Slip-Robust InEKF for Wheeled Robots

Theodor Chakhachiro[1], Ryan Feng[1], Vladimir Krokhmal[1], Xihang Yu[2], and Hongyu Zhou[3]

***Abstract*—In this paper, we propose a novel Slip-Robust Invariant Extended Kalman Filter (InEKF) for wheeled robot state estimation and kinematic center tracking. We perform an observability analysis on the measurement model of the wheel encoders which leads to unobservable orientation around the gravity vector, as well as unobservable position of the robot. We also provide angular velocity and acceleration biases correction and estimation. We implement this method on a Clearpath Husky Robot traveling over a parking lot near the Ford Motor Company Robotics Building. The results validate the proposed algorithm and compete with Visual SLAM and odometry methods which require camera measurements.**

I. Introduction

The localization problem is critical for the successful deployment of any mobile robotics platform. In order to reason about and navigate through any environment, an accurate estimate of the robot's location is required.

A wide range of sensors can be leveraged towards this goal. On wheeled robots, the focus of our work, these can include wheel encoders, IMUs, GPS, cameras, LiDAR, and more. However, many of these sensors only provide useful data under specific conditions. For example, camera-based state estimation methods suffer in darkness or in feature-sparse environments, while GPS will not work indoors. On the other hand, encoders and IMUs are cheap and flexible sensors usable in nearly any environment and present on almost all wheeled robotic platforms. However, the primary issue of relying on encoders is wheel slip. When the rotation of the wheels does not provide an accurate estimate of the robot's actual motion (i.e. when the wheels slip), encoder-based approaches can fail.

We propose an Invariant Extended Kalman Filter (InEKF) for the fusion of encoder and IMU data with an explicit wheel slip noise term. The inclusion of this term aims to model the errors caused by wheel slip in order to maintain accurate state estimation. The contributions of this work are listed as follows:

- Derivation of an InEKF for encoder-IMU fusion on wheeled robots with wheel slip.
- Wheel encoders measurement model.
- Estimation and tracking of the kinematic center of a wheeled robot.
- Observability analysis on the measurement model of the wheel encoders.

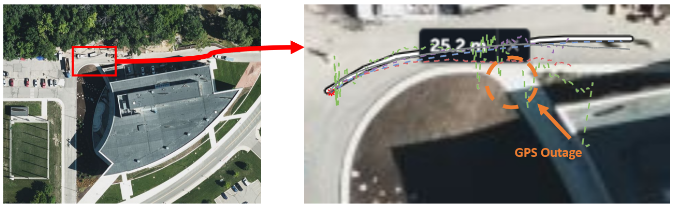

Fig. 1. Ford Motor Company Robotics Building Parking lot trajectory results of the proposed Pseudo-Encoder InEKF algorithm (violet). Encoder-Only InEKF (red), ORB-SLAM2 (blue), GPS data (green) and Visual Odometry (**black**) trajectory results are also plotted for comparison.

- Experimental validation of this approach on a Clearpath Husky robot.
- Open source software available for download at https://github.com/XihangYU630/inekf_wheeled.git.

The remainder of this paper is organized as follows. Section II discusses prior related work in this area. Section III presents the our formulation of this state estimation problem. The filter is derived in Section IV with additional observability analysis in Section V. Section VI presents the results of our method on data collected from a Husky robot. Finally, some conclusions and future work are discussed in Section VII.

II. Related Work

Previous research for husky mobile robot has introduced one algorithm to estimate heading [1]. Although this highly accurate algorithm has estimate one state of husky, it is far away from solving dead-reckoning problem. Recent breakthrough in wheeled robot or autonomous cars has considered different sensor measurements to increase the robustness of state estimation systems [2]–[5]. In spite of their success, these systems, involving camera and LiDAR, require expensive computation and energy consumption. Moreover, IMU-based localization system should be run in parallel for emergent sensor failure such as GPS outage.

Kalman filters are routinely used to integrate the outputs of IMUs. When the IMU is mounted on a car, it is common practice to make the Kalman filter incorporate side information about the specificity of wheeled vehicle dynamics, such as approximately null lateral and upward velocity assumption in the form of pseudo-measurements [6]–[12]. We propose a method to incorporate newly-developed Invariant Extended Kalman filter with pseudo-measurement. After witnessing success of previous work using wheel encoder data fusion [13], [14]. We use wheel encoder data as another way of correction.

A video presentation of this paper is available here.

[1]Theodor Chakhachiro, Ryan Feng, and Vladimir Krokhmal are with the Robotics Institute. {teochiro, rzfeng, krokhmal}@umich.edu

[2]Xihang Yu is with the Dept. of Computer Science. xihangyu@umich.edu

[3]Hongyu Zhou is with the Dept. of Aerospace Engineering. zhouhy@umich.edu

The authors are listed in alphabetical order.

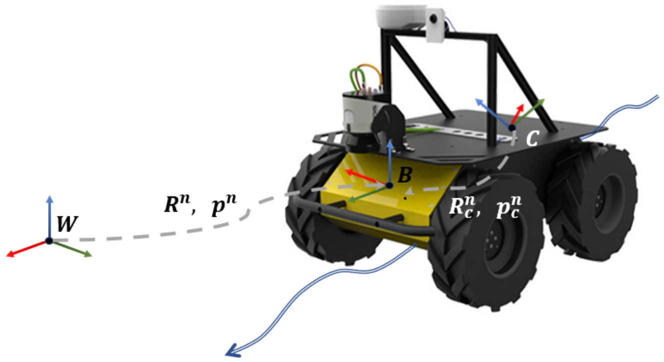

Fig. 2. Husky Robot System where $\boldsymbol{W}$ represents the world frame, R^n and p^n represent the orientation and position of the IMU ($\boldsymbol{B}$) at index n in the world frame. R_c^n and p_c^n represent the orientation and position of the kinematic center ($\boldsymbol{C}$) in the IMU frame.

Slip is very common for husky robot partly due to the way it is designed, i.e. using the difference of forward velocities between two wheels. To address robot state estimation in case of slip, [15] and [16] add force sensors or accelerometers address state estimation. However, these sensors are vulnerable to ground impact. As for wheeled robot, [17] and [18] utilize filtering-based method. However, these algorithms need high computation resource. Considering all above, we adopt the algorithm to add noise covariance in measurement model to incorporate wheel slip as in [19].

III. Problem Formulation

The problem formation is partly adopted based on [19]. Our goal is to estimate the state of a wheel robot, i.e., the orientation $\boldsymbol{R} \in SO(3)$, velocity $\boldsymbol{v} \in \mathbb{R}^3$ and position $\boldsymbol{p} \in \mathbb{R}^3$ of the robot in the world frame, via the inertial measurement unit (IMU), wheel kinematics and pesudo measurement using Invariant EKF. We assume the IMU frame is aligned with the robot frame, and the robot works in a 2D plane and its wheels remain contact to the ground. The robot system and the variables are illustrated in Fig. 2. There are three different frames: *i) the world frame* W*; ii) the IMU frame* B*; and iii) the kinematic center frame* C. The kinematic center frame is an ideal frame attached to the robot, whose orientation and position w.r.t. the IMU frame is denoted $\boldsymbol{R}_c \in SO(3)$ and $\boldsymbol{p}_c \in \mathbb{R}^3$, which represents the misalignment between the IMU frame and the kinematic center frame.

