x_{B_1}} & \dots & \frac{\partial {}^N x_{B_{k-2}}}{\partial {}^N x_{B_{k-2}}} & \frac{\partial {}^N x_{B_{k-2}}}{\partial {}^N x_{B_{k-1}}} \\ \frac{\partial f({}^N x_{B_{k-1}}, u_k, w_k)}{\partial {}^N x_{B_0}} & \frac{\partial f({}^N x_{B_{k-1}}, u_k, w_k)}{\partial {}^N x_{B_1}} & \dots & \frac{\partial f({}^N x^T_{B_{k-1}}, u^T_k, w_k)}{\partial {}^N x_{B_{k-2}}} & \frac{\partial f({}^N x^T_{B_{k-1}}, u^T_k, w_k)}{\partial {}^N x_{B_{k-1}}} \end{bmatrix} $$

$$ = \begin{bmatrix} I & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & I & 0 \\ 0 & \dots & 0 & J_{f_x} \end{bmatrix} $$

EKF Pose-Based SLAM

Prediction

$$
\begin{aligned}
{}^N\bar{P}_k &= F_{1_k}\,{}^N P_{k-1} F_{1_k}^T + F_{2_k} Q_k F_{2_k}^T \\
&= \begin{bmatrix} I & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & I & 0 \\ 0 & \dots & 0 & J_{f_x} \end{bmatrix}
\begin{bmatrix} {}^N P_{B_0} & \dots & {}^N P_{B_{0,k-1}} \\ \vdots & \ddots & \vdots \\ {}^N P_{B_{k-1,0}} & \dots & {}^N P_{B_{k-1}} \end{bmatrix}
\begin{bmatrix} I & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & I & 0 \\ 0 & \dots & 0 & J_{f_x}^T \end{bmatrix} \\
&+ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ J_{f_w} \end{bmatrix} Q_k \begin{bmatrix} 0 & 0 & \dots & J_{f_w}^T \end{bmatrix} \\
&= \left[\begin{array}{ccc|c} {}^N P_{B_0} & \dots & {}^N P_{B_{0,k-2}} & {}^N P_{B_{0,k-1}} J_{f_x}^T \\ \vdots & \ddots & \vdots & \vdots \\ {}^N P_{B_{k-1,0}} & \dots & {}^N P_{B_{k-2}} & {}^N P_{B_{k-2,k-0}} J_{f_x}^T \\ \hline J_{f_x}\,{}^N P_{B_{k-1,0}} & \dots & J_{f_x}\,{}^N P_{B_{k-1,k-2}} & J_{f_x}\,{}^N P_{B_{k-1}} J_{f_x}^T \end{array}\right]
+ \left[\begin{array}{ccc|c} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \dots & J_{f_w} Q_k J_{f_w}^T \end{array}\right] \\
&= \left[\begin{array}{c|c} {}^N P_{B_{0\dots k-2,0\dots k-2}} & {}^N P_{B_{0\dots k-2,k-1}} J_{f_x}^T \\ \hline J_{f_x}\,{}^N P_{B_{k-1,0\dots k-2}} & J_{f_x}\,{}^N P_{B_{k-1}} J_{f_x}^T + J_{f_w} Q_k J_{f_w}^T \end{array}\right].
\end{aligned}
$$

EKF Pose-Based SLAM

Prediction

$${}^N\bar{P}_k = F_{1_k}{}^N P_{k-1} F_{1_k}^T + F_{2_k} Q_k F_{2_k}^T$$

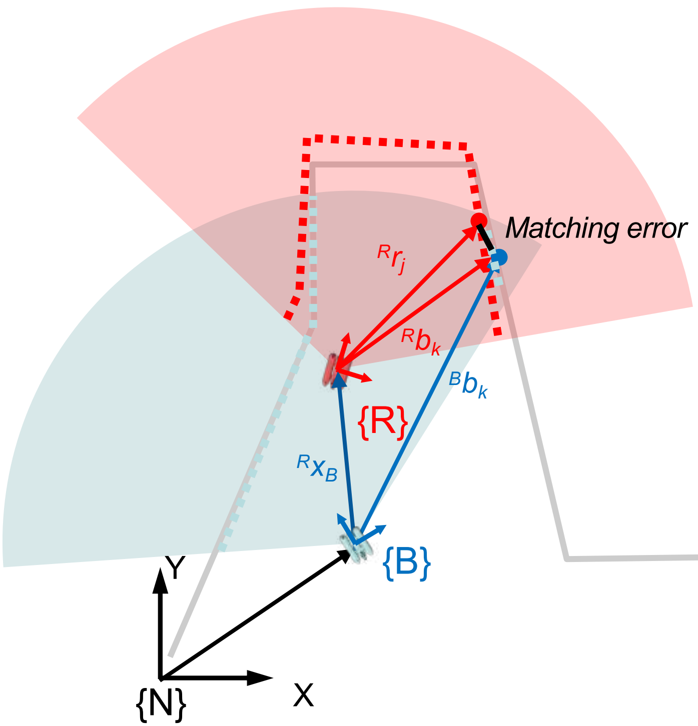
$$\mathbf{Method}\ \texttt{Prediction}({}^N\hat{x}_{k-1}, {}^N P_{k-1}, {}^N\hat{\bar{x}}_k, u_k, Q_k)$$

$${}^N\hat{\bar{x}}_k = \begin{bmatrix} {}^N\hat{x}_{B_0} \\ \vdots \\ {}^N\hat{x}_{B_{k-2}} \\ f({}^N\hat{x}_{B_{k-1}}, u_k, \hat{w}_k) \end{bmatrix};$$

$${}^N\bar{x}_k = \left[\begin{array}{c|c} {}^N P_{B_{1\ldots k-2,1\ldots k-2}} & {}^N P_{B_{1\ldots k-2,k-1}} J_{f_x}^T \\ \hline J_{f_x}{}^N P_{B_{k-1,1\ldots k-2}} & J_{f_x}{}^N P_{B_{k-1}} J_{f_x}^T + J_{f_w} Q_k J_{f_w}^T \end{array}\right];$$

$$\mathbf{return}\ [{}^N\hat{\bar{x}}_k, {}^N\bar{P}_k]$$

$$= \left[\begin{array}{c|c} {}^N P_{B_{0\ldots k-2,0\ldots k-2}} & {}^N P_{B_{0\ldots k-2,k-1}} J_{f_x}^T \\ \hline J_{f_x}{}^N P_{B_{k-1,0\ldots k-2}} & J_{f_x}{}^N P_{B_{k-1}} J_{f_x}^T + J_{f_w} Q_k J_{f_w}^T \end{array}\right].$$



Registration

Function ICP($^{\mathcal{R}}\boldsymbol{S_r}, {}^{\mathcal{B}}\boldsymbol{S_b}, {}^{\mathcal{R}}\boldsymbol{x_B}$)

  // **New Scan:** $^{\mathcal{R}}\boldsymbol{S_r} = [{}^{\mathcal{R}}\boldsymbol{r_1} \; {}^{\mathcal{R}}\boldsymbol{r_2} \; \dots \; {}^{\mathcal{R}}\boldsymbol{r_N}]$

  // **Reference Scan:** $^{\mathcal{B}}\boldsymbol{S_b} = [{}^{\mathcal{B}}\boldsymbol{b_1} \; {}^{\mathcal{B}}\boldsymbol{b_2} \; \dots \; {}^{\mathcal{B}}\boldsymbol{b_S}]$

  // **Initial Transformation guess:** $^{\mathcal{R}}\boldsymbol{x_B}$

  $i = 1$;

  while $(i < maxIterations)$ **and(not** $converged$) **do**

    for $j = 1$ **to** N **do** // Every point in the R Scan

      $^{\mathcal{B}}\boldsymbol{c_j} = \underset{^{\mathcal{B}}\boldsymbol{b_k} \in {}^{\mathcal{B}}S_b}{\operatorname{argmin}} \{ || {}^{\mathcal{R}}\boldsymbol{x_B} \oplus {}^{\mathcal{B}}\boldsymbol{b_k} - {}^{\mathcal{R}}\boldsymbol{r_j} || \}$; // Compute the Nearest Neighbor

    end

    // LS minimization of the Matching error

    $^{\mathcal{R}}\boldsymbol{x_B} = \underset{^{\mathcal{R}}\boldsymbol{x_B}}{\operatorname{argmin}} \{ \sum_j^N \boldsymbol{e_j^T} \cdot \boldsymbol{e_j} \}$ where $\boldsymbol{e_j} = {}^{\mathcal{R}}\boldsymbol{x_B} \oplus {}^{\mathcal{B}}\boldsymbol{c_j} - {}^{\mathcal{R}}\boldsymbol{r_j}$;

    $i = i + 1$;

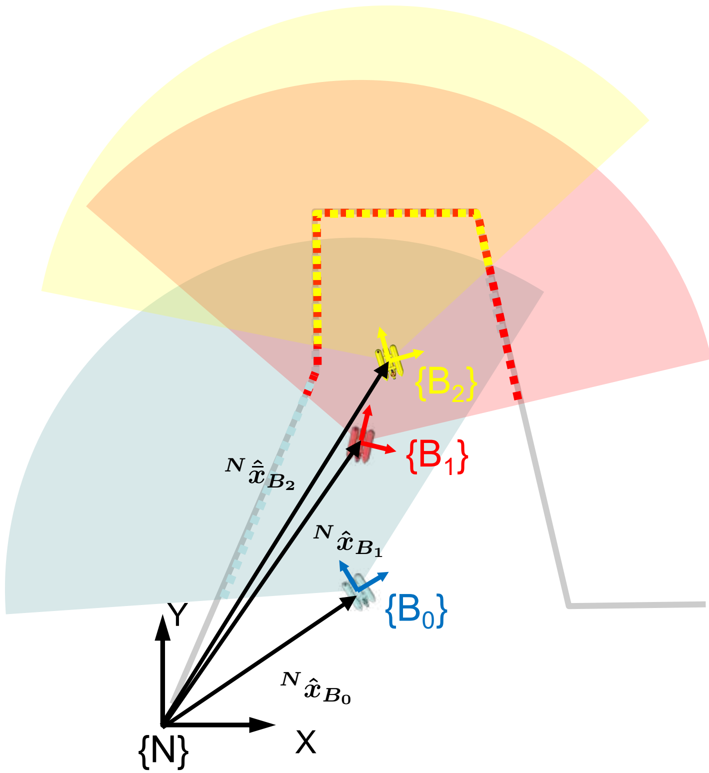
  end

  return $^{\mathcal{R}}\boldsymbol{x_B}$;

end

Algorithm 1: Iterative Closest Point

EKF Pose-Based SLAM



$$B_0\text{–}frame \equiv \mathcal{B}\text{–}frame$$
$$B_1\text{–}frame \equiv \mathcal{R}\text{–}frame$$
$$B_2\text{–}frame \equiv \mathcal{Y}\text{–}frame$$

Pose Constraints

1. Compute overlapping scans

$$\mathcal{H}_p = OverlappingScans({}^N\hat{\bar{\boldsymbol{x}}}_{\boldsymbol{k}}, \mathcal{M})$$
$$\mathcal{H}_p = [0\ 1\ 2]$$

2. Register overlapping scans

$$\boxed{\begin{matrix}[{}^{B_k}\boldsymbol{x}_{\mathcal{Y}}, {}^{B_k}\boldsymbol{R}_{\mathcal{Y}}] \\ [{}^{B_k}\boldsymbol{x}_{\mathcal{B}}, {}^{B_k}\boldsymbol{R}_{\mathcal{B}}] \\ [{}^{B_k}\boldsymbol{x}_{\mathcal{R}}, {}^{B_k}\boldsymbol{R}_{\mathcal{R}}]\end{matrix}}\begin{matrix} = Registration({}^{B_k}\boldsymbol{S}_{\boldsymbol{k}}, {}^{\mathcal{Y}}\boldsymbol{S}_{\boldsymbol{y}}) \\ = Registration({}^{B_k}\boldsymbol{S}_{\boldsymbol{k}}, {}^{\mathcal{B}}\boldsymbol{S}_{\boldsymbol{b}}) \\ = Registration({}^{B_k}\boldsymbol{S}_{\boldsymbol{k}}, {}^{\mathcal{R}}\boldsymbol{S}_{\boldsymbol{r}}),\end{matrix}$$

