Supplementary Materials: WING: Wheel-Inertial Neural Odometry with Ground Manifold Constraints

Kunyi Zhang[1,3] - kunyizhang@zju.edu.cn
Chenxing Jiang[2,3,4] - chenxingjiang@zju.edu.cn

CONTENTS

[1]State Key Laboratory of Industrial Control Technology, Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China.
[2]Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology.
[3]Alibaba DAMO Academy Autonomous Driving Lab, Hangzhou 311121, China.
[4]Huzhou Institute, Zhejiang University, Huzhou 313000, China.

I. Preliminaries

A. IMU Kinematics

IMU measurements include the gyroscope $\widetilde{\boldsymbol{\omega}}$ and non-gravitational acceleration $\widetilde{\boldsymbol{a}}$, which are measured in the IMU frame (the $\mathcal{I}$ frame) and given by:

$$
\begin{aligned}
{}^{\mathcal{I}}\widetilde{\boldsymbol{\omega}} &= {}^{\mathcal{I}}\boldsymbol{\omega} + {}^{\mathcal{I}}\boldsymbol{b}_{\boldsymbol{\omega}} + \boldsymbol{n}_{\boldsymbol{\omega}}, && \text{(1a)}\\
{}^{\mathcal{I}}\widetilde{\boldsymbol{a}} &= {}^{\mathcal{I}}\boldsymbol{a} + {}^{\mathcal{I}}\boldsymbol{b}_{\boldsymbol{a}} + {}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}^{\top}{}^{\mathcal{G}}\mathbf{g} + \boldsymbol{n}_{\boldsymbol{a}}, && \text{(1b)}
\end{aligned}
$$

where ${}^{\mathcal{I}}\boldsymbol{\omega}$ and ${}^{\mathcal{I}}\boldsymbol{a}$ are the true angular velocity and acceleration, ${}^{\mathcal{G}}\mathbf{g} = [0, 0, 9.8]$ is the gravity vector in the gravity-aligned frame (the $\mathcal{G}$ frame), ${}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}$ is the rotation matrix from the $\mathcal{I}$ frame to the $\mathcal{G}$ frame, $\boldsymbol{n}_{\boldsymbol{\omega}}$ and $\boldsymbol{n}_{\boldsymbol{a}}$ are the additive Gaussian white noise in gyroscope and acceleration measurements, $\boldsymbol{b}_{\boldsymbol{\omega}}$ and $\boldsymbol{b}_{\boldsymbol{a}}$ are the bias of IMU modeled as random walk:

$$
\begin{aligned}
\boldsymbol{n}_{\boldsymbol{\omega}} &\sim \mathcal{N}(0, \boldsymbol{\Sigma}^2_{\boldsymbol{\omega}}), \quad \dot{\boldsymbol{b}}_{\boldsymbol{\omega}} = \boldsymbol{n}_{\boldsymbol{b}_{\boldsymbol{a}}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}^2_{\boldsymbol{b}_{\boldsymbol{\omega}}}), && \text{(2a)}\\
\boldsymbol{n}_{\boldsymbol{a}} &\sim \mathcal{N}(0, \boldsymbol{\Sigma}^2_{\boldsymbol{a}}), \quad \dot{\boldsymbol{b}}_{\boldsymbol{a}} = \boldsymbol{n}_{\boldsymbol{b}_{\boldsymbol{\omega}}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}^2_{\boldsymbol{b}_{\boldsymbol{a}}}). && \text{(2b)}
\end{aligned}
$$

B. Wheel Odometer Kinematics

In this paper, we consider a two-wheel ground robot model, which means that the forward velocity ${}^{\mathcal{B}}\widetilde{v}_x$ and angular velocity ${}^{\mathcal{B}}\widetilde{\omega}_z$ perpendicular to the motion plane can be derived from two wheel speeds:

$$
\begin{aligned}
{}^{\mathcal{B}}\widetilde{v}_x &= \frac{\widetilde{\omega}_l * r_l + \widetilde{\omega}_r * r_r}{2}, && \text{(3a)}\\
{}^{\mathcal{B}}\widetilde{\omega}_z &= \frac{\widetilde{\omega}_r * r_r - \widetilde{\omega}_l * r_l}{w_b}, && \text{(3b)}
\end{aligned}
$$

where $\widetilde{\omega}_l$ and $\widetilde{\omega}_r$ are speeds of the left and right wheels, r_l and r_r are the radii of two wheels, and w_b is the wheelbase between two wheels.

C. Yaw Independent Rotation Convention

A novel rotation convention is proposed in [1], which decouples the rotation into yaw angle and tilt vector. Thus, the estimations of velocity and tilt could avoid being affected by the drift of yaw angle when updating by body velocity. The rotation is represented as follows:

$$
\begin{aligned}
\mathbf{R} &= \mathbf{R}_{\psi}\mathbf{R}_{\phi}, && \text{(4)}\\
\mathbf{R}_{\psi} &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, && \text{(4a)}\\
\mathbf{R}_{\phi} &= \begin{bmatrix} \frac{z_2^2 + z_1^2 z_3}{z_1^2 + z_2^2} & \frac{z_1 z_2 (z_3 - 1)}{z_1^2 + z_2^2} & z_1 \\ \frac{z_1 z_2 (z_3 - 1)}{z_1^2 + z_2^2} & \frac{z_1^2 + z_2^2 z_3}{z_1^2 + z_2^2} & z_2 \\ -z_1 & -z_2 & z_3 \end{bmatrix}, && \text{(4b)}
\end{aligned}
$$

where ψ is the yaw angle, $\boldsymbol{z} = [z_1, z_2, z_3]^{\top}$ is the tilt vector in the 2-sphere S^2 ($\|\boldsymbol{z}\| = 1$). Differentiating the tilt matrix $\mathbf{R}_{\phi} = \mathbf{R}_{\psi}^{\top}\mathbf{R}$ yields:

$$
\begin{aligned}
\dot{\mathbf{R}}_{\phi} &= \dot{\mathbf{R}}_{\psi}^{\top}\mathbf{R} + \mathbf{R}_{\psi}^{\top}\dot{\mathbf{R}} \\
&= -[\dot{\psi}\boldsymbol{e}_3]_{\times}\mathbf{R}_{\psi}^{\top}\mathbf{R} + \mathbf{R}_{\psi}^{\top}\mathbf{R}[\boldsymbol{\omega}]_{\times} \\
&= -[\dot{\psi}\boldsymbol{e}_3]_{\times}\mathbf{R}_{\phi} + \mathbf{R}_{\phi}[\boldsymbol{\omega}]_{\times},
\end{aligned} \qquad \text{(5)}
$$

where $\boldsymbol{e}_3 = [0, 0, 1]^\top$ and $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^\top$. Differentiating the last column and last row of tilt matrix $\mathbf{R}_\phi$ yields:

$$\dot{z}_1 = \dot{\psi} z_2 + \omega_2 \frac{z_2^2 + z_1^2 z_3}{z_1^2 + z_2^2} - \omega_1 \frac{z_1 z_2 (z_3 - 1)}{z_1^2 + z_2^2}, \tag{6a}$$

$$\dot{z}_2 = -\dot{\psi} z_1 - \omega_1 \frac{z_1^2 + z_2^2 z_3}{z_1^2 + z_2^2} + \omega_2 \frac{z_1 z_2 (z_3 - 1)}{z_1^2 + z_2^2}, \tag{6b}$$

$$\dot{z}_3 = -\omega_2 z_1 + \omega_1 z_2, \tag{6c}$$

$$-\dot{z}_1 = -\omega_3 z_2 - \omega_2 z_3, \tag{6d}$$

$$-\dot{z}_2 = \omega_3 z_1 + \omega_1 z_3. \tag{6e}$$

Bringing Eq.(6d) into Eq.(6a) or Eq.(6e) into Eq.(6b) and simplifying gives us:

$$\dot{\psi} = \omega_3 - \frac{\omega_1 z_1 + \omega_2 z_2}{1 + z_3}. \tag{7}$$

There is an alternative representation of the tilt vector based on the stereographic coordinates $\boldsymbol{s} = (s_1, s_2)^\top$, which are used to give a minimal parameterization of the tilt. The stereographic coordinates are expressed as the functions of $\boldsymbol{z}$:

$$s_1 = \frac{z_1}{1 + z_3}, \tag{8a}$$

$$s_2 = \frac{z_2}{1 + z_3}. \tag{8b}$$

Differentiating Eq.(8b) and simplifying yields:

$$\dot{s}_1 = \frac{1}{2}\omega_2 (s_1^2 - s_2^2 + 1) + \omega_3 s_2 - \omega_1 s_1 s_2, \tag{9a}$$

$$\dot{s}_2 = \frac{1}{2}\omega_1 (s_1^2 - s_2^2 - 1) - \omega_3 s_1 + \omega_2 s_1 s_2. \tag{9b}$$

Similarly, the tilt matrix in Eq.(4b) and the yaw angle derivative in Eq.(7) could be represented by $\boldsymbol{s}$:

$$\mathbf{R}_\phi = \begin{bmatrix} \frac{1 - s_1^2 + s_2^2}{1 + s_1^2 + s_2^2} & \frac{-2 s_1 s_2}{1 + s_1^2 + s_2^2} & \frac{2 s_1}{1 + s_1^2 + s_2^2} \\ \frac{-2 s_1 s_2}{1 + s_1^2 + s_2^2} & \frac{1 + s_1^2 - s_2^2}{1 + s_1^2 + s_2^2} & \frac{2 s_2}{1 + s_1^2 + s_2^2} \\ \frac{-2 s_1}{1 + s_1^2 + s_2^2} & \frac{-2 s_2}{1 + s_1^2 + s_2^2} & \frac{1 - s_1^2 - s_2^2}{1 + s_1^2 + s_2^2} \end{bmatrix}, \tag{10}$$

and

$$\dot{\psi} = -\omega_1 s_1 - \omega_2 s_2 + \omega_3. \tag{11}$$

Note that, the yaw angle ψ is wrapped into $[0, 2\pi)$ in the proposed state estimation system.

D. B-spline Curve and Surface

1) B-spline basis function: Given a partition $X = \{x_i \mid x_i \leq x_{i+1},\ i = 0, 1, \cdots, t\}$ on the parameter x axis, the B-spline basis function is defined by the following recursive equation (namely De Boor's Algorithm):

$$B_{i,0}(x) = \begin{cases} 1, & x_i \leq x \leq x_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{12a}$$

$$B_{i,p}(x) = \frac{x - x_i}{x_{i+p} - x_i} B_{i,p-1}(x) + \frac{x_{i+p+1} - x}{x_{i+p+1} - x_{i+1}} B_{i+1,p-1}(x), \tag{12b}$$

where the B-spline basis function $B_{i,p}(x)$ is a polynomial function of degree p (or order $p+1$, $p \geq 0$) in x.

