2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025)
October 19-25, 2025. Hangzhou, China

2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 979-8-3315-4393-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IROS60139.2025.11246800

A Robust Distributed Odometry for Mobile Robots with Steerable Wheels

Wang Xi, Jiaming Guo, Chenyang Wang, Shukun Wu and Jianping He

***Abstract*— Odometry estimation remains a critical challenge for wheeled robots, as reducing its drift directly mitigates dependency on external localization systems. This paper proposes a distributed odometry framework for steerable wheels, named ICF-DO, which is applicable to both Steerable Wheeled Mobile Robots (SWMRs) and cooperative multi-single-wheel robot systems. The proposed method features low computational complexity and reduced drift, while demonstrating strong robustness in communication-restricted scenarios. Additionally, singularity can be processed in a distributed manner in the proposed framework. Experimental validation on a real physical SWMR platform demonstrates the effectiveness and practicality of the proposed method.**

I. Introduction

Wheeled robots have found extensive applications across industrial and service domains. Emerging research envisions distributed intelligence architectures, where each wheel module possesses autonomous decision-making capabilities. Such intelligent wheels typically comprise independent drive and steering actuators. When aggregated into multi-wheel chassis systems, this paradigm enhances operational stability, payload capacity, and maneuverability through redundant actuation and adaptive kinematics.

This concept builds upon established frameworks including the Steerable Wheeled Mobile Robot (SWMR) and Pseudo-Omnidirectional Vehicle (POV) architectures, which achieve complex mobility patterns through synchronized steering control. A SWMR consists of multiple steerable wheels that steer around a vertical axis (as shown in Fig.1), enabling omnidirectional motion.

The steering and rotational states of wheels in SWMR inherently encode partial chassis motion, enabling reconstruction of chassis dynamics from wheel state observations. The widely adopted Pacejka Magic Formula [1] provides empirical modeling of tire lateral forces and self-aligning moments during maneuvers. Chassis motion evolves according to constraints of wheel dynamics, and consensus on chassis control objectives among wheels amplifies this implicit coupling. Consequently, wheel encoder-based global odometry demonstrates physical validity when wheels maintain proper kinematic alignment.

Motivation: Despite extensive utilization, the nonlinearity of SWMR dynamics introduces significant challenges, particularly with kinematic singularities [2] and representation singularities [3]. These singularities can lead to wheel misalignment, which disrupts the system's odometry and overall control. Maintaining precise wheel alignment becomes infeasible as the number of wheels increases. Thus, odometry algorithms need to selectively fuse raw sensor data. Existing research has prioritized singularity-compatible control of SWMR and odometry enhancement under such configurations. For non-standard chassis designs with auxiliary actuators, potential field methods offer singularity avoidance through structural optimization [4]. When precise system modeling is achievable, tire friction models enable odometry estimation via dynamic formulations [5]. For conventional SWMR systems, two primary odometry estimation approaches exist. That is, Schwesinger *et al.* [6] employs least-squares optimization to estimate odometry for SWMR with zero wheel turning radius. The Forward Kinematic Actuation Model (FKAM) [7] utilizes damped least squares to circumvent singularities. Alternative methodologies derive odometry through instantaneous center of rotation (ICR) estimation, exemplified by the approach developed by Clavien *et al.* [8] and a singularity-robust virtual-wheel approach [9].

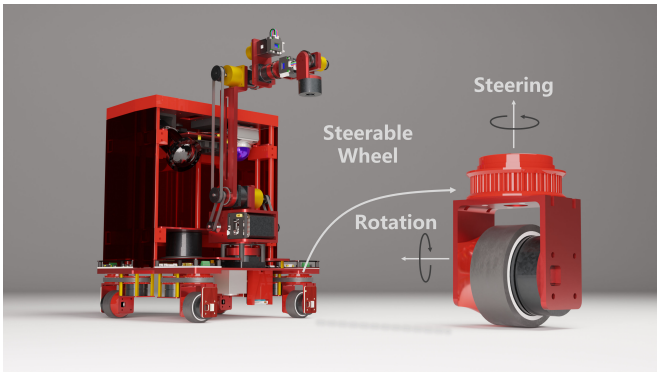


Fig. 1: Illustration of SWMR. The direction of each wheel is steerable.

However, current methods may exhibit some issues: I) insufficient integration of model prediction and multi-sensor data for precise odometry estimation; II) lack of computationally efficient algorithms capable of fully utilizing multi-wheel data streams; III) architectural incompatibility with distributed robotic systems, hindering scalability.

Therefore, the above problems underscore the need for a unified odometry framework that combines model-based constraints with sensor fusion, while supporting distributed implementation. This motivates the study of the distributed odometry framework proposed in this paper.

