



Automated Systems & Technologies  
25-26 May 2015 • St. Petersburg, Russia

## **APPLYING CONDITIONAL PARTICLE FILTERS FOR BEARING-ONLY SLAM PROBLEM**

D.G.Arsenjev, N.A.Berkovskii

SPbSPU  
Polytechnicheskaya Street, 29  
195251 St. Petersburg, RF  
[berkovsk@mail.ru](mailto:berkovsk@mail.ru)

### **Abstract**

The recurrent algorithm for solving 2D Bearing-only SLAM problem is proposed. This algorithm is based on the Sequential Monte Carlo method and Rao-Blackwellisation technique, decomposing the state-vector into two parts which are the robot's and the landmarks' positions. The trajectories of the robot are modeled independently while the landmarks' coordinates are modeled as conditional distributions. These conditional distributions are found using the individual particle filters corresponding to each generated trajectory of the robot. Our method has linear complexity growth with respect to number of the landmarks. The generalization of the proposed algorithm to 3D case is trivial.

### **INTRODUCTION**

Bearing-only Simultaneous Localization and Mapping (SLAM) is attractive choice because of the inexpensive navigational equipment of the mobile robot, since a range finder is not used and a single camera is only required [1,2,3]. In comparison with "classical" SLAM, when a range finder is assumed to be available, the main difficulty is that the prior uncertainty in the disposition of landmarks is very high, so Extended Kalman Filter (EKF), which is used in well-known Fast SLAM 1.0 and Fast SLAM 2.0 algorithms [4], appears to be unsuitable. Apparently, the only EKF-based method to solve

Bearing-only SLAM is the Gaussian Sum Filter (GSF) [5], but it has exponential growth of the complexity with respect to the landmark number [2]. An alternative approach is proposed in [1], where the trajectories of the robot are modeled independently using particle filters [6, 7] and the trapezoids, whose shapes change sequentially, serve for modeling the landmark uncertainties. The algorithm proposed in this paper, in some respects, is allied to the method described in [1]. We also apply the decomposition of the state-space and the Rao-Blackwellisation technique [6, 8] but instead of the trapezoids individual particle filters are used for approximating the conditional distributions corresponding to each modeled trajectory of the robot. In this context, the set of the particles representing the state-vector and set of their weights may be called the main particle filter. We believe such an approach produces the high accuracy since when the number of the particles in each individual filter is large enough, the approximation of the conditional distribution is practically exact. At the great number of independent trajectories applying these individual particle filters is time consuming of course, but removing the particles with too small weights significantly speed up the computational process. Our method has linear complexity growth with respect to the landmark number and can be trivially generalized to 3D case. It should be note that in [9] we have proposed another method based on the particle filters for solving Bearing-only SLAM. That method doesn't use the individual particle filters so it works very fast. Essentially, method proposed in [9] is significantly simplified version of the algorithm given in this paper. In [9], only one trajectory of the robot constructed in some empirical way is employed and, as a result, the method from [9] yields high accuracy of estimating the state-vector exclusively at the low values of system noises. The algorithm proposed in this paper, contrary, is applicable for any values of the system noises; moreover, it may be considered to be a standard the other suboptimal methods are compared with. The paper is organized as follows: in section 2 the problem is formulated and denotations are introduced, in section 3 the proposed algorithm is described, and our simulation results are represented in section 4.

## PROBLEM FORMULATION

We use the kinematic discrete model of the robot's motion. Let a mobile robot, whose coordinates are  $\mathbf{r}=(r_x, r_y)$ , moves on the  $x-y$  plane along a scheduled trajectory (Fig. 1).

