2024 XXVI Robotics Mexican Congress (COMRob)
November 13-15, 2024, Torreón, Coahuila, México

Pose Estimation of a Differential-Drive Mobile Robot Using EKF-Based Sensor Fusion

Fernando-Rafael Martínez-Algarín
Tecnológico Nacional de México, I. T. Tuxtla Gutiérrez
Turix-Dynamics Diagnosis and Control Group
C.ª Panamericana S/N, 29050 Tuxtla Gutiérrez, México
Email: M18270404@tuxtla.tecnm.mx

Ildeberto Santos-Ruiz
Tecnológico Nacional de México, I. T. Tuxtla Gutiérrez
Turix-Dynamics Diagnosis and Control Group
C.ª Panamericana S/N, 29050 Tuxtla Gutiérrez, México
Email: ildeberto.dr@tuxtla.tecnm.mx

Francisco-Ronay López-Estrada
Tecnológico Nacional de México, I. T. Tuxtla Gutiérrez
Turix-Dynamics Diagnosis and Control Group
C.ª Panamericana S/N, 29050 Tuxtla Gutiérrez, México
Email: frlopez@tuxtla.tecnm.mx

Joaquín-Eduardo Domínguez-Zenteno
Tecnológico Nacional de México, I. T. Tuxtla Gutiérrez
Turix-Dynamics Diagnosis and Control Group
C.ª Panamericana S/N, 29050 Tuxtla Gutiérrez, México
Email: joaquin.dz@tuxtla.tecnm.mx

2024 XXVI Robotics Mexican Congress (COMRob) | 979-8-3315-3339-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/COMRob64055.2024.10777435

***Abstract*—Localization is an important process in navigation systems. In this stage, the robot gathers the necessary data to navigate safely and reach its destination. This paper presents a robust pose estimation method for a differential-drive robot using sensor fusion techniques, along with the design, instrumentation, and implementation in a warehouse robot used for transporting loads, focusing on real-world applications. The objective is to accurately determine the robot's position and orientation by integrating data from wheel encoders and an Inertial Measurement Unit (IMU) mounted on the robot. The wheel encoders allow the estimation of the linear and angular velocities of the robot, while the IMU offers complementary data on the robot's acceleration and angular rates. The data is fused through the Extended Kalman Filter (EKF), considering the kinematics of the differential-drive robot and the uncertainties of the sensor measurements. The results from simulations and real-world experiments demonstrate an improvement in the pose estimation accuracy.**

***Index Terms*—Pose estimation, Differential drive, Mobile robot, Extended Kalman filter, Sensor fusion.**

I. Introduction

Autonomous robots are reliable systems utilized for performing tasks such as human and goods transportation. They are used in both outdoor and indoor environments because they provide many features such as speed and safety, which makes them perfect to execute tasks where accuracy and time are relevant. To develop proper autonomous robots, it is required to equip them with sensors to gather information from the environment and from itself. The information allows the robot to estimate its pose, detect obstacles, and make clever decisions to achieve goals.

It is important to gather information from various types of sensors, as each sensor has its own advantages and disadvantages. For instance, some are more resistant to noise, while others are significantly affected by environmental factors. When using many sensors, it is required to fuse the gathered information. For that purpose, sensor fusion techniques allow the integration of information that comes from different sources to produce a more reliable and robust information than that provided by any sensor individually [1].

The selection of sensors is directly related to the type of environment and the tasks that the robot performs. The most used sensors in outdoor environments are wheel encoders and GPS, but when a robot operates in indoor environments GPS cannot be used, and other problems arise in performing the localization task [2]. To adress this, the robot is equipped with sensors such as encoders, IMU, and LiDAR [3]. Each of them offers different types of information, such as linear displacement deduced from wheel rotation, angular velocity, rotations, magnetic field interference, and the relative distance from the robot to objects or walls. Such information can be used to infer the robot's states with the proper sensor fusion method to determine where the robot is.

Many sensor fusion methods exist, but the most commonly used for state estimation in navigation systems are the Extended Kalman Filter (EKF), the Unscented Kalman Filter (UKF), and the Particle Filter (PF) [4]. The EKF allows for the effective fusion of information from heterogeneous sources effectively [5] and offers advantages such as low computational cost and denoising for nonlinear systems [6].

