

---

# Visual-Inertial Localization for Skid-Steering Robots with Kinematic Constraints

Xingxing Zuo - xingxingzuo@zju.edu.cn  
Mingyang Li - mingyangli@alibaba-inc.com  
Guoquan Huang - ghuang@udel.edu

Institute of Cyber-System and Control, Zhejiang University, China  
A.I. Labs, Alibaba Group, China  
Department of Mechanical Engineering, University of Delaware, Delaware, USA

# Contents

<b>1</b>	<b>ICR-based Kinematics of Skid-Steering Robots</b>	<b>1</b>
1.1	Notations . . . . .	1
1.2	Kinematics of Skid-steering Robot . . . . .	1
<b>2</b>	<b>Kinematics-Constrained Visual-Inertial Localization</b>	<b>2</b>
2.1	Image Processing . . . . .	3
2.2	IMU Prediction . . . . .	3
2.3	Motion Manifold Constraints . . . . .	4
2.4	ICR-based Kinematic Constraints . . . . .	4
	<b>References</b>	<b>8</b>

# 1 ICR-based Kinematics of Skid-Steering Robots

## 1.1 Notations

In this technique report, we consider a robotic platform navigating with respect to a global reference frame,  $\{\mathbf{G}\}$ . The platform is equipped with a camera, an IMU, and wheel odometers, whose frames are denoted by  $\{\mathbf{C}\}$ ,  $\{\mathbf{I}\}$ ,  $\{\mathbf{O}\}$  respectively. To present transformation, we use  ${}^{\mathbf{A}}\mathbf{p}_{\mathbf{B}}$  and  ${}^{\mathbf{A}}\mathbf{R}_{\mathbf{B}}$  to denote position and rotation of frame  $\{\mathbf{B}\}$  with respect to  $\{\mathbf{A}\}$ . We use  $\hat{\mathbf{x}}$  and  $\delta\mathbf{x}$  to represent the current estimated value and error state for variable  $\mathbf{x}$ . Furthermore, we reserve the symbol  $\check{x}$  to denote the inferred measurement mean value of  $x$  throughout this technique report.

## 1.2 Kinematics of Skid-steering Robot

In this work, we employ the ICR parameters [1] to approximately model the kinematics of a skid-steering robot. Specifically, as shown in Fig. 1, we denote  $\mathbf{ICR}_v = (X_v, Y_v)$  the ICR position of the robot frame, and  $\mathbf{ICR}_l = (X_l, Y_l)$  and  $\mathbf{ICR}_r = (X_r, Y_r)$  the ones of the left and right wheels, respectively. The relation between the readings of wheel odometer measurements and the ICR parameters can be derived as follows:

$$\begin{aligned} Y_l &= -\frac{o_l - \mathbf{O}v_x}{\mathbf{O}\omega_z}, \quad Y_r = -\frac{o_r - \mathbf{O}v_x}{\mathbf{O}\omega_z} \\ Y_v &= \frac{\mathbf{O}v_x}{\mathbf{O}\omega_z}, \quad X_v = X_l = X_r = -\frac{\mathbf{O}v_y}{\mathbf{O}\omega_z} \end{aligned} \quad (1)$$

where  $\mathbf{O}\mathbf{v} = [\mathbf{O}v_x, \mathbf{O}v_y, \mathbf{O}v_z]^\top$  is the robot's 3D linear velocity in local odometry frame, and  $\mathbf{O}\omega_z$  denotes the angular velocity along the robot's  $z$  axis, which is the abbreviation of  $\mathbf{O}\omega_z$ . From Eq. 1, the kinematic models of the skid-steering robot can be expressed as:

$$\begin{bmatrix} \mathbf{O}v_x \\ \mathbf{O}v_y \\ \mathbf{O}\omega_z \end{bmatrix} = \frac{1}{\Delta Y} \begin{bmatrix} -Y_r & Y_l \\ X_v & -X_v \\ -1 & 1 \end{bmatrix} \begin{bmatrix} o_l \\ o_r \end{bmatrix} \quad (2)$$

where  $\Delta Y = Y_l - Y_r$ . We can see that after the linear velocities of left and right wheels  $o_l, o_r$  are know, the motion of the robot can be determined by the 3 ICR parameters  $[X_v, Y_l, Y_r]$ .

Moreover, we introduce two additional scale factors,  $[\alpha_l, \alpha_r]$ , to compensate for the possible effects, e.g., due to tire inflation and interface roughness. With the scale factors and Eq. 1, we can express the motion variables as:

$$\begin{bmatrix} \mathbf{O}v_x \\ \mathbf{O}v_y \\ \mathbf{O}\omega_z \end{bmatrix} = g(\boldsymbol{\xi}, o_l, o_r) = \frac{1}{\Delta Y} \begin{bmatrix} -Y_r & Y_l \\ X_v & -X_v \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_l & 0 \\ 0 & \alpha_r \end{bmatrix} \begin{bmatrix} o_l \\ o_r \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{bmatrix} X_v \\ Y_l \\ Y_r \\ \alpha_l \\ \alpha_r \end{bmatrix} \quad (3)$$

where  $\Delta Y = Y_l - Y_r$ , and  $\boldsymbol{\xi}$  is the entire set of kinematic parameters.