Challenges: In distributed scenarios, odometry for multi-

The authors are with the Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China. E-mail address: {bddwyx,gan_ju,theoldchun,wsk798,jphe}@sjtu.edu.cn

single-wheel robot system faces some challenges. A single wheel, on its own, cannot control or observe the state of chassis. Coordinated motion and odometry estimation rely on the interaction between multiple wheels. Furthermore, the number of wheels, their locations, and the communication topology between them may vary over time, introducing further complexity to the control and estimation processes. To the best of our knowledge, few attempt has been conducted on distributed odometry for steerable wheels.

Contributions: In this paper, we present a distributed odometry algorithm for steerable wheels, featuring both effectiveness and practicality. Main contributions of this research are as follows:

1) Building upon insights into steerable wheel dynamics, we formulate an odometry estimation model, subsequently validated through field experiments.
2) We develop a distributed odometry framework, demonstrating high adaptability to changes in communication topology, and resilience to communication failures.
3) The novel data fusion strategy achieves over 45% reduction in root mean square error (RMSE) compared to the conventional approach.
4) The odometry requires less than $50\,\mu s$ per iteration on real-time resource-constrained embedded system, while maintaining low positional drift, thereby ensuring its applicability in practical scenarios.

II. Kinematic Modeling

In this section, we analyze the inverse and forward relationships between the wheel states and the chassis state, and introduce the phenomenon of kinematic singularity present in SWMR. The modeling established here is the foundation of the distributed odometry design in the following section.

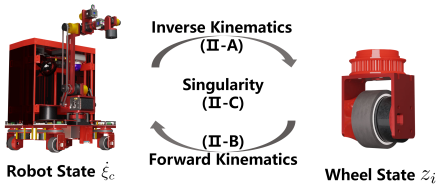


Fig. 2: Organization of Section II.

In subsequent contents, bold symbols denote concatenated vector/matrix quantities. Key notations appear in Table I.

A. Inverse Kinematics Model

Given a chassis velocity, Inverse Kinematics can obtain the desired steer angle and rotational velocity of each wheel. Let $\mathbb{C}$ denotes the coordinate system fixed to the chassis, with chassis center C located at the origin, and $\mathbb{W}$ be a world-fixed inertial coordinate system. Define two variables φ_i and ϑ_i representing the steering angle and rotation angle of the i-th wheel, respectively, where $i \in \{1, 2, \cdots, n\}$.

Let $(v_i^x, v_i^y)^\top$ represent the output velocity of the i-th wheel in the coordinate frame $\mathbb{C}$, with kinematic relation

$$v_i^x = -r\dot{\vartheta}_i \sin\varphi_i, \quad v_i^y = r\dot{\vartheta}_i \cos\varphi_i.$$

TABLE I: Key Notations for Robot and Its State Estimation.

Notation	Definition
$\mathbb{C}$, $\mathbb{W}$	Chassis coordinate, World coordinate
$\xi_{\{c,w\}}$	Robot state in coordinate $\{\mathbb{C}, \mathbb{W}\}$ (3×1)
x, y, ψ	Positions and orientation of robot
φ, ϑ	Steering, Rotation angle of wheel, driven by motors
v_i^x, v_i^y	Output velocity of wheel i in $\mathbb{C}$
z_i	Indirect measurement of wheel i (2×1)
H_i	Observation matrix of z_i
$\hat{\xi}_c$, J_c	Centralized estimation and its information matrix in $\mathbb{C}$
$\hat{\xi}_c^-$, J^-	Prior velocity estimation and its information matrix
$\hat{\xi}_i$, J_i	Estimation of wheel i and its information matrix in $\mathbb{C}$
t, k	Time step and consensus iteration at t

Given a desired robot velocity $\dot{\xi}_c = (\dot{x}_c, \dot{y}_c, \dot{\psi}_c)^\top$ in $\mathbb{C}$, with H_i defined in the equation, the ideal velocity of the i-th wheel follows from rigid body kinematics as

$$\begin{pmatrix} v_i^x \\ v_i^y \end{pmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & -l_i^y - d\sin\varphi_i \\ 0 & 1 & l_i^x + d\cos\varphi_i \end{bmatrix}}_{H_i} \dot{\xi}_c + d \begin{pmatrix} -\sin\varphi_i \\ \cos\varphi_i \end{pmatrix} \dot{\varphi}_i. \tag{1}$$

The Inverse Kinematics Model (1) reveals the relationship between the desired wheel states and the chassis velocity, forming the basis for SWMR control. However, odometry estimation requires reconstructing the chassis trajectory from the wheel states. We therefore introduce Modified FKAM - a forward kinematics approach - in the following subsection.

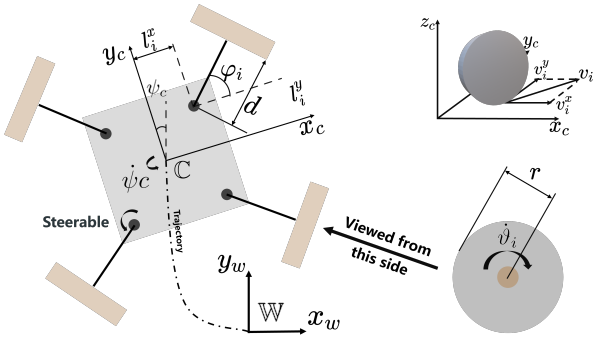


Fig. 3: Schematic Model of SWMR with $d \neq 0$.

