IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

A New Approach to Real-Time Odometry Calibration Using an Adaptive Particle Filter Design

Bibiana Fariña[1], Jonay Toledo[2] and Leopoldo Acosta (IEEE member) [3]

***Abstract*—This paper presents a novel calibration system for odometric sensors using an Adaptive Particle Filter (Adaptive-PF) to achieve high precision pose estimation and improve localization in wheeled mobile robots. The system reduces for intrinsic systematic errors in the odometric sensor by adjusting its parameters in realtime. Likewise, a comparative analysis of resampling methods —multinomial, stratified, systematic, and residual resampling— is conducted to evaluate their impact on calibration performance. The system validation is demonstrated by its implementation in an autonomous wheelchair, where the localization module integrates wheel encoders, an Inertial Measurement Unit (IMU), and a LIDAR sensor, providing robust navigation in dynamic environments. Experimental results demonstrate that systematic approach and resampling based on the effective number of particles (N_{eff}) yield the best performance. Additionally, the system dynamically adjusts prediction error based on the differences between LIDAR and odometry data. It also adapts the number of particles according to the dispersion and uncertainty, optimizing computational time without sacrificing accuracy. The proposed system outperforms another well-known method, namely the DKF (Dual Kalman Filter). Consequently, this research introduces a new Adaptive-PF for odometric parameter calibration under changing conditions.**

***Index Terms*—Calibration. Real-time parameter adjustment, Mobile Robots, Odometry, Particle Filter**

I. Introduction

Autonomous mobile robots rely on accurate localization systems to navigate safely. Real-time calibration and fusion data from multiple sensors play an important role in maintaining robust navigation, even in dynamic and changing environments. One of the most used methods for optimizing localization is the Particle Filter (PF), an algorithm that stands out for its ability to handle nonlinear state models with non-Gaussian noise [1]. Likewise, it has been widely implemented in various fields due to its versatility and robustness.

In mobile robots equipped with odometry systems, systematic errors can affect the accuracy of the estimated pose. To reduce these errors, real-time calibration of the odometry sensor is necessary to dynamically adjust system parameters during operation. In this paper we propose a PF-based calibration, analyzing the advantages and disadvantages of different strategies. Furthermore, an Adaptive Particle Filter (Adaptive-PF) is presented as a novel solution to this challenge. Unlike the standard version, this filter dynamically adjusts its parameters based on the received observations or the characteristics of the environment. This can include modifying the number of particles, adjusting the particle dispersion during resampling, or adapting the motion and measurement models based on feedback from the environment. This adaptability is particularly useful in mobile robot localization, as it allows the system to respond to changes in the environment and to situations where sensor uncertainty is high.

The validation of the proposed odometry calibration system was conducted using an autonomous wheelchair as a prototype for the mobile robotic platform. The advantages and disadvantages of different strategies used in the PF have been analyzed experimentally improving the precision and robustness of the wheelchair localization system and ensuring reliable navigation in dynamic environments.

Moreover, the proposed method is applicable to a variety of robotic platforms with an odometric system that needs to adjust some parameters, including wheeled, tracked, or legged robots. Furthermore, it is not limited to a specific set of sensors, it can integrate devices such as GNSS, radars, 3D cameras, range cameras, and other sensors that provide real-time pose measurements. This fact allows the implementation of the method across different scenarios and types of robots.

The paper is organized as follows, Section II reviews related work. Section III describes the PF for odometric calibration. Section IV compares calibration strategies within the PF. Section V presents the proposed Adaptive-PF. Experimental results with the wheelchair appear in Section VI, and conclusions in Section VII.

II. Previous Work

In localization systems of mobile robots, sensor calibration plays an important role in improving the accuracy of pose estimation. Traditionally, this calibration was performed offline, conducted in controlled environments prior to navigation, allowing for the correction of systematic sensor errors. In [2], vehicle odometry parameters are adjusted based on the discrepancies in the final pose along a specified trajectory. Similarly, in [3], a comparable approach is employed for mobile robots, where the robot navigates a square path to refine its odometric parameters. However, this approach has significant limitations as it cannot adapt to changes that occur during actual robot operation, such as wheel wear, terrain changes, or others factors that affect sensor accuracy. Performing calibration online, in contrast, enables the detection and correction of these dynamic changes during operation, improving localization accuracy over time and continuously adapting system parameters to current conditions.

A widely used method is the Kalman Filter (KF), which is particularly useful for estimating states in systems with

[1] [2] [3] are with Computer Science and System Department, Universidad de La Laguna, 38200 Santa Cruz de Tenerife, Spain; bfarinaj@ull.edu.es (B.F), jonay@isaatc.ull.es (J.T), leo@isaatc.ull.es (L.A).

Manuscript received xxxx xx, xxxx; revised xxxx xx, xxxx.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

limited nonlinearities and Gaussian noise [4]. This method includes the Augmented Kalman filter (AKF) that adjusts parameters by including them in the state vector [5] [6]. In this case the parameters are dynamically adjusted based on the observations, which allows improving estimation accuracy under changing conditions. For instance, [5] applies an AKF to an Autonomous Guided Vehicle (AGV). In [7] and [8], the authors implement an AKF to calibrate parameters using the pose (X, Y, θ) provided by a LIDAR. However, the literature is sparse, and its successful implementation in real world and indoors environments depends on several factors, like the availability of an external sensor that provides an absolute pose and sensor synchronization [9]. However, most sensors measure relative changes, typically translated into velocity measurements. To address this problem, [10] presents a realistic implementation of the AKF where it also solves the non-synchronization of sensorial data. On the other hand, the Dual Kalman Filter (DKF) allows simultaneous estimation of system states and model parameters, using two independent KFs simultaneously, one to estimate the state and the other to correct the parameters from the output of the previous state [11]. [12] proposes a DKF for an autonomous vehicle. Likewise, [13] uses a DKF to estimate the state and the unknown parameter of an optical parametric oscillator. However, despite their utility, these methods face important challenges in non-linear systems, where their performance can be affected.