A. System Model

The IMU measures the acceleration $\tilde{\boldsymbol{a}}$ and angular velocity $\tilde{\boldsymbol{\omega}}$ in the IMU frame, both of which are corrupted by white Gaussian noise $\boldsymbol{w}_a, \boldsymbol{w}_\omega$ and bias $\boldsymbol{b}_a, \boldsymbol{b}_\omega \in \mathbb{R}^3$, respectively. Hence, the IMU measurements are

$$\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} + \boldsymbol{b}_\omega + \boldsymbol{w}_\omega, \quad \tilde{\boldsymbol{a}} = \boldsymbol{a} + \boldsymbol{b}_a + \boldsymbol{w}_a. \tag{1}$$

We model the dynamics of the bias as Gaussian random walk, i.e.,

$$\dot{\boldsymbol{b}}_\omega = \boldsymbol{w}_{b\omega}, \quad \dot{\boldsymbol{b}}_a = \boldsymbol{w}_{ba}. \tag{2}$$

Based on the IMU measurements, the dynamics of robot state becomes

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\left(\tilde{\boldsymbol{\omega}} - \boldsymbol{b}_\omega - \boldsymbol{w}_\omega\right)^\times, \tag{3}$$

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \ \dot{\boldsymbol{v}} = \boldsymbol{R}\left(\tilde{\boldsymbol{a}} - \boldsymbol{b}_a - \boldsymbol{w}_a\right) + \boldsymbol{g}. \tag{4}$$

where $\boldsymbol{g}$ is the acceleration due to gravity and $(\cdot)^\times$denote a 3×3 skew symmetric matrix, s.t. $x \times y = x^\times y, \forall x, y \in \mathbb{R}^3$.

We assume $\boldsymbol{R}_c$ and $\boldsymbol{p}_c$ are constants corrupted by white Gaussian noise $\boldsymbol{w}_{pc}$ and $\boldsymbol{w}_{Rc}$. Hence, their dynamics models are

$$\dot{\boldsymbol{R}}_c = \boldsymbol{R}_c \boldsymbol{w}_{Rc}^\times, \quad \dot{\boldsymbol{p}}_c = \boldsymbol{w}_{pc}. \tag{5}$$

B. Wheeled Robot Kinematics Measurements

Let $\boldsymbol{v}_l$ and $\boldsymbol{v}_r$ be the velocity of the center of left and right wheels in IMU frame, $\boldsymbol{\omega}_l$ and $\boldsymbol{\omega}_r$ be the angular velocity of the left and right wheels in IMU frame, $\boldsymbol{r}_{1l}$ and $\boldsymbol{r}_{1r}$ be the vector from the center of the left and right wheel to the contact point, and $\boldsymbol{r}_2$ be the vector from the left to right wheel.

Assume that there is no slippery between the contact point on the wheel and the floor, the velocity at the contact point should be zero, i.e., $\boldsymbol{\omega}_l^\times \boldsymbol{r}_{1l} = \boldsymbol{v}_l$.

Now we can connect $\boldsymbol{v}_l$ and $\boldsymbol{v}_r$ with $\boldsymbol{R}$, $\boldsymbol{v}$, and $\boldsymbol{\omega}$ by

$$2\boldsymbol{R}^\top \boldsymbol{v} = \boldsymbol{v}_l + \boldsymbol{v}_r, \quad \boldsymbol{\omega}^\times \boldsymbol{r}_2 = \boldsymbol{v}_r - \boldsymbol{v}_l \tag{6}$$

Using $\boldsymbol{\omega}_l^\times \boldsymbol{r}_{1l} = \boldsymbol{v}_l$, we have

$$\boldsymbol{\omega}_l^\times \boldsymbol{r}_{1l} - \frac{1}{2}\boldsymbol{\omega}^\times \boldsymbol{r}_2 = \boldsymbol{R}^\top \boldsymbol{v}. \tag{7}$$

Similarly, for the right wheel we have

$$\boldsymbol{\omega}_l^\times \boldsymbol{r}_{1r} + \frac{1}{2}\boldsymbol{\omega}^\times \boldsymbol{r}_2 = \boldsymbol{R}^\top \boldsymbol{v}. \tag{8}$$

Consider the noisy measurements of $\boldsymbol{\omega}$ and $\boldsymbol{\omega}_l$, and error caused by slippery, the measurement model becomes

$$\begin{aligned} \tilde{\boldsymbol{\omega}}_l^\times \boldsymbol{r}_{1l} - \frac{1}{2}\tilde{\boldsymbol{\omega}}^\times \boldsymbol{r}_2 &= \boldsymbol{R}^\top \boldsymbol{v} + \boldsymbol{n}_f \\ \tilde{\boldsymbol{\omega}}_l^\times \boldsymbol{r}_{1r} + \frac{1}{2}\tilde{\boldsymbol{\omega}}^\times \boldsymbol{r}_2 &= \boldsymbol{R}^\top \boldsymbol{v} + \boldsymbol{n}_f \end{aligned} \tag{9}$$

where $\boldsymbol{n}_f$ is the white noise caused by slippery.

C. Pseudo Measurement

The velocity of the robot represented in the kinematics center frame is

$$\boldsymbol{v}_c = \begin{bmatrix} v_c^{for} \\ v_c^{lat} \\ v_c^{ver} \end{bmatrix} = \boldsymbol{R}_c^\top \boldsymbol{R}^\top \boldsymbol{v} + \boldsymbol{\omega}_c^\times \boldsymbol{R}_c^\top \boldsymbol{p}_c \tag{10}$$

where v_c^{for}, v_c^{lat}, and v_c^{ver} are forward, lateral, and vertical speeds. In the kinematics center frame, we assume that the lateral and vertical speeds are roughly zero, and are corrupted by Gaussian white noise. Hence, we obtain two scalar pseudo observations,

$$\boldsymbol{y}_c = \begin{bmatrix} y_c^{lat} \\ y_c^{ver} \end{bmatrix} = \begin{bmatrix} v_c^{lat} \\ v_c^{ver} \end{bmatrix} + \boldsymbol{n}_c, \tag{11}$$

where $\boldsymbol{n}_c$ is Gaussian white noise. The filter is then fed with the pseudo-measurement of $\boldsymbol{y}_c = \boldsymbol{0}$.

D. Problem Formulation

Based on the system model (2)-(5) and measurement models (9) and (44), we define the state variables:

$$\boldsymbol{x} := (\boldsymbol{R}, \boldsymbol{v}, \boldsymbol{p}, \boldsymbol{b}_\omega, \boldsymbol{b}_a, \boldsymbol{R}_c, \boldsymbol{p}_c).$$

Our goal is to use the measurements from the IMU, wheel kinematics and pseudo measurements to estimate $\boldsymbol{x}$. Note that our method does not require measurements from a camera. By including the noise term $\boldsymbol{n}_f$ in measurement (9), the estimation is slip-robust.

IV. Invariant Extended Kalman Filter

A. Math Prerequisite

Denote a $n \times n$ matrix Lie group $\mathscr{G}$ and its associated Lie algebra $\mathfrak{g}$, representing the tangent space of $\mathscr{G}$ at the identity. We define the wedge operator as

$$(\cdot)^\wedge : \mathbb{R}^n \to \mathfrak{g}, \tag{12}$$

such that $\forall \boldsymbol{\xi} \in \mathbb{R}^n$, we can associate it with the elements in the matrix Lie group through the exponential map

$$\exp(\cdot) : \mathbb{R}^n \to \mathscr{G}, \quad \exp(\boldsymbol{\xi}) = \exp_{\mathrm{m}}\left(\boldsymbol{\xi}^\wedge\right), \tag{13}$$

where $\exp_{\mathrm{m}}(\cdot)$ is the matrix exponential.

