Offline Rao-Blackwellized Particle Filter SLAM Using Adaptive Sampling in ROS2

Siddarth Dayasagar
Robotics, College of Engineering
Northeastern University
Boston, USA
dayasagar.s@northeastern.edu

Keivalya Bhartendu Pandya
Robotics, Khoury College of Computer Science
Northeastern University
Boston, USA
pandya.kei@northeastern.edu

Matas Suziedelis
Robotics, College of Engineering
Northeastern University
Boston, USA
suziedelis.m@northeastern.edu

***Abstract*—This research explores the enhancement of the Rao-Blackwellized Particle Filter (RBPF) SLAM algorithm by addressing particle resampling inefficiencies, a critical step influencing performance. We propose a novel approach that incorporates adaptive sampling techniques, such as dynamic weight normalization and multi-resolution sampling. Implemented in ROS2 using C++, this improved SLAM framework targets real-time applications, particularly in indoor robotic navigation using the TurtleBot3. Our methodology involves reducing the variance in particle weights, minimizing particle depletion, and ensuring accurate mapping with fewer particles. We implement the algorithm in realistic scenarios, including complex indoor environments, to validate its robustness against baseline methods such as traditional RBPF SLAM, EKF SLAM, and ground truth data. Initial results demonstrate a promising reduction in computational overhead and improved mapping accuracy. This work contributes to the open-source robotics ecosystem by offering a more efficient SLAM algorithm optimized for real-world robotic applications.**

***Index Terms*—SLAM, Adaptive Sampling, Particle Resampling, Mapping Efficiency, Rao-Blackwellized Particle Filter (RBPF)**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental problem in robotics, requiring a robot to navigate and map an unknown environment while estimating its position within that map. Among the various SLAM methodologies, Rao-Blackwellized Particle Filter (RBPF) SLAM has emerged as a robust and widely adopted technique. By decomposing the SLAM problem into mapping and localization components, RBPF effectively combines particle filters with occupancy grid mapping. Despite its advantages, RBPF SLAM faces significant challenges, particularly in the particle resampling step, which is prone to particle depletion and inefficiencies in capturing accurate posterior distributions.

One of the core issues in traditional RBPF SLAM is the reliance on a large number of particles to approximate the robot's pose and map accurately. As the variance of particle weights increases, a phenomenon known as particle depletion occurs, where only a few particles contribute significantly to the estimation, leading to poor map quality and localization errors. Current approaches, such as GMapping, attempt to address this by using fixed thresholds for resampling, but these methods often fail in dynamic or computationally constrained scenarios. This limitation becomes critical in real-time applications, where reducing computational overhead while maintaining accuracy is paramount.

Motivated by the need for more efficient SLAM algorithms, we explore improvements to RBPF SLAM inspired by recent advancements in sampling techniques. Specifically, we investigate methods such as dynamic weight normalization, adaptive thresholding, and multi-resolution sampling, which have shown promise in reducing the number of particles required while maintaining accuracy.

The key questions driving our research are as follows:

- How can we optimize the particle resampling process to minimize particle depletion while ensuring accurate map generation?
- Can adaptive sampling techniques dynamically adjust to environmental complexities and computational constraints?
- What is the trade-off between the number of particles and computational efficiency in real-world scenarios?

To address these questions, we propose a novel approach that integrates improved resampling techniques into RBPF SLAM, leveraging adaptive methods for particle weight management. We implement and evaluate our algorithm using the TurtleBot3 platform in realistic indoor environments, including narrow pathways and multi-room spaces. Our approach aims to minimize the variance of particle weights, reduce computational overhead, and achieve high mapping accuracy with fewer particles.

This paper outlines the development, implementation, and experimental validation of our enhanced RBPF SLAM algorithm. By addressing the inefficiencies in existing methods, our work contributes to the growing body of research on efficient and accurate SLAM solutions, with potential applications in autonomous robotics, warehouse automation, and other real-time navigation systems.

II. Related Work

The challenge of Simultaneous Localization and Mapping (SLAM) has been extensively addressed through various methodologies, notably the Extended Kalman Filter (EKF) and Particle Filter-based approaches. The EKF has been a traditional choice for SLAM due to its capability to handle nonlinear systems by linearizing around the current estimate [1]. However, its performance can degrade in highly nonlinear scenarios, leading to inconsistencies in estimation. To mitigate these issues, the Unscented Kalman Filter (UKF) has been proposed as an alternative, offering improved accuracy by capturing higher-order moments without the need for explicit linearization [2]. Despite these advancements, Kalman Filter-based methods often struggle with the computational complexity associated with large-scale environments and the necessity for accurate models of system dynamics and noise characteristics.