In nonlinear systems, such as mobile robot motion modeling, the Kalman Filter has notable disadvantages due to its assumption of almost linear relationships between states and measurements. This approach is not suitable for the complexities of many real robotic systems, where system dynamics and sensor noises are often non-Gaussian and highly non-linear, which can lead to inaccurate estimations and cumulative errors [14]. To overcome these limitations, the Particle Filter (PF) has become a robust alternatively in robot localization. The PF approximates the probability distribution of system states using multiple particles, providing more accurate pose estimations even in complex environments [1].

The PF works by sampling a finite number of particles that represent possible states of the system. Each particle is weighted based on its consistency with observations, and the particles that best represent the probability of the true state are selected. However, its main drawback is the high computational cost, as ensuring accurate estimates can require a large number of particles, increasing processing time [15]. This limitation has been addressed by different strategies that lead to algorithm optimizations, including particle resampling methods [16], the resampling frequency, or the implementation of an Adaptive-PF, which dynamically modify the number of particles based on observations [17]. Additionally, [18] proposes a localization algorithm using a PF with parallel processing on GPUs to reduce computational costs. Similarly, [19] explores the implementation of PF using multicore architectures to optimize performance and reduce processing time.

The PF has been widely used in mobile robot localization, especially in pose estimation and multi-sensor data fusion. For example, [20] describes a system that employs ultrasonic sensors and a PF for precise indoor robot localization while avoiding obstacles. Likewise, [21] presents a real time self-localization algorithm for mobile robots using a PF. However, as far as the authors are concerned, no studies specifically address real-time odometry parameter calibration. Despite this, the use of PF for parameter calibration follows the same logic as in the AKF, with the added advantage of handling more complex and non-linear dynamics.Therefore, in wheeled robots, where odometry is prone to intrinsic systematic errors such as wheel diameter variation, these can be corrected in real time by an Adaptive-PF, which continuously adjusts the odometry parameters, ensuring greater localization accuracy.

III. PF-BASED CALIBRATION OF ODOMETRIC PARAMETERS

This section describes the algorithm designed for the calibration of odometric parameters. It presents the overall structure of the method, additional improvements and optimizations to the algorithm will be discussed later.

In the case of our robot, which uses differential drive kinematics, the state vector is represented as follows:

$$\chi = [V, W, R_R, R_L, D] \tag{1}$$

where V is the linear velocity of the robot, W is the angular velocity, R_R and R_L are the robot radii of the right and left wheel respectively, and D is the distance between the wheels.

The parameter correction is based on PF. The algorithm represents a distribution of possible solutions of the robot's pose, and then use sensor measurements to update this distribution for obtaining accurate estimation. The state relies on odometric equations to predict the robot's movement over time.

The PF structure involves the following steps [22]:

1) **Initialization**: A set of particles (S^i, where $i = 1...N$) is uniformly initialized within the robot's state space, with each particle χ_i representing a possible robot state. In the implementation described in the paper, the particle set consists of 5000 particles, **uniformly distributed within predefined ranges** for each parameter. The parameters R_R, R_L, and D are **initialized using a uniform distribution over the intervals** $[0.305, 0.335]\,\mathrm{m}$ for R_R, $[0.305, 0.335]\,\mathrm{m}$ for R_L, and $[0.55, 0.75]\,\mathrm{m}$ for D, based on known wheel specifications. The velocities (V, W) are both initialized to zero, reflecting the assumption of no initial movement.
2) **Prediction**: The prediction function is used to estimate the next state of each particle:

$$\hat{S}_k^i = f(S_{k-1}^i, u_{k-1}) + \epsilon_{k-1}^i \tag{2}$$

where u_k represents the available control inputs, ϵ_{k-1}^i is the process noise that captures the uncertainty in the robot's motion, and $f(\cdot)$ is a function describing the robot's prediction model. In this approach, $f(\cdot)$ corresponds to the odometric equations. The next state is derived from the current state and the angular velocity of each wheel (ω_r, ω_l) using eq. 3.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

$$\begin{bmatrix} V \\ W \\ R_r \\ R_l \\ D \end{bmatrix}_k = \begin{bmatrix} \frac{\omega_r R_{r_{k-1}} + \omega_l R_{l_{k-1}}}{2} \\ \frac{\omega_r R_{r_{k-1}} - \omega_l R_{l_{k-1}}}{D_{k-1}} \\ R_{r_{k-1}} \\ R_{l_{k-1}} \\ D_{k-1} \end{bmatrix} \tag{3}$$

The prediction noise (ϵ^i_{k-1}) is dynamically adjusted by introducing small random perturbations following a Gaussian distribution with a mean of 0 and a standard deviation of 0.0005. These perturbations are applied to R_R, R_L, and D, along with uniform sampling to account for uncertainties in the velocities (V and W).