B. Modified FKAM

FKAM [7] is a widely-applied forward kinematic model. However, it solely relies on the steering angle φ_i while neglecting the rotation speed $\dot{\vartheta}_i$. In this subsection, we refine FKAM by adopting its core insight and derive a clearer formulation, named Modified FKAM.

Define auxiliary variable z_i by

$$z_i = \begin{pmatrix} v_i^x + d\dot{\varphi}_i \sin\varphi_i \\ v_i^y - d\dot{\varphi}_i \cos\varphi_i \end{pmatrix}. \tag{2}$$

Since the steering angle φ_i and rotation angle ϑ_i can be measured using motor encoders, z_i can be treated as an indirect measurement of the chassis velocity $\dot{\xi}_c$.

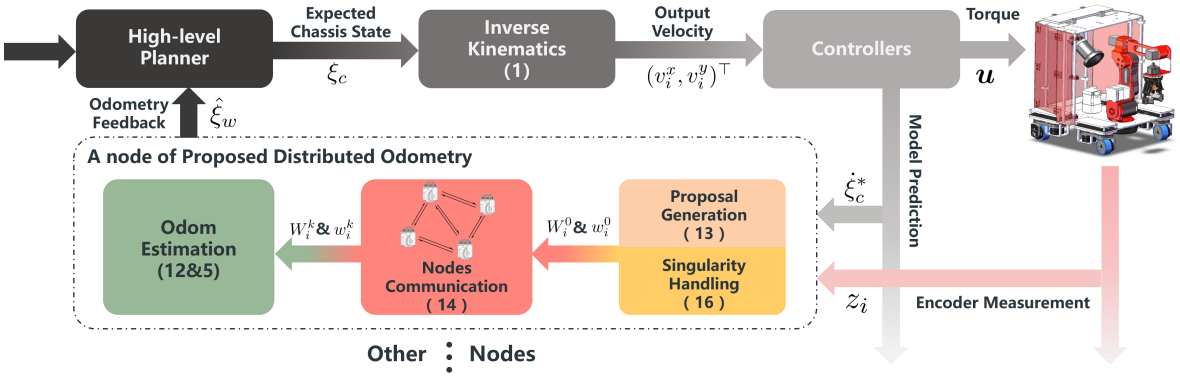


Fig. 4: System Block Diagram and Architecture of the Proposed Odometry Framework.

(1) can be rewitten as $z_i = H_i\dot{\xi}_c$. Concatenating the measurement values $\boldsymbol{z}_{2n\times 1} = (z_1^\top, z_2^\top, \ldots, z_n^\top)^\top$ and the observation matrix $\boldsymbol{H}_{2n\times 3} = (H_1^\top, H_2^\top, \ldots, H_n^\top)^\top$, the chassis velocity $\dot{\xi}_c$ can be estimated by

$$\hat{\dot{\xi}}_c = \boldsymbol{H}^\dagger(\varphi_1, \varphi_2, \cdots, \varphi_n)\,\boldsymbol{z}, \tag{3}$$

where $\boldsymbol{H}^\dagger$ represents the damped pseudoinverse of $\boldsymbol{H}$, i.e.,

$$\boldsymbol{H}^\dagger = \left(\boldsymbol{H}^\top\boldsymbol{H} + \lambda^2 I_{3\times 3}\right)^{-1}\boldsymbol{H}^\top,$$

with $\lambda \in \mathbb{R}$ the damping factor. The damped pseudoinverse not only addresses the issue of $\boldsymbol{H}^\top\boldsymbol{H}$ being rank-deficient in certain cases, but also enhances the robustness of the velocity estimation model against disturbances.

However, due to the time-varying nature of H_i, equation (3) incurs high computational costs. Additionally, prior information on the control inputs should also be leveraged for odometry estimation. Thus, in Sec. III, we propose a novel distributed odometry framework.

C. Kinematics Singularity Analysis

Singularity is a common issue in nonlinear robotic systems, occurring when certain desired trajectories lead to some expected outputs approaching infinity. In the context of SWMR, the phenomenon of singularity also appears.

Under the *no lateral skidding* assumption, each wheel is subject to the constraint that lateral component of wheel speed equals zero, i.e., $f_i(\varphi_i)^\top\dot{\xi}_c = 0$, where

$$f_i(\varphi_i) = \begin{bmatrix}\cos\varphi_i & \sin\varphi_i & l_i^x\sin\varphi_i - l_i^y\cos\varphi_i\end{bmatrix}^\top. \tag{4}$$

According to Equation (4) in [10], the relationship between the steering speed $\dot{\varphi}_i$ and the desired chassis state satisfies

$$\dot{\varphi}_i = \frac{-f_i(\varphi_i)\ddot{\xi}_c}{\frac{\mathrm{d}f_i(\varphi_i)}{\mathrm{d}\varphi_i}\dot{\xi}_c}. \tag{5}$$

As the denominator approaches zero, the required steering speed $\dot{\varphi}_i$ approaches infinity, causing the chassis to fail in tracking the desired trajectory. Therefore, singularity is a critical issue in SWMR control, and improper handling can lead to catastrophic result.