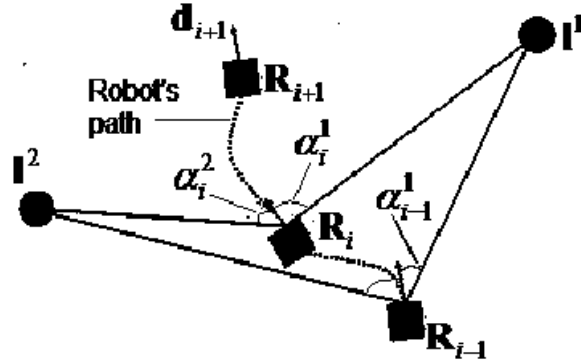


Figure.1. The motion of robot and taking of the measurements in bearing-only SLAM.

The robot starts at the known point  $\mathbf{r}_0 = (r_{x0}, r_{y0})$  and then makes  $N_{st}$  consecutive movements (steps). By  $\theta$  denote the orientation of the robot with respect to the  $x$  axis. Suppose there are  $N_l$  landmarks on the plane and let  $\mathbf{l}^s = (l_x^s, l_y^s)$ ,  $s=1,2,\dots,N_l$  be their coordinates. The robot's motion can be described as follows:

$$\Delta \mathbf{r}_i = ((\rho_i + e_i^\rho) \cos \theta_{i-1}, (\rho_i + e_i^\rho) \sin \theta_{i-1}), \quad (1)$$

$$\theta_i = \theta_{i-1} + \Delta \theta_i + e_i^\theta, \quad (2)$$

where  $\rho_i, \Delta \theta_i$  are control parameters forming the robot's path,  $i \in \overline{1, N_{st}}$ . The system noise processes  $e_i^\rho, e_i^\theta$  are described by their probability density functions (p.d.f.)  $\mathbf{N}(e_i^\rho, 0, \sigma_i^\rho)$  and  $\mathbf{N}(e_i^\theta, 0, \sigma_i^\theta)$ , where  $\mathbf{N}(x, m, \sigma)$  is the p.d.f. of the normal distribution with mean  $m$ , variance  $\sigma^2$  and argument  $x$ . The random variables  $e_i^\rho$  and  $e_j^\theta$  (for any  $i, j$ ) as well as  $e_i^\rho$  and  $e_j^\rho, e_i^\theta$  and  $e_j^\theta$  (for  $i \neq j$ ) are assumed to be independent of each other. Let  $\mathbf{r}_i = (r_{xi}, r_{yi})$  be the robot position at the step  $i$ , then we have

$$\mathbf{r}_i = \mathbf{r}_{i-1} + \Delta \mathbf{r}_i, \quad (3)$$

where  $\Delta \mathbf{r}_i$  is found by (1) и (2). The model defined by (1) – (3) is more suitable for a walking robot, but it may be also used for a wheeled robot. The initial orientation  $\theta_0$  is known. Let  $\alpha_i^s$  be the angle between the vectors  $\mathbf{d}_i = (\cos \theta_i, \sin \theta_i)$  and  $\mathbf{l}^s - \mathbf{r}_i$ ,  $\alpha_i^s \in (-\pi, \pi]$ . More precisely,

$$\alpha_i^s = \arccos \left( \frac{\mathbf{d}_i \cdot (\mathbf{l}^s - \mathbf{r}_i)}{\|\mathbf{l}^s - \mathbf{r}_i\|} \right) \operatorname{sgn} \begin{vmatrix} \cos \theta_i & \sin \theta_i \\ l_x^s - r_{xi} & l_y^s - r_{yi} \end{vmatrix}, \quad (4)$$

where  $\| \cdot \|$  is a vector magnitude. Consider the subset  $M \subset \overline{0, N_{st}}$ ,  $0 \in M$ . If  $i \in M$  then for all  $s \in \overline{1, N_l}$  the noisy measurements  $m_i^s$  are supposed to be available and

$$m_i^s = \alpha_i^s + e_{is}^\alpha, \quad (5)$$

where  $e_{is}^\alpha$  are independent random variables with p.d.f.  $\mathbf{N}(e_{is}^\alpha, 0, \sigma_i^\alpha)$ .

