2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Mobile Robot Localization Based on FEM Stress Analysis Using Pressure Sensors Under Floor

Daiyannan Chen and Yonghoon Ji

***Abstract*— Traditional mobile robot localization techniques relying on on-board sensors often face significant limitations, such as visual occlusion, overreliance on visual features, and high computational costs. To address these challenges and monitor the entire environment in real time, this paper proposes an innovative localization framework that uses pressure sensors embedded under the floor to estimate a mobile robot pose based on its weight distribution. In order to reduce computational complexity, a mathematical model based on the Kirchhoff–Love plate theory is established to describe the ground deformation under external loads. This model is then used to simulate and analyze the stress distribution using the finite element method (FEM), forming the basis for the localization. The robot pose is iteratively estimated using a particle filter-based approach, which dynamically adjusts based on observed and predicted pressure distributions to arrive at the optimal result. Using environmental feedback rather than relying on on-board sensors, our approach eliminates the need to equip robots with dedicated localization hardware, reducing cost and system complexity.**

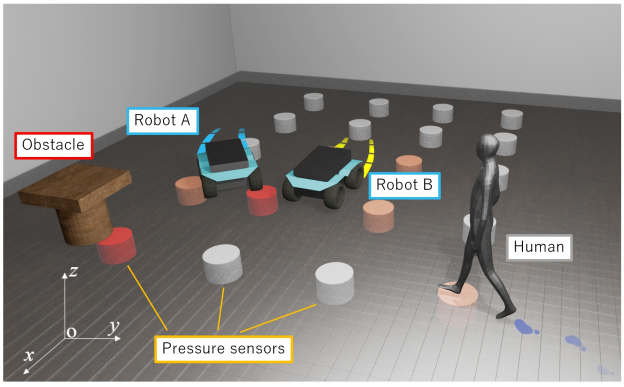


Fig. 1. Localization system using pressure sensors under the floor.

I. Introdution

Accurate localization is one of the most critical components in mobile robotics, as it allows robots to understand their pose within an environment and navigate effectively [1]–[3]. Over the years, researchers have developed a variety of indoor-based localization methods, including vision-based simultaneous localization and mapping (V-SLAM) [4], [5], and light detection and ranging (LiDAR) technology [6], [7], highlighting their applications, challenges, and opportunities for improvement. In terms of V-SLAM technology, cameras capture detailed visual information, enabling robots to identify and differentiate between features such as walls, doors, and objects. The V-SLAM relies on clear and consistent views to detect and track features in image data. If the field of view of the camera is obstructed or the environment lacks distinct visual features (e.g., plain walls or open spaces), the algorithm may not be able to locate the robot [5], [8]. Similar problems also occur with LiDAR. LiDAR provides precise distance measurements, enabling robots to generate highly detailed maps of their surroundings. However, LiDAR may struggle to detect or accurately represent transparent obstacles, such as glasses or acrylic boards [7].

As an alternative approach to localization, we aim to find a novel localization method that relies on environmental sensors instead of on-board sensors. Pressure sensors, as a commonly used medium for robots to sense the external environment, are widely used in various fields such as industrial robotic arms and human-computer interaction [9]–[11]. [9] compared the effect of the pressure sensor measurement on the ground reaction force by investigating the accuracy of the measurement under standard loading conditions. A notable study by [10] demonstrated a method in which sensors installed on the robot's foot captured terrain characteristics, aiding in surface classification. However, these approaches face significant challenges in localization tasks because of their dependence on pressure distribution in the area near the robot and lack recognition of the global environment. On the other hand, commercially available systems such as pressure sensing mats from Tekscan Inc. have shown great promise in capturing detailed force distribution data [12]. However, their design prioritizes high-precision biomechanical and industrial use cases by using a huge array of sensors, making them less efficient and scalable for robotic localization tasks in dynamic or expansive environments. We hope that by arranging fewer pressure sensors, the sensor system can be applied to robot localization in a wide range of indoor environments.

