Appraisal of Different Particle Filter Resampling Schemes Effect in Robot Localization

Imbaby I. Mahmoud[1], Asmaa Abd El Tawab[2], May Salama[3] and Howida A. Abd El-Halym[4]

[1]*Nuclear Research Center, Atomic Energy Authority,Imbabyisma@yahoo.com*
[2]*Nuclear Research Center, Atomic Energy Authority,asm28aa@yahoo.com*
[3]*Benha University, Faculty of Eng., Shoubra , Eletronic and Comm. Dept.,msalama@megacom-int.com*
[4]*Nuclear Research Center, Atomic Energy Authority,howidaaa@yahoo.com*

ABSTRACT

This paper considers the effect of the Resampling schemes in the behavior of Particle Filter (PF) based robot localizer. The investigated schemes are Multinomial Resampling, Residual Resampling, Residual Systematic Resampling, Stratified Resampling and Systematic Resampling. An algorithm is built in Matlab environment to host these schemes. The performances are evaluated in terms of computational complexity and error from ground truth and the results are reported. The results showed that the localization plan which adopts the Systematic or Stratified Resampling scheme achieves higher accuracy localization while decreasing consumed computational time. However, the difference is not significant. Moreover, a particle excitation strategy is proposed. This strategy achieved significant improvement in the behavior of PF based robot localization.

Keywords: *Particle Filter, Robot Localization, Resampling*

I. INTRODUCTION

The problem of robot localization consists of answering the question Where am I from a robot's point of view. This means the robot has to find out its location relative to the environment. When we talk about location, pose, or position we mean the x and y coordinates and heading direction of a robot in a global coordinate system.

The mobile robot localization problem comes in many different flavors .The most simple localization problem is position tracking while the initial robot pose is known, and the problem is to compensate incremental errors in a robot's odometry. More challenging is the global localization problem [1], where a robot is not told its initial pose but instead has to determine it from scratch.

Several methods are employed to deal with robot localization problem [2, 3] Recently Particle Filter (PF) becomes dominant approach used for solving this problem. This is due to its ability to handle non-linear non-Gaussian problem, typical characteristic of localization problem [4, 5]. Several implementations of PF are reported [6- 8].

Resampling approaches of PF are introduced to overcome the degeneracy problem of original PF [9], which proved to be the bottleneck of whole PF algorithm. In this paper, the effect of different resampling approaches in the behavior of PF based robot localization problem is investigated. Resampling approaches namely : Multinomial Resampling (Mult R), Systematic Resampling (SR), Residual Resampling (RR), Residual Systematic Resampling (RSR) and Stratified Resampling (STR) are hosted in a PF algorithm which is

implemented in Matlab environment and their performance are evaluated in terms of computational complexity and amount of error from ground truth and the results are reported.

This paper is organized as follows: section2 presents overview of the resampling approaches and their implementation algorithms, section 3 studied the effect in Robot Localization by using different resampling approaches, section 4 shows the discussion of the obtained results, section 5 is devoted to conclusion.

II. OVERVIEW OF RESAMPLING APPROACHES

A. Multinomial Resampling Approach

The simplest approach to resampling is based on an idea at the core of what is called bootstrap method [10].

This approach works as follows: Firstly N sorted random numbers uniformly distributed in [0, 1] are selected. Then the cumulative sums of the particle weights are calculated. Finally, the number of the sorted random numbers that appear in each interval of the cumulative sum represents the number of copies of this particular particle which are going to be propagated to the next generation. Intuitively, if a particle has a large weight then many random numbers are going to survive.

For comparison purposes, we estimate variance for each approach The resampling from a multinomial distribution possesses a variance namely $\text{var}(N_i) = N\widetilde{\omega}_t^{(i)}\left(1 - \widetilde{\omega}_t^{(i)}\right)$.where $\widetilde{\omega}_t^{(i)}$ is the weight of particle i=1,..., N at certain me t [11, 12].

B. Residual Resampling

This procedure involves the following steps. Firstly, set $\widetilde{N}_i = \left\lfloor N\widetilde{\omega}_t^{(i)} \right\rfloor$

Where $\left\lfloor \ \ \right\rfloor$ denotes the integer part.

