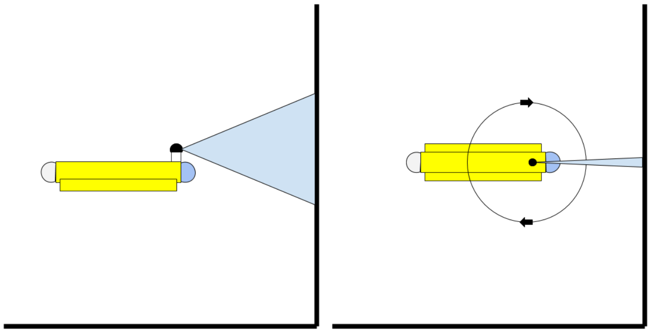
Rao-Blackwellized Particle Filter with grid-mapping for AUV SLAM using Forward-Looking Sonar

Stian Skaalvik Sandøy[†], Takumi Matsuda[‡], Toshihiro Maki[‡], Ingrid Schjølberg[†]

***Abstract*—This paper adresses underwater localization for an AUV using SLAM and Forward Looking Sonar (FLS) data. The proposed method is Rao-Blackwellized Particle Filter (RBPF) with grid mapping. The original RBPF uses multiple suggested trajectories, where each of which has a occupancy grid map. Analogously, range measurements from the FLS are computed for each trajectory. A scan matching procedure is then performed for the measurements of each trajectories and the best matches are then chosen. The performance is confirmed through simulations and experiments. The suggested method enables SLAM in enclosed underwater environments using noisy FLS measurements without any type of artificial landmarks or assumption on the type of environment, and without any increment in runtime complexity for long operation times.**

I. Introduction

Nowadays, there is plenty of activity concerning underwater inspection operations. One example is in the Norwegian aquaculture industry, where inspections are crucial to prevent fish escape due to damage on a fish cage or on equipment [1]. In the past few years, underwater inspection operations have been performed by divers or manually operated ROVs, which can turn out dangerous for the former and costly for the latter. In addition, the lack of high communication bandwidth in underwater environments leads to a need for tethered vehicles in live monitoring, which can be a hazard in the management of the vehicles. This motivates the use of cable-free autonomous underwater vehicles (AUVs), under supervisory human control. This should be able to decrease the need for online data transfer since the AUV can gather information autonomously and then transfer it after inspection. AUVs are commonplace nowadays, but mostly in open water surveys. There are only a few examples of survey operations in more confined environments and one challenge, among others, is to find the position of the vehicle relative to its surroundings [2]–[5]. This problem is popularly known as Simultaneous Localization and Mapping (SLAM) [6]. In recent years, it has been increasingly more used in underwater navigation [7] and a reason why is that it is fundamental for increased autonomy in partly or completely unknown environments. The most attractive SLAM solution is with the usage of unknown data associations, which means doing SLAM in environments without making any assumptions or using artificial landmarks. This provides a more flexible solution that can be used in any environment without having to reconfigure the setup for each operation. Regarding acoustic sensor evaluation in aquaculture net pens, sonars were found to be able to determine a range from an underwater vehicle to a fish net [8]. Therefore, a solution using FLS in a RBPF framework will be investigated here. The main contribution of this paper is the development of a RBPF solution for online underwater navigation scenarios using a FLS as shown in Figure 1a and 1b. The odometry data is obtained from a Doppler Velocity Logger (DVL) and a Fiber Optic Gyro (FOG), and depth measurements from a pressure sensor. Since the AUV has no prior knowledge of the environment, it needs to determine its position online along with building a map. The developed SLAM approach is based on methods presented in [9], that, to the best of the authors' knowledge, has not yet been implemented before in this type of system.



(a) Side view (b) Top view

Fig. 1. Forward Looking Sonar span

II. Related Work

In recent years, underwater SLAM has gained increasing popularity [7]. Many SLAM approaches have been developed for different types of sensor input and environments. The typical sensors on ground vehicles are monocular and stereo camera, and Light Detection And Ranging (LiDAR) with high accuracy and fast update frequency. In underwater environments these kinds of measurements are hard to obtain and therefore are replaced with active sonars which are more robust in underwater environments. However, the accuracy and update rate can still not match that of the LiDAR and camera. In this section we present three of the most important concepts for the RBPF, which are; grid mapping, scan-matching and the

[†] Department of Marine Technology, Norwegian University of Science and Technology, Otto Nielsens Vei 10, 7052 Trondheim, Norway
stian.s.sandoy@ntnu.no, ingrid.schjolberg@ntnu.no

[‡] Institute of Industrial Science, The University of Tokyo 4-6-1 Komaba, Meguro-ku, Tokyo 153-8505, Japan
maki@iis.u-tokyo.ac.jp,matsuda@iis.u-tokyo.ac.jp

over all SLAM framework. This section will dive into previous work in these fields.

A. Occupancy Grid Maps

Occupancy Grid maps are a way of representing space by a finite number of grids. To each grid a value between $[0, 1]$ is assigned, which is the probability of a grid cell being occupied [6]. The value is given by sensor updates from a vehicle range sensor. The concept of occupancy grid mapping was first brought up in [10] and [11].

