



**HAL**  
open science

# A new resampling algorithm for particle filters and its application in global localization within symmetric environments

Seif Eddine Seghiri, Noura Mansouri, Ahmed Chemori

► **To cite this version:**

Seif Eddine Seghiri, Noura Mansouri, Ahmed Chemori. A new resampling algorithm for particle filters and its application in global localization within symmetric environments. Transactions of the Institute of Measurement and Control, inPress, 10.1177/01423312241267042 . lirmm-04675915

**HAL Id: lirmm-04675915**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-04675915v1>**

Submitted on 23 Aug 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# A New Resampling Algorithm for Particle Filters and its Application in Global Localization within Symmetric Environments

Journal Title  
XX(X):1-24  
©The Author(s) 2023  
Reprints and permission:  
sagepub.co.uk/journalsPermissions.nav  
DOI: 10.1177/ToBeAssigned  
www.sagepub.com/

SAGE

Seif eddine Seghiri<sup>1</sup>, Noura Mansouri<sup>1</sup> and Ahmed Chemori<sup>2</sup>

## Abstract

Mobile robots are undergoing tremendous development, which makes them employed in many fields. In this area, global localization in symmetric indoor environments is a commonly encountered problem. One of the commonly used algorithms to solve it, is the Adaptive Monte Carlo Localization (AMCL), which is based on the particle filter algorithm. In this paper, we developed a new algorithm for resampling used within the Adaptive Monte Carlo Localization (AMCL) framework that we called Effective Samples Resampling (ESR). The proposed algorithm is based on a deterministic sample selection and it is thoroughly tested in real time. Using a considerable amount of simulations, the efficacy and robustness of the AMCL using this technique are validated and compared to certain conventional approaches. They are also tested and validated in various real-time operating conditions using the Robot Operating System (ROS). The obtained results are quite satisfying in terms of resampling quality, implementation complexity, and convergence time when compared to random resampling approaches where a sample-based probability density given by high quality sensors might destabilize localization. The global localization is well handled when the proposed algorithm is involved, compared to standard resampling algorithms that can often be overconfident and fail in some scenarios when there is a lot of symmetry in the considered map of the environment.

## Keywords

Adaptive Monte Carlo Localization, Particle filter, Resampling, Global localization, ROS, Autonomous Navigation

---

<sup>1</sup>LARC, University of Constantine 1, Constantine, Algeria

<sup>2</sup>LIRMM, University of Montpellier, CNRS, Montpellier, France

## Corresponding author:

Seif eddine Seghiri, LARC, University of Constantine 1, Constantine, Algeria.  
Email: seifeddine.seghiri@umc.edu.dz

## Introduction

Navigation is a difficult challenge for autonomous robotic systems. The robotic system must complete four steps in order to properly navigate its environment: perception, localization, cognition, [Julius Fusic et al. \(2022\)](#) and motion control [Shen et al. \(2023\)](#). Localization is one of the most essential challenges among these four components; it is the phase in which the robot must be able to continuously estimate its position in the environment under consideration. [Forbes \(2013\)](#); [Yilmaz and Temeltas \(2021\)](#). According to the type of information initially available, three main sub-problems can be discerned in mobile robot localization: *Position Tracking*, *Global Localization* and *Kidnapped Robot Problem* [Zhang et al. \(2012\)](#). All of these sub-problems are different from each other in terms of computational requirements and complexity, especially for large symmetric environments. In position tracking problem, the initial pose of the robot is assumed to be known, so the localization algorithm tracks the robot using its control inputs and measurements [Naab and Zheng \(2021\)](#). For the global localization problem, the robot attempts to locate itself on a map without knowing its initial pose. Hence, more effective techniques are needed to determine it [Su et al. \(2017\)](#); [Se et al. \(2001\)](#). Another difficult localization challenge is the kidnapped robot problem. It is the hardest among the three [Thrun et al. \(2000a,b\)](#); [Desrochers et al. \(2015\)](#), and could be seen as a variant of the global localization problem, where localization failures are defined as a kidnapping problem [Kuptamete and Aunsri \(2022\)](#).

Furthermore, the global localization and the kidnapping problem are deeply connected to the environment's symmetric redundancy and the quality of the sensors [Laaksonen and Kyrki \(2008\)](#); [Thomas et al. \(2021\)](#). To resolve mobile robot localization problems, different approaches have been proposed in the literature [Yu et al. \(2020\)](#); [Roy and Chowdhury \(2021\)](#). Monte Carlo Localization (MCL) algorithm is a probabilistic localization technique that has been shown to be a reliable approach to mobile robot localization. It is a Particle Filter (PF) based algorithm, which uses sampling techniques to represent the robot's pose belief [Dellaert et al. \(1999\)](#); [Rosas-Cervantes and Lee \(2020\)](#). However, this algorithm uses a fixed small sample set size when the position of the robot is approximately known, and a larger size if not, which makes it computationally expensive. The Adaptive Monte Carlo Localization (AMCL) algorithm, proposed in [Fox \(2001\)](#), uses adaptive sample set size to increase the efficiency of MCL algorithm [Fox et al. \(2003\)](#); [Thrun et al. \(2005\)](#). It uses Kullback-Leibler divergence to infer the difference between the sample-based and the target densities [Guan et al. \(2019\)](#). Zhang et al. [Zhang et al. \(2012\)](#) proposed another adaptive type of MCL technique called Self Adaptive Monte Carlo Localization (SAMCL). The authors used the concept of Similar Energy Region (SER). Instead of randomly distributing particles in the map, the method consists of using a set of poses having similar energy with the robot, to minimize the area of search for the robot and then spreading global samples in SER. MCL, AMCL and SAMCL techniques are often used in cases where methods like Extended Kalman Filter (EKF), Unscented Kalman Filter (UKF), and Multi-Hypothesis Localization (MHL) fail [Ahmad and Namerikawa \(2013\)](#); [Zhang et al. \(2020\)](#); [Huang and Dissanayake \(2007\)](#); [Jensfelt and Kristensen \(2001\)](#); [Moreira et al. \(2020\)](#); [Panigrahi and Bisoy \(2021\)](#). PF enables not only Gaussian distributions to be handled without linearization but it can also evaluate the conditions when the dynamic system has an arbitrary probability density, which is almost the case in all real-time applications [Arulampalam et al. \(2002\)](#); [Karpinen and Vihola \(2021\)](#). Three steps are needed to form the whole PF algorithm: prediction, measurement update and resampling the steps [Roumeliotis and Bekey \(2000\)](#); [Doucet et al. \(2000\)](#); [Zhang et al. \(2019\)](#). In the prediction step, each particle is treated as a pseudo-robot. Once the robot is controlled, this step will predict the new pose of each particle using a probabilistic motion

model [Aycard and Brouard \(2020\)](#). In the measurement update step, each particle will have a weight that indicates whether this particle fits or not. This weight is determined by the likelihood of the sensor measurements at that particle's location and orientation. The resampling step means redistributing the particles according to the weight of each one. The large-weight particles are more likely to survive than small-weight particles. This step is needed because, without resampling for a few iterations, very few particles will have large weight [Li et al. \(2015\)](#). However, this step may cause the divergence and/or the inaccuracy of the filter and even algorithm failure [Meng et al. \(2021\)](#). These issues are known as particle degeneracy and particle impoverishment [Nicely and Wells \(2019\)](#).

## Related works

Resampling is an important step in the PF algorithm. It must be optimized to avoid a high delay in the localization process. For that, several resampling schemes have been proposed in the literature; Systematic Resampling (SR) [Carpenter et al. \(1999\)](#), Deterministic Systematic Resampling (DSR) [Kozierski et al. \(2013\)](#), Stratified Resampling (StR) [Kitagawa \(1996\)](#) with its variants, Low-variance Resampling (LVR) [Thrun et al. \(2005\)](#), Metropolis Resampling (MR) [Murray et al. \(2016\)](#), Rejection Resampling (RR) [Murray et al. \(2013\)](#), partial resampling, partial stratified resampling, partial deterministic resampling [Bolic et al. \(2003\)](#) and Kullback-Liebler divergence resampling [Alam and Gustafsson \(2020\)](#); [Li et al. \(2012\)](#); [Doucet et al. \(2009\)](#). After a large number of simulations, [Kozierski et al. \(2013\)](#), [Hol et al. \(2006\)](#), and [Li et al. \(2015\)](#) concluded that resampling approaches such as RR, MR, and partial resampling result in significantly poor resampling quality, unlike SR, StR, and DSR algorithms.

Systematic resampling (SR) is computationally efficient but may lead to sample correlation. To address this, deterministic systematic resampling (DSR) has been developed, which reduces sample impoverishment and enhances predictability, albeit at the cost of increased implementation complexity and potential bias. Stratified resampling (StR) aims to minimize variance and preserve a diverse set of particles. However, its effectiveness is contingent upon precise stratification, which may not adequately represent all areas of the state space. Furthermore, while the Low-variance resampling (LVR) algorithm specifically targets the reduction of sample impoverishment by eliminating particles with low weights, it risks introducing bias. In contrast, StR generally maintains better diversity among particles, but its performance significantly depends on the accurate adjustment and proportion of strata to particle weights, as indicated in prior studies [Kozierski et al. \(2013\)](#); [Li et al. \(2015\)](#).