3) **Update**: In this step, external sensor measurements of linear (V_m) and angular (W_m) velocities are used to update the particle weights by comparing them to the odometric model's predictions. Sensors like GPS, Doppler radar, or cameras can be used, provided their data is converted into linear and angular speeds.
The particle weights (ρ^i_k) are computed by evaluating the likelihood of the measured velocities, z_k, given the predicted velocities from the odometric model, $\hat{S}^i_k$. This is done using a probability function $P(z_k|\hat{S}^i_k)$. Once the weight corresponding to each individual measurement is determined, the overall weight of the particle is obtained by combining all available measurements. This process, described by Eq. 4, considers M independent measurements, where the total weight is computed as the product of the likelihoods associated with each measurement. The same procedure is applied to all particles in the set, yielding the weighted particle set S^i_k and their corresponding weights ρ^i_k.

$$\rho^i_k \approx P(z^j_k|\hat{S}^i_k) = P(z_1|S^i_k) \cdot P(z_2|S^i_k) \cdot \ldots \cdot P(z_M|S^i_k) \tag{4}$$

This approach ensures that particles whose predicted states closely match the sensor data are assigned higher probabilities. Each measurement is modeled as a Gaussian distribution ($\mathcal{N}$) with mean μ and standard deviation σ, where the mean corresponds to the predicted velocity values (V_m, W_m), and σ is determined by the noise characteristics of the sensor. A larger σ value indicates higher measurement uncertainty, leading to a broader probability distribution and, consequently, a greater spread of weights across multiple particles.

4) **Resampling**: Based on the weights, resampling is performed to generate a new set of particles. The first step is the normalization of the weights, Eq. 5, followed by a resampling process (Section IV describes different resampling strategies), which selects the new particles according to the probability distribution of $\tilde{\rho}^i_k$. Particles with higher weights are more likely to be selected, while those with lower weights are less likely to be chosen.

$$\tilde{\rho}^i_k = \frac{\rho^i_k}{\sum_{j=1}^{N} \rho^j_k} \tag{5}$$

The resulting radii at each time step are obtained as the weighted average of the particles (eq. 6). The winning particle is not selected because it is highly sensitive to potential errors in the measurements. Instead, this approach acts as a filter for the data, making the estimation process more robust.

$$\chi_k(R_R, R_L) = \sum_{i=1}^{N} \tilde{\rho}^i_k S^i_k \tag{6}$$

Figure 1 shows the process of online parameter calibration.

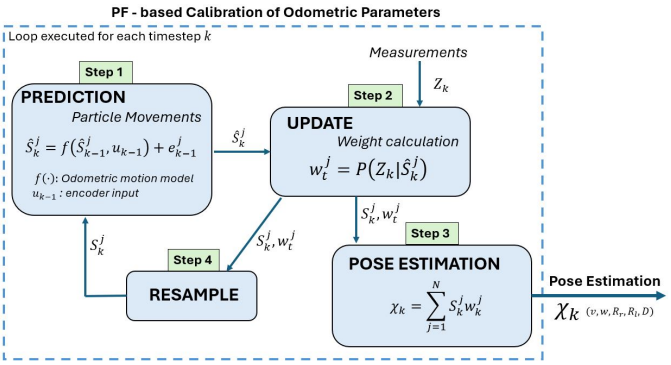


Fig. 1: Proposed scheme for the odometry calibration.

IV. Analysis and Comparison of PF Strategies

Different approaches to optimize the efficiency of the PF are explored, including several resampling techniques and the optimal resampling frequency to reduce computational time. This section analyzes and compares the experimental performance of these techniques for odometry parameter correction in an autonomous robot. Although D is part of the model, it is not analyzed here, as it remains constant during operation and has limited impact on pose estimation. The focus is on the wheel radii, which are more sensitive to real-world variations and more influential in odometric errors.

A. Robot sensor system description

This method is validated in the localization system of a smart wheelchair, the prototype is shown in Figure 2, but it can be applied to any wheeled robot. The wheelchair is designed to improve the mobility of severely disabled people, helping users move with less effort and improving their quality of life. The user controls the wheelchair by specifying the desired final pose, while a set of sensors detects obstacles and reconstructs the environment to ensure safe and autonomous navigation [23] [24] [25]. The wheelchair makes low-level decisions, such as avoiding obstacles and selecting the best path to the desired destination, where the localization subsystem plays an important role in the movement process.

The wheelchair contains the following sensors:

- *Odometry*. Incremental encoders coupled to the wheel motors to capture motion in real time. These encoders provide a resolution of 8800 pulses per wheel revolution, corresponding to an angular resolution of 0.04 degrees. With an wheel radius of 0.32 meters, the system can

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Fig. 2: Intelligent wheelchair with the set of sensors used in the localization module.

achieve a positional resolution of 0.1 mm, making it very accurate for short-distance movements. The estimated pose accumulates the error over time, leading to increasing uncertainty as the robot moves. This problem makes online calibration necessary for an accurate localization.

- *IMU*. An MPU9250 module. The IMU consists of a 3-axis gyroscope, accelerometer, and compass. The gyroscope measures angular velocity, while the accelerometer captures linear acceleration. Although the compass can determine the north direction, its reliability decreases indoors due to magnetic interference, which limits its contribution to the localization system. Therefore we will use only the measurements provided by the gyroscope.
- *Lidar*. The Sick LMS111 sensor is placed on the back of the wheelchair. The pose estimation provided by the sensor is calculated using the scan matching technique called Iterative Closest Point (ICP) [26]. It estimates the incremental pose of the robot and dynamically calculates the sensor uncertainty at each time, by comparing two consecutive scans.