Interestingly, as a special configuration when  $\boldsymbol{\xi} = [0, \frac{b}{2}, \frac{-b}{2}, 1, 1]^\top$ , with  $b$  being the distance between left and right wheels, Eq. 3 can be simplified as:

$$\mathbf{O}v_x = \frac{o_l + o_r}{2}, \quad \mathbf{O}\omega_z = \frac{o_r - o_l}{b}, \quad \mathbf{O}v_y = 0 \quad (4)$$

This is the kinematic model for a wheeled robot moving without slippage (e.g., a differential drive robot), and used by most existing work for localizing wheeled robots [2, 3]. However, in the case of skid-steering robots under consideration, if directly applying Eq. 4, the localization accuracy would be significantly degraded. It is important to point out that as  $\xi$  cannot remain constant due to different motions and terrains [1, 4], we will perform online “calibration” to estimate these kinematic parameters along with the navigation states as in [5–7].

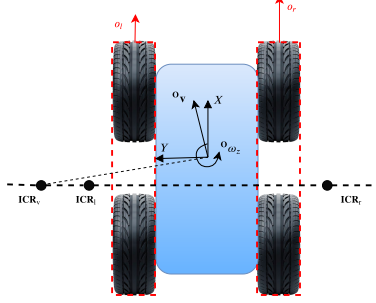


Figure 1: The odometer measurements and the instantaneous center of rotation (ICR) of a skid-steering robot on the ground plane.

## 2 Kinematics-Constrained Visual-Inertial Localization

We develop a window-BA estimator for the proposed kinematics-constrained visual-inertial localization for a skid-steering robot equipped with a camera, an IMU, and wheel encoders. For simplicity, although not necessary, we assume known extrinsic transformations between sensors. At each time step, we optimize the following window of states, whose typically oldest state will be marginalized out when moving to the next window in order to bound computational cost:

$$\mathbf{x} = \{\mathcal{G}\mathcal{T}, \mathbf{G}\mathbf{v}_{\mathbf{I}_k}, \mathbf{b}_a, \mathbf{b}_\omega, \xi, \mathcal{F}, \mathbf{m}\} \quad (5)$$

In the above expression,  $\mathcal{G}\mathcal{T} = \{\mathcal{G}_{\mathbf{O}_{k-s}}\mathbf{T}, \dots, \mathcal{G}_{\mathbf{O}_{k-1}}\mathbf{T}, \mathcal{G}_{\mathbf{O}_k}\mathbf{T}\}$  denotes the cloned poses in the sliding window at time  $\{k-s, \dots, k\}$ .  $\mathcal{G}_{\mathbf{O}_k}\mathbf{T} = \{\mathcal{G}_{\mathbf{O}_k}\mathbf{R}, \mathcal{G}_{\mathbf{O}_k}\mathbf{p}_{\mathbf{O}_k}\}$  represents the 6DOF pose of the robot at time  $k$ . We choose the odometry frame is the base sensor frame and the system is initialized by the initial position of odometer while the direction of  $z$  is aligned with the gravity.  $\mathcal{F}$  contains all the 3D global positions of visual features.  $\mathbf{G}\mathbf{v}_{\mathbf{I}_k}, \mathbf{b}_a, \mathbf{b}_\omega$  are the IMU velocity in global frame, acceleration bias and angular velocity bias, respectively. Note that we estimate online the ICR kinematic parameters  $\xi$  and thus include them in the state as well. Lastly,  $\mathbf{m}$  denotes the parameters related to the motion manifold constraints enforcing local smooth ground planar motion. As illustrated in Fig. 2, the sliding window BA is our estimation engine whose cost function includes the following constraints:

$$\mathcal{C} = \mathcal{C}_{prior} + \mathcal{C}_{proj} + \mathcal{C}_I + \mathcal{C}_O + \mathcal{C}_{manifold} \quad (6)$$

which includes the prior of the states remaining in the current sliding window after marginalization [8], the projection error of visual features, the IMU integration constraints [7, 14], the odometry-induced kinematic constraints, and the motion manifold constraints.

In particular, the prior factor related with the remaining states  $\mathbf{x}_r$  in current sliding window is associated with the measurements incident to the marginalized states

$$\mathcal{C}_{prior}(\mathbf{x}_r) = \frac{1}{2} \|\mathbf{x}_r \boxminus \hat{\mathbf{x}}_r\|_{\Lambda_{marg}}^2 + \mathbf{g}_{marg}^\top (\mathbf{x}_r \boxminus \hat{\mathbf{x}}_r) \quad (7)$$