To the best of our knowledge, no prior research has examined or explored the magnitude of the resampling effect in mobile robotics localization. This paper proposes a novel resampling algorithm for the AMCL algorithm and compares it extensively to four other algorithms, in global localization within symmetric environments. It introduces a new concept of Effective Samples Resampling referred to as ESR, which basically divides the particle set into two subsets using the number of effective particles introduced in [Kong et al. \(1994\)](#); [Martino et al. \(2016\)](#) as a threshold. The effective subset will contain the effective particles which will be sorted, and from that, the non-effective subset will contain all the remaining particles. At the end, the non-effective particles will be replaced by the *highly* effective particles held in the effective subset.

The proposed approach is tested in simulation, compared to other resampling algorithms and then tested using a real-time data collected by a real robotic system. The main contribution of this work can be summarized in the following points:

- The ESR algorithm addresses the common failures of previous resampling methods related to ambiguity and symmetry in the environment, thereby enhancing the robustness and reliability of global localization tasks.
- It effectively prevents particle degeneracy, reducing the need for numerous particles and enhancing global localization accuracy. The proposed algorithm is also able to prevent the AMCL algorithm from converging to a local minima.
- The proposed algorithm is rigorously tested in simulations, compared with existing resampling methods, and validated using real-time data from an operational robotic system, ensuring its applicability and effectiveness in practical scenarios.

The remainder of the paper is structured as follows. In the second section, we discuss the fundamentals of PF and resampling methods, as well as a brief discussion of the resampling algorithms employed in this work. The proposed ESR algorithm is introduced in the third section. In addition, simulation and experimental results are presented in the fourth section. Finally, we outline our key findings and future studies.

## Background on Particle Filter and Resampling Algorithms

Particle Filters were introduced in the early 1990s as a solution to the non-linear, non-Gaussian state estimation problem [Gordon et al. \(1993\)](#). Their flexibility and effectiveness soon made them popular across various fields, including robotics, computer vision, economics, meteorology, and signal processing.

This category of filters are a non-parametric implementation of the *Bayes filter*, employing importance sampling to model state estimates from a target distribution. This method assigns weights to samples (particles) to quantify deviations from a proposed distribution, particularly useful in complex estimation problems where analytical solutions are impractical [Hsu et al. \(2016\)](#); [Doucet et al. \(2000\)](#); [Elvira et al. \(2021\)](#). PF approximates the system's state belief as  $bel(x_{0:t}) = p(x_{0:t} | z_{1:t}, u_{1:t})$ , where  $x_{0:t}$  represents the states from time zero to  $t$ , and  $u_{1:t}$  and  $z_{1:t}$  are the control inputs and measurements, respectively.

At first, let us recall the *Bayes rule* and the *Markov assumption*. For constant probabilities  $p(y) > 0$ , *Bayes rule* relates conditionals of type  $p(x|y)$  to their “inverse”  $p(y|x)$  as follows:

$$p(x | y) = \frac{p(y | x)p(x)}{p(y)} = \frac{p(y | x)p(x)}{\int_{x'} p(y | x')p(x')dx'} \quad (1)$$

$$p(x | y, z) = \frac{p(y | x, z)p(x | z)}{p(y | z)} \quad (2)$$

*Markov assumption* assumes that the next state depends only on the current state. By applying a series of Bayes rule (1), (2) and Markov assumption to the pose belief formula (3) [Thrun et al. \(2005\)](#):

$$bel(x_{0:t}) = p(x_{0:t} | z_{1:t}, u_{1:t}) = \eta p(z_t | x_{0:t}, z_{1:t-1}, u_{1:t}) p(x_{0:t} | z_{1:t-1}, u_{1:t}) \quad (3)$$

with  $\eta = p(z_t | z_{1:t-1}, u_{1:t})$  which serves as a normalization constant.

We derive the *Bayes filter* formula [Ji et al. \(2017\)](#):

$$\begin{aligned} bel(x_{0:t}) &= \eta p(z_t | x_t) p(x_t | x_{t-1}, u_t) p(x_{0:t-1} | z_{1:t-1}, u_{1:t-1}) \\ &= \eta p(z_t | x_t) p(x_t | x_{t-1}, u_t) bel(x_{0:t-1}) \end{aligned} \quad (4)$$

The particle filter algorithm comprises three steps: prediction, measurement update, and resampling. In the prediction step, a particle  $m$  at pose  $x_{t-1}$  denoted by  $x_{t-1}^m$  is controlled the same way as the real robot at time  $t-1$ . At time  $t$  this particle will be denoted by  $x_t^{[m]}$ . All particles should be sampled from the so called probabilistic motion model, denoted by  $\widehat{bel}(x_{0:t}) = p(x_t | x_{t-1}, u_t) bel(x_{0:t-1})$ . In the measurement update step, the first term  $\eta p(z_t | x_t)$  in Equation (4) is introduced, and used to determine the fitness of each particle. Under normal conditions, one can show that the weight  $w_t^{[m]}$  of the  $m$ -th particle can be recursively computed according to the importance sampling technique given in Equation (5):

$$w_t^{[m]} = \frac{\text{target distribution}}{\text{proposal distribution}} \quad (5)$$

where the *target distribution* is  $bel(x_{0:t})$  introduced in (4), and the *proposal distribution* is  $\widehat{bel}(x_{0:t})$ :

$$w_t^{[m]} = \eta \frac{p(z_t | x_t) p(x_t | x_{t-1}, u_t) p(x_{0:t-1} | z_{1:t-1}, u_{1:t-1})}{p(x_t | x_{t-1}, u_t) p(x_{0:t-1} | z_{1:t-1}, u_{1:t-1})} \quad (6)$$

leading to:

$$w_t^{[m]} \propto p(z_t | x_t) \quad (7)$$

The resampling step aims at preventing the degeneracy of the propagated particles and redistributing them according to the target distribution  $bel(x_{0:t})$  [Shah and Kroese \(2018\)](#). This could be done by modifying the particles' set  $S_t$  distributed according to  $\widehat{bel}(x_{0:t})$  defined by Equation (8) to a new one  $\tilde{S}_t$  distributed according to  $bel(x_{0:t})$ :

$$S_t = \left\{ \left\langle x_t^{[m]}, w_t^{[m]} \right\rangle \right\}_{m=1, \dots, M} \quad (8)$$

The robustness of the PF algorithm is significantly influenced by the chosen resampling method, which can yield diverse outcomes across different algorithms [Kang et al. \(2018\)](#); [Zhou et al. \(2021\)](#). Since the resampling step implies that the propagated particles with small weight should be removed [Wang et al. \(2020\)](#), the state space exploration will be shifted to the parts of the state space with large probability masses [Li et al. \(2015\)](#).

Let us consider  $\pi(x_t)$  the weighted approximate sample-based density at time  $t$  before resampling:

$$\pi(x_t) = \sum_{j=1}^M w_j(t) \delta(x - x_j(t)) \quad (9)$$

After resampling, it will be replaced by the unweighted density  $\tilde{\pi}(x_t)$  given by Equation (10):

$$\tilde{\pi}(x_t) = \sum_{i=1}^M \frac{1}{M} \delta(x - x_i(t)) = \sum_{j=1}^M \frac{c_j}{M} \delta(x - x_j(t)) \quad (10)$$

where  $c_j$  is the number of copies of the original particle  $x_j(t)$  in  $\tilde{S}_t$ , and  $\delta$  is the Dirac delta function.

Although the resampling step mitigates the divergence issue in the particle filter, it introduces several undesirable effects. A primary concern is particle impoverishment, where particles with higher importance weights are selected multiple times, while those with lower weights are likely discarded. In global localization within symmetric environments, the resampling method significantly impacts particle filters, as particles tend to have similar weights. Additionally, resampling can lead to increased latency.

Among many resampling methods existing in literature, we mention; Stratified, Systematic, Deterministic Systematic and Low-variance resampling for their high resampling quality [Kozierski et al. \(2013\)](#); [Hol et al. \(2006\)](#); [Li et al. \(2015\)](#):

- 1 **Stratified Resampling (StR)** partitions the interval  $(0, 1]$  into  $M$  equal segments, resampling within each segment [Kitagawa \(1996\)](#).
- 2 **Systematic Resampling (SR)** uses a single random number to determine resampling intervals, improving efficiency [Carpenter et al. \(1999\)](#).
- 3 **Deterministic Systematic Resampling (DSR)** eliminates randomness by using a fixed number for interval determination [Kozierski et al. \(2013\)](#).
- 4 **Low-variance Resampling (LVR)** ensures particles are resampled with minimal variance, reducing redundancy [Thrun et al. \(2005\)](#).

Particle Filters are commonly used in robotics for state estimation and localization [Yilmaz and Bayram \(2023\)](#), often integrated with the Robot Operating System (ROS). ROS provides a flexible framework for writing robot software, which includes tools and libraries for obtaining, building, writing, and running code across multiple computers [Legleiter and Dille \(2024\)](#); [Zhu et al. \(2024\)](#).

