Mobile Robot Localization and Prediction by UKF with Neural Network Aided Modeling

Yongyue Xu[1,2], Jia Chen[1,2], Jinya Su[1,2,3], Yunda Yan [4], Shihua Li[1,2,3]
[1]*School of Automation,* Southeast University, Nanjing, China
[2]*Key Laboratory of Measurement and Control of CSE, Ministry of Education, Southeast University, Nanjing, China*
[3]*Institute of Intelligent Unmanned Systems, Southeast University, Nanjing 210096, China.*
[4]*Department of Computer Science, University College London, London WC1E 6BT, UK*

***Abstract*—State estimation is crucial for mobile robot navigation, enabling tasks like motion planning and obstacle avoidance. Global Positioning System (GPS) signals are unreliable indoors due to obstructions, Ultra-Wideband (UWB) systems offer a robust alternative for indoor positioning. However, traditional IMU-UWB integration methods suffer rapid degradation in accuracy when low-frequency UWB measurement data is unreliable, primarily due to their dependence on the IMU kinematic model. In this paper, we propose an enhanced localization framework that synergizes kinematic modeling with neural network-aided techniques to enhance the IMU process model under GPS-denied and low-frequency measurement conditions. By embedding the improved model within an Unscented Kalman Filter (UKF), our approach markedly improves both single-step and multi-step prediction accuracy. Comparatively experimental results using a Mecanum-wheeled mobile robot demonstrate that the proposed system not only improves localization precision but also enhances robustness, providing a reliable solution for indoor 2D mobile robot navigation. The dataset and code are available at: https://github.com/YyX-ssr/MLP-Aided-UKF**

***Index Terms*—Neural Network, Learning Aided Modeling, Robot Localization, UKF**

I. Introduction

Recent advances in robotics have automated tasks traditionally performed by humans, such as visual inspection [1], [2]. In these systems, precise state estimation is essential for navigation and obstacle avoidance, and a variety of sensors, including cameras, radar, LiDAR, the Global Positioning System (GPS), and Inertial Measurement Units (IMU), support this task. However, sensor performance often degrades due to uncertain environmental factors. For example, although GPS provides absolute position data, its signals are susceptible to blockage, multipath effects, and shading, rendering it unreliable in indoor or GPS-denied environments [3]. Similarly, while Ultra-Wideband (UWB) is well-suited for indoor applications, it is prone to measurement loss from interference. In contrast, IMU-based inertial navigation systems (INS) resist external interference and consume low power, making them ideal for integration with UWB in indoor navigation [4], [5].

*This work was supported by the National Natural Science Foundation of China under Grant 62303110, the Start-Up Research Fund of Southeast University under Grant RF1028623226, and the Taihu Lake Innovation Fund for the School of Future Technology of Southeast University. Corresponding author: Prof. Jinya Su. (sucas@seu.edu.cn)

An IMU comprises a tri-axial accelerometer and gyroscope that continuously record the body's acceleration and angular velocity. Using a kinematic model, these measurements yield estimates of attitude, velocity, and position, providing crucial components for navigation systems. However, low-cost Micro-Electro-Mechanical Systems (MEMS) IMUs are vulnerable to various error sources, including bias, noise, misalignment, and scale factor inaccuracies [6]. In conventional inertial navigation, carrier attitude is obtained by integrating angular velocity, and position is derived by double integrating the transformed acceleration. Even minor errors quickly accumulate during integration, causing unbounded drift and localization failure within seconds.

IMU data is often fused with absolute measurements using a Kalman Filter (KF) [7]. KF is popular in state estimation, target tracking, and navigation because of its optimality for linear systems with Gaussian noise and its straightforward implementation [8], [9]. Its estimation process generally comprises two steps: a prediction step based on the system's process model and an update step that refines the state using observation data [10]. In integrated navigation systems, the IMU's kinematic model typically serves as the process model, while GPS/UWB measurements are used as observations to update and correct the predicted state. However, since the IMU kinematic model is inherently nonlinear, nonlinear KF variants such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) are generally employed for improved accuracy [11].

An accurate process model is critical for enhancing integrated navigation, particularly in situations where measurement data is unreliable. Classical kinematic models, grounded in Newtonian physics, are often undermined by real-world IMU measurement errors, such as bias, noise, and misalignment, which significantly contribute to rapid inertial navigation drift [12]. Prior efforts to mitigate this drift have focused on specific scenarios. For example, in pedestrian dead reckoning, the inherent periodicity of human walking is leveraged to detect steps and estimate stride length, thereby reducing error accumulation [13]. Similarly, the zero-velocity update (ZUPT) method, which detects stationary periods using an IMU on a pedestrian's foot, can be incorporated as observational data into the KF framework to enhance accuracy [14]. However,

these strategies typically depend on context-specific assumptions that limit their broader applicability.

