VOL. 18, NO. 3-2, 2019, 17-24
www.elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

Enhanced Localization with Adaptive Normal Distribution Transform Monte Carlo Localization for Map Based Navigation Robot

T.Y. Lim[1], **C. F. Yeong**[2*], **E. L. M. Su**[3], **S.M. Shithil**[2], **S.F. Chik**[3], **F. Duan**[4] and **P.J.H. Chin**[5]

[1]Malaysia Japan Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Johor Bahru
[2]Centre for Artificial Intelligence and Robotics, Universiti Teknologi Malaysia, Johor Bahru
[3]School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru
[4]Department of Automation, College of Computer and Control Engineering, Nankai University, Tianjin, China
[5]DF Automation and Robotics Sdn. Bhd, Johor Bahru, Malaysia

[*]Corresponding author: cfyeong@utm.my, Tel: 607-5557163

Abstract: Map-based navigation is the common navigation method used among the mobile robotic application. The localization plays an important role in the navigation where it estimates the robot position in an environment. Monte Carlo Localization (MCL) is found as the widely used estimation algorithm due to it non-linear characteristic. There are classifications of MCL such as Adaptive MCL (AMCL), Normal Distribution Transform MCL (NDT-MCL) which can perform better than the MCL. However, AMCL is adaptive to particles but the position estimation accuracy is not optimized. NDT-MCL has good position estimation but it requires higher number of particles which results in higher computational effort. The objective of the research is to design and develop a localization algorithm which can achieve better performance in term of position estimation and computational effort. The new MCL algorithm which is named as Adaptive Normal Distribution Transform Monte Carlo Localization (ANDT-MCL) is then designed and developed. It integrates Kullback–Leibler divergence, Normal Distribution Transform and Systematic Resampling into the algorithm. Three experiments are conducted to evaluate the performance of proposed ANDT-MCL in simulated environment. These experiments include evaluating the performance of ANDT-MCL with different path shape, distance and velocity. In the end of the research work, the proposed ANDT-MCL is successfully developed. It is adaptive to the number of particles used, higher position estimation and lower computational effort than existing algorithms. The algorithm can produce better position estimation with less computational effort in any kind paths and is consistent in long journey as well as can outperform in high speed navigation.

Keywords: Localization, Map based navigation, MCL, NDT-MCL, AMCL

Article History: received 4 September 2019; accepted 1 December 2019; published 24 December 2019.

1.INTRODUCTION

Localization systems are an essential enabling component of mobile robotic systems [1]. The localization plays a significant role in the navigation where it estimates the position of the robot in the environment. Over the years, there are many localization algorithms being researched intensively in order to efficiently and effectively combine both of the feedbacks to provide good robot position feedback to the navigation system. Those widely researched algorithms are Extended Kalman Filter (EKF)[2], Unscented Kalman Filter[3,4] (UKF) and Particle Filter (PF), also known as Monte Carlo Localization (MCL). MCL is found as the widely used estimation algorithm for robot localization due to its non-linear estimation characteristic[3,4,5].The MCL algorithm uses particles to predict the state of the robot such as position and orientation when it moves and senses the environment [6]. There are variants of MCL being innovated in order to improve the limitations existed in MCL for example computational time and position estimation accuracy.

Adaptive MCL (AMCL) has been proposed by Dieter Fox [7] to improve the computational time of MCL. AMCL is a robot localization algorithm for navigating in 2D environment. It, which is also utilizes the Kullback-Lknown aeibler Divergence s KLD-sampling in the Monte Carlo localization to estimate the state of a robot in a known map. With the KLD sampling, it is able to adapt the number of particles needed for MCL rather than fixed number of particles when moving in the environment[8,9]. Besides, another variant of MCL is also proposed which is Normal Distribution Transform Monte Carlo Localization (NDT-MCL). It is a piecewise continuous representation which also uses the particle filter to estimate the state of a robot in a known map by representing the space as a set of normal distributions[10]. With this representation, it improves the position estimation of MCL.