3. Observation equation

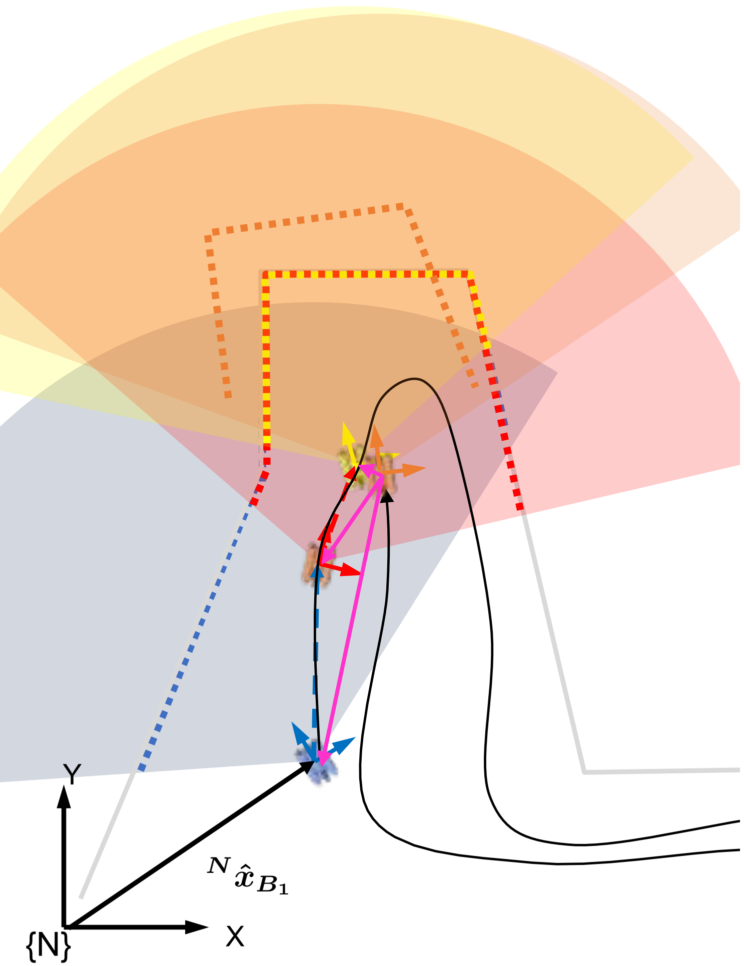
$$\boldsymbol{z_r} = \boldsymbol{h}({}^N\bar{\boldsymbol{x}}_{\boldsymbol{k}}, \boldsymbol{v}_{\boldsymbol{r_k}})$$

$$\begin{bmatrix}\boldsymbol{z_y} \\ \boldsymbol{z_b} \\ \boldsymbol{z_r}\end{bmatrix} = \begin{bmatrix}\boldsymbol{h_r}({}^N\bar{\boldsymbol{x}}_{\boldsymbol{B_k}}, {}^N\boldsymbol{x}_{\mathcal{Y}}, \boldsymbol{v}_{\boldsymbol{y_k}}) \\ \boldsymbol{h_r}({}^N\bar{\boldsymbol{x}}_{\boldsymbol{B_k}}, {}^N\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{v}_{\boldsymbol{b_k}}) \\ \boldsymbol{h_r}({}^N\bar{\boldsymbol{x}}_{\boldsymbol{B_k}}, {}^N\boldsymbol{x}_{\mathcal{R}}, \boldsymbol{v}_{\boldsymbol{r_k}})\end{bmatrix}$$

$$\begin{bmatrix}{}^{B_k}\boldsymbol{x}_{\mathcal{Y}} \\ {}^{B_k}\boldsymbol{x}_{\mathcal{B}} \\ {}^{B_k}\boldsymbol{x}_{\mathcal{R}}\end{bmatrix} = \begin{bmatrix}((\ominus {}^N\bar{\boldsymbol{x}}_{\boldsymbol{B_k}}) \oplus {}^N\boldsymbol{x}_{\mathcal{Y}}) + \boldsymbol{v}_{\boldsymbol{y_k}} \\ ((\ominus {}^N\bar{\boldsymbol{x}}_{\boldsymbol{B_k}}) \oplus {}^N\boldsymbol{x}_{\mathcal{B}}) + \boldsymbol{v}_{\boldsymbol{b_k}} \\ ((\ominus {}^N\bar{\boldsymbol{x}}_{\boldsymbol{B_k}}) \oplus {}^N\boldsymbol{x}_{\mathcal{R}}) + \boldsymbol{v}_{\boldsymbol{r_k}}\end{bmatrix}$$

EKF Pose-Based SLAM

Loop Closing



Pose Constraints

1. Compute overlapping scans

$$\mathcal{H}_p = OverlappingScans({}^{N}\hat{\bar{x}}_k, \mathcal{M})$$
$$\mathcal{H}_p = [0\ 1\ 2]$$

2. Register overlapping scans

$$\begin{aligned}
\boxed{[{}^{B_k}x_{\mathcal{Y}}, {}^{B_k}R_{\mathcal{Y}}]} &= Registration({}^{B_k}S_k, {}^{\mathcal{Y}}S_y)\\
\boxed{[{}^{B_k}x_{\mathcal{B}}, {}^{B_k}R_{\mathcal{B}}]} &= Registration({}^{B_k}S_k, {}^{\mathcal{B}}S_b)\\
\boxed{[{}^{B_k}x_{\mathcal{R}}, {}^{B_k}R_{\mathcal{R}}]} &= Registration({}^{B_k}S_k, {}^{\mathcal{R}}S_r),
\end{aligned}$$

3. Observation equation

$$z_r = h({}^{N}\bar{x}_k, v_{r_k})$$

$$\begin{bmatrix} z_y \\ z_b \\ z_r \end{bmatrix} = \begin{bmatrix} h_r({}^{N}\bar{x}_{B_k}, {}^{N}x_{\mathcal{Y}}, v_{y_k}) \\ h_r({}^{N}\bar{x}_{B_k}, {}^{N}x_{\mathcal{B}}, v_{b_k}) \\ h_r({}^{N}\bar{x}_{B_k}, {}^{N}x_{\mathcal{R}}, v_{r_k}) \end{bmatrix}$$

$$\begin{bmatrix} {}^{B_k}x_{\mathcal{Y}} \\ {}^{B_k}x_{\mathcal{B}} \\ {}^{B_k}x_{\mathcal{R}} \end{bmatrix} = \begin{bmatrix} ((\ominus {}^{N}\bar{x}_{B_k}) \oplus {}^{N}x_{\mathcal{Y}}) + v_{y_k} \\ ((\ominus {}^{N}\bar{x}_{B_k}) \oplus {}^{N}x_{\mathcal{B}}) + v_{b_k} \\ ((\ominus {}^{N}\bar{x}_{B_k}) \oplus {}^{N}x_{\mathcal{R}}) + v_{r_k} \end{bmatrix}$$

EKF Pose-Based SLAM

Observation Jacobians

$$
\boldsymbol{H_k} = \left.\frac{\partial \boldsymbol{h}({}^N\bar{\boldsymbol{x}}_{\boldsymbol{k}}, \boldsymbol{v_k})}{\partial {}^N\bar{\boldsymbol{x}}_{\boldsymbol{k}}}\right|_{{}^N\bar{\boldsymbol{x}}_{\boldsymbol{k}}={}^N\hat{\bar{\boldsymbol{x}}}_{\boldsymbol{k}},\boldsymbol{w_k}=\boldsymbol{0}}
$$

$$
= \begin{array}{c c}
 & \begin{array}{ccccccccccc} 1 & 2 & \dots & y & \dots & b & \dots & r & \dots & n_p \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \end{array} &
\begin{bmatrix}
\frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial{}^N\boldsymbol{x}_{B_0}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial{}^N\boldsymbol{x}_{B_1}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial{}^N\boldsymbol{x}_{\mathcal{Y}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial{}^N\boldsymbol{x}_{\mathcal{B}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial{}^N\boldsymbol{x}_{\mathcal{R}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial{}^N\boldsymbol{x}_{B_k}} \\
\frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial{}^N\boldsymbol{x}_{B_0}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial{}^N\boldsymbol{x}_{B_1}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial{}^N\boldsymbol{x}_{\mathcal{Y}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial{}^N\boldsymbol{x}_{\mathcal{B}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial{}^N\boldsymbol{x}_{\mathcal{R}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial{}^N\boldsymbol{x}_{B_k}} \\
\frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k}\right)}{\partial{}^N\boldsymbol{x}_{B_0}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k}\right)}{\partial{}^N\boldsymbol{x}_{B_1}} & \dots & \frac{\partial \boldsymbol{h}_{\boldsymbol{r}}({}^N\bar{\boldsymbol{x}}_k,\boldsymbol{v}_{y_k})}{\partial{}^N\boldsymbol{x}_{\mathcal{Y}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k}\right)}{\partial{}^N\boldsymbol{x}_{\mathcal{B}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k}\right)}{\partial{}^N\boldsymbol{x}_{\mathcal{R}}} & \dots & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k}\right)}{\partial{}^N\boldsymbol{x}_{B_k}}
\end{bmatrix}
\end{array}
$$

$$
= \begin{array}{c c}
 & \begin{array}{cccccccccc} 1 & 2 & \dots & y & \dots & b & \dots & r & \dots & n_p \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \end{array} &
\begin{bmatrix}
\boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{J_{2\oplus}} & \dots & \dots & \dots & \dots & \dots & \boldsymbol{J_{1\oplus}}\boldsymbol{J_{\ominus}} \\
\boldsymbol{0} & \boldsymbol{0} & \dots & \dots & \dots & \boldsymbol{J_{2\oplus}} & \dots & \dots & \dots & \boldsymbol{J_{1\oplus}}\boldsymbol{J_{\ominus}} \\
\boldsymbol{0} & \boldsymbol{0} & \dots & \dots & \dots & \dots & \dots & \boldsymbol{J_{2\oplus}} & \dots & \boldsymbol{J_{1\oplus}}\boldsymbol{J_{\ominus}}
\end{bmatrix}
\end{array}
$$

$$
\boldsymbol{V_k} = \left.\frac{\partial \boldsymbol{h}({}^N\bar{\boldsymbol{x}}_{\boldsymbol{k}}, \boldsymbol{v_k})}{\partial \boldsymbol{v_k}}\right|_{\bar{\boldsymbol{x}}_{\boldsymbol{k}}={}^N\hat{\bar{\boldsymbol{x}}}_{\boldsymbol{k}},\boldsymbol{w_k}=\boldsymbol{0}}
$$

$$
= \begin{bmatrix}
\frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial\boldsymbol{v}_{y_k}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial\boldsymbol{v}_{b_k}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k}\right)}{\partial\boldsymbol{v}_{r_k}} \\
\frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial\boldsymbol{v}_{y_k}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial\boldsymbol{v}_{b_k}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k}\right)}{\partial\boldsymbol{v}_{r_k}} \\
\frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k}\right)}{\partial\boldsymbol{v}_{y_k}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k}\right)}{\partial\boldsymbol{v}_{b_k}} & \frac{\partial\left((\ominus{}^N\bar{\boldsymbol{x}}_{B_k})\oplus{}^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k}\right)}{\partial\boldsymbol{v}_{r_k}}
\end{bmatrix}
= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}
$$

EKF Pose-Based SLAM

Observation Jacobians

$$H_k = \left.\frac{\partial h({}^N\bar{\boldsymbol{x}}_k, \boldsymbol{v}_k)}{\partial {}^N\bar{\boldsymbol{x}}_k}\right|_{{}^N\bar{\boldsymbol{x}}_k={}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{w}_k=0}$$

$$= \begin{array}{c|cccc} & 1 & 2 & \dots & y \\ \hline 1 & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k})}{\partial^N\boldsymbol{x}_{B_0}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k})}{\partial^N\boldsymbol{x}_{B_1}} & \dots & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k})}{\partial^N\boldsymbol{x}_{\mathcal{Y}}} \\ 2 & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k})}{\partial^N\boldsymbol{x}_{B_0}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k})}{\partial^N\boldsymbol{x}_{B_1}} & \dots & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k})}{\partial^N\boldsymbol{x}_{\mathcal{Y}}} \\ 3 & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k})}{\partial^N\boldsymbol{x}_{B_0}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k})}{\partial^N\boldsymbol{x}_{B_1}} & \dots & \frac{\partial h_r({}^N\bar{\boldsymbol{x}}_k,\boldsymbol{v}_{y_k})}{\partial^N\boldsymbol{x}_{\mathcal{Y}}} \end{array}$$

$$= \left[\begin{array}{c|cccccccccc} & 1 & 2 & \dots & y & \dots & b & \dots & r & \dots & n_p \\ \hline 1 & \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{J_{2\oplus}} & \dots & \dots & \dots & \dots & \dots & \boldsymbol{J_{1\oplus}J_{\ominus}} \\ 2 & \boldsymbol{0} & \boldsymbol{0} & \dots & \dots & \dots & \boldsymbol{J_{2\oplus}} & \dots & \dots & \dots & \boldsymbol{J_{1\oplus}J_{\ominus}} \\ 3 & \boldsymbol{0} & \boldsymbol{0} & \dots & \dots & \dots & \dots & \dots & \boldsymbol{J_{2\oplus}} & \dots & \boldsymbol{J_{1\oplus}J_{\ominus}} \end{array}\right]$$

$$V_k = \left.\frac{\partial h({}^N\bar{\boldsymbol{x}}_k, \boldsymbol{v}_k)}{\partial \boldsymbol{v}_k}\right|_{\bar{\boldsymbol{x}}_k={}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{w}_k=0}$$

$$= \begin{bmatrix} \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k})}{\partial\boldsymbol{v}_{y_k}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k})}{\partial\boldsymbol{v}_{b_k}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{Y}}+\boldsymbol{v}_{y_k})}{\partial\boldsymbol{v}_{r_k}} \\ \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k})}{\partial\boldsymbol{v}_{y_k}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k})}{\partial\boldsymbol{v}_{b_k}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{B}}+\boldsymbol{v}_{b_k})}{\partial\boldsymbol{v}_{r_k}} \\ \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k})}{\partial\boldsymbol{v}_{y_k}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k})}{\partial\boldsymbol{v}_{b_k}} & \frac{\partial((\ominus^N\bar{\boldsymbol{x}}_{B_k})\oplus^N\boldsymbol{x}_{\mathcal{R}}+\boldsymbol{v}_{r_k})}{\partial\boldsymbol{v}_{r_k}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & 0 & 0 \\ 0 & \boldsymbol{I} & 0 \\ 0 & 0 & \boldsymbol{I} \end{bmatrix}$$

Method `ObservationMatrix`$(\mathcal{H}_p, {}^N\hat{\bar{\boldsymbol{x}}}_k, \boldsymbol{z_m}, \boldsymbol{R_m}, \boldsymbol{z_p}, \boldsymbol{R_p})$