This paper focuses on the implementation of an EKF for pose estimation in a differential-drive robot. The prototype is designed to deliver packages with a maximum allowable load of 120 kg. The EKF was implemented offline as a first approach to develop a localization algorithm fusing encoder and IMU data. Localization is the initial process that an autonomous robot performs to carry out tasks. This work will be useful in future projects that develop autonomous warehouse mobile robots.

The paper is organized as follows: Section II describes the kinematic model of the differential-drive vehicle. Section III presents equations related to the robot's odometry using wheel encoders. Section IV presents the sensor fusion approach used in this paper. Section V describes the prototype developed.

Section VI shows the results obtained. Finally, Section VII presents the conclusions of the paper.

II. Differential drive kinematics

The differential mobile robot is a non-holonomic system that operates under certain movement restrictions caused by mechanical and geometrical factors.

It has two motorized wheels that can be controlled independently, allowing the robot to change its heading. Additionally, it has castor wheels that provide support and can be neglected in the kinematic analysis. Two caster wheels were used for the robot presented in this paper.

Fig. 1 shows the diagram of the robot and the variables involved in the kinematic analysis.

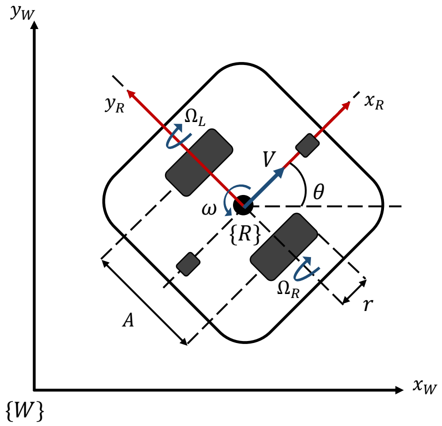

Fig. 1. Differential-drive mobile robot diagram.

For modeling the robot kinematics, the following conditions were assumed [7]:

1) The center of gravity of the robot is located at the origin of the coordinate system {R}.
2) The wheels make contact with the ground in a single point.
3) The wheels have a constant and equal radius.
4) The robot navigates on a flat surface.

Three variables define the robot's pose ξ: two position coordinates (x, y) and an orientation angle θ related to a terrestrial reference frame (see Fig. 1).

$$\xi = \begin{bmatrix} x & y & \theta \end{bmatrix}^\top \tag{1}$$

The control inputs of the robot are the linear speed v and the angular speed ω, so that:

$$\dot{\xi} = f(\xi, v, \omega) \tag{2}$$

By assuming that the control inputs can be changed arbitrarily, the motion of the robot is described by the following kinematic model:

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} \tag{3}$$

III. Odometry

Odometry is a technique used in robotics to track the evolution of the robot's pose for navigation purposes. It consists in measuring the rotation of the wheels and using the information to estimate linear displacement and heading [8].

The sensors used in this paper for that purpose are incremental encoders, which send signal pulses related to the angular displacement of the wheels. The angular displacement of the right wheel, $\Delta\varphi_r$, and the left wheel, $\Delta\varphi_l$, are calculated using the following equations:

$$\Delta\varphi_r = \frac{2\pi}{N} n_r, \tag{4}$$

$$\Delta\varphi_l = \frac{2\pi}{N} n_l, \tag{5}$$

where n_r and n_l are the numbers of ticks registered by the right and the left encoders, and N is the total amount of ticks for a full wheel revolution.

Knowing the relative angular displacement, we can compute the angular speed of each wheel Ω_r and Ω_l dividing it by the sampling time T_s as follows:

$$\Omega_r = \Delta\varphi_r / T_s, \tag{6}$$

$$\Omega_l = \Delta\varphi_l / T_s. \tag{7}$$

The angular velocity of each wheel is related to the linear velocity and the angular velocity of the robot, as shown in the following equations:

$$v = \frac{r}{2}\left(\Omega_r + \Omega_l\right), \tag{8}$$

$$\omega = \frac{r}{A}\left(\Omega_r - \Omega_l\right), \tag{9}$$

where r is the wheel radius and A is the distance from the center of one wheel to another, as shown in Fig. 1.