We define the adjoint map $\mathrm{Ad}_\chi : \mathfrak{g} \to \mathfrak{g}$, which converse from local to global coordinate, as,

$$\mathrm{Ad}_{\boldsymbol{\chi}}\left(\boldsymbol{\xi}^\wedge\right) = \boldsymbol{\chi}\boldsymbol{\xi}^\wedge\boldsymbol{\chi}^{-1}, \quad \forall \boldsymbol{\chi} \in \mathscr{G} \tag{14}$$

Consider a system defined on matrix Lie group $\mathscr{G}$ and its associate Lie algebra $\mathfrak{g}$ with input u_t,

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\chi}_t = f_{u_t}\left(\boldsymbol{\chi}_t\right), \quad \boldsymbol{\chi}_t \in \mathscr{G}, \quad t \geq 0 \tag{15}$$

The right invariant error between the two trajectories $\boldsymbol{\chi}_t$ and $\overline{\boldsymbol{\chi}}_t$ are defined as

$$\boldsymbol{\eta}_t = \overline{\boldsymbol{\chi}}_t\boldsymbol{\chi}_t^{-1} \tag{16}$$

If the system in (15) is group affine, i.e.,

$$f_{u_t}\left(\boldsymbol{\chi}_1\boldsymbol{\chi}_2\right) = f_{u_t}\left(\boldsymbol{\chi}_1\right)\boldsymbol{\chi}_2 + \boldsymbol{\chi}_1 f_{u_t}\left(\boldsymbol{\xi}_2\right) - \boldsymbol{\chi}_1 f_{u_t}\left(\boldsymbol{I}_d\right)\boldsymbol{\chi}_2 \tag{17}$$

then the error dynamics are independent of state and can be expressed as [20]

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\eta}_t = g_{u_t}\left(\boldsymbol{\eta}_t\right) := f_{u_t}\left(\boldsymbol{\eta}_t\right) - \boldsymbol{\eta}_t f_{u_t}\left(\boldsymbol{I}_d\right) \tag{18}$$

Recall that the first order approximation of the exponential map is $\boldsymbol{\eta}_t = \exp\left(\boldsymbol{\xi}_t\right) \approx \boldsymbol{I} + \boldsymbol{\xi}_t^\wedge$, we can linearize the error dynamics as

$$g_{u_t}\left(\exp\left(\boldsymbol{\xi}_t\right)\right) = \left(\boldsymbol{A}_t\boldsymbol{\xi}_t\right)^\wedge + o\left(\|\boldsymbol{\xi}_t\|\right) \approx \left(\boldsymbol{A}_t\boldsymbol{\xi}_t\right)^\wedge \tag{19}$$

where the the Jacobian matrix $\boldsymbol{A}_t$ is independent of $\boldsymbol{\xi}_t$.

Combining (18)-(19), we obtain a linear differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\xi}_t = \boldsymbol{A}_t\boldsymbol{\xi}_t \tag{20}$$

The log-linear property of error propagation [20] suggests that if the initial error satisfies $\boldsymbol{\eta}_0 = \exp\left(\boldsymbol{\xi}_0\right)$, the linearized error dynamics (18) can fully represent the nonlinear error defined in (16) as

$$\boldsymbol{\eta}_t = \exp\left(\boldsymbol{\xi}_t\right), \quad t \geq 0 \tag{21}$$

Hence, for a system satisfying the group affine property, designing a observor based on the linear differential equation (20) avoids problems of nonlinear observer design.

B. Estimation Error Dynamics

Based on the system model (2)-(5), we give the definition of the error between the estimated states $\hat{\boldsymbol{x}}$ and the real states $\boldsymbol{x}$, where $\hat{(\cdot)}$ denotes the estimated states.

The IMU $(\boldsymbol{R}, \boldsymbol{v}, \boldsymbol{p})$ are embedded in the double direct spatial isometry $SE_2(3)$ [20], i.e.,

$$\boldsymbol{\chi} := \begin{bmatrix} \boldsymbol{R} & \boldsymbol{v} & \boldsymbol{p} \\ \boldsymbol{0}_{1\times3} & 1 & 0 \\ \boldsymbol{0}_{1\times3} & 0 & 1 \end{bmatrix} \tag{22}$$

Without bias, the dynamics of states $(\boldsymbol{R}, \boldsymbol{v}, \boldsymbol{p})$ are group affine. Thus, we define the right-invariant error and linearize it,

$$\begin{aligned} \boldsymbol{\eta} = \hat{\chi}\chi^{-1} &= \begin{bmatrix} \hat{\boldsymbol{R}}\boldsymbol{R}^\top & \hat{\boldsymbol{v}} - \hat{\boldsymbol{R}}\boldsymbol{R}^\top\boldsymbol{v} & \hat{\boldsymbol{p}} - \hat{\boldsymbol{R}}\boldsymbol{R}^\top\boldsymbol{p} \\ \boldsymbol{0}_{1\times3} & 1 & 0 \\ \boldsymbol{0}_{1\times3} & 0 & 1 \end{bmatrix} \\ &\approx \begin{bmatrix} \boldsymbol{I} + \boldsymbol{\xi}_R\wedge & \boldsymbol{\xi}_v & \boldsymbol{\xi}_p \\ \boldsymbol{0}_{1\times3} & 1 & 0 \\ \boldsymbol{0}_{1\times3} & 0 & 1 \end{bmatrix} \end{aligned} \tag{23}$$

We define $\boldsymbol{\xi}_{IMU} := \left[\boldsymbol{\xi}_R^\top, \boldsymbol{\xi}_v^\top, \boldsymbol{\xi}_p^\top\right]^\top \in \mathbb{R}^9$. Using $\boldsymbol{\xi}_{IMU}$ we can formulate the standard InEKF. However, augmenting the error of IMU biases and kinematics center misalignment will result in an imperfect InEKF [21], i.e, the state matrix of linearized system (20) becomes state dependent.

The estimation error of $\boldsymbol{R}_c$ is

$$\eta_c = \hat{\boldsymbol{R}}_c\boldsymbol{R}_c^\top = \exp\left(\boldsymbol{\xi}_{R_c}\right) \approx \boldsymbol{I} + \boldsymbol{\xi}_{R_c} \tag{24}$$

For IMU biases and kinematics center position, the estimation error are calculated in Euclidean space, i.e:

$$\boldsymbol{e}_{b\omega} = \hat{\boldsymbol{b}}_{\boldsymbol{\omega}} - \boldsymbol{b}_{\boldsymbol{\omega}}, \quad \boldsymbol{e}_{ba} = \hat{\boldsymbol{b}}_{\boldsymbol{a}} - \boldsymbol{b}_a, \quad \boldsymbol{e}_{pc} = \hat{\boldsymbol{p}}_c - \boldsymbol{p}_c \tag{25}$$

Concatenating the error in (23)-(25) gives the linearized estimation error of the entire system,

$$\boldsymbol{\xi} := \left[\boldsymbol{\xi}_{IMU}^T, \boldsymbol{e}_{b\omega}^\top, \boldsymbol{e}_{ba}^\top, \boldsymbol{\xi}_{Rc}^\top, \boldsymbol{e}_{pc}^\top\right]^\top \in \mathbb{R}^{21}, \tag{26}$$

and the linearized estimation error dynamics are as follows,

$$\begin{aligned} &\frac{d}{dt}\left(\hat{\boldsymbol{R}}\boldsymbol{R}^\top\right) \approx \left(\hat{\boldsymbol{R}}\left(\boldsymbol{w}_\omega - \boldsymbol{e}_{b\omega}\right)\right)^\times, \\ &\frac{d}{dt}\left(\hat{\boldsymbol{v}} - \hat{\boldsymbol{R}}\boldsymbol{R}^\top\boldsymbol{v}\right) \approx \boldsymbol{g}^\times\boldsymbol{\xi}_R + \hat{\boldsymbol{R}}\left(\boldsymbol{w}_a - \boldsymbol{e}_{ba}\right) \\ &+ \hat{\boldsymbol{v}}^\times\boldsymbol{\xi}_R\left(\boldsymbol{w}_\omega - \boldsymbol{e}_{b\omega}\right), \\ &\frac{d}{dt}\left(\hat{\boldsymbol{p}} - \hat{\boldsymbol{R}}\boldsymbol{R}^\top\boldsymbol{p}\right) \approx \boldsymbol{\xi}_v + \hat{\boldsymbol{p}}^\times\hat{\boldsymbol{R}}\left(\boldsymbol{w}_\omega - \boldsymbol{e}_{b\omega}\right) \\ &\frac{d}{dt}\boldsymbol{e}_{b\omega} = \boldsymbol{w}_{b\omega}, \quad \frac{d}{dt}\boldsymbol{e}_{b\omega} = \boldsymbol{w}_{ba}, \\ &\frac{d}{dt}\left(\hat{\boldsymbol{R}}_c\boldsymbol{R}_c^\top\right) = \boldsymbol{w}_{Rc}^\times, \quad \frac{d}{dt}\boldsymbol{e}_{pc} = \boldsymbol{w}_{pc} \end{aligned} \tag{27}$$