III. Distributed Odometry Design

A. Formulation of Distributed Odometry

1) System Dynamics: The exact system dynamics from motor input torques to ξ_c are intrinsically intractable, involving complex mechanical processes and unknown parameters. However, leveraging the predictive nature of controllers (e.g., LQR), we obtain a controller-predicted state $\dot{\xi}_c^*(t)$. Assuming the prior velocity estimation linearly combines previous velocity and model prediction of the controller, and the residual between real and approximated dynamics is treated as process noise $\boldsymbol{\gamma}_\xi$ with covariance Q. Consequently, the dynamics and observation model can be formulated by

$$\dot{\xi}_c(t+1) = \alpha\dot{\xi}_c(t) + (1-\alpha)\dot{\xi}_c^*(t) + \boldsymbol{\gamma}_\xi \tag{6a}$$

$$\boldsymbol{z} = \boldsymbol{H}\dot{\xi}_c + \boldsymbol{\gamma}_z, \tag{6b}$$

where wheel measurement errors (deviations from real velocities) are characterized by $\boldsymbol{\gamma}_z$ with zero mean and block-diagonal covariance $\boldsymbol{R} = \mathrm{diag}(R_1, R_2, \ldots, R_n)$.

Remark 1: Compensating for prediction latency, (6a) demonstrates practical efficacy. The parameter $\alpha \in [0, 1]$ is empirically determined, with $\alpha = 1$ reducing the method to a stationary state estimator.

2) Centralized Odometry: Before introducing the distributed formulation, we first derive the centralized estimator. Let $J_c = \mathrm{Cov}^{-1}(\hat{\dot{\xi}}_c)$ be the information matrix of the velocity estimate. The prior state estimate and its corresponding information matrix are denoted by $\hat{\dot{\xi}}_c^-$ and J^-, respectively. According to the Kalman filter framework, the maximum a posteriori estimate [11] at time t in the information form is formulated by

$$\begin{cases}\hat{\dot{\xi}}_c = J_c^{-1}\left(J^-\hat{\dot{\xi}}_c^- + \boldsymbol{H}^\top\boldsymbol{R}^{-1}\boldsymbol{z}\right)\\ J_c = \left(J^- + \boldsymbol{H}^\top\boldsymbol{R}^{-1}\boldsymbol{H}\right)\end{cases}, \tag{7}$$

with prior estimate equation

$$\begin{cases}\hat{\dot{\xi}}_c^-(t) = \alpha\hat{\dot{\xi}}_c(t-1) + (1-\alpha)\dot{\xi}_c^*(t-1)\\ J^-(t) = \left(\alpha^2\left(J_c(t-1)\right)^{-1} + Q\right)^{-1}\end{cases}, \tag{8}$$

where at time $t = 0$, $\hat{\dot{\xi}}_c^-(0)$ and $J^-(0)$ can be initialized following the classical Kalman Filter procedure. Equations (7) and (8) represent centralized and Kalman-filter-like forms of the proposed method.

We now consider the centralized estimation problem as a process in which all wheels collectively reach a consensus on the optimal estimate. Thus, we need to reformulate the centralize form of odometry.

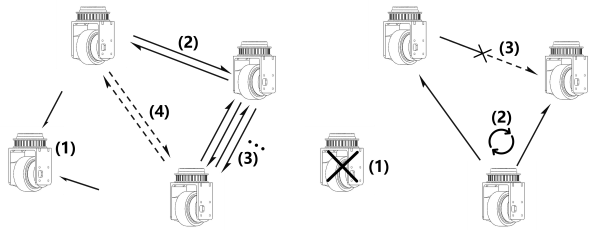
3) Reformulation of Centralized Odometry: Let $\hat{\dot{\xi}}_i$ denote the state estimate of the i-th wheel. The prior estimate and the corresponding information matrix for the i-th wheel, according to (8), are denoted as $\hat{\dot{\xi}}_i^-$ and J_i^-, respectively. By referring to (26) and (27) in [12], and assuming that each wheel has reached a consensus on the prior estimates $\hat{\dot{\xi}}_i(t-1)$. Then, using the diagonal property of $\boldsymbol{R}$, a decoupled manner of (7) at time t is

$$\begin{cases} \hat{\dot{\xi}}_c = J_c^{-1} \sum_{i=1}^{n} \left(\frac{J_i^-}{n} \hat{\dot{\xi}}_i^- + H_i^\top R_i^{-1} z_i \right) \\ J_c = \sum_{i=1}^{n} \left(\frac{J_i^-}{n} + H_i^\top R_i^{-1} H_i \right) \end{cases}. \tag{9}$$

Remark 2: Although (9) appears to provide the required distributed odometry, real-world deployment presents additional challenges. Communication between wheels is not as straightforward as in a centralized discussion, time-varying communication topologies and bandwidth constraints must be considered. In the next subsection, we explore the distributed implementation of this process.