In this paper, from the perspective of pre-establishing an environment for localization rather than for each robot, we suggest a novel concept to estimate a robot pose using pressure sensors installed under the floor as shown in Figure 1. We measure the pressure distribution of the robot on the ground by arranging a sensor matrix under the floor. The basic idea is to estimate the location of each load point, considering the bending problem of the thin plate due to the load of each wheel. Inferring the location of the load using the values of the pressure sensor matrix is a nonlinear problem; thus, the particle filter-based approach is applied

All authors are with Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology, Japan. {s2520026, ji-y}@jaist.ac.jp

as follows. Initially, we predict a possible location based on the robot movement in the previous frame and generate random particles that represent possible positions near the predicted location. By calculating the pressure distribution based on finite element method (FEM) stress analysis at the corresponding position of the predicted particles, we obtain the value of the pressure sensor matrix at the position of the particle. By comparing the calculated value with the observed value, we can iteratively resample the particles and estimate the optimal pose.

The following part of the paper is structured into four sections. In Section II, we introduced the sensor system modeling and defined node constraints for FEM analysis to solve stress at different points on the ground. In Section III, we proposed a pose estimation method for mobile robots based on particle filters. The validation of the task considered is discussed in Section IV through experimental results. We end the paper by providing conclusions and future work that can be undertaken.

II. FEM Analysis

The full governing equations of plate bending involve multiple coupled differential equations, which can be computationally demanding to solve. To simplify the analysis while maintaining accuracy, we introduce the Kirchhoff-Love plate theory, which assumes:

1) Neglect of thickness changes and interlayer compressive stress.
2) Preservation of perpendicularity to the midplane.

This implies that the effects of shear stresses, τ_{xz} and τ_{yz} are negligible. As a result, the shear strains γ_{xz} and γ_{yz} are also zero. From this, the compatibility conditions for displacements can be derived as:

$$\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}=0, \quad \frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}=0, \tag{1}$$

where u and v are the displacements in the x and y directions, respectively, w is the vertical deflection of the plate.

3) In-plane constraints for points in the midplane.

The points in the midplane of the plate are assumed not to move in the in-plane directions. This results in a kinematic relationship between displacements and deflection w:

$$u=z\frac{\partial w}{\partial x}, \quad v=z\frac{\partial w}{\partial y}. \tag{2}$$

By relying on these assumptions, the theory reduces the computational complexity of analyzing plate behavior while providing a solid foundation for integrating FEM. This makes it particularly suitable for applications that require large-scale real-time stress distribution calculations, such as the robot localization system that uses underfloor pressure sensors investigated in this study.

A. Basic Equations of Deformation

Since the bending strains can be expressed as:

$$\epsilon_{xx}=-z\frac{\partial^2 w}{\partial x^2}, \quad \epsilon_{yy}=-z\frac{\partial^2 w}{\partial y^2}, \quad \gamma_{xy}=-2z\frac{\partial^2 w}{\partial x \partial y}, \tag{3}$$

where ϵ_{xx} and ϵ_{yy} are the bending strains along the x and y axes, γ_{xy} is the shear strain due to bending.

The kinematic equations link the bending moments to the bending stresses and bending strains. The bending stresses are expressed as:

$$\sigma_{xx}=-E\epsilon_{xx}, \quad \sigma_{yy}=-E\epsilon_{yy}, \quad \tau_{xy}=-\tfrac{E}{2(1+\mu)}\gamma_{xy}, \tag{4}$$

where E is Young's modulus and μ is Poisson's ratio. σ_{xx}, σ_{yy}, and τ_{xy} are the bending stresses in the respective directions.

B. Definition of Displacement Field Function

In our case, each node in the finite element model is defined with three degrees of freedom: vertical deflection w, rotation of the x-axis θ_x, rotation of the y-axis θ_y from four directions of neighbor nodes. The constraints for a unit node are given as:

$$\begin{aligned} w(x,y)&=\sum_{i=1}^{4} N_i(x,y)w_i \\ \theta_x(x,y)&=\sum_{i=1}^{4} N_{ix}(x,y)\theta_{xi} \\ \theta_y(x,y)&=\sum_{i=1}^{4} N_{iy}(x,y)\theta_{yi} \end{aligned} \tag{5}$$

where $N_i(x,y)$ are the shape functions associated with i -th node.