More recently, machine learning has shown promise in overcoming the limitations of conventional inertial navigation. IONet [15] pioneered this approach by employing a Long Short-Term Memory (LSTM) network to learn location transformations from raw IMU data in polar coordinates, while RoNIN [16] adopts architectures based on ResNet, LSTM, and Temporal Convolutional Networks (TCN) to regress velocity. Other studies have focused on calibrating inertial sensors via neural networks to achieve higher measurement accuracy [17], [18]. Additionally, some works integrate learning-based models with the KF framework; for instance, TLIO [19] uses neural networks to estimate both displacement and uncertainty end-to-end for the KF update step, and in [20], a TCN is utilized to regress an agricultural robot's linear velocity for incorporation into the prediction step of an EKF. Despite the superior feature extraction capabilities of recurrent neural networks (e.g., LSTM and GRU) and TCNs, these approaches can undermine the Markov assumption of the KF by introducing cross-correlations among time-series inputs, complicating the accurate determination of the covariance matrix.

Although learning-based modeling methods have demonstrated potential, they suffer from poor interpretability. In this study, we propose a hybrid modeling approach that combines deep learning with a mechanism-based model to improve the interpretability of the system and enhance the state prediction capability of the IMU model. Furthermore, we incorporate this predictive model into an Unscented Kalman Filter (UKF) framework to improve traditional integrated navigation, where the proposed framework is shown in Fig. 1. In light of the Markov assumption, we opted not to employ recurrent network architectures such as LSTM or GRU. Instead, we utilized a Multilayer Perceptron (MLP) that regresses the changes in linear velocity and orientation angles from the last step's state variables and IMU measurements.

Although learning-based modeling methods have demonstrated potential, they suffer from limited interpretability. In this study, we propose a hybrid modeling approach that fuses deep learning with a kinematic-based model to improve system transparency and enhance the state prediction capability of the IMU process model. Furthermore, we incorporate this predictive model into an Unscented Kalman Filter (UKF) framework to advance traditional integrated navigation, as shown in Fig. 1. In light of the Markov assumption, we opted against using recurrent network architectures such as LSTM or GRU and instead utilized a Multilayer Perceptron (MLP) that regresses changes in linear velocity and orientation angles from the previous step's state variables and IMU measurements. Our main contributions are summarized as follows:

(1) MLPs are employed to model variations in linear velocity and orientation, and their output is integrated with the traditional model to yield an enhanced process model that enhances model predictive performance.
(2) Considering the process model's nonlinearity, we fuse it with UKF for integrated navigation, improving both accuracy and robustness of state estimation.
(3) Using a Mecanum-wheeled robot, our method is validated to outperform traditional models in model prediction and indoor positioning accuracy (velocity in particular).

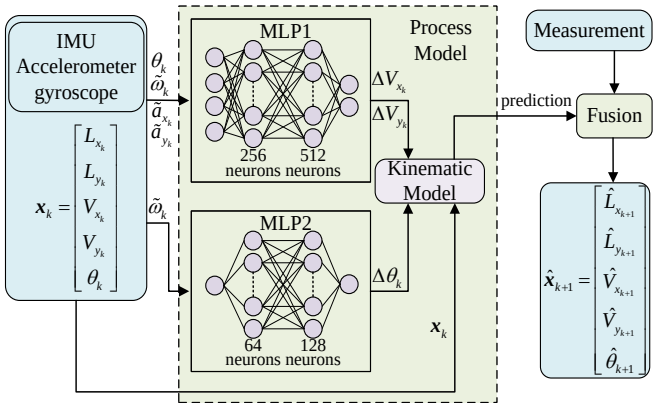

Fig. 1. Framework of the proposed learning-aided state estimation

II. Preliminaries

A. INS Error Analysis

MEMS IMU measures the acceleration $\boldsymbol{a}$ and the angular velocity $\boldsymbol{\omega}$ of the body. However, these measurements suffer from different errors [17],

$$\tilde{\boldsymbol{u}} = (\boldsymbol{S} + \boldsymbol{N})\boldsymbol{u} + \boldsymbol{b} + \boldsymbol{\eta} \tag{1}$$

where $\tilde{\boldsymbol{u}} \in R^3$ and $\boldsymbol{u} \in R^3$ represent measurement and ground truth of an IMU; $\boldsymbol{S} \in R^{3\times3}$ is the ratio between the measurement and the truth, named as scale factor; $\boldsymbol{N} \in R^{3\times3}$ denotes axis-misalignment error; $\boldsymbol{b} \in R^3$ is zero bias, representing measurement of accelerometer or gyroscope when IMU is stationary; $\boldsymbol{\eta} \in R^3$ denotes measurement noise, generally assumed to be zero-mean Gaussian white noise.