Representing (27) in matrix form gives

$$\dot{\boldsymbol{\xi}} = \boldsymbol{A}\boldsymbol{\xi} + \boldsymbol{B}_{\hat{\boldsymbol{x}}}\boldsymbol{w} \tag{28}$$

where

$$\boldsymbol{A} = \left[\begin{array}{cccccc} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\hat{\boldsymbol{R}} & \mathbf{0}_3 & \mathbf{0}_{3\times 6} \\ \boldsymbol{g}^\times & \mathbf{0}_3 & \mathbf{0}_3 & -\hat{\boldsymbol{v}}^\times \hat{\boldsymbol{R}} & -\hat{\boldsymbol{R}} & \mathbf{0}_{3\times 6} \\ \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 & -\hat{\boldsymbol{p}}^\times \hat{\boldsymbol{R}} & \mathbf{0}_3 & \mathbf{0}_{3\times 6} \\ & \mathbf{0}_{12\times 9} & & & \mathbf{0}_{12} & \end{array}\right] \tag{29}$$

$$\boldsymbol{B}_{\hat{\boldsymbol{x}}} = \left[\begin{array}{cc} \mathrm{Ad}_{\hat{\chi}} & \mathbf{0}_{9\times 12} \\ \mathbf{0}_{12\times 9} & \boldsymbol{I}_{12} \end{array}\right], \mathrm{Ad}_{\hat{\chi}} = \left[\begin{array}{ccc} \hat{\boldsymbol{R}} & \mathbf{0}_3 & \mathbf{0}_3 \\ \hat{\boldsymbol{v}}^\times \hat{\boldsymbol{R}} & \hat{\boldsymbol{R}} & \mathbf{0}_3 \\ \hat{\boldsymbol{p}}^\times \hat{\boldsymbol{R}} & \mathbf{0}_3 & \hat{\boldsymbol{R}} \end{array}\right] \tag{30}$$

$$\boldsymbol{w} := \mathrm{vec}\left(\boldsymbol{w}_\omega, \boldsymbol{w}_a, \mathbf{0}_{3\times 1}, \boldsymbol{w}_{b_a}, \boldsymbol{w}_{b_\omega}, \boldsymbol{w}_{R_C}, \boldsymbol{w}_{p_c}\right) \tag{31}$$

Note $\boldsymbol{w}$ corresponds to the noise with covariance is $\boldsymbol{P}$. The dynamics of $\boldsymbol{P}$ is governed by the Ricatti equation, i.e.,

$$\frac{\mathrm{d}}{\mathrm{dt}}\boldsymbol{P} = \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}^\top + \boldsymbol{Q}, \quad \boldsymbol{Q} = \boldsymbol{B}_{\hat{\boldsymbol{x}}}\,\mathrm{Cov}(\boldsymbol{w})\boldsymbol{B}_{\hat{\boldsymbol{x}}}^\top \tag{32}$$

C. Propagation Steps

Assuming zero-order hold of the input, the discrete error dynamics between two consecutive time steps t_k and t_{k+1} is

$$\begin{aligned} \hat{\boldsymbol{R}}_{k+1}^- &= \hat{\boldsymbol{R}}_k^+ \exp\left(\left(\tilde{\boldsymbol{w}}_k - \hat{\boldsymbol{b}}_{\omega,k}^+\right)\Delta t\right) \\ \hat{\boldsymbol{v}}_{k+1}^- &= \hat{\boldsymbol{v}}_k^+ + \hat{\boldsymbol{R}}_k^+\left(\tilde{\boldsymbol{a}}_k - \hat{\boldsymbol{b}}_{a,k}^+\right)\Delta t + \boldsymbol{g}\Delta t \\ \hat{\boldsymbol{p}}_{k+1}^- &= \hat{\boldsymbol{p}}_k^+ + \hat{\boldsymbol{v}}_k^+\Delta t + \frac{1}{2}\hat{\boldsymbol{R}}_k^+\left(\tilde{\boldsymbol{a}}_k - \hat{\boldsymbol{b}}_{a,k}^+\right)\Delta t^2 + \frac{1}{2}\boldsymbol{g}\Delta t^2 \\ \hat{\boldsymbol{b}}_{\omega,k+1}^- &= \hat{\boldsymbol{b}}_{\omega,k}^+, \quad \hat{\boldsymbol{b}}_{a,k+1}^- = \hat{\boldsymbol{b}}_{a,k}^+ \\ \hat{\boldsymbol{R}}_{c,k+1}^- &= \hat{\boldsymbol{R}}_{c,k}^+, \quad \hat{\boldsymbol{p}}_{c,k+1}^- = \hat{\boldsymbol{p}}_{c,k}^+ \end{aligned} \tag{33}$$

where $\Delta t = t_{k+1} - t_k, (\cdot)_k^+$ denotes the estimated states at time t_k with all measurements until t_k are processed, and $(\cdot)_k^-$ denotes the state estimated at time t_k through propagation. Given the system (28) at time t_k, the covariance propagation equation is

$$\Phi_k = \exp_{\mathrm{m}}\left(\boldsymbol{A}_k\Delta t\right), \quad \boldsymbol{P}_{k+1} = \Phi_k\boldsymbol{P}_k\Phi_k^\top + \boldsymbol{Q}_k, \tag{34}$$

where $\boldsymbol{Q}_k \approx \Phi_k\boldsymbol{Q}\Phi_k^\top\Delta t$. For more details on the system discretization, please refer to [22].

D. Update Steps with Encoder

The wheel kinematics measurement (9) formulates the rightinvariant observation [20], which exploits the geometry property of the system,

$$\boldsymbol{y} = \boldsymbol{\chi}^{-1}\boldsymbol{b} + \boldsymbol{s} \tag{35}$$

where

$$\begin{aligned} \boldsymbol{y} &= \left[-\left(\tilde{\boldsymbol{\omega}}_l^\times\boldsymbol{r}_1 + \frac{1}{2}\tilde{\boldsymbol{\omega}}^\times\boldsymbol{r}_2\right)^\top, -1, 0\right]^\top \\ \boldsymbol{b} &= \left[\mathbf{0}_{1\times 3}, -1, 0\right]^\top, \quad \boldsymbol{s} = \left[\boldsymbol{n}_f^\top, 0, 0\right]^{\mathrm{T}} \end{aligned} \tag{36}$$