2) B-spline curve: Given control points $\mathbb{C} = \{C_i \mid 0 \leq i \leq n, i \in \mathbb{Z}\}$ (array in $1D$) and a vector of non-decreasing knots $X = \{x_1, x_2, \cdots, x_t\}$, the B-spline curve of p degree can be defined as:

$$\mathcal{C}(x) = \sum_{i=0}^{n} B_{i,p}(x) C_i, \tag{13}$$

where $\{B_{i,p}(x)\}$ are the B-spline basis functions of degree p, the line $\mathbb{L} = \{C_0 C_1 \cdots C_n\}$ is the control polygon of the B-spline curve, and the numbers of the control point vector and the knot vector have the following relationship:

$$t = n + p + 1. \tag{14}$$

3) B-spline surface: Given control points $\mathbb{C} = \{C_{i,j} \mid 0 \leq i \leq n, 0 \leq j \leq m, i, j \in \mathbb{Z}\}$ (grid in $2D$) and two vectors of non-decreasing knots $X = \{x_1, x_2, \cdots, x_t\}$ and $Y = \{y_1, y_2, \cdots, y_s\}$, the B-spline surface of $p \times q$ degree can be defined as:

$$\mathcal{S}(x, y) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,p}(x) B_{j,q}(y) C_{i,j}, \tag{15}$$

where $\{B_{i,p}(x)\}$ and $\{B_{j,q}(x)\}$ are the B-spline basis functions of degree p and q, the grid $\mathbb{G} = \{C_{i,0} C_{i,1} \cdots C_{i,m}, C_{0,j} C_{1,j} \cdots C_{n,j} |\ 0 \leq i \leq n, 0 \leq j \leq m,\ i, j \in \mathbb{Z}\}$ is the control polygon of the B-spline surface, and the numbers of the control point vector and the knot vector have the following relationship:

$$t = n + p + 1, \tag{16a}$$

$$s = m + q + 1. \tag{16b}$$

4) Uniform cubic B-spline curve: Considering the B-spline curve of degree 3 (namely $p = 3$) with the uniform knot vector:

$$X = \{x_1, x_2, \cdots, x_t \mid x_{i+1} - x_i = \text{const}, 1 \leq i \leq t, i \in \mathbb{Z}\}. \tag{17}$$

We represent the B-spline basis functions by the local parameter $u \in [0, 1]$ instead of the global parameter $x \in [x_i, x_{i+1}]$ on each knot interval by performing the transformation as:

$$x = x(u) = (1 - u) x_i + u x_{i+1}, u \in [0, 1]. \tag{18}$$

Then we can derive the cubic B-spline basis functions as:

$$\begin{cases} B_{0,3}(u) = \dfrac{1}{6}\left(-u^3 + 3u^2 - 3u + 1\right), \\ B_{1,3}(u) = \dfrac{1}{6}\left(3u^3 - 6u^2 + 4\right), \\ B_{2,3}(u) = \dfrac{1}{6}\left(-3u^3 + 3u^2 + 3u + 1\right), \\ B_{3,3}(u) = \dfrac{1}{6} u^3, \end{cases} \quad 0 \leq u = \frac{x - x_i}{x_{i+1} - x_i} \leq 1. \tag{19}$$

The cubic B-spline curve $\mathcal{C}(x) = \sum_{i=0}^{n} B_{i,3}(x) C_i$ can be represented in compact matrix form:

$$\mathcal{C}(u) = \boldsymbol{u} \mathbf{B} \boldsymbol{c}^{\top}, \tag{20}$$

where

$$\boldsymbol{u}^\top = \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}, \mathbf{B} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}, \boldsymbol{c}^\top = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{bmatrix}. \tag{21}$$

5) Uniform dual cubic B-spline surface: Similarly, considering the B-spline surface of degree 3×3 (namely $p = q = 3$) with the uniform knot vectors:

$$\begin{aligned} X &= \{x_1, x_2, \cdots, x_t \mid x_{i+1} - x_i = \text{const}, 1 \le i \le t, i \in \mathbb{Z}\}, \\ Y &= \{y_1, y_2, \cdots, y_s \mid y_{j+1} - y_j = \text{const}, 1 \le j \le s, j \in \mathbb{Z}\}. \end{aligned} \tag{22}$$

We represent the B-spline basis functions by the local parameter $u \in [0, 1]$ and $v \in [0, 1]$ instead of the global parameter $x \in [x_i, x_{i+1}]$ and $y \in [y_j, y_{j+1}]$ on each knot interval by performing the transformations as:

$$\begin{aligned} x &= x(u) = (1-u)x_i + ux_{i+1}, u \in [0, 1], \\ y &= y(v) = (1-v)y_j + yv_{j+1}, v \in [0, 1]. \end{aligned} \tag{23}$$

Then, the cubic B-spline surface $\mathcal{S}(x, y) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,3}(x)B_{j,3}(y)C_{i,j}$ can also be represented in compact matrix form:

$$\mathcal{S}(u, v) = \boldsymbol{u}\mathbf{B}\mathbf{C}\mathbf{B}^\top\boldsymbol{v}^\top, \tag{24}$$

where

$$\boldsymbol{u}^\top = \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}, \boldsymbol{v}^\top = \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}, \mathbf{B} = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} c_{0,0} & c_{0,1} & c_{0,2} & c_{0,3} \\ c_{1,0} & c_{1,1} & c_{1,2} & c_{1,3} \\ c_{2,0} & c_{2,1} & c_{2,2} & c_{2,3} \\ c_{3,0} & c_{3,1} & c_{3,2} & c_{3,3} \end{bmatrix}. \tag{25}$$

E. Ground Manifold Constraint

1) Manifold under global representation: Given that there is a ground manifold that can be represented as a dual cubic B-spline surface, then any three-dimensional (3D) point $\boldsymbol{p} = (x, y, z)$ on the manifold satisfies the following constraint equation:

$$z = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,3}(x)B_{j,3}(y)c_{i,j}. \tag{26}$$

Let $\mathcal{M}$ denote the manifold, then we have

$$\mathcal{M}(\boldsymbol{p}) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_{i,3}(x)B_{j,3}(y)c_{i,j} - z = 0. \tag{27}$$

Similarly, we transfer Eq.(27) to the form of the local coordinates (u, v) with each piece of the surface represented as follows:

$$\begin{aligned} \mathcal{M}(\boldsymbol{p}) &= \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,3}(u)B_{j,3}(v)c_{i,j} - z = 0 \\ &= \boldsymbol{u}\mathbf{B}\mathbf{C}\mathbf{B}^\top\boldsymbol{v}^\top - z = 0 \end{aligned} \tag{28}$$

where

$$
\begin{aligned}
u &= \frac{x - x_i}{x_{i+1} - x_i}, x \in [x_i, x_{i+1}], \\
v &= \frac{y - y_j}{y_{j+1} - y_j}, y \in [y_j, y_{j+1}].
\end{aligned} \tag{29}
$$

Let $d = x_{i+1} - x_i = y_{j+1} - y_j$ denotes the constant interval of the uniform knot vector of dual cubic B-spline surface, the transformation can be represented as follows:

$$
\begin{aligned}
u &= k_x x + b_x \in [0, 1],\ k_x = 1/d,\ b_x = -g_x/d, \\
v &= k_y y + b_y \in [0, 1],\ k_y = 1/d,\ b_y = -g_y/d,
\end{aligned} \tag{30}
$$

where $(g_x, g_y) = (x_i, y_j)$ is the global coordinate of the ground grid. So, the ground manifold in Eq.(28) can be abbreviated as follows:

$$
\begin{aligned}
\mathcal{M}(\boldsymbol{p}) &= \boldsymbol{u}\mathbf{B}\mathbf{C}\mathbf{B}^\top \boldsymbol{v}^\top - z = 0, \\
&= \boldsymbol{x}\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top \mathbf{K}_y^\top \boldsymbol{y}^\top - z = 0,
\end{aligned} \tag{31}
$$

where

$$
\boldsymbol{x}^\top = \begin{bmatrix} x^3 \\ x^2 \\ x \\ 1 \end{bmatrix}, \boldsymbol{y}^\top = \begin{bmatrix} y^3 \\ y^2 \\ y \\ 1 \end{bmatrix}, \mathbf{K}_x = \begin{bmatrix} k_x^3 & 0 & 0 & 0 \\ 3k_x^2 b_x & k_x^2 & 0 & 0 \\ 3k_x b_x^2 & 2k_x b_x & k_x & 0 \\ b_x^3 & b_x^2 & b_x & 1 \end{bmatrix}, \mathbf{K}_y = \begin{bmatrix} k_y^3 & 0 & 0 & 0 \\ 3k_y^2 b_y & k_y^2 & 0 & 0 \\ 3k_y b_y^2 & 2k_y b_y & k_y & 0 \\ b_y^3 & b_y^2 & b_y & 1 \end{bmatrix}, \tag{32}
$$

and $\mathbf{B}$ and $\mathbf{C}$ are the same as in Eq.(25).