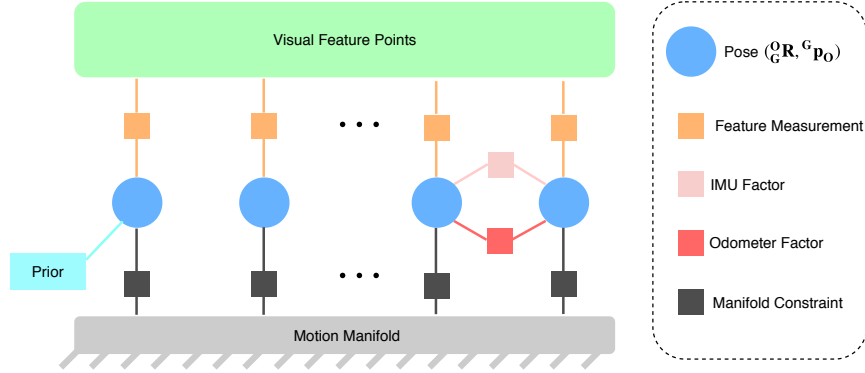


Figure 2: In the proposed kinematics-constrained visual-inertial localization for skid-steering robots, five different constraints are used in the sliding-window BA: A prior encapsulates the information about the current states due to marginalization of states and measurements; Visual feature measurements connect the feature points in the map and the robot pose at the time when the image was recorded; IMU integration factor summarizes the sequential IMU raw measurements between the two images (keyframes); Odometry-induced kinematic factor summarizes the sequential odometer measurements between the two images; Motion manifold constraints enforce local smooth planar motions. Note that the IMU factor and the Odometry-induced kinematic factor only existing in the newest keyframe and the second newest keyframe.

where  $\hat{\mathbf{x}}_r$  is the current state estimate at the time of marginalization, and  $\Lambda_{margin}$  and  $\mathbf{g}_{margin}$  are the marginalized Hessian and gradient [8], respectively. The boxminus  $\boxminus$  denotes the generalized minus operation, since we need perform computations operate on manifold.

## 2.1 Image Processing

Keyframe policy are used in our framework for computational saving. Only some keyframes will be optimized in the back-end. Whereas, the non-keyframe will be dropped immediately without any extra operations unlike existing method [9, 10] which also need to extract features and analyses the features distribution for keyframe selection. Once a new image is coming, we use the odometer pose prediction ( see Sec. 2.4) for keyframe selections. We use a simple heuristic for keyframe selection: the odometer prediction has a translation or rotation over a certain threshold (in all the experiments, 0.2 meter and 3 degrees are the criteria).

Corner feature points are extracted in a fast way [11] and tacked by KLT optical flow algorithm [12]. The tracked features are further checked by a RANSAC scheme with a fundamental matrix model [13]. For a 3D feature  $\mathbf{G}\mathbf{p}_{fj} \in \mathcal{F}$  in the global frame has an observation  $\mathbf{z}_{i,j}$  in  $C_i$ th camera keyframe, the reprojection error is

$$c_{proj}(\mathbf{G}_i\mathbf{R}, \mathbf{G}\mathbf{p}_{O_i}, \mathbf{G}\mathbf{p}_{fj}) = \|\mathbf{z}_{i,j} - \pi(\mathbf{G}_i\mathbf{R}, \mathbf{G}\mathbf{p}_{C_i}, \mathbf{G}\mathbf{p}_{fj})\|_{\Lambda_{proj}}^2 \quad (8)$$

$$\mathbf{G}_i\mathbf{R} = \mathbf{G}_i\mathbf{R}_C\mathbf{G}_C\mathbf{R}, \quad \mathbf{G}\mathbf{p}_{C_i} = \mathbf{G}\mathbf{p}_{O_i} + \mathbf{G}_i\mathbf{R}^O\mathbf{p}_C$$

. In the above expression,  $\pi(\cdot)$  denote the projection function, and  $\Lambda_{proj}$  represent the inverse covariance of the observation  $\mathbf{z}_{i,j}$ .

## 2.2 IMU Prediction

The raw measurements of IMU is the acceleration and angular velocity measurements in local IMU frame.:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega}_I + \mathbf{b}_\omega + \mathbf{n}_\omega \quad (9a)$$

$$\mathbf{a}_m = \mathbf{a}_I + \mathbf{I}_G\mathbf{R}\mathbf{g} + \mathbf{b}_a + \mathbf{n}_a \quad (9b)$$

where  $\mathbf{G}g$  is the global gravity, and the raw measurements are deteriorated by the time-varying bias  $\mathbf{b}_\omega, \mathbf{b}_a$ , and white noises  $\mathbf{n}_\omega, \mathbf{n}_a$ . We predict the pose at current key frame time  $t(k)$  by intermediate sequential IMU measurements  $\mathbf{i}_0, \mathbf{i}_1, \dots, \mathbf{i}_m \in \mathcal{I}_m$  between  $t(k-1)$  and  $t(k)$ . The integration of IMU are omitted here due to limited spaces, and please refer to [7, 14] for details. After integrating all the intermediate IMU measurements, we obtain the state prediction  $\hat{\mathbf{x}}_{I_k} = [\mathbf{G}\hat{\mathbf{p}}_{\mathbf{O}_k}, \mathbf{G}\hat{\mathbf{O}}_k \hat{\mathbf{R}}, \mathbf{G}\hat{\mathbf{v}}_{I_k}, \hat{\mathbf{b}}_a, \hat{\mathbf{b}}_\omega] = f(\hat{\mathbf{x}}_{I_{k-1}}, \mathcal{I}_m)$  at time  $t_k$ , where  $\hat{\mathbf{b}}_a$ , and  $\hat{\mathbf{b}}_\omega$  do not evolve between two keyframes. We can write the IMU constraints here:

$$\mathcal{C}_I(\mathbf{G}\mathbf{p}_{\mathbf{O}_k}, \mathbf{G}\mathbf{O}_k \mathbf{R}, \mathbf{G}\mathbf{p}_{\mathbf{O}_{k-1}}, \mathbf{G}\mathbf{O}_{k-1} \mathbf{R}, \mathbf{G}\mathbf{v}_{I_k}, \mathbf{b}_a, \mathbf{b}_\omega) = \|\hat{\mathbf{x}}_{I_k} \ominus \mathbf{x}_{I_k}\|_{\Lambda_{IMU}}^2 \quad (10)$$