In this paper we proposed to design and develop a

localization algorithm which can achieve better performance in term of position estimation and computational effort. The new MCL algorithm which is named as Adaptive Normal Distribution Transform Monte Carlo Localization (ANDT-MCL) is then designed and developed. It integrates Kullback–Leibler divergence, Normal Distribution Transform and Systematic Resampling into the algorithm.

Three experiments are conducted to evaluate the performance of proposed ANDT-MCL in simulated environment. These experiments include evaluating the performance of ANDT-MCL with different path shape, distance and velocity.

2. DESIGN AND DEVELOPMENT

MCL produces insufficient position estimation accuracy[12]. One of the reasons of the poor estimation performance is due to the grid-based representation used by MCL. The occupancy grid map representation used in the MCL are discretized into fixed size may include sensor noises which then cannot represent the environment accurately. Besides, MCL uses the particles to represent the likely position of the robot in the environment. The greater the number of particles, the more accurate the position estimation. However, it can result in heavy computational time. The basic MCL algorithm overview is shown in Figure 1. Hence, a piecewise continuous static NDT map is proposed to use in ANDT-MCL measurement update to replace the use of occupancy grid map. ANDT-MCL also includes NDT[18] in the particles weightage update during the measurement update as show in the Figure 2 from step 3 to step 8. Whereas, the KLD[71] has the behavior of number of particles adaption which is proposed and implemented together during the resampling stage as shown in Figure 2 from step 14 to step 20. An overview of a time-update ANDT-MCL algorithm is outlined in Figure 2 and the design of ANDT-MCL modifies the MCL based on the Figure 1.

1. **Inputs:** $S_{t-1} = \{\langle x_{t-1}^k, w_{t-1}^k \rangle \mid k = 1, \dots, n_{t-1}\}$, control measurement u_{t-1}, observation z_t, map m
2. $S_t = \emptyset, \alpha = 0, n = N$
3. **for** $k := 1, \dots, n_{t-1}$ **do**
   // Sampling: Predict next state (Eq. 4.3)
4. Sample x_t^k from $p(x_t|x_{t-1}, u_{t-1})$ using x_{t-1} and u_{t-1}
5. $w_t^k := p(z_t \mid x_t^k, m)$ // Compute each particle weight
6. $\alpha := \alpha + w_t^k$ // Update normalization factor
7. **for** $k := 1, \dots, n_{t-1}$ **do**
8. $w_t^k := w_t^k / \alpha$ // Normalize weights
9. **for** $k := 1, \dots, n_{t-1}$ **do**
   // Resampling: Draw state from current distribution
10. Resample new particle x_t^k with probability $\propto w_t^k$
11. $S_t := S_t \cup \{\langle x_t^k, w_t^k \rangle\}$ // Insert sample into sample set
12. Compute current pose $\bar{x}_t$ by using S_t // Obtain current robot position estimation
13. **return** S_t

Figure 1. Basic MCL algorithm overview

At each iteration, the algorithm takes the previous sample set S_{t-1} containing data position states x_{t-1}, weightages w_{t-1}, previous number of samples n_{t-1} as the inputs. Besides, the inputs also contain current observations z_t. Several parameters are initialized as shown in step 2. Step 3 is to transform the current observations z_t to a set of normal distribution parameters $\bar{z}_t := \{u_i, \sum_i\}_{i=1}^{N_{zt}}$ with regular cell size. Step 5 is the prediction phase (or sampling phase) in which each sample updates its state using motion model. Step 6 shows the likelihood model of $\bar{z}_t$ from x_t^k and m is computed and is denoted as L_2^k. The weightage w_t of each sample is calculated based on the L_2^k and previous weightage, w_{t-1} as shown in step 7. The normalization factor, α is updated after that. With the normalization factor, the weight w_t is normalized to value 1.0. The next step is resampling where the samples are being resampled based on important weight. The new samples obtained are put into the sample set S_t in step 12. For each new sample inserted into sample set S_t, it is checked if it falls into an empty bin. The number of supported bins k is increased by one and the current bin is marked non-empty. The Equation 4.10 is then used to update the number $\hat{n}$ of samples required for the current estimate of k. The additional step as shown in step 17 is to check the minimum number of samples $\hat{n}_{min}$ has been passed by (default: 10).