IV. Sensor fusion

A robot requires reliable information about its states to navigate safely. In many situations, a single sensor does not provide reliable enough information. To address this problem, robots are equipped with many sensors that provide information about their states, and with a sensor fusion algorithm, the gathered information is used to generate more reliable state estimations.

The sensor fusion method used in this paper is the EKF, which is a recursive algorithm based on probability that computes the best states estimations of a non-linear dynamical system given discrete time measurements that relate to the states of the robot [9], allowing the use of information provided by different sensors. It is important to use more than just wheel encoders, as displacement estimations from wheel encoders are characterized by uncertainties caused by wheel slip or changes in the wheel radius [10].

For this work focused on a warehouse robot that operates in indoor environments, the robot was equipped with wheel encoders and an IMU. The data provided by encoders were used for the prediction step and IMU data were used for the update step, as shown in Fig. 2.

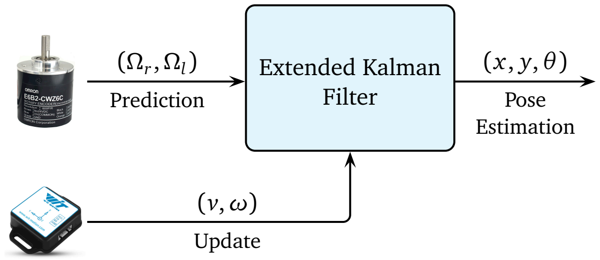

Fig. 2. EKF-based sensor fusion approach.

To implement the extended Kalman filter, it is necessary to discretize the model shown in (3). Using the forward Euler method and considering the control inputs $\begin{bmatrix} v & \omega \end{bmatrix}^\top$ as additional state variables [11], the states of the robot at the instant $k+1$ with the encoder measurements as inputs, $u = [\Omega_r \;\; \Omega_l]^\top$, are defined as:

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \\ v_{k+1} \\ \omega_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + T_s v_k \cos\left(\theta_k + T_s \omega_k\right) \\ y_k + T_s v_k \sin\left(\theta_k + T_s \omega_k\right) \\ \theta_k + T_s \omega_k \\ \frac{r}{2}\left[(\Omega_r)_k + (\Omega_l)_k\right] \\ \frac{r}{A}\left[(\Omega_r)_k - (\Omega_l)_k\right] \end{bmatrix} \tag{10}$$

The new state depends on the previous k-th state and inputs $(\Omega_r)_k$ and $(\Omega_l)_k$. With this approach, we can predict the next robot states using the information provided by encoders and then use IMU data to correct the estimations.

For performing EKF, it is important to consider both process and measurement noise, denoted as p_k and m_k. These noises are assumed Gaussian with zero mean and covariance matrices (Q_k and R_k), as described by:

$$p_k \sim \mathcal{N}(0, Q_k) \tag{11}$$

$$m_k \sim \mathcal{N}(0, R_k) \tag{12}$$

Then, the stochastic model of the system considering the noise is:

$$X_{k+1} = f(X_k, u_k) + p_k, \tag{13}$$

where f is the non-linear function on the right side of (10) that establishes the relation of the states and inputs at the previous time step.

An advantage of the EKF is that it estimates states using non-linear system models. To achieve this, the model is linearized around the current operating point, computing the Jacobian matrix of the system model. The model linearization at time step k is shown as follows:

$$F_k = \frac{\partial f(X_k, u_k)}{\partial X_k} = \begin{bmatrix} 1 & 0 & F_{1,3} & F_{1,4} & F_{1,5} \\ 0 & 1 & F_{2,3} & F_{2,4} & F_{2,5} \\ 0 & 0 & 1 & 0 & T_s \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{14}$$

where

$$F_{1,3} = -T_s\, v_k \sin(\theta_k + T_s \omega_k), \tag{15}$$

$$F_{1,4} = T_s \cos(\theta_k + T_s \omega_k), \tag{16}$$

$$F_{1,5} = -T_s^2\, v_k \sin(\theta_k + T_s \omega_k), \tag{17}$$

$$F_{2,3} = T_s\, v_k \cos(\theta_k + T_s \omega_k), \tag{18}$$

$$F_{2,4} = T_s \sin(\theta_k + T_s \omega_k), \tag{19}$$

$$F_{2,5} = T_s^2\, v_k \cos(\theta_k + T_s \omega_k). \tag{20}$$

A measurement model is required to fuse different sensor information. It describes how the information gathered by each sensor relates to the system states, as described in the next equation:

$$z_k = h(X_k) + m_k, \tag{21}$$

where h is a non-linear function that relates the states X_k to the measurements.