Secondly, perform a multinomial resampling procedure to select the remaining $\overline{N}_t = N - \sum_{i-1}^{N} \widetilde{N}_i$ samples with new weights $\omega_t'^{(i)} = \overline{N}_t^{-1}\left(\widetilde{\omega}_t^i N - \widetilde{N}_i\right)$. Finally, add the results to the current $\widetilde{N}_i$. For this scheme, the variance can be estimated from the following relation $\left(\text{var}(Ni) = \overline{N}_t \omega_t'^i \left(1 - \omega_t'^i\right)\right)$[9]. We notice that it is smaller than the one given by the multinomial approach. From computational point of view this procedure consumes less time. This algorithm can be found in [13].

C. Systematic Resampling

The SR consists of two loops, the first loop of length *N* to calculate the replication factor of the *N* particles, and the second loop with variable length that depends on the weight of the particles. *N* is the input number of particles, *M* is the number of particles generated after resampling, and *w* is an array of normalized weights from the importance step. The output *r* is an array of replication factors showing how many times each particle is replicated. This algorithm can be found in [13].

The SR is simple to implement and its computational complexity is O(M).This strategy introduces a variance on number of children even smaller than the residual resampling scheme**.**

D. Residual Systematic Rresampling

It is a resampling approach [14] which is based on stratified resampling, and is referred to it as RSR.

In the RSR algorithm, firstly the number of replication factors of the particles is calculated. Because this method does not guarantee that the number of resampled particles is *M*, the residual number of particles *Mr* is

computed. Secondly, resampling of residuals is performed by SR with N input and Mr output particles. To apply SR, the residual weights $w_t^{m^*}$ are normalized and the new weights w_t^m are obtained for $m=1, 2,\ldots, M$. Finally, the produced replication factors are summed. Since Ref. [14] did not formulate algorithm, we state it as follows:

Output	An array of replication factors $\{r^m\}_{m=1}^M$ at time instant t, t>0 using RR.
Input	An array of $\{w_t^m\}_{m=1}^N$ input and output number of particles, N and M.
Algorithm	(r)=RR(w, N, M)

- $Mr=M$ // Initializing the number of particles left for resampling
- For $m=1: N$ // The First step to compute the replications

$$r^m = \lfloor w^m \rfloor M$$

$$w_r^{m^*} = w^m . M - r^m$$

// The residues of the weights. $$M_r = M_r - r^m$$

- End For
- If $Mr > 0$ // The Second step to process residuals.

  - calculate total weight $Wr = sum[\{w_t^{i^*}\}_{i-1}^M]$

    - For $m = 1: N$

$$w_r^m = w_r^{m^*} / W_r$$

    - End For
  (rr)= SR (w_r, N, Mr)   // SR with residual weights.
 - For $m = 1: M$ // Updating the replication factors.

$$r^m = r^m + r_r^m$$

  - End If
- End If

E. Stratified Rresampling

Stratified resampling [15] is based on ideas used in survey sampling and consists in pre-partitioning the (0, 1] interval into N disjoint sets, (0, 1] = (0, 1/N] U $\cdots$ U ({N − 1}/N, 1]. The U^i 's are then drawn independently in each of these sub-intervals: $U^i \sim$ U (({i − 1} /N, i/N]), where U ([a, b]) denotes the uniform distribution on the interval [a, b]. This strategy introduces a variance on resampling, $var\ (N_i) = \overline{N}_t \omega_t'^{(i)}\left(1 - \overline{N}_t \omega_t'^{(i)}\right)$. In the following this method is formulated as the algorithm and given in pseudo-code form:

Output	An array of replication factors $\{r^m\}_{m=1}^M$ at time instant t, $t>0$ using STR.
Input	An array of $\{w_t^m\}_{m=1}^N$ input and output number of particles, N and M, respectively.
Algorithm	(r)=ST (w, N, M)

-aux=(rand(1,N))./N;   // Generate random number array
  i =(0:1/N:(1-1/N));
  U=aux+i;
-j=1
- S=w(j)   // Initializing the cumulative sum of weights
- For $m=1: N$ // The main loop
    - K=0   // The replication factor counter
   -While (S < U(m))   // The resampling loop
      - j= j+1
      - S=S + w (j)   // Updating the CSW