1. **Inputs:** $S_{t-1} = \{\langle x_{t-1}^k, w_{t-1}^k \rangle \mid k = 1, \dots, n_{t-1}\}$, control measurement u_{t-1}, observation z_t, map m, bounds ε and δ, bin size Δ, minimum number of samples $\hat{n}_{min}$
2. $S_t = \emptyset, n = N, \hat{n} = 0, k = 0, \alpha = 0$
3. $\bar{z}_t := \{u_i, \sum_i\}_{i=1}^{N_{zt}}$ from z_t // Compute NDT (Eq. 4.4 & 4.5)
4. **for** $k := 1, \dots, n_{t-1}$ **do**
   // Sampling: Predict next state (Eq. 3.3)
5. Sample x_t^k from $p(x_t|x_{t-1}, u_{t-1})$ using x_{t-1} and u_{t-1}
6. $L_2^k := L_2^k(\bar{z}_t \mid x_t^k, m)$ // Compute likelihood (Eq. 4.6)
7. $w_t^k := w_{t-1}^k L_2^k$ // Compute each particle weight
8. $\alpha := \alpha + w_t^k$ // Update normalization factor
9. **for** $k := 1, \dots, n_{t-1}$ **do**
10. $w_t^k := w_t^k / \alpha$ // Normalize weights
11. **do**
    // Resampling: Draw state from current distribution
12. Resample new particle x_t^n with probability $\propto w_t^n$
13. $S_t := S_t \cup \{\langle x_t^n, w_t^n \rangle\}$ // Insert selected sample into sample set
14. **if** (x_t^n falls into empty bin b) **then**
15. $k := k + 1$ // Update number of supported bins
16. $b :=$ non-empty // Mark bin
17. **if** $n \geq \hat{n}_{min}$ **then**
    // Update number of desired samples (Eq. 4.10)
18. $\hat{n} := \frac{k-1}{2\varepsilon}\left\{1 - \frac{2}{9(k-1)} + \sqrt{\frac{2}{9(k-1)}}\, z_{1-\delta}\right\}^3$
19. $n := n + 1$ // Update number of n samples
20. **while** ($n < \hat{n}$ and $n < \hat{n}_{min}$) // Until KL-bound limit
20. Compute current pose $\bar{x}_t$ by using S_t // Obtain current robot position estimation
21. **return** S_t

Figure 2. ANDT-MCL algorithm overview

In overall, the value of number samples n and the desired number of samples $\hat{n}$ will change over time. In the beginning of sampling, k value increases with about every new sample inserted into sample set S_t because initially all bins are empty. When the k value increases, the value of the number of desired samples $\hat{n}$ increases as well. However, when more and more bins become non-empty over time, the $\hat{n}$ increases only occasionally. According to the algorithm, n increases every time when each new sample is inserted, so it will cause the sampling to stop as shown in step 20 when n eventually reach $\hat{n}$. After all of this, the last step is to compute the current pose $\bar{x}_t$ by using S_t.

KLD number of particles adaption is integrated during the resampling stage of ANDT-MCL as shown in Figure 2 from step 14 to step 20. It is also found out that the typical resampling method can be improved together with KLD in the ANDT-MCL as shown in step 12 in Figure 2 to improve the position estimation. The proposed resampling

method is called SR resampling. The pseudocode of the SR resampling is shown in Figure 3.

```
    xx[] = SR(x, w, N)
1.  j = 0, accumW = w[j]
2.  u = rand()/N
3.  for i = 0..(N − 1)
4.      while accumW < u
5.          j = j + 1
6.          accumW = accumW + w[j]
7.      end
8.      xx[i] = x[j]
9.      u = u + 1/N
10. end
```

Figure 3. Systematic resampling pseudocode

The resampled particles are stored in the new xx sample set which are then undergo the KLD process. x is the robot pose particle in the initial sample set which will undergo resampling algorithm. w is the weighting of each of the samples(particle) in the sample set[16]. N is the total number of samples(particles) needed.