Based on the fundamental equations, we define the two rotational directions as follows:

$$N_{ix}=-\frac{1}{8}(1+\frac{x}{x_i})\left[(1+\frac{y}{y_i})^2\right], \tag{6}$$

$$N_{iy}=\frac{x_i}{8}(1-\frac{x}{x_i})(1+\frac{y}{y_i})\left[2+\frac{y}{y_i}\right]. \tag{7}$$

By organizing these expressions, we obtain:

$$[N]=\begin{bmatrix}[N_i] & [N_j] & [N_m] & [N_p]\end{bmatrix}, \tag{8}$$

$$[N_i]=\begin{bmatrix}N_i & N_{ix} & N_{iy}\end{bmatrix}, \tag{9}$$

where i, j, m, and p represent the four directions of the adjacent nodes, respectively..

C. Establishment of Unit Stiffness Matrix Equation

Based on the obtained displacement field function, we substitute it into the geometric equation, which can be expressed as:

$$\boldsymbol{\varepsilon_b}=z\mathbf{B_b}\mathbf{d_e}, \tag{10}$$

where $\mathbf{B_b}$ represents the bending strain-displacement matrix, $\mathbf{d_e}$ represents the nodal displacement vector of element.

Next, the strain energy U can be written based on the principle of minimum potential energy, as follows:

$$U = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon}^T \mathbf{D} \boldsymbol{\varepsilon} \, d\Omega, \tag{11}$$

where $\boldsymbol{\varepsilon}$ is the strain vector, $\mathbf{D}$ represents the constitutive matrix of the thin plate and Ω is the plate domain. Combining the kinematic equation with the strain energy expression, the unit stiffness matrix $\mathbf{K}^e$ for a thin plate element can be expressed as:

$$\mathbf{K}^e = \frac{t^3}{12} \int_{A^e} \mathbf{B}_b^T \mathbf{D} \boldsymbol{\varepsilon} \mathbf{B}_b \, dA^e, \tag{12}$$

where t is the plate thickness, A^e is the element area. Finally, using the stiffness matrix at each point of the thin plate surface, we can conduct FEM analysis for the entire structure.

D. FEM Stress Analysis

The stiffness matrices of the computed element K_e are assembled into the global stiffness matrix:

$$\mathbf{K}_{\text{global}} = \sum_{e=1}^{n} \mathbf{K}^e, \tag{13}$$

where n is the total number of elements in the model. The system of equations derived from the global stiffness matrix is solved to obtain the nodal displacements:

$$\mathbf{K}_{\text{global}} \mathbf{d} = \mathbf{f}, \tag{14}$$

where $\mathbf{d}$ is the global displacement vector and $\mathbf{f}$ is the global force vector.

Once the nodal displacements are obtained, the element strains and stresses can be computed using the strain displacement matrix introduced in Section II.B. Thus, we can obtain the results of the FEM analysis of the pressure distribution across the entire plane. The estimated pressure distribution is compared with the actual measured values from the pressure sensors to estimate the robot pose as described in the following section.

E. Simulation and Result

We set up the simulation environment for the robot and its pressure sensors as shown in Figure 2. The robot is modeled with four wheels and nine pressure sensors are arranged in a 3×3 grid on the ground to measure the pressure distribution caused by the robot's weight. We assumed that the floor is a flat, homogeneous surface, and the robot's wheels are modeled as four points where the pressure is exerted.

Figure 3 represents the normal stress distribution calculated by FEM. The pressure sensor stress shows a strong correlation with the distance from the load of the wheels. The values of the four pressure sensors around the load are significantly higher than those of the other points. In the optimization process of localization, to simplify the calculation, we take the values of these four points as input data to estimate the robot pose.