- End While	
- *k=k+1*	
- *R(m)=K*	// Storing the replication factors
- End For	

III. IMPLEMENTATION OF ROBOT LOCALIZATION ALGORITHM USING DIFFERENT RESAMPLING APPROACHES

We study a localization system which employs radio beacons that provide the ability to measure range only [16]. Obtaining range from radio beacons has the advantage that line of sight between the beacons and the transponder is not required, and the data association problem can be completely avoided. In this work six radio beacons are distributed over a 30*30 meter area and robot is programmed to drive in a repeating path. A particle filter method is used to fuse range data with dead reckoning data collected from a real system which integrates proprioceptive measurements from wheel encoders, gyros, and accelerometers to localize the robot. Matlab environment is used for experimenting with localization process. A program for particle filter algorithm is written to host different resampling approaches.
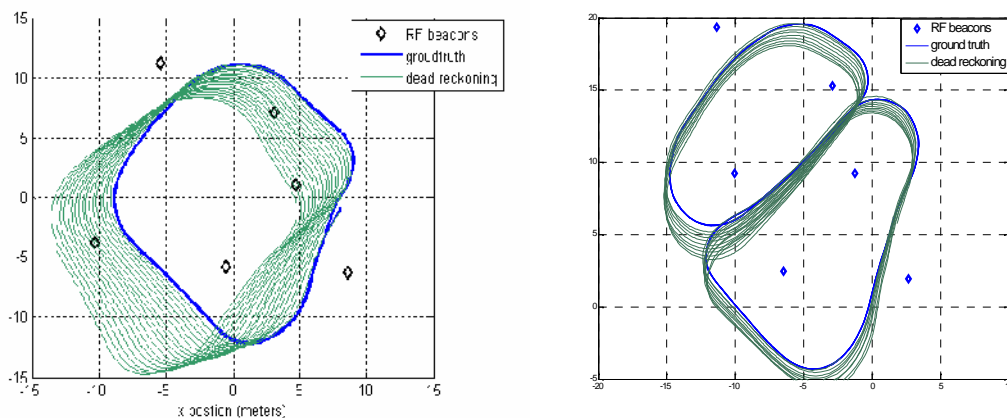

Fig. 1: (a) Circular path (b) Figure-8 path data: ground truth and dead reckoning

Fig [1] shows the dead reckoning path, ground truth path and tag locations for the (a) circular path dataset and (b) figure-8 path dataset. We notice from this figure that the dead reckoning tends to drift away from the true path over time. This is due to increasing errors in odometry.

A. The results of the resampling approaches in a circular path and figure-8 path

In the implemented Matlab experiments, the particle filter requires no initial estimate of the robot's position. In these experiments, the first 5000 step (about 20 minute) of the robot's travel are clipped from results plot, giving the filter time to converge. Figs (2-6) show the results of studied resampling approaches.

Table [1] & [2] summarizes the results of these figures concerning the error in the estimates of the studied resampling approaches.