B. Implementation of Distributed Odometry

In the field of sensor networks, consensus-based distributed state estimation has been widely studied [13], [14]. We selected ICF [12] as the baseline for our odometry.



(a) Communication Features (b) Challenges for Odometry

Fig. 5: Features and Challenges of Distributed Odometry. (a1) Receives data from others; (a2) Bidirectional communication, easily implemented via broadcasting; (a3) Multiple data exchanges per cycle; (a4) Fully connected topology is unnecessary. (b1) Node failure; (b2) Time-varying topology; (b3) Data loss.

Considering the structure of (9), we define two auxiliary variables respectively by

$$\begin{cases} w_i^0 = \frac{J_i^-(t)}{n} \dot{\xi}^-(t) + H_i^\top R_i^{-1} z_i(t) \\ W_i^0 = \frac{J_i^-(t)}{n} + H_i^\top R_i^{-1} H_i \end{cases}. \tag{10}$$

Clearly, these variables satisfy

$$\sum_{i=1}^{n} W_i^0 = J_c, \quad J_c^{-1} \sum_{i=1}^{n} w_i^0 = \hat{\dot{\xi}}_c.$$

The subsequent challenge lies in developing a distributed computation framework for (10) without centralized coordination or communication.

Let $\mathcal{N}_i(t)$ denote the set of wheels directly communicating with the i-th wheel. The distributed computation is achieved via average consensus algorithm, referring to [13]. The iteration process of our distributed odometry is designed as

$$\begin{cases} w_i^k = w_i^{k-1} + \epsilon \sum_{j \in \mathcal{N}_i} \left(w_j^{k-1} - w_i^{k-1} \right) \\ W_i^k = W_i^{k-1} + \epsilon \sum_{j \in \mathcal{N}_i} \left(W_j^{k-1} - W_i^{k-1} \right) \end{cases}, \tag{11}$$

where $\epsilon \in \mathbb{R}^+$ denotes a topology-dependent constant.

Clearly, (11) is one kind of classical average consensus processes. Thus, according to the classical average consensus convergence theory [13], as $k \to \infty$, (11) gives results that

$$\lim_{k \to \infty} \left(W_i^k \right)^{-1} w_i^k = \hat{\dot{\xi}}_c, \quad \lim_{k \to \infty} n W_i^k = J_c$$

as formalized in (9). Consequently, through average consensus propagation of the auxiliary variables w_i and W_i, we achieve fully distributed estimation of the system state $\hat{\dot{\xi}}_c$ without centralized coordination.

C. Singularity Handling

In Sec. II-C, we analyze the negative impact of singularity on control. Thus, to obtain accurate odometry data, it is essential to handle the encoder data from the affected wheels.

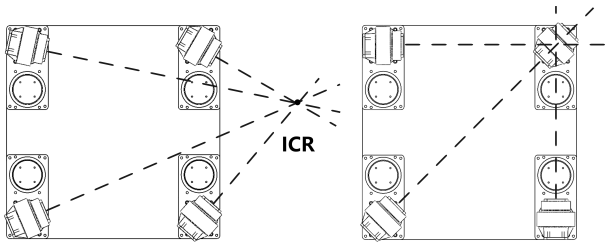


Fig. 6: Singularity from the ICR Perspective. The right subfigure illustrates a singular configuration when $d = 0$.

From another perspective, singularity occurs when the ICR coincides with the center of wheel on the chassis plane. See Figure 6 as an example. Specifically, according to rigid

body kinematics, given an ICR location, the expected output velocity of i-th wheel can be simply calculated as

$$\begin{pmatrix} v_i^x \\ v_i^y \end{pmatrix} = \dot{\psi}_c \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \underbrace{\begin{pmatrix} l_i^x + d\cos\varphi_i - x_{\text{ICR}} \\ l_i^y + d\sin\varphi_i - y_{\text{ICR}} \end{pmatrix}}_{d_i^{\text{ICR}}}. \tag{12}$$

When ICR approaches the center axis of a wheel, prohibitively high steering speeds are required, thus the control performance may be compromised. Given the mutually exclusive operational regions of individual wheels, at most one wheel is affected by singularity at any given moment. Consequently, the odometry only requires to process potentially erroneous measurements originating from a singular wheel.

According to rigid body kinematics, the relation between ICR location and vehicle velocity $\dot{\xi}_c$ can be described by

$$\begin{pmatrix} x_{\text{ICR}} \\ y_{\text{ICR}} \end{pmatrix} = \frac{1}{\dot{\psi}_c} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \dot{x}_c \\ \dot{y}_c \end{pmatrix}, \tag{13}$$

which indicates that each wheel can determine the current ICR position. Using the above equation, distributed singularity detection can be achieved.