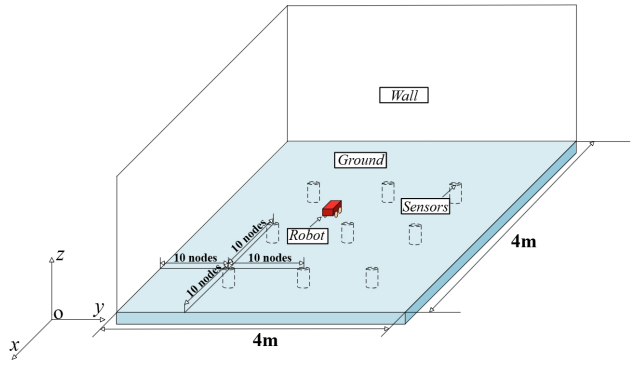


Fig. 2. Pressure sensor-based robot localization system in simulation environment.

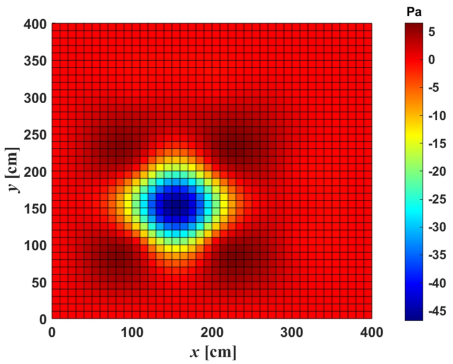


Fig. 3. Simulation result of normal stress distribution of the floor.

III. Robot Localization Based on Particle Filter

As presented in Section II, the stress distribution in the plate is calculated using FEM with the kirchhoff-love plate theory when the load position is known. Next, a particle filter-based approach is applied to estimate the pose of the robot. Specifically, the pressure values are predicted by performing FEM on each of the particles distributed according to the movement status of the mobile robot, and then compared with the measured value from the actual pressure sensor to optimize the distribution of the particles until they converge around the actual position of the robot.

A. Model Initialization

The state vector of the mobile robot in the 2D plane is defined as:

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix}, \tag{15}$$

where x_t and y_t represent the coordinates of the center of mass of the robot in a global coordinate system. θ_t denotes the robot orientation, defined as the angle between the robot's forward direction and the global x-axis. A region around the initial position (x_0, y_0) of the robot is defined as the initial search space for particles. Each particle is assigned a state vector:

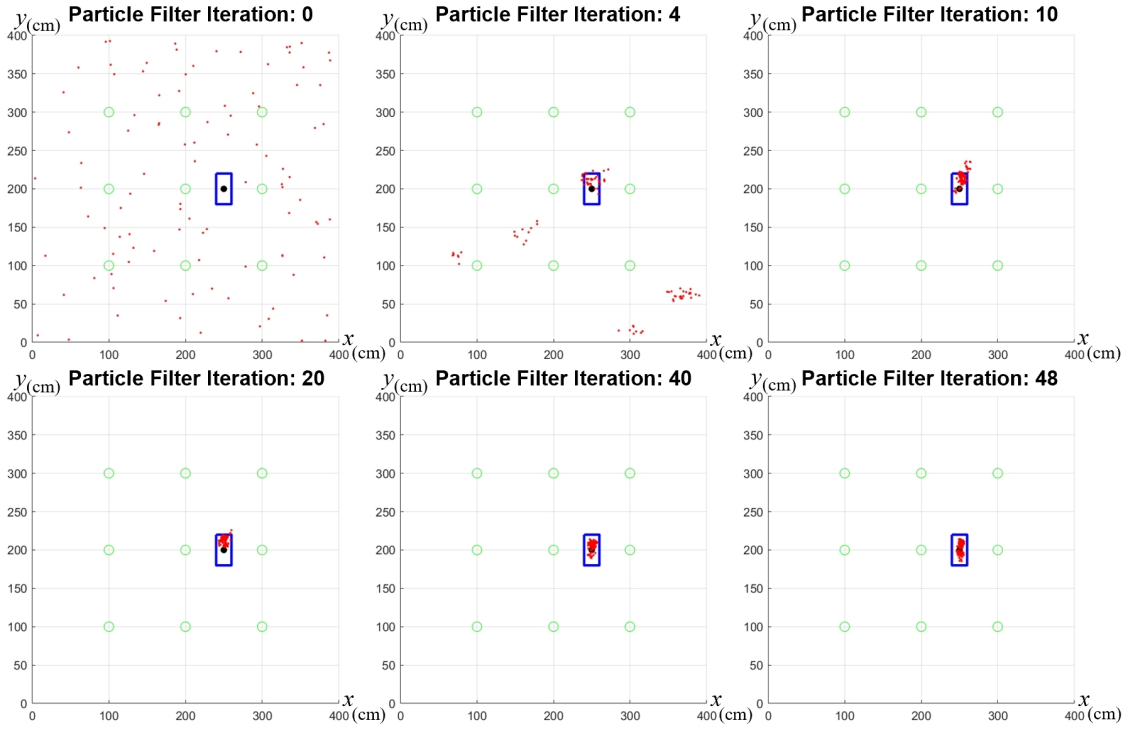


Fig. 4. Simulation results of particle filter-based robot position estimation. Each subfigure shows the particle distribution and estimated position at different iterations.

$$\mathbf{x}_0^{[i]} = \begin{bmatrix} x_0^{[i]} \\ y_0^{[i]} \\ \theta_0^{[i]} \end{bmatrix}, \tag{16}$$

where, $(x_0^{[i]}, y_0^{[i]})$ are sampled from a uniform distribution around the initial position, and $\theta_0^{[i]}$ is sampled from a uniform distribution in the range $[0, 2\pi]$. In our problem setting, the initial state of the robot is unknown; thus, the initial position is assigned at the center of the plane, and the particles are distributed over the entire area on the plane. The initial weight of each particle can be expressed as:

$$w_0^{[i]} = \frac{1}{M}, \tag{17}$$

where M represents the number of the generated particles.

B. Model State Estimation

The robot's motion is defined using a differential drive kinematics model. Therefore, the state of each particle $\mathbf{x}_t^{[i]} = [x_t^{[i]} \; y_t^{[i]} \; \theta_t^{[i]}]^\top$ is predicted as follows:

$$x_t^{[i]} = x_{t-1}^{[i]} + \left(v_t \cos\theta_{t-1}^{[i]} + \eta_x\right) \Delta t, \tag{18}$$

$$y_t^{[i]} = y_{t-1}^{[i]} + \left(v_t \sin\theta_{t-1}^{[i]} + \eta_y\right) \Delta t, \tag{19}$$

$$\theta_t^{[i]} = \theta_{t-1}^{[i]} + (\omega_t + \eta_\theta)\, \Delta t \tag{20}$$

where v_t and ω_t are the robot's linear and angular velocities at time t by computing the displacement and orientation change between the current frame and the previous frame, respectively. Δt is the duration of the time step. η_x, η_y, and η_θ represent Gaussian noise terms added to account for motion uncertainties.

The weight $w_t^{[i]}$ of each particle is updated based on the likelihood of the observed sensor data given the particle's predicted state:

$$w_t^{[i]} = \exp\left(-\frac{1}{2}(\mathbf{z}_t - \mathbf{z}_t^{[i]})^\top \mathbf{R}^{-1} (\mathbf{z}_t - \mathbf{z}_t^{[i]})\right), \tag{21}$$

where $\mathbf{z}_t$ is the actual pressure distribution from the sensors. $\mathbf{z}_t^{[i]}$ is the predicted pressure distribution based on FEM for the particle i. $\mathbf{R}$ is the covariance matrix that represents the sensor noise.