B. Scan Matching

Scan Matching is the alignment of a point cloud in correspondence to previous ones. Nearly all SLAM approaches have some form of scan matcher, and the most popular is called the Iterative Closest Point (ICP) [12]. It takes an unaligned point cloud and a reference point cloud as input, and aims to, first, find correspondence for each point, and then, to minimize the distance between the correspondences. In [13], a variant of the ICP algorithm was implemented, which was named probabilistic Iterative Correspondence (pIC), and incorporated the uncertainty of points by using a probabilistic length called Mahalanobis distance instead of a metric measure as used in the general ICP. This method was developed to take into account noisy range measurements and errors in the position due to the odometry error. This was in [14] further extended by using a FLS in addition to removing distortions in each point cloud due to vehicle motions, a method named Mechanical Scanning Imaging Sonar pIC (MSISpIC). Another popular method for scan matching is called the Normal Distributions Transform (NDT) [15]. It uses local probability distributions based on each cell in a grid map, along with a reference point cloud and the following one. Its advantage is that no correspondence needs to be established between the points, which is ICP's biggest bottleneck. Another scan matching algorithm is one that maximizes the probability of each point cloud through a gradient decent search of the vehicle pose in relation to an occupancy grid map. This approach was applied quite successfully in [16].

C. Simultaneous Localization and Mapping

There are three main SLAM paradigms, namely the Extended Kalman filter, the Graph-SLAM and the Particle Filter (PF). An example of the Extended Kalman Filter used for AUV SLAM is presented in [17], where the algorithm keeps track on the associations between different poses. Associations and alignment are performed by a pIC scan matcher. Graph-SLAM has, the last years, become more and more popular due to the increasing computational power available [18]. The RBPF is the most popular PF solution for SLAM schemes, and the algorithm was first suggested in [19] using known landmark associations. It was later customized for grid map representation, such that unknown data associations could be used in [9]. The algorithm can be summarized in the following steps:

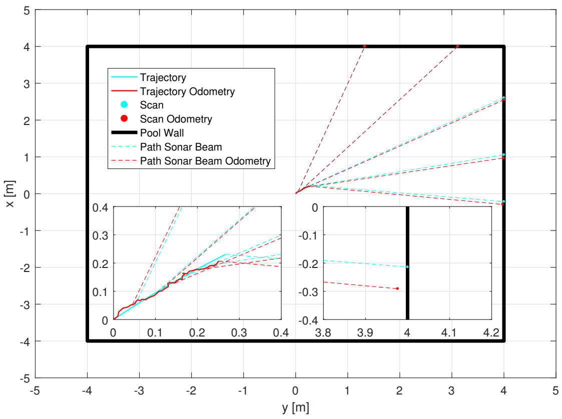


Fig. 2. A ground truth and odometry trajectory are plotted in red and cyan, respectively. Simulated scans are shown in dotted cyan and red. The red and cyan dashed lines illustrate sonar beams for ground truth and odometry, respectively.

1) Sampling: Samples the next generation of particles from a proposal distribution given by previous particle set, odometery data and a scan matched point cloud.
2) Importance Weighing: Compute weight of each particle by the proposal distribution.
3) Resampling: If the effective number of particles is smaller than a certain threshold, the particles are chosen by considering their respective weights computed in step number 2.
4) Map Estimation: Insert point cloud for each particle into their respective map.

The complexity of the RBPF is $O(NM)$ [9], where N is the number of particles, and M is the size of the map. This is due to the fact that each particle has a map, and under resampling each map has to be copied. However, due to adaptive resampling, many of the steps have a complexity of $O(N)$ instead, which corresponds to linear complexity.

III. Overall Framework

This section describes the overall system framework. The vehicle dynamics are explained in Section III-A and the RBPF design in Section III-B.

A. Vehicle Overview

The discrete 2D vehicle dynamics used are summarized by the following:

$$\eta_t = \eta_{t-1} + R(\psi_{t-1})(\nu_{t-1} + w_\nu)\Delta t \tag{1}$$

where $\eta = \begin{bmatrix} x & y & \psi \end{bmatrix}^T$ is the position given in the North-East-Down (NED) frame [20], $\nu = \begin{bmatrix} u_{dvl} & v_{dvl} & r_{fog} \end{bmatrix}^T$ is given in the body frame and $R(\psi) \in \mathbf{R}^{3\times3}$ is the $2D$ rotation matrix, that gives the corresponds between a vector in the body frame and the NED frame. Further, $w_\nu \sim \mathcal{N}(\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T, Q)$, where $Q = \text{diag}(\sigma^2_{dvl}, \sigma^2_{dvl}, \sigma^2_{fog}) \in \mathbf{R}^{3\times3}$.

Algorithm 1 Scan building for each particle trajectory

```
 1: $N \leftarrow number\ of\ particles$
 2: $M \leftarrow number\ of\ endpoint$
 3: $e_{\text{buffer}} \leftarrow zeros(2, M, N)$
 4: $count \leftarrow 1$
 5: repeat
 6:     $u_{t-1} \leftarrow [u, v, r]$
 7:     if $count > M$ then
 8:         $S_t \leftarrow RBPF(u_{t-1}, e_{\text{buffer}})$
 9:         $count = 1$
10:         $e_{\text{buffer}} \leftarrow zeros(2, M, N)$
11:     else
12:         $S_t \leftarrow RBPF(u_{t-1})$
13:     end if
14:     $z_t \leftarrow [r, \theta]$
15:     for i=1 to N do
16:         $\eta_t^i \leftarrow S_t^i$
17:         $e_{\text{buffer}} \leftarrow$ insertMeasurement($e_t(z_t, \eta_t^i)$)
18:     end for
19:     $count \leftarrow count + 1$
20: until End of SLAM
```