A conventional IMU motion model typically performs predictions under the assumption that inertial sensors have been successfully calibrated. By disregarding the aforementioned error sources, the following motion model can be established,

$$\begin{cases} \boldsymbol{R}_b^n(k+1) = \boldsymbol{R}_b^n(k)\boldsymbol{R}_{b_{k+1}}^{b_k} \\ \boldsymbol{v}_n(k+1) = \boldsymbol{v}_n(k) + \boldsymbol{a}_n(k)dt \\ \boldsymbol{p}_n(k+1) = \boldsymbol{p}_n(k) + \boldsymbol{v}_n(k)dt + \frac{1}{2}\boldsymbol{a}_n(k)dt^2 \end{cases} \tag{2}$$

where subscripts n and b denote navigation frame and body frame, respectively, while $\boldsymbol{R}$, $\boldsymbol{v}$, and $\boldsymbol{p}$ represent rotation matrix, linear velocity, and position coordinates. $\boldsymbol{R}_b^n$ represents rotation matrix from body frame to navigation frame. Here dt and k denote sampling time and the k-th sampling interval. The relationship between $\boldsymbol{R}_{b_{k+1}}^{b_k}$, $\boldsymbol{a}_n$ and IMU measurements ($\tilde{\boldsymbol{\omega}}$ and $\tilde{\boldsymbol{a}}_b$) is given in Eq. (3),

$$\begin{cases} \boldsymbol{R}_{b_{k+1}}^{b_k} = \boldsymbol{I} + \sin(\boldsymbol{\sigma})\frac{[\boldsymbol{\sigma}\times]}{\boldsymbol{\sigma}} + (1-\cos(\boldsymbol{\sigma}))\frac{[\boldsymbol{\sigma}\times]^2}{\boldsymbol{\sigma}^2} \\ \boldsymbol{\sigma} = \tilde{\boldsymbol{\omega}}(k)dt \\ \boldsymbol{a}_n(k) = \boldsymbol{R}_b^n(k-1)\tilde{\boldsymbol{a}}_b(k) - \boldsymbol{g}_n \end{cases} \tag{3}$$

In the absence of corrective measures, the error rapidly increases as it accumulates through integration.

B. Unscented Kalman Filter

UKF employs unscented transformation to estimate the mean and variance of a distribution after passing through a nonlinear transformation. Unlike EKF, UKF does not require calculating the Jacobian matrix and is capable of achieving second-order accuracy. Therefore, it has been widely applied to nonlinear system filtering. Suppose a nonlinear system is given in Eq. (4), where both $\boldsymbol{\eta_k}$ and $\boldsymbol{\mu_k}$ are zero-mean white noises, with $\boldsymbol{Q}$ and $\boldsymbol{R}$ being their variance matrix, respectively. $\boldsymbol{x}_k$, $\boldsymbol{z}_k$ and u_k are state vector, observation and input, while $f(\cdot)$ and $h(\cdot)$ represent the process and measurement model.

$$\begin{cases} \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{\eta}_k) & \boldsymbol{\eta}_k \sim N(0, \boldsymbol{Q}) \\ \boldsymbol{z}_k = h(\boldsymbol{x}_k, \boldsymbol{\mu}_k) & \boldsymbol{\mu}_k \sim N(0, \boldsymbol{R}) \end{cases} \tag{4}$$

Then the procedure of UKF can be summarized as follows:

1) Initialization: Considering that noises are not additive, we construct an augmented state variable and covariance matrix,

$$\boldsymbol{x}_k^{(a)} = \begin{bmatrix} \boldsymbol{x_k} \\ \boldsymbol{\eta_k} \\ \boldsymbol{\mu_k} \end{bmatrix}, \boldsymbol{P}_k^{(a)} = diag(\boldsymbol{P}_k, \boldsymbol{Q}, \boldsymbol{R}) \tag{5}$$

Set initial values for state variable and covariance matrix as

$$\boldsymbol{x}_{0|0}^{(a)} = \begin{bmatrix} E(\boldsymbol{x}_0) \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{P}_{0|0}^{(a)} = diag\left(\boldsymbol{P}_0, \boldsymbol{Q}, \boldsymbol{R}\right) \tag{6}$$

2) Model prediction: Assuming the posterior estimates and covariance at time k are given by $\boldsymbol{x}_{k|k}^{(a)}$ and $\boldsymbol{P}_{k|k}^{(a)}$, the prior state estimate and covariance at time k+1 can be obtained based on unscented transformation. First, sigma points $\boldsymbol{x}_k^{(i)}, i = 0, 1, \cdots, 2n+1$, are generated from the augmented state estimate $\boldsymbol{x}_{k|k}^{(a)}$ and covariance matrix $\boldsymbol{P}_{k|k}^{(a)}$ through the sampling strategy defined in (7), where $(\cdot)_{i-n}$ denotes the $(i-n)$-th column of the matrix inside the parentheses, and n represents the state dimension. These $2n+1$ sigma points are subsequently propagated through the process model to yield transformed vectors $\boldsymbol{x}_{k+1|k}^{(i)} = f(\boldsymbol{x}_k^{(i)}, \boldsymbol{u}_k)$. The prior estimated mean $\boldsymbol{x}_{k+1|k}$ and covariance matrix $\boldsymbol{P}_{k+1|k}$ are then computed using the weighting scheme specified in (8) and the aggregation rules outlined in (9). λ, α and β are the parameters that need to be adjusted.