- $\boldsymbol{H_p} = \boldsymbol{0}$;
- **for** $i = 1$ *to length*$(\boldsymbol{z_p})$ **do**
  - $\boldsymbol{H_p}[i, 1] = \boldsymbol{J_{1\oplus}}(\ominus^N\hat{\bar{\boldsymbol{x}}}_{B_k}, {}^N\hat{\boldsymbol{x}}_k[\boldsymbol{\mathcal{H}_p}[i]])\boldsymbol{J_{\ominus}}({}^N\hat{\bar{\boldsymbol{x}}}_{B_k})$;
  - $\boldsymbol{H_p}[i, \mathcal{H}_p[i]] = \boldsymbol{J_{2\oplus}}({}^N\hat{\bar{\boldsymbol{x}}}_{B_k})$;
- $\boldsymbol{z_k} = \begin{bmatrix} \boldsymbol{z_m} \\ \boldsymbol{z_p} \end{bmatrix}$; $\boldsymbol{R_k} = \begin{bmatrix} \boldsymbol{R_m} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R_p} \end{bmatrix}$;
- $\boldsymbol{H_k} = \begin{bmatrix} \boldsymbol{J_{h_x}}(\hat{\bar{\boldsymbol{x}}}_k) \\ \boldsymbol{H_p} \end{bmatrix}$;
- $\boldsymbol{V_k} = \begin{bmatrix} \boldsymbol{J_{h_v}}(\hat{\bar{\boldsymbol{x}}}_k) \\ \boldsymbol{V_p} \end{bmatrix}$;
- return $[\boldsymbol{z_k}, \boldsymbol{R_k}, \boldsymbol{H_k}, \boldsymbol{V_k}]$;

Method `h`$({}^N\hat{\boldsymbol{x}}_k, \mathcal{H}_p)$

- $h_{mf} = parent.h(\hat{\bar{\boldsymbol{x}}}_k)$;
- **for** $i = 1$ *to length*$(\mathcal{H}_p)$ **do**
  - $j = \mathcal{H}_p[i]$;
  - **if** $j \neq 0$ **then**
    - $h_{mf} = \begin{bmatrix} h_{mf} \\ \ominus^N\hat{\bar{\boldsymbol{x}}}_{B_k}\oplus^N\boldsymbol{x_k}[j] \end{bmatrix}$;
- **return** $\boldsymbol{h}_{mf}$;

EKF Pose-Based SLAM

Update

1. The Standard EKF Update equations are used:

$$\boldsymbol{K}_k = {}^{\boldsymbol{N}}\bar{\boldsymbol{P}}_k \boldsymbol{H}_k^T (\boldsymbol{H}_k {}^{\boldsymbol{N}}\bar{\boldsymbol{P}}_k \boldsymbol{H}_k^T + \boldsymbol{V}_k \boldsymbol{R}_k \boldsymbol{V}_k^T)^{-1}$$

$${}^{\boldsymbol{N}}\hat{\boldsymbol{x}}_{\boldsymbol{k}} = {}^{\boldsymbol{N}}\hat{\bar{\boldsymbol{x}}}_{\boldsymbol{k}} + \boldsymbol{K}_k \left(\boldsymbol{z}_k - \boldsymbol{h}({}^{\boldsymbol{N}}\hat{\bar{\boldsymbol{x}}}_{\boldsymbol{k}}, \boldsymbol{z}_{\boldsymbol{p}}, \mathcal{H}_p)\right)$$

$${}^{\boldsymbol{N}}\boldsymbol{P}_{\boldsymbol{k}} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) {}^{\boldsymbol{N}}\bar{\boldsymbol{P}}_k (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)^T$$

Pose based EKF SLAM using Point Clouds

Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner


```
Method Localization(x̂_0, P_0)
  [^N x̂_0, ^N P_0] = [^N x̂_{B_0}, ^N P_{B_0}];                      // Initialize SLAM state vector
  [^{B_0} S_0, ^{B_0} R_{S_0}] = GetScan();;                          // Read the Scan from the sensor
  [^N x̄̂_0, ^N P_0] = AddNewPose(^N x̂_0, ^N P̄_0);                    // Grow the state vector
  M = [^{B_0} S_0];                                                   // Store the scan in the map
  for k = 1 to steps do
    [u_k, Q_k] = GetInput();                                          // Get input to the motion model
    [^N x̄̂_k, ^N P_k] = Prediction(^N x̂_{k-1}, ^N P_{k-1}, u_k, Q_k);
    [z_m, R_m] = GetMeasurement();                                    // Read navigation sensors
    if ScanAvailable then
      [^{B_k} S_k, ^{B_k} R_{S_k}] = GetScan();
      [^N x̄̂_k, ^N P̄_k] = AddNewPose(^N x̄̂_k, ^N P̄_k);               // Grow the state vector
      M[k] = ^{B_k} S_k;                                              // Store the scan in the map
      H_p = OverlappingScans(^N x̄̂_k, M);                             // Get pairs of overlapping scans
      for i = 1 to length(H_p) do                                     // for all overlaps
        j = H_p[i];                                                   // Get the pair: ^{B_j} S_j overlaps ^{B_k} S_k
        ^{B_j} S_j = M[j]; ^{B_k} S_k = M[k];                         // Get the scans from the map
        ^{B_j} x_{B_k} = (⊖ ^N x_{B_j}) ⊕ ^N x_{B_k} ;                // Scans Displacement guess
        [z_{p_i}, R_{p_i}] = Register(^{B_j} S_j, ^{B_k} S_k, ^{B_j} x_{B_k});   // Scan displacement mean & cov
        i = i + 1;                                                    // Go to next pair
    H_p = DataAssociation(^N x̄̂_k, ^N P̄_k, z_p, R_p);                 // select compatible registrations
    [z_k, R_k, H_k, V_k] = ObservationMatrix(H_p, ^N x̄̂_k, z_m, R_m, z_p, R_p);
    [^N x̂_k, ^N P_k] = Update(^N x̄̂_k, ^N P̄_k, z_k, R_k, H_k, V_k, H_p);
```

> To implement a PEKFSLAM we need to:

1. Program the ***GetInput(·)*** function
2. Define the **motion model:** $^{N}\bar{x}_{B_k} = f\left({}^{N}x_{B_{k-1}}, u_k, w_k\right)$
   - Compute the motion model Jacobians:

$$J_{f_x} = \frac{\partial f\left({}^{N}x_{B_{k-1}}, u_k, w_k\right)}{\partial {}^{N}x_{B_{k-1}}} \quad , \quad J_{f_w} = \frac{\partial f\left({}^{N}x_{B_{k-1}}, u_k, w_k\right)}{\partial w_k}$$

3. Program the ***GetScan()*** function
4. Define the **observation model:**

$$z_k = h({}^{N}\bar{x}_k, v_{p_k})$$

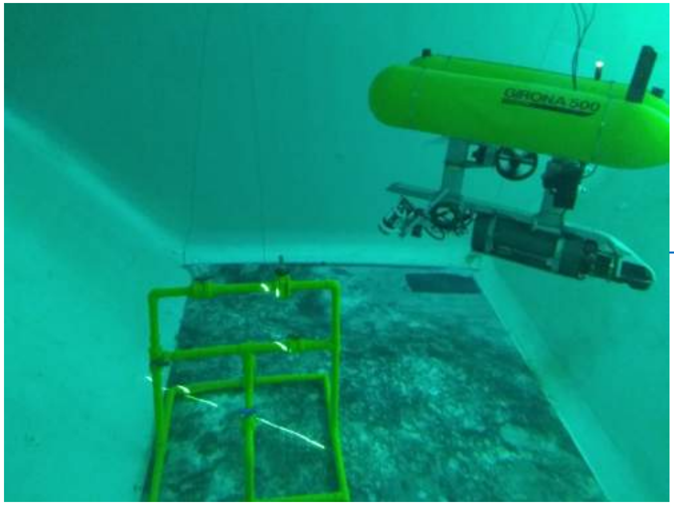
$$\begin{bmatrix} z_{ij} \\ \vdots \\ z_{pq} \end{bmatrix} = \begin{bmatrix} h_{ij}({}^{N}\bar{x}_k, v_{ij_k}) \\ \vdots \\ h_{pq}({}^{N}\bar{x}_k, v_{pq_k}) \end{bmatrix} = \begin{bmatrix} \Delta x_{ij} \\ \vdots \\ \Delta x_{pq} \end{bmatrix}$$

   - Compute the observation Jacobians:

$\forall i, j$ compute:

$$J_{h_x} = \frac{\partial h_{ij}({}^{N}\bar{x}_k, v_k)}{\partial {}^{N}x_{B_k}} \quad , \quad J_{h_v} = \frac{\partial h_{ij}({}^{N}\bar{x}_k, v_k)}{\partial v_k}$$

5. Implement the ***Register(·)*** *function*



GetInput()

$$\boldsymbol{u_k} = \begin{bmatrix} {}^N\phi_B \\ {}^N\theta_B \\ {}^N\psi_B \end{bmatrix}$$

Motion Model

State Vector:

$$\boldsymbol{{}^N x_{B_k}} = [\underbrace{{}^N x_B \ {}^N y_B \ {}^N z_B}_{\boldsymbol{{}^N \eta_{1_{B_k}}}} \ \underbrace{{}^B u_B \ {}^B v_B \ {}^B w_B}_{\boldsymbol{{}^B \nu_k}} \ \boldsymbol{{}^N x_{B_n}} \ \ldots \ \boldsymbol{{}^N x_{B_1}}]^T$$

Motion Model:

$$\boldsymbol{{}^N \bar{x}_{B_k}} = \boldsymbol{f}\left(\boldsymbol{{}^N x_{B_{k-1}}}, \boldsymbol{u_k}, \boldsymbol{w_k}\right) = \begin{bmatrix} \boldsymbol{{}^N \eta_{1_{B_{k-1}}}} + \boldsymbol{{}^N R_B(u_k + w_{\eta_{2_k}})} \left(\boldsymbol{{}^B \nu_{k-1}} \Delta t + \boldsymbol{w_{\dot{\nu}_k}} \frac{\boldsymbol{\Delta t^2}}{\boldsymbol{2}}\right) \\ \boldsymbol{{}^B \nu_{k-1}} + \boldsymbol{w_{\dot{\nu}_k}} \Delta t \end{bmatrix}$$

$$\text{where } \boldsymbol{u_k} = \begin{bmatrix} {}^N\phi_B & {}^N\theta_B & {}^N\psi_B \end{bmatrix}^T$$

$$\boldsymbol{w_k} = \begin{bmatrix} \boldsymbol{w_{\eta_{2_k}}^T} & \boldsymbol{w_{\dot{\nu}_k}^T} \end{bmatrix}^T = \begin{bmatrix} w_\phi & w_\theta & w_\psi & w_{\dot{u}} & w_{\dot{v}} & w_{\dot{w}} \end{bmatrix}^T$$

$$\boldsymbol{w_k} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q_k}); \boldsymbol{Q_k} = diag(\sigma^2_{w_\phi}, \sigma^2_{w_\theta}, \sigma^2_{w_\psi}, \sigma^2_{w_{\dot{u}}}, \sigma^2_{w_{\dot{v}}}, \sigma^2_{w_{\dot{w}}})$$

Jacobians:

$$\boldsymbol{J_{f_x}} = \frac{\partial \boldsymbol{f}\left(\boldsymbol{{}^N x_{B_{k-1}}}, \boldsymbol{u_k}, \boldsymbol{w_k}\right)}{\partial \boldsymbol{{}^N x_{B_{k-1}}}} = \begin{bmatrix} \boldsymbol{I_{3\times3}} & \boldsymbol{{}^N R_B(u_k)} \Delta t \\ \boldsymbol{0_{3\times3}} & \boldsymbol{I_{3\times3}} \end{bmatrix}$$

$$\boldsymbol{J_{f_w}} = \frac{\partial \boldsymbol{f}\left(\boldsymbol{{}^N x_{B_{k-1}}}, \boldsymbol{u_k}, \boldsymbol{w_k}\right)}{\partial \boldsymbol{w_k}} = \begin{bmatrix} \boldsymbol{J_R} \cdot \boldsymbol{{}^B \hat{\nu}_{k-1}} \Delta t & \boldsymbol{{}^N R_B(u_k)} \frac{\Delta t^2}{2} \\ \boldsymbol{0_{3\times3}} & \boldsymbol{I_{3\times3}} \Delta t \end{bmatrix}$$

$$\text{where } \boldsymbol{J_R} = \left.\frac{\partial \boldsymbol{{}^N R_B(u_k + w_{\eta_{2_k}})}}{\partial \boldsymbol{w_{\eta_{2_k}}}}\right|_{\boldsymbol{{}^N x_{B_{k-1}} = {}^N \hat{x}_{B_{k-1}}}, \boldsymbol{w_k = 0}}$$

View poses

> The State Vector contains only the position: $^{N}\boldsymbol{x}_k = [\boldsymbol{\eta}_{1_{B_k}}^T \ \boldsymbol{\nu}_1^T]^T$

> A View Pose can be computed combining the position and the input attitude:

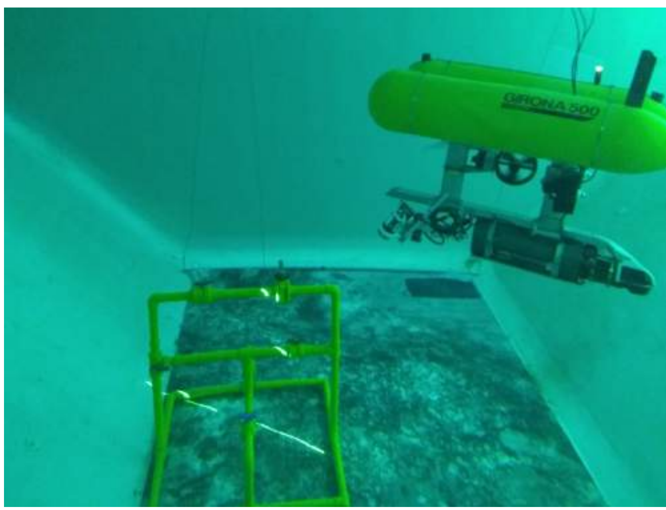
$$\left.\begin{aligned} ^{N}\boldsymbol{x}_k &= [\boldsymbol{\eta}_{1_{B_k}}^T \ \boldsymbol{\nu}_1^T]^T \\ \boldsymbol{u}_k &= \boldsymbol{\eta}_{2_{B_k}} = \begin{bmatrix}\phi & \theta & \psi\end{bmatrix}^T \\ \boldsymbol{w}_{\eta_2} &= \boldsymbol{N}(0, \boldsymbol{R}_{AHRS}) \end{aligned}\right\} \Rightarrow \boldsymbol{\eta}_i = \begin{bmatrix}\boldsymbol{\eta}_{1_i} \\ \boldsymbol{\eta}_{2_i}\end{bmatrix} = \underbrace{\begin{bmatrix}\boldsymbol{I} \\ 0\end{bmatrix}}_{\boldsymbol{F}} \boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k) + \underbrace{\begin{bmatrix}0 \\ \boldsymbol{I}\end{bmatrix}}_{\boldsymbol{G}} (\boldsymbol{u}_k + \boldsymbol{w}_{\eta_2})$$