3. METHODOLOGY

In experiment 1 to 3, the proposed algorithm of ANDT-MCL is validated in different condition which are in different kind of paths, different path lengths and different speeds. The performance of ANDT-MCL is compared with NDT-MCL and AMCL by using the Gazebo simulated environment which is built for the experiment as shown in Figure 4

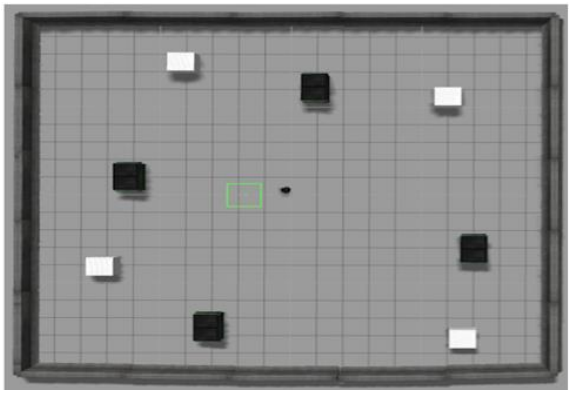


Figure 4. The simulated environment built for the experiment

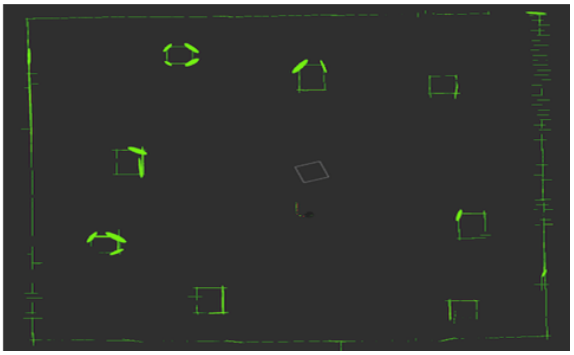


Figure 5. The NDT static map built by NDT- Mapping

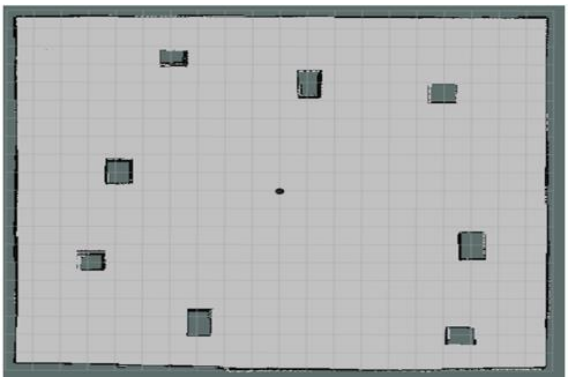


Figure 6. The occupancy static grid map built by G-Mapping

The static NDT-Mapping algorithm is pre-run by the robot to obtain a NDT static map which is used by ANDT-MCL and NDT-MCL when robot is navigating in the environment. The G-Mapping algorithm is also pre-run to obtain an occupancy static grid map which is used by AMCL algorithm. The outputs of the maps are showed in Figure 5 and Figure 6. The setting used by each of the algorithm are similar and are shown in Table 1. In order to carry out the localization algorithm, initially the Gazebo with the simulated environment (refer Figure 4 is brought up. The map corresponding to the localization used is selected which are NDT map for ANDT-MCL and NDT-MCL whereas occupancy grid map for AMCL. The type of localization algorithm is assigned before running the experiment. The robot position is initialized at the origin (refer Figure 4)

In Experiment1 ANDT-MCL algorithm is validated by navigating robot in different kind of paths such as- straight line path, triangular path and also the square path which are showed in Figure 7. The reason of using these paths is to test how will be the complexity of the path affecting the localization performance. The reason of using linear path because it's more predictable and preferable as well as it is safer in the production. After executing the algorithm the robot is asked to move according to the path shown in Figure 7 Finally, the overall processes are repeated with another two path assigned as shown in Figure 7.