Since the difference in error results as indicated in Table [1] & [2] is not significant, a particle excitation strategy is experimented. It is based on adding a Gaussian perturbation around each particle position. Table [3] shows the average error for the studied resampling approaches in case of excitation of the particle.
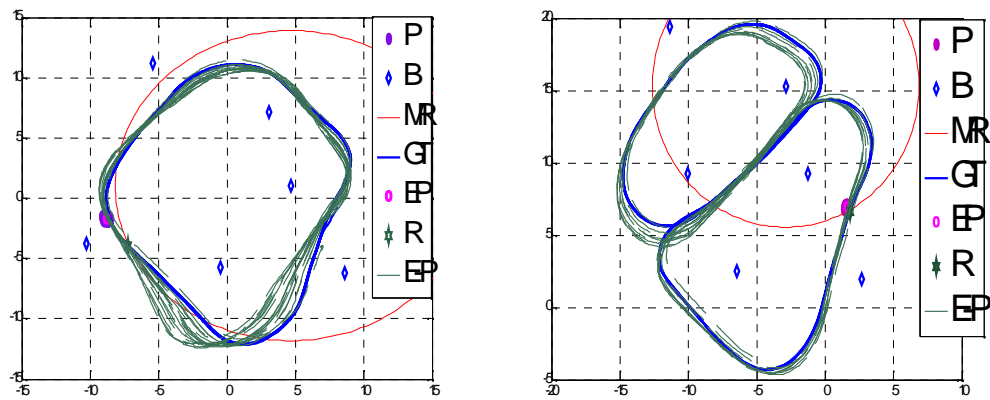
Fig. 2: systematic resampling, where the acronyms P, B, MR, GT, EP, R, and E-P denotes particles, beacons, measured range, ground truth, estimated pose, Robot and estimated path respectively.
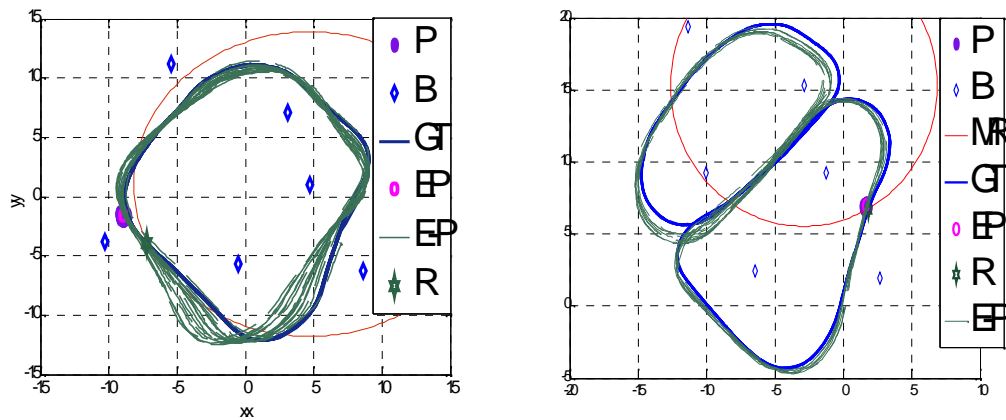

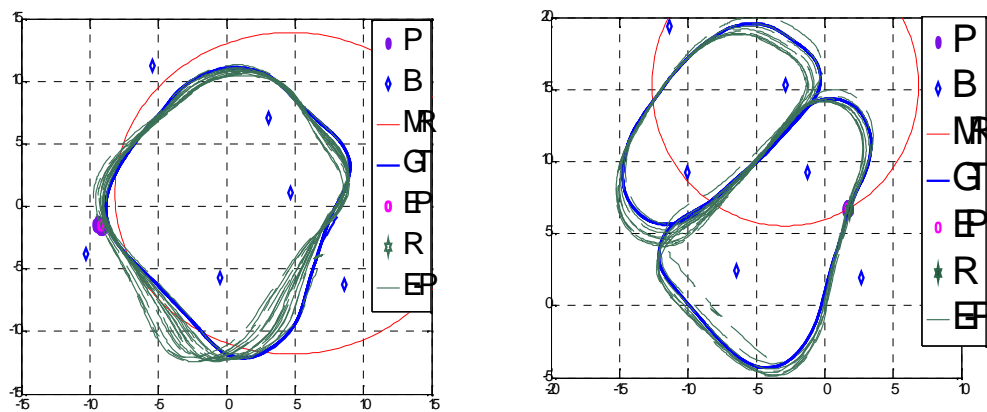Fig. 3: Residual systematic resampling
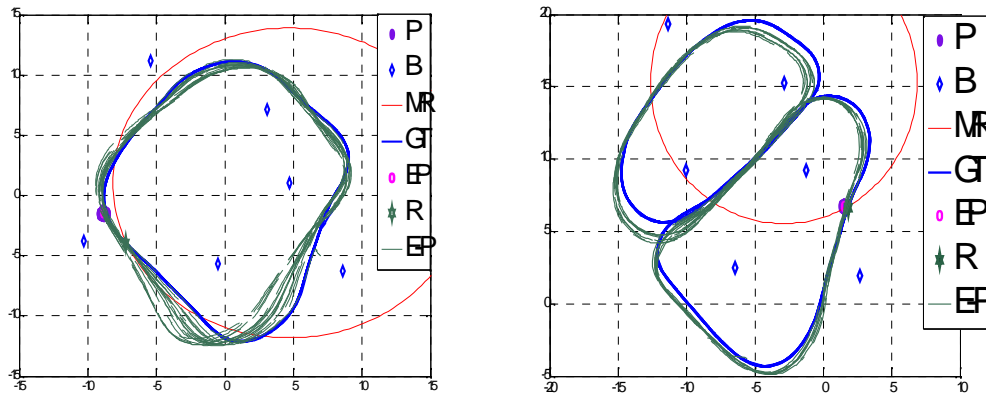