The update equation for χ are given by

$$\hat{\boldsymbol{\chi}}^+ = \exp\left(\boldsymbol{L}\left(\hat{\boldsymbol{\chi}}^-\boldsymbol{y} - \boldsymbol{b}\right)\right)\hat{\boldsymbol{\chi}}^- \tag{37}$$

$$\boldsymbol{\eta}^+ = \exp\left(\boldsymbol{L}\left(\boldsymbol{\eta}^-\boldsymbol{b} - \boldsymbol{b} + \boldsymbol{\chi}^-\boldsymbol{s}\right)\right)\hat{\boldsymbol{\eta}}^- \tag{38}$$

where L is the observation gain matrix. Using the first order approximation we obtain

$$\begin{aligned} \boldsymbol{\eta}^+ &= \exp\left(\boldsymbol{\xi}_{IMU}^+\right) \approx \left(\boldsymbol{I} + \boldsymbol{\xi}_{IMU}^+\right)^\wedge \\ &\approx \left(\boldsymbol{I} + \boldsymbol{\xi}_{IMU}^-\right)^\wedge + \left(\boldsymbol{L}\boldsymbol{\Pi}\left(\boldsymbol{\eta}^-\boldsymbol{b} - \boldsymbol{b} + \boldsymbol{\chi}^-\boldsymbol{s}\right)\right)^\wedge \end{aligned} \tag{39}$$

where $\boldsymbol{\Pi} = [\boldsymbol{I}_3, \mathbf{0}_{3\times 2}]$ is used to select the first three rows of $\boldsymbol{\eta}^-\boldsymbol{b} - \boldsymbol{b} + \boldsymbol{\chi}^-\boldsymbol{s}$. The update equation for $\boldsymbol{\xi}_{IMU}$ is then

$$\boldsymbol{\xi}_{IMU}^+ = \boldsymbol{\xi}_{IMU}^- - \boldsymbol{L}\left(\left[\mathbf{0}_3, \boldsymbol{I}_3, \mathbf{0}_3\right]\boldsymbol{\xi}_{IMU}^- - \left[\mathbf{0}_{1\times 3}, \left(\hat{\boldsymbol{R}}\boldsymbol{n}_f\right)^\top, \mathbf{0}_{1\times 3}\right]^\top\right) \tag{40}$$

Note that the above update is only applied to the IMU states. Expanding the update equation to full state, we now have

$$\boldsymbol{\xi}^+ = \boldsymbol{\xi}^- - \boldsymbol{K}\left(\boldsymbol{H}\boldsymbol{\xi} - \left[\mathbf{0}_{1\times 3}, \left(\hat{\boldsymbol{R}}^f\right)^\top, \mathbf{0}_{1\times 15}\right]^\top\right) \tag{41}$$

where the Kalman gain $\boldsymbol{K}$ is given by

$$\begin{aligned} \boldsymbol{K} &= \boldsymbol{P}\boldsymbol{H}^\top\boldsymbol{S}^{-1}, \quad \boldsymbol{S} = \boldsymbol{H}\boldsymbol{P}^-\boldsymbol{H}^\top + \boldsymbol{N} \\ \boldsymbol{N} &= \hat{\boldsymbol{R}}\,\mathrm{Cov}\left(\boldsymbol{n}_f\right)\hat{\boldsymbol{R}}^\top, \quad \boldsymbol{H} = \left[\mathbf{0}_3, \boldsymbol{I}_3, \mathbf{0}_{3\times 15}\right] \end{aligned} \tag{42}$$

Denote $\boldsymbol{\delta} := \left[\boldsymbol{\delta}_{IMU}^\top, \boldsymbol{\delta}_{b\omega}^\top, \boldsymbol{\delta}_{ba}^\top, \boldsymbol{\delta}_{Rc}^\top, \boldsymbol{\delta}_{pc}^\top\right]^\top = \boldsymbol{K}\boldsymbol{\Pi}(\hat{\chi}\boldsymbol{y} - \boldsymbol{b})$ the correction term, we then have the update equations for the full states,

$$\begin{aligned} \hat{\boldsymbol{\chi}}^+ &= \exp\left(\boldsymbol{\delta}_{IMU}\right)\hat{\boldsymbol{\chi}}^-, \\ \boldsymbol{b}_a^+ &= \boldsymbol{b}_a^- + \boldsymbol{\delta}_{ba}, \quad \boldsymbol{b}_\omega^+ = \boldsymbol{b}_\omega^- + \boldsymbol{\delta}_{b\omega} \\ \boldsymbol{R}_c^+ &= \exp\left(\boldsymbol{\delta}_{Rc}\right)\boldsymbol{R}_c^-, \quad \boldsymbol{p}_c^+ = \boldsymbol{p}_c^- + \boldsymbol{\delta}_{pc} \\ \boldsymbol{P}^+ &= (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\boldsymbol{P}^-(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})^\top + \boldsymbol{K}\boldsymbol{N}\boldsymbol{K}^\top \end{aligned} \tag{43}$$

E. Update Steps with Pseudo Measurement

The pseudo measurement $\boldsymbol{y}_c$ with covariance $\mathbf{N}_c$ is defined as follows:

$$\boldsymbol{y}_c = \begin{bmatrix} v_c^{lat} \\ v_c^{ver} \end{bmatrix} = \mathbf{0}, \tag{44}$$

Replacing $\boldsymbol{N}$ and $\boldsymbol{H}$ in equations (42) with $\boldsymbol{N}_c$ and $\boldsymbol{H}_c$, we have the update term

$$\boldsymbol{\delta}_c = \boldsymbol{K}\left(\boldsymbol{y}_c - \tilde{\boldsymbol{y}}_c\right), \tag{45}$$

where

$$\boldsymbol{H}_c = \boldsymbol{A}_c\begin{bmatrix}\mathbf{0} & \boldsymbol{R}^\top & \mathbf{0} & -p_c^\times & \mathbf{0} & \boldsymbol{B}_c & \boldsymbol{C}_c\end{bmatrix} \tag{46}$$

$$\boldsymbol{A}_c = \begin{bmatrix}\boldsymbol{I}_2 & \mathbf{0}_2\end{bmatrix}, \ \boldsymbol{B}_c = \boldsymbol{R}_c^\top\boldsymbol{R}^\top\boldsymbol{v}^\times, \ \boldsymbol{C}_c = -\left(\boldsymbol{\omega} - \boldsymbol{b}_\omega\right)_\times \tag{47}$$

We then compute an updated states and covariance $\boldsymbol{P}^+$ in the same way as in (43).

V. Observability Analysis

This section discusses the observability of the filter. Previous studies have proven the unobservability of RI-EKF in [20] and [23]. As the encoder measurements is identical to leg kinematics measurement model in [19], we could obtain the unobservable states of the filter without nonlinear observability analysis. Thus we only discuss filter of pseudo-measurement model.

The observability matrix is as follows:

$$\hat{\boldsymbol{O}} = \begin{bmatrix} \boldsymbol{H}_k^- \\ \boldsymbol{H}_{k+1}^- \boldsymbol{\Phi}_k^+ \\ \boldsymbol{H}_{k+2}^- \boldsymbol{\Phi}_{k+1}^+ \boldsymbol{\Phi}_k^+ \\ \vdots \end{bmatrix} \tag{48}$$

where

$$\boldsymbol{H}_k^- = \boldsymbol{A}_c \begin{bmatrix} \mathbf{0} & \boldsymbol{R}_k^{-\top} & \mathbf{0} & -\boldsymbol{p}_{\mathrm{c,k}}^{-\times} & \mathbf{0} & \boldsymbol{B}_{c,k}^- & \boldsymbol{C}_{c,k}^- \end{bmatrix} \tag{49}$$

$$\boldsymbol{A}_c = \begin{bmatrix} \boldsymbol{I}_2 & \mathbf{0}_2 \end{bmatrix}, \ \boldsymbol{B}_{c,k}^- = \boldsymbol{R}_{c,k}^{-\top} \boldsymbol{R}_k^{-\top} \boldsymbol{v}_k^{\times}, \ \boldsymbol{C}_{c,k}^- = -\left(\boldsymbol{\omega}_k - \boldsymbol{b}_{\omega,k}\right)^{\times} \tag{50}$$

$$\begin{aligned} \boldsymbol{\Phi}_k^+ &= \exp\left(\boldsymbol{A}_k \Delta t\right) = \boldsymbol{I} + \boldsymbol{A}_k \Delta t \\ &= \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & -\boldsymbol{R}_k^+ \Delta t & \mathbf{0}_3 & \mathbf{0}_{3\times6} \\ \boldsymbol{g}^{\times} \Delta t & \mathbf{I}_3 & \mathbf{0}_3 & -\boldsymbol{v}_k^{+\times} \boldsymbol{R}_k^+ \Delta t & -\boldsymbol{R}_k^+ \Delta t & \mathbf{0}_{3\times6} \\ \mathbf{0}_3 & \boldsymbol{I}_3 \Delta t & \mathbf{I}_3 & -\boldsymbol{p}_k^{+\times} \boldsymbol{R}_k^+ \Delta t & \mathbf{0}_3 & \mathbf{0}_{3\times6} \\ & \mathbf{0}_{12\times9} & & & \boldsymbol{I}_{12} & \end{bmatrix} \end{aligned} \tag{51}$$