An overview of a sensor setup is shown in Table I. A Fiber Optic Gyro (FOG) and a Doppler Velocity Logger (DVL) provide the odometry data, and a pressure sensor provides depth information. Note that the depth measurement is not included in the vehicle's dynamics due to the accurate measurement obtained from the depth sensor. Further, range data, denoted as z_t, is obtained from a FLS. A collection of these measurements is a scans and can be represented as a point cloud. Figure 1a and 1b illustrate the reachable space of the FLS beam from the side and top view, respectively. The beam is wide when seen from the side and narrow from the top view. Furthermore, the FLS is mechanically rotating over the yaw angle as illustrated in Figure 1b. When obtaining ranges from the FLS we assume that range and direction are given directly from the sensor. The measurement is given by the following equation:

$$z_t = \begin{bmatrix} r & \theta \end{bmatrix}^T + \begin{bmatrix} \omega_r & \omega_\theta \end{bmatrix}^T \tag{2}$$

where r is the range from the FLS, θ is the angle of the FLS relative to the vehicle's yaw angle. Further, $\omega_r = \mathcal{N}(0, \sigma_r^2)$ and $\omega_\theta = \mathcal{N}(0, \sigma_\theta^2)$, which means that both are Gaussian distributed with mean zero and standard deviation σ_r and σ_θ, respectively. To obtain the NED frame position of each FLS measurement at time t, denoted as e_t, we need to use the following relation:

$$e_t(z_t, \eta_t) = R(\psi) \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \tag{3}$$

B. Rao-Blackwellized Particle Filter using Forward-Looking Sonar

The key idea of this paper is that the FLS data is integrated into the RBPF presented in this section. The problem can be visualized in Figure 2 from a simulated ground truth in a solid cyan line and the scan, which is the collection of all the cyan dots. When the same scan is viewed from the trajectory based on noisy odometry in the red dots, there is a clear distortion. In [9], the RBPF uses multiple suggested trajectories each of which has a map. Analogously, a scan for each trajectory has also been adapted, which increase the chances of finding the true path. The procedure of is summarized in Algorithm 1. Initialization is performed on Line 1-4, by defining the number of particles and desired FLS measurements in each scan. For each iteration the odometry, denoted u_{t-1}, is used in the RBPF to update all particles and obtaining the new particle set S_t given as:

$$S_t = \{\eta_t^1, \ldots, \eta_t^i, \ldots, \eta_t^N\} \tag{4}$$

At Line 14-18 is the FLS measurement inserted into e_{buffer} given by Equation (3) for each particle. The structure of the e_{buffer} set is shown in Equation (5). At Line 19, the count of the number of scans inserted is incremented. The procedure is repeated until the scan buffer reaches a predefined size, denoted as M. When e_{buffer} is full, it is sent into the RBPF at Line 8. Note that the count and buffer are reinitialized at Line 9 and 10.

$$\begin{aligned} e_{\text{buffer}} = \{&\{e_{t_1}^1, \ldots, e_{t_M}^1\}, \ldots, \{e_{t_1}^i, \ldots, e_{t_M}^i\}, \\ &\ldots, \{e_{t_1}^N, \ldots, e_{t_M}^N\}\} \end{aligned} \tag{5}$$

The RBPF function in Algorithm 1 is summarized in Algorithm 2. It is similar to the one given in [9], but with some differences. The first difference is the scan matching algorithm used. It is to time consuming to acquire enough measurements to use an gradient decent search approach due to a low update rate from the FLS in comparison to a LiDAR, as used in [9]. For that reason, the ICP algorithm is chosen instead, since it only requires one reference scan for each particle. The reference scans are stored after each iteration,as seen at Line 16. Note that also when resampling at Line 25, a copy of the scans is required. The reference scan is stored in e_{ref} and has the following structure:

$$\begin{aligned} e_{\text{ref}} = \{&\{e_{t_1}^1, \ldots, e_{t_{M \times R}}^1\}, \ldots, \{e_{t_1}^i, \ldots, e_{t_{M \times R}}^i\}, \\ &\ldots, \{e_{t_1}^N, \ldots, e_{t_{M \times R}}^N\}\} \end{aligned} \tag{6}$$

where R is the number of scans stored. When all R scans have been inserted, the old scans are replaced, as in a sliding window approach. The ICP variant used here is an implementation from *MATLAB* described in [21]. Furthermore, on Lines 2-17, the measurement update is performed for each particle. Firstly, in Line 3, a new pose x is suggested from the ICP algorithm. If the root mean square error (rmse) for the scan match is larger than a threshold T_1, the particle is updated on the basis of odometry in Line 5-6. If not, a set of points $x^{1:K} = \{x^1, x^2, \ldots, x^K\}$ is chosen from an uniform distribution, denoted as $\mathcal{U}$ at Line 8. This distribution

Algorithm 2 RBPF [9]

```
1:  if $e_{\text{buffer}}$ $available$ then
2:      for $i = 1$ to $N$ do
3:          $x, rmse \leftarrow$ ICP($e^i_{\text{buffer}}$,$e^i_{\text{ref}}$)
4:          if $rmse > T_1$ then
5:              $S^i_t \leftarrow S^i_{t-1} + \mathcal{N}(u_{t-1}, Q)$
6:              $w_i = p(e_{\text{buffer}} | m^i_{t-1}, {S_t}^i)$
7:          else
8:              $x^{1:K} \leftarrow \mathcal{U}(x, \epsilon)$
9:              $w^{1:K} \leftarrow p(x^k | u_{t-1}, S^i_{t-1}) p(e_{\text{buffer}} | m^i_{t-1}, x^k)$
10:             $\mu_t \leftarrow$ mean($x^{1:K}, w^{1:K}$)
11:             $\Sigma_t \leftarrow$ cov($\mu_t$,$x^{1:K}, w^{1:K}$)
12:             $S^i_t \leftarrow \mathcal{N}(\mu_t, \Sigma_t)$
13:             $w^i_t \leftarrow w^i_{t-1} \sum_{k=1}^{K} w_k$
14:         end if
15:         $m^i_t \leftarrow$ updateMap($S^i_t$, $e_{\text{buffer}}$)
16:         $e_{\text{ref}} \leftarrow$ insertScan($S^i_t$, $e_{\text{buffer}}$)
17:     end for
18: else
19:     for $i = 1$ to $N$ do
20:         $S^i_t \leftarrow S^i_{t-1} + \mathcal{N}(u_{t-1}, Q)$
21:     end for
22: end if
23: $N_{\text{eff}} = 1 / \sum_{i=1}^{N} (\tilde{w}_i)$
24: if $N_{\text{eff}} < T_2$ then
25:     $\{S_t, m_t, e_{\text{ref}}\}$ = resample($S_t, m_t, e_{\text{ref}}, w_t$)
26:     $w^i_t \leftarrow 1/N$
27: end if
```