Fig. 4: Stratified resampling
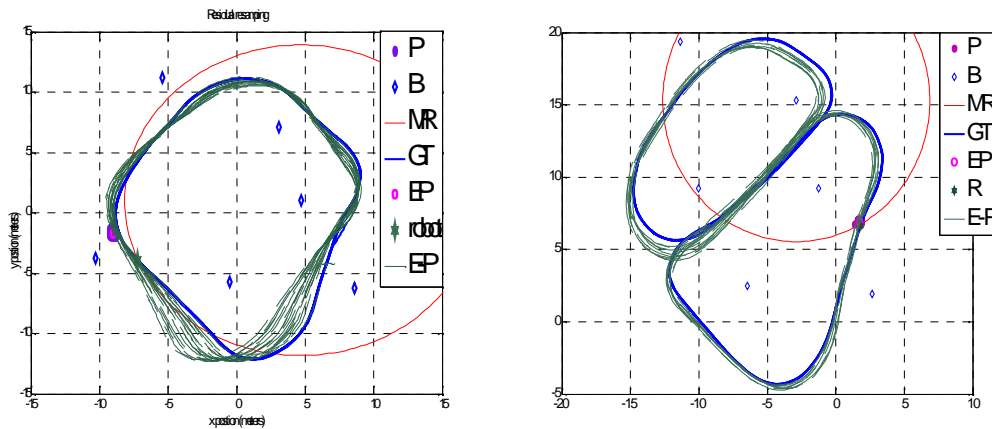
Fig. 5: Multinomial resampling

Fig. 6: Residual resampling

Table 1: Results of Error Calculation Using Different Resampling Approaches in PF in a Circular Path Data.

Error/in meter	Dead reckoning	RSR	Stratified	Residual	Systematic	Multinomial
XTE_abs_avg	1.9114	0.487	0.4917	0.4646	0.4591	0.462
XTE_abs_max	6.5952	2.9215	3.2119	3.0187	2.818	2.7637
XTE_abs_std	1.5820	0.5078	0.5338	0.5139	0.4885	0.4751
ATE_abs_avg	1.6226	1.3183	1.3146	1.3211	1.3203	1.3079
ATE_abs_max	6.8282	4.4358	4.7448	4.5695	4.333	4.2072
ATE_abs_std	1.5014	0.75	0.7749	0.7644	0.7351	0.7551
Cartesian_abs_avg	2.8043	1.4482	1.446	1.4423	1.4399	1.429
Cartesian_abs_max	8.9253	4.4398	4.7491	4.5702	4.336	4.2229
Cartesian_abs_std	2.0035	0.8356	0.8743	0.854	0.8121	0.8233

XTE:Cross Track Error,How far left or right of the true position our estimation is,Orthogonal to the true heading,ATE: Along Track Error, Tangential component of the position error ,Cartesian error: Total Euclidean distance error /in meter .

Table 2: Results of Error Calculation Using Different Resampling Approaches in PF in a Figure-8 Path Data

Error in meter	Deadreckoning	RSR	Stratified	Residual	Systematic	Multinomial
XTE_abs_avg	0.5223	0.363	0.3521	0.3638	0.351	0.4083
XTE_abs_max	2.3385	1.364	1.4609	1.4365	1.4886	1.4879
XTE_abs_std	0.4417	0.3035	0.3109	0.3106	0.3008	0.3222

ATE_abs_avg	0.7538	0.6339	0.6213	0.644	0.625	0.6371
ATE_abs_max	2.6212	1.778	1.8228	1.7768	1.8795	2.0892
ATE_abs_std	0.5447	0.4163	0.4458	0.4389	0.413	0.4626
Cartesian_abs_avg	0.9997	0.8081	0.7934	0.8157	0.7902	0.8483
Cartesian_abs_max	2.6284	1.7791	1.828	1.781	1.8831	2.09
Cartesian_abs_std	0.5775	0.4237	0.3943	0.4132	0.3877	0.4133

Table 3: Resampling Approaches Results in Case of Excitation of the Particle in a Circular Path