The wheelchair is equipped with sensors that measure (V) or (W) of the robot. However, the proposed approach is general and can be applied with any other set of sensors capable of providing robot velocities.

B. Re-sampling Methods

Resampling is an important component of the PF algorithm. It ensures the elimination of particles with low weights and prevents the state estimation from being predominantly influenced by few particles. In particular, this study compares the efficacy and computational cost of various resampling techniques to estimate the odometry radii. The most commonly used resampling methods are:

- **Multinomial Resampling**: N random numbers $u_n \sim U(0,1)$ are generated independently to select particles based on their normalized weights. Each particle is chosen by finding the smallest index j such that:

$$C_j \geq u_n \tag{7}$$

where C_j is the cumulative sum of the normalized weights. This method is simple but has high variance, which can lead to rapid particle degeneration.

- **Stratified Resampling**: The interval $(0,1]$ is divided into N subintervals $\left(\frac{n-1}{N}, \frac{n}{N}\right)$ that generate a random number in each interval, n. This method ensures a balanced selection, reducing variance and improving particle diversity.
- **Systematic Resampling**: A single random number $u_1 \sim U(0, \frac{1}{N})$ is generated, and the remaining values are computed as:

$$u_n = u_1 + \frac{n-1}{N}, \quad n = 2, \ldots, N \tag{8}$$

This method ensures uniform selection, further reducing variance efficiently.

- **Residual Resampling**: Particles with weights $\rho_i > 1/N$ are deterministically replicated a number of times equal to the integer part of $n_i = N\rho_i$. Then, the remaining particles $m = N - \sum n_i$ are selected randomly using the residuals of the weights:

$$\hat{\rho}_i = N\left(\rho_i - \frac{n_i}{N}\right) \tag{9}$$

This combines deterministic and random selection to minimize variance.

Figure 3 shows the experimental comparison of the resampling methods used to correct odometry parameters. The comparison includes two metrics: the normalized error in estimating wheel radii and the average computation time per iteration, including the resampling process.

The wheel radii error refers to the difference between the final radius estimated by the algorithm and the true radius values, which were obtained through manual measurements. The initial radius values were intentionally set incorrectly to test the system's ability to recalibrate under realistic variations. These errors are normalized by taking the maximum error among all methods and scaling the others proportionally, so the maximum error is equal to 1. Similarly, the computation times are normalized in the same way and are represented as the translucent bars in the figure. The black error bars at the top of each column indicate the standard deviation of the respective metric, which has also been scaled by a factor to ensure visibility in the plot.

This comparison demonstrates that the differences in the error and computation time between these methods are minimal, indicating similar performance. However, systematic resampling has been selected as the preferred method due to its balance between simplicity and efficiency. Although computation times are similar between methods, systematic resampling provides computational efficiency while maintaining low levels of error and variance. Additionally, its practical advantages and strong experimental results for odometry correction make it a suitable choice for this application.

C. Re-sampling Frequency

In this section, we compare different resampling frequencies used in the PF. We analyze three strategies:

- **Always resampling**: The resampling is performed at every iteration, regardless of the particle distribution. This method ensures that particle degeneracy is consistently avoided, but it introduces additional processing time due

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

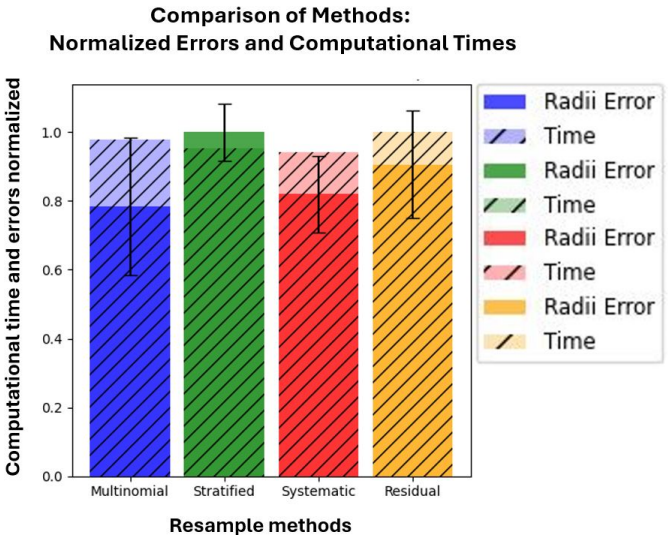


Fig. 3: Error and computation time by resampling method.

to the constant resampling, which may be unnecessary when the particle set is already well distributed.

- **Resampling based on the effective number of particles**: Effective number of particles N_{eff} is calculated as:

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} \rho_i^2} \tag{10}$$

Resampling happens when N_{eff} falls below a predefined threshold N_{th}. We use a threshold of $0.5 \times N$, following common practice in the PF literature [27] [28]. This approach balances the need for resampling by only applying it when particle degeneration is significant, thus reducing unnecessary computations time while maintaining the filter accuracy.

- **Resampling when the maximum weight falls below a threshold**: Resampling is performed only when the maximum particle weight ρ_{max} falls below a certain threshold. It aims to ensure that no single particle dominates the distribution significantly. It can delay resampling, conserving computational resources, but may compromise performance if particle degeneration occurs too late.

Figure 4 compares the error in estimating the system state across multiple trials with different deviations in the initial estimation of the odometric radii. Additionally, the average computational time of the PF algorithm has been evaluated during the paths, considering the different resampling frequencies for each technique mentioned in the previous section. We experimentally compared the error in the radius estimation and the computation time using the following frequencies: constant resampling (green column), resampling based on N_{eff} (blue column), and resampling when the maximum weight is below a threshold (orange column).