$$\boldsymbol{x}_k^{(i)} = \begin{cases} \boldsymbol{x}_{k|k}^{(a)} & i = 0 \\ \boldsymbol{x}_{k|k}^{(a)} + (\sqrt{(n+\lambda)\boldsymbol{P}_{k|k}^{(a)}})_i & i = 1 \cdots n \\ \boldsymbol{x}_{k|k}^{(a)} - (\sqrt{(n+\lambda)\boldsymbol{P}_{k|k}^{(a)}})_{i-n} & i = n+1 \cdots 2n \end{cases} \tag{7}$$

$$\begin{cases} \psi_m^0 = \frac{\lambda}{n+\lambda}, \psi_c^0 = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta \\ \psi_m^i = \psi_c^i = \frac{1}{2(n+\lambda)}, i \neq 0 \end{cases} \tag{8}$$

$$\begin{cases} \boldsymbol{x}_{k+1|k} = \sum_{i=0}^{2n} \psi_m^i \boldsymbol{x}_{k+1|k}^{(i)} \\ \boldsymbol{P}_{k+1|k} = \sum_{i=0}^{2n} \psi_c^i \begin{bmatrix} (\boldsymbol{x}_{k+1|k}^{(i)} - \boldsymbol{x}_{k+1|k}) \times \\ (\boldsymbol{x}_{k+1|k}^{(i)} - \boldsymbol{x}_{k+1|k})^T \end{bmatrix} \end{cases} \tag{9}$$

3) Measurement update: Analogous to the prediction step, the update phase initiates by generating sigma points from the prior state estimate $\boldsymbol{x}_{k+1|k}$ and covariance $\boldsymbol{P}_{k+1|k}$. These points are propagated through the measurement model $h(\cdot)$ to yield transformed observation vectors $\boldsymbol{z}_{k+1}^{(i)} = h(\boldsymbol{x}_{k+1|k}^{(i)})$ for $i = 0, \ldots, 2n$. The measurement mean $\hat{\boldsymbol{z}}_{k+1}$ and innovation covariance $\boldsymbol{P}_{zz,k+1}$ are then computed via the weighted summation in Eq. (9). Finally, the posterior state estimate $\boldsymbol{x}_{k+1|k+1}$ and its covariance $\boldsymbol{P}_{k+1|k+1}$ are obtained through Eq. (10), where the cross-covariance $\boldsymbol{P}_{xz,k+1}$ follows Eq. (11).

$$\begin{cases} \boldsymbol{K}_{k+1} = \boldsymbol{P}_{xz}\boldsymbol{P}_z^{-1} \\ \boldsymbol{x}_{k+1|k+1} = \boldsymbol{x}_{k+1|k} + \boldsymbol{K}_{k+1}(\boldsymbol{z}_{k+1} - \hat{\boldsymbol{z}}_{k+1}) \\ \boldsymbol{P}_{k+1|k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{K}_{k+1}\boldsymbol{P}_z\boldsymbol{K}_{k+1}^T \end{cases} \tag{10}$$

$$\boldsymbol{P}_{xz} = \sum_{i=0}^{2n} \begin{bmatrix} \psi_c^i(\boldsymbol{x}_{k+1|k}^{(i)} - \boldsymbol{x}_{k+1|k}) \\ \cdot(\hat{\boldsymbol{z}}_{k+1}^{(i)} - \hat{\boldsymbol{z}}_{k+1})^T \end{bmatrix} \tag{11}$$

III. METHODOLOGY

A. System Description

Consider a 2D scenario as depicted in Fig. 2, where $X_nO_nY_n$ represents the navigation coordinate frame and $X_bO_bY_b$ denotes the body-fixed coordinate frame. We simplify (2) by defining the system state at the kth instant as $\boldsymbol{x}_k = \begin{bmatrix} L_{x_k} & L_{y_k} & V_{x_k} & V_{y_k} & \theta_k \end{bmatrix}^T$, where L_{x_k} and L_{y_k} are the robot's Cartesian coordinates in the navigation frame, θ_k denotes the heading angle, and V_{x_k}/V_{y_k} represents the linear velocities in the navigation frame. The system input vector is given by $\tilde{\boldsymbol{u}}_k = \begin{bmatrix} \tilde{a}_{x_k} & \tilde{a}_{y_k} & \tilde{\omega}_k \end{bmatrix}^T$, containing IMU-measured two-axis accelerations ($\tilde{a}_{x_k}$, $\tilde{a}_{y_k}$) and angular rate ($\tilde{\omega}_k$) in the body frame. It is important to note that the input is generally contaminated by a multitude of error sources, including measurement noise. Our objective focuses on developing an enhanced process model with improved error-bound characteristics.