is centered around x with a size given by ϵ. In Line 9, each point x^k gets a weight depending on the probability of location given the odometry , the scan and the occupancy grid map [22]. Thereafter, the weighted mean and covariance are computed. This is further used to estimate an Gaussian distribution, from where the next particle pose is drawn, as seen in Line 12. The particle weight is updated in Line 13 followed by an update of the map and scan reference for each particle. In Line 23, the effective number of particles is computed. This is a measure on how well the weights are distributed on each particle and are used to determine, by a threshold T_2, when the resampling step is necessary. Note that $\tilde{w}_i$ in Line 23 denotes the normalized particle weight. For more details and derivation of RBPF, see [9].

IV. Simulations

This section presents results of the simulations of the method described in Section III. The parameters used in the simulations are shown in Table II and I. As illustrated in Figure 2, the simulation environment is a 8m×8m pool and all measurements are taken at the same depth. The AUV follows a path along the pool walls based on its odometry data while the FLS is scanning in 360 degree. Two simulations are performed with the following difference:

1) No sonar noise included.
2) Sonar noise included.

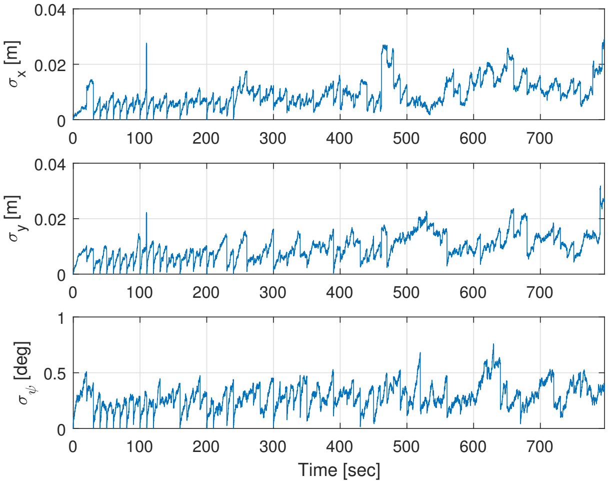


Fig. 3. Particle covariance of simulation 2.

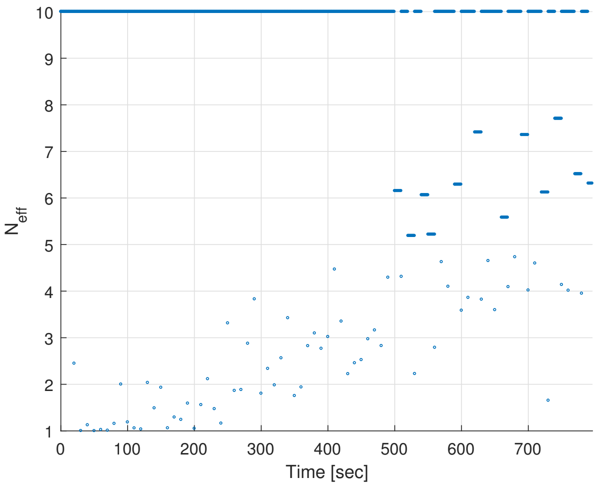


Fig. 4. Effective number of particles in simulation 2.

Simulations 1 and 2 are presented in Figures 5 and 6, respectively. They show the estimated trajectories along with the grid map. The main plot illustrates an overview, where odometry is plotted in a dashed blue line, the weighted particle mean in a red solid line and the ground truth in a thick green solid line. The occupancy grid map for the particle with highest weight is shown by a gray scale ranging from $[0.5, 1]$. The right subplot in both figures is a blown up view of the trajectories. One color corresponds to one round and is enumerated in the legend. Table III shows the RMSE errors for each of the simulations for both the particle mean and the odometry trajectory. Furthermore, in Figures 4 and 3, the effective number of particles and the particle covariance estimative are plotted for simulation 2.