An IMU is a sensor that provides information such as accelerations, angular speed, and magnetic field data. The information considered for the update process is the angular speed ω_{IMU} (relative to z-axis) and the acceleration a_{IMU} in the direction of the vehicle's movement (x-axis of the car). Integrating a_{IMU} over time gives the linear velocity v_k:

$$v_k = \int_0^{kT_s} a_{\text{IMU}}(t)\, \mathrm{d}t, \tag{22}$$

$$\omega_k = \omega_{\text{IMU}}(kT_s). \tag{23}$$

Then, the measurement model considering the IMU sensors is defined by:

$$h = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} \tag{24}$$

It also requires a linearization of the measurement model h; the linearization is given as follows:

$$H_k = \frac{\partial h}{\partial X_k} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{25}$$

From the transition and measurement models given by (10) and (24), with the Jacobian matrices (14) and (25), the EKF can be performed. The pose estimation process is summarized in Algorithm 1.

At each time step of the algorithm, an *a priori* estimate, denoted by the superscript x_k^-, is obtained by evaluating the nonlinear state model; this step is called "Prediction". Then, from the IMU measurements, an *a posteriori* estimate is obtained to correct the prior estimate; this step is called "Update". The matrix P_k is the uncertainty in the estimate $\widehat{x}_k$, which tends to decrease over time.

V. Robot Prototype

A. Design

The robot was designed using the CAD software SolidWorks, considering the motors, wheels, and battery dimensions. The robot has a width and length of 0.72 m, and a height of 0.22 m. PTR steel profiles were selected to provide robustness and resistance to the prototype, allowing it to carry

Algorithm 1 EKF-based pose estimation

```
Input: $u_k = [\Omega_r, \Omega_l]$, $z_k = [v_{\text{IMU}}, \omega_{\text{IMU}}]$
Output: $\widehat{x}_k$
$k \leftarrow 0$
Initialize $\widehat{x}_0^-$ and $P_0^-$
while {robot is operating} do
    $u_k \leftarrow$ Encoder measurements: $[\Omega_r, \Omega_l]^\top$
    $z_k \leftarrow$ IMU data: $[v_{\text{IMU}}, \omega_{\text{IMU}}]^\top$
        $v_{\text{IMU}}$ is obtained from $a_{\text{IMU}}$ according to (22).
  Update:
    $H_k \leftarrow \left(\partial h/\partial x\right)\big|_{\widehat{x}_k^-}$
    $K_k \leftarrow P_k^- H_k^\top \left(H_k P_k^- H_k^\top + R_k\right)^{-1}$
    $\widehat{x}_k \leftarrow \widehat{x}_k^- + K_k \left(z_k - h(\widehat{x}_k^-)\right)$
    $P_k \leftarrow (I - K_k H_k)\, P_k^-$
  Prediction:
    $\widehat{x}_{k+1}^- \leftarrow f(\widehat{x}_k, u_k)$
    $F_k \leftarrow \left(\partial f/\partial x\right)\big|_{\widehat{x}_k, u_k}$
    $P_{k+1}^- \leftarrow F_k P_k F_k^\top + Q_k$
    $k \leftarrow k + 1$
end while
```

a maximum load of 120 kg. The physical parameters of the robot used in the kinematic model are $A = 0.595\,\text{m}$ (the distance between the wheels) and $r = 0.1016\,\text{m}$ (the wheel radius). A 3D view of the design is shown in Fig. 3.