In our validation, we will use ROS bagfiles, which are a format for recording and playing back ROS message data. Bagfiles enable the capturing of sensor data and other messages, allowing for offline real-time testing and analysis of algorithms with consistent and repeatable data inputs.

## Proposed Effective Samples Resampling (ESR)

The ESR algorithm, proposed in this paper assumes a deterministic selection of the very high-weighted at the cost of removing the extremely low-weighted samples. After sorting the weights, the number of effective sample size  $\widehat{ESS}$  (also known as  $N_{\text{eff},t}$ ) is used as a threshold [Kong et al. \(1994\)](#):

$$\sum_{m=1}^{M-N_{\text{eff},t}} p_r^{[m]}(x_t) = \sum_{m=N_{\text{eff},t}+1}^M p_s^{[m]}(x_t) \quad (11)$$

where  $p_r^{[m]}(x_t)$  and  $p_s^{[m]}(x_t)$  denote the probability of removing and selecting  $x_t^{[m]}$  respectively.

The ESR algorithm uses a weights' indices manipulation technique to guarantee the completeness of

Equation (11). It takes as inputs the particles set  $S_t$  and their weights  $W_t$  and from that a set  $\Omega_t$  of weights with their indices is then constructed as follows:

$$\Omega_t = \left\{ \left\langle W_t^{[m]}, m \right\rangle \right\}_{m=1, \dots, M} \quad (12)$$

Afterward, the set  $\Omega_t$  is ascendingly sorted with respect to the weight value. Therefore, each weight will retain its previous position within the unsorted weights set. It is worth noting that ESR algorithm is dealing with two sets instead of just one. The first set is the particle set  $S_t$ , the second is the weights set  $\Omega_t$  which provides a kind of memory to the PF.

This gives each particle the possibility to recall its position in  $S_t$ . According to the obtained position, the corresponding particle is going to be replicated, deleted or just left to be propagated in the following iterations.

The expected number of times  $M_t^{[m]}$  that the particle  $x_t^{[m]}$  is often resampled proportionally to its weight  $w_t^{[m]}$  as defined by the Equation of unbiasedness (13) Liu and Chen (1998):

$$E(M_t^{[m]} | w_t^{[m]}) \propto M w_t^{[m]} \quad (13)$$

It can be clearly seen that Equation (11) is directly related to  $N_{\text{eff},t}$ , where  $N_{\text{eff},t} = \alpha M$  with  $\alpha \in [0, 1]$ . Thus the ESR algorithm is classified as a compound sequential resampling technique Li et al. (2015) as described in Equation (14):

$$M_t^{[m]} \in \left[ 0, \frac{1}{\alpha} \right] \quad (14)$$

$N_{\text{eff},t}$  is often used for analyzing large datasets Acosta et al. (2021) or to reduce the frequency of resampling Martino et al. (2016, 2017). Such that resampling takes place whenever  $N_{\text{eff},t}$  is beyond a certain threshold  $\gamma$ .

For example, if one chooses  $\gamma = 0.5M$ , this means that resampling will occur only if the number of ineffective particles exceeds half of the total number of particles:

$$\begin{cases} N_{\text{eff},t} \leq \gamma, & \text{resample} \\ N_{\text{eff},t} > \gamma, & \text{do not resample} \end{cases} \quad (15)$$

However, the ESR algorithm will lean on this number, to split-up the set  $\Omega_t$  into effective and ineffective particles, in order to replace only the ineffective by the *highly* effective particles.

The algorithm starts the particles insertion in the unweighted set  $\tilde{S}_t$ , from the last weight's particle in  $\Omega_t$ . Then, the  $M - N_{\text{eff},t}$  highest weights' particles in  $\Omega_t$  will be replicated one more time to finally fill up  $\tilde{S}_t$  with there corresponding weights' particles. Algorithm 1 summarizes the different steps of the proposed ESR algorithm.

It is worth noting that steps 2-4 in Algorithm 1 could be implemented in the measurement update step of the PF algorithm.

### Analysis of the ESR Algorithm

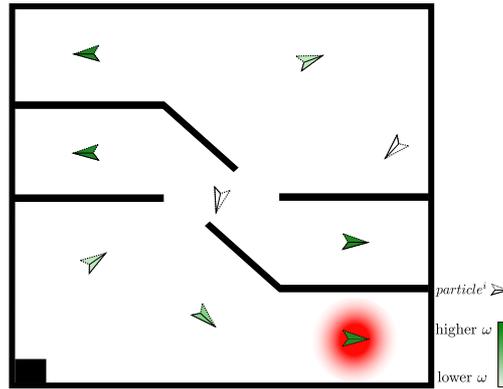
Figure 1 effectively illustrates the challenges associated with the AMCL algorithm, which utilizes particle filtering for robot global localization. In this representation, particles are depicted as small arrows, with

**Algorithm 1** ESR**Require:**  $S_t, W_t, N_{\text{eff},t}$ 

```

1:  $\tilde{S}_t = \Omega_t = \emptyset$ 
2: for  $m = 1$  to  $M$  do
3:   add  $\langle W_t^{(m)}, m \rangle$  to  $\Omega_t$ 
4: end for
5: sort  $(\Omega_t)$  with respect to  $W_t$ 
6:  $ii = 1, jj = 0$ 
7: while  $ii \leq M$  do
8:   add  $x_t^{[\Omega_t^{[M-jj]}.m]}$  to  $\tilde{S}_t$ 
9:    $ii = ii + 1, jj = jj + 1$ 
10:  if  $jj > N_{\text{eff},t}$  then
11:     $jj = 0$ 
12:  end if
13: end while

```



**Figure 1.** Visualization of particle weights and their corresponding potential poses using the AMCL algorithm for global localization.

their potential poses and weights indicated by varying shades of green (the darker the green, the higher the weight). The true robot position is marked by a shaded red circle in the bottom right corner.

A key observation from Figure 1 is the potential vulnerability of *probabilistic* resampling to erroneously remove significant particles, including those that accurately represent the robot's true position. Theoretically, all of the equally dark-green particles must be preserved. For instance, the removal of the particle near the true position could hinder the algorithm's ability to converge to the correct location. This scenario underscores the inherent risk of sample impoverishment, where good particles are randomly discarded, leading to possible localization failures. This vulnerability is particularly pronounced in symmetric environments, where probabilistic resampling might exclude the essential

particle that represents the true position of the robot, potentially compromising the entire particle set's accuracy.

To address this, the Effective Samples Resampling (ESR) algorithm 1 is designed to:

- deterministically eliminate only particles with *very low* weights, thereby preserving those with higher probabilities and improving the robustness of the localization process.
- replicate the  $M - N_{\text{eff},t}$  high-weighted particles after sorting the particles set  $\Omega_t$  with respect to the weights  $W_t$ .

Therefore, by implementing this resampling algorithm, the likelihood of retaining significant particles increases, which in turn enhances the accuracy and reliability of the AMCL algorithm.

## Simulation and Experimental results

The algorithms described in section previously as well as the proposed ESR algorithm were simulated, tested and compared to each other using ROS melodic installed on an 8GB of RAM and i5 CPU PC running under Ubuntu 18.04 operational system. The mobile robot used for simulation is called Turtlebot2 shown in Figure 2a. It is a low-cost differential-drive mobile robot, composed of a Yujin Kobuki mobile platform, sensors and actuators, such as optical encoders and DC motors. Additionally, It integrates a Microsoft XBOX 360 Kinect sensor. For real-time experiments, a ROS bag file of the PR2 (mobile manipulation platform represented in Figure 2c) recorded by Willow Garage is employed. The robot is equipped with various types of sensors such as a three-axis accelerometer, Microsoft Kinect and Hokuyo UTM-30LX Lidar sensors. In addition, it incorporates 32 brushed DC motors for controlling its different joints and wheels. The PR2 software system is fully developed in ROS. As a result, all PR2 features are accessible through ROS interfaces Bohren et al. (2011). The robot's ROS package used for simulation combines the Kobuki platform with the Kinect ROS packages. This combination allows simulation and implementation of different mobile robotics tasks such as perception, localization, mapping, navigation and control using ROS navigation stack.

The simulation environment is called Turtlebot2 stage, it is a 2D maze like environment. It represents a couple of symmetric rooms as illustrated in Figure 2b. Furthermore, simulation and real-time experiments uses the AMCL ROS package for localization. All the AMCL parameters used for tracking and/or global localization are summarized in Table 1.