> Adding a new View Pose can be achieved by:

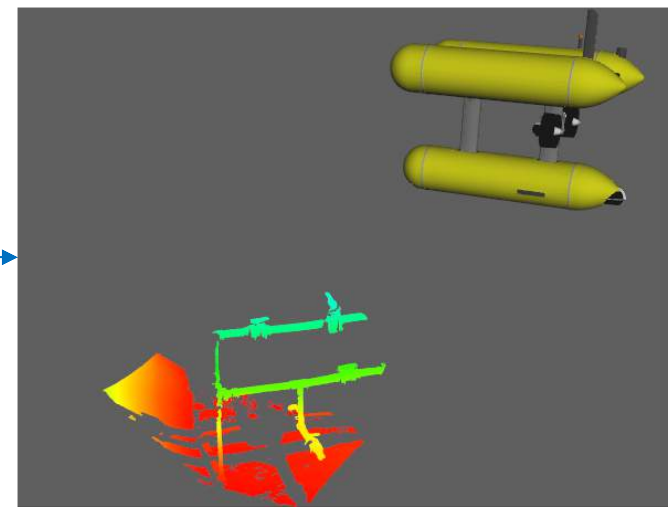
$$\bar{\boldsymbol{x}}_k^+ = \begin{bmatrix}\boldsymbol{F}\boldsymbol{f}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_k, \boldsymbol{w}_k) + \boldsymbol{G}(\boldsymbol{u}_k + \boldsymbol{w}_{\eta_2}) \\ \boldsymbol{x}_k\end{bmatrix}; \ \boldsymbol{F} = \begin{bmatrix}\boldsymbol{I} \\ 0\end{bmatrix}; \ \boldsymbol{G} = \begin{bmatrix}0 \\ \boldsymbol{I}\end{bmatrix}$$

Pose based EKF SLAM using Point Clouds

Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner

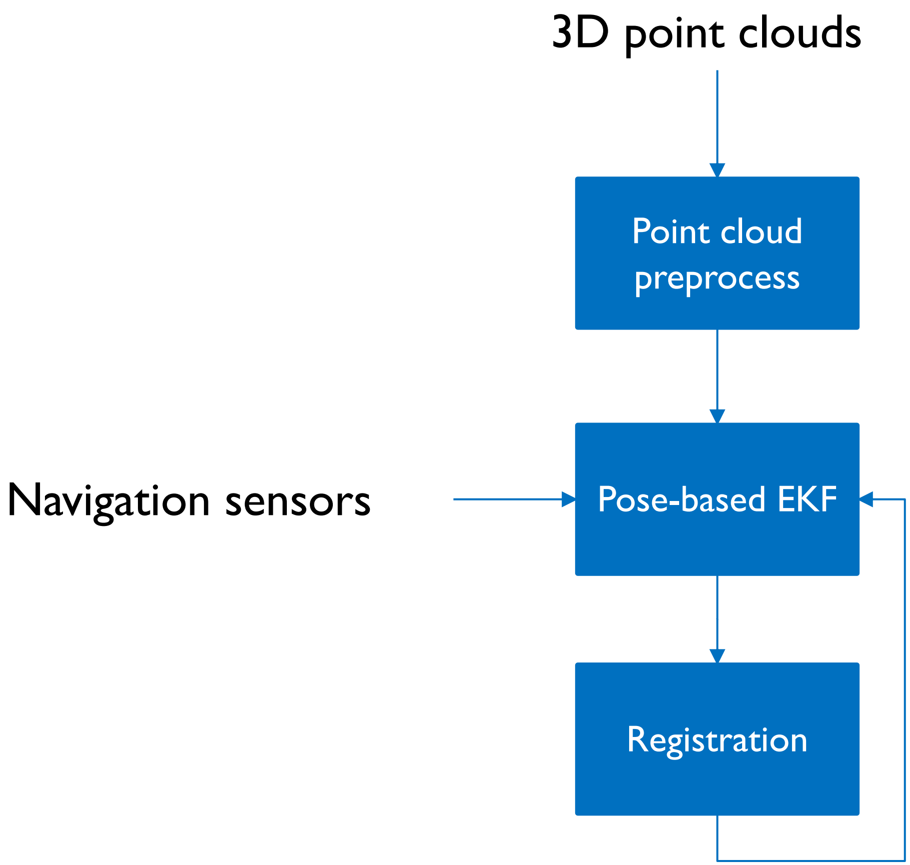


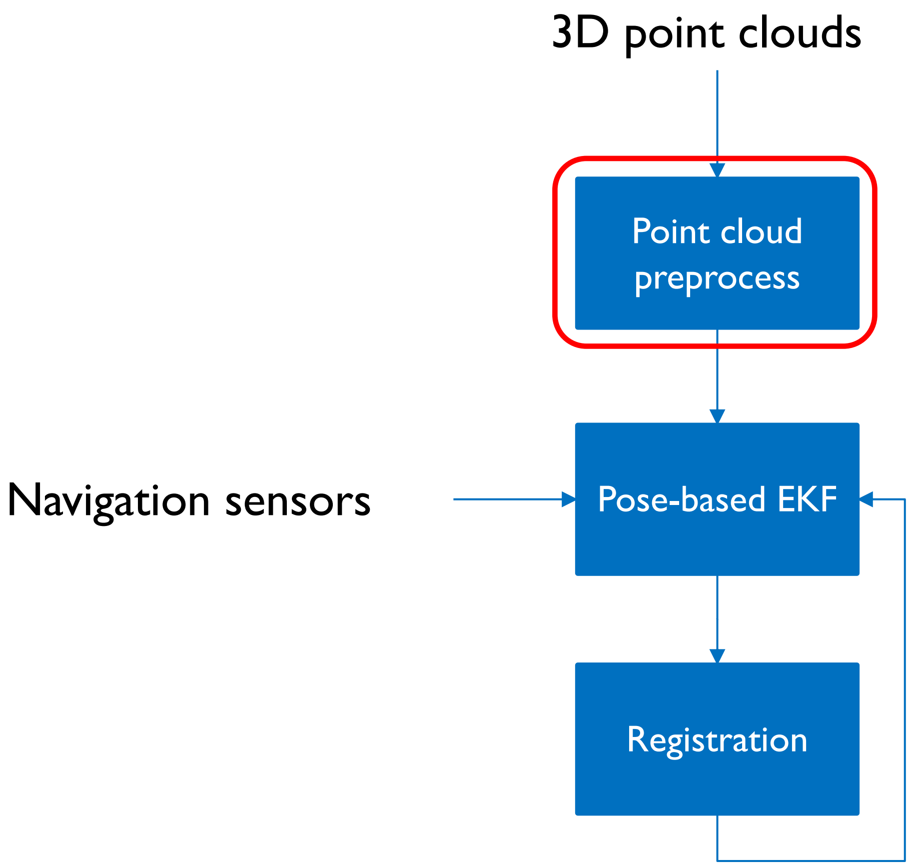
GetScan()



Pose based EKF SLAM using Point Clouds

Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner

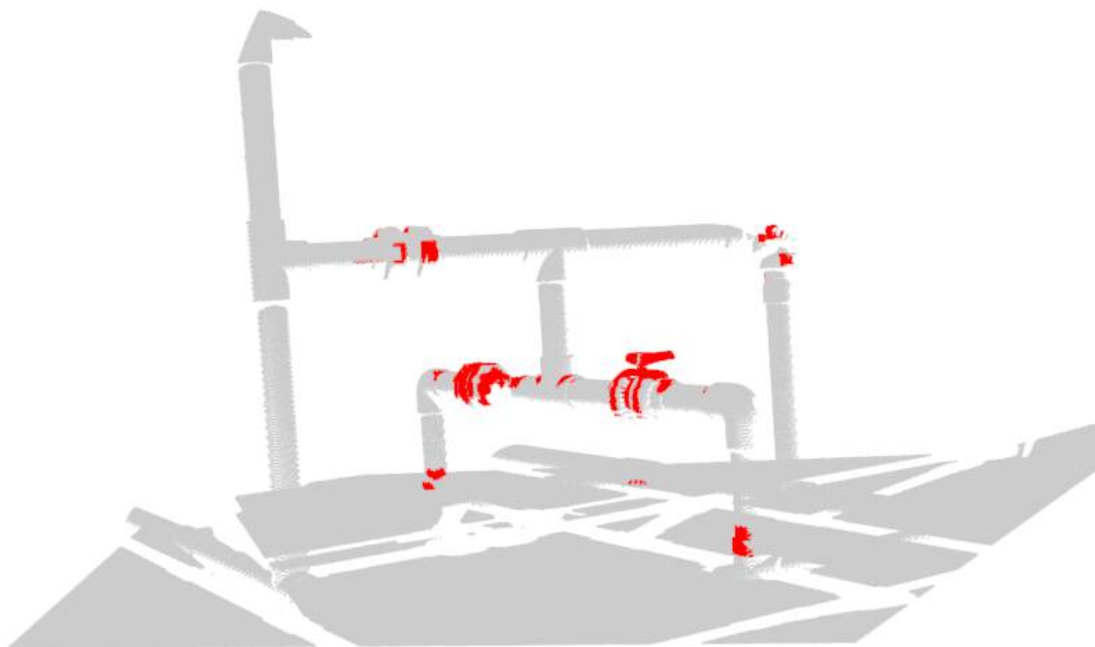




3D point clouds
Point cloud preprocess
Navigation sensors
Pose-based EKF
Registration

Point cloud preprocess

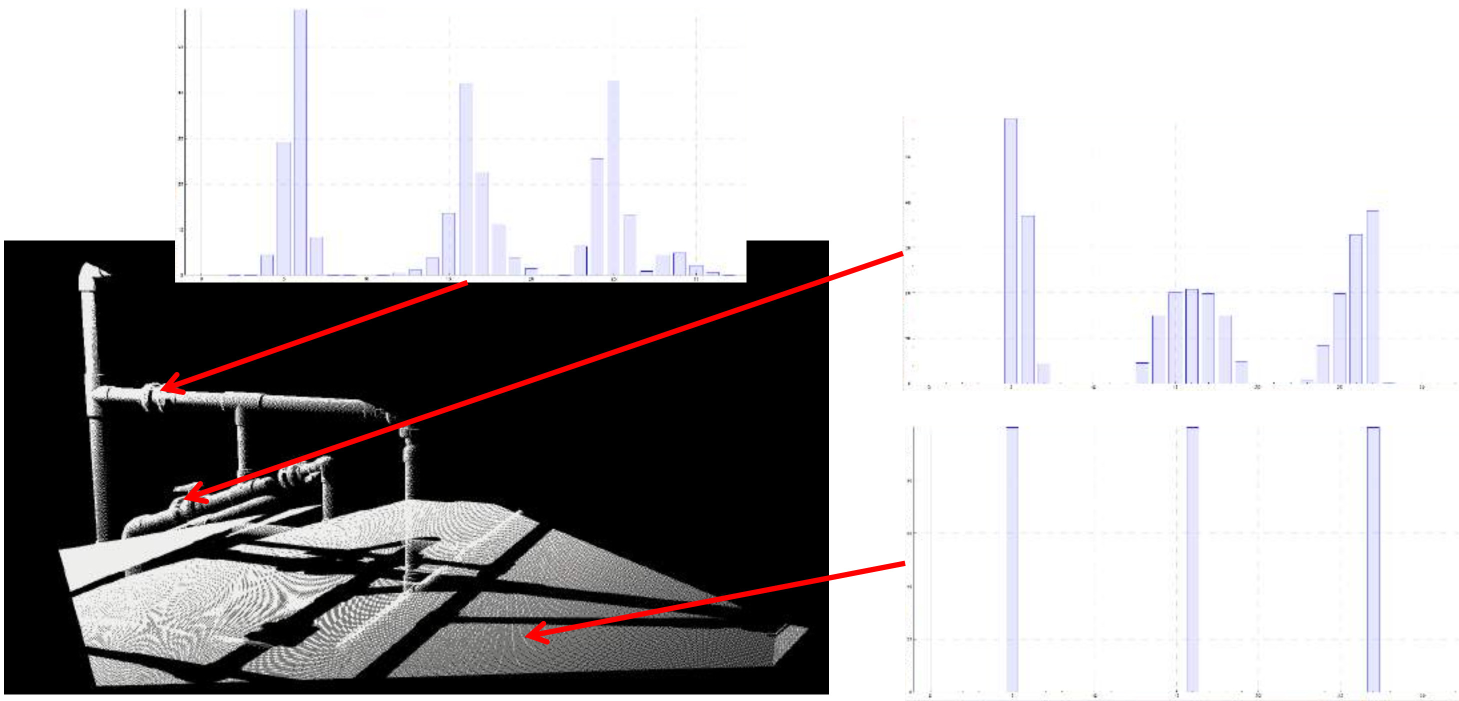
- Key points extraction
  - Remove planar surfaces (RANSAC, Fischler et al. 1981)
  - Remove points with curvature (Pauly et al., 2002) below threshold