Let us introduce the notations  $\mathbf{R}_i = (\mathbf{r}_i, \theta_i)$ ,  $\mathbf{L} = (\mathbf{l}^1, \mathbf{l}^2, \dots, \mathbf{l}^{N_r})$ , and  $\mathbf{m}_{1:i} = \{m_k^s\}$  where  $s \in \overline{1, N_l}$  and  $k \in \overline{1, i} \cap M$ . Our aim is to estimate recursively in time the posterior p.d.f.  $f(\mathbf{R}_i, \mathbf{L} | \mathbf{m}_{1:i})$  and, in particular, approximately calculate the conditional mean  $E_{f(\mathbf{R}_i, \mathbf{L} | \mathbf{m}_{1:i})}(\mathbf{R}_i, \mathbf{L})$ ,  $i \in \overline{1, N_{st}}$ , and covariance matrix of the state vector  $(\mathbf{R}_i, \mathbf{L})$  with respect to  $f(\mathbf{R}_i, \mathbf{L} | \mathbf{m}_{1:i})$ . Covariance matrix is needed for evaluating the mean-square error of estimate.

## DESCRIBING THE PROPOSED METHOD

The following factorization (6) of the conditional p.d.f. is the theoretical base for the proposed method:

$$f(\mathbf{R}_{0:i}, \mathbf{L} | \mathbf{m}_{1:i}) = f(\mathbf{L} | \mathbf{R}_{0:i}, \mathbf{m}_{1:i}) f(\mathbf{R}_{0:i} | \mathbf{m}_{1:i}), \quad (6)$$

where  $\mathbf{R}_{0:i} = \{\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_i\}$ .

The formula (6) follows from the Bayesian rules [6]. The details of the implementation are reported below.

*Initialization:* Put  $\mathbf{R}_{0(j)} = \mathbf{R}_0$  for all the initial states of the robot and  $q_{0(j)} = (N)^{-1}$  for initial the weights of trajectories,  $j \in \overline{1, N}$  (initialization of the main filter). Generate vectors  $\mathbf{l}_{0(m)}^{s,j}$ ,  $m \in \overline{1, N_1}$  with an a priori density  $f(\mathbf{l}_0^{s,j})$  for each  $j \in \overline{1, N}$  и  $s \in \overline{1, N_L}$  (initialization of conditional filters) and set  $w_{0(m)}^{s,j} = N_1^{-1}$ ,  $m \in \overline{1, N_1}$ . Modeling with the density  $f(\mathbf{l}_0^{s,j})$  is performed as follows.

Firstly, random variables  $u_{(m)}^{s,j}$ ,  $m \in \overline{1, N_1}$  are sampled from the uniform distribution on the interval  $[u_{\min}, u_{\max}]$ , where  $u_{\min}$  and  $u_{\max}$  are the design parameters, that are equal to the minimum and maximum distances of landmarks from the robot we expect a priori. Secondly, the random variables  $\varphi_{(m)}^{s,j}$ ,  $m \in \overline{1, N_1}$  with density  $\mathbf{N}(\varphi_{(m)}^{s,j}, \overline{\varphi^s}, \sigma_0^\alpha)$  are modeled, where  $\overline{\varphi^s}$  is the angle between the  $x$  axis and the ray forming angle  $m_0^s$  with the initial vector  $\mathbf{d}_0 = (\cos \theta_0, \sin \theta_0)$ . Finally, put

$$\begin{cases} l_{0(m)x}^{s,j} = u_{(m)}^{s,j} \cos(\varphi_{(m)}^{s,j}) + r_{x0} \\ l_{0(m)y}^{s,j} = u_{(m)}^{s,j} \sin(\varphi_{(m)}^{s,j}) + r_{y0} \end{cases},$$

*Transition*  $(i-1) \rightarrow i$ ,  $i \in \mathfrak{N}$ . Let there be realizations  $\mathbf{R}_{i-1(j)}$  with weights  $q_{i-1(j)}$ ,  $j \in \overline{1, N}$ , sets of constant particles  $\{\mathbf{l}_{i-1(m)}^{s,j}\}_{m=1}^{N_1}$  with weights  $w_{i-1(m)}^{s,j}$ .