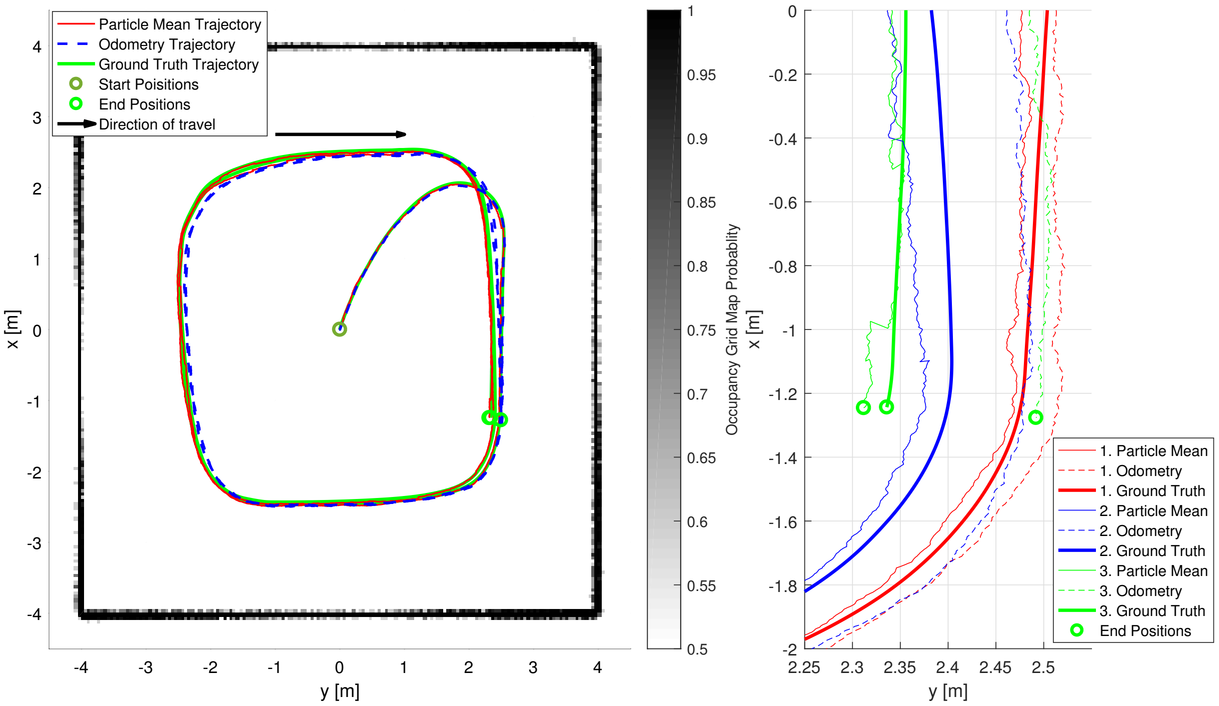


Fig. 5. Simulation results without sonar noise. In the right subplot is each round enumerated in the legend.

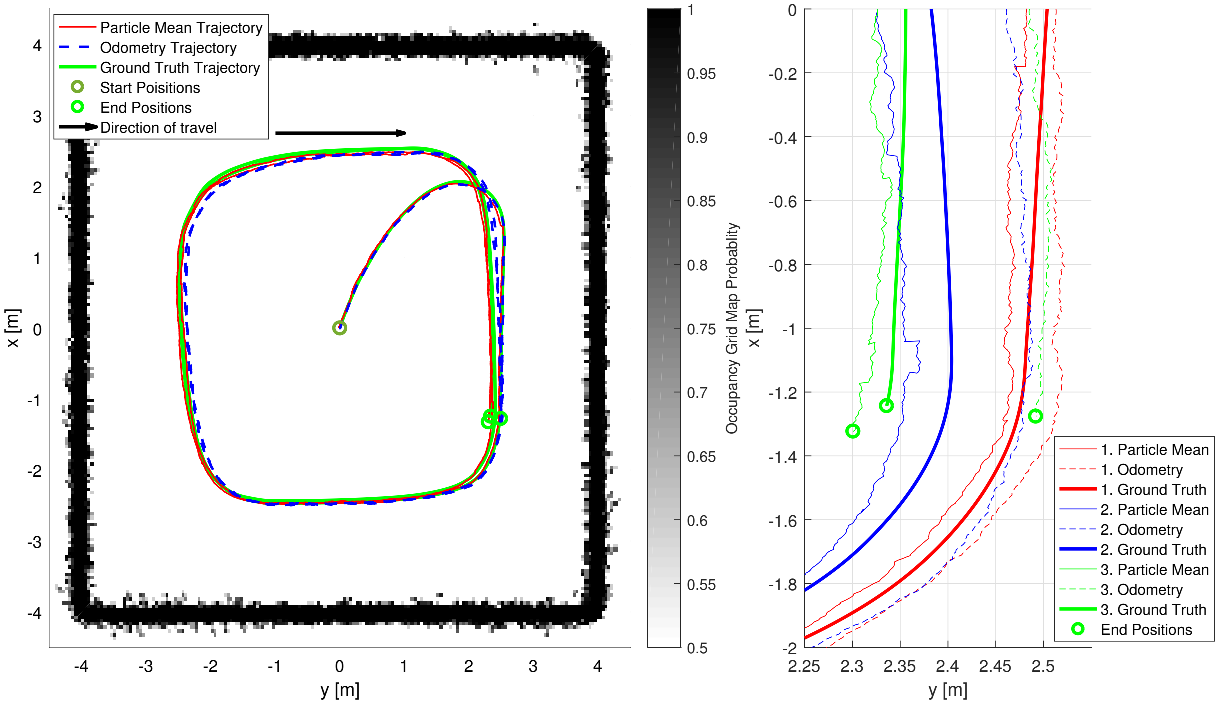


Fig. 6. Simulation results with sonar noise. In the right subplot is each round enumerated in the legend.