Point cloud preprocess

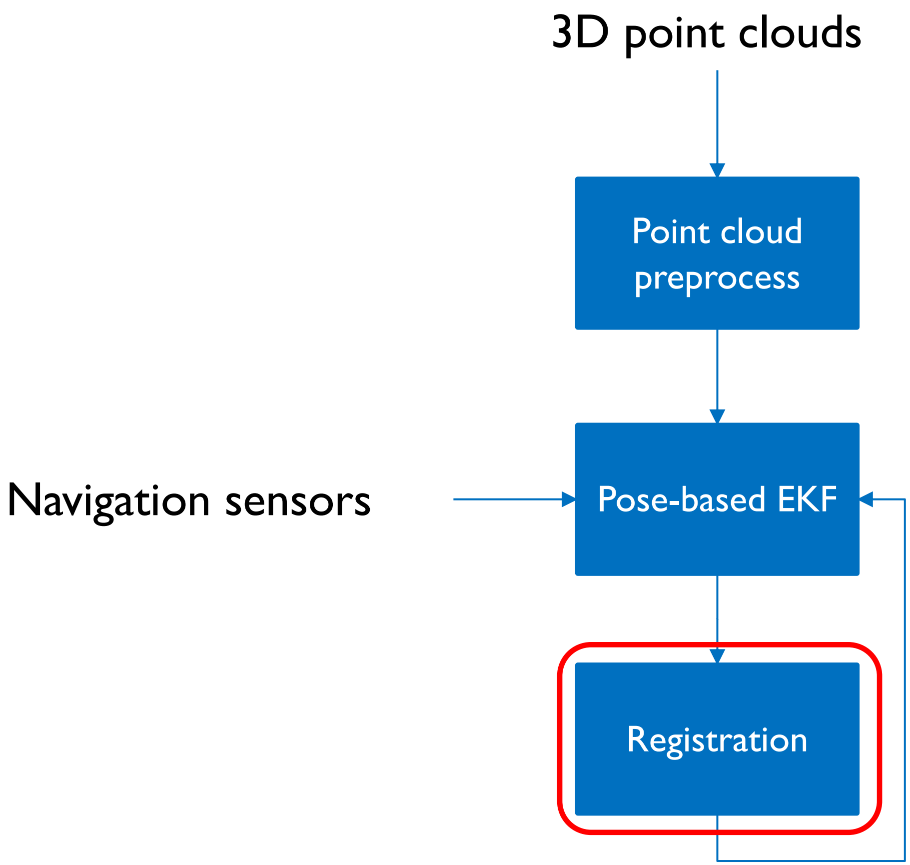
- Key points extraction
  - Remove planar surfaces (RANSAC, Fischler et al. 1981)
  - Remove points with curvature (Pauly et al., 2002) below threshold

- Feature extraction: Fast point feature histogram (Rusu et al., 2009)



Pose based EKF SLAM using Point Clouds

Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner

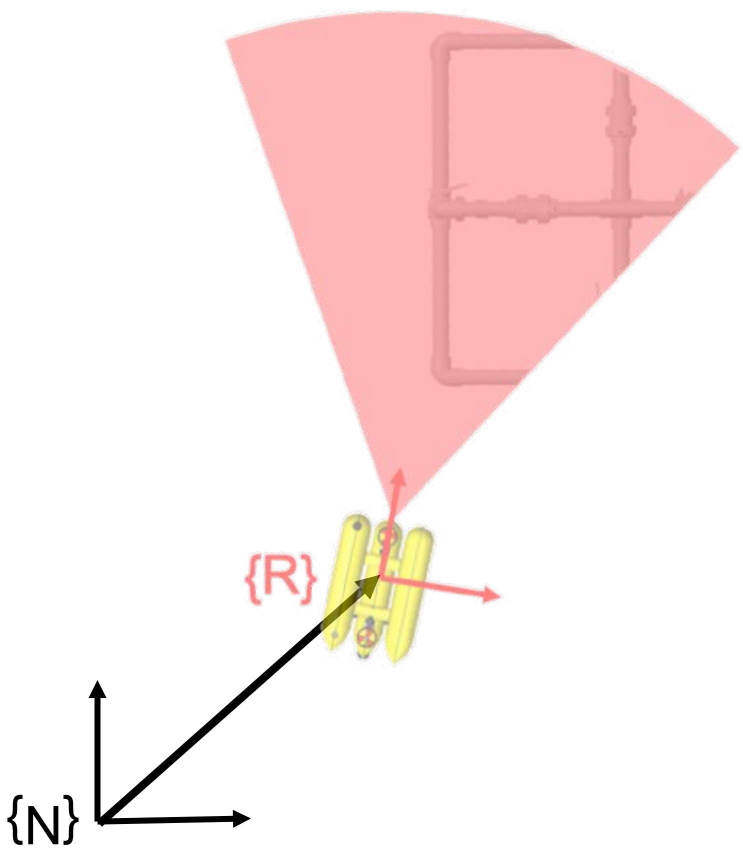


EKF Pose-Based SLAM

> The AUV takes 1 scans and adds the View-Pose to the map

$$\mathcal{M} = [{}^{R}\boldsymbol{S}_{\boldsymbol{R}}]$$

$${}^{N}\hat{\boldsymbol{x}}_{\boldsymbol{k}} = \begin{bmatrix} {}^{N}\hat{\boldsymbol{x}}_{\boldsymbol{R}} \\ {}^{N}\hat{\hat{\boldsymbol{x}}}_{\boldsymbol{B_k}} \end{bmatrix}$$

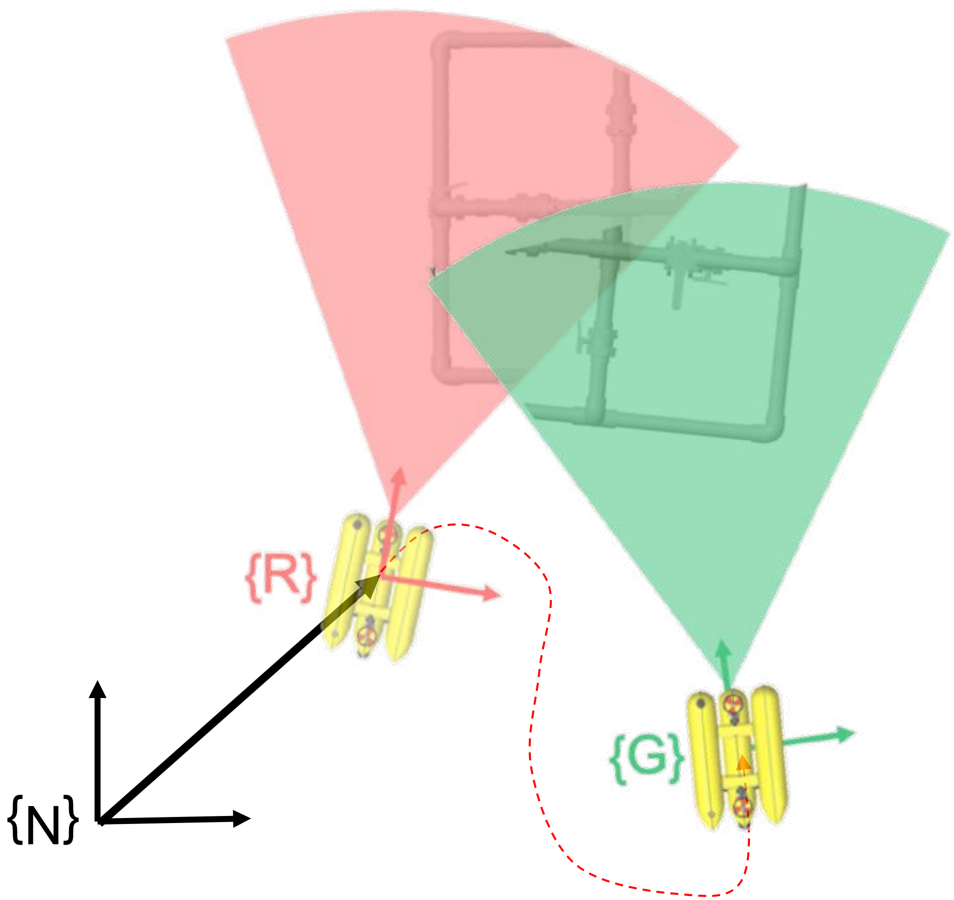


EKF Pose-Based SLAM

> The AUV moves and take another scans from the new View pose

$$\mathcal{M} = [{}^R\boldsymbol{S}_R,\ {}^G\boldsymbol{S}_G]$$

$${}^N\hat{\boldsymbol{x}}_k = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_R \\ {}^N\hat{\boldsymbol{x}}_G \\ {}^N\hat{\boldsymbol{x}}_{B_k} \end{bmatrix}$$



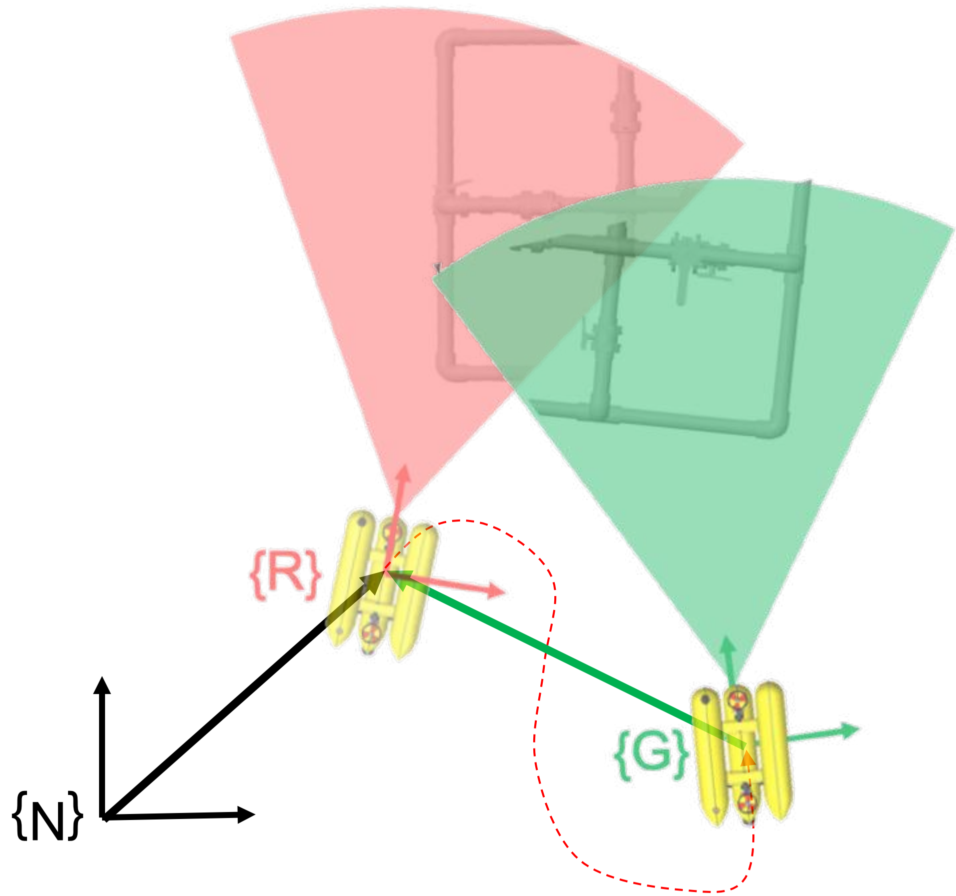
EKF Pose-Based SLAM

> The Observation function computes the expected robot displacement between scans.

$$\mathcal{M} = [{}^{R}S_{R},\ {}^{G}S_{G}]$$

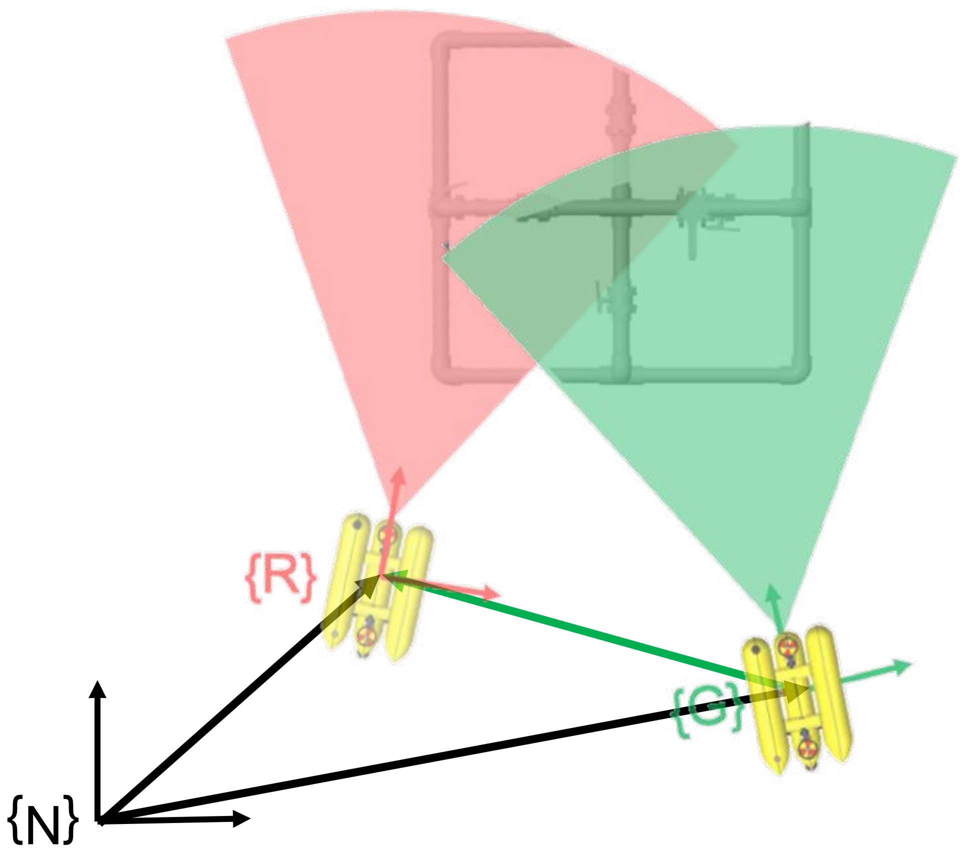
$${}^{N}\hat{x}_k = \begin{bmatrix} {}^{N}\hat{x}_R \\ {}^{N}\hat{x}_G \\ {}^{N}\hat{x}_{B_k} \end{bmatrix}$$

$$z_{RG} = h_{RG}({}^{N}x_k, v_k) = \ominus {}^{N}x_R \oplus {}^{N}x_G + v_{RG}$$