Within the ICF framework, the presence of naive nodes is allowed, meaning that the lack of measurements from some nodes at any given time does not affect the overall distributed estimation. The smaller the distance $\|\boldsymbol{d}_i^{\text{ICR}}\|$ (defined in (12)), the lower the measurement quality. Thus, we can select an appropriate threshold d_{min} and exclude the measurements from wheels with $\|d_i^{\text{ICR}}\| \leq d_{\text{min}}$. For such wheels, the measurement information matrix is set to $R_i^{-1} = \mathbf{0}$. Thus, we design Algorithm 1, named ICF-DO, as the proposed distributed odometry in this research.

D. Algorithm Analysis

1) Effectiveness: Compared to Modified FKAM (3), the ICF-based velocity estimation leverages more prior model information, thus demonstrates lower drifting rate compared to conventional methods. Notably, even in scenarios where prior model information is unavailable, reliable odometry can still be achieved by configuring $\alpha = 1$.

2) Convergence and Robustness: Let $\mathcal{G}$ denote the communication graph and $\mathcal{L}(\mathcal{G})$ be its corresponding Laplacian matrix, with eigenvalues $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$. Define

$$\overline{W} = \frac{1}{n}\sum W_i^0, \quad \overline{w} = \frac{1}{n}\sum w_i^0,$$

and $\rho = 1 - \epsilon\lambda_2$. According to discrete-time consensus theory [14], when the maximum degree Δ_{max} of $\mathcal{G}$ and ϵ satisfies $0 < \epsilon < \frac{1}{2\Delta_{max}}$, the errors exhibit exponential convergence

$$\|W_i^k - \overline{W}\| = \mathcal{O}(\rho^k), \quad \|w_i^k - \overline{w}\| = \mathcal{O}(\rho^k).$$

Since only the information matrix J_i depends on the number of wheels n, the odometry remains operational even when partial nodes have erroneous perceptions of n, as numerically validated in [12]. Furthermore, Experiment J in [12] demonstrates that the method maintains functionality under model uncertainties.

Algorithm 1 ICF-based Distributed Odometry (ICF-DO)

Input: Direct measurement v_i and model prediction $\dot{\xi}^*(t)$ from the controller.

Initialize measurement z_i using (2) and consensus iteration counter $k = 0$.

Compute prior state estimate $\dot{\xi}_i^-(t)$ and information matrix $J_i^-(t)$ as follows:

$$\dot{\xi}_i^-(t) = \alpha\hat{\dot{\xi}}(t-1) + (1-\alpha)\dot{\xi}^*(t-1),$$
$$J_i^-(t) = \left(\alpha^2 \left(J_i(t-1)\right)^{-1} + Q\right)^{-1}.$$

if $\|\boldsymbol{d}_i^{\text{ICR}}\| \leq d_{min}$ **then**
  Set $R_i^{-1} = \mathbf{0}$ temporarily for this step.
end if
Generate consensus proposal W_i^0 and w_i^0 by (10).
Start broadcasting the latest consensus proposal W_i^k and w_i^k until the timer interrupt occurs.
while Timer interrupt has not occurred **do**
  if A packet from neighboring node j is received **then**
    Increment k: $k \leftarrow k + 1$.
    Perform average consensus by (11).
  end if
end while
Terminate proposal broadcasting.
Compute local state estimate $\hat{\dot{\xi}}_i(t)$ and the corresponding information matrix $J_i(t)$ as follows:

$$\hat{\dot{\xi}}_i(t) = \left(W_i^k\right)^{-1} w_i^k,$$
$$J_i(t) = nW_i^k.$$

Numerically integrate global state ξ_w

$$\hat{\xi}_w = \int_0^t \mathcal{R}_c^w(\hat{\xi}_w(\tau))\hat{\dot{\xi}}_c(\tau)\,\mathrm{d}\tau$$

via Runge-Kutta method.

Output: Global state estimation $\hat{\xi}_w$.

3) Practicality: In large-scale deployment scenarios, the dominant computational cost of proposed method stems from average consensus, which scales as $\mathcal{O}(\Delta_{max}k)$. This complexity characterization demonstrates the proposed algorithm's capability to maintain real-time performance in practical implementations.

The proposed method is well-suited for real-time embedded computation. In practical scenarios, average consensus cannot run infinitely, and a balance must be struck between the number of iterations and the odometry computation time. Since the odometry update needs to be completed before the controller computation within the corresponding control period, data exchanges between nodes can continue until the end of the control period. The odometry computation is then performed at the beginning of the next period.