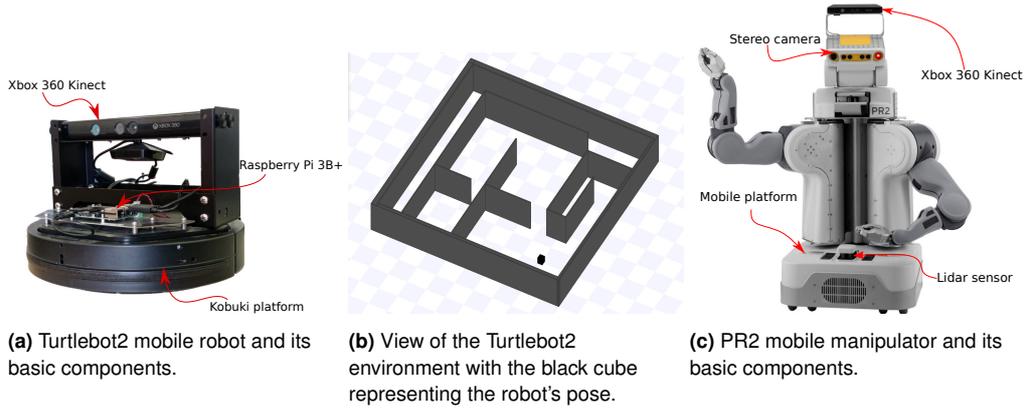
### Simulation results

In this section, Turtlebot2 mobile robot is controlled via a Lyapunov-based control scheme described by Equation (16), in order to control both linear and angular velocities  $v$  and  $\omega$  to arrive at a desired polar coordinates  $x_D = [\rho, \alpha, \beta]^T$  Park and Kuipers (2011); Siegwart et al. (2011). This scheme is based on the resulted pose estimate from the AMCL algorithm and  $x_D$ .

$$\begin{cases} v = k_v \rho \cos \alpha \\ \omega = k_v \sin \alpha \cos \alpha + k_\alpha \alpha \end{cases} \quad (16)$$

where  $k_v$  and  $k_\alpha$  are positive constants.

The AMCL ROS package uses an adaptive number of particles varying from 5000 to 500 for position



**Figure 2.** View of all basic components of Turtlebot2 and PR2 (Personal Robot 2).

**Table 1.** A list of all the parameters of the AMCL ROS node. This adjustment is done to assess the ESR algorithm's efficiency for global localization without modifying the number of filter updates before resampling [Fox \(2001\)](#).

Parameter	Value
Probabilistic Motion Model type	Differential Motion Model
Probabilistic Sensor Model	Likelihood Field Model
Motion Model Parameters	$\alpha = [0.2, 0.2, 0.3, 0.2, 0.1]^T$
$min_{particles}$ (Minimum allowed number of particles)	Depends on the scenario
$max_{particles}$ (Maximum allowed number of particles)	1
$resample\_interval$ (Number of filter updates required before resampling)	0.05
$KLD_{error}$ (Max error between $bel(x_{0:t})$ and $\pi(x_t)$ )	0.99
$KLD_z$ (Upper standard normal quantile for $(1-p)$ )	30
$max_{beams}$ (How many beams will be used)	0.5
$z_{hit}$ (Mixture weight for the $z_{hit}$ part of $p(z_t   x_t)$ )	0.05
$z_{short}$ (Mixture weight for the $z_{short}$ part of $p(z_t   x_t)$ )	0.5
$z_{max}$ (Mixture weight for the $z_{max}$ part of $p(z_t   x_t)$ )	0.5
$z_{rand}$ (Mixture weight for the $z_{rand}$ part of $p(z_t   x_t)$ )	0.5
$\sigma_{hit}$ (Standard deviation for Gaussian model used in $z_{hit}$ part of $p(z_t   x_t)$ )	0.2
$\lambda_{short}$ (Exponential decay parameter for $z_{short}$ part of $p(z_t   x_t)$ )	0.1
$[\alpha_{slow}, \alpha_{fast}]$ (Exponential decay rate for the slow and fast average weight)	[0.0, 0.0]
$update\_distance$ (Translational movement required before performing a filter update)	0.2 m
$update\_angle$ (Rotational movement required before performing a filter update)	0.5 rad

tracking and from 50000 to 500 for global localization. In order to compare the performances of the above described resampling algorithms with the ESR algorithm, the ground truth and the estimated trajectories are plotted. As well as the Root Squared Error ( $RSE$ ) regarding their  $x$  and  $y$  locations.

Throughout this section, the terms GT, ET and RT will refer to the ground truth, estimated and reference trajectories respectively.

$$RSE = \sqrt{(x - x_g)^2 + (y - y_g)^2} \quad (17)$$

where  $x_g$  and  $y_g$  are the robot's ground truth positions. The Python ROS node that controls the robot via Equation (16) is driving the robot along a pre-defined point-to-point trajectory  $x_{ref}$ .

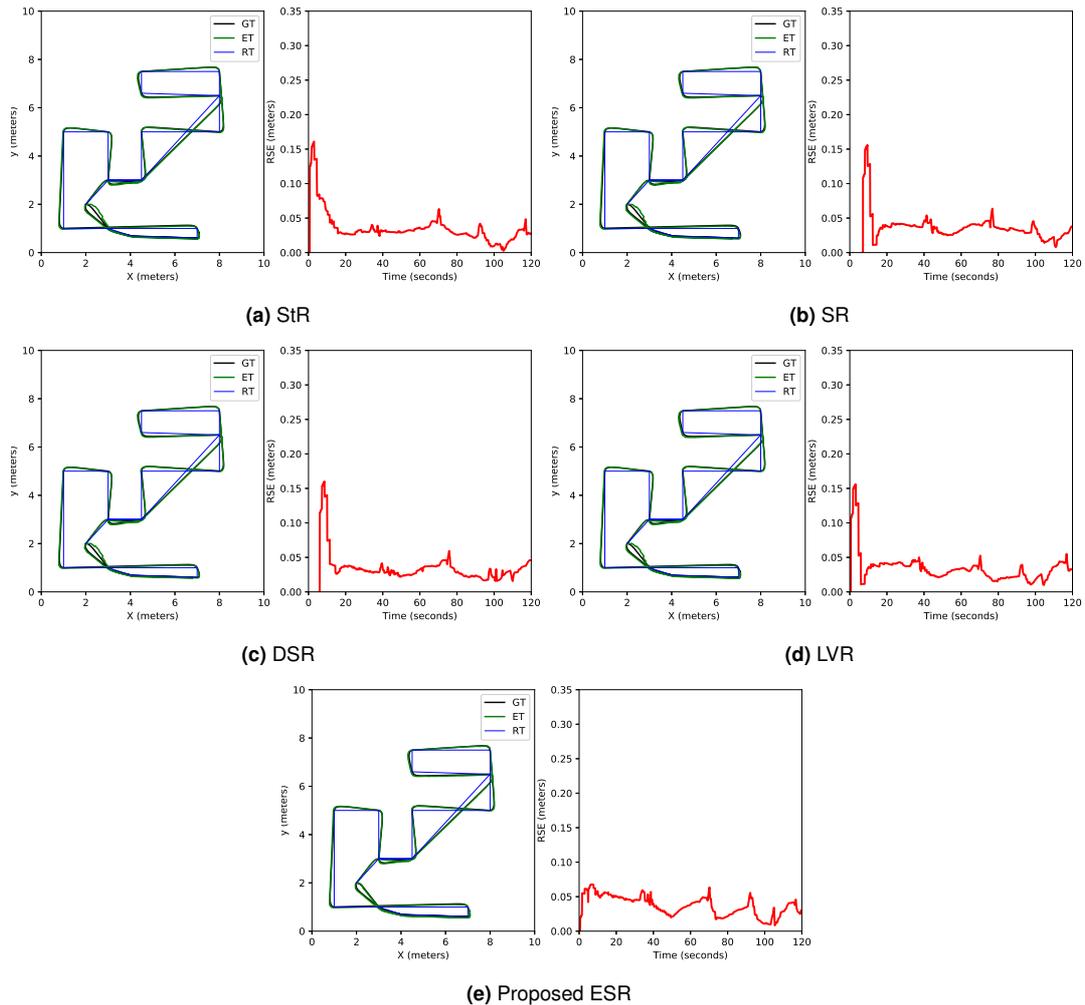
The reason behind choosing this trajectory is to augment the localization difficulty, considering its length and symmetry redundancy.

*Simulation Scenario 1: Pose Tracking* In this first scenario, the initial pose of the robot is assumed to be known. The AMCL algorithm tracks the robot along  $x_{ref}$ . The robot is initially located at  $x_0 = [2.0, 2.0, 0.0]^T$ , and the number of particles is varying between 5000 and 500 particles. Figures 3 describe how the  $RSE$  and ET is evolving in simulation time for each of the resampling schemes. Figure 3a describes how the error is changing when using Stratified resampling, while Figure 3b illustrates the same scenario using Systematic resampling. Deterministic Systematic resampling simulation's results are illustrated in Figure 3c. Figure 3d shows how the tracking process is handled using Low-variance resampling. Figure 3e illustrates the evolution of the error between ET and GT when the Effective Samples Resampling (ESR) algorithm is involved. It can be seen that there is no significant difference between them while tracking the robot's pose. Moreover, all of these algorithms monitor the robot with a certain offset without faults.