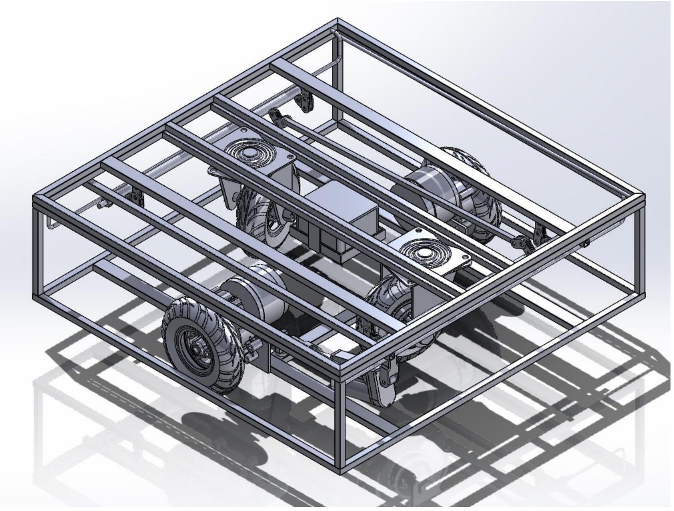

Fig. 3. CAD model of the warehouse robot.

B. Sensors and electronics instrumentation

The robot was equipped with incremental wheel encoders and an IMU as sensors. The incremental wheel encoders selected are the B-E6B2-CWZ6C model (from BERM), which are directly coupled to the exposed shafts of the DC Yalu-MY1016ZL motors. The IMU selected is the BWT901CL, manufactured by WitMotion. It was placed at the center of the robot, with the x-axis facing forward. A Raspberry Pi 4 development board was used to control the robot, and a laptop was used to acquire data via serial from an IMU and an ESP 32 that processed the ticks counted by the encoders. Other elements were used to create the system. Fig. 4 shows the electronic components and their interactions.

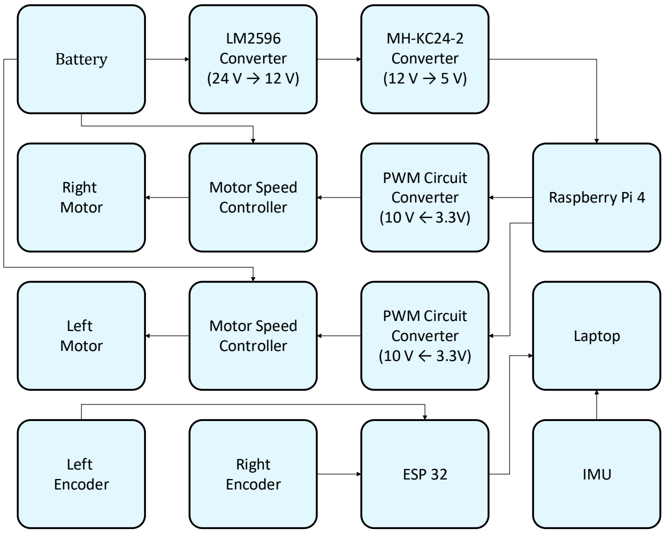
Battery
LM2596 Converter (24 V → 12 V)
MH-KC24-2 Converter (12 V → 5 V)
Right Motor
Motor Speed Controller
PWM Circuit Converter (10 V ← 3.3V)
Raspberry Pi 4
Left Motor
Motor Speed Controller
PWM Circuit Converter (10 V ← 3.3V)
Laptop
Left Encoder
Right Encoder
ESP 32
IMU

Fig. 4. Robot instrumentation.

The differential-drive robot used for the experiments is shown in Fig. 5.

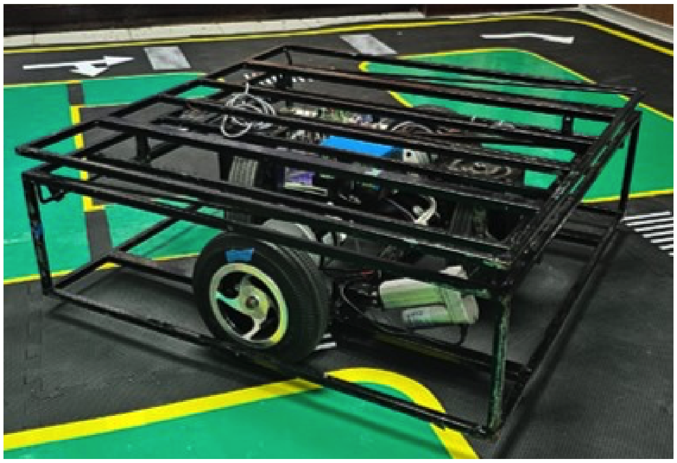

Fig. 5. Assembled warehouse robot.