TABLE I
SENSOR OVERVIEW

Sensor	Notation	Update Rate [Hz]	Noise Std.
FOG	r	5	σ_{FOG} [deg/s]
DVL	ν	5	σ_{DVL} [m/s]
Pressure sensor	d	10	0.02 [m]
Tritech Micron	z	20	$(0.08[\text{m}], 1.5[\text{deg}])$

TABLE II
RBPF PARAMETERS

RBPF Parameter	Value
N	10
M	200
K	500
R	10
ϵ	$(0.01, 0.01, 2.5)$[m,m,deg]
σ_{FOG}	0.075[deg/s]
σ_{DVL}	$0.00084 + 0.038\sqrt{\lvert\nu\rvert}$ [m/s]
$(p_{\text{hit}}, p_{\text{miss}})$	(0.6, 0.5)
T_1	0.1
T_2	$0.5N$
Resolution of grid map	0.05[m]

V. EXPERIMENTS

This section describes experiments conducted in a pool of 7.5m × 7.5m at Institute of Industrial Science, The University of Tokyo. Section V-A presents a noise analysis of the FLS and Section V-B shows the conducted SLAM experiment.

A. FLS Noise Analysis Experiment

In order to ensure correct noise properties implemented into simulations, an noise analysis was performed in a water tank with the FLS, a Tritech Micron [23]. The tests were conducted by pointing the sonar towards the pool floor instead of pointing it forward. This allowed us to measure the range of the beam for different depths and also compare the results with the ranges obtained from the DVL. The result is shown in the sensor overview at Table I. The standard deviation of the range was 0.08m after removing outliers. The angular standard deviation was not measured, but it was set to 1.5 degrees which correspond to the beam width. Note that the DVL and FOG standard deviation are also given in Table I and are obtained from [24].

B. RBPF SLAM Experiment

A pool experiment was conducted in order to verify the simulation results. A trail of 1100 seconds were performed and the vehicle used was the Tri-Dog 1 (TD1) [25], shown in Figure 7. TD1 was programed to run a predefined path doing 4 rounds in a square of 2.5m × 2.5m at a speed of 0.075m/s and diving one meter down at the second round. However, since the proposed method only consider the XY-plane, the depth will not be taken into consideration. The vehicle navigated using odometry data from the DVL and FOG while gathering range data from the FLS, which was scanning in a $\pm 60\,\text{deg}$ sector relative to the AUV's heading. In post-processing, all gathered sensor data was run through the RBPF SLAM in the same manner as the simulated data. All parameters in the filter have the same values as used in the simulated cases. Note that odometry measurements are in a 5 Hz rate while the update frequency of the FLS in 20 Hz, therefore is an interpolated scheme for each particle necessary at one time instant to the next. However, since we have low velocity 0.075m/s in this case, we assume that the the errors are low. An overview of the run is shown in Figure 8 and the particle covariance and effective number of samples is shown in Figures 9 and 10, respectively. Since there are no ground truth available, it is difficult to obtain a value for the RMSE. However, some shells were scattered over the pool floor to have a reference point and pictures where taken of the floor. Since the camera configurations are known the position is computed for the first and last encounter. With no error the shell should have the same position. The computed position one shell is illustrated in the right subfigure of Figure 8, where the red and blue star illustrate the computed shell position for the odometry trajectory for first and last encounter respectively, and in a red and blue circle for the SLAM trajectory. The errors from the first to the last encounter is shown in Table IV, where the SLAM trajectory has a much lower error than the odometry.

TABLE III
RMSE FOR ESTIMATED VS. GROUND TRUTH IN SIMULATIONS

Error Measure	RMSE [m]	RMSE [deg]
Odometry RMSE	0.0985	0.7096
SLAM RMSE without sonar noise	0.0382	0.2746
SLAM RMSE with sonar noise	0.0590	0.4928

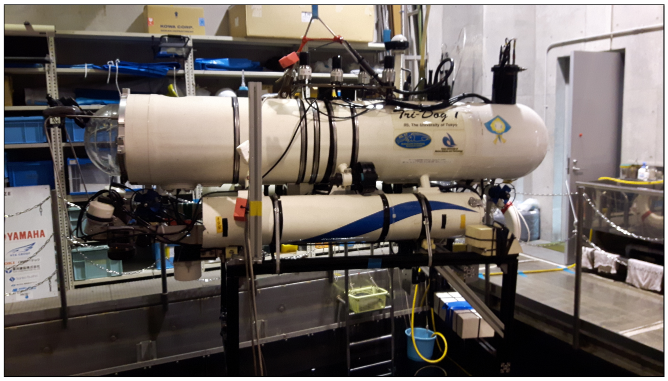


Fig. 7. The hovering type AUV called Tri-Dog 1 (TD1).

VI. DISCUSSION

This section discusses the results presented in Section IV and V. Starting with simulation 1, where no sonar noise is included, the RBPF shows an excellent accuracy as can be seen from the low RMSE in Table III. For Simulation 2, there is an expected increase when the sonar noise is included, but it still manages to follow the ground truth trajectory, which can