*Simulation Scenario 2: Global localization* In this scenario, the robot's initial pose is no longer known. Therefore, global localization will be aimed while simulation results will be illustrated through Figures 4. All of the resampling algorithms were used in the AMCL ROS node with a particles size  $M \in [500, 50000]$ . These figures describe how the  $RSE$  and the estimated trajectory are evolving versus simulation time for each of the resampling schemes. Figure 4a describes how the error is varying when the resampling is handled by Stratified resampling algorithm. Figure 4b illustrates the same scenario but using Systematic resampling. Deterministic Systematic resampling simulation's results are illustrated in Fig4c. Figure 4d shows how the AMCL ROS node is trying to globally localize the robot using Low-variance resampling, while Figure 4e illustrates the evolution of the error between the ground truth  $x_g$  and  $y_g$  coordinates and the estimated one of the robot using the proposed Effective Samples Resampling (ESR) algorithm. Furthermore, we can see how efficient the suggested approach is. It can also be observed that, with the exception of the proposed ESR technique, none of the resampling algorithms were able to estimate the pose of the robot. It not only guaranteed convergence, but it also converged in a short period of time. AMCL algorithm is still one of the best probabilistic mobile robot global localization algorithms. However, in an ideal map, such as the one shown in Figure 2b, with a lot of symmetry. The existence of numerous particles with almost equal weights could be a problem. In most cases, all particles with equal weights should be selected. When a random resampling algorithm attempts to resample well-fitting particles from evenly weighted group of particles. It is possible that it will alleviate one or more of these particles. One of the particles that was deleted, on the other hand, may represent the mobile robot's true state. Effective Samples Resampling (ESR) algorithm is able to split up the particles set  $S_t$  into effective and ineffective sets, before replacing the *extremely ineffective* with the *extremely effective* particles. Consequently, all well-fitting particles are saved.

To further showcase the impact of the number of particles adaptation on the resampling quality and convergence, we conduct two other global localization simulations in another symmetric environment. Figure 5b illustrates the environment, while Figure 5a provides its corresponding 2D grid map along with the reference trajectory depicted in an orange line.

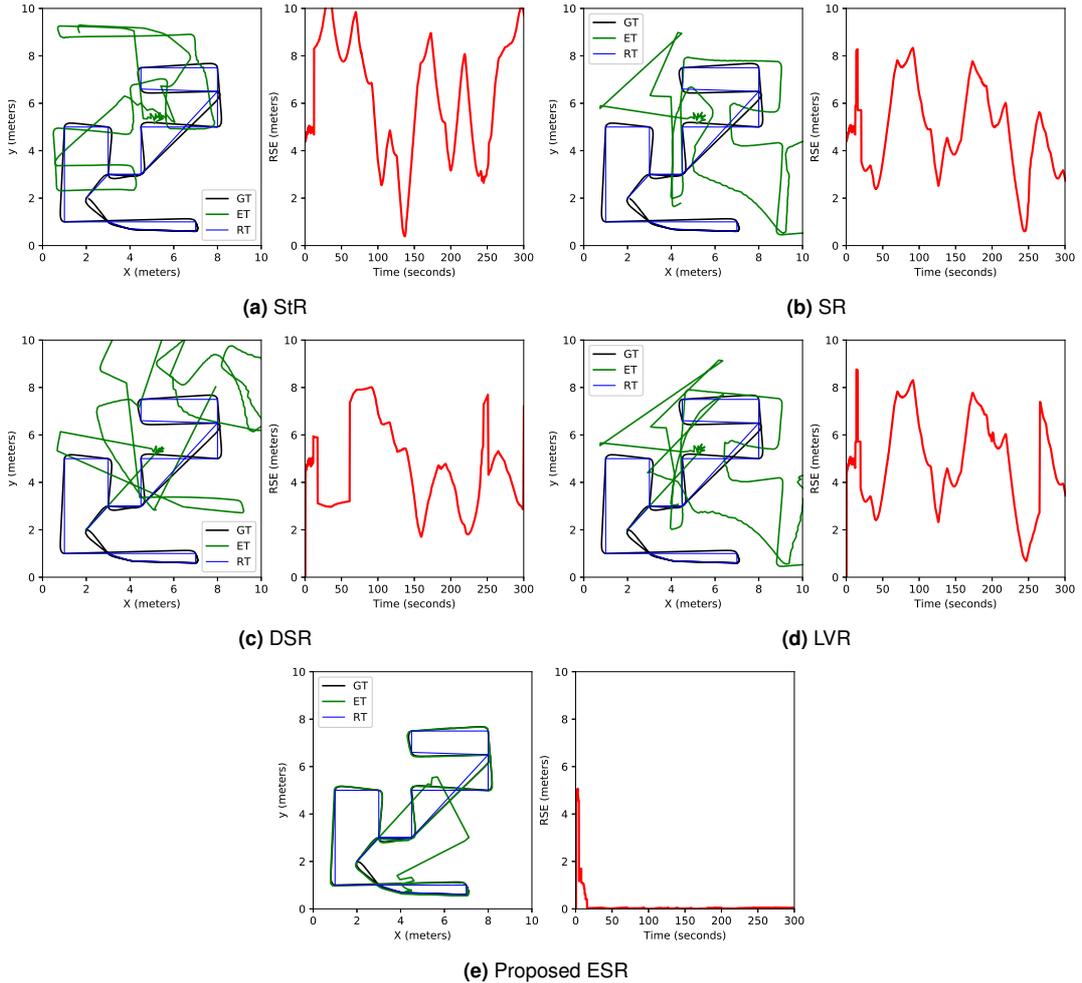
In the first simulation case, the number of particles varies between a minimum of 500 and a maximum of 70000. This test highlights the robustness of the Effective Samples Resampling (ESR) algorithm,



**Figure 3.** Illustrating the tracking error evolution for all algorithms, plotting the ground truth and AMCL estimated trajectories, and showcasing the error (red curve) between them.

where it not only converges faster but also achieves a lower Root Mean Squared Error (RMSE) compared to other algorithms. Figure 6 illustrates the tracking errors and RMSE for each algorithm.

In the second simulation case, we fix the number of particles at 70000 for all algorithms to assess their performance under the same conditions. All algorithms demonstrate convergence, yet the ESR algorithm continues to exhibit superior performance in terms of convergence speed and RMSE. Figure 7 depicts these results, emphasizing the comparative efficiency of our resampling algorithm.

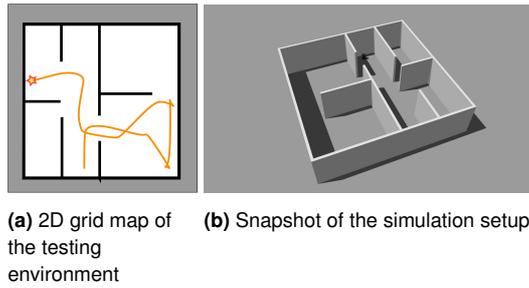


**Figure 4.** Error evolution of all algorithms aiming at global localization. The figure depicts ground truth and AMCL estimated trajectories, as well as the error (red curve) between them.

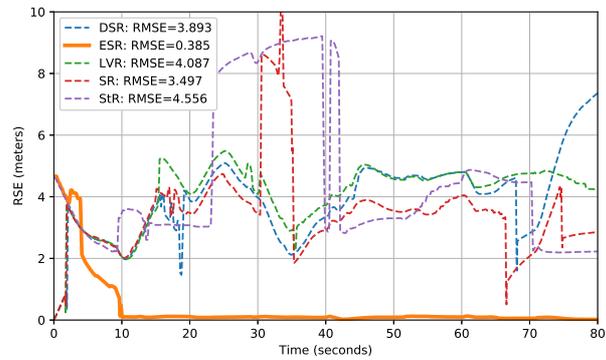
### Real-time experimental results

In this section, a  $58.25\text{ m} \times 47.25\text{ m}$  occupancy grid map of the Willow environment illustrated in Figure 8, and a ROS bag file collected by Willow PR2 looping inside the Willow environment were used, in order to test the AMCL ROS package while implementing different resampling techniques.

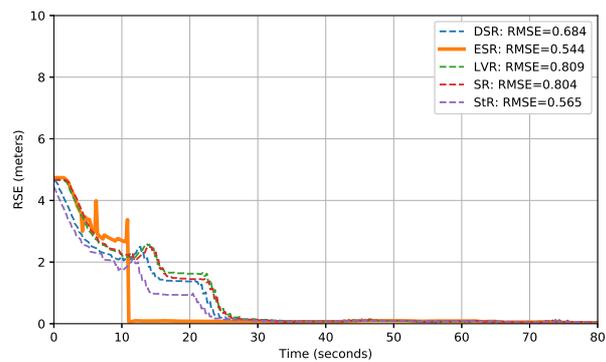
The advantage of using this map and a bag file, lies in allowing the validation of different resampling schemes under the same conditions and using the same real-time experimental data, which results in a fair comparison. The process of localization is running on the same control PC described above using ROS.



**Figure 5.** Illustrative views of the symmetric testing environment used in simulations.



**Figure 6.** Tracking errors and RMSE with adaptive particle size.



**Figure 7.** Tracking errors and RMSE with fixed particle size.



**Figure 8.** An illustrative view of the Willow environment 58.25 m x 47.25 m occupancy grid map, with 0.05 m/pixel resolution.