EKF Pose-Based SLAM

> Scans are registered to get the robot displacement & the PEKFSLAM is updated



$$\mathcal{M} = [{}^R S_R, {}^G S_G]$$

$$^N\hat{x}_k = \begin{bmatrix} ^N\hat{x}_R \\ ^N\hat{x}_G \\ ^N\hat{x}_{B_k} \end{bmatrix}$$

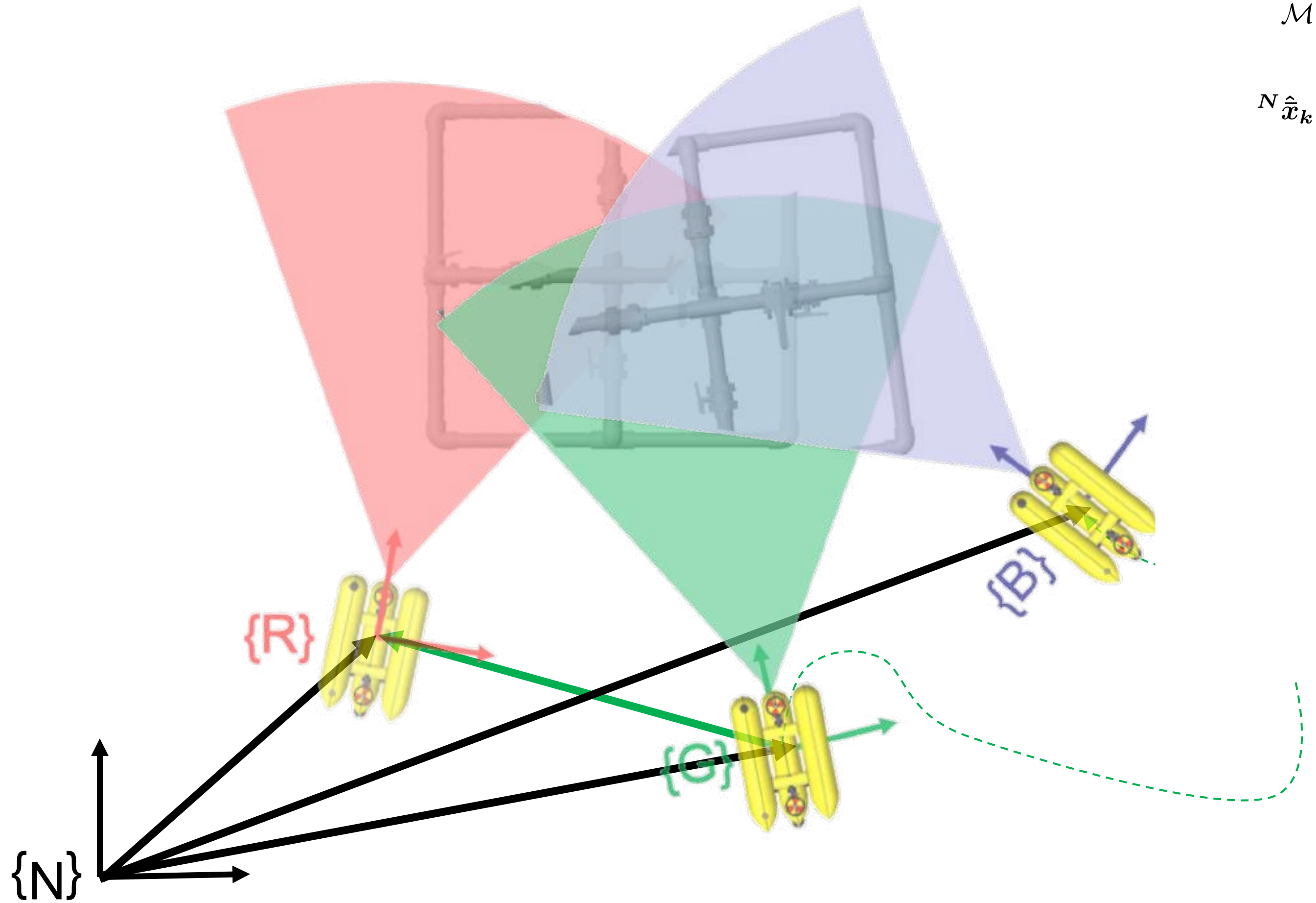
$$z_{RG} = h_{RG}(^N x_k, v_k) = \ominus {}^N x_R \oplus {}^N x_G + v_{RG}$$

$$[\boldsymbol{z_{RG}}, \boldsymbol{R_{RG}}] = Register({}^R S_R, {}^G S_G)$$

$$\boldsymbol{^N\hat{x}_k} = \begin{bmatrix} \boldsymbol{^N\hat{x}_G} \\ \boldsymbol{^N\hat{x}_R} \\ \boldsymbol{^N\hat{x}_{B_k}} \end{bmatrix} = Update(\cdot)$$

EKF Pose-Based SLAM

> The AUV moves and take another scans from the new View pose

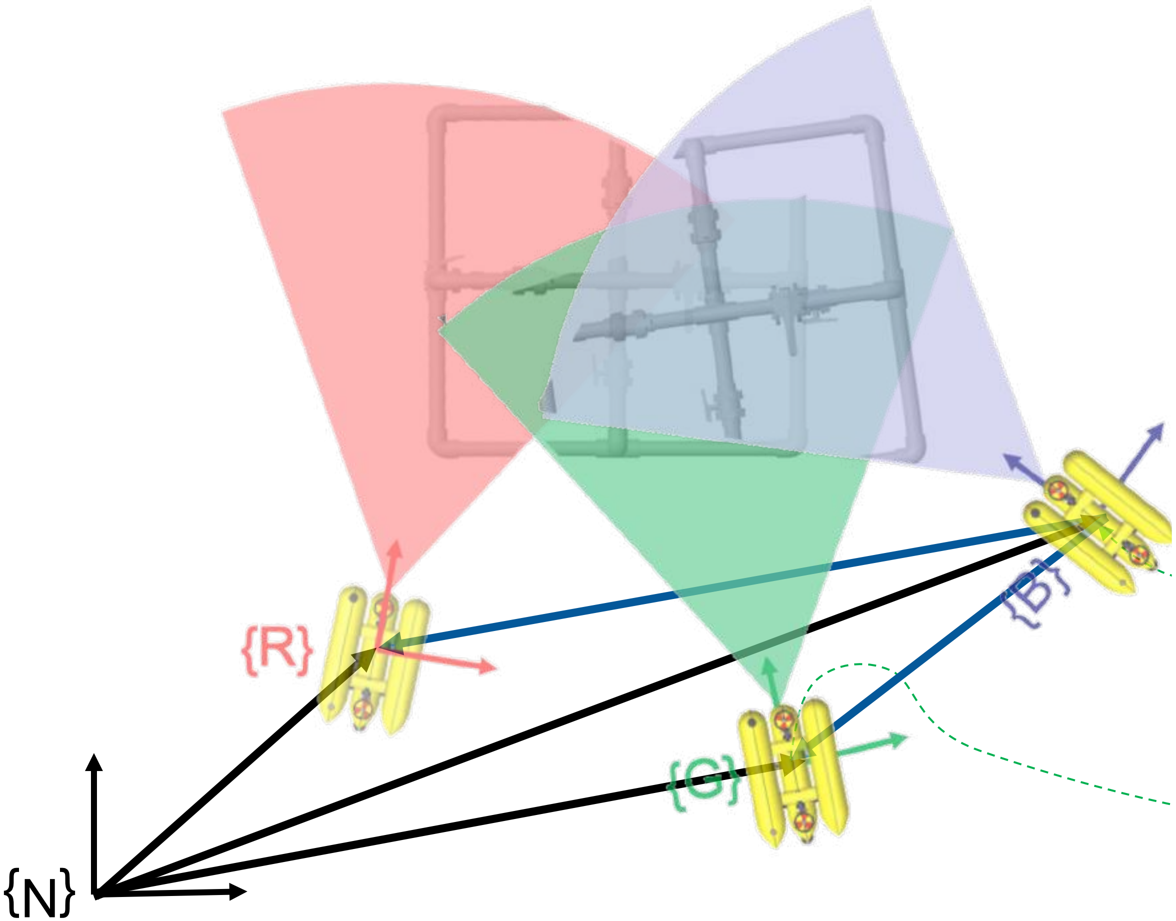


$$\mathcal{M} = [{}^{R}\boldsymbol{S}_{\boldsymbol{R}},\ {}^{G}\boldsymbol{S}_{\boldsymbol{G}},\ {}^{B}\boldsymbol{S}_{\boldsymbol{B}}]$$

$$^{N}\hat{\bar{\boldsymbol{x}}}_{\boldsymbol{k}} = \begin{bmatrix} ^{N}\hat{\boldsymbol{x}}_{\boldsymbol{R}} \\ ^{N}\hat{\boldsymbol{x}}_{\boldsymbol{G}} \\ ^{N}\hat{\boldsymbol{x}}_{\boldsymbol{B}} \\ ^{N}\hat{\bar{\boldsymbol{x}}}_{\boldsymbol{B_k}} \end{bmatrix}$$

EKF Pose-Based SLAM

> The Observation function computes the expected robot displacement between scans.



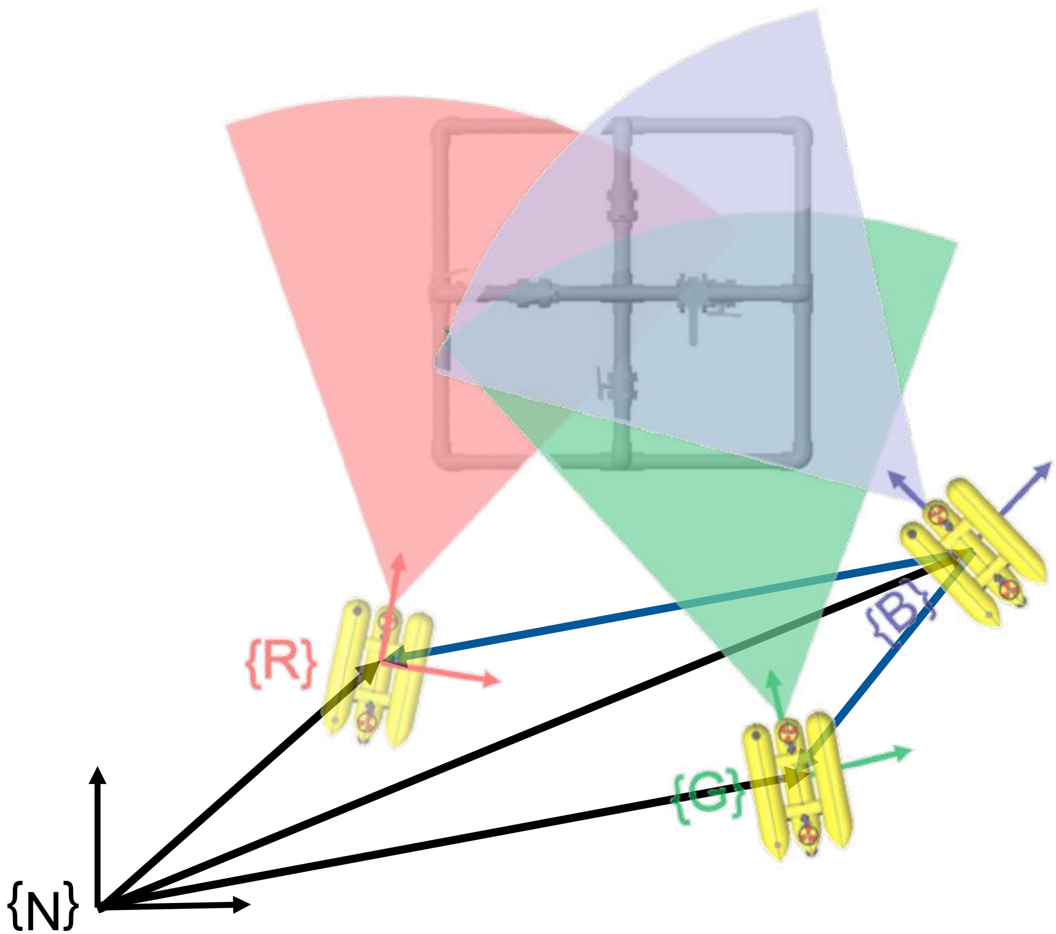
$$\mathcal{M} = [{}^{R}S_{R},\ {}^{G}S_{G},\ {}^{B}S_{B}]$$

$${}^{N}\hat{x}_k = \begin{bmatrix} {}^{N}\hat{x}_R \\ {}^{N}\hat{x}_G \\ {}^{N}\hat{x}_B \\ {}^{N}\hat{x}_{B_k} \end{bmatrix}$$

$$z_k = \begin{bmatrix} h_{BG}({}^{N}x_k, v_k) \\ h_{BR}({}^{N}x_k, v_k) \end{bmatrix}$$