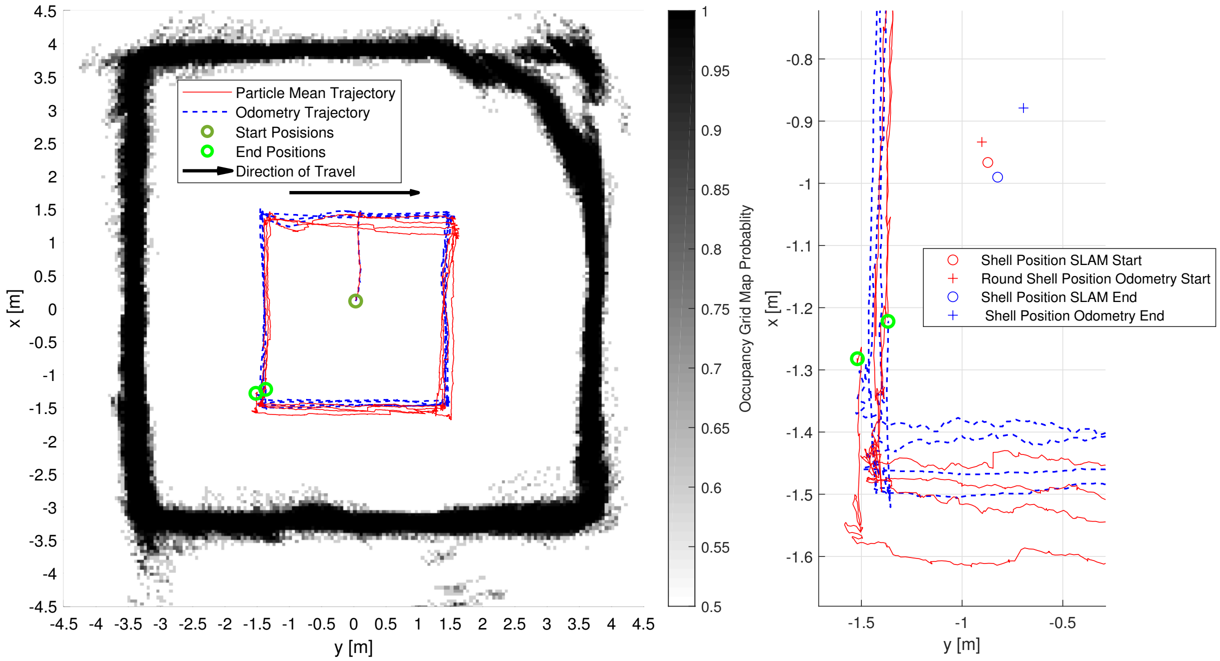


Fig. 8. Results from the experiment.

TABLE IV
ESTIMATE OF SHELL POSITION ERROR FROM FIRST TO LAST ENCOUNTER

Trajectory	dx **[m]**	dy **[m]**	$\sqrt{dx^2 + dy^2}$**[m]**
Odometry Trajectory	0.0547	0.2068	0.2139
SLAM Trajectory	-0.0236	0.0487	0.0553

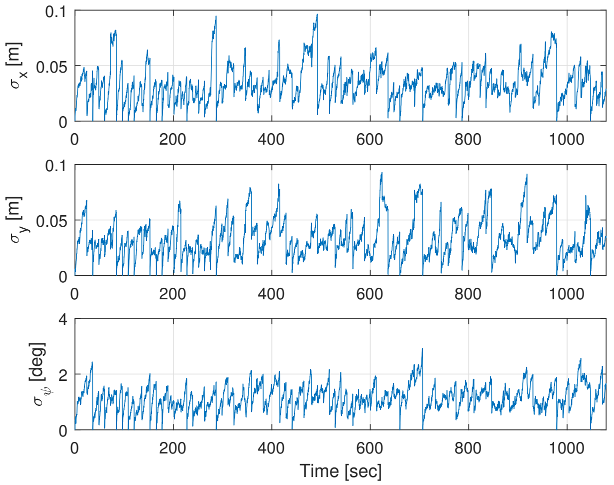


Fig. 9. Particle covariance for the experiment

also be seen in Figure 6, and from the low RMSE values in Table III. Besides, in the grid maps of Figure 5 and 6 the sonar noise is noticeable by the thickness of the walls. Note that the resolution of each cell is 0.05 meters. For the noisy sonar case, the update of the map is an issue if cells are considered free when updating, which lowers its probability value. So, in this case, the probability value of a free update in the inverse sensor model is set to 0.5, which leads to not updating and inserting only the endpoint into the grid map.

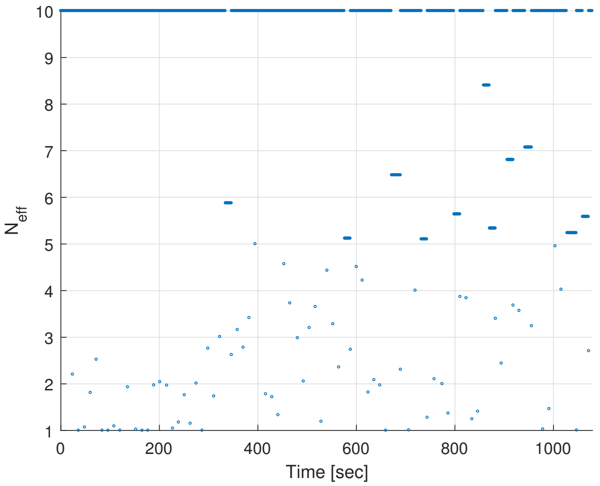


Fig. 10. Effective number of particles for the experiment

Particle covariance of simulation 2 and the experiment run are plotted in Figure 3 and 9, respectively, where the values plotted correspond to the diagonal elements of the particle covariance matrices. As seen in the figures, the covariance increases due to the odometry error, but for each measurement update the covariance gets reduced and is thereby bounded. Note that the experimental particle covariance is about two times higher than the simulated one, which is likely due to the presence of outliers in the sonar measurements in the real case. This can also be seen in the occupancy grid map in Figure 8, and leads to worse scan matching conditions for the ICP algorithm, making the measurement update more dependent on the odometry, as seen in Line 5-6 of Algorithm 2. However, the method is robust despite the outliers as seen by the relative estimated shell position from the first to last encounter, illustrated in Figure 8.

The effective number of particles is plotted in Figure 4 and 10 for simulation 2 and experiment respectively. It indicates how well the weights are distributed over the particles. Their effective number is always at a maximum of 10 after each resampling, and decreases monotonically after each measurement update. If the effective number of particles is higher than 5, it is possible to skip the resampling step since the particles' spread is large enough.