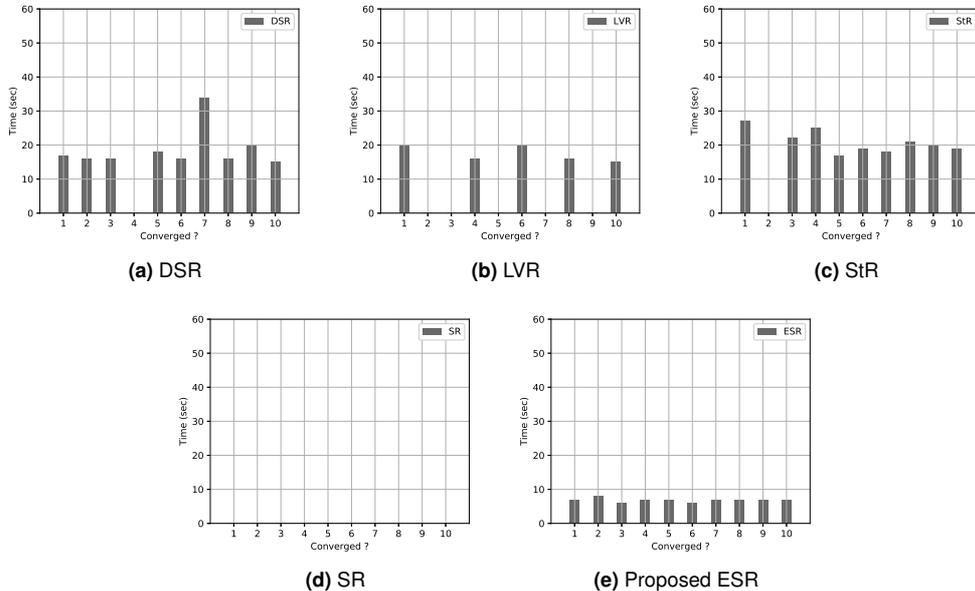
The reason behind choosing the Willow dataset to test the algorithm lies in the symmetry redundancy in the map and the high quality of the Lidar sensor. Whereas the purpose of choosing a Lidar sensor with high-quality measurements is to maximize the number of particles sensing almost the same thing when seeing corners, walls, and other symmetric geometries Cappé et al. (2007); Gustafsson et al. (2002). As a result, increasing the likelihood that a resampling strategy will delete an excellent particle. It can be clearly seen from the numerical simulation results section that the global localization in a simulation environment (characterized with a true symmetry compared to the real world) often fails. Accordingly, to increase the challenge of global localization one may increase the number of particles leading to an increase in the computational requirements as well as the chance of the global localization process to fail. Several runs of the PR2 bag file were conducted for every resampling algorithm. By changing the number of particles parameter of the AMCL ROS package, we can measure the impact of the resampling phase on the global localization from one resampling technique to another. We propose to conduct two scenarios for each resampling method, where both first and second scenarios run ten times, for each test. The time needed by the AMCL ROS node to correctly converge to the robot's location is saved for every test, along with a Boolean value set to *true* every time the AMCL algorithm converges to the correct pose and to *false* if not.

To provide a clearer comparison of the tracking error performance of the algorithms, we have selected the trials with the smallest Root Mean Square Error (RMSE) for each algorithm. This approach is necessary because some algorithms may fail to converge in certain trials, complicating direct performance comparisons across all trials. By focusing on the trials with the least RMSE, we ensure that the comparison is based on the most successful execution of each algorithm, thereby providing a more accurate representation of their tracking accuracy. This method highlights the potential of each algorithm when it performs at its best, allowing for a fair and meaningful comparison.

*Experimental Scenario 1* The mobile robot moves along a trajectory defined in the bag file, and the AMCL package runs to globally localize it. The allowed maximum number of particles is set to 70000 and the minimum is to 500 particles.

**Table 2.** Scenario 1: AMCL convergence time (in seconds) for global localization using all algorithms across ten trials.

Method	Convergence	Convergence time (sec)
StR	1,0,1,1,1,1,1,1,1,1	27,-,22,25,17,19,18,21,20,19
SR	0,0,0,0,0,0,0,0,0,0	-,-,-,-,-,-,-,-,-,-
DSR	1,1,1,0,1,1,1,1,1,1	17,16,16,-,18,16,34,16,20,15
LVR	1,0,0,1,0,1,0,1,0,1	20,-,-,16,-,20,-,16,-,15
ESR	1,1,1,1,1,1,1,1,1,1	7,8,6,7,7,6,7,7,7,7



**Figure 9.** Scenario 1: AMCL algorithm's pose estimation time, using all algorithms. Bar height represents time in seconds. Absence of the bar indicates localization failure.

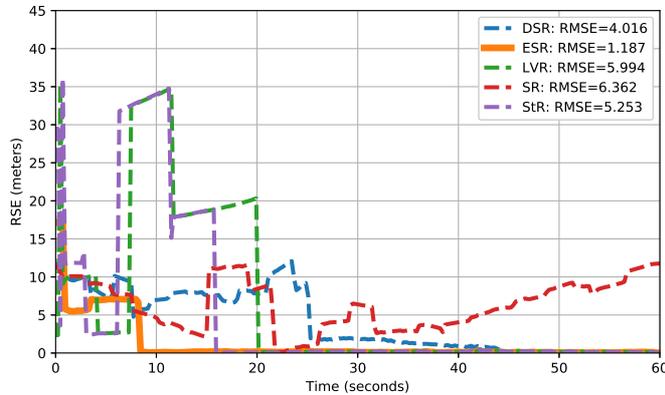
The results of the first scenario are given in Table 2. In Figures 9 each bar is happened to be in a specified time step and it is characterized by its height. The presence of the bar means that the localization process succeed at that time step and its height is indicating the convergence time in seconds. It is clearly remarkable that the SR algorithm was not able to guarantee the global localization as depicted in Figure 9d. The StR algorithm converges in almost 90% of the experiments with an average time about 20.8 sec as shown in Figure 9c.

In Figure 9a, the DSR algorithm is involved. It converges pretty much like the StR algorithm but a little bit quicker with an average time 18.6 sec. The LVR algorithm was able to converge in almost half of the experiments as illustrated in Figure 9b.

The proposed ESR algorithm offers the best performances. Figure 9e shows how it guaranteed the

convergence of the AMCL algorithm every time with the same setup. Furthermore, it is significantly quicker and more stable than the other algorithms, with an average time of around 6.9 sec.

Figure 10 illustrates the Root Squared Error (RSE) of each algorithm. The figure provides a clear, quantitative comparison of their performance in terms of accuracy and time convergence time. The plot includes both instantaneous RSE values over time and the Root Mean Squared Error (RMSE) for each method, encapsulating their overall error metrics throughout the first scenario.



**Figure 10.** Root Squared Error (RSE) and Root Mean Squared Error (RMSE) comparisons of DSR, SR, LVR, StR, and the proposed ESR algorithm in the first experimental scenario.

The figure illustrates the RMSE values for each algorithm, measured in meters, which serve as indicators of the effectiveness of each algorithm in minimizing global localization errors in the first real-time scenario:

Algorithm	RMSE (meters)	Convergence Time (secs)
DSR	4.016	43.78
LVR	5.994	20.11
SR	6.362	Did not converge
StR	5.253	15.76
Proposed ESR	<b>1.187</b>	<b>7.88</b>

**Table 3.** Performance comparison of different algorithms in real-time scenario 1

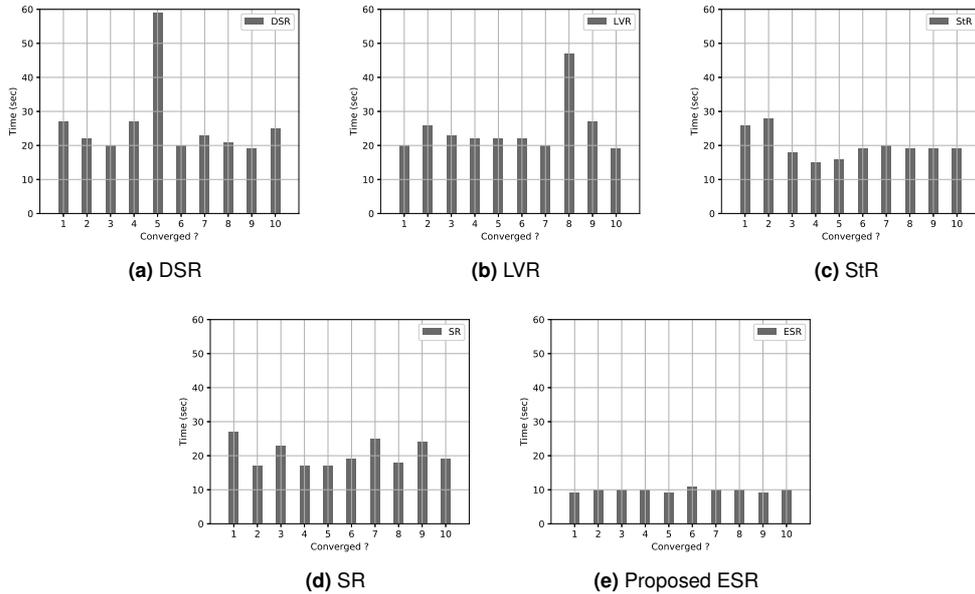
These findings demonstrate the superior performance of the ESR algorithm in reducing error accumulation and efficiently converging compared to the existing resampling algorithms in global localization. Table 3, describes the impact of the particles set size on the resampling algorithms’ convergence.