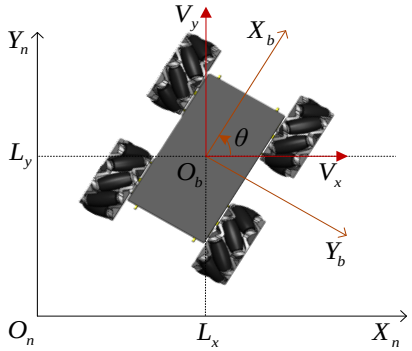

Fig. 2. Description of the system reference frame

In this study, under the assumption that the states are fully observable, the system model can be expressed by (12), where $\boldsymbol{I}_{5\times5}$ is a 5th-order identity matrix.

$$\begin{cases} \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \tilde{\boldsymbol{u}}_k) \\ \boldsymbol{z}_k = \boldsymbol{I}_{5\times5}\boldsymbol{x}_k + \boldsymbol{\mu}_k \end{cases} \tag{12}$$

B. Learning Aided Modeling

1) Modeling: In contrast to the end-to-end approach that directly models process dynamics in the form of $\boldsymbol{x}_{k+1} = f_{net}(\boldsymbol{x}_k, \tilde{\boldsymbol{u}}_k)$, an alternative strategy involves constructing the model based on state increments [21], expressed as $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + f_{net}(\boldsymbol{x}_k, \tilde{\boldsymbol{u}}_k)$. This approach is able to enhance model's interpretability while simultaneously reducing the demands on data volume. Guided by this conceptual framework, we propose the following modeling paradigm,

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\begin{bmatrix} \Delta\hat{V}_{x_k} \\ \Delta\hat{V}_{y_k} \\ \Delta\hat{\theta}_k \end{bmatrix} \\ \boldsymbol{A} = \begin{bmatrix} 1 & 0 & dt & 0 & 0 \\ 0 & 1 & 0 & dt & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0.5dt & 0 & 0 \\ 0 & 0.5dt & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \begin{bmatrix} \Delta\hat{V}_{x_k} & \Delta\hat{V}_{y_k} \end{bmatrix}^T = g_{n1}(\theta_k, \tilde{\boldsymbol{u}}_k), \Delta\hat{\theta}_k = g_{n2}(\tilde{\omega}_k) \end{cases} \tag{13}$$

The functions $g_{n1}(\cdot)$ and $g_{n2}(\cdot)$ represent the trainable neural networks in our architecture, corresponding to MLP1 and MLP2 as depicted in Fig. 1. We establish a decoupled estimation framework through disjoint modeling of translational velocity and angular displacement. This model operates through two systematic stages: First, the translational velocity and incremental orientation changes of the agent are derived from the neural networks' outputs, enabling the estimation of instantaneous velocity, orientation angular and acceleration. Subsequently, positional displacement is calculated based on the kinematic model.

2) Learning Components: In this study, linear velocity and angular variation are modeled independently using two MLPs, MLP1 and MLP2, each with two hidden layers. MLP1 takes four inputs, the previous time step's angular state and IMU measurements, and processes them through hidden layers of 256 and 512 neurons, respectively, with ReLU activations to capture nonlinear relationships. A fully connected layer then outputs the estimated linear velocity variation. In contrast, MLP2 accepts a single input, the previous time step's angular velocity measurement, and employs hidden layers with 64 and 128 neurons, respectively.

3) Loss function: During training, mean squared error (MSE) in (14) served as the loss function, and the model was optimized using the Adam optimizer. To mitigate overfitting, the dataset was partitioned into training, validation, and testing subsets, with the model achieving the best performance on the validation set being selected as the final model.

$$J = \frac{1}{N_d}\sum_{i=1}^{N_d}\left\| \begin{bmatrix} \Delta V_{x_i} & \Delta V_{y_i} & \Delta\theta_i \end{bmatrix}^T - \begin{bmatrix} \Delta\hat{V}_{x_i} & \Delta\hat{V}_{y_i} & \Delta\hat{\theta}_i \end{bmatrix}^T \right\|_2 \tag{14}$$

IV. Experimental Validation

A. Experimental Setup and Training

Data collection platform is a Mecanum-wheeled mobile robot, as shown in Fig. 3(a). This platform is equipped with a ZED2i camera that integrates an IMU for sensor data, while ground-truth is obtained via a motion capture system (four optical cameras and retro-reflection markers) and managed by a Jetson Orin NX. The IMU operates at 200 Hz, and the motion capture system at 100 Hz. Two trajectories, a circular and a figure-eight, each lasting approximately 160 seconds, were designed. The figure-eight trajectory was used for training, with the circular trajectory split equally between validation and testing.