### 2.3 Motion Manifold Constraints

The manifold constraints includes two parts related to the position and rotation of the robot respectively. Firstly, as the skid-steer robot navigates on ground surface, the positions  $\mathbf{G}\mathbf{p}_{\mathbf{O}}$  during a period of time can be approximated by the quadratic polynomial [15] (in our implementation, this holds in the whole sliding window):

$$M_p(\mathbf{G}\mathbf{p}_{\mathbf{O}}) = \frac{1}{2} \begin{bmatrix} \mathbf{G}p_{\mathbf{O}x} \\ \mathbf{G}p_{\mathbf{O}y} \end{bmatrix}^\top \mathbf{A} \begin{bmatrix} \mathbf{G}p_{\mathbf{O}x} \\ \mathbf{G}p_{\mathbf{O}y} \end{bmatrix} + \mathbf{B}^\top \begin{bmatrix} \mathbf{G}p_{\mathbf{O}x} \\ \mathbf{G}p_{\mathbf{O}y} \end{bmatrix} + \mathbf{G}p_{\mathbf{O}z} + c \quad (11)$$

where

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 \\ a_2 & a_3 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (12)$$

. The manifold parameters are

$$\mathbf{m} = [a_1, a_2, a_3, b_1, b_2, c]^\top$$

. And all the robot position insides the sliding window should satisfy Eq. 11 .

In addition, the rotations of the robot  $\mathbf{G}\mathbf{R}_{\mathbf{O}}$  should satisfy the following constraints

$$M_r(\mathbf{G}\mathbf{R}, \mathbf{G}\mathbf{p}_{\mathbf{O}}) = \|\llbracket \mathbf{G}\mathbf{R} \mathbf{e}_3 \rrbracket_{12} * \frac{\partial M_p}{\partial \mathbf{G}\mathbf{p}_{\mathbf{O}}}\|_{\Lambda_{mr}}^2 \quad (13)$$

$\llbracket \mathbf{v} \rrbracket_{12}$  denotes the first and second rows of the symmetric matrix of the 3D vector  $\mathbf{v}$ . The above expression ensures the fact that the roll and pitch of the ground robot should be consistent with the normal vector of the motion manifold (ground surface). In general, the motion manifold constraints, related with all the poses inside the sliding window, are

$$\mathcal{C}_{manifold}(\mathbf{G}\mathcal{T}, \mathbf{m}) = \sum_{i=k-s}^k \|M_p(\mathbf{G}\mathbf{p}_{\mathbf{O}_i})\|_{\Lambda_{mp}}^2 + M_r(\mathbf{G}\mathbf{O}_i \mathbf{R}, \mathbf{G}\mathbf{p}_{\mathbf{O}_i}) \quad (14)$$

### 2.4 ICR-based Kinematic Constraints

We now derive the ICR-based kinematic constraints based on the wheel encoders' measurements of the skid-steering robot. Specifically, by assuming the supporting manifold of the robot is locally planar between  $t_{k-1}$  and  $t_k$ , the local linear and angular velocities,  $\mathbf{O}^{(t)}\mathbf{v}$  and  $\mathbf{O}^{(t)}\boldsymbol{\omega}$ , are a function of the wheel encoders' measurements of the left and right wheels  $o_{lm}(t)$  and  $o_{rm}(t)$  as well as the ICR kinematic parameters  $\boldsymbol{\xi}$  [see (3)]:

$$\begin{aligned} \left[ \mathbf{O}^{(t)}\mathbf{v}^T, \mathbf{O}^{(t)}\boldsymbol{\omega}^T \right]^\top &= \mathbf{\Pi} g(\boldsymbol{\xi}(t), o_l(t), o_r(t)) \\ &= \mathbf{\Pi} g(\boldsymbol{\xi}(t), o_{lm}(t) - n_l(t), o_{rm}(t) - n_r(t)) \end{aligned} \quad (15)$$

where  $\mathbf{\Pi} = \begin{bmatrix} \mathbf{e}_1^T & \mathbf{e}_2^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{e}_3^T \end{bmatrix}^T$  is the selection matrix with  $\mathbf{e}_i$  being a  $3 \times 1$  unit vector with the  $i$ th element of 1,  $o_{lm}(t)$  and  $o_{rm}(t)$  are odometer readings of left and right wheels, respectively,  $n_l(t)$  and  $n_r(t)$  are the odometry noise modeled as zero-mean white Gaussian. Once the instantaneous local velocities of the robot are available, with the initial conditions  $\mathbf{O}_{(t)}^{\mathbf{O}(t)} \mathbf{R} \big|_{t=t_{k-1}} = \mathbf{I}_{3 \times 3}$  and  $\mathbf{O}_{(t)}^{\mathbf{O}(t)} \mathbf{p}_{\mathbf{O}(t)} \big|_{t=t_{k-1}} = \mathbf{0}_{3 \times 1}$ , we can integrate the following differential equations in the time interval  $t \in [t_{k-1}, t_k]$ :