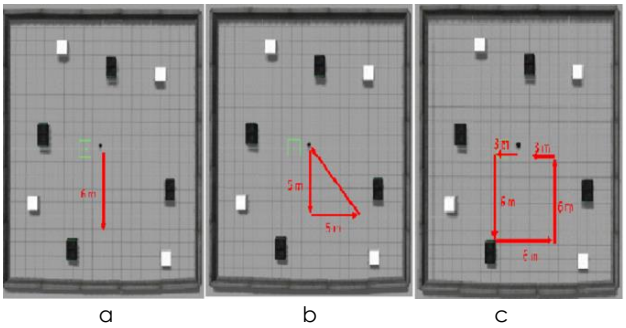


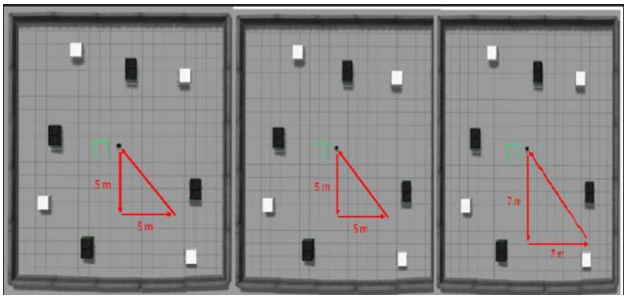
Figure 7. Type of paths (a) Straight path (b) Triangular path (c) Square path

In Experiment 2 the robot is also navigated on a same path but different path lengths. The path that is chosen in this experiment is triangular path with different total path lengths which are triangles made up of side path of 3meters, 5meters and 7meters which are shown in Figure

8. The reason of using these paths is to identify the consistency of the position estimation throughout long distance and also to test how will be the path length of the path affecting the localization performance. After the execution of algorithm, then the robot is asked to move according to the path assigned with certain path length. After that, the overall processes are repeated with another different path length as shown in Figure 8.

Table 1. Settings for the experiment

	ANDT-MCL	NDT-MCL	AMCL
KLD-Error	0.3	-	0.3
Quantile	0.99	-	0.99
Bin-Size	0.05	-	0.05
Particles	4000 -> 100	4000	4000 -> 100
Update Distance (m)	0.1	0.1	0.1



(a) (b) (c)

Figure 8. Different path lengths of triangle path (a) 3 meters (side) (b) 5 meters (side) (c) 7 meters (side)

In the Experiment 3 ANDT-MCL is tested by navigating the robot on a straight path but at different speed. The speeds that are chosen in this experiment are 0.3ms-1, 0.65ms-1 and 1.0ms-1 which are shown in Figure 9. The reason of using these speeds is to test how will be the speeds of the moving robot affecting the localization performance. After the execution of the algorithm, the robot is asked to move the straight path with different speed. Then the overall processes are repeated with another different speed as shown in Figure 9.