Plug (49) and (51) into (48), we obtain the observability matrix:

$$\hat{\boldsymbol{O}} = \begin{bmatrix} \boldsymbol{A}_c \boldsymbol{O}_1 \\ \boldsymbol{A}_c \boldsymbol{O}_2 \\ \boldsymbol{A}_c \boldsymbol{O}_3 \\ \vdots \end{bmatrix} \tag{52}$$

By analyzing the first three rows of $\hat{\boldsymbol{O}}$, we can obtain its null space. Write out $\boldsymbol{O}_1$ up to $\boldsymbol{O}_3$:

$$\begin{aligned} \boldsymbol{O}_1 &= \begin{bmatrix} \mathbf{0} & \boldsymbol{R}_k^{-\top} & \mathbf{0} & -\boldsymbol{p}_{c,k}^{-\times} & \mathbf{0} & \boldsymbol{B}_{c,k}^- & \boldsymbol{C}_{c,k}^- \end{bmatrix}, \\ \boldsymbol{O}_2 &= \begin{bmatrix} \#_{1,0} & \#_{1,1} & \mathbf{0} & \#_{1,3} & \#_{1,4} & \boldsymbol{B}_{c,k+1}^- & \boldsymbol{C}_{c,k+1}^- \end{bmatrix}, \\ \boldsymbol{O}_3 &= \begin{bmatrix} \#_{2,0} & \#_{2,1} & \mathbf{0} & \#_{2,3} & \#_{2,4} & \boldsymbol{B}_{c,k+2}^- & \boldsymbol{C}_{c,k+2}^- \end{bmatrix} \end{aligned} \tag{53}$$

where

$$\begin{aligned} \#_{1,0} &= \boldsymbol{R}_{k+1}^{-\top} \boldsymbol{g}^{\times} \Delta t, \\ \#_{1,1} &= \boldsymbol{R}_{k+1}^{-\top}, \\ \#_{1,3} &= -\boldsymbol{R}_{k+1}^{-\top} \boldsymbol{v}_k^{\times} \boldsymbol{R}_k^+ \Delta t - \left(\boldsymbol{p}_{c,k+1}^-\right)^{\times}, \\ \#_{1,4} &= -\boldsymbol{R}_{k+1}^{-\top} \boldsymbol{R}_k^+ \Delta t, \\ \#_{2,0} &= \boldsymbol{R}_{k+1}^{-\top} \boldsymbol{g}^{\times} \Delta t, \\ \#_{2,1} &= 2\boldsymbol{R}_{k+2}^{-\top}, \\ \#_{2,3} &= \boldsymbol{R}_{k+2}^{-\top} \left(-\boldsymbol{v}_k^{\times} \boldsymbol{R}_k^+ \Delta t - \boldsymbol{v}_{k+1}^{\times} \boldsymbol{R}_{k+1}^+ \Delta t\right) - \left(\boldsymbol{p}_{c,k+1}^-\right)^{\times}, \\ \#_{2,4} &= -\boldsymbol{R}_{k+2}^{-\top} \left(-\boldsymbol{R}_k^+ \Delta t - \boldsymbol{R}_{k+1}^+ \Delta t\right) \end{aligned} \tag{54}$$

Because the gravity vector only has a z component, the third column of $\hat{O}$ is all zero. Therefore, a rotation about the gravity vector is unobservable. In addition, the absolute position of the robot is unobservable because the seventh to ninth columns of $\hat{O}$ are all zeros. For the rest of state variables, they are observable when multiple different measurements are obtained.

VI. Numerical Results

A. Data Set

To evaluate the performance of our proposed methods, we use a data set collected by the Clearpath Husky robot. This experiment was carried out near the parking lot of the Ford Motor Company Robotics Building at the University of Michigan in An Arbor on a rainy day. This terrain is appropriate for our implementation since its an outdoor environment where slipping in most likely to occur because of the nature of the ground as well as the accumulated rain.

The Husky robot is equipped with IMU, GPS, ZED Stereo Camera, and encoders for each of the four wheels. IMU records linear acceleration and angular velocity of the robot with a frequency of 100 Hz. GPS collects data on the latitude, longitude, and the altitude of the Husky with a frequency of 20 Hz. Encoders record the angular velocity of each wheel at the rate of 100 Hz. The camera collects the data on the pose and orientation of the robot at 30 Hz. The data obtained from the camera provides an opportunity to calculate linear and angular velocity of the robot for visual odometry.

During the course of the experiment, an ORB-SLAM2 stereo-vision based method is run on the Husky. This algorithm produces pose and orientation estimates at the rate of 15 Hz.

GPS, ORB-SLAM2, and Visual Odometry data are processed to yield position of the robot in the world frame. The trajectories produced by these sources are used as benchmarks to evaluate the performance of the proposed method. Recorded IMU and encoders measurements data is used in the prediction and correction step of the proposed InEKF method to calculate accurate estimate of our state vector.

B. Parameter Tuning

We perform some parameter tuning to optimize the results presented later on. For correct state vector convergence as well as robustness against slip, the covariance of the noise should be tuned ”well-enough”. We also perform a parameter tuning method similar to [19]. However, this method led to unsatifactory results and thus was quickly dropped.

C. Results

Here, we present the results of our proposed methods on the parking lot dataset. First, we show state estimation and trajectory tracking of our Encoder-Only InEKF algorithm followed by the results of our Pseudo-Encoder InEKF algorithm. Also, we conduct a state convergence study for the latter aligning with our observability analysis. Finally, we compare the performance of these algorithms and discuss the results.

For all these results, both Matlab and Python scripts were run on a HP Laptop with an Intel core i7-8550 CPU, 1.8GHz, and 16GB RAM.

1) Encoder-Only Results: We compare our proposed method to ORB-SLAM2, a Visual Odometry algorithm running on the ZED camera of the Husky robot and to GPS. As a benchmark, we choose Visual Odometry since our InEKF is an odometry method that utilizes only IMU and encoder data for propagation and correction of our state vector. Fig. 3 shows

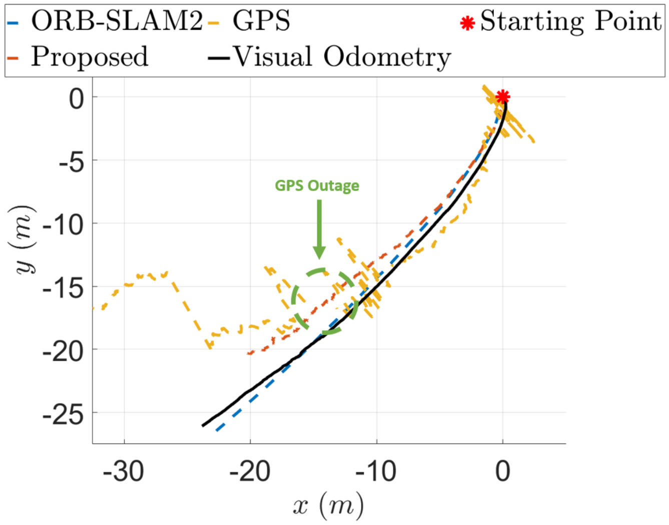

Fig. 3. Trajectory results of the proposed Encoder-Only InEKF algorithm (red). ORB-SLAM2 (blue), GPS data (yellow) and Visual Odometry (**black**) trajectory results are also plotted for comparison. GPS outage (green) also occurs midway through the trajectory making it unreliable.