After the weight update step, the weights of all particles are normalized to ensure they sum to 1:

$$\tilde{w}_t^{[i]} = \frac{w_t^{[i]}}{\sum_{j=1}^{M} w_t^{[j]}}, \tag{22}$$

where $\tilde{w}_t^{[i]}$ is the normalized weight of particle i, and M is the total number of particles.

Then, the systematic resampling algorithm is employed for its computational efficiency as follows:

1) Calculate the cumulative sum of weights:

$$C_j = \sum_{i=1}^{j} \tilde{w}_t^{[i]}, \quad j = 1, \ldots, M. \tag{23}$$

2) Generate a random number u_1 uniformly distributed in $[0, \frac{1}{M}]$.

3) For $i = 1, \ldots, M$, calculate:

$$u_i = u_1 + \frac{i-1}{M}. \tag{24}$$

4) Select particle j such that:

$$C_{j-1} < u_i \leq C_j. \tag{25}$$

As mentioned above, the state estimation process integrates prediction, observation, normalization, and resampling steps to iteratively refine the robot pose estimation. The robot pose at time t, $\hat{\mathbf{x}}_t = [\hat{x}_t, \hat{y}_t, \hat{\theta}_t]^\top$, is calculated as the weighted average of all particles:

$$\hat{\mathbf{x}}_t = \sum_{i=1}^{M} w_t^{[i]} \mathbf{x}_t^{[i]}, \tag{26}$$

where $\mathbf{x}_t^{[i]}$ represents the i-th particle's state, and $w_t^{[i]}$ is its corresponding weight.

IV. Verification Experiments

We conducted experiments that included both simulation and real environments to verify our localization framework based on FEM stress analysis using pressure sensors under floor.

A. Simulation Test in MATLAB

We use the same simulation environment as in Section II.E as shown in Figure 2. To simulate sensor responses, we modeled the pressure exerted by each of the four robot wheels on the floor. The pressure sensors collected data corresponding to the total pressure at their respective locations. This model provided a pressure distribution that would be used as input for the particle filter localization algorithm. To reduce computational complexity, we only considered the four sensors with the highest pressure readings at each time step. This allows us to focus on the area with the most significant pressure distribution, thus reducing the number of particles that need to be evaluated.

Figure 4 illustrates the simulation results of using a particle filter to estimate the robot pose. Each subplot represents different iterations of the filter, showing the convergence of particles over time. The results demonstrate the effectiveness of the particle filter in estimating the robot's initial position with increasing accuracy over iterations. Figure 5 illustrates a comparison between the estimated trajectory generated by the particle filter and the robot trajectory. The results show that the estimated trajectory closely matches the actual trajectory, indicating the high accuracy of the proposed method to predict the robot pose. As shown in Figure 6, we compared the distance error achieved using the proposed particle filter-based method with simple position prediction and the Kalman filter. The proposed method achieves the smallest distance error among the three approaches. The simple position prediction method exhibits the highest distance error, as it does not account for environmental observations, causing it to accumulate significant errors over time. The Kalman filter improves upon this by incorporating sensor

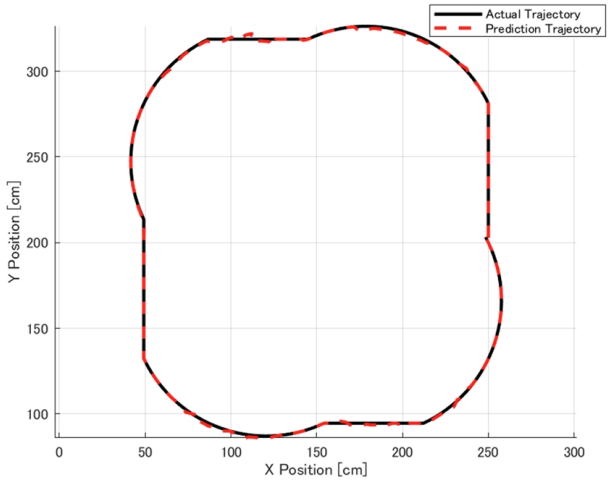


Fig. 5. Simulation result of trajectory prediction.