EKF Pose-Based SLAM

> Scans are registered to get the robot displacement & the PEKFSLAM is updated



$$\mathcal{M} = [{}^R S_R,\ {}^G S_G,\ {}^B S_B]$$

$${}^N\hat{x}_k = \begin{bmatrix} {}^N\hat{x}_R \\ {}^N\hat{x}_G \\ {}^N\hat{x}_B \\ {}^N\hat{x}_{B_k} \end{bmatrix}$$

$$z_k = \begin{bmatrix} h_{BG}({}^N x_k, v_k) \\ h_{BR}({}^N x_k, v_k) \end{bmatrix} = \begin{bmatrix} \ominus {}^N x_B \oplus {}^N x_G + v_{BG} \\ \ominus {}^N x_B \oplus {}^N x_R + v_{BR} \end{bmatrix}$$

$$[\boldsymbol{z_{BG}}, \boldsymbol{R_{BG}}] = Register({}^B S_B, {}^G S_G)$$

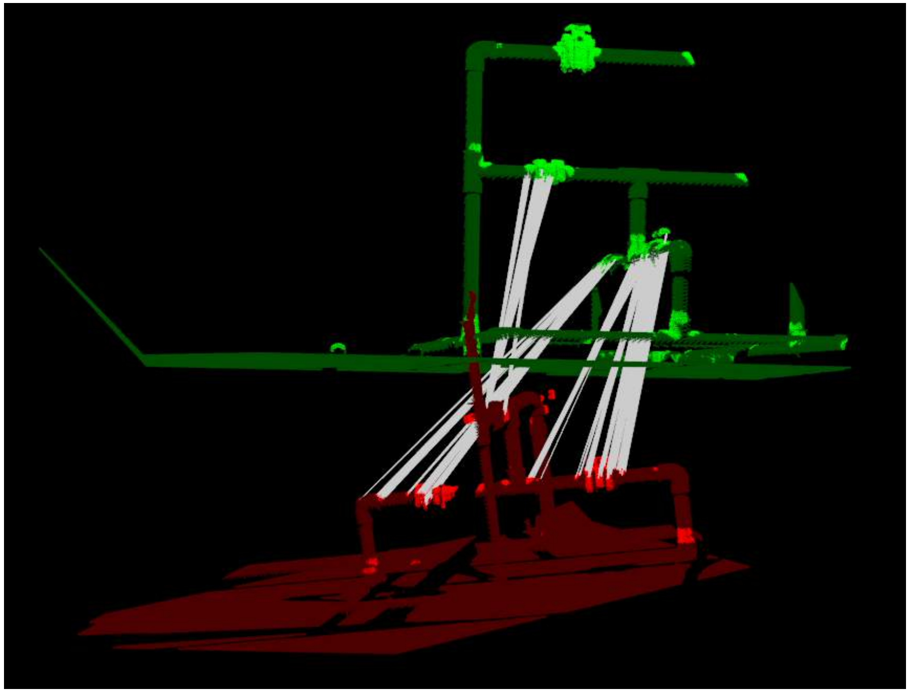
$$[\boldsymbol{z_{BR}}, \boldsymbol{R_{BR}}] = Register({}^B S_B, {}^R S_R)$$

$$\boldsymbol{z_k} = \begin{bmatrix} \boldsymbol{z_{BG}} \\ \boldsymbol{z_{BR}} \end{bmatrix} \ ; \ \boldsymbol{R_k} = \begin{bmatrix} \boldsymbol{R_{BG}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R_{BR}} \end{bmatrix}$$

$${}^N\hat{\boldsymbol{x}}_{\boldsymbol{k}} = \begin{bmatrix} {}^N\hat{\boldsymbol{x}}_{\boldsymbol{G}} \\ {}^N\hat{\boldsymbol{x}}_{\boldsymbol{R}} \\ {}^N\hat{\boldsymbol{x}}_{\boldsymbol{B}} \\ {}^N\hat{\boldsymbol{x}}_{\boldsymbol{B_k}} \end{bmatrix} = Update(\ldots, \boldsymbol{z_k}, \boldsymbol{R_k})$$

Registration algorithm

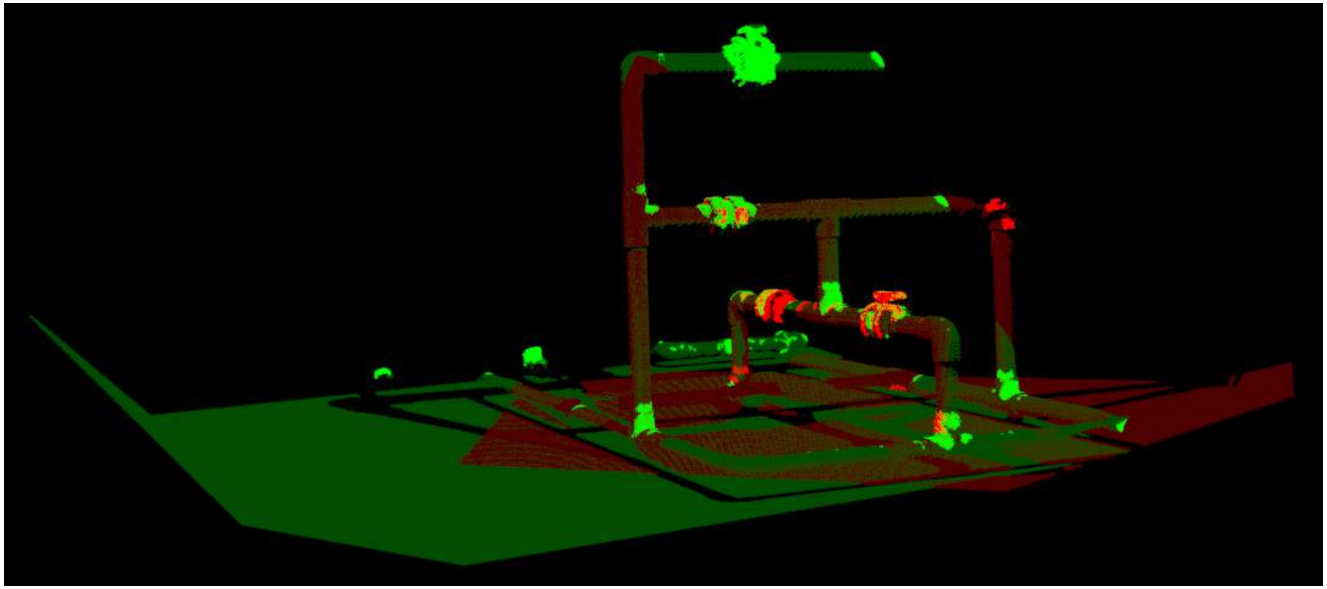
- Coarse registration
  - Feature association
  - Roto-translation using Singular Value Decomposition (J. Besl and McKay, 1992)



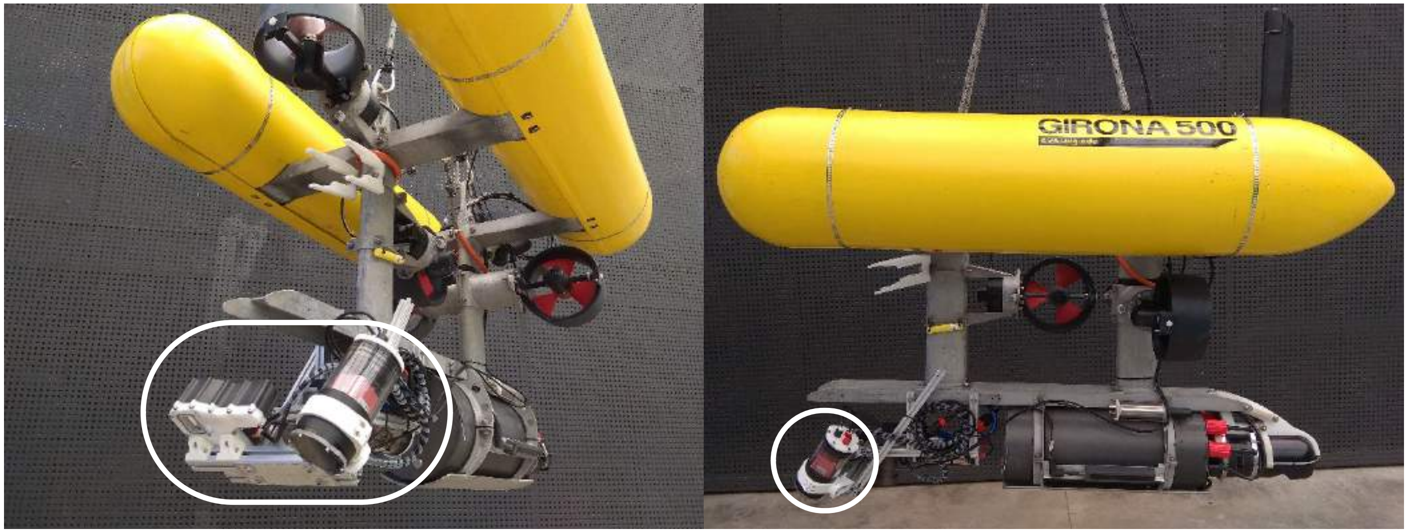
Registration algorithm

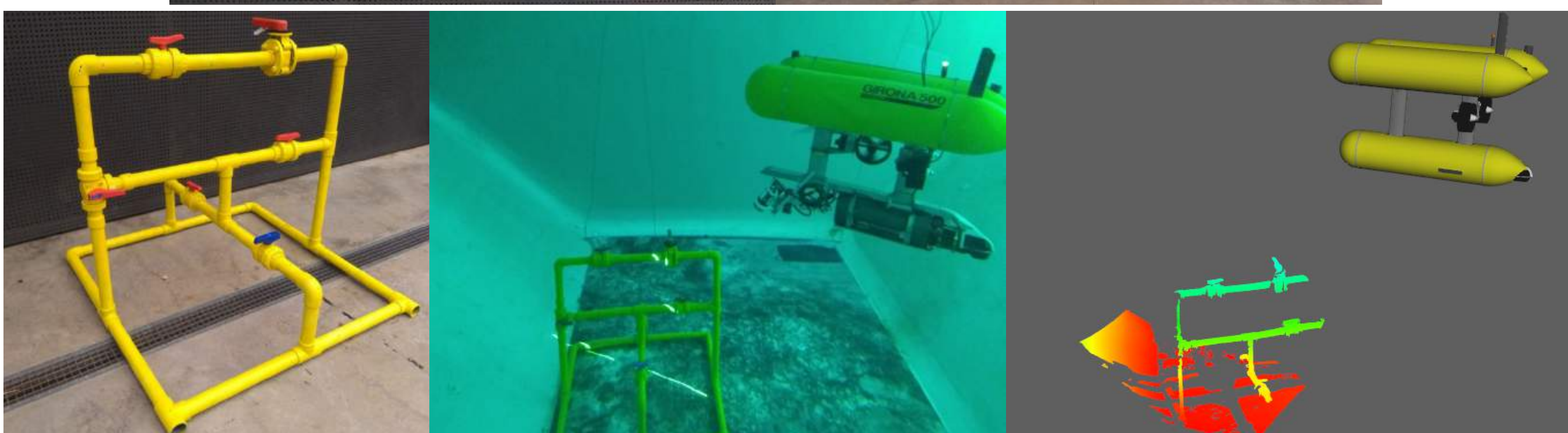
- Coarse registration
  - Feature association
  - Roto-translation using Singular Value Decomposition (J. Besl and McKay, 1992)