1. Generate  $\mathbf{R}_{i(j)}$  with a density  $f(\mathbf{R}_{i(j)} | \mathbf{R}_{i-1(j)})$  and denote the results as

$\mathbf{R}_{i(j)}^*$ . Calculate unnormalized weights

$$Q_{i(j)} = q_{i-1(j)} \prod_{s=1}^{N_L} \left( \sum_{m=1}^{N_1} f(\mathbf{m}_i^s | \mathbf{l}_{i-1(m)}^{s,j}, \mathbf{R}_{i(j)}^*) w_{i-1(m)}^{s,j} \right),$$

normalize them so that their sum equals unit, and obtain the weights  $q_{i(j)}$ ,  $j \in \overline{1, N}$ .

2. Calculate unnormalized weights

$$W_{i(m)}^{s,j} = f(\mathbf{m}_i^s | \mathbf{R}_{i(j)}^*, \mathbf{l}_{i-1(m)}^{s,j}) w_{i-1(m)}^{s,j},$$

normalize them, and obtain the weights  $w_{i-1(m)}^{*s,j}$ ,  $j \in \overline{1, N}$ .

3. Calculate  $N_{eff} = \left( \sum_{j=1}^N (q_{i(j)})^2 \right)^{-1}$ . If  $N_{eff} \leq \alpha N$ , where  $\alpha \in [0,1]$  is a

threshold specified by the user, we resample  $\mathbf{R}_{i(j)}$ ,  $j \in \overline{1, N}$  with a discrete density

$$f_N^i(\mathbf{R}_{i(j)} | \mathbf{m}_{ii}) = \sum_{j=1}^N q_{i(j)} \delta(\mathbf{R}_{i(j)} - \mathbf{R}_{i(j)}^*),$$

and put  $q_{i(j)} = N^{-1}$ ,  $j \in \overline{1, N}$ . Together with the particles  $\mathbf{R}_{i(j)}$  the

associated sets  $\{\mathbf{l}_{i-1(m)}^{s,j}\}_{m=1}^{N_1}$  with weights  $w_{i-1(m)}^{*s,j}$ , are resampled; in this

way, they form a new family of sets  $\{\mathbf{l}_{i(m)}^{s,j}\}_{m=1}^{N_1}$ ,  $j \in \overline{1, N}$ , with weights

$w_{i(m)}^{s,j}$ . And if  $N_{eff} \leq \alpha N$ , the resampling is not performed and we put

$$w_{i(m)}^{s,j} = w_{i-1(m)}^{*s,j}.$$

4. For each  $\mathbf{R}_{i(j)}$ ,  $j \in \overline{1, N}$ , we independently model vectors  $\mathbf{l}_i^{s(j)}$ ,  $s \in \overline{1, N_L}$ , with discrete densities

$$f_{N_1}^i(\mathbf{l}_i^{s(j)}) = \sum_{m=1}^{N_1} \delta(\mathbf{l}_i^{s(j)} - \mathbf{l}_{i(m)}^{s,j}) w_{i(m)}^{s,j}$$

in this way, we obtain a cloud of points of the form  $\{(\mathbf{R}_{i(j)}, \mathbf{l}_i^{(j)})\}_{j=1}^N$ ,

$\mathbf{l}_i^{(j)} = \{\mathbf{l}_i^{s(j)}\}_{s=1}^{N_L}$ , with weights  $q_{i(j)}$ ,  $j \in \overline{1, N}$ . The points  $\{(\mathbf{R}_{i(j)}, \mathbf{l}_i^{(j)})\}_{j=1}^N$  with weights  $q_{i(j)}$ ,  $j \in \overline{1, N}$ , are considered as a discrete approximation for the density  $f(\mathbf{R}_i, \mathbf{L} | \mathbf{m}_{1:i})$ . By calculating the different characteristics of these discrete distributions (mathematical expectations, and covariance matrices of different subvectors), we obtain approximate values of corresponding characteristics of the distributions with density  $f(\mathbf{R}_i, \mathbf{L} | \mathbf{m}_{1:i})$ . In particular,

$$E[(\mathbf{R}_i, \mathbf{L}) | \mathbf{m}_{1:i}] = \sum_{j=1}^N (\mathbf{R}_{i(j)}, \mathbf{l}_i^{(j)}) q_{i(j)}$$