VI. Results

The EKF performance was evaluated through several experiments in an indoor environment. The data from wheel encoders and IMU were collected by the Raspberry Pi 4 at a sampling rate of 100 Hz and processed offline using MATLAB on a desktop computer. Fig. 6 shows the plot of a typical one-minute data set, which was used to validate the pose estimation method.

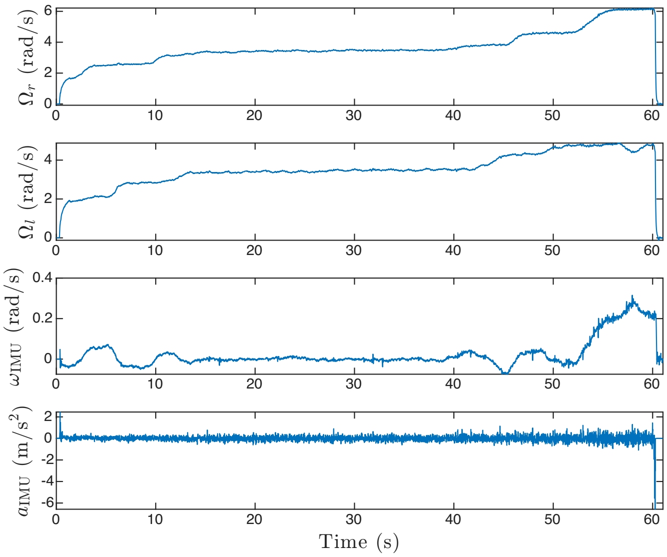

Fig. 6. Sensor signals in a one-minute run.

The trajectory of the robot was estimated using two approaches. The first approach uses only the information provided by wheel encoders, and the second approach consists of sensor fusion of wheel encoders and IMU.

In the implementation of the EKF, the process noise covariance matrix Q and the measurement noise covariance matrix R play crucial roles in the filter's performance. For our EKF, these matrices were determined empirically through a series of experiments and observations. The matrix Q represents the uncertainty in the system model and was adjusted to obtain the best performance of the estimator. It is defined as $Q = \text{diag}([1, 1, 1, 1, 1]) \times 10^{-1}$. On the other hand, the matrix R, which captures the uncertainty in the sensor measurements, was determined based on the stationary state measurement noise of the IMU. It is defined as $R = \text{diag}([1, 1]) \times 1.8 \times 10^{-2}$. These values were chosen based on experimental data to ensure the filter accurately represents the system dynamics and measurement noise.

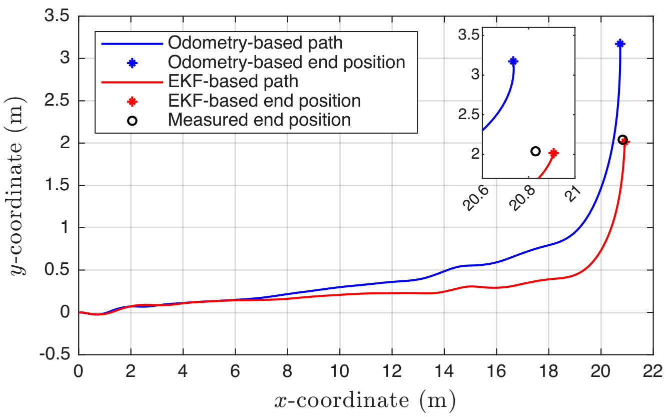

Fig. 7. Test path to evaluate EKF-based pose estimation.