Fig. 3(b) shows that during model training process, the training loss continuously decreases until convergence, whereas the validation loss begins to rise after an initial decline, indicating the occurrence of overfitting. Consequently, the model exhibiting the best performance on the validation set is selected as the final model. The predictive performance of the finalized model for lateral velocity variations in the testing dataset is visually demonstrated in Fig. 3(c).

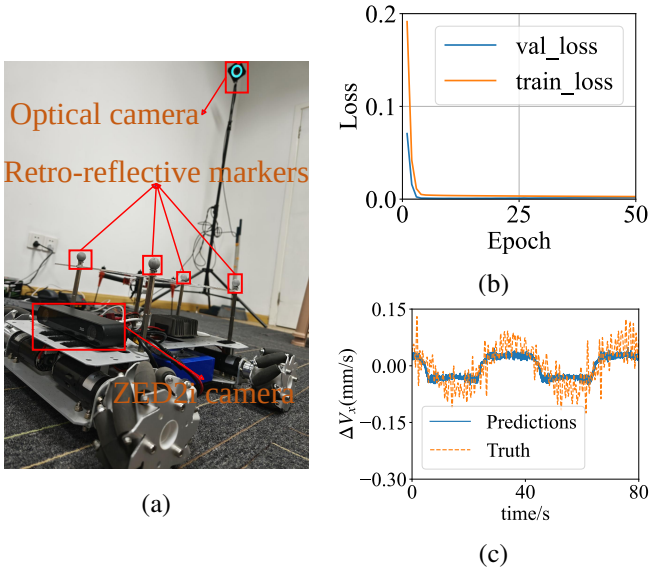

Fig. 3. (a) Experimental platform, (b) Loss variation curves and (c) Prediction for ΔV_x on testing dataset

B. Model Prediction Capability

1) Single-step Prediction: In this section, we first validate the testing capability of the proposed model (denoted as M1). The conventional model shown in Eq. (2) is denoted as M2. Table I summarizes the comparative results, and demonstrates that the proposed M1 significantly outperforms M2 in single-step prediction, particularly in terms of linear velocity and angular prediction accuracy.

2) Multiple-step Prediction: Considering model prediction is important in decision making, we further test the prediction performance of the proposed M1 against the conventional M2. Fig. 4 shows the error distribution when predicting the state

TABLE I
MSE OF SINGLE-STEP PREDICTION FOR M1 AND M2

	$L_x(mm)$	$L_y(mm)$	$\theta(rad)$	$V_x(mm/s)$	$V_y(mm/s)$
M1	**0.0314**	**0.0303**	**3.02e-7**	**0.0014**	**0.0036**
M2	0.0315	0.0305	8.69e-7	3.57	2.5328

at time step $k+20$ using M1 and M2. It can be observed that, compared to M2, the proposed M1 model has a lower mean prediction error (with the distribution center closer to zero) and a smaller variance (indicating a narrower distribution range).

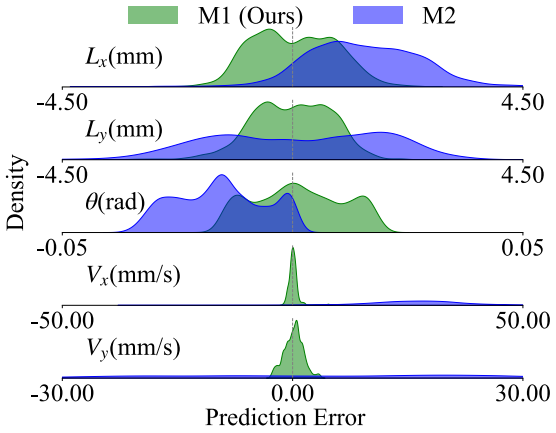

Fig. 4. Distribution of prediction errors over 20 Steps

Fig.5 illustrates the variation in the mean absolute error and variance of predictions as the number of prediction steps increases for both M1 and M2. It can be observed that the growth rate of both error and variance in M1 is significantly lower than that in M2. To better visualize the impact of state prediction errors, we take the 100-step(0.5s) prediction as an example and plot the results (both spatially and temporally) in Fig.6, where each estimated point represents the predicted value obtained in 100-step based on M1 and M2. It can be observed that model M1 demonstrates a significant improvement in prediction capability compared to M2. The improvement in multi-step prediction accuracy of the process equation also indirectly reflects the system's enhanced robustness in handling measurement information loss.