5. Assign  $i = i + 1$  and go to step 1.

As compared to known algorithms for the filtration of vectors of the form  $(\mathbf{R}_i, \mathbf{L})$ , the novelty is modeling the uncertainty of the parameter  $\mathbf{L}$  by use of conditional particle filters. This is efficient when measurements are nonlinear in the parameter and the a priori uncertainty in the parameter is too large to apply the GKF. Besides, all numerous achievements that appeared in the field of particle filters during the past two decades can be used both for the main filter and for conditional filters. For conditional filters, one should apply tools intended for the case of a constant vector.

## SIMULATION

We consider the discrete model of circular motion along the circle  $x^2 + y^2 = 1$ . The robot starts at the point  $(1, 0)$  with the step length  $\rho_i = 2\pi/36 = 0,174$  and angles  $\Delta\theta_i = 10^\circ$ . Thus if the noises were absent the robot would arrive to the start point in 36 steps. However, the system noises processes  $e_i^p$  and  $e_j^o$  make the robot change its path. Our aim is to estimate the state-vector after 32 steps, that is, when 8/9 of the whole circle is completed. In our experiments, all the linear sizes are expressed in relative numbers, so if other step length is needed one can just multiply by appropriate factor all of the model linear parameters. In our simulation, 5 landmarks (4 out of the circle and 1 within the circle) are modeled. For generating exterior landmarks the uniform distribution in the ring  $1.55 < \sqrt{x^2 + y^2} < 5$  is used and the interior landmarks are sampled from the uniform distribution in the circle  $\sqrt{x^2 + y^2} < 0.45$ . We have put  $u_{\min} = 0.5$ ;  $u_{\max} = 6$  therefore the distances all the modeled landmarks from any point belonging the robot's path are within  $[u_{\min}, u_{\max}]$ . For the set  $M$ , the four

different variants have been tried, namely,  $M_1 = \{0,1,2,\dots,32\}$ ,  $M_2 = \{0,2,4,\dots,32\}$ ,  $M_3 = \{0,4,8,\dots,32\}$  and  $M_4 = \{0,8,16,24,32\}$ . It has turned out that using  $M_1$  and  $M_2$  yields only weak increasing of the computational accuracy compared with taking measurements at the steps whose numbers are from  $M_3$ . At the same time,  $M_3$  has appeared to be better choice than  $M_4$ . Evidently, the method is less time consuming when  $M_3$  is used rather than  $M_2$  or  $M_1$ . Because of this, we have taken  $M = M_3$  in all the results following below. Further, increasing the  $N_1$  beyond 200 hasn't produced the significant accuracy improvement so  $N_1 = 200$  in all our examples. The parameters  $\sigma_i^\rho$ ,  $\sigma_i^\theta$ ,  $N$  has been varied and the algorithm has been performed many times at each set of these parameters' values. The standard deviation of the measurement noise  $\sigma_i^\alpha = 1.5^\circ$  for all the examples. The typical results at  $\sigma_i^\rho = 0.05\rho_i$ ,  $\sigma_i^\theta = 1^\circ$  and  $N = 200$  are depicted in Fig. 5 and Fig. 6. Note  $\sigma_i^\rho$  is expressed by the percent of  $\rho_i$  for simplicity.