2) Ground manifold constraint: Furthermore, given that the robot travels on the ground that is parameterized as a segmented B-spline surface as Eq.(31), which indicates that the poses of the tangent points between the wheels and the ground satisfy two kinds of constraints, one is that the position ${}^{\mathcal{G}}\boldsymbol{p}_w$ is on the surface, and the other is that the orientation ${}^{\mathcal{G}}_{w}\mathbf{R}$ is parallel to the normal of the surface manifold:

$$
\mathcal{M}\left({}^{\mathcal{G}}\boldsymbol{p}_w\right) = 0, \tag{33a}
$$

$$
{}^{\mathcal{G}}_{w}\mathbf{R} \cdot \boldsymbol{e}_3 \times \nabla\mathcal{M}\left({}^{\mathcal{G}}\boldsymbol{p}_w\right) = \mathbf{0}, \tag{33b}
$$

where $\boldsymbol{e}_3 = [0, 0, 1]^\top$ is the third axis, $\nabla = \frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial y}\vec{j} + \frac{\partial}{\partial z}\vec{k}$ is a vector differential operator, and $\mathbf{0}$ is the zero vector. The pose of any wheel tangent point $\{{}^{\mathcal{G}}_{w}\mathbf{R}, {}^{\mathcal{G}}\boldsymbol{p}_w\}$ can be converted by extrinsic transformation $\{{}^{w}_{\mathcal{I}}\mathbf{R}, {}^{\mathcal{I}}\boldsymbol{t}_w\}$ from the IMU pose $\{{}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}, {}^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}}\}$.

$$
{}^{\mathcal{G}}\boldsymbol{p}_w = {}^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}} + {}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}\ {}^{\mathcal{I}}\boldsymbol{t}_w, \tag{34a}
$$

$$
{}^{\mathcal{G}}_{w}\mathbf{R} = {}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}\ {}^{\mathcal{I}}_{w}\mathbf{R}. \tag{34b}
$$

Assuming that the $\mathcal{I}$ frame and the w frame have the same rotation representation, we substitute Eq.(4), Eq.(31) and Eq.(34) into equation Eq.(33), and after simplification we can derive:

$$
\boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{y} - z_w = 0, \tag{35a}
$$

$$
\boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial}\boldsymbol{y}_w + 2\frac{s_1\sin(\psi) + s_2\cos(\psi)}{1 - s_1^2 - s_2^2} = 0, \tag{35b}
$$

$$
\boldsymbol{\partial}\boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{y}_w + 2\frac{s_1\cos(\psi) - s_2\sin(\psi)}{1 - s_1^2 - s_2^2} = 0, \tag{35c}
$$

where

$$\boldsymbol{x}_w = \begin{bmatrix} x_w^3 \\ x_w^2 \\ x_w \\ 1 \end{bmatrix}, \boldsymbol{\partial x}_w = \begin{bmatrix} 3x_w^2 \\ 2x_w \\ 1 \\ 0 \end{bmatrix}, \boldsymbol{y_w} = \begin{bmatrix} y_w^3 \\ y_w^2 \\ y_w \\ 1 \end{bmatrix}, \boldsymbol{\partial y_w} = \begin{bmatrix} 3y_w^2 \\ 2y_w \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} x_\mathcal{I} \\ y_\mathcal{I} \\ z_\mathcal{I} \end{bmatrix} + {}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}\ {}^{\mathcal{I}}\boldsymbol{t}_w, \tag{36}$$

$\mathbf{K}_x$, $\mathbf{K}_y$, $\mathbf{B}$ and $\mathbf{C}$ have the definition as Eq.(32), $\{s_1, s_2, \psi\}$ are the abbreviations of IMU rotation $\{{}^{\mathcal{G}}_{\mathcal{I}}s_1, {}^{\mathcal{G}}_{\mathcal{I}}s_2, {}^{\mathcal{G}}_{\mathcal{I}}\psi\}$. Furthermore, we can transfer Eq.(35) to a vector form w.r.t. the matrix $\mathbf{C}$:

$$(\boldsymbol{y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{x}_w\mathbf{K}_x\mathbf{B})Vec(\mathbf{C}) - z = 0, \tag{37a}$$

$$(\boldsymbol{\partial y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{x}_w\mathbf{K}_x\mathbf{B})Vec(\mathbf{C}) + 2\frac{s_1\sin(\psi) + s_2\cos(\psi)}{1 - s_1^2 - s_2^2} = 0, \tag{37b}$$

$$(\boldsymbol{y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B})Vec(\mathbf{C}) + 2\frac{s_1\cos(\psi) - s_2\sin(\psi)}{1 - s_1^2 - s_2^2} = 0, \tag{37c}$$

where $\otimes$ is the Kronecker product, $Vec(\mathbf{C})$ is the column straightening function that transforms a matrix into a vector.

II. Network Processing

In our previous work DIDO [2], we attempt to use ResNet [3] to process the raw IMU data and remove the biases of accelerometer and gyroscope with some positive results. Here, we still adopt a similar network to process the raw data from two interoceptive sensors (IMU and wheel odometer).

A. IMU De-Bias Net

In *De-Bias Net*, we utilize the ResNet-1D [3] to learn the kinematic characteristics of IMU for de-biasing. The respective input features of the *De-Bias Net*s are historical raw accelerometer measurements ${}^{\mathcal{I}}\widetilde{\boldsymbol{a}}$ and gyroscope measurements ${}^{\mathcal{I}}\widetilde{\boldsymbol{\omega}}$, the outputs are $\{{}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{a}}, diag(\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{v}_{i,i+n}})\}$ and $\{{}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}, diag(\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{q}_{i,i+n}})\}$ at every moment, separately. The Mean Square Error (MSE) loss functions and the Negative Log Likelihood (NLL) loss functions of relative velocity and relative rotation (in quaternion) are defined on the following integrated increments:

$$\mathcal{L}_{\text{MSE},\boldsymbol{a}} = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{v}_{i,i+n} - \widehat{\boldsymbol{v}}_{i,i+n}\|_2^2, \tag{38a}$$

$$\mathcal{L}_{\text{NLL},d\boldsymbol{v}} = \frac{1}{2N}\sum_{i=1}^{N} \log\det(\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{v}_{i,i+n}}) + \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{v}_{i,i+n} - \widehat{\boldsymbol{v}}_{i,i+n}\|^2_{\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{v}_{i,i+n}}}, \tag{38b}$$

$$\mathcal{L}_{\text{MSE},\boldsymbol{\omega}} = \frac{1}{N}\sum_{i=1}^{N} \|\text{Log}\big((\widehat{\boldsymbol{q}}_{i,i+n})^*\,\boldsymbol{q}_{i,i+n}\big)\|_2^2, \tag{38c}$$

$$\mathcal{L}_{\text{NLL},d\boldsymbol{q}} = \frac{1}{2N}\sum_{i=1}^{N} \log\det(\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{q}_{i,i+n}}) + \frac{1}{N}\sum_{i=1}^{N} \|\text{Log}\big((\widehat{\boldsymbol{q}}_{i,i+n})^*\,\boldsymbol{q}_{i,i+n}\big)\|^2_{\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{q}_{i,i+n}}}, \tag{38d}$$

where

$$\widehat{\boldsymbol{v}}_{i,i+n} = \int_i^{i+n} {}^{\mathcal{G}}_{\mathcal{I}_t}\mathbf{R}({}^{\mathcal{I}_t}\widetilde{\boldsymbol{a}} - {}^{\mathcal{I}_t}\widehat{\boldsymbol{b}}_{\boldsymbol{a}})dt, \tag{39a}$$

$$\boldsymbol{v}_{i,i+n} = {}^{\mathcal{G}}\boldsymbol{v}_{\mathcal{I}_{i+n}} - {}^{\mathcal{G}}\boldsymbol{v}_{\mathcal{I}_i}, \tag{39b}$$

$$\widehat{\boldsymbol{q}}_{i,i+n} = \int_i^{i+n} \frac{1}{2}{}^{\mathcal{I}_i}_{\mathcal{I}_t}\boldsymbol{q}\ ({}^{\mathcal{I}_t}\widetilde{\boldsymbol{\omega}} - {}^{\mathcal{I}_t}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}})dt, \tag{39c}$$

$$\boldsymbol{q}_{i,i+n} = ({}^{\mathcal{G}}_{\mathcal{I}_i}\boldsymbol{q})^*\ {}^{\mathcal{G}}_{\mathcal{I}_{i+n}}\boldsymbol{q}, \tag{39d}$$

and the capitalized logarithmic map and exponential map of quaternion are defined as in [4]:

$$\begin{aligned} &\mathrm{Log} : S^3 \to \mathfrak{so}(3); \quad \boldsymbol{q} \in S^3 \overset{\mathrm{Log}}{\longmapsto} \boldsymbol{u} \in \mathfrak{so}(3), \\ &\mathrm{Exp} : \mathfrak{so}(3) \to S^3; \quad \boldsymbol{u} \in \mathfrak{so}(3) \overset{\mathrm{Exp}}{\longmapsto} \boldsymbol{q} \in S^3. \end{aligned} \tag{40}$$

Note that, the subscript i and $i+n$ is the abbreviation for time t_i and t_{i+n}, $\widehat{\boldsymbol{v}}_{i,i+n}$ and $\widehat{\boldsymbol{q}}_{i,i+n}$ are the intermediate variables calculated from the estimated values ${}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{a}}$ and ${}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}$ of the *De-Bias Net* by the forward Euler method actually, $\boldsymbol{v}_{i,i+n}$ and $\boldsymbol{q}_{i,i+n}$ are the ground truth velocity increment and relative rotation of each input sliding window, n is the window size, $\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{v}_{i,i+n}}$ and $\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{q}_{i,i+n}}$ are the estimated covariance matrices, and N is the batch size. In training, ${}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}$ is the ground truth rotation. The quaternion conjugate, quaternion multiplication and quaternion logarithm operations are defined in [4].

Considering the following kinematic models as utilized in the training:

$$\boldsymbol{v}_{i,k} = \int_i^k {}^{\mathcal{G}}_{\mathcal{I}_t}\mathbf{R}({}^{\mathcal{I}_t}\widetilde{\boldsymbol{a}} - {}^{\mathcal{I}_t}\widehat{\boldsymbol{b}}_{\boldsymbol{a}} - \boldsymbol{n}_{\boldsymbol{a}})dt, \tag{41a}$$

$$\boldsymbol{q}_{i,k} = \int_i^k \frac{1}{2}{}^{\mathcal{I}_i}_{\mathcal{I}_t}\boldsymbol{q}\ ({}^{\mathcal{I}_t}\widetilde{\boldsymbol{\omega}} - {}^{\mathcal{I}_t}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}} - \boldsymbol{n}_{\boldsymbol{\omega}})dt, \tag{41b}$$

where ${}^{\mathcal{G}}_{\mathcal{I}_t}\mathbf{R}$ is the ground truth, ${}^{\mathcal{I}_t}\widehat{\boldsymbol{b}}_{\boldsymbol{a}}$ and ${}^{\mathcal{I}_t}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}$ are estimated values of the *De-Bias Net*, $\boldsymbol{n}_{\boldsymbol{a}}$ and $\boldsymbol{n}_{\boldsymbol{\omega}}$ are the input noise of the integral system in Eq.(41). We discretize the above equations to obtain the nominal state equations:

$$\boldsymbol{v}_{i,k+1} = \boldsymbol{v}_{i,k} + {}^{\mathcal{G}}_{\mathcal{I}_k}\mathbf{R}({}^{\mathcal{I}_k}\widetilde{\boldsymbol{a}} - {}^{\mathcal{I}_k}\widehat{\boldsymbol{b}}_{\boldsymbol{a}})\Delta t_k, \tag{42a}$$

$$\boldsymbol{q}_{i,k+1} = \boldsymbol{q}_{i,k}\ \boldsymbol{q}\{({}^{\mathcal{I}_t}\widetilde{\boldsymbol{\omega}} - {}^{\mathcal{I}_t}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}})\Delta t_k\}, \tag{42b}$$