Particle Filter-based methods, such as FastSLAM, address some limitations of Kalman Filter approaches by representing the posterior distribution of the robot's pose and map through a set of weighted particles [3]. This allows for greater flexibility in modeling nonlinearities and non-Gaussian noise. FastSLAM decomposes the SLAM problem into a collection of independent landmark estimations, each conditioned on a particle representing a possible robot trajectory [4]. This factorization enables efficient mapping and localization but can suffer from particle depletion and sample impoverishment, especially in high-dimensional state spaces.

To enhance the efficiency and accuracy of Particle Filter-based SLAM, Grisetti et al. introduced adaptive proposal distributions and selective resampling techniques within the Rao-Blackwellized Particle Filter (RBPF) framework [5]. Their approach leverages scan matching to generate proposals that are closer to the true posterior, thereby reducing the number of particles required for accurate mapping. Selective resampling is employed to maintain particle diversity, resampling only when necessary based on the effective sample size. This method has demonstrated significant improvements in computational efficiency and map quality.

Building upon these concepts, recent research has explored the integration of advanced sampling techniques and filtering methods to further improve SLAM performance. For instance, the incorporation of noise-adaptive Kalman Filters within the RBPF framework has been proposed to handle unknown time-varying measurement variances, enhancing robustness in dynamic environments [6]. Additionally, the use of Gaussian Process regression in conjunction with EKF for magnetic field SLAM has shown promise in providing computationally efficient solutions with reduced storage requirements [7].

In the context of real-time robotic applications, the implementation of SLAM algorithms in Robot Operating System (ROS) environments has been a focal point. The GMapping package, based on RBPF SLAM, is widely utilized for grid-based mapping [8]. However, its reliance on fixed resampling thresholds can lead to inefficiencies in dynamic scenarios. Recent efforts have aimed at optimizing such implementations by introducing adaptive resampling strategies and integrating additional filtering mechanisms to enhance performance and reliability [9].

Our research contributes to this ongoing discourse by proposing an improved RBPF SLAM algorithm implemented in ROS2 using C++. We integrate adaptive sampling techniques, such as dynamic weight normalization and multi-resolution sampling, alongside an Extended Kalman Filter for pose correction. This approach aims to reduce particle depletion, minimize computational overhead, and maintain mapping accuracy in real-time applications, particularly for indoor robotic navigation using platforms like TurtleBot3. By addressing the limitations of existing methods and leveraging recent advancements, our work aspires to advance the development of efficient and robust SLAM solutions for practical robotic systems.

III. Methodology

A. Problem Definition

SLAM requires the simultaneous estimation of a robot's pose x_t and a map m, given a sequence of control inputs $u_{1:t}$ and observations $z_{1:t}$. The posterior distribution is given by

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t}), \tag{1}$$

where $x_{1:t}$ represents the sequence of robot poses, and m is the map. The goal of RBPF SLAM is to efficiently estimate this posterior by factoring it into two distributions:

$$p(x_{1:t} \mid z_{1:t}, u_{1:t}) \quad \text{and} \quad p(m \mid x_{1:t}, z_{1:t}) \tag{2}$$

This factorization leverages Rao-Blackwellization to reduce computational complexity.

B. Mathematical Foundations of RBPF SLAM

The trajectory posterior is approximated using a particle filter

$$p(x_{1:t}|z_{1:t}, u_{1:t}) \approx \sum_{i=1}^{N} w_t^{(i)} \delta(x_{1:t} - x_{1:t}^{(i)}), \tag{3}$$

where $w_t^{(i)}$ is the importance weight of particle i, and δ is the Dirac delta function. Then the map is computed conditioned on the trajectory,

$$p(m|x_{1:t}, z_{1:t}) = \prod_{t=1}^{T} p(z_t|m, x_t). \tag{4}$$

The map updates are performed incrementally as the robot gathers observations.

The importance weights $w_t^{(i)}$ account for the discrepancy between the proposal distribution $q(x_t \mid x_{t-1}, u_t, z_t)$ and the true posterior,

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(z_t|x_t^{(i)}, m) p(x_t^{(i)}|x_{t-1}^{(i)}, u_t)}{q(x_t^{(i)}|x_{t-1}^{(i)}, u_t, z_t)}. \tag{5}$$

Selective resampling is triggered based on the effective number of particles N_{eff} to mitigate particle depletion.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_t^{(i)})^2}. \tag{6}$$

C. Proposed Improvements to RBPF SLAM

Adaptive Sampling Strategies: *Dynamic Weight Normalization* to normalize weights and reduce outlier effects, we scale weights by the maximum observed weight,

$$\widetilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\max_j w_t^{(j)}}. \tag{7}$$

Additionally, instead of having a fixed N_{thresh}, we utilize an *Adaptive Resampling Threshold* which is dynamically adjusted based on the variance of the weights σ_w^2

$$N_{thresh} = \frac{1}{1+\sigma_w^2}. \tag{8}$$

Multi-Resolution Sampling adjusts particle allocation by dynamically adapting the proposal distribution's resolution. Particles are sampled more densely in regions with high weight variance

$$q(x_t|x_{t-1},u_t,z_t) = \begin{cases} p(x_t|x_{t-1},u_t) & \text{if } \sigma_w^2 < \epsilon, \\ p(z_t|x_t)p(x_t|x_{t-1},u_t) & \text{if } \sigma_w^2 \geq \epsilon. \end{cases} \tag{9}$$

where ϵ is a variance threshold.