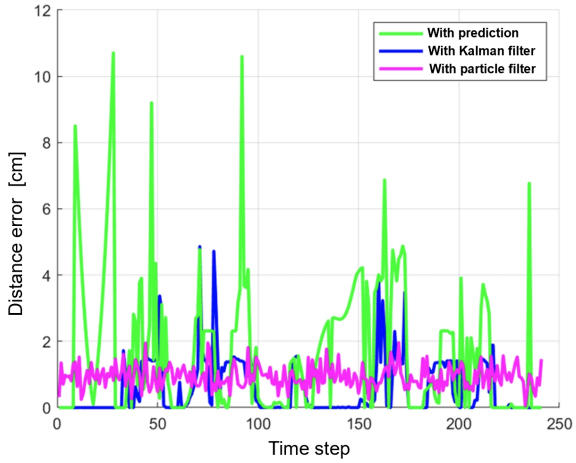
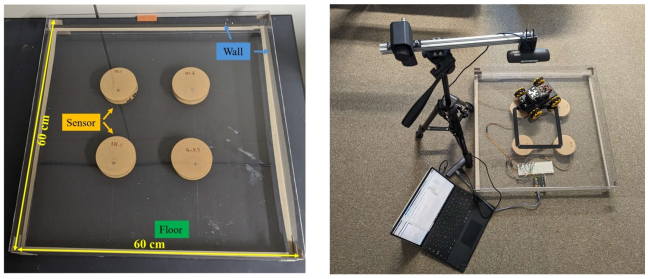


Fig. 6. Comparison result of error distance between proposed framework and other methods.

measurements. However, its performance is limited by the assumption of Gaussian noise and linear system dynamics, which do not fully capture the complexities of the indoor localization problem.

B. Real-World Environment Test

The real-world experimental setup is designed to evaluate the feasibility and precision of pressure-based localization in an indoor environment. As shown in Figure 7 (a), a floor was constructed using an epoxy resin board with dimensions of 60 cm × 60 cm. To enclose the testing area and prevent external disturbances, 10 cm high epoxy resin walls were installed around the perimeter of the floor. A 2 × 2 pressure sensor array, constructed with HX711 pressure sensors, was embedded beneath the simulated floor. The sensors were connected to an Arduino Uno, which collected pressure data and transmitted them to a computer via serial communication. The data were then processed using a particle filter algorithm to estimate the actual contact position based on the measured pressure distribution.



(a) (b)

Fig. 7. Real-world experimental setup with pressure sensor system (a) and mobile robot operation (b).

The validation experiment was performed with a mobile robot moving to estimate its trajectory over time. A camera positioned above the setup continuously detected an AprilTag attached to the robot, providing ground-truth position data throughout the experiment, as shown in Figure 7 (b). Every 50 ms, the robot collected data from the four pressure sensors as the observed value to estimate the robot's position. Figure 8 presents the test results that illustrate the ground-truth trajectory (marked in blue) and the estimated trajectory (marked in red). The results indicate that, while the particle filter-based algorithm successfully tracks the robot's motion, deviations from the planned trajectory are also evident. The error analysis in Figure 9 further demonstrates the inconsistencies in position estimation. The error fluctuates over time, suggesting that certain sections of the trajectory are more prone to inaccuracies, possibly due to abrupt turns.

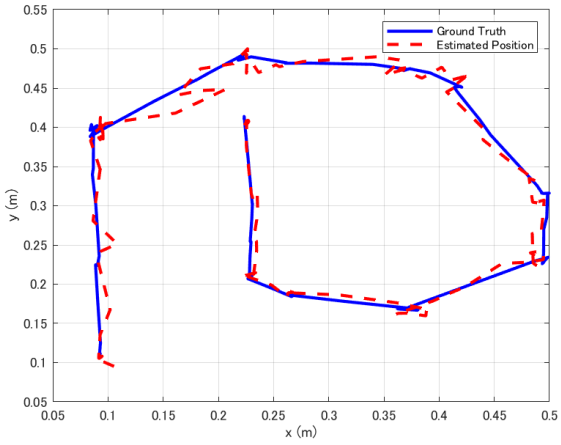


Fig. 8. Experimental results of robot trajectory estimation.