The results show that both the N_{eff} and the constant resampling method provide high accuracy in estimating the radii, while the method based on the maximum weight tends to produce the least accurate results. In terms of runtime, the constant resampling method is the slowest, followed by N_{eff}, with the maximum weight method being the fastest. Based on these findings, the N_{eff} method can be considered as the best alternative, as it balances accuracy and computational cost.

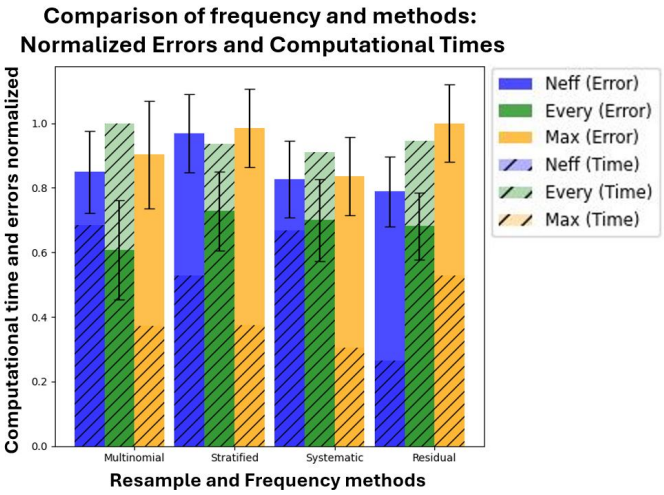


Fig. 4: Comparison of resampling methods and frequencies.

These results are also reflected in Table I, which summarizes the findings from multiple experiments. The table shows the root mean square error (RMSE) of the linear and angular odometric velocities obtained by parameter correction. It also includes the average final errors for each estimated radius, and the average execution time in each iteration.

This comparison demonstrates the similar performance of the four methods, as highlighted in the previous section, where the advantages of systematic resampling were also mentioned. Therefore, based on this experimental comparison, we can conclude that the systematic method with a resampling frequency based on N_{eff} is the best alternative for our case.

V. Adaptive PF proposed

In this section, we propose an Adaptive Particle Filter (Adaptive-PF) to improve the performance of the calibration. There are several approaches in the literature that suggest online modifying the number of particles to optimize PF. KLD sampling [29] adjusts the number of particles based on the Kullback-Leibler Divergence (KLD) between the particle and the target distributions. The KLD measures the difference between these two distributions: if it exceeds a predefined threshold, the number of particles is increased, and, if it is lower, the number of particles is reduced. This approach ensures that the filter maintains an accurate state by adapting the number of particles according to data variability [30].

In this work, we chose to adjust the number of particles directly based on their dispersion and in a proportional way. This adjustment ensures that the sample space is always adequately covered. Thus, when greater particle dispersion is observed, the system increases N to explore a larger space and improve estimation accuracy. In contrast, when the dispersion is smaller, N is reduced, optimizing computational performance without compromising accuracy.

The Adaptive-PF starts with an initial count of 8000 particles, which are dynamically adjusted based on the state

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: Comparison of resampling methods and frequency techniques.

Method	Freq.	Err Final Rr (m)	Err Final Rl (m)	Err Velocity		Time (ms)
				V (m/s)	w(rad/s)	
Stratified	neff	0.002623	0.00080	0.005167	0.01750	1.11274
	max	0.001967	0.003010	0.006599	0.026072	0.71592
	every	0.001645	0.00012373	0.0038214	0.016504	1.7896
Multinomial	neff	0.0006603	0.001845	0.006157	0.017341	1.31104
	max	0.0019943	0.001488	0.00764	0.029885	0.71322
	every	0.0006329	0.001029	0.0055483	0.0162425	1.9134
Residual	neff	0.001351	0.001967	0.00376	0.016399	1.01067
	max	0.0011549	0.002067	0.091024	0.024952	0.506289
	every	0.000729	0.0011176	0.002707	0.015864	1.81052
Systematic	neff	0.001983	0.001067	0.006877	0.01689	1.2079
	max	0.001868	0.0020041	0.007044	0.028287	0.58438
	every	0.000459	0.00063199	0.0049737	0.0145311	1.74098

dispersion of the radii. Specifically, the adjustment is designed to remain precise, where N is updated proportionally so that a dispersion of 0.01 mm in a radius corresponds to one particle. This proportional adjustment allows the system to maintain a fine balance between exploration and computational efficiency, ensuring accurate estimations throughout calibration.

In addition to adapting the particle number, this method also adjusts the process error, ϵ^i_{k-1} proportionally to the similarity between the linear and angular velocities of LIDAR (V_m, W_m) and the PF odometry (V_O, W_O) obtained by weighted average of the particles. In this case, the standard deviation (σ) of ϵ^i_{k-1} is dynamically determined by eq. 11

$$\sigma_{\epsilon^i_{k-1}} = \alpha \cdot \sqrt{(V_m - V_O)^2 + (W_m - W_O)^2} \qquad (11)$$

Here, α is a coefficient that adjusts the influence of the error. In this implementation $\alpha = 0.0001$. The random nature of the process error is bounded by the value $\sigma_{\epsilon^i_{k-1}}$, which means that the perturbations introduced during the prediction phase are scaled according to this value. As the radii are corrected, it is expected that the velocities become more similar, resulting in a decrease in $\sigma_{\epsilon^i_{k-1}}$, thereby reducing the noise and improving the prediction accuracy. Conversely, a significant discrepancy between the velocities indicates increased system noise, leading to a broader error range. In such cases, with higher data uncertainty, the particle filter increases the dispersion and the number of particles to maintain accuracy.