*Experimental Scenario 2* Like in the previous scenario, the robot moves along the same trajectory and the global localization is handled by the AMCL ROS package. However, this time the number of particles is fixed to 70000, so it is no longer adapted. Additionally, illustrations of how resampling algorithms can

be sensitive to the particles' number and adaptation can be seen through Figures 11. Furthermore, the obtained results for this scenario are presented in Table 4.

**Table 4.** Scenario 2: AMCL convergence time (in seconds) for global localization using all algorithms across ten trials.

Method	Convergence	Convergence time (sec)
StR	1,1,1,1,1,1,1,1,1,1	26,28,18,15,16,19,20,19,19,19
SR	1,1,1,1,1,1,1,1,1,1	27,17,23,17,17,19,25,18,24,19
DSR	1,1,1,1,1,1,1,1,1,1	27,22,20,27,59,20,23,21,19,25
LVR	1,1,1,1,1,1,1,1,1,1	20,26,23,22,22,22,20,47,27,19
ESR	1,1,1,1,1,1,1,1,1,1	9,10,10,10,9,11,10,10,9,10

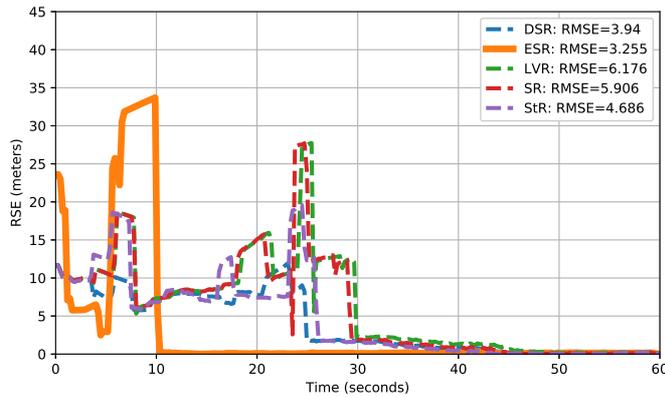


**Figure 11.** Scenario 1: AMCL algorithm's pose estimation time, using all algorithms. Bar height represents time in seconds. Absence of the bar indicates localization failure.

It can be noticed that all of the resampling algorithms have well handled the global localization. However, the time taken by the AMCL algorithm to find where the robot is in the map sometimes is too long. It is remarkable that unlike in the previous case where the number of particles is changing within the interval [70000, 500], the SR algorithm now guarantees the global localization always with an average time of 20.6 sec as can be seen in Figure 11d. The StR algorithm converges every time as well, with an average time about 19.9 sec as represented in Figure 11c. In the case of the DSR algorithm, global localization is perfectly handled, taking 26.3 sec in average to converge as depicted in Figure 11a. The LVR algorithm converges in all of the experiments as illustrated in Figure 11b with an average

time of 24.8 *sec*. The Effective Samples Resampling (ESR) algorithm provides the best performances in terms of stability and computational time compared to the others as illustrated in Figure 11e. It is stable and fast with an average convergence time of about 9.8 *sec*. In the case of StR algorithm there is a peak that exhibiting the worst convergence time. This latter is due to the randomness present in the process of resampling, as opposed to the proposed ESR algorithm.

Figure 12 illustrates the Root Squared Error (RSE) of each algorithm in the second scenario, where each algorithm was tested with a fixed number of 70,000 particles. The figure provides a clear, quantitative comparison of their performance in terms of accuracy and convergence time. The plot includes both instantaneous RSE values over time and the Root Mean Squared Error (RMSE) for each method, encapsulating their overall error metrics throughout the scenario.



**Figure 12.** Root Squared Error (RSE) and Root Mean Squared Error (RMSE) comparisons of DSR, SR, LVR, StR, and the proposed ESR algorithm in the second experimental scenario.

The figure depicts the RMSE values for each algorithm, measured in meters, which serve as indicators of the effectiveness of each algorithm in minimizing global localization errors in the second real-time scenario. As it can be observed from Table 5, fixing the number of particles affects the convergence of the resampling algorithm.

Algorithm	RMSE (meters)	Convergence Time (secs)
DSR	3.94	42.31
LVR	6.176	46.50
SR	5.906	44.20
StR	4.686	40.15
Proposed ESR	<b>3.255</b>	<b>10.1</b>

**Table 5.** Performance comparison of different algorithms in real-time scenario 2

These findings demonstrate the superior performance of the ESR algorithm in terms of error reduction, efficiency in convergence, and overall robustness compared to the other resampling algorithms in global localization under fixed particle count conditions.

Overall, using existing resampling algorithms requires careful tuning of the number of particles. The proposed ESR algorithm not only addresses this issue but also outperforms all other resampling algorithms in terms of effective and robust global localization in symmetric environments.

## Conclusion

Overconfident resampling comes from the existence of many particles weighted with almost the same value. Ideally, all equally weighted particles should be selected. Whenever a random resampling algorithm is attempting to resample good fit particles, from equally weighted ones. It actually might remove one or more of these particles. Whereas, one of the removed particles could be the actual pose of the mobile robot. In this paper we presented a non-theoretical comparison between a proposed resampling algorithm namely Effective Samples Resampling (ESR) and some of the typical algorithms, for the resampling phase of the Adaptive Monte Carlo Localization (AMCL) algorithm. Yet, it demonstrates the effectiveness of the proposed algorithm. Both simulation and real-time implementations were successfully conducted.

The resampling phase of the AMCL is critical, and it needs to be well implemented. Some resampling methods frequently fail for global localization when the map of the environment has a lot of symmetry, in contrast to the proposed ESR algorithm, which offers excellent performance and has the shortest convergence time. The efficiency of this algorithm is attributed to the high probability of resampling the *very* effective particles at the expense of the *highly* ineffective ones, due to its determinability.

The implementational overhead of our approach is very small. However, it includes a sorting procedure that can be a little bit expensive. Typically, ESR algorithm's computational complexity is highly dependent on the involved sorting algorithm. The global localization in very large environments that requires a substantial particle set size (e.g  $M \geq 10^7$ ), could cause a high delay. This can be resolved by implementing a faster sorting algorithm or exploiting an FPGA (Field Programmable Gate Array) to accelerate sorting or resampling.

Our approach has primarily been applied to mobile robot localization in symmetric environments, demonstrating the effectiveness of our newly developed resampling algorithm for particle filters. Beyond this, the versatility of the technique allows it to be beneficial in other areas as well. Notably, it can significantly enhance Simultaneous Localization And Mapping (SLAM) and improve state-space estimation across various control systems. This broad applicability highlights the potential of our algorithm to contribute to advancements in diverse fields within robotics and control engineering.

## References

- Acosta J, Alegría A, Osorio F and Vallejos R (2021) Assessing the effective sample size for large spatial datasets: A block likelihood approach. *Computational Statistics & Data Analysis* 162: 107282.
- Ahmad H and Namerikawa T (2013) Extended kalman filter-based mobile robot localization with intermittent measurements. *Systems Science & Control Engineering* 1(1): 113–126.
- Alam SA and Gustafsson O (2020) Improved particle filter resampling architectures. *Journal of Signal Processing Systems* 92(6): 555–568.

- Arulampalam MS, Maskell S, Gordon N and Clapp T (2002) A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing* 50(2): 174–188.
- Aycard O and Brouard C (2020) A new tool to initialize global localization for a mobile robot. In: *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, pp. 1296–1303.
- Bohren J, Rusu RB, Jones EG, Marder-Eppstein E, Pantofaru C, Wise M, Mösenlechner L, Meeussen W and Holzer S (2011) Towards autonomous robotic butlers: Lessons learned with the pr2. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 5568–5575.
- Bolic M, Djuric PM and Hong S (2003) New resampling algorithms for particle filters. In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)*., volume 2. IEEE, pp. II–589.
- Cappé O, Godsill SJ and Moulines E (2007) An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE* 95(5): 899–924.
- Carpenter J, Clifford P and Fearnhead P (1999) Improved particle filter for nonlinear problems. *IEE Proceedings-Radar, Sonar and Navigation* 146(1): 2–7.
- Dellaert F, Fox D, Burgard W and Thrun S (1999) Monte carlo localization for mobile robots. In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 2. IEEE, pp. 1322–1328.
- Desrochers B, Lacroix S and Jaulin L (2015) Set-membership approach to the kidnapped robot problem. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3715–3720.
- Doucet A, Godsill S and Andrieu C (2000) On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing* 10(3): 197–208.
- Doucet A, Johansen AM et al. (2009) A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering* 12(656-704): 3.
- Elvira V, Miguez J and Djurić PM (2021) On the performance of particle filters with adaptive number of particles. *Statistics and Computing* 31(6): 1–18.
- Forbes JR (2013) Adaptive approaches to nonlinear state estimation for mobile robot localization: an experimental comparison. *Transactions of the Institute of Measurement and Control* 35(8): 971–985.
- Fox D (2001) Kld-sampling: Adaptive particle filters and mobile robot localization. *Advances in Neural Information Processing Systems (NIPS)* 14(1): 26–32.
- Fox V, Hightower J, Liao L, Schulz D and Borriello G (2003) Bayesian filtering for location estimation. *IEEE pervasive computing* 2(3): 24–33.
- Gordon NJ, Salmond DJ and Smith AF (1993) Novel approach to nonlinear/non-gaussian bayesian state estimation. In: *IEE Proceedings F-radar and signal processing*, volume 140. IET, pp. 107–113.
- Guan RP, Ristic B, Wang L and Palmer JL (2019) Kld sampling with gmapping proposal for monte carlo localization of mobile robots. *Information Fusion* 49: 79–88.
- Gustafsson F, Gunnarsson F, Bergman N, Forssell U, Jansson J, Karlsson R and Nordlund PJ (2002) Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing* 50(2): 425–437.
- Hol JD, Schon TB and Gustafsson F (2006) On resampling algorithms for particle filters. In: *2006 IEEE nonlinear statistical signal processing workshop*. IEEE, pp. 79–82.
- Hsu CC, Yeh SS and Hsu PL (2016) Particle filter design for mobile robot localization based on received signal strength indicator. *Transactions of the Institute of Measurement and Control* 38(11): 1311–1319.