As shown in Fig.7, after generating the consensus proposal based on the encoder data (10), nodes can fully utilize the communication bandwidth to execute the average consensus

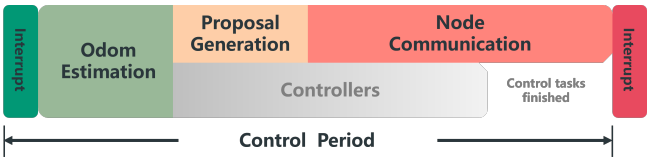


Fig. 7: Operations within Each Control Period. Each period begins with a timer interrupt, during which control computation and communication reception occur simultaneously. Consensus updates continue until the next timer interrupt.

until timer interrupt signals the start of next period. This ensures that the controller can use the latest odometry data while maximizing the estimation accuracy by fully utilizing communication bandwidth.

IV. Real-world Experiments

A. Experimental Settings

Real-world experiments were conducted on a self-designed POV platform, *Fines*. *Fines* consists of four wheels positioned at the corners of a 240 mm square. The wheel's rotational axis is aligned with its center, i.e., $d = 0$. The wheels' steering and rotation are controlled by integrated FOC driver boards, with encoder precision of 14 bits. Conductive slip rings enable infinite wheel rotation without wire twisting. All motors are controlled via an STM32F446-based main control board, using RS485/CAN bus communication. The embedded control framework employed is *FineMote*.

The robot's ground truth reference position is provided by the *FZMotion* Motion Capture System and transmitted to the robot via MQTT. All relevant data is collected by the MCU, exported as Comma-Separated Values (CSV) files using STM32CubeMonitor, and visualized using MATLAB. Parameter α in (8) is set to 0.5 in the following experiments.

B. Comparison with Modified FKAM

The first experiment compares the proposed method with the centralized Modified FKAM under a fully connected topology. The primary comparison metric is the root mean square error (RMSE) between the estimated trajectories of both odometry methods and the positioning data provided by the Motion Capture System, serving as a benchmark for evaluating odometry accuracy. The error of state ξ_w is evaluated in two components: the distance error for the first two dimensions and the angular error for the third dimension.

As illustrated in Fig. 8 (localization error distribution) and Table II (RMSE comparison), the proposed method demonstrates reduced drift and sustained odometry accuracy. While covariance parameter selection leads to increased mean square error in orientation estimation, positional localization remains minimally affected by angular error.

C. Robustness Test

In this experiment, we simulated the performance of odometry in an environment with poor communication and compared it to the ideal case of a fully connected topology.

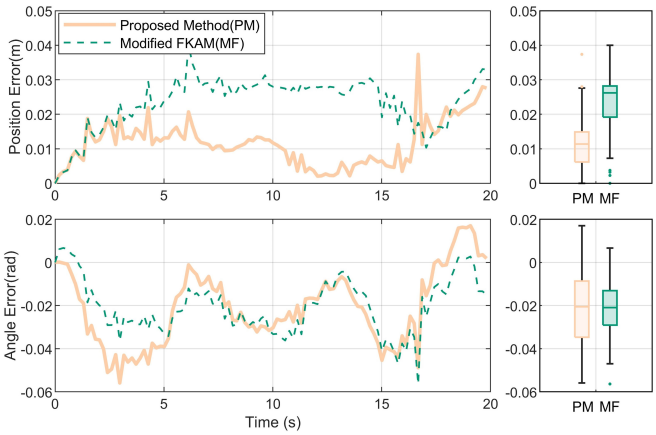


Fig. 8: Results of Comparison between the Two Methods.

TABLE II: RMSE of Position and Angular Error between the Two Methods (Average of 10 Trials)

	RMSE(Position)	**RMSE**(Angle)
Modified FKAM	24.6 mm	1.52×10^{-2}rad
Proposed Method	13.4 mm	1.94×10^{-2}rad

During each control period, each wheel had only one opportunity for unidirectional data transmission with its neighbors, with a success rate of just 5%. To simulate more extreme conditions, the second wheel was completely disconnected at $t = 10$ s. For generality and to clearly demonstrate the robustness, gaussian noise with a standard deviation of 0.05 m/s was added to the velocity data obtained from the high-precision encoders, which serves as z_i.

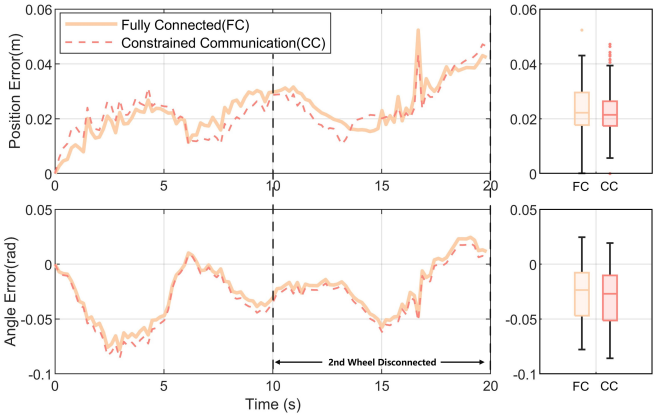


Fig. 9: Results of Robustness Test.