Regarding the run time, it takes 5.3 minutes to run through the experiment simulation offline in MATLAB using a HP EliteBook with Intel(R) Core(TM) i7-6500U CPU at 2.5GHz. In comparison, the full operation takes about 18 minutes, which means that it uses only 29.5% of the time when run offline. These numbers indicates that the data can be processed before a new update is performed, therefore, being able to be used in online applications.

VII. Conclusions

In this paper we presented a method called Rao-Blackwellized Particle Filter using grid maps in a new application. This method was here implemented for SLAM in enclosed underwater environments using noisy FLS measurements. This approach has the benefit of not depending on any type of artificial landmarks or assumption about the type of environment. Simulations of the algorithm showed that, by correcting the measurements and this way bounding the error, the performance was improved in comparison to the odometry trajectory. In order to confirm the simulation results, an experiment was performed in a test basin at Institute of Industrial Science, in The University of Tokyo, with a hovering type AUV called TD1, which gathered data that was later post-processed using the proposed algorithm. The experimental results support the simulations by showing that the AUV drifts less, having a more accurate positioning then when relying on odometry data only.

Acknowledgments

This work is funded by the Norwegian Research Council projects Reducing Risk in Aquaculture (254913)

References

[1] . Jensen, T. Dempster, E. B. Thorstad, I. Uglem, and A. Fredheim, "Escapes of fishes from norwegian sea-cage aquaculture: causes, consequences and prevention," *Aquaculture Environment Interactions*, vol. 1, no. 1, pp. 71–83, 2010. [Online]. Available: http://www.int-res.com/abstracts/aei/v1/n1/p71-83/

[2] A. Mallios, P. Ridao, D. Ribas, M. Carreras, and R. Camilli, "Toward autonomous exploration in confined underwater environments," *Journal of Field Robotics*, vol. 33, no. 7, pp. 994–1012, 2016. [Online]. Available: http://dx.doi.org/10.1002/rob.21640

[3] N. Fairfield, G. Kantor, and D. Wettergreen, "Real-time slam with octree evidence grids for exploration in underwater tunnels," *Journal of Field Robotics*, vol. 24, no. 1-2, pp. 03–21, 2007. [Online]. Available: http://dx.doi.org/10.1002/rob.20165

[4] D. Ribas, P. Ridao, J. D. Tardós, and J. Neira, "Underwater slam in man-made structured environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 898–921, 2008. [Online]. Available: http://dx.doi.org/10.1002/rob.20249

[5] A. Mallios, P. Ridao, D. Ribas, and E. Hernández, "Scan matching slam in underwater environments," *Autonomous Robots*, vol. 36, no. 3, pp. 181–198, 2014. [Online]. Available: https://doi.org/10.1007/s10514-013-9345-0

[6] S. Thrun, W. Burgard, and D. Fox, "Probabilistic robotics. 2005," *Massachusetts Institute of Technology, USA*, 2005.

[7] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[8] P. Rundtop and K. Frank, "Experimental evaluation of hydroacoustic instruments for rov navigation along aquaculture net pens," *Aquacultural Engineering*, vol. 74, pp. 143–156, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0144860916300012

[9] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[10] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.

[11] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI magazine*, vol. 9, no. 2, p. 61, 1988.

[12] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics '91*, vol. 1611. SPIE, Conference Proceedings, p. 21.

[13] L. Montesano, J. Minguez, and L. Montano, "Probabilistic scan matching for motion estimation in unstructured environments," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Conference Proceedings, pp. 3499–3504.

[14] J. D. Hernandez, K. Istenic, N. Gracias, N. Palomeras, R. Campos, E. Vidal, R. García, and M. Carreras, "Autonomous underwater navigation and optical mapping in unknown natural environments," *Sensors (Basel, Switzerland)*, vol. 16, no. 8, p. 1174, 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5017340/

[15] P. Biber and W. Straßer, "The normal distributions transform: A new approach to laser scan matching," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3. IEEE, Conference Proceedings, pp. 2743–2748.

[16] S. Kohlbrecher, O. v. Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, Conference Proceedings, pp. 155–160.

[17] A. Mallios, P. Ridao, E. Hernandez, D. Ribas, F. Maurelli, and Y. Petillot, "Pose-based slam with probabilistic scan matching algorithm using a mechanical scanned imaging sonar," in *OCEANS 2009-EUROPE*, Conference Proceedings, pp. 1–6.

[18] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[19] A. Doucet, N. d. Freitas, K. P. Murphy, and S. J. Russell, "Rao-blackwellised particle filtering for dynamic bayesian networks," in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. 720075: Morgan Kaufmann Publishers Inc., Conference Proceedings, pp. 176–183.

[20] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[21] 2017. [Online]. Available: https://www.mathworks.com/help/vision/ref/pcregrigid.html?s_tid=doc_ta

[22] 2018. [Online]. Available: https://www.mathworks.com/help/robotics/ug/occupancy-grids.html?s_tid=gn_loc_drop

[23] 2018. [Online]. Available: http://www.tritech.co.uk/product/small-rov-mechanical-sector-scanning-sonar-tritech-micron

[24] T. Maki, “Research on autonomous observation of seafloor environment,” Thesis, 2008.

[25] H. Kondo, “Development of an autonomous underwater vehicle ”tri-dog” toward practical use in shallow water,” *J Robot Mechatron*, vol. 13, no. 2, pp. 205–211, 2001. [Online]. Available: https://ci.nii.ac.jp/naid/20000873426/en/