with the following discrete error state models:

$$\delta\boldsymbol{v}_{i,k+1} = \delta\boldsymbol{v}_{i,k} + {}^{\mathcal{G}}_{\mathcal{I}_k}\mathbf{R}\mathbf{v}_{\mathbf{i}} = \delta\boldsymbol{v}_{i,k} + \mathbf{v}_{\mathbf{i}}, \tag{43a}$$

$$\delta\boldsymbol{\theta}_{i,k+1} = \mathbf{R}^\top\{({}^{\mathcal{I}_t}\widetilde{\boldsymbol{\omega}} - {}^{\mathcal{I}_t}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}})\Delta t_k\}\delta\boldsymbol{\theta}_{i,k} + \boldsymbol{\theta}_{\mathbf{i}}, \tag{43b}$$

$$\mathbf{i} = \begin{bmatrix}\mathbf{v}_{\mathbf{i}} \\ \boldsymbol{\theta}_{\mathbf{i}}\end{bmatrix} \sim \mathcal{N}(0, \mathbf{Q}_{\mathbf{i}}), \quad \mathbf{Q}_{\mathbf{i}} = \Delta t^2 \begin{bmatrix}\boldsymbol{\Sigma}^2_{\boldsymbol{a}} & \mathbf{0}_3 \\ \mathbf{0}_3 & \boldsymbol{\Sigma}^2_{\boldsymbol{\omega}}\end{bmatrix}, \tag{43c}$$

where $\mathbf{i}$ is the perturbation impulse vector as in [4] and the time interval Δt_k are assumed to be equal as Δt. Note that, there is no error propagation about the rotation since the ground truth ${}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}$ is provided. We combine the two states to yield the following error state equation:

$$\delta\boldsymbol{x}_{i,k+1} = \begin{bmatrix}\delta\boldsymbol{v}_{i,k+1} \\ \delta\boldsymbol{\theta}_{i,k+1}\end{bmatrix} = \boldsymbol{f}(\boldsymbol{x}_{i,k}, \delta\boldsymbol{x}_{i,k}, \boldsymbol{u}_k, \mathbf{i}) \tag{44a}$$

$$= \mathbf{F}_{\mathbf{x},k}(\boldsymbol{x}_{i,k}, \boldsymbol{u}_k)\delta\boldsymbol{x}_{i,k} + \mathbf{F}_{\mathbf{i},k}\ \mathbf{i}, \tag{44b}$$

whose error state prediction equations are written:

$$\delta\widehat{\boldsymbol{x}}_{i,k+1} = \mathbf{F}_{\mathbf{x},k}(\boldsymbol{x}_{i,k}, \boldsymbol{u}_k)\delta\widehat{\boldsymbol{x}}_{i,k}, \tag{45a}$$

$$\mathbf{P}_{i,k+1} = \mathbf{F}_{\mathbf{x},k}\mathbf{P}_{i,k}\mathbf{F}_{\mathbf{x},k}^{\top} + \mathbf{F}_{\mathbf{i},k}\mathbf{Q}_{\mathbf{i}}\mathbf{F}_{\mathbf{i},k}^{\top}, \tag{45b}$$

where the Jacobi matrices are:

$$\mathbf{F}_{\mathbf{x},k} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{R}^{\top}\{({}^{\mathcal{I}_k}\widetilde{\boldsymbol{\omega}} - {}^{\mathcal{I}_k}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}})\Delta t\} \end{bmatrix}, \quad \mathbf{F}_{\mathbf{i}} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}, \tag{46a}$$

so we can obtain the propagation of the covariance as follows:

$$\begin{aligned}
\mathbf{P}_{i,i+1} &= \mathbf{F}_{\mathbf{x},i+1}\mathbf{P}_{i,i}\mathbf{F}_{\mathbf{x},i+1}^{\top} + \mathbf{Q}_{\mathbf{i}}, \\
\mathbf{P}_{i,i+2} &= \mathbf{F}_{\mathbf{x},i+2}\mathbf{F}_{\mathbf{x},i+1}\mathbf{P}_{i,i}\mathbf{F}_{\mathbf{x},i+1}^{\top}\mathbf{F}_{\mathbf{x},i+2}^{\top} + \mathbf{F}_{\mathbf{x},i+2}\mathbf{P}_{i,i}\mathbf{F}_{\mathbf{x},i+2}^{\top} + \mathbf{Q}_{\mathbf{i}}, \\
&\vdots \\
\mathbf{P}_{i,i+n} &= \mathbf{F}_{\mathbf{x},i+n}\cdots\mathbf{F}_{\mathbf{x},i+1}\mathbf{P}_{i,i}\mathbf{F}_{\mathbf{x},i+1}^{\top}\cdots\mathbf{F}_{\mathbf{x},i+n}^{\top} \\
&\quad + \mathbf{F}_{\mathbf{x},i+n}\cdots\mathbf{F}_{\mathbf{x},i+2}\mathbf{Q}_{\mathbf{i}}\mathbf{F}_{\mathbf{x},i+2}^{\top}\cdots\mathbf{F}_{\mathbf{x},i+n}^{\top} \\
&\quad + \cdots \\
&\quad + \mathbf{F}_{\mathbf{x},i+n}\mathbf{Q}_{\mathbf{i}}\mathbf{F}_{\mathbf{x},i+n}^{\top} \\
&\quad + \mathbf{Q}_{\mathbf{i}},
\end{aligned} \tag{47}$$

Assuming that the initial covariance matrix is:

$$\mathbf{P}_{i,i} = \begin{bmatrix} \mu\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \nu\mathbf{I}_3 \end{bmatrix}, \ \mu{>}0, \nu{>}0, \tag{48}$$

then we can derive:

$$\mathbf{P}_{i,i+n} = \begin{bmatrix} \mu\mathbf{I}_3 + n\Delta t^2\boldsymbol{\Sigma}_{\boldsymbol{a}}^2 & \mathbf{0}_3 \\ \mathbf{0}_3 & \nu\mathbf{I}_3 + n\Delta t^2\boldsymbol{\Sigma}_{\boldsymbol{\omega}}^2 \end{bmatrix}. \tag{49}$$

Furthermore, since the initial values of each input window are ground truth in training and the initial covariance matrix $\mathbf{P}_{i,i}$ is very small (but cannot be set to 0), then we approximate that

$$\mathbf{P}_{i,i+n} \approx n\Delta t^2 \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{a}}^2 & \mathbf{0}_3 \\ \mathbf{0}_3 & \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^2 \end{bmatrix}. \tag{50}$$

The estimated covariance matrices $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{v}_{i,i+n}}^2$ and $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{q}_{i,i+n}}^2$ in Eq.(38b) and Eq.(38d) output from the *De-Bias Net* could be approximated to the theoretical value:

$$\mathbf{P}_{i,i+n} \approx n\Delta t^2 \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{a}}^2 & \mathbf{0}_3 \\ \mathbf{0}_3 & \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^2 \end{bmatrix} \Longleftarrow \begin{bmatrix} \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{v}_{i,i+n}}^2 & \mathbf{0}_3 \\ \mathbf{0}_3 & \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{q}_{i,i+n}}^2 \end{bmatrix}. \tag{51}$$

B. Wheel Odometer Net

Commonly, it is assumed that there is no speed in the non-forward direction:

$${}^{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}} = \begin{bmatrix} {}^{\mathcal{B}}\widetilde{v}_{\mathcal{B}_x} & 0 & 0 \end{bmatrix}^{\top}, \tag{52}$$

where ${}^{\mathcal{B}}\widetilde{v}_{\mathcal{B}_x}$ is the observation value from wheel odometer. However, the above nonholonomic constraint are not always strictly satisfied due to slippage. In addition, the extrinsic parameters $\{{}^{\mathcal{I}}_{\mathcal{B}}\mathbf{R}, {}^{\mathcal{I}}\boldsymbol{t}_{\mathcal{B}}\}$ between the IMU frame ($\mathcal{I}$) and the wheel frame ($\mathcal{B}$) require real-time calibration due to the effect of the vibrations and suspension system in motion. We aim to introduce IMU data into the network to tackle wheel speed outliers and time-varying extrinsic parameters. Considering the following velocity transfer

equation:

$$ {}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}} = {}^{\mathcal{I}}_{\mathcal{B}}\mathbf{R}\ {}^{\mathcal{B}}\boldsymbol{v}_{\mathcal{B}} - {}^{\mathcal{I}}\boldsymbol{\omega} \times {}^{\mathcal{I}}\boldsymbol{t}_{\mathcal{B}} - {}^{\mathcal{I}}\dot{\boldsymbol{t}}_{\mathcal{B}}, \tag{53} $$

we feed the same neural network the ResNet-1D with raw wheel speed ${}^{\mathcal{B}}\widetilde{v}_{\mathcal{B}_x}$, raw gyroscope measurements ${}^{\mathcal{I}}\widetilde{\boldsymbol{\omega}}$ within a sliding window. The similar MSE and NLL loss functions of body velocity are defined as:

$$ \mathcal{L}_{\text{MSE},\boldsymbol{v}} = \frac{1}{N}\sum_{i=1}^{N} \|{}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}i} - {}^{\mathcal{I}}\widehat{\boldsymbol{v}}_{\mathcal{I}i}\|_2^2, \tag{54a} $$

$$ \mathcal{L}_{\text{NLL},\boldsymbol{v}} = \frac{1}{2N}\sum_{i=1}^{N} \log\det(\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{v}_i}) + \frac{1}{N}\sum_{i=1}^{N} \|{}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}i} - {}^{\mathcal{I}}\widehat{\boldsymbol{v}}_{\mathcal{I}i}\|^2_{\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{v}_i}}, \tag{54b} $$

where $\{{}^{\mathcal{I}}\widehat{\boldsymbol{v}}_{\mathcal{I}}, diag(\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{v}})\}$ are the network outputs, ${}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}}$ is the ground truth velocity in the $\mathcal{I}$ frame.

C. Training details

Training details

In detail, the Mean Square Error (MSE) loss is replaced by the negative log likelihood NLL loss after the (MSE) loss stabilizes and converges as in [5].

When NLL loss is utilized, the MSE index will not be the optimal.