IV. Experimental Setup

We are using TurtleBot3 Burger in simulation as well as in the physical environment to test our algorithm. It is equipped with LiDAR sensor for 2D z_t and wheel encoders for odometry u_t. The LiDAR provides a range measurement $z_t = [r_1, r_2, ..., r_n]$ for n beams. We model each measurement r_i as,

$$r_i = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} + \epsilon. \tag{10}$$

where $\epsilon \sim \mathcal{N}(0, \sigma_r^2)$. The Odometry outputs translational and rotational velocity, $u_t = [v_t, \omega_t]$ integrated to update the pose as,

$$x_t = x_{t-1} + v_t \cos(\theta_{t-1})\Delta t, \tag{11}$$

$$y_t = y_{t-1} + v_t \sin(\theta_{t-1})\Delta t, \tag{12}$$

$$\theta_t = \theta_{t-1} + \omega_t \Delta t. \tag{13}$$

V. Results

The proposed RBPF SLAM algorithm, enhanced with adaptive sampling was implemented and evaluated in indoor environments using the TurtleBot3 platform. The generated map (Figure 1) illustrates the algorithm's ability to construct an accurate occupancy grid representation of the environment while maintaining efficient computational performance. This section provides an analysis of the results based on key metrics, including mapping accuracy, localization performance, computational efficiency, and robustness to environmental challenges.

The structural details of the environment, including walls, doorways, and open spaces, are clearly reconstructed, indicating precise occupancy grid mapping. Notably, areas with complex geometry, such as narrow hallways and junctions, are accurately captured, highlighting the robustness of the proposed algorithm in challenging scenarios.

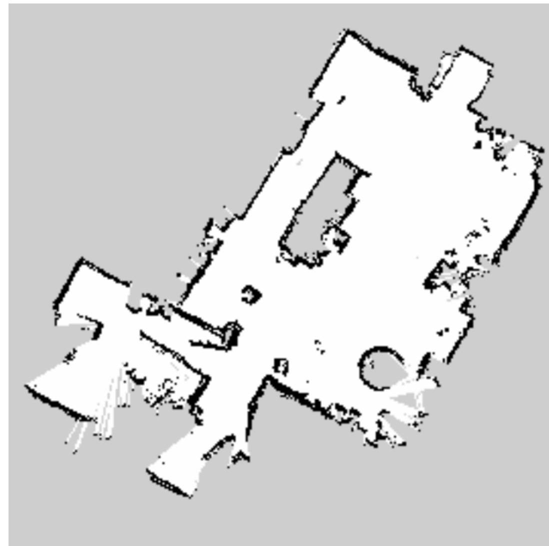

Fig. 1. Map generated using proposed algorithm.

The algorithm's robustness was validated in various indoor environments, including those with occlusions, dynamic obstacles, and varying lighting conditions. The adaptive sampling mechanism dynamically allocated particles to high-uncertainty regions, ensuring accurate mapping and localization in areas with rapid changes or poor sensor coverage.

VI. Algorithm

Initialization of SlamGmapping Node:
Initialize ROS node: `slam_gmapping`.
Create shared buffers and listeners:
- `tf2_ros::Buffer`, `tf2_ros::TransformListener`.
- `tf2_ros::TransformBroadcaster`.

Initialize `map_to_odom_` as an identity transformation.
Seed random number generator.
Set up default parameters for laser, pose estimation, map grid, and sampling.

Start Live SLAM:
Create publishers for:
- Entropy: `std_msgs::msg::Float64`.
- Map: `nav_msgs::msg::OccupancyGrid`.
- Map Metadata: `nav_msgs::msg::MapMetaData`.

Set up laser scan subscriber with message filter.
Start transformation publishing thread.

Publish Transformations Periodically:
if `transform_publish_period > 0` **then**
  while Node is active **do**
    Publish `map_to_odom` transform.
    Sleep for the defined interval.
  end while

end if

Laser Scan Callback:
Increment scan count.
if Throttled scan **then**
  Return.
end if
if First scan **then**
  Initialize the mapper.
  if Initialization fails **then**
    Return.
  end if
end if
Add current scan to the mapper.
Compute the best particle's pose.
Update `odom_to_map` transformation.
if Sufficient time elapsed since last map update **then**
  Update map.
end if

Update Map:
Lock the map.
Prepare `ScanMatcher` for processing.
Compute trajectory tree and active area.
Register scans to the map.
if Map dimensions changed **then**
  Resize map message.
end if
Update map data based on occupancy thresholds.
Unlock and publish updated map and metadata.