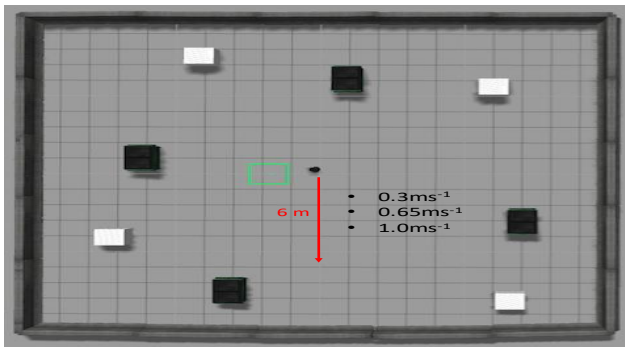


Figure 9. Simulated environment for various speed experiment

In each complete laser scanner's scanning callback, the execution time is also calculated from the start of the ANDT-MCL algorithm to end of the algorithm. The ground truth position and the estimated position along the path is recorded. The errors are calculated from each of the ground truth positions with estimated positions. The average error from all the error from each points is calculated. These processes are repeated 10 times. After repeating 10 times for each localization algorithm, the whole processes are repeated with another two localization algorithm. There are total of 90 sets of execution are run in this experiment. The average error, standard deviation of the error, average execution time and standard deviation of execution time are calculated from the 10 runs, which leads to 9 sets of outputs. The performance evaluation is carried out by analyzing the output of average position estimation error with its standard deviation and computational time with its standard deviation. The lower the position estimation error and the lower the computational time, the better the localization algorithm.

4. RESULTS AND DISCUSSION

The average error results from Experiment 1 are showed in Figure 10. When comparing ANDT-MCL and NDT-MCL the result does not show any significant difference in both straight and square path as shown in Table 3 and Table 4. However, in triangular path, the ANDT-MCL shows significant improvement than NDT-MCL in which it results in 0.0333m average error while 0.0352m average error from NDT-MCL as shown in Table 2.

The outputs produced by AMCL show higher average error compared to ANDT-MCL. The percentage of improvement are 157% for straight path, 18% for triangular path as well as 74% for square path. All of them shows the significant improvement from ANDT-MCL over AMCL except in triangular path as in Table 3 to Table 4.

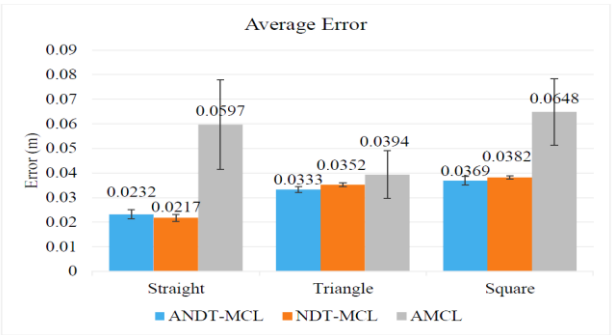


Figure 10. The average error and standard deviation produced by each of the algorithms

Table 2. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in triangular path

Triangle			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	5.92
	AMCL	0.082	18.32

Table 3 Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in straight line path

Straight Line			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.072	-6.17
	AMCL	0.000	157.51

Table 4. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in square path

Square			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.085	2.88
	AMCL	0.000	74.71

In term of execution time performance, the results are shown in Figure 11 below. When comparing ANDT-MCL and NDT-MCL, ANDT-MCL shows huge improvement over the NDT-MCL as shown in Table 5 to Table 7. However, the average execution time of AMCL is less than the ANDT-MCL. Only the straight path which does not show the significant different between the two performances by ANDT-MCL and AMCL. From the results obtained, in term average position error, ANDT-MCL has similar performance as NDT-MCL because both of them are using NDT algorithm for the static map and also for update stage for the weighting the particles which results in better performance than AMCL. With the use of NDT map and the integration of NDT in AMCL, the weighting of the particles is weighted more efficiently and it results in better position estimation.

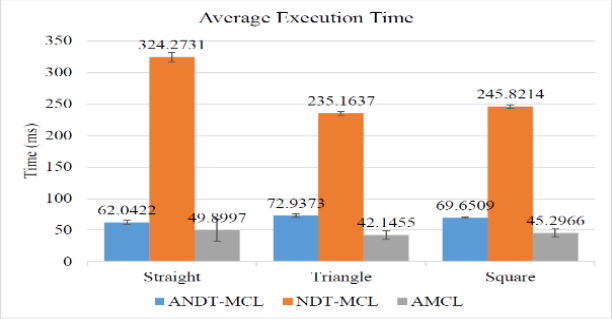


Figure 11. The Average Execution Time and Standard Deviation Produced by Each of the Algorithms

Table 5. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in straight line path