$$\begin{aligned} \mathbf{O}_{(t)}^{\mathbf{O}(t)} \dot{\mathbf{R}} &= \mathbf{O}_{(t)}^{\mathbf{O}(t)} \mathbf{R} \cdot \lfloor \mathbf{O}_{(t)} \boldsymbol{\omega} \rfloor \\ \mathbf{O}_{k-1} \dot{\mathbf{p}}_{\mathbf{O}(t)} &= \mathbf{O}_{k-1} \mathbf{v}_{\mathbf{O}(t)} = \mathbf{O}_{(t)}^{\mathbf{O}(t)} \mathbf{R} \cdot \mathbf{O}_{(t)} \mathbf{v} \end{aligned} \quad (16)$$

This integration will result in the relative pose  $\{\mathbf{O}_{k-1} \mathbf{p}_{\mathbf{O}_k}, \mathbf{O}_{k-1} \mathbf{R}\}$ , which is then used to propagate the global pose from  $t_k$  to  $t_{k+1}$ :

$$\mathbf{G} \mathbf{p}_{\mathbf{O}_k} = \mathbf{G} \mathbf{p}_{\mathbf{O}_{k-1}} + \mathbf{G}_{\mathbf{O}_{k-1}} \mathbf{R} \cdot \mathbf{O}_{k-1} \mathbf{p}_{\mathbf{O}_k} \quad (17)$$

$$\mathbf{G}_{\mathbf{O}_k} \mathbf{R} = \mathbf{G}_{\mathbf{O}_{k-1}} \mathbf{R} \cdot \mathbf{O}_{k-1} \mathbf{R} \quad (18)$$

where  $\mathbf{O}_{k-1} \mathbf{p}_{\mathbf{O}_k}$  and  $\mathbf{O}_{k-1} \mathbf{R}$  represents the relative motion between timestamps. Additionally, we model the ICR kinematic parameter  $\boldsymbol{\xi}$  as a random walk to capture its time-varying characteristics:

$$\dot{\boldsymbol{\xi}}(t) = \mathbf{n}_{\boldsymbol{\xi}}(t) \quad (19)$$

where  $\mathbf{n}_{\boldsymbol{\xi}}$  is zero-mean white Gaussian noise.

Based on the ICR-based kinematic model (17) and (19), we predict the pose and kinematic parameter at the newest keyframe time  $t_k$ ,  $\hat{\mathbf{x}}_{\mathbf{O}_k} = \begin{bmatrix} \mathbf{G}_{\mathbf{O}_k} \hat{\mathbf{R}}, \mathbf{G} \hat{\mathbf{p}}_{\mathbf{O}_k}, \hat{\boldsymbol{\xi}} \end{bmatrix} = f(\hat{\mathbf{x}}_{\mathbf{O}_k}, \mathcal{O}_m)$ , by integrating all the intermediate odometry measurements  $\mathbf{o}_0, \mathbf{o}_1, \dots, \mathbf{o}_m \in \mathcal{O}_m$  specially,  $t(0) = t(k-1), t(m) = t(k)$ . And the raw measurements of wheel odometry are the linear velocities of left and right wheels, such as  $\mathbf{o}_0 = [o_{lm0}, o_{rm0}]^\top \in \mathbb{R}^{2 \times 1}$ . As a result, the odometer-induced kinematic constraint can be generically written in the following form:

$$\mathcal{C}_{\mathbf{O}}(\mathbf{G} \mathbf{p}_{\mathbf{O}_k}, \mathbf{G}_{\mathbf{O}_k} \mathbf{R}, \mathbf{G} \mathbf{p}_{\mathbf{O}_{k-1}}, \mathbf{G}_{\mathbf{O}_{k-1}} \mathbf{R}, \boldsymbol{\xi}) = \|\hat{\mathbf{x}}_{\mathbf{O}_k} \boxminus \mathbf{x}_{\mathbf{O}_k}\|_{\boldsymbol{\Lambda}_{odom}}^2 \quad (20)$$

where  $\boldsymbol{\Lambda}_{odom}$  represents the inverse covariance (information) obtained via covariance propagation.