As shown in Fig. 9, the limited communication did not significantly affect the performance of the odometry. Even after one wheel was fully disconnected, the odometry still produced results nearly identical to those under ideal communication conditions.

D. Embedded Deployment Experiment

This experiment evaluates the real-time performance of the proposed method, deployed on the *FineMote* embedded

framework with CMSIS-DSP acceleration package, where four MCU code segments centrally simulate distributed odometry estimation across four wheels. A Hermite interpolation curve with 7 knots is used as the reference trajectory. The chassis state ξ_w is regulated via full-state feedback control based on odometry-tracking errors.

TABLE III: Knots Description of the Hermite Curve

Knot	x_w(m)	y_w(m)	ψ_w	$\dot{x}_w$(m/s)	$\dot{y}_w$(m/s)	$\dot{\psi}_w$
1	0	0	0	0	0	0
2	-1.0	1.0	$\pi/2$	-0.05	0	0
3	-1.5	0.5	$\pi/4$	0	-0.05	0
4	-1.0	0	0	0.05	-0.05	0
5	-0.5	-0.5	$-\pi/4$	0	-0.05	0
6	-1.0	-1.0	$-\pi/2$	-0.05	0	0
7	-2.0	0	0	0	0	0

Performance is quantified by two metrics: (1) the RMSE of the actual trajectory relative to the ideal reference, and (2) the computational latency of a single odometry update.

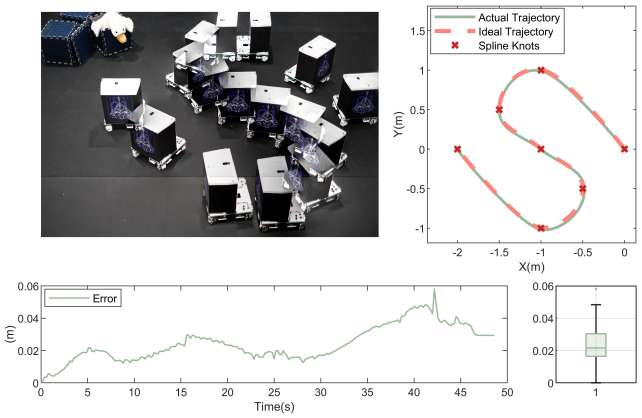


Fig. 10: Reference Trajectory and Path-following Results.

In our experiments, the proposed method maintained a low drift rate (under 0.5% path-length normalized error) while achieving high tracking accuracy (RMSE < 3 cm) with a basic controller. The complete computational pipeline - encompassing state estimation and consensus computation - requires less than 50 µs per iteration, representing only 5% of the available time budget in 1 kHz control applications.

TABLE IV: Results of Embedded Deployment

Metric	RMSE	Computational Cost
Value	26.6 mm	48.44 µs

V. Conclusions

This study proposes a distributed odometry method for robots with steerable wheels. Under the ICF framework, the method effectively combines controller model predictions with wheel encoder measurements, demonstrating superior performance over conventional FKAM method.

As a distributed odometry, the proposed method achieves robustness under challenging conditions including packet loss and sensor fails. The lightweight computation allows real-time implementation on embedded platforms with minimal resource consumption, as verified through physical experiments on a real SWMR platform.

References

[1] H. B. Pacejka and E. Bakker, "The magic formula tyre model," *Vehicle System Dynamics*, vol. 21, no. S1, pp. 1–18, 1992.

[2] B. Thuilot, B. d'Aandrea Novel, and A. Micaelli, "Modeling and feedback control of mobile robots equipped with several steering wheels," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 3, pp. 375–390, 1996.

[3] P. R. Giordano, M. Fuchs, A. Albu-Schaffer, and G. Hirzinger, "On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[4] A. Dietrich, T. Wimböck, A. Albu-Schäffer, and G. Hirzinger, "Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[5] K. Berntorp, B. Olofsson, and A. Robertsson, "Path tracking with obstacle avoidance for pseudo-omnidirectional mobile robots using convex optimization," in *American Control Conference (ACC)*, 2014.

[6] U. Schwesinger, C. Pradalier, and R. Siegwart, "A novel approach for steering wheel synchronization with velocity/acceleration limits and mechanical constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[7] M. Sorour, A. Cherubini, R. Passama, and P. Fraisse, "Kinematic modeling and singularity treatment of steerable wheeled mobile robots with joint acceleration limits," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[8] L. Clavien, M. Lauria, and F. Michaud, "Instantaneous centre of rotation estimation of an omnidirectional mobile robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[9] C. Stöger, H. Gattringer, and A. Müller, "The virtual wheel concept for the singularity-free kinematics and dynamics modeling and control of pseudo-omnidirectional vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4798–4804, 2021.

[10] M. Sorour, A. Cherubini, P. Fraisse, and R. Passama, "Motion discontinuity-robust controller for steerable mobile robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 452–459, 2016.

[11] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[12] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, 2013.

[13] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[14] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.