III. Space-based Sliding Window Filtering

A. State

In this work, we fuse the neural processed IMU and wheel odometer data using a filtering framework. The state vector $\mathcal{X}$ includes IMU state $\mathcal{X}_I$, space-based sliding window state $\mathcal{X}_S$ and space-based sliding control mesh $\boldsymbol{c}$, which is represented as follows:

$$ \mathcal{X} = \begin{bmatrix} \mathcal{X}_I{}^{\top} & \mathcal{X}_S{}^{\top} & \boldsymbol{c}^{\top} \end{bmatrix}^{\top}, \tag{55} $$

$$ \mathcal{X}_I = \begin{bmatrix} {}^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}}{}^{\top} & {}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}}{}^{\top} & {}^{\mathcal{G}}_{\mathcal{I}}\psi & {}^{\mathcal{G}}_{\mathcal{I}}\boldsymbol{s}^{\top} \end{bmatrix}^{\top}, \tag{55a} $$

$$ \mathcal{X}_S = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \cdots & \boldsymbol{\xi}_n \end{bmatrix}^{\top}, \quad \boldsymbol{\xi}_j = \begin{bmatrix} {}^{\mathcal{G}}\boldsymbol{p}^{\top}_{\mathcal{I}_j} & {}^{\mathcal{G}}_{\mathcal{I}_j}\psi & {}^{\mathcal{G}}_{\mathcal{I}_j}\boldsymbol{s}^{\top} \end{bmatrix}^{\top}, \tag{55b} $$

$$ \boldsymbol{c} = Vec(\mathbf{C}), \tag{55c} $$

where ${}^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}}$, ${}^{\mathcal{G}}_{\mathcal{I}}\psi$ and ${}^{\mathcal{G}}_{\mathcal{I}}\boldsymbol{s}$ are the position, yaw angle and tilt vector of the ground robot in the $\mathcal{G}$ frame, ${}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}}$ is the velocity of the robot in $\mathcal{I}$ frame, $\boldsymbol{\xi}_j$ is the jth state of the space-based sliding window, and $\boldsymbol{c}$ is the control mesh vector obtained by column straightening of the 4×4 control mesh matrix $\mathbf{C}$.

This work leverages a uniform dual cubic B-spline surface to characterize the ground that the vehicle travels, so a control polygon with a fixed interval (parameter d) horizontally is shown as all circles in Fig. 1. In fact, we optimize only states associated with the mesh where the vehicle is traveling currently, so that the trajectories ($\mathcal{X}_S$, yellow lines) and control mesh points ($\mathbf{C}$, 4×4 yellow circles) in Fig. 1 are the current optimized states.

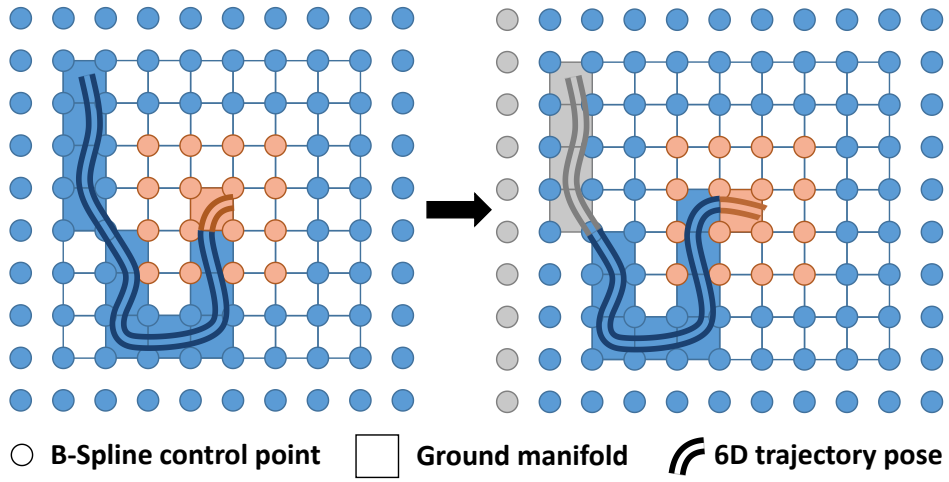

Fig. 1: The demonstration of space-based sliding window update strategy. Yellow, blue and gray represent the active, the static and the marginalized states or information in the proposed system, respectively.

B. IMU Process Model

In this work, we leverage the neural processed IMU measurements to drive the state estimation system, whose full process model is given as follows:

$$
\begin{aligned}
{}^{\mathcal{G}}\dot{\boldsymbol{p}}_{\mathcal{I}} &= {}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}\,{}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}}, && \text{(56a)}\\
{}^{\mathcal{I}}\dot{\boldsymbol{v}}_{\mathcal{I}} &= {}^{\mathcal{I}}\boldsymbol{a} - {}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}^{\top}{}^{\mathcal{G}}\mathbf{g} - {}^{\mathcal{I}}\boldsymbol{\omega} \times {}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}}, && \text{(56b)}\\
{}^{\mathcal{G}}_{\mathcal{I}}\dot{\psi} &= \begin{bmatrix} -s_1 & -s_2 & 1 \end{bmatrix} {}^{\mathcal{I}}\boldsymbol{\omega}, && \text{(56c)}\\
{}^{\mathcal{G}}_{\mathcal{I}}\dot{\boldsymbol{s}} &= \frac{1}{2}\begin{bmatrix} -2s_1s_2 & s_1^2 - s_2^2 + 1 & 2s_2 \\ s_1^2 - s_2^2 - 1 & 2s_1s_2 & -2s_1 \end{bmatrix} {}^{\mathcal{I}}\boldsymbol{\omega}, && \text{(56d)}
\end{aligned}
$$

where the rotation ${}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R} = \mathbf{R}_{\psi}\mathbf{R}_{\phi}$ is expressed as a form of yaw-independent as [1], and

$$
\begin{aligned}
{}^{\mathcal{I}}\boldsymbol{\omega} &= {}^{\mathcal{I}}\widetilde{\boldsymbol{\omega}} - {}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}} - \boldsymbol{n}_{\boldsymbol{\omega}}, \quad \boldsymbol{n}_{\boldsymbol{\omega}} \sim \mathcal{N}(0, \widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{\omega}}), && \text{(57a)}\\
{}^{\mathcal{I}}\boldsymbol{a} &= {}^{\mathcal{I}}\widetilde{\boldsymbol{a}} - {}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{a}} - \boldsymbol{n}_{\boldsymbol{a}}, \quad \boldsymbol{n}_{\boldsymbol{a}} \sim \mathcal{N}(0, \widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{a}}), && \text{(57b)}
\end{aligned}
$$

where ${}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}}$ and ${}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{a}}$ are the outputs of the *De-Bias Net* in Sec.II-A, and

$$
\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{\omega}} = \frac{1}{n\Delta t^2}\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{q}_{i,i+n}}, \quad \widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{a}} = \frac{1}{n\Delta t^2}\widehat{\boldsymbol{\Sigma}}^2_{\boldsymbol{v}_{i,i+n}}. \tag{58a}
$$

The differential equations Eq.(56) can be discretized as follows:

$$
\begin{aligned}
\mathcal{X}_{\mathcal{I}_{k+1}} &= \mathbf{F}(\mathcal{X}_{\mathcal{I}_k}, \boldsymbol{u}_k) \\
&= \mathbf{F}_{\mathcal{I}}\mathcal{X}_{\mathcal{I}_k} + \mathbf{F}_{\boldsymbol{n}}\boldsymbol{n}_k,
\end{aligned} \tag{59}
$$

where

$$
\begin{aligned}
\mathcal{X}_{\mathcal{I}_k} &= \begin{bmatrix} {}^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_k} & {}^{\mathcal{I}_k}\boldsymbol{v} & {}^{\mathcal{G}}_{\mathcal{I}_k}\psi & {}^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s} \end{bmatrix}^{\top}, \\
\boldsymbol{u}_k &= \begin{bmatrix} {}^{\mathcal{I}_k}\widehat{\boldsymbol{\omega}} - \boldsymbol{n}_{\boldsymbol{\omega}} \\ {}^{\mathcal{I}_k}\widehat{\boldsymbol{a}} - \boldsymbol{n}_{\boldsymbol{a}} \end{bmatrix} = \begin{bmatrix} {}^{\mathcal{I}}\widetilde{\boldsymbol{\omega}} - {}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{\omega}} - \boldsymbol{n}_{\boldsymbol{\omega}} \\ {}^{\mathcal{I}}\widetilde{\boldsymbol{a}} - {}^{\mathcal{I}}\widehat{\boldsymbol{b}}_{\boldsymbol{a}} - \boldsymbol{n}_{\boldsymbol{a}} \end{bmatrix},
\end{aligned} \tag{60}
$$

$$\mathbf{F}_{\mathcal{I}}=\begin{bmatrix}\mathbf{I}_3 & {}^{\mathcal{G}}_{\mathcal{I}_k}\mathbf{R}\Delta t_k & \frac{\partial^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_{k+1}}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\psi} & \frac{\partial^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_{k+1}}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s}}\\ \mathbf{0}_3 & \mathbf{I}_3-\lfloor\boldsymbol{\omega}_k\rfloor_\times\Delta t_k & \mathbf{0}_{3\times1} & \frac{\partial^{\mathcal{I}_{k+1}}\boldsymbol{v}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s}}\\ \mathbf{0}_{1\times3} & \mathbf{0}_{1\times3} & 1 & \frac{\partial^{\mathcal{G}}_{\mathcal{I}_{k+1}}\psi}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s}}\\ \mathbf{0}_{2\times3} & \mathbf{0}_{2\times3} & \mathbf{0}_{2\times1} & \frac{\partial^{\mathcal{G}}_{\mathcal{I}_{k+1}}\boldsymbol{s}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s}}\end{bmatrix},\quad \mathbf{F}_{\boldsymbol{n}}=\begin{bmatrix}\mathbf{0}_3 & \mathbf{0}_3\\ \frac{\partial^{\mathcal{I}_{k+1}}\boldsymbol{v}}{\partial\boldsymbol{n}_{\boldsymbol{a}}} & \frac{\partial^{\mathcal{I}_{k+1}}\boldsymbol{v}}{\partial\boldsymbol{n}_{\boldsymbol{\omega}}}\\ \mathbf{0}_{1\times3} & \frac{\partial^{\mathcal{G}}_{\mathcal{I}_{k+1}}\psi}{\partial\boldsymbol{n}_{\boldsymbol{\omega}}}\\ \mathbf{0}_{2\times3} & \frac{\partial^{\mathcal{G}}_{\mathcal{I}_{k+1}}\boldsymbol{s}}{\partial\boldsymbol{n}_{\boldsymbol{\omega}}}\end{bmatrix}. \tag{61}$$