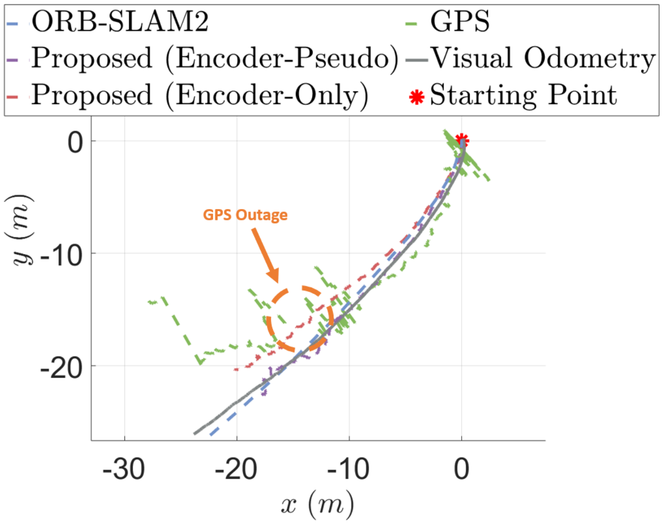

Fig. 4. Trajectory results of the proposed Pseudo-Encoder InEKF algorithm (violet). Encoder-Only InEKF (red), ORB-SLAM2 (blue), GPS data (green) and Visual Odometry (**black**) trajectory results are also plotted for comparison.

the trajectory results of our Encoder-Only algorithm. It can be clearly seen that the latter closely follows the Visual methods initially and drifts off later since the position of the Husky is not observable. The results are still meaningful since the overall drift is less that 10% of the 25 m trajectory where the robot is under constant effect of slip. The estimation results of the state vector are shown in Figs. 5-9. It can be observed that the velocity of the Husky converges to the ZED camera velocity from Visual Odometry and is even less noisy than the latter. Moreover, the position of the Husky drifts away after some time since it is not observable and the orientation is correctly captured up to a deviation of 0.05 rad after some tuning. It is important to note that the angular velocity bias b_ω along y converges fastly to a steady state value of 0.5 rad/s. This algorithm takes around 7 seconds to run on Matlab for the full trajectory.

2) Encoder-Pseudo measurement Results: Similarly, we compare the proposed Pseudo-Encoder InEKF algorithm to the methods previously mentioned as well as our Encoder-Only InEKF algorithm.. Fig. 4 shows that the trajectory resulting from this algorithm provides minimal to no deviation from ORB-SLAM2 and Visual Odometry. The main difference is the sudden changes in direction from our proposed algorithm, thus correctly tracking the slip effect on the Husky robot, making the trajectory shorter compared to the benchmark. Figs. 10-14 show that we are now able to track the orientation R_c and position p_c of the kinematic center of the Husky, previously assumed to be constant. This algorithm takes around 8 seconds to run on Matlab for the full trajectory.

3) Convergence Test Results: We also perform a convergence test on this algorithm to validate the observability analysis. Indeed, Figs. 15-17 shows that for an initial orientation $\psi = 1.13\ rad$ about the z-axis, the orientation ϕ about the x-axis and the orientation θ about the y-axis correctly converge to the ground truth since these states are observable contrary to ψ which is not able recover and track the correct orientation. Finally, the position of the Husky robot diverges since it is not observable. These results align with the observability analysis performed earlier.

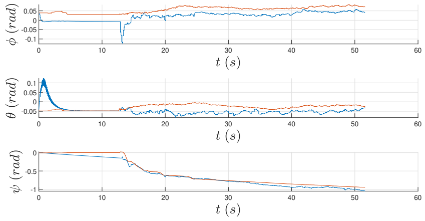

Fig. 5. Orientation Estimation results of the proposed Encoder-Only InEKF algorithm (blue) versus Visual Odometry (**orange**). $\phi(t)$ represents the orientation around the x-axis, $\theta(t)$ represents the orientation around the y-axis and $\psi(t)$ represents the orientation around the z-axis.

4) Comparison with IMU-VO filter: In addition, we also used body velocity of visual odometry (VO) as an alternative source of body velocity. Velocity twist information is obtained by taking difference of poses in visual odometry. The standard IMU-VO filter does not outperform our filter because the measured body velocity is quite noisy as shown in Fig. 6.

VII. Conclusion

This paper proposes a novel Slip-Robust Invariant Extended Kalman Filter for wheeled robots. Two algorithms are implemented and tested on a Husky robot utilizing only IMU and encoder data for correct localization and state estimation. Future work includes CNN integration to predict noise parameters as well as environment-adaptive slip covariance estimation which can be utilized to detect and correctly classify the type of

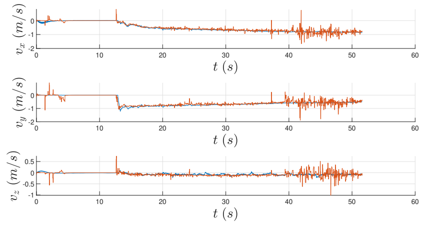

Fig. 6. Velocity Estimation results of the proposed Encoder-Only InEKF algorithm (blue) versus Visual Odometry (**orange**). It can be observed that the proposed method's velocity estimation is smoother than the results from visual odometry.

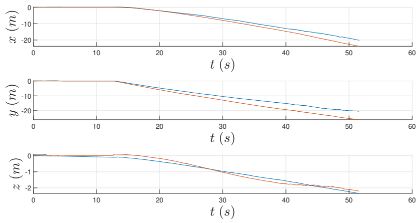

Fig. 7. Position Estimation results of the proposed Encoder-Only InEKF algorithm (blue) versus Visual Odometry (**orange**).

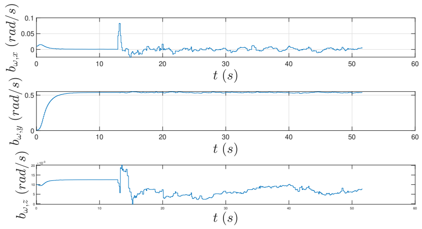

Fig. 8. Angular Velocity Bias $b_\omega(t)$ Estimation results of the proposed Encoder-Only InEKF algorithm.

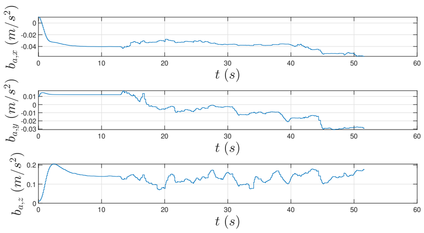

Fig. 9. Linear Acceleration Bias $b_a(t)$ Estimation results of the proposed Encoder-Only InEKF algorithm.

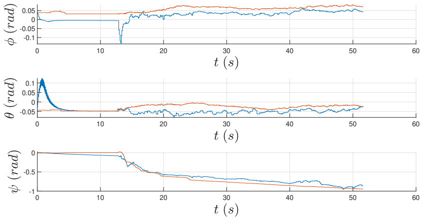

Fig. 10. Orientation Estimation results of the proposed Pseudo-Encoder InEKF algorithm (blue) versus Visual Odometry (**orange**). $\phi(t)$ represents the orientation around the x-axis, $\theta(t)$ represents the orientation around the y-axis and $\psi(t)$ represents the orientation around the z-axis.