Straight Line			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	422.67
	AMCL	0.054	-19.57

Table 6. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in triangular path

Triangle			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	222.42
	AMCL	0.000	-42.22

Table 7. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in square path

Square			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	252.93
	AMCL	0.000	-34.97

AMCL which uses the typical method of weighting algorithm has poorer result compared to the proposed ANDT-MCL. In term of execution time, ANDT-MCL can outperform better than NDT-MCL. It can adapt the number of particles needed throughout the navigation, hence it reduces the computational effort. However, the AMCL has less computational time than ANDT-MCL because it does not include algorithm like NDT in its weighting phase hence the computational time is reduced.

From experiment 2 the average error results are showed in Figure 12. When comparing ANDT-MCL and NDT-MCL the result does not show any significant difference in both 3 and 7 meters path length as shown in Table 8 and Table10. However for 5 meters path lengths ANDT-MCL shows significant improvement as shown in Table 9. ANDT-MCL shows better outputs in three different path lengths. But AMCL shows higher average error compared to ANDT-AMCL. The percentage of improvement are 63% for 3meters(side) path, 18% for 5meters(side) path as well as 15% for 7meters(side) path as shown in Table 8 to Table 10. Besides, from these 3 different path lengths, ANDT-MCL maintains the performance at average 0.0342m error while AMCL shows inconsistent average position estimation.

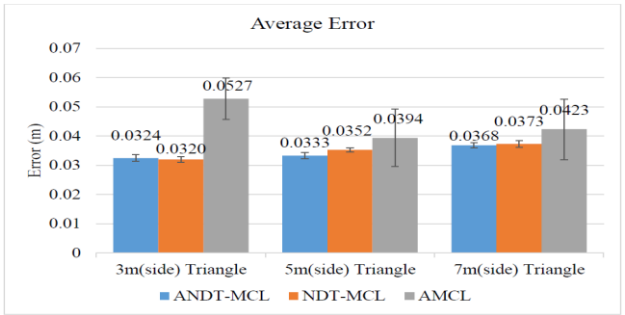


Figure 12. The Average Error and Standard Deviation Produced by Each of the Algorithms

In term of execution time performance ANDT-MCL shows huge improvement over NDT-MCL as shown in Figure 13. The significant improvements are shown in Table 11 to Table 13.

Table 8. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in 9meters path

3m(side) Triangle			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.359	-1.39
	AMCL	0.000	62.53

Table 9. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in 15meters path

5m(side) Triangle			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	5.92
	AMCL	0.082	18.32

Table 10. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in 21meters path

7m(side) Triangle			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.300	1.34
	AMCL	0.129	14.91

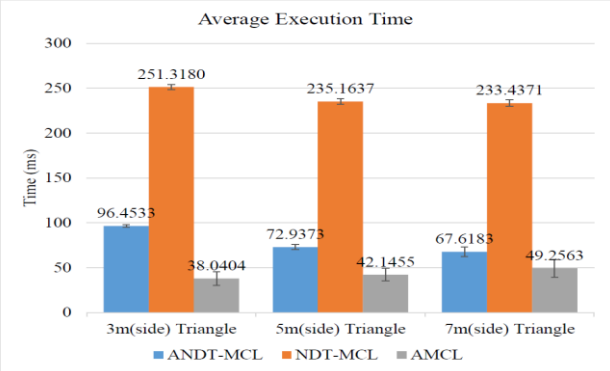


Figure 13. The Average Execution Time and Standard Deviation Produced by Each of the Algorithms

Table 11. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in 9meters path

3m(side) Triangle			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	160.56
	AMCL	0.000	-60.56

However, the average execution time of AMCL is less than the ANDT-MCL. From the results obtained, the performance behavior and execution time are almost similar to the previous experiment (different kind of path). All of the localization shows increase in average position errors when the speed of the robot is increased. ANDT-MCL shows the best position estimation performance among these three speeds. The average error results obtained from the Experiment 3 are showed as following Figure 14. While comparing between ANDT-MCL and NDT-MCL, for 0.3ms-1result doesn't show significant difference as shown in Table 14. But ANDT-MCL has better result than NDT-MCL for 0.65ms-1 with significant difference as shown in Table 15. However, with 1.0ms-1, the ANDT-MCL only has better result than NDT-MCL but with significance difference as shown in Table 16. While comparing ANDT-MCL shows better outputs in all of the three different path lengths. The outputs produced by AMCL are 0.0597m (0.3ms-1 speed), 0.1004m (0.65ms-1 speed) and 0.1108m (1.0ms-1 speed) in which the average errors have higher average error compared to ANDT-MCL as shown in Table 14 to Table 16.

Table 12. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in 15meters path

5m(side) Triangle			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	222.42
	AMCL	0.000	-42.22

Table 13. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL in 21meters path

7m(side) Triangle			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	245.23
	AMCL	0.000	-27.16

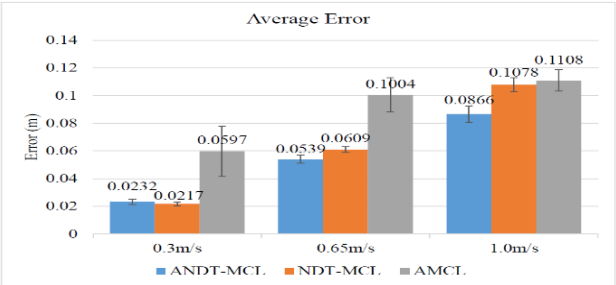


Figure 14. The Average Error and Standard Deviation Produced by Each of the Algorithms

Table 14. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL at 0.3m/s

0.3m/s			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.072	-6.17
	AMCL	0.000	157.51

The execution time performance results are shown in Figure 15. ANDT-MCL shows huge improvement over NDT-MCL as shown in Table 17 to Table 19. As like previous two experiments the average execution time of AMCL is less than ANDT-MCL.

Table 15. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL at 0.65m/s

0.65m/s			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	12.88
	AMCL	0.000	86.04

Table 16. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL at 1.0m/s

1.0m/s			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	24.53
	AMCL	0.000	27.98

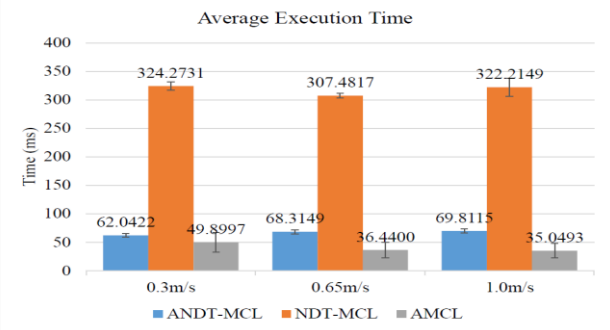


Figure 15. The Average Execution Time and Standard Deviation Produced by Each of the Algorithms

Table 17. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL at 0.3m/s

0.3m/s			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	422.67
	AMCL	0.054	-19.57

Table 18. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL at 0.65m/s

0.65m/s			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	350.10
	AMCL	0.000	-46.66

Table 19. Statistically analysis for ANDT-MCL with NDT-MCL and AMCL at 1.0m/s

1.0m/s			
From	**To**	**P - Value**	**Improve (%)**
ANDT-MCL	NDT-MCL	0.000	361.55
	AMCL	0.000	-49.80

5. CONCLUSION

In this study, ANDT-MCL is successfully developed by integrating MCL with NDT and KLD particles adaption technique with SR resampling. . The algorithm is adaptive to the number of particles used, higher position estimation than current algorithms, higher consistency and lower computational effort. ANDT-MCL performs better than NDT-MCL in terms of position estimation and computational time whereas it also performs better than the AMCL in terms of position estimation and consistency. From the Experiments 1-3 we can conclude that ANDT-MCL can cope with different kind of path complexity, path lengths and high speed movement. So this algorithm can be used to produce better position