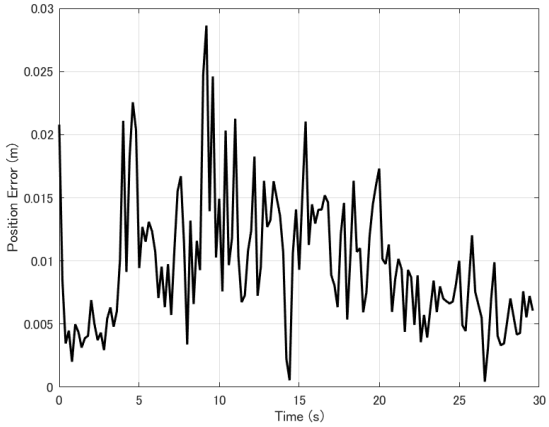


Fig. 9. Error of prediction trajectory from the actual planned trajectory.

V. Conclusion and Future Work

This study demonstrated a novel approach to mobile robot localization in indoor environments by integrating Kirchhoff–Love plate theory, FEM stress analysis, and a particle filter algorithm. Through theoretical modeling, simulation, and experimental validation, the proposed method was shown to effectively estimate the position of a robot based on pressure distributions measured by sensors embedded under the floor.

However, real-time implementation challenges persist. Future work should focus on optimizing the algorithm so as to increase the sampling frequency to reduce estimation delays and improving the sensor model to better accommodate environmental uncertainties. By refining these aspects, the real-time applicability of the system can be improved, bringing it closer to a precise and reliable trajectory estimation.

References

[1] I. Ullah, D. Adhikari, H. Khan, M. S. Anwar, S. Ahmad, and X. Bai, "Mobile robot localization: Current challenges and future prospective," *Computer Science Review*, vol. 53, 2024.

[2] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6019–6039, 2022.

[3] S. Campbell, N. O'Mahony, A. Carvalho, L. Krpalkova, D. Riordan, and J. Walsh, "Where am i? localization techniques for mobile robots a review," *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pp. 43–47, 2020.

[4] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (slam) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229–241, 2001.

[5] A. Yarovoi and Y. K. Cho, "Review of simultaneous localization and mapping (slam) for construction robotics applications," *Automation in Construction*, vol. 162, 2024.

[6] C. Debeunne and D. Vivet, "A review of visual-lidar fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, 2020.

[7] H. Yin, X. Xu, S. Lu, X. Chen, R. Xiong, S. Shen, C. Stachniss, and Y. Wang, "A survey on global lidar lo calization: Challenges, advances and open problems," *International Journal of Computer Vision*, pp. 1–33, 2024.

[8] B. Alsadik and S. Karam, "The simultaneous localization and mapping (slam)-an overview," *Journal of Applied Science and Technology Trends*, vol. 2, no. 02, pp. 147–158, 2021.

[9] B. Talu, I. Isik, B. Candiri, Y. Candiri, and R. B. Yapalıkan, "Effects of sensor size, surface material, and contact area on pressure measurements in thin-film pressure sensors," *IEEE Sensors Journal*, vol. 24, no. 2, pp. 1591–1597, 2024.

[10] Q. Yuan and J. Wang, "Design and experiment of the nao humanoid robot's plantar tactile sensor for surface classification," *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 931–935, 2017.

[11] R. B. Mishra, N. El-Atab, A. M. Hussain, and M. M. Hussain, "Recent progress on flexible capacitive pressure sensors: From design and materials to applications," *Advanced materials technologies*, vol. 6, no. 4, 2021.

[12] Tekscan, "Pressure sensing mat systems," 2024, accessed: 2024-12-28. [Online]. Available: https://www.tekscan.com/pressure-sensing-mat-systems