Error/in meter	Dead reckoning	RSR	Stratified	Residual	Systematic	Multinomial
XTE_abs_avg	1.9114	0.4123	0.4506	0.4502	0.4264	0.5146
ATE_abs_avg	1.6226	1.0681	1.0813	1.0871	0.7443	1.1733
Cartesian_abs_avg	2.8043	1.1789	1.2118	1.2214	0.9194	1.3325

Resampling is always necessary in order to make more effective use of a finite sample set. As a consequence, sample impoverishment is unavoidable. Increasing the sample set diversity is helpful to identify new possible modes and prevent some samples from dying out too early [17]. Therefore we assumed each sample is representing a Gaussian kernel. The process of generating samples from Gaussian kernel called it excitation strategy. The samples generated from the kernel may span the state space more evenly, which helps find some new potentially good samples. Regardless those samples from excitation process are not theoretically guaranteed to approximate the posterior asymptotically, they do better concerning error values than PF when sample impoverishment is severe. Fig [8] demonstrates these results.
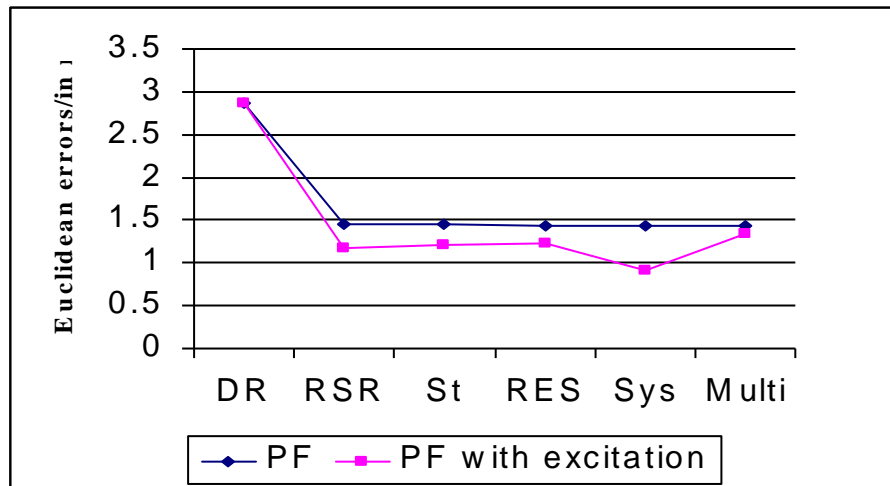

Fig. 7: Comparison between resampling approaches in PF and PF with excitation

IV. DISCUSSION OF THE RESULTS

From Table [1] & [2], we notice the slight difference in calculated error among different resampling approaches. This is agreed well with theoretical prediction based on variance [18]. Also, this can be contributed to the fact that all approaches have the same basic strategy. From table [2], the effect of excitation strategy is very prominent. Even using the excitation strategy enhanced the effect of resampling approaches as in the case of systematic resampling.

Considering computational complexity and time consumed in a Matlab run, systematic resampling is favorable. Fig [8] shows the time consumed by each resampling approach as a function of the number of particles. Systematic resampling shows the least consumed time compared with all other studied resampling schemes.
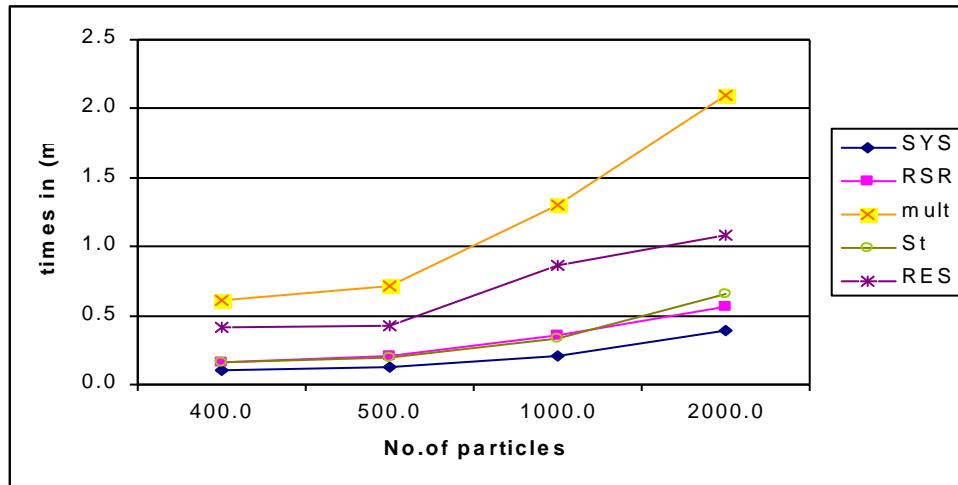
Fig. 8: Time consumed for different resampling approaches as a function of the number of particles