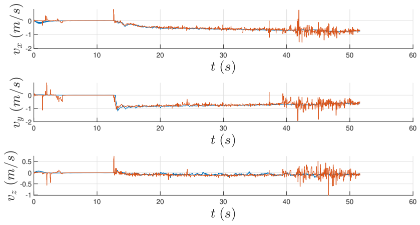

Fig. 11. Velocity Estimation results of the proposed Pseudo-Encoder InEKF algorithm (blue) versus Visual Odometry (**orange**). It can be observed that the proposed method's velocity estimation is smoother than the results from visual odometry.

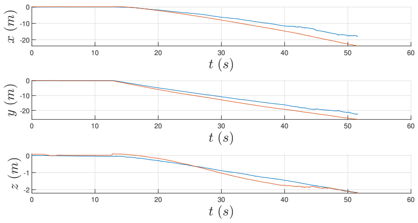

Fig. 12. Position Estimation results of the proposed Pseudo-Encoder InEKF algorithm (blue) versus Visual Odometry (**orange**).

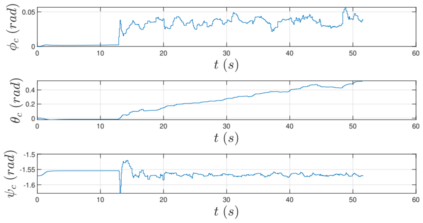

Fig. 13. Kinematic Center Orientation Estimation results of the proposed Pseudo-Encoder InEKF algorithm (blue) versus Visual Odometry (**orange**). $\phi(t)$ represents the orientation around the x-axis, $\theta(t)$ represents the orientation around the y-axis and $\psi(t)$ represents the orientation around the z-axis.

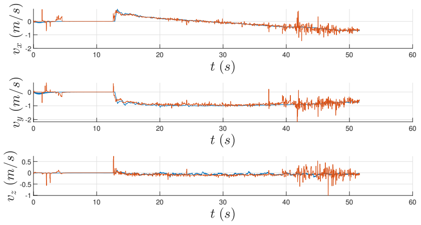

Fig. 16. Velocity Convergence results of the proposed Pseudo-Encoder InEKF algorithm (blue) versus Visual Odometry (**orange**). It can be observed that the proposed method's velocity estimation is smoother than the results from visual odometry.

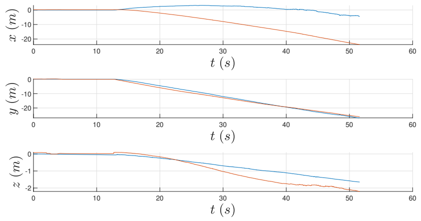

Fig. 17. Position Convergence results of the proposed Pseudo-Encoder InEKF algorithm (blue) versus Visual Odometry (**orange**).

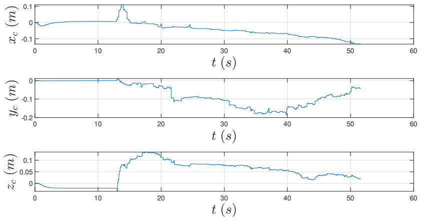

Fig. 14. Kinematic Center Position Estimation results of the proposed Pseudo-Encoder InEKF algorithm (blue) versus Visual Odometry (**orange**).

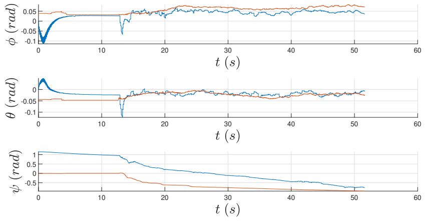

Fig. 15. Orientation Convergence results of the proposed Pseudo-Encoder InEKF algorithm (blue) versus Visual Odometry (**orange**). $\phi(t)$ represents the orientation around the x-axis, $\theta(t)$ represents the orientation around the y-axis and $\psi(t)$ represents the orientation around the z-axis.

terrain the robot is acting on. The latter can render the usage of a camera for detection obsolete. Finally, a SLAM integration of this algorithm would yield better results if implemented since this paper only solves the odometry problem.

Acknowledgment

We would like to thank Prof. Maani Ghaffari, Justin Lin and Sangli Teng for their immeasurable help and support as well as Joseph Wilson and Tingjun Li for collecting data.

References

[1] M. Gallant, J. A. Marshall, and B. K. Lynch, "Estimating the heading of a husky mobile robot with a lidar compass based on direction maps," 2014. 1

[2] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, pp. 194–220, 2017. 1

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013. 1

[4] J.-E. Deschaud, "Imls-slam: Scan-to-model matching based on 3d data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2480–2485. 1

[5] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017. 1

[6] F. Zheng, H. Tang, and Y.-H. Liu, "Odometry-vision-based ground vehicle motion estimation with se(2)-constrained se(3) poses," *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2652–2663, 2019. 1

[7] M. Brossard, A. Barrau, and S. Bonnabel, "Ai-imu dead-reckoning," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 4, pp. 585–595, 2020. 1

[8] F. Zheng and Y.-H. Liu, "Se(2)-constrained visual inertial fusion for ground vehicles," *IEEE Sensors Journal*, vol. 18, no. 23, pp. 9699–9707, 2018. 1

[9] M. Buczko, V. Willert, J. Schwehr, and J. Adamy, "Self-validation for automotive visual odometry," *2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–6, 2018. 1

[10] G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, pp. 731–747, 2001. 1

[11] D. Simon, "Kalman filtering with state constraints: A survey of linear and nonlinear algorithms," *Control Theory Applications, IET*, vol. 4, pp. 1303 – 1318, 09 2010. 1

[12] C. Kilic, J. N. Gross, N. Ohi, R. M. Watson, J. Strader, T. Swiger, S. Harper, and Y. Gu, "Improved planetary rover inertial navigation and wheel odometry performance through periodic use of zero-type constraints," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 552–559, 2019. 1

[13] D.-H. Yi, T.-J. Lee, D.-I. Cho *et al.*, "A new localization system for indoor service robots in low luminance and slippery indoor environment using afocal optical flow sensor based sensor fusion," *Sensors*, vol. 18, no. 1, p. 171, 2018. 1

[14] A. K. Nasir and H. Roth, "Pose estimation by multisensor data fusion of wheel encoders, gyroscope, accelerometer and electronic compass," *IFAC Proceedings Volumes*, vol. 45, no. 4, pp. 49–54, 2012. 1

[15] K. Kaneko, F. Kanehiro, S. Kajita, M. Morisawa, K. Fujiwara, K. Harada, and H. Hirukawa, "Slip observer for walking on a low friction floor," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 634–640. 2

[16] H. Takemura, M. Deguchi, J. Ueda, Y. Matsumoto, and T. Ogasawara, "Slip-adaptive walk of quadruped robot," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 124–141, 2005. 2

[17] X. Zhang, S. Yuan, X. Yin, X. Li, X. Qu, and Q. Liu, "Estimation of skid-steered wheeled vehicle states using stukf with adaptive noise adjustment," *Applied Sciences*, vol. 11, no. 21, p. 10391, 2021. 2

[18] Z. Ullah, Z. Xu, Z. Lei, and L. Zhang, "A robust localization, slip estimation, and compensation system for wmr in the indoor environments," *Symmetry*, vol. 10, no. 5, p. 149, 2018. 2

[19] S. Teng, M. W. Mueller, and K. Sreenath, "Legged robot state estimation in slippery environments using invariant extended kalman filter with velocity update," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3104–3110. 2, 4, 5

[20] A. Barrau and S. Bonnabel, "The invariant extended kalman filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2016. 3, 4

[21] A. Barrau, "Non-linear state error based extended kalman filters with applications to navigation," Ph.D. dissertation, Mines Paristech, 2015. 3

[22] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, "Contact-aided invariant extended kalman filtering for robot state estimation," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020. 4

[23] R. Hartley, M. G. Jadidi, J. W. Grizzle, and R. M. Eustice, "Contact-aided invariant extended kalman filtering for legged robot state estimation," *arXiv preprint arXiv:1805.10410*, 2018. 4