Here we illustrate the odometer integration in detail. If we use midpoint method for numerical integration to compute  $\mathbf{G} \mathbf{p}_{\mathbf{O}_k}, \mathbf{G}_{\mathbf{O}_k} \mathbf{R}$  in Eq. 17 by  $\mathcal{O}_m$ . Starting from  $t(0)$ , the pose prediction at  $t(1)$ , can be obtained from the integration in the midpoint method:

$$\mathbf{G} \mathbf{p}_{\mathbf{O}_1} = \mathbf{G} \mathbf{p}_{\mathbf{O}_0} + \mathbf{G}_{\mathbf{O}_0} \mathbf{R} \mathbf{O}_0 \mathbf{p}_{\mathbf{O}_1}, \quad \mathbf{O}_0 \mathbf{p}_{\mathbf{O}_1} = \frac{1}{2} \Delta t \left( \mathbf{O}_0 \mathbf{v} + \mathbf{O}_1 \mathbf{R} \mathbf{O}_1 \mathbf{v} \right) \quad (21a)$$

$$\mathbf{G}_{\mathbf{O}_1} \mathbf{R} = \mathbf{G}_{\mathbf{O}_0} \mathbf{R} \mathbf{O}_1 \mathbf{R}, \quad \mathbf{O}_1 \mathbf{R} = \exp\{\bar{\boldsymbol{\omega}} \Delta t\}, \quad \bar{\boldsymbol{\omega}} = \left[ 0 \quad 0 \quad \frac{\mathbf{O}_0 \boldsymbol{\omega}_z + \mathbf{O}_1 \boldsymbol{\omega}_z}{2} \right]^\top \quad (21b)$$

In the above expressions, the linear velocities  $\mathbf{O}_0 \mathbf{v}, \mathbf{O}_1 \mathbf{v}$  and the angular velocities  $\mathbf{O}_0 \boldsymbol{\omega}_z, \mathbf{O}_1 \boldsymbol{\omega}_z$  at time  $t(0), t(1)$  can be computed from the raw measurement of wheel odometry  $\mathbf{o}_0, \mathbf{o}_1$  by the ICR model in Eq. 3. Note that the vertical linear velocity  $\mathbf{O}_0 \mathbf{v}, \mathbf{O}_1 \mathbf{v}$  remains zero here.

Besides the poses, the 5 ICR parameters including locations  $\mathbf{ICR} = [X_v, Y_l, Y_r]^\top$  and correction factor  $\boldsymbol{\alpha} = [\alpha_l, \alpha_r]^\top$  are also propagated in this step.  $\mathbf{ICR}$  and  $\boldsymbol{\alpha}$  are affected by random walk noise as shown in Eq. 19. So the mean value of the propagated  $\mathbf{ICR}, \boldsymbol{\alpha}$  will remain still. In general, the state  $\mathbf{x}_0 = \left[ \mathbf{G} \mathbf{p}_{\mathbf{O}_0}, \mathbf{G}_{\mathbf{O}_0} \mathbf{R}, \mathbf{O}_0 \mathbf{ICR}, \mathbf{O}_0 \boldsymbol{\alpha} \right]^\top$  at time  $t(0)$  will be propagated to  $\mathbf{x}_1 = \left[ \mathbf{G} \mathbf{p}_{\mathbf{O}_1}, \mathbf{G}_{\mathbf{O}_1} \mathbf{R}, \mathbf{O}_1 \mathbf{ICR}, \mathbf{O}_1 \boldsymbol{\alpha} \right]^\top$  by

odometry measurements in this propagation. We will derive the close-form error state transition matrix  ${}^1_0\Phi$  in this step.

$${}^1_0\Phi = \begin{bmatrix} \mathbf{I}_{3 \times 3} & -\mathbf{G}_{\mathbf{O}_0} \mathbf{R} [\mathbf{O}_0 \mathbf{p}_{\mathbf{O}_1}] & \Phi_A & \Phi_B \\ \mathbf{0}_{3 \times 3} & \mathbf{O}_1^{\mathbf{O}_0} \mathbf{R}^{\top} & \Phi_C & \Phi_D \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{0}_{2 \times 3} & \mathbf{I}_{2 \times 2} \end{bmatrix} \quad (22)$$

The block matrix  $\Phi_A, \Phi_B, \Phi_C, \Phi_D$  can be computed by taking the derivative of Eq. 21 with respect to  $\mathbf{O}_0 \mathbf{ICR}$  and  $\mathbf{O}_0 \alpha$ . We omit some superscripts and subscripts for concise expression, which will not cause confusion.

$$\Phi_A = \frac{\partial \mathbf{G} \delta \mathbf{p}_{\mathbf{O}_1}}{\partial \delta \mathbf{ICR}} = \frac{\Delta t}{2} \mathbf{G}_{\mathbf{O}_0} \mathbf{R} * \{ \mathbf{M}_{v0} + \mathbf{O}_1^{\mathbf{O}_0} \mathbf{R} \mathbf{M}_{v1} - \mathbf{O}_1^{\mathbf{O}_0} \mathbf{R} [\mathbf{v}_{\mathbf{O}_1}] \mathbf{J}_r \mathbf{M}_{\omega} \} \quad (23)$$

Among the above expression,

$$\mathbf{M}_{v0} = \frac{1}{\Delta Y^2} \begin{bmatrix} 0 & Y_r & -Y_l \\ \Delta Y & -X_v & X_v \\ 0 & 0 & 0 \end{bmatrix} * (\alpha_l o_{lm0} - \alpha_r o_{rm0}) \quad (24)$$

$$\mathbf{M}_{v1} = \frac{1}{\Delta Y^2} \begin{bmatrix} 0 & Y_r & -Y_l \\ \Delta Y & -X_v & X_v \\ 0 & 0 & 0 \end{bmatrix} * (\alpha_l o_{lm1} - \alpha_r o_{rm1}) \quad (25)$$

$$\mathbf{M}_{\omega} = \frac{1}{\Delta Y^2} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\alpha_l(o_{lm0}+o_{lm1})-\alpha_r(o_{rm0}+o_{rm1})}{2} & -\frac{\alpha_l(o_{lm0}+o_{lm1})-\alpha_r(o_{rm0}+o_{rm1})}{2} \end{bmatrix} \quad (26)$$