Specifically,

$$\begin{aligned}
\frac{\partial^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_{k+1}}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\psi}&=\begin{bmatrix}-\sin\psi & -\cos\psi & 0\\ \cos\psi & -\sin\psi & 0\\ 0 & 0 & 0\end{bmatrix}\mathbf{R}_\phi{}^{\mathcal{I}}\boldsymbol{v}\Delta t_k,\\
\frac{\partial^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_{k+1}}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s}}&=\begin{bmatrix}\frac{\partial^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_{k+1}}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}s_1} & \frac{\partial^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_{k+1}}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}s_2}\end{bmatrix},\\
\frac{\partial^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_{k+1}}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}s_1}&=\frac{\mathbf{R}_\psi}{1+s_1^2+s_2^2}\begin{bmatrix}-4s_1(1+s_2^2) & -2s_2(1-s_1^2+s_2^2) & 2(1-s_1^2+s_2^2)\\ -2s_2(1-s_1^2+s_2^2) & 4s_1s_2^2 & -4s_1s_2\\ -2(1-s_1^2+s_2^2) & 4s_1s_2 & -4s_1\end{bmatrix}{}^{\mathcal{I}}\boldsymbol{v}\Delta t_k,\\
\frac{\partial^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_{k+1}}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}s_2}&=\frac{\mathbf{R}_\psi}{1+s_1^2+s_2^2}\begin{bmatrix}4s_1^2s_2 & -2s_2(1+s_1^2-s_2^2) & -4s_1s_2\\ -2s_2(1+s_1^2-s_2^2) & -4s_2(1+s_1^2) & 2(1+s_1^2-s_2^2)\\ 4s_1s_2 & -2(1+s_1^2-s_2^2) & -4s_2\end{bmatrix}{}^{\mathcal{I}}\boldsymbol{v}\Delta t_k,\\
\frac{\partial^{\mathcal{I}_{k+1}}\boldsymbol{v}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s}}&=\frac{-g}{1+s_1^2+s_2^2}\begin{bmatrix}-2(1-s_1^2+s_2^2) & 4s_1s_2\\ 4s_1s_2 & -2(1+s_1^2-s_2^2)\\ -4s_1 & -4s_2\end{bmatrix}\Delta t_k,\\
\frac{\partial^{\mathcal{G}}_{\mathcal{I}_{k+1}}\psi}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s}}&=-\begin{bmatrix}\omega_1 & \omega_2\end{bmatrix}\Delta t_k,\\
\frac{\partial^{\mathcal{G}}_{\mathcal{I}_{k+1}}\boldsymbol{s}}{\partial^{\mathcal{G}}_{\mathcal{I}_k}\boldsymbol{s}}&=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}+\begin{bmatrix}\omega_2s_1-\omega_1s_2 & -\omega_1s1-\omega_2s_2+\omega_3\\ \omega_1s1+\omega_2s_2-\omega_3 & -\omega_2s_s-\omega_1s_2\end{bmatrix}\Delta t_k,\\
\frac{\partial^{\mathcal{I}_{k+1}}\boldsymbol{v}}{\partial\boldsymbol{n}_{\boldsymbol{a}}}&=-\mathbf{I}_3\Delta t_k,\\
\frac{\partial^{\mathcal{I}_{k+1}}\boldsymbol{v}}{\partial\boldsymbol{n}_{\boldsymbol{\omega}}}&=-\left\lfloor{}^{\mathcal{I}}\boldsymbol{v}\right\rfloor_\times\Delta t_k,\\
\frac{\partial^{\mathcal{G}}_{\mathcal{I}_{k+1}}\psi}{\partial\boldsymbol{n}_{\boldsymbol{\omega}}}&=\begin{bmatrix}s_1 & s_2 & -1\end{bmatrix}\Delta t_k,\\
\frac{\partial^{\mathcal{G}}_{\mathcal{I}_{k+1}}\boldsymbol{s}}{\partial\boldsymbol{n}_{\boldsymbol{\omega}}}&=\begin{bmatrix}s_1s_2 & \frac{1}{2}(-1-s_1^2+s_2^2) & -s_2\\ \frac{1}{2}(1-s_1^2+s_2^2) & -s_1s_2 & s_1\end{bmatrix}\Delta t_k,
\end{aligned}\tag{62}$$

where $\{{}^{\mathcal{I}}\boldsymbol{v},\psi,\phi,s_1,s_2,\omega_1,\omega_2,\omega_3\}$ are the abbreviations of $\{{}^{\mathcal{I}_k}\boldsymbol{v},{}^{\mathcal{G}}_{\mathcal{I}_k}\psi,{}^{\mathcal{G}}_{\mathcal{I}_k}\phi,{}^{\mathcal{G}}_{\mathcal{I}_k}s_1,{}^{\mathcal{G}}_{\mathcal{I}_k}s_2,{}^{\mathcal{I}_k}\omega_1,{}^{\mathcal{I}_k}\omega_2,{}^{\mathcal{I}_k}\omega_3\}$ and $\lfloor\boldsymbol{a}\rfloor_\times$ is the skew symmetric matrix of any vector $\boldsymbol{a}$.

C. Observation Model

There are two kinds of measurements in the proposed system.

1) Neural velocity measurement: We could obtain triaxial pseudo-measurements of the IMU's velocity through a convolutional neural network in Sec.II-B:

$$\boldsymbol{h}_{\boldsymbol{v}}(\mathcal{X})={}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}}={}^{\mathcal{I}}\widehat{\boldsymbol{v}}_{\mathcal{I}}+\boldsymbol{n}_{\boldsymbol{v}},\tag{63}$$

where $\boldsymbol{n_v} \sim \mathcal{N}(0, \widehat{\boldsymbol{\Sigma}}_{\boldsymbol{v}}^2)$, ${}^{\mathcal{I}}\widehat{\boldsymbol{v}}_{\mathcal{I}} = \begin{bmatrix} {}^{\mathcal{I}}\widehat{\boldsymbol{v}}_{\mathcal{I}_x} & {}^{\mathcal{I}}\widehat{\boldsymbol{v}}_{\mathcal{I}_y} & {}^{\mathcal{I}}\widehat{\boldsymbol{v}}_{\mathcal{I}_z} \end{bmatrix}^\top$ and $diag(\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{v}}^2)$ is the output of the *Wheel-Odometer Net*. Specifically, the noise $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{v}}^2$ considers only for 3 elements of the main axis.

The jacobian of Eq.(63) is:

$$\begin{aligned} \frac{\partial \boldsymbol{h_v}(\mathcal{X})}{\partial \mathcal{X}} &= \begin{bmatrix} \frac{\partial \boldsymbol{h_v}(\mathcal{X})}{\partial \mathcal{X}_I} & \frac{\partial \boldsymbol{h_v}(\mathcal{X})}{\partial \mathcal{X}_S} & \frac{\partial \boldsymbol{h_v}(\mathcal{X})}{\partial \boldsymbol{c}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_{3\times(3+6n+16)} \end{bmatrix}, \end{aligned} \tag{64}$$

2) Manifold constraint: It's shown that all the poses on the proposed manifold should satisfy the equation constraints as Eq.(33) and Eq.(35), but the IMU motion system is affected by noise in practice. Similar to the approach of using planar constraints in [6], and using quadratic surface constraints in [7], we also convert the B-spline surface constraints into the observation equation with the Gaussian white noise:

$$\boldsymbol{h}_{\boldsymbol{\mathcal{M}}}(\mathcal{X}) = Vec \begin{pmatrix} \boldsymbol{h}_{g_x-3,g_y-3}(\mathcal{X}) & \cdots & \boldsymbol{h}_{g_x-3,g_y}(\mathcal{X}) & \cdots & \boldsymbol{h}_{g_x-3,g_y+3}(\mathcal{X}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{h}_{g_x,g_y-3}(\mathcal{X}) & \cdots & \boldsymbol{h}_{g_x,g_y}(\mathcal{X}) & \cdots & \boldsymbol{h}_{g_x,g_y+3}(\mathcal{X}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{h}_{g_x+3,g_y-3}(\mathcal{X}) & \cdots & \boldsymbol{h}_{g_x+3,g_y}(\mathcal{X}) & \cdots & \boldsymbol{h}_{g_x+3,g_y+3}(\mathcal{X}) \end{pmatrix}, \tag{65}$$

where $\boldsymbol{h}_{\boldsymbol{\mathcal{M}}_{u,v}}$ is short for $\boldsymbol{h}_{u,v}$ and each one has $3 \times n_{u,v}$ equations. We divide Eq.(65) into two parts, one part representing the current manifold which corresponds to the most intermediate ground (the active ground manifold) in the left subfigure of the Fig.1:

$$\begin{aligned} \boldsymbol{h}_{g_x,g_y}(\mathcal{X}) &= \begin{bmatrix} \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}_1}(\mathcal{X}) \\ \vdots \\ \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}_{n_{g_x,g_y}}}(\mathcal{X}) \end{bmatrix}, \forall\ \boldsymbol{\xi}_j \in \mathcal{X}_S \in \mathcal{M}_{g_x,g_y}, \\ \boldsymbol{h}_{u,v,\boldsymbol{\xi}}(\mathcal{X}) &= \begin{bmatrix} (\boldsymbol{y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{x}_w \mathbf{K}_x \mathbf{B}) Vec(\mathbf{C}_{g_x,g_y}) - z_w \\ (\boldsymbol{\partial y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{x}_w \mathbf{K}_x \mathbf{B}) Vec(\mathbf{C}_{g_x,g_y}) + 2\frac{s_1 \sin(\psi)+s_2 \cos(\psi)}{1-s_1^2-s_2^2} \\ (\boldsymbol{y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{\partial x}_w \mathbf{K}_x \mathbf{B}) Vec(\mathbf{C}_{g_x,g_y}) + 2\frac{s_1 \cos(\psi)-s_2 \sin(\psi)}{1-s_1^2-s_2^2} \end{bmatrix} = \mathbf{0}_{3\times 1} + \boldsymbol{n}_{\boldsymbol{\mathcal{M}}}, \end{aligned} \tag{66}$$

where $\boldsymbol{\xi} = \begin{bmatrix} x & y & z & \psi & s_1 & s_2 \end{bmatrix}^\top, \mathbf{C}_{g_x,g_y} = \mathbf{C}, \boldsymbol{n}_{\boldsymbol{\mathcal{M}}} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\boldsymbol{\mathcal{M}}}^2)$, The other part representing the other 48 manifolds (the static ground manifolds) except the most intermediate manifold:

$$\begin{aligned} \boldsymbol{h}_{u,v}(\mathcal{X}) &= \begin{bmatrix} \boldsymbol{h}_{u,v,\boldsymbol{\xi}_1}(\mathcal{X}) \\ \vdots \\ \boldsymbol{h}_{u,v,\boldsymbol{\xi}_{n_{u,v}}}(\mathcal{X}) \end{bmatrix}, \forall\ \boldsymbol{\xi}_j \in \mathcal{M}_{u,v}, \begin{Bmatrix} \forall u \in \{g_x, g_x \pm 1, g_x \pm 2, g_x \pm 3\} \\ \forall v \in \{g_y, g_y \pm 1, g_y \pm 2, g_y \pm 3\} \end{Bmatrix} \Big| \{u, v\} \neq \{g_x, g_y\} \Bigg\}, \\ \boldsymbol{h}_{u,v,\boldsymbol{\xi}}(\mathcal{X}) &= \begin{bmatrix} (\boldsymbol{y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{x}_w \mathbf{K}_x \mathbf{B}) Vec(\mathbf{C}_{u,v}) - z_w \\ (\boldsymbol{\partial y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{x}_w \mathbf{K}_x \mathbf{B}) Vec(\mathbf{C}_{u,v}) + 2\frac{s_1 \sin(\psi)+s_2 \cos(\psi)}{1-s_1^2-s_2^2} \\ (\boldsymbol{y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{\partial x}_w \mathbf{K}_x \mathbf{B}) Vec(\mathbf{C}_{u,v}) + 2\frac{s_1 \cos(\psi)-s_2 \sin(\psi)}{1-s_1^2-s_2^2} \end{bmatrix} = \mathbf{0}_{3\times 1} + \boldsymbol{n}_{\boldsymbol{\mathcal{M}}} \\ &= \begin{bmatrix} \boldsymbol{y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{x}_w \mathbf{K}_x \mathbf{B} \\ \boldsymbol{\partial y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{x}_w \mathbf{K}_x \mathbf{B} \\ \boldsymbol{y}_w \mathbf{K}_y \mathbf{B} \otimes \boldsymbol{\partial x}_w \mathbf{K}_x \mathbf{B} \end{bmatrix} Vec(\mathbf{C}_{u,v}) + \begin{bmatrix} -z_w \\ 2\frac{s_1 \sin(\psi)+s_2 \cos(\psi)}{1-s_1^2-s_2^2} \\ 2\frac{s_1 \cos(\psi)-s_2 \sin(\psi)}{1-s_1^2-s_2^2} \end{bmatrix}, \end{aligned} \tag{67}$$

where

$$Vec(\mathbf{C}_{u,v}) = \mathbf{M}_S(u,v) Vec(\mathbf{C}_{g_x,g_y}) + \mathbf{M}_N(u,v) Vec(\mathbf{C}_{u,v}), \tag{68}$$

$\mathbf{M}_S(u,v)$ is the construction matrix that selects the intersection control points of $\mathbf{C}_{g_x,g_y}$ and $\mathbf{C}_{g_x,g_y}$ in $\mathbf{C}_{g_x,g_y}$ and sorts them in the order of the vector $Vec(\mathbf{C}_{u,v})$, and $\mathbf{M}_N(u,v)$ neglects the intersection control points and keeps the original order.

Note that, all vectors and matrices in Eq.(66) and Eq.(67) are defined in the same way as in the Sec.I-E. Among the 7×7 grids (namely manifolds) shown in the left subfigure of the Fig.1, only the current sliding window poses (the active state $\mathcal{X}_S = \{\boldsymbol{\xi} \mid \boldsymbol{\xi} \in \mathcal{M}_{g_x,g_y}\}$, which locate in the active ground manifold $\mathcal{M}_{g_x,g_y}$) and the current control mesh (the active control points in $\mathbf{C} = \mathbf{C}_{g_x,g_y}$, which shapes $\mathcal{M}_{g_x,g_y}$) are estimated. And the poses on the other 48 manifolds (the static ground manifolds $\{\mathcal{M}_{u,v} \mid u \neq g_x, v \neq g_y\}$) and the other 64 control points (the static control points) are utilized to better optimize all active states.

The jacobian of the Eq.(66) with respect to the state $\mathcal{X}$ is as follows:

$$\frac{\partial \boldsymbol{h}_{g_x,g_y}(\mathcal{X})}{\partial \mathcal{X}} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_{g_x,g_y}(\mathcal{X})}{\partial \mathcal{X}_I} & \frac{\partial \boldsymbol{h}_{g_x,g_y}(\mathcal{X})}{\partial \mathcal{X}_S} & \frac{\partial \boldsymbol{h}_{g_x,g_y}(\mathcal{X})}{\partial \boldsymbol{c}} \end{bmatrix}_{3n_{g_x,g_y}\times(9+6n_{g_x,g_y}+16)}, \tag{69a}$$

$$\frac{\partial \boldsymbol{h}_{g_x,g_y}(\mathcal{X})}{\partial \mathcal{X}_I} = \mathbf{0}_{3n_{g_x,g_y}\times 9}, \tag{69b}$$

$$\frac{\partial \boldsymbol{h}_{g_x,g_y}(\mathcal{X})}{\partial \mathcal{X}_S} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}_1}(\mathcal{X})}{\partial \boldsymbol{\xi}_1} & \mathbf{0}_{3\times 6} & \cdots & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{3\times 6} & \frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}_2}(\mathcal{X})}{\partial \boldsymbol{\xi}_2} & \cdots & \mathbf{0}_{3\times 6} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{3\times 6} & \mathbf{0}_{3\times 6} & \cdots & \frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}_{n_{g_x,g_y}}}(\mathcal{X})}{\partial \boldsymbol{\xi}_{n_{g_x,g_y}}} \end{bmatrix}_{3n_{g_x,g_y}\times 6n_{g_x,g_y}}, \tag{69c}$$

$$\frac{\partial \boldsymbol{h}_{g_x,g_y}(\mathcal{X})}{\partial \boldsymbol{c}} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}_1}(\mathcal{X})}{\partial \boldsymbol{c}} \\ \vdots \\ \frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}_{n_{u,v}}}(\mathcal{X})}{\partial \boldsymbol{c}} \end{bmatrix}_{3n_{g_x,g_y}\times 16}, \tag{69d}$$

$$\frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}}(\mathcal{X})}{\partial \boldsymbol{c}} = \begin{bmatrix} \boldsymbol{y}_w\mathbf{K}_y\mathbf{B}\otimes \boldsymbol{x}_w\mathbf{K}_x\mathbf{B} \\ \boldsymbol{\partial y}_w\mathbf{K}_y\mathbf{B}\otimes \boldsymbol{x}_w\mathbf{K}_x\mathbf{B} \\ \boldsymbol{y}_w\mathbf{K}_y\mathbf{B}\otimes \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B} \end{bmatrix}_{3\times 16}, \tag{69e}$$

$$\frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}}(\mathcal{X})}{\partial \boldsymbol{\xi}} = \begin{bmatrix} \frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}}(\mathcal{X})}{\partial \boldsymbol{p}} & \frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}}(\mathcal{X})}{\partial \psi} & \frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}}(\mathcal{X})}{\partial \boldsymbol{s}} \end{bmatrix}_{3\times 6}, \tag{69f}$$

$$\frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}}(\mathcal{X})}{\partial \boldsymbol{p}} = \begin{bmatrix} \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{y}_w^\top & \boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & -1 \\ \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & \boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial^2 y}_w^\top & 0 \\ \boldsymbol{\partial^2 x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{y}_w^\top & \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & 0 \end{bmatrix}, \tag{69g}$$

$$\frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}}(\mathcal{X})}{\partial \psi} = \begin{bmatrix} \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{y}_w^\top & \boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & -1 \\ \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & \boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial^2 y}_w^\top & 0 \\ \boldsymbol{\partial^2 x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{y}_w^\top & \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & 0 \end{bmatrix} \frac{\partial \mathbf{R}\boldsymbol{t}}{\partial \psi} + 2\begin{bmatrix} 0 \\ \frac{s_1\cos\psi - s_2\sin\psi}{1-s_1^2-s_2^2} \\ \frac{-s_1\sin\psi - s_2\cos\psi}{1-s_1^2-s_2^2} \end{bmatrix}, \tag{69h}$$

$$\frac{\partial \boldsymbol{h}_{g_x,g_y,\boldsymbol{\xi}}(\mathcal{X})}{\partial \boldsymbol{s}} = \begin{bmatrix} \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{y}_w^\top & \boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & -1 \\ \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & \boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial^2 y}_w^\top & 0 \\ \boldsymbol{\partial^2 x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{y}_w^\top & \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{C}\mathbf{B}^\top\mathbf{K}_y^\top\boldsymbol{\partial y}_w^\top & 0 \end{bmatrix} \frac{\partial \mathbf{R}\boldsymbol{t}}{\partial \boldsymbol{s}} \tag{69i}$$

$$+ 2\begin{bmatrix} 0 & 0 \\ \frac{(1+s_1^2-s_2^2)\sin\psi + 2s_1s_2\cos\psi}{(1-s_1^2-s_2^2)^2} & \frac{(1-s_1^2+s_2^2)\cos\psi + 2s_1s_2\sin\psi}{(1-s_1^2-s_2^2)^2} \\ \frac{(1+s_1^2-s_2^2)\sin\psi - 2s_1s_2\cos\psi}{(1-s_1^2-s_2^2)^2} & \frac{(-1+s_1^2-s_2^2)\cos\psi + 2s_1s_2\sin\psi}{(1-s_1^2-s_2^2)^2} \end{bmatrix}, \tag{69j}$$

$$\frac{\partial \mathbf{R}\boldsymbol{t}}{\partial \psi} = \begin{bmatrix} -\sin\psi & \cos\psi & 0 \\ \cos\psi & -\sin\psi & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{R}_\phi \boldsymbol{t}, \tag{69k}$$

$$\frac{\partial \mathbf{R}\boldsymbol{t}}{\partial s_1} = \frac{\mathbf{R}_\psi}{(1+s_1^2+s_2^2)^2} \begin{bmatrix} -4s_1(1+s_2^2) & -2s_2(1-s_1^2+s_2^2) & 2(1-s_1^2+s_2^2) \\ -2s_2(1-s_1^2+s_2^2) & 4s_1s_2^2 & -4s_1s_2 \\ -2(1-s_1^2+s_2^2) & 4s_1s_2 & -4s_1 \end{bmatrix} \boldsymbol{t}, \tag{69l}$$

$$\frac{\partial \mathbf{R}\boldsymbol{t}}{\partial s_2} = \frac{\mathbf{R}_\psi}{(1+s_1^2+s_2^2)^2} \begin{bmatrix} 4s_1^2s_2 & -2s_2(1+s_1^2-s_2^2) & -4s_1s_2 \\ -2s_2(1+s_1^2-s_2^2) & -4s_2(1+s_1^2) & 2(1+s_1^2-s_2^2) \\ 4s_1s_2 & -2(1+s_1^2-s_2^2) & -4s_2 \end{bmatrix} \boldsymbol{t}, \tag{69m}$$

where $\{\mathbf{R}, \mathbf{R}_\psi, \mathbf{R}_\phi, \boldsymbol{t}\}$ are the abbreviations of $\{{}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}, {}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}_\psi, {}^{\mathcal{G}}_{\mathcal{I}}\mathbf{R}_\phi, {}^{\mathcal{I}}\boldsymbol{t}_w\}$.