Figure 5 shows the behavior of this Adaptive-PF method. The graph presents the evolution of the particle number N for a path, where the conditions vary midway, increasing data uncertainty. The method adapts by adjusting the number of particles, which is directly proportional to the dispersion. This demonstrates how the approach responds to changing conditions in the system.

VI. Results

This section presents the results of the proposed Adaptive-PF using the wheelchair. The method was validated through multiple paths, analyzing wheel radii and odometry poses. Figure 6.a shows a Monte Carlo simulation with random initial misaligned radii for both wheels, where the radii converge to actual values over time. Figure 6.b depicts a scenario with

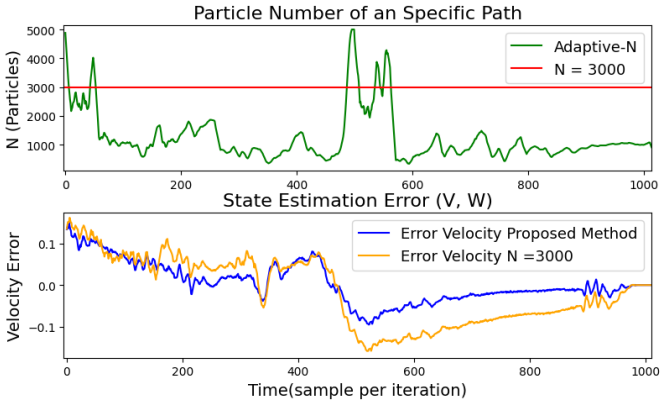


Fig. 5: Evolution of particle count during a trajectory. State estimation error is compared between the proposed method and a fixed-count baseline (N = 3000)

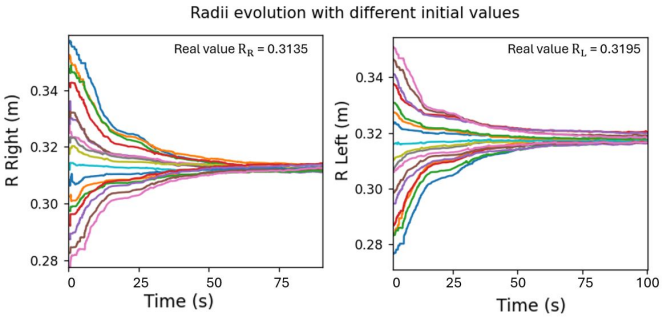
changing conditions after 55 seconds, such as a new user boarding. This affects the radii, but the method adjusts process noise and recalibrates, preserving localization accuracy.

Furthermore, we have tested how this new odometry approach improves the odometric estimation of the wheelchair pose during a trajectory. For this purpose, the estimated pose from our method and odometry without calibration was compared against the reference trajectory. In this case, the ground truth is obtained using a Velodyne sensor, which uses the LOAM algorithm to calculate the robot pose [31]. It is a high precision sensor, but it is worth mentioning that this sensor in a low-cost localization module may be cost prohibitive.

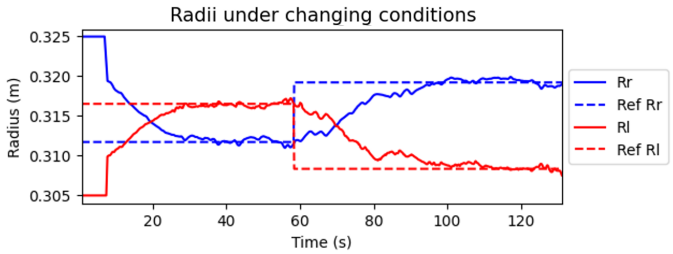
Figure 7, show three methods: traditional odometry without any adjustments (green line), odometry with calibration via Adaptive-PF (blue line), and the ground truth provided by the Velodyne LIDAR (black line). In the figure, the trajectory estimated by our proposed method is more accurate, resulting in a odometry pose closely to the ground truth.

Additionally, a comparison has been conducted with a well-known approach, the Dual Kalman Filter (DKF). Figure 8 shows how both methods correct the wheel radii over time. However, our method converges significantly faster to the actual values. In this case, the proposed method quickly adapts

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) Adaptive-PF Radii Evolution in Monte Carlo simulation.



(b) Radii evolution using Adaptive-PF in changing conditions.

Fig. 6: Radius adjustment to true values using Adaptive-PF.

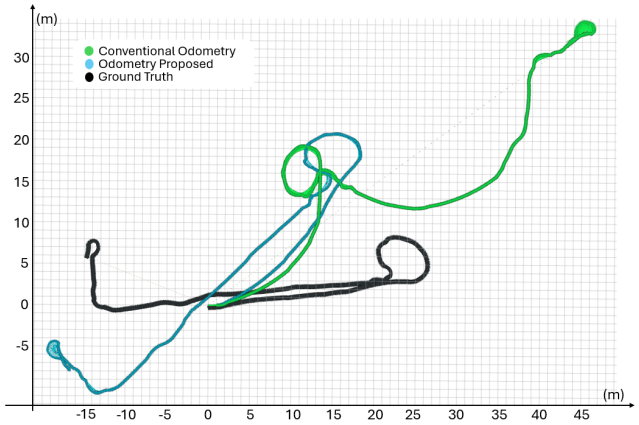


Fig. 7: Odometry pose estimation for a specific path.

to the change by adjusting the radii and maintaining the localization accuracy, while the DKF responds slower.