- Huang S and Dissanayake G (2007) Convergence and consistency analysis for extended kalman filter based slam. *IEEE Transactions on robotics* 23(5): 1036–1049.
- Jensfelt P and Kristensen S (2001) Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation* 17(5): 748–760.
- Ji G, Wang Y, Zhao S, Liu Y, Zhang K, Yao B and Zhou S (2017) Bayesian hybrid state estimation for unequal-length batch processes with incomplete observations. *International Journal of Control, Automation and Systems* 15(6): 2480–2491.
- Julius Fusic S, Hariharan K, Sitharthan R and Karthikeyan S (2022) Scene terrain classification for autonomous vehicle navigation based on semantic segmentation method. *Transactions of the Institute of Measurement and Control* 44(13): 2574–2587.
- Kang H, Lee HS, Kwon YB and Park YH (2018) A sequential estimation algorithm of particle filters by combination of multiple independent features in evidence. *International Journal of Control, Automation and Systems* 16(3): 1263–1270.
- Karppinen S and Vihola M (2021) Conditional particle filters with diffuse initial distributions. *Statistics and computing* 31(3): 1–14.
- Kitagawa G (1996) Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of computational and graphical statistics* 5(1): 1–25.
- Kong A, Liu JS and Wong WH (1994) Sequential imputations and bayesian missing data problems. *Journal of the American statistical association* 89(425): 278–288.
- Kozierski P, Lis M and Zietkiewicz J (2013) Resampling in particle filtering - comparison. *Studia z Automatyki i Informatyki* 38: 35–64.
- Kuptamete C and Aunsri N (2022) A review of resampling techniques in particle filtering framework. *Measurement* : 110836.
- Laaksonen J and Kyrki V (2008) Localization in ambiguous environments using multiple weak cues. *Intelligent Service Robotics* 1(4): 281–288.
- Legleiter CJ and Dille M (2024) A robot operating system (ros) package for mapping flow fields in rivers via particle image velocimetry (piv). *SoftwareX* 26: 101711. DOI:<https://doi.org/10.1016/j.softx.2024.101711>. URL <https://www.sciencedirect.com/science/article/pii/S2352711024000827>.
- Li T, Bolic M and Djuric PM (2015) Resampling methods for particle filtering: classification, implementation, and strategies. *IEEE Signal processing magazine* 32(3): 70–86.
- Li T, Sattar TP and Sun S (2012) Deterministic resampling: unbiased sampling to avoid sample impoverishment in particle filters. *Signal Processing* 92(7): 1637–1645.
- Liu JS and Chen R (1998) Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association* 93(443): 1032–1044.
- Martino L, Elvira V and Louzada F (2016) Alternative effective sample size measures for importance sampling. In: *2016 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, pp. 1–5.
- Martino L, Elvira V and Louzada F (2017) Effective sample size for importance sampling based on discrepancy measures. *Signal Processing* 131: 386–401.
- Meng J, Wang S, Xie Y, Jiang L, Li G and Liu C (2021) Efficient re-localization of mobile robot using strategy of finding a missing person. *Measurement* 176: 109212.
- Moreira AP, Costa P and Lima J (2020) New approach for beacons based mobile robot localization using kalman filters. *Procedia Manufacturing* 51: 512–519.

- Murray LM, Lee A and Jacob PE (2013) Rethinking resampling in the particle filter on graphics processing units. *arXiv preprint arXiv:1301.4019* 135.
- Murray LM, Lee A and Jacob PE (2016) Parallel resampling in the particle filter. *Journal of Computational and Graphical Statistics* 25(3): 789–805.
- Naab C and Zheng Z (2021) Application of the unscented kalman filter in position estimation a case study on a robot for precise positioning. *Robotics and Autonomous Systems* : 103904.
- Nicely MA and Wells BE (2019) Improved parallel resampling methods for particle filtering. *IEEE Access* 7: 47593–47604.
- Panigrahi PK and Bisoy SK (2021) Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University-Computer and Information Sciences* .
- Park JJ and Kuipers B (2011) A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment. In: *2011 IEEE International Conference on Robotics and Automation*. IEEE, pp. 4896–4902.
- Rosas-Cervantes V and Lee SG (2020) 3d localization of a mobile robot by using monte carlo algorithm and 2d features of 3d point cloud. *International Journal of Control, Automation and Systems* 18(11): 2955–2965.
- Roumeliotis SI and Bekey GA (2000) Bayesian estimation and kalman filtering: A unified framework for mobile robot localization. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 3. IEEE, pp. 2985–2992.
- Roy P and Chowdhury C (2021) A survey of machine learning techniques for indoor localization and navigation systems. *Journal of Intelligent & Robotic Systems* 101(3): 1–34.
- Se S, Lowe D and Little J (2001) Local and global localization for mobile robots using visual landmarks. In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 1. IEEE, pp. 414–420.
- Shah R and Kroese DP (2018) Without-replacement sampling for particle methods on finite state spaces. *Statistics and Computing* 28(3): 633–652.
- Shen C, Bai J, Shi B and Chen Y (2023) Trajectory tracking control for wheeled mobile robot subject to generalized torque constraints. *Transactions of the Institute of Measurement and Control* 45(7): 1258–1270.
- Siegwart R, Nourbakhsh IR and Scaramuzza D (2011) *Introduction to autonomous mobile robots*. Cambridge, MA, USA: MIT press.
- Su Z, Zhou X, Cheng T, Zhang H, Xu B and Chen W (2017) Global localization of a mobile robot using lidar and visual features. In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, pp. 2377–2383.
- Thomas A, Mastrogiovanni F and Baglietto M (2021) An integrated localization, motion planning and obstacle avoidance algorithm in belief space. *Intelligent Service Robotics* 14(2): 235–250.
- Thrun S, Beetz M, Bennewitz M, Burgard W, Cremers AB, Dellaert F, Fox D, Haehnel D, Rosenberg C, Roy N et al. (2000a) Probabilistic algorithms and the interactive museum tour-guide robot minerva. *The International Journal of Robotics Research* 19(11): 972–999.
- Thrun S, Burgard W and Fox D (2005) *Probabilistic robotics*. Intelligent robotics and autonomous agents. Cambridge, Mass: MIT Press. ISBN 978-0-262-20162-9.
- Thrun S, Fox D, Burgard W et al. (2000b) Monte carlo localization with mixture proposal distribution. In: *AAAI/IAAI*. pp. 859–865.
- Wang F, Wang Y, He J, Sun F, Li X and Zhang J (2020) Visual object tracking via iterative ant particle filtering. *IET Image Processing* 14(8): 1636–1644.

- Yilmaz A and Temeltas H (2021) Integration of affine icp into the precise localization problem of smart-agvs: Procedures, enhancements and challenges. *Transactions of the Institute of Measurement and Control* 43(8): 1695–1709.
- Yilmaz MK and Bayram H (2023) Particle filter-based aerial tracking for moving targets. *Journal of Field Robotics* 40(2): 368–392.
- Yu S, Yan F, Zhuang Y and Gu D (2020) A deep-learning-based strategy for kidnapped robot problem in similar indoor environment. *Journal of Intelligent & Robotic Systems* 100(3): 765–775.
- Zhang L, Zapata R and Lepinay P (2012) Self-adaptive monte carlo localization for mobile robots using range finders. *Robotica* 30(2): 229–244.
- Zhang Qb, Wang P and Chen Zh (2019) An improved particle filter for mobile robot localization based on particle swarm optimization. *Expert Systems with Applications* 135: 181–193.
- Zhang X, Zhao L, Zhong W and Gu F (2020) Performance analysis of resampling algorithms of parallel/distributed particle filters. *IEEE Access* 9: 4711–4725.
- Zhou N, Lau L, Bai R and Moore T (2021) A genetic optimization resampling based particle filtering algorithm for indoor target tracking. *Remote Sensing* 13(1): 132.
- Zhu A, Pauwels P, Torta E, Zhang H and De Vries B (2024) Data linking and interaction between bim and robotic operating system (ros) for flexible construction planning. *Automation in Construction* 163: 105426. DOI: <https://doi.org/10.1016/j.autcon.2024.105426>. URL <https://www.sciencedirect.com/science/article/pii/S0926580524001626>.