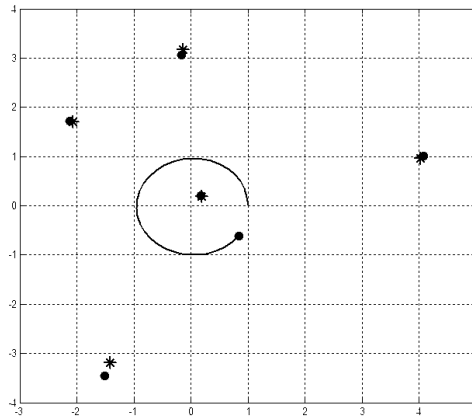


Figure 5. The point estimation: true trajectory of the robot, true landmarks' positions (stars), and point estimates at the 32 step (stars).

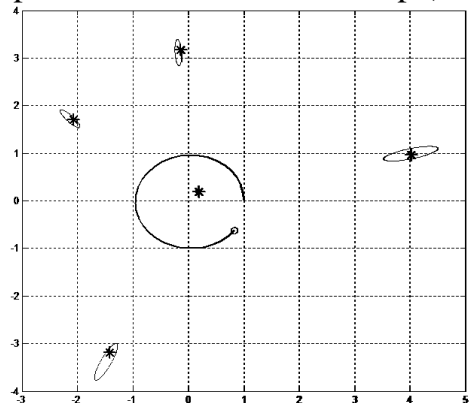


Figure 6. The confidence regions: the true trajectory of the robot, landmarks' positions (stars), and 95% confidence ellipses at the 32 step.

## CONCLUSIONS

Here is what we are planning in respect of the method proposed.

- Compare our method with the algorithm described in [1].
- Combine our method with a strategy of optimal choosing landmarks in case there are many ones.
- Generalize the proposed methods to 3D case.

## REFERENCES

- [1] M. H. Mirabdollah, B. Mertching. Monocular SLAM: Using trapezoids to model landmark uncertainties//2012 IEEE International Conference on Robotics and Biometrics, ROBIO 2012 – Conference Digest, art. no. 6491013, pp 482-488.
- [2] K.E.Bekris, M.Glick, L.E. Kavraki. Evaluation of Algorithms for bearing-only SLAM// [ICRA 2006](#): pp.1937-1943.
- [3] M Bryson, S Sukkarieh [Bearing-only SLAM for an airborne vehicle](#)- Australasian Conf. Robot. Autom.(ACRA 2005), Sydney , 2005.
- [4] S. Thrun, M. Montemerlo, D. Koller, B.Wegbreit, J. Nieto, and E. Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association, 2004. // <http://robots.stanford.edu/papers/Thrun03g.html>.
- [5] Kwok, N.M. Bearing Only SLAM using a SPRT Based Gaussian Sum Filter.//ICRA 2005, pp. 1109 – 1114.
- [6] Doucet A. Sequential Monte-Carlo methods in practice. /Doucet.A., Freitas N., Gordon N.– N.Y., Shpringer-Verlag, 2001.
- [7] B Ristic, S. Arulampalam, and N. Gordon. Beyond the Kalman Filter: Particle Filters for Tracking Applications. Artech House, 2004.
- [8] Stepanov O.A., Toropov A.B. Application of the Monte Carlo Methods with Partial Analytical Integration Techniques to the Problem of Navigation System Aiding. Proceeding of 20-th Saint Petersburg International Conference on Integrated Navigation Systems. Russia 2013, pp.299-301.
- [9] D.G. Arseniev, N.A. Berkovskii. Bearing-only SLAM: A fast algorithm without using Extended Kalman filter.//WSEAS Transactions on signal processing. Vol. 10, pp.199-204. (2014)
- [10] N. Bergman: Posterior Cramer-Rao Bounds for Sequential Estimation, in: A. Doucet, N.Freitas, N.Gordon. Sequential Monte-Carlo methods in practice. N.Y. Shpringer-Verlag, (2001).
- [11] Z. Long, N. Ruixin, P.K. Varsney: Conditional Posterior Cramer-Rao Lower Bounds for nonlinear recursive Filtering, 12<sup>th</sup> International Conference on Information fusion, Seattle, WA, USA, Julu 6-9, p.1528, (2009).