C. Navigation Performance

In this section, we integrate model M1 with the UKF for navigation and state estimation, embedding M1 into the prediction step of the UKF. In our experiments, we assume that the full state is observable (which is a reasonable assumption, as the position, velocity, and attitude of a mobile robot can be obtained through sensors such as GPS and UWB). To simulate UWB measurements, Gaussian white noise is added to the ground truth from Motion Capture data, and the measurement frequency is reduced to 50 Hz. The simulated navigation process is as follows: when UWB data is unavailable, only IMU data is used for prediction and when UWB data is

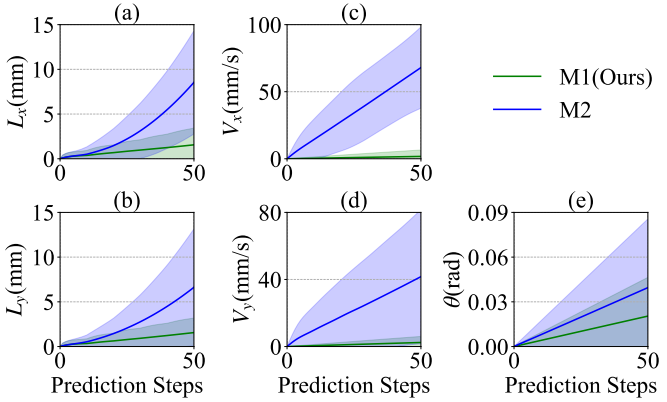

Fig. 5. Curve of MAE variation with prediction steps, with shaded area representing the 95% confidence interval calculated from variance

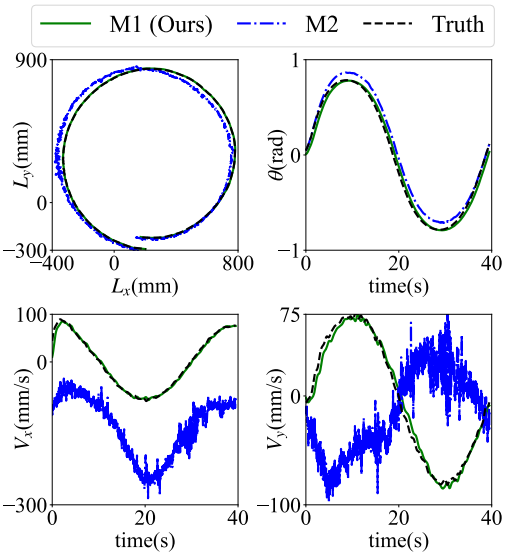

Fig. 6. Trajectory prediction performance: each state estimation point represents the estimated value obtained from 100-step model prediction

available, it is used to update the predicted values. The final navigation performance is shown in Fig. 7, with quantitative results summarized in Table II.

The lack of significant improvement in position and orientation estimation accuracy may be related to the parameter settings of the process noise covariance matrix in the UKF. However, it can still be observed that our proposed model significantly improves the estimation of linear velocity while maintaining comparable accuracy in position and orientation estimation. This improvement is crucial for tasks such as obstacle avoidance and path planning where trajectory prediction is required.

V. CONCLUSIONS

This study considers the problem of mobile robot localization and prediction by model based approach. Considering the limited performance of classical kinematic modeling, Neural network aided modeling is adopted to improve the modeling

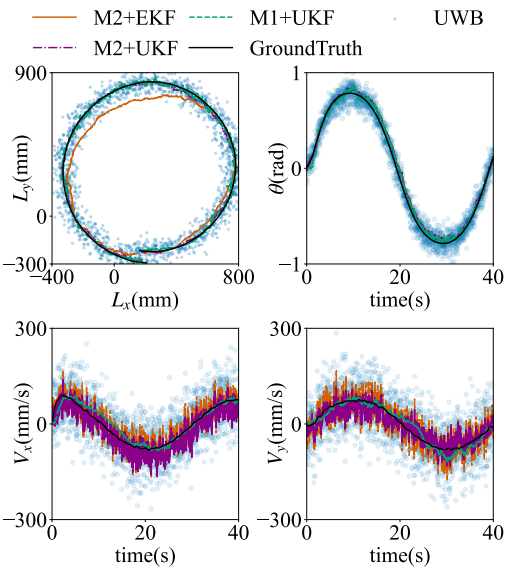

Fig. 7. Comparison of IMU/UWB integrated navigation performance under different process models and filtering algorithms

TABLE II
MAE OF STATE ESTIMATION FOR DIFFERENT METHODS

	L_x	L_y	θ	V_x	V_y
M2+EKF	34.95	36.40	**0.03**	24.21	25.19
M2+UKF	**7.35**	9.72	**0.03**	23.09	19.37
M1+UKF	7.89	**8.41**	**0.03**	**11.48**	**11.61**

performance, where two Multilayer Perceptron (MLP) are learned to regresses the changes in linear velocity and orientation angles. The proposed learning-aided modeling method is validated against the classical model by a Mecanum-wheeled mobile platform, where low-frequency UWB measurement (Motion Capture data added with noises) and IMU are integrated in UKF framework. Comparative experimental results demonstrate that, 1) MLP can learn the variation for better modeling; 2) the proposed learning-aided modeling provides enhanced single-step and multiple-step model based prediction; 3) the proposed modeling integrated with UKF provides better state filtering, particularly in linear velocity estimation, which is crucial for navigation tasks such as obstacle avoidance and motion planning. Future work will further improve the modeling performance and consider more practical experimental validation including outdoor validations.