The estimated trajectories of the robot are shown in Fig. 7. The zoomed-in region highlights the last estimated positions of the robot. The blue dot represents the end estimated position using wheel odometry, the red dot is the position calculated by EKF, and the black circle is the ground truth, which was measured. The known end position is used for computing the norm of position error, E, using the Euclidean distance formula as follows:

$$E = \sqrt{(x_{\text{true}} - x_{\text{est}})^2 + (y_{\text{true}} - y_{\text{est}})^2}, \qquad (26)$$

where x_{est} and y_{est} are the estimated coordinates, whereas x_{true} and y_{true} are the measured coordinates. The orientation error $\widetilde{\theta}$ is computed as the difference between the true (measured) angle and the estimated angle:

$$\widetilde{\theta} = \theta_{\text{true}} - \theta_{\text{est}}. \qquad (27)$$

Table I illustrates the improvement in the pose estimation between the different approaches.

TABLE I
FINAL POSITION AND ANGLE ERROR.

Estimation method	**Position error, E**	**Orientation error, $\widetilde{\theta}$**
Odometry	1.136 m	−5.286°
EKF	0.082 m	−2.863°

VII. CONCLUSIONS

The proposed pose estimation method for a differential-drive robot shows an improvement in the robustness and accuracy of the estimations based on the experimental results, validating the effectiveness of the EKF in fusing data provided by wheel encoders and IMU. The use of more sensors in future research will make it convenient to use the Robot Operating System (ROS) to develop the communication among the sensors, the Raspberry Pi 4, and the circuits to optimize data sharing and perform the EKF in real-time without using a laptop. These efforts will facilitate the integration of sensors such as LiDAR and camera-based systems to give the robot the ability to perceive its environment and refine pose estimation. The aim of the following work is to contribute to the advancement of reliable and smart autonomous navigation systems.

REFERENCES

[1] M. L. Fung, M. Z. Q. Chen, and Y. H. Chen, "Sensor fusion: A review of methods and applications," in *2017 29th Chinese Control And Decision Conference (CCDC)*, pp. 3853–3860, 2017.

[2] A. H. Kassem and M. Asem, "A low-cost robotic system for simultaneous localization and mapping," *Journal of Engineering and Applied Science*, vol. 71, no. 1, p. 158, 2024.

[3] A. Charroud, K. E. Moutaouakil, and A. Yahyaouy, "Fast and accurate localization and mapping method for self-driving vehicles based on a modified clustering particle filter," *Multimedia Tools and Applications*, vol. 82, pp. 18435–18457, May 2023.

[4] Y. Zhuang, X. Sun, Y. Li, J. Huai, L. Hua, X. Yang, X. Cao, P. Zhang, Y. Cao, L. Qi, J. Yang, N. El-Bendary, N. El-Sheimy, J. Thompson, and R. Chen, "Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches," *Information Fusion*, vol. 95, pp. 62–90, 2023.

[5] M. Tang, B. Zhou, X. Zhong, X. Liu, and Q. Li, “Enhanced indoor positioning through human-robot collaboration,” *Urban Informatics*, vol. 3, no. 1, p. 7, 2024.

[6] X. Fan, N. Reginald, and B. Fidan, “Adaptive path following for a differential drive robot with EKF-based localization,” *IFAC-PapersOnLine*, vol. 55, no. 38, pp. 166–171, 2022. 13th IFAC Symposium on Robot Control SYROCO 2022.

[7] W. Lv, Y. Kang, and J. Qin, “Indoor localization for skid-steering mobile robot by fusing encoder, gyroscope, and magnetometer,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 6, pp. 1241–1253, 2019.

[8] M. A. Mahmud, M. S. Aman, H. Jiang, A. Abdelgawad, and K. Yelamarthi, “Kalman filter based indoor mobile robot navigation,” in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 1949–1953, 2016.

[9] F. E. Daum, *Extended Kalman Filters*, pp. 751–753. Cham: Springer International Publishing, 2021.

[10] U. Onyekpe, V. Palade, A. Herath, S. Kanarachos, and M. E. Fitzpatrick, “Whonet: Wheel odometry neural network for vehicular localisation in gnss-deprived environments,” *Engineering Applications of Artificial Intelligence*, vol. 105, p. 104421, 2021.

[11] B. Fariña, J. Toledo, and L. Acosta, “Sensor fusion algorithm selection for an autonomous wheelchair based on EKF/UKF comparison,” *International Journal of Mechanical Engineering and Robotics Research*, vol. 12, no. 1, pp. 1–7, 2023.