V. CONCLUSION

The paper presents a study for the effect of resampling approaches in the behavior of P.F based robot localizer. A P.F algorithm is implemented in Matlab environment to host different resampling strategies. Different resampling approaches namely Multinomial Resampling, Residual Resampling, Residual Systematic Resampling, Stratified Resampling and Systematic Resampling are investigated. Quantitative evaluation of these approaches is determined in terms of the error calculated between the ground truth and localized position, and time complexity. Results showed that the localization plan which adopts the Systematic or Stratified Resampling scheme achieves higher accuracy localization while decreasing consumed computational time. Moreover, an excitation strategy for particles is also implemented. The effect of excitation strategy shows significant improvement of the P.F algorithm independent of resampling approaches. However, Systematic Resampling shows high improvement with the excitation strategy introduced.

References

[1] D. Fox, W. Burgard, and S. Thrun. " Markov localization for mobile robots in dynamic environments." *Journal of Artificial Intelligence Research*, 11:391–427,1999.

[2] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. "Monte carlo localization: Efficient position estimation for mobile robots." In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Orlando, FL, 1999.

[3] Ioannis M.Rekleitis "A particle filter tutorial for mobile robot localization" Technical Report TR-CIM-04-02,McGill University, Canada H3A 2A7,2004.

[4] James Carpenter and Peter Clifford and Paul Fearnhead "An Improved Particle Filter for Non-linear Problems" Department of Statistics, University of Oxford, 1999.

[5] Simon Maskell, Neil Gordon, and Tim Clapp "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking "M. Sanjeev Arulampalam, IEEE Transactions on Signal Processing, Vol. 50, NO. 2,FEBRUARY 2002.

[6] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. "Robust Monte Carlo localization for mobile robots" Artificial Intelligence, pp. 99-141, 2000.

[7] J. Liu and R. Chen. "Sequential Monte Carlo methods for dynamic systems"Journal of American Statistical Association, 93, 1998.

[8] A. Doucet, N. de Freitas, and N. Gordon, "Sequential Monte Carlo Methods in Practice", Springer, 2001.

[9] Rudolph van der Merwe, Nando de Freitas, Arnaud Doucet, Eric Wan "The Unscented Particle Filter " Cambridge University ,August 16, 2000

[10] B. Efron and R. J. Tibshirani. "An Introduction to the Bootstrap" Chapman & Hall, 1993.

[11] Carpenter, J., Clifford, P. and Fearnhead, "Building robust simulation-based filters for evolving data sets" Department of Statistics, Oxford University P (1999).

[12] Liu, J. S. and Chen, R. "Sequential Monte Carlo methods for dynamic systems" Journal of the American Statistical Association vol.93 (1998).

[13] Howida Abd El-Latif Abd El-Halym "Particle Filters For Object Tracking: Enhanced Algorithm and Efficient Implementations " Ph.D Thesis, Cairo University,2010.

[14] M. Boli´c, P. M. Djuri´c, S. Hong, "New Resampling Algorithms for ParticleFilters", IEEE Proceedings of ICASSP, Hong Kong, 2003.

[15] P. Fearnhead. "Sequential *Monte Carlo methods in filtter theory*.PhD thesis" University of Oxford, 1998.

[16] Derek Kurth "Range-Only Robot Localization and SLAM with Radio" Robotics Institute, Carnegie Mellon Univ., tech. report CMU-RI-TR-04-29 May 2004.

[17] Jeroen D.Hol,Thomas B. Schon, Fredrik Gustafsson "On Resampling Algorithms For Particle Filters" Cambridge, United Kingdom,2006.

[18] Ping Wang, James M. Rehg, "A Modular Approach to the Analysis and Evaluation of Particle Filters for Figure Tracking," IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06), 2006.