$\mathbf{J}_r$  is the abbreviation of  $\mathbf{J}_r(\bar{\omega} \Delta t)$ , denoting the right Jacobian of  $\bar{\omega} \Delta t$ . If  $\theta = \theta \mathbf{a}$ , the following holds:

$$\mathbf{J}_r(\theta) = \frac{\sin \theta}{\theta} \mathbf{I} + \left(1 - \frac{\sin \theta}{\theta}\right) \mathbf{a} \mathbf{a}^{\top} - \frac{1 - \cos \theta}{\theta} [\mathbf{a}] \quad (27)$$

$$\Phi_B = \frac{\partial \mathbf{G} \delta \mathbf{p}_{\mathbf{O}_1}}{\partial \delta \alpha} = \frac{\Delta t}{2} \mathbf{G}_{\mathbf{O}_0} \mathbf{R} * \{ \mathbf{N}_{v0} + \mathbf{O}_1^{\mathbf{O}_0} \mathbf{R} \mathbf{N}_{v1} - \mathbf{O}_1^{\mathbf{O}_0} \mathbf{R} [\mathbf{v}_{\mathbf{O}_1}] \mathbf{J}_r \mathbf{N}_{\omega} \} \quad (28)$$

$$\mathbf{N}_{v0} = \frac{1}{\Delta Y} \begin{bmatrix} -Y_r o_{lm0} & Y_l o_{rm0} \\ X_v o_{lm0} & -X_v o_{rm0} \\ 0 & 0 \end{bmatrix} \quad (29)$$

$$\mathbf{N}_{v1} = \frac{1}{\Delta Y} \begin{bmatrix} -Y_r o_{lm1} & Y_l o_{rm1} \\ X_v o_{lm1} & -X_v o_{rm1} \\ 0 & 0 \end{bmatrix} \quad (30)$$

$$\mathbf{N}_{\omega} = \frac{1}{\Delta Y} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{(o_{lm0}+o_{lm1})}{2} & \frac{(o_{rm0}+o_{rm1})}{2} \end{bmatrix} \quad (31)$$



$$\Phi_C = \frac{\partial \mathbf{G}_{\mathbf{O}_1} \delta \theta}{\partial \delta \mathbf{ICR}} = \Delta t \frac{1}{\Delta Y^2} \mathbf{J}_r \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{\alpha_l(o_{lm0} + o_{lm1})}{2} - \frac{\alpha_r(o_{rm0} + o_{rm1})}{2} - \frac{\alpha_l(o_{lm0} + o_{lm1})}{2} + \frac{\alpha_r(o_{rm0} + o_{rm1})}{2} \end{bmatrix} \quad (32)$$

$$\Phi_D = \frac{\mathbf{G}_{\mathbf{O}_1} \delta \theta}{\partial \delta \alpha} = \Delta t \frac{1}{\Delta Y} \mathbf{J}_r \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\frac{(o_{lm0} + o_{lm1})}{2} & \frac{(o_{rm0} + o_{rm1})}{2} \end{bmatrix} \quad (33)$$

As for the propagated covariances, we can obtain them by:

$$\mathbf{P}_1 = {}_0^1 \Phi \mathbf{P}_0 {}_0^1 \Phi^\top + \mathbf{G}_0 \mathbf{Q}_0 \mathbf{G}_0^\top \quad (34)$$

where  $\mathbf{Q}_0 \in \mathbb{R}^{9 \times 9}$  is the covariance of noises affecting  $[\mathbf{ICR}^\top, \alpha^\top, \mathbf{o}_0^\top, \mathbf{o}_1^\top]^\top$ . Since we start the integration from  $t_0$ ,  $\mathbf{P}_0$  is zero matrix. And  $\mathbf{G}_0$  is the Jacobian with respect to the noises,

$$\mathbf{G}_0 = \begin{bmatrix} \Phi_A & \Phi_B & \mathbf{G}_A & \mathbf{G}_B \\ \Phi_C & \Phi_D & \mathbf{G}_C & \mathbf{G}_D \\ \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 2} \\ \mathbf{0}_{2 \times 3} & \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \end{bmatrix} \quad (35)$$