The jacobian of the Eq.(67) with respect to the state $\mathcal{X}$ is as follows:

$$
\begin{aligned}
\frac{\partial \boldsymbol{h}_{u,v}(\mathcal{X})}{\partial \mathcal{X}} &= \begin{bmatrix} \frac{\partial \boldsymbol{h}_{u,v}(\mathcal{X})}{\partial \mathcal{X}_I} & \frac{\partial \boldsymbol{h}_{u,v}(\mathcal{X})}{\partial \mathcal{X}_S} & \frac{\partial \boldsymbol{h}_{u,v}(\mathcal{X})}{\partial \boldsymbol{c}} \end{bmatrix}_{3n_{u,v}\times(9+6n_{u,v}+16)}, \\
\frac{\partial \boldsymbol{h}_{u,v}(\mathcal{X})}{\partial \mathcal{X}_I} &= \mathbf{0}_{3n_{u,v}\times 9}, \\
\frac{\partial \boldsymbol{h}_{u,v}(\mathcal{X})}{\partial \mathcal{X}_S} &= \mathbf{0}_{3n_{u,v}\times 6n_{u,v}}, \\
\frac{\partial \boldsymbol{h}_{u,v}(\mathcal{X})}{\partial \boldsymbol{c}} &= \begin{bmatrix} \frac{\partial \boldsymbol{h}_{u,v,\boldsymbol{\xi}_1}(\mathcal{X})}{\partial \boldsymbol{c}} \\ \vdots \\ \frac{\partial \boldsymbol{h}_{u,v,\boldsymbol{\xi}_{n_{u,v}}}(\mathcal{X})}{\partial \boldsymbol{c}} \end{bmatrix}_{3n_{u,v}\times 16}, \\
\frac{\partial \boldsymbol{h}_{u,v,\boldsymbol{\xi}}(\mathcal{X})}{\partial \boldsymbol{c}} &= \begin{bmatrix} \boldsymbol{y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{M}_S(u,v) \\ \boldsymbol{\partial y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{x}_w\mathbf{K}_x\mathbf{B}\mathbf{M}_S(u,v) \\ \boldsymbol{y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B}\mathbf{M}_S(u,v) \end{bmatrix}_{3\times 16}.
\end{aligned}
\tag{70}
$$

D. Extended Kalman Filtering Process

1) Initialization: At the time of initialization, there is no a priori information about the ground manifold, so the space-based sliding window state $\mathcal{X}_S$ and the space-based sliding control vector $\boldsymbol{c}$ do not exist in the initial state vector. We initialize the IMU state $\mathcal{X}_I$ and its covariance of the proposed filter and other parameters as in TABLE.I.

	${}^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}}$	${}^{\mathcal{I}}\boldsymbol{v}_{\mathcal{I}}$	${}^{\mathcal{G}}_{\mathcal{I}}\psi$	${}^{\mathcal{G}}_{\mathcal{I}}\boldsymbol{s}$	$\boldsymbol{c}$	d	$\boldsymbol{n}_{\boldsymbol{\mathcal{M}}}$
val	${}^{\mathcal{G}}\boldsymbol{p}_{\mathcal{I}_0}$	${}^{\mathcal{I}_0}\boldsymbol{v}_{\mathcal{I}_0}$	${}^{\mathcal{G}}_{\mathcal{I}_0}\psi$	${}^{\mathcal{G}}_{\mathcal{I}_0}\boldsymbol{s}$	LSM	5	\
diag(cov)	1e−8	1e−4	1e−6	1e−4	1e−4	\	1e−8

TABLE I: Initialization

When the robot enters the second ground manifold (before manifold constraint update), the control vector $\boldsymbol{c}$ is solved by the least squares method as follows:

$$
\begin{aligned}
&\boldsymbol{c} = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\boldsymbol{b}, \quad \forall\ \boldsymbol{\xi}_j \in \mathcal{X}_S \in \mathcal{M}_{g_x,g_y}, \\
&\mathbf{A} = \begin{bmatrix} \mathbf{A}_{\xi_1} \\ \mathbf{A}_{\xi_2} \\ \vdots \\ \mathbf{A}_{\xi_n} \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_{\boldsymbol{\xi}_1} \\ \boldsymbol{b}_{\boldsymbol{\xi}_2} \\ \vdots \\ \boldsymbol{b}_{\boldsymbol{\xi}_n} \end{bmatrix}, \\
&\mathbf{A}_\xi = \begin{bmatrix} \boldsymbol{y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{x}_w\mathbf{K}_x\mathbf{B} \\ \boldsymbol{\partial y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{x}_w\mathbf{K}_x\mathbf{B} \\ \boldsymbol{y}_w\mathbf{K}_y\mathbf{B} \otimes \boldsymbol{\partial x}_w\mathbf{K}_x\mathbf{B} \end{bmatrix}, \boldsymbol{b}_{\boldsymbol{\xi}} = \begin{bmatrix} z_w \\ -2\frac{s_1\sin(\psi)+s_2\cos(\psi)}{1-s_1^2-s_2^2} \\ 2\frac{s_1\cos(\psi)-s_2\sin(\psi)}{1-s_1^2-s_2^2} \end{bmatrix}.
\end{aligned}
\tag{71}
$$

2) State propagation: We can obtain the discrete state propagation equations from the derivations in the Sec.III-B:

$$\mathcal{X}_{\mathcal{I}_{k+1}} = \mathbf{F}(\mathcal{X}_{\mathcal{I}_k}, \boldsymbol{u}_k), \tag{72a}$$

$$\mathbf{P}_{k+1} = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^\top + \mathbf{B}_k \mathbf{W} \mathbf{B}_k^\top, \tag{72b}$$

$$\mathbf{A}_k = \begin{bmatrix} \mathbf{F}_\mathcal{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{B}_k = \begin{bmatrix} \mathbf{F}_{\boldsymbol{n}} \\ \mathbf{0} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{a}}^2 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\boldsymbol{\omega}}^2 \end{bmatrix}, \tag{72c}$$

where $\mathbf{F}_\mathcal{I}$ and $\mathbf{F}_{\boldsymbol{n}}$ are derived in Eq.(61).

3) State augmentation: In the proposed system, the dimension of the sliding window state $\mathcal{X}_S$ grows as the ground robot runs longer in a ground manifold, state augmentations are performed at the process model frequency. During an augmentation step, the state dimension is incremented with cloning:

$$\mathbf{P}_{k+1} = \bar{\mathbf{A}}_k \mathbf{P}_k \bar{\mathbf{A}}_k^\top + \bar{\mathbf{B}}_k \mathbf{W} \bar{\mathbf{B}}_k^\top, \tag{73a}$$

$$\bar{\mathbf{A}}_k = \begin{bmatrix} \mathbf{F}_\mathcal{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{F}_\mathcal{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \bar{\mathbf{B}}_k = \begin{bmatrix} \mathbf{F}_{\boldsymbol{n}} \\ \mathbf{0} \\ \mathbf{F}_{\boldsymbol{n}} \\ \mathbf{0} \end{bmatrix}, \tag{73b}$$

where $\bar{\mathbf{A}}_k$ and $\bar{\mathbf{B}}_k$ are copy and state augmentation operations. After that, the dimension of the sliding window state $\mathcal{X}_S$ increases by 6. Once the robot runs into the next ground manifold, all the old sliding window state in the last manifold will be pruned in the marginalization step.

4) Measurement update: There are two observation updates in the proposed filtering, one is a network-corrected IMU velocity observation with a frequency of $100Hz$, and the other is based on a pseudo-measurement of the ground manifold with an update frequency that depends on the fixed ground interval parameter d and the robot's velocity.

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top \left(\mathbf{H}\mathbf{P}\mathbf{H}^\top + \mathbf{R}\right)^{-1}, \tag{74a}$$

$$\mathcal{X} \leftarrow \mathcal{X} + \mathbf{K}\big(\boldsymbol{z} - \boldsymbol{h}(\mathcal{X})\big), \tag{74b}$$

$$\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}(\mathbf{I} - \mathbf{K}\mathbf{H})^\top + \mathbf{K}\mathbf{R}\mathbf{K}^\top, \tag{74c}$$

where $\mathbf{H}$, $\mathbf{R}$, $\boldsymbol{h}(\mathcal{X})$ and $\boldsymbol{z}$ denote the Jacobi matrix, measurement covariance, measurement model and measurement value provided by Sec.III-C, respectively.

5) Slinding and Marginalization: As shown in the Fig.1, when the ground travels to the next ground manifold, the trajectory poses that locate in the previous ground and the control points that are not longer shaping the previous ground manifold are both removed from the active state and dropped into the static state. This operation is similar to the marginalization in the time-sliding window based optimization in [8], and we use Schur complement to perform the Jacobi matrix removal operation. As mentioned before, the poses and control points in the static state are used to optimize the latest manifold (which is shaped by the active control mesh). Only when the static state cannot affect the latest manifold are they actually marginalized, that is, dropped into the marginalized state.

REFERENCES

[1] J. Svacha, G. Loianno, and V. Kumar, "Inertial yaw-independent velocity and attitude estimation for high-speed quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1109–1116, 2019.

[2] K. Zhang, C. Jiang, J. Li, S. Yang, T. Ma, C. Xu, and F. Gao, "Dido: Deep inertial quadrotor dynamical odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9083–9090, 2022.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[4] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[5] D. Chen, N. Wang, R. Xu, W. Xie, H. Bao, and G. Zhang, "Rnin-vio: Robust neural inertial navigation aided visual-inertial odometry in challenging scenes," in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2021, pp. 275–283.
[6] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "Vins on wheels," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5155–5162.
[7] M. Zhang, X. Zuo, Y. Chen, Y. Liu, and M. Li, "Pose estimation for ground robots: On manifold representation, integration, reparameterization, and optimization," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1081–1099, 2021.
[8] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.