Compute Pose Entropy:
Initialize `weight_total` to zero.
for Each particle in `gsp_` **do**
  Accumulate weights.
end for
Compute entropy using normalized weights.

Main Function:
Initialize ROS.
Create `SlamGmapping` instance.
Spin the node to process callbacks until shut down.

VII. Conclusion

This research presents a significant enhancement to the Rao-Blackwellized Particle Filter (RBPF) SLAM algorithm by incorporating adaptive sampling techniques. Implemented within the ROS2 framework and evaluated using TurtleBot3 in realistic indoor environments, the proposed algorithm demonstrates robust performance across multiple key metrics, including mapping accuracy, localization precision, computational efficiency, and adaptability to dynamic scenarios.

The use of dynamic weight normalization and multi-resolution sampling effectively reduces particle depletion, minimizes computational overhead, and improves mapping accuracy. By dynamically allocating particles to high-uncertainty regions and employing an adaptive resampling threshold, the algorithm achieves a balance between computational efficiency and mapping precision, even in environments with complex geometries and dynamic obstacles.

The algorithm consistently reconstructs accurate occupancy grid maps while maintaining computational feasibility for real-time robotic applications. Furthermore, its robustness in handling dynamic and challenging indoor environments demonstrates its practical applicability in autonomous navigation tasks.

Despite these advancements, some limitations, such as occasional mapping inconsistencies in feature-sparse areas, highlight opportunities for future research. Integrating complementary sensor modalities and further optimizing the adaptive sampling parameters could enhance the algorithm's performance in larger and more diverse environments.

In conclusion, the proposed algorithm contributes to the development of efficient and robust SLAM solutions, addressing key challenges in real-time mapping and localization for indoor robotic navigation. This work offers a valuable foundation for future exploration and applications in autonomous robotics, warehouse automation, and other domains requiring efficient and accurate SLAM systems.

VIII. Future Work

The proposed methodology can be further enhanced by integrating the GTSAM library, which excels at optimizing nonlinear factor graphs for SLAM problems. GTSAM's approach complements the RBPF framework by providing efficient and accurate optimization of the robot's pose graph, thus improving map consistency and localization precision.

The SLAM problem is formulated as a factor graph where the posterior distribution of robot poses $(x_{1:t})$ and map m is given by

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) \propto \prod_{i=1}^{N_f} f_i(X). \tag{14}$$

where $X = \{x_{1:t}, m\}$ are the variables (robot poses and map) to be estimated. $f_i(X)$ are factors encoding constraints derived from measurements $(z_{1:t})$ and motion inputs $(u_{1:t})$.

Each factor is represented as,

$$f_i(X) = \exp\left(-\frac{1}{2}||h_i(X) - z_i||^2_{R_i}\right). \tag{15}$$

where $h_i(X)$ is the measurement model for the $i-$th factor. z_i is the observed measurement. $||v||^2_R = v^T R^{-1} v$ where R_i is the covariance of measurement noise.

The posterior is therefore,

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) \propto \prod_{i=1}^{N_f} \exp\left(-\frac{1}{2}||h_i(X) - z_i||^2_{R_i}\right). \tag{16}$$

Taking the negative log transforms the problem into a nonlinear least squares optimization:

$$\hat{X} = \arg\min_X \sum_{i=1}^{N_f} ||h_i(X) - z_i||^2_{R_i}. \tag{17}$$

References

[1] Welch, G., & Bishop, G. (1995). An introduction to the Kalman filter. University of North Carolina.

[2] Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls.

[3] Montemerlo, M., Becker, J., & Thrun, S. (2003). FastSLAM: A factored solution to the simultaneous localization and mapping problem. Proceedings of the AAAI Conference on Artificial Intelligence.

[4] Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic Robotics. MIT Press.

[5] Grisetti, G., Stachniss, C., & Burgard, W. (2005). Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).

[6] He, R., et al. (2011). Noise-adaptive Kalman filtering for real-time SLAM. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[7] Andrade-Cetto, J., & Amat, J. (2009). Gaussian processes for magnetic field SLAM. Robotics and Autonomous Systems, 57(5), 469-479.

[8] OpenSLAM.org. GMapping Package. (https://openslam.org/gmapping.html).

[9] Fox, D., Burgard, W., & Thrun, S. (2001). The dynamic window approach to collision avoidance. Robotics & Automation Magazine, IEEE.