This fact is also supported by Table II, where multiple trajectories have been analyzed. The table summarizes the RMSE of wheelchair positions over time for different methods (Adaptive-PF, DKF, and without calibration), as well as the average final error of the estimated radii. Additionally, in the Monte Carlo simulations with various initial values, we recorded the average execution time per iteration and the convergence time for both methods. Convergence is considered to have been reached when the error falls below 0.5% of the true value. The results indicate that, while the Particle Filter approach (Adaptive-PF) has a higher computational cost per iteration, it converges more quickly to the true parameters than DKF, resulting in a lower pose error and a more accurate trajectory. Moreover, the odometry derived from our algorithm yields lower errors compared to the other method, highlighting

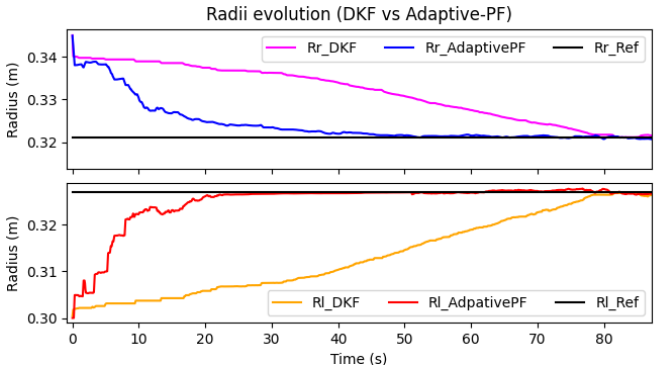


Fig. 8: Radii evolution (DKF vs Adaptive-PF)

the effectiveness of the proposed calibration.

TABLE II: Results for different odometry methods.

Metric	Adaptive-PF	DKF	No-calibration
Err_{Rr} (m)	0.00186	0.00211	-
Err_{Rl} (m)	0.00191	0.00185	-
$RMSE_{(x,y)}$ (m)	3.3761	5.438	17.114
$RMSE_{\theta}$ (rad)	0.5853	0.9410	2.0601
Average iteration time (ms)	0.8911	0.4596	0.1377
Convergence time (s)	32.55	60.73	-

To evaluate the generalization of the method, we used the available FusionPortable v2 dataset [32], which features sensor data from a different robotic platform and several environments. As shown in Figure 9, Adaptive-PF accurately estimates the wheel radii from incorrect initial values and converges to the true parameters. These results demonstrate the method's robustness and adaptability across different systems.

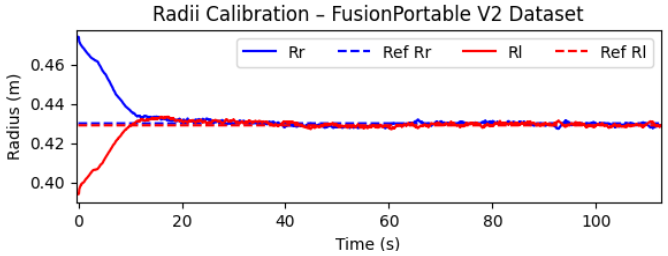


Fig. 9: Wheel radii evolution using the FusionPortable v2.

VII. Conclusions

This work presents an innovative PF based algorithm to dynamically calibrate the odometric parameters of an autonomous robot, improving the accuracy of its localization system. It represents a significant contribution to mobile robotics and localization, as no prior literature has specifically addressed odometric calibration using an Adaptive-PF. The developed algorithm corrects errors in the wheel radii estimation adapting to changes in operating conditions, such as variations in user weight or environmental characteristics. A comparative analysis was conducted by its implementation in an autonomous wheelchair, highlighting systematic resampling and effective particle number (N_{eff})-based resampling as the most efficient method, balancing accuracy and computational

cost. Furthermore, it also presents an adaptive approach that dynamically adjusts both, the number of particles based on dispersion and, the process error prediction noise according to the similarity between LIDAR measurements and odometry velocities, ensuring greater localization accuracy while remaining computationally feasible.

Experimental results confirmed that adjusting odometric parameters improves pose estimation over a well-known method. Thus, our approach improve the PF algorithm by reducing common systematic odometry errors in real systems.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution from the Spanish Ministry of Science and Technology under the SIRTAPE project DPI2017-90002-R.

REFERENCES

[1] D. Talwar and S. Jung, "Particle filter-based localization of a mobile robot by using a single lidar sensor under slam in ros environment," in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, 2019, pp. 1112–1115.

[2] K. Lee and Woojin Chung, "Calibration of kinematic parameters of a car-like mobile robot to improve odometry accuracy," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 2546–2551.

[3] Y. Maddahi, "Design and laboratory tests of wheeled mobile robots," in *Proceedings of the 4th WSEAS/IASME Int. Conf. on System Science and Simulation in Engineering*, 01 2005, p. 185–190.

[4] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, 2015, pp. 74–77.

[5] T. Larsen, M. Bak, N. Andersen, and O. Ravn, "Location estimation for an autonomously guided vehicle using an augmented kalman filter to autocalibrate the odometry," in *First International Conference on Multisource-Multisensor Information Fusion 1998, FUSION'98 ; Conference date: 01-01-1998*, 1998.