REFERENCES

[1] Pierluigi Rea and Erika Ottaviano. Design and development of an inspection robotic system for indoor applications. *Robotics and Computer-Integrated Manufacturing*, 49:143–151, 2018.

[2] Jinya Su, Xiaoyong Zhu, Shihua Li, and Wen-Hua Chen. Ai meets uavs: A survey on ai empowered uav perception systems for precision agriculture. *Neurocomputing*, 518:242–270, 2023.

[3] Christopher Doer and Gert F. Trommer. Radar visual inertial odometry and radar thermal inertial odometry: Robust navigation even in challenging visual conditions. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 331–338, 2021.

[4] Daquan Feng, Junjie Peng, Yuan Zhuang, Chongtao Guo, Tingting Zhang, Yinghao Chu, Xiaoan Zhou, and Xiang-Gen Xia. An adaptive imu/uwb fusion method for nlos indoor positioning and navigation. *IEEE Internet of Things Journal*, 10(13):11414–11428, 2023.

[5] Thien Hoang Nguyen, Thien-Minh Nguyen, and Lihua Xie. Range-focused fusion of camera-imu-uwb for accurate and drift-reduced localization. *IEEE Robotics and Automation Letters*, 6(2):1678–1685, 2021.

[6] Yunqi Gao, Dianxi Shi, Ruihao Li, Zhe Liu, and Wen Sun. Gyro-net: Imu gyroscopes random errors compensation method based on deep learning. *IEEE Robotics and Automation Letters*, 8(3):1471–1478, 2022.

[7] Lujuan Dang, Badong Chen, Yulong Huang, Yonggang Zhang, and Haiquan Zhao. Cubature kalman filter under minimum error entropy with fiducial points for ins/gps integration. *IEEE/CAA Journal of Automatica Sinica*, 9(3):450–465, 2022.

[8] Jinya Su, Baibing Li, and Wen-Hua Chen. On existence, optimality and asymptotic stability of the kalman filter with partially observed inputs. *Automatica*, 53:149–154, 2015.

[9] Tiancheng Li, Jinya Su, Wei Liu, and Juan M Corchado. Approximate gaussian conjugacy: parametric recursive filtering under nonlinearity, multimodality, uncertainty, and constraint, and beyond. *Frontiers of Information Technology & Electronic Engineering*, 18(12):1913–1939, 2017.

[10] Masoud Khodarahmi and Vafa Maihami. A review on kalman filter models. *Archives of Computational Methods in Engineering*, 30(1):727–747, 2023.

[11] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[12] Changhao Chen and Xianfei Pan. Deep learning for inertial positioning: A survey. *IEEE transactions on intelligent transportation systems*, 2024.

[13] Robert Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys & Tutorials*, 15(3):1281–1293, 2013.

[14] Isaac Skog, Peter Handel, John-Olof Nilsson, and Jouni Rantakokko. Zero-velocity detection—an algorithm evaluation. *IEEE transactions on biomedical engineering*, 57(11):2657–2666, 2010.

[15] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[16] Sachini Herath, Hang Yan, and Yasutaka Furukawa. Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 3146–3152. IEEE, 2020.

[17] Huakun Liu, Xin Wei, Monica Perusquía-Hernández, Naoya Isoyama, Hideaki Uchiyama, and Kiyoshi Kiyokawa. Duet: Improving inertial-based odometry via deep imu online calibration. *IEEE Transactions on Instrumentation and Measurement*, 72:1–13, 2023.

[18] Mahdi Abolfazli Esfahani, Han Wang, Keyu Wu, and Shenghai Yuan. Orinet: Robust 3-d orientation estimation with a single particular imu. *IEEE Robotics and Automation Letters*, 5(2):399–406, 2019.

[19] Wenxin Liu, David Caruso, Eddy Ilg, Jing Dong, Anastasios I Mourikis, Kostas Daniilidis, Vijay Kumar, and Jakob Engel. Tlio: Tight learned inertial odometry. *IEEE Robotics and Automation Letters*, 5(4):5653–5660, 2020.

[20] Yayun Du, Swapnil Sayan Saha, Sandeep Singh Sandha, Arthur Lovekin, Jason Wu, S Siddharth, Mahesh Chowdhary, Mohammad Khalid Jawed, and Mani Srivastava. Neural-kalman gnss/ins navigation for precision agriculture. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9622–9629. IEEE, 2023.

[21] Pratyaksh Prabhav Rao, Alessandro Saviolo, Tommaso Castiglione Ferrari, and Giuseppe Loianno. Learning long-horizon predictions for quadrotor dynamics. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12758–12765, 2024.