- Fine registration: Point to point ICP



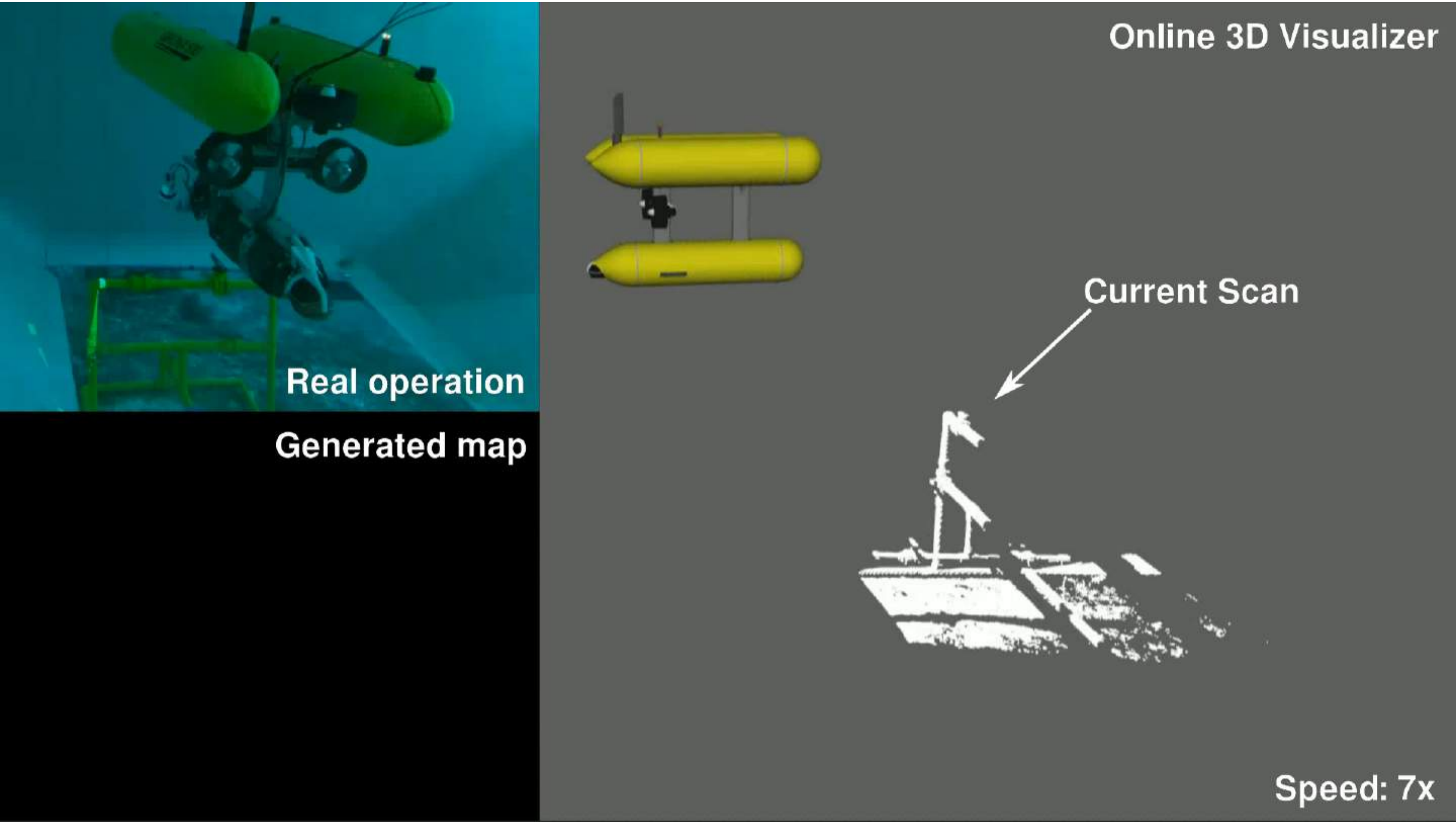
Experiments and results





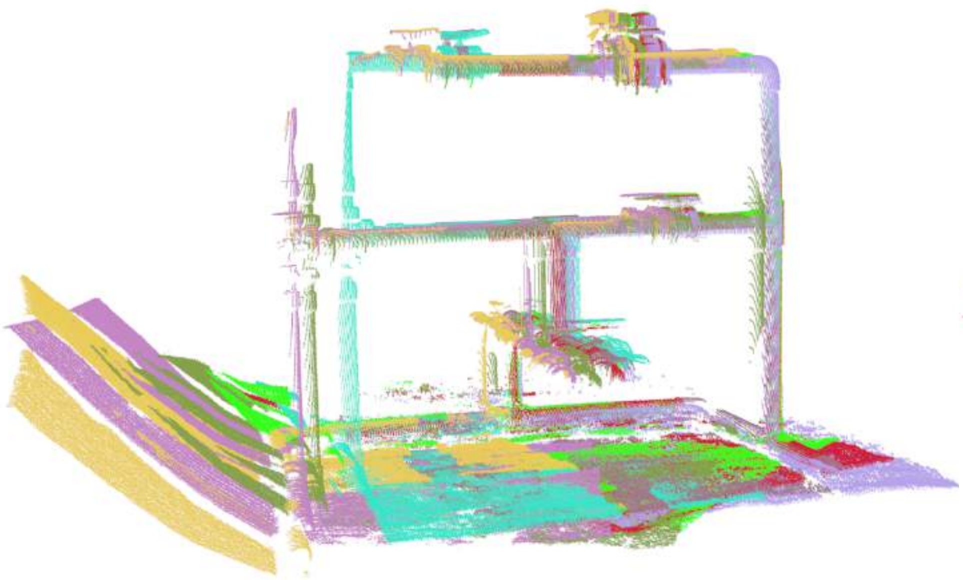
Pose based EKF SLAM using Point Clouds

Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner

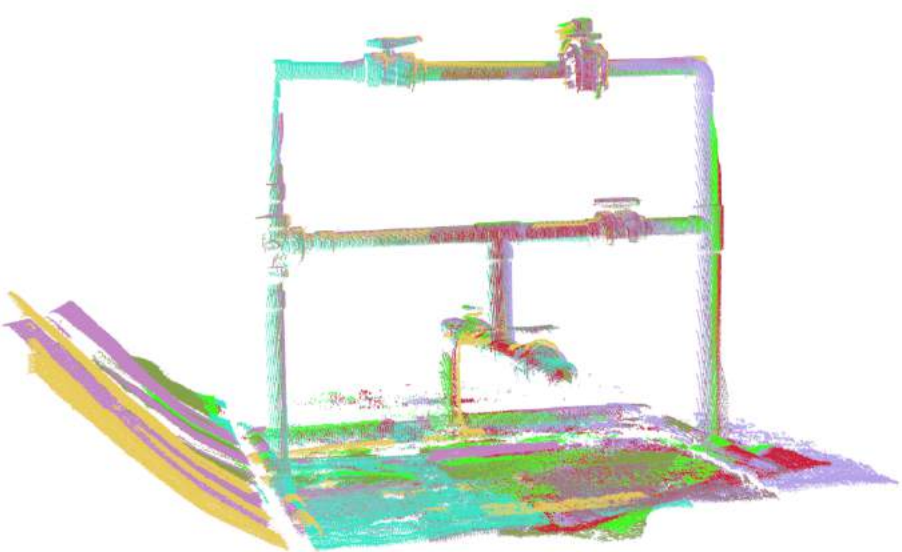


Experiments and results

Dead reckoning

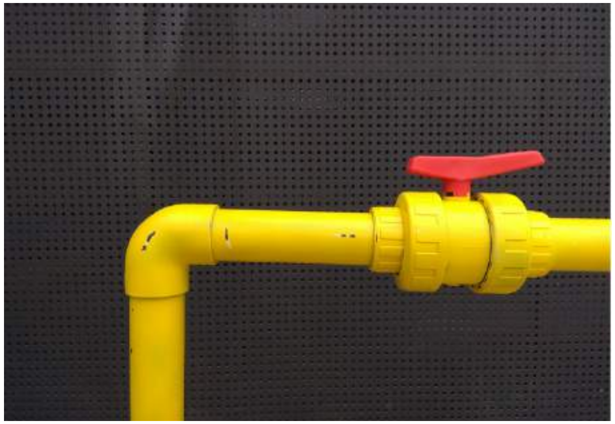
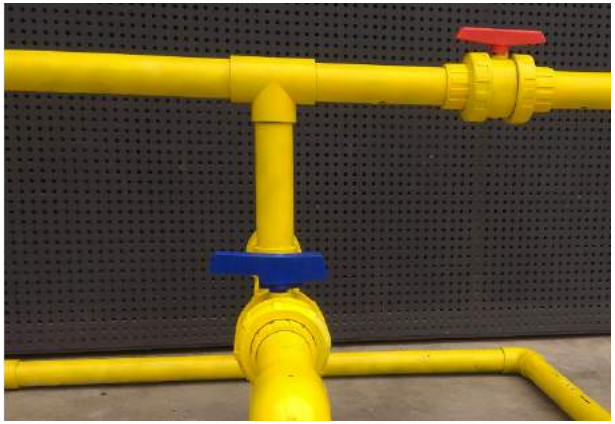


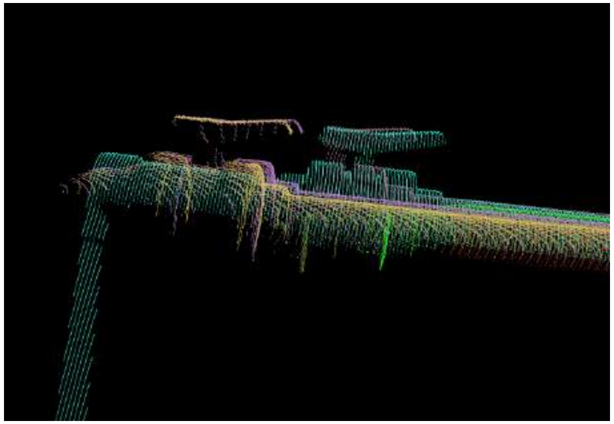
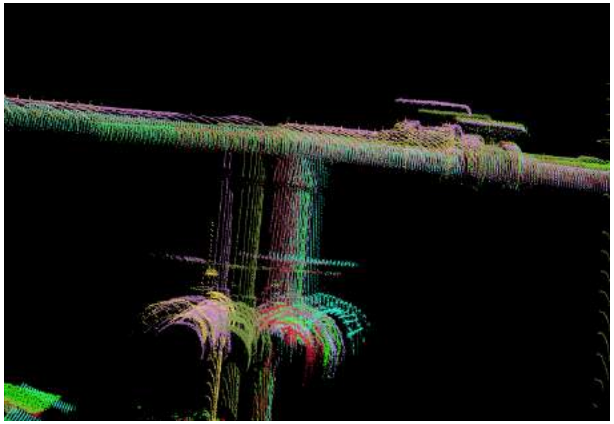
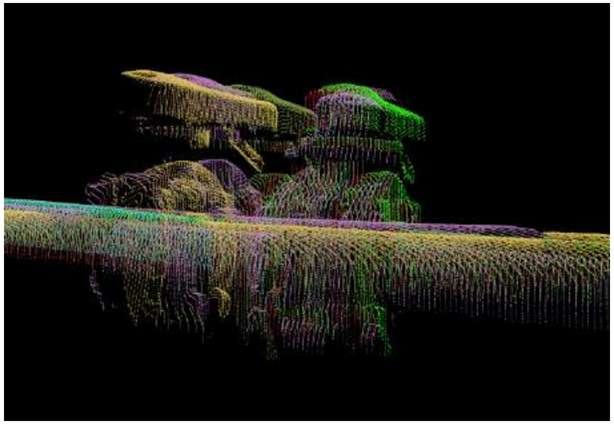
SLAM



Experiments and results

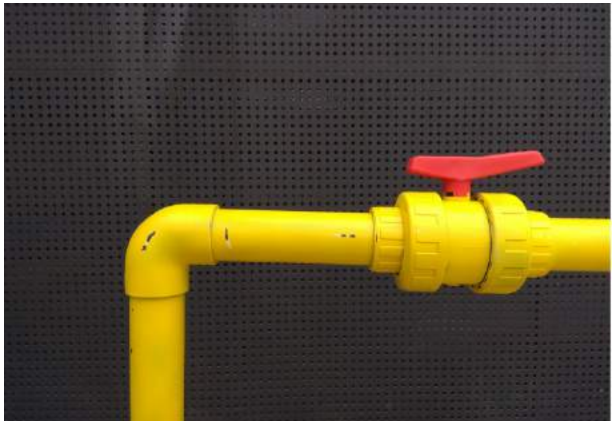
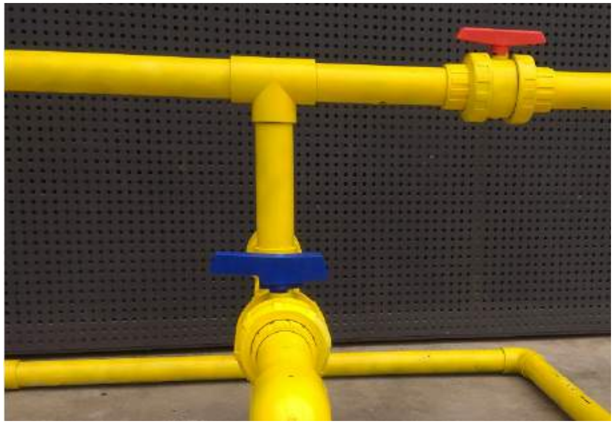
Structure details: dead reckoning navigation

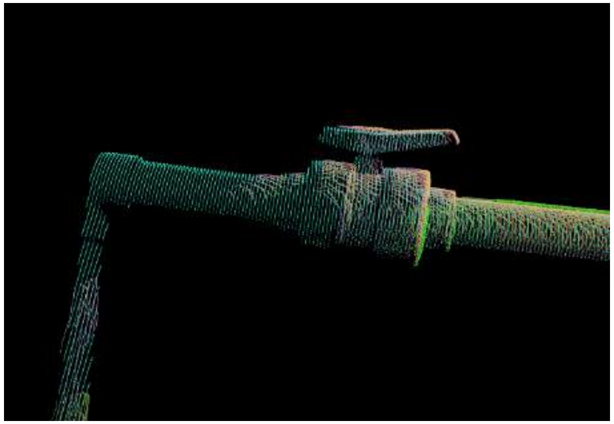
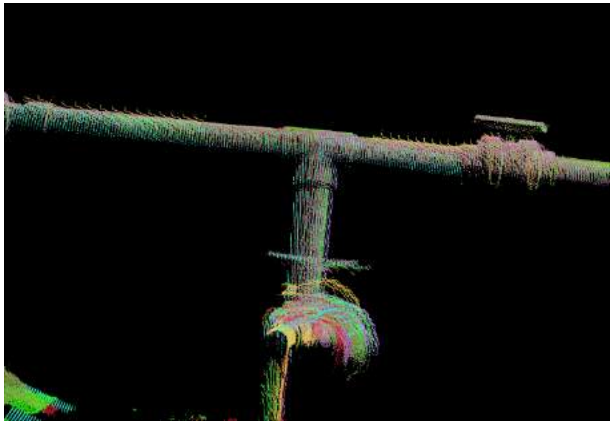
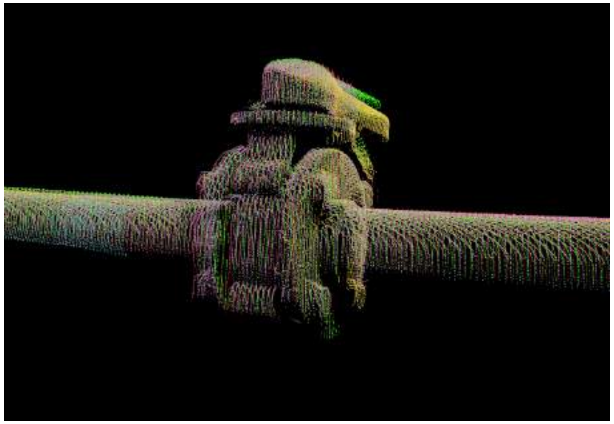



Experiments and results

Structure details: SLAM

Experiments and results

Robot uncertainty