[6] I. Hassanzadeh and M. Fallah, "Design of augmented extended and unscented kalman filters for differential-drive mobile robots," *Journal of Applied Sciences*, vol. 8, 12 2008.

[7] A. Martinelli, N. Tomatis, A. Tapus, and R. Siegwart, "Simultaneous localization and odometry calibration for mobile robot," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, vol. 2, 2003, pp. 1499–1504 vol.2.

[8] A. Martinelli, "Modeling and estimating the odometry error of a mobile robot," *IFAC Proceedings Volumes*, vol. 34, no. 6, pp. 407–412, 2001, 5th IFAC Symposium on Nonlinear Control Systems 2001, St Petersburg, Russia, 4-6 July 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017352096

[9] S. Maeyama, Y. Takahashi, and K. Watanabe, "A solution to slam problems by simultaneous estimation of kinematic parameters including sensor mounting offset with an augmented ukf," *Advanced Robotics*, vol. 29, 09 2015.

[10] B. Fariña, J. Toledo, and L. Acosta, "Augmented kalman filter design in a localization system using onboard sensors with intrinsic delays," *IEEE Sensors Journal*, vol. 23, no. 11, pp. 12 105–12 113, 2023.

[11] E. Wan and A. Nelson, "Dual kalman filtering methods for nonlinear prediction, smoothing, and estimation," *Advances in Neural Information Processing Systems*, 08 2000.

[12] C. Li, Y. Liu, L. Sun, Y. Liu, M. Tomizuka, and W. Zhan, "Dual extended kalman filter based state and parameter estimator for model-based control in autonomous vehicles," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 327–333.

[13] Q. Yu, S. Yokoyama, D. Dong, D. McManus, and H. Yonezawa, "Simultaneous estimation of parameters and the state of an optical parametric oscillator system," 2021. [Online]. Available: https://arxiv.org/abs/2111.07249

[14] I. Ullah, X. Su, X. Zhang, D. Choi, Z. Hou, and J. Zhu, "Evaluation of localization by extended kalman filter, unscented kalman filter, and particle filter-based techniques," *Wireless Communications and Mobile Computing*, vol. 2020, p. 15, 10 2020.

[15] Z. Xue and H. Schwartz, "A comparison of mobile robot pose estimation using nonlinear filters: Simulation and experimental results," *International Journal of Mechatronics and Automation*, vol. 5, p. 92, 01 2015.

[16] T. Li, M. Bolic, and P. M. Djuric, "Resampling methods for particle filtering: Classification, implementation, and strategies," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 70–86, 2015.

[17] D. Fox, "Kld-sampling: Adaptive particle filters," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. MIT Press, 2001. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2001/file/c5b2cebf15b205503560c4e8e6d1ea78-Paper.pdf

[18] Y. Yang, W. Zhang, F. Li, D. Zhang, and Y. Liu, "3d/6dof particle filtering location algorithm based on gpu parallel acceleration," in *2023 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2023, pp. 1527–1532.

[19] C. Paz, G. Perez Paina, L. Canali, and J. Toloza, "Filtro de partículas para localización de robots móviles implementado en arquitecturas multi-núcleo," 11 2012.

[20] T. Grami and A. Sghaier Tlili, "Indoor mobile robot localization based on a particle filter approach," in *2019 19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 2019, pp. 47–52.

[21] F. Luo, B. Du, and Z. Fan, "Mobile robot localization based on particle filter," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, 2014, pp. 3089–3093.

[22] S. Katwe, N. Iyer, M. Khan, M. Peters, and M. Mahale, "Particle filter based localization of autonomous vehicle," in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, 2021, pp. 1–6.

[23] R. Arnay, J. Hernández-Aceituno, J. Toledo, and L. Acosta, "Laser and optical flow fusion for a non-intrusive obstacle detection system on an intelligent wheelchair," *IEEE Sensors Journal*, vol. 18, no. 9, pp. 3799–3805, May 2018.

[24] N. Morales, R. Arnay, J. Toledo, A. Morell, and L. Acosta, "Safe and reliable navigation in crowded unstructured pedestrian areas," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 74 – 87, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197615002638

[25] R. Arnay, N. Morales, A. Morell, J. Hernandez-Aceituno, D. Perea, J. T. Toledo, A. Hamilton, J. J. Sanchez-Medina, and L. Acosta, "Safe and reliable path planning for the autonomous vehicle verdino," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 2, pp. 22–32, Summer 2016.

[26] M. Brossard, S. Bonnabel, and A. Barrau, "A new approach to 3d icp covariance estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 744–751, 2020.

[27] A. Sveier and O. Egeland, "Dual quaternion particle filtering for pose estimation," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 5, pp. 2012–2025, 2021.

[28] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[29] R. P. Guan, B. Ristic, L. Wang, and J. L. Palmer, "Kld sampling with gmapping proposal for monte carlo localization of mobile robots," *Information Fusion*, vol. 49, pp. 79–88, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253517303305

[30] H. Zhang, N. Chen, and G. Fan, "An improved localization algorithm for intelligent robot," in *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2019, pp. 1–5.

[31] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, 2014.

[32] H. Wei, J. Jiao, X. Hu, J. Yu, X. Xie, J. Wu, Y. Zhu, Y. Liu, L. Wang, and M. Liu, "Fusionportablev2: A unified multi-sensor dataset for generalized slam across diverse platforms and scalable environments," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 02783649241303525, 0. [Online]. Available: https://doi.org/10.1177/02783649241303525