In the above expression,  $\mathbf{G}_A = \frac{\partial \mathbf{G}_{\mathbf{O}_1} \delta \mathbf{p}_{\mathbf{O}_1}}{\partial \delta \mathbf{o}_0}$ ,  $\mathbf{G}_B = \frac{\partial \mathbf{G}_{\mathbf{O}_1} \delta \mathbf{p}_{\mathbf{O}_1}}{\partial \delta \mathbf{o}_1}$ ,  $\mathbf{G}_C = \frac{\partial \mathbf{G}_{\mathbf{O}_1} \delta \theta}{\partial \delta \mathbf{o}_0}$ ,  $\mathbf{G}_D = \frac{\partial \mathbf{G}_{\mathbf{O}_1} \delta \theta}{\partial \delta \mathbf{o}_1}$ ,

$$\mathbf{G}_A = \frac{\Delta t}{2\Delta Y} \mathbf{G}_{\mathbf{O}_0} \mathbf{R} * \left( \begin{bmatrix} -\alpha_l Y_r & \alpha_r Y_l \\ \alpha_l X_v & -\alpha_r X_v \\ 0 & 0 \end{bmatrix} - \frac{1}{2} \mathbf{O}_0 \mathbf{R} [\mathbf{v}_{\mathbf{O}_1}] \mathbf{J}_r \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\alpha_l & \alpha_r \end{bmatrix} \right) \quad (36)$$

$$\mathbf{G}_B = \frac{\Delta t}{2\Delta Y} \mathbf{G}_{\mathbf{O}_0} \mathbf{R} * \left( \begin{bmatrix} \mathbf{O}_0 \mathbf{R} & \begin{bmatrix} -\alpha_l Y_r & \alpha_r Y_l \\ \alpha_l X_v & -\alpha_r X_v \\ 0 & 0 \end{bmatrix} \\ \mathbf{O}_1 \mathbf{R} & \begin{bmatrix} -\alpha_l Y_r & \alpha_r Y_l \\ \alpha_l X_v & -\alpha_r X_v \\ 0 & 0 \end{bmatrix} \end{bmatrix} - \frac{1}{2} \mathbf{O}_0 \mathbf{R} [\mathbf{v}_{\mathbf{O}_1}] \mathbf{J}_r \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\alpha_l & \alpha_r \end{bmatrix} \right) \quad (37)$$

$$\mathbf{G}_C = \mathbf{G}_D = \frac{\Delta t}{2\Delta Y} \mathbf{J}_r \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -\alpha_l & \alpha_r \end{bmatrix} \quad (38)$$

In analogy to Eq. 21, we can compute  ${}^G \mathbf{p}_{\mathbf{O}_k}$ ,  $\mathbf{G}_{\mathbf{O}_k} \mathbf{R}$  by the intermediate measurements  $\mathcal{O}_l$  iteratively. In addition, the covariance  $\mathbf{P}_k$  at time  $t(k)$  can be also obtained iteratively like Eq. 34, and we have  $\Lambda_{odom} = \mathbf{P}_k^{-1}$ .

## References

- [1] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo. “Approximating kinematics for tracked mobile robots”. In: *The International Journal of Robotics Research* 24.10 (2005), pp. 867–878.
- [2] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis. “VINS on wheels”. In: *IEEE International Conference on Robotics and Automation*. 2017, pp. 5155–5162.
- [3] M. Quan, S. Piao, M. Tan, and S.-S. Huang. “Tightly-coupled Monocular Visual-odometric SLAM using Wheels and a MEMS Gyroscope”. In: *arXiv:1804.04854* (2018).
- [4] G. Huskic, S. Buck, and A. Zell. “Path following control of skid-steered wheeled mobile robots at higher speeds on different terrain types”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 3734–3739.
- [5] M. Li and A. I. Mourikis. “Online temporal calibration for camera–IMU systems: Theory and algorithms”. In: *The International Journal of Robotics Research* (2014).
- [6] A. Censi, A. Franchi, L. Marchionni, and G. Oriolo. “Simultaneous calibration of odometry and sensor parameters for mobile robots”. In: *IEEE Transactions on Robotics* 29.2 (2013), pp. 475–492.
- [7] M. Li and A. I. Mourikis. “High-precision, consistent EKF-based visual-inertial odometry”. In: *The International Journal of Robotics Research* 32.6 (2013), pp. 690–711.
- [8] K. Eickenhoff, P. Geneva, and G. Huang. “Closed-form Preintegration Methods for Graph-based Visual-Inertial Navigation”. In: *International Journal of Robotics Research* 38.5 (2019), pp. 563–586.
- [9] T. Qin, P. Li, and S. Shen. “VINS-mono: A robust and versatile monocular visual-inertial state estimator”. In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1004–1020.
- [10] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. “Keyframe-based visual–inertial odometry using nonlinear optimization”. In: *The International Journal of Robotics Research* 34.3 (2015), pp. 314–334.
- [11] E. Rosten and T. Drummond. “Machine learning for high-speed corner detection”. In: *European conference on computer vision*. Springer. 2006, pp. 430–443.
- [12] B. D. Lucas, T. Kanade, et al. “An iterative image registration technique with an application to stereo vision”. In: (1981).
- [13] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [14] A. I. Mourikis and S. I. Roumeliotis. “A multi-state constraint Kalman filter for vision-aided inertial navigation”. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE. 2007, pp. 3565–3572.
- [15] M. Zhang, Y. Chen, and M. Li. “Large-Scale Vision-Aided Localization For Ground Vehicle”. In: *submitted* (2019).
- [16] X. Zuo, M. Zhang, Y. Chen, Y. Liu, G. Huang, and M. Li. *Visual Localization for Skid-Steering Robots*. Tech. rep. Available: [http://udel.edu/~ghuang/papers/tr\\_icr.pdf](http://udel.edu/~ghuang/papers/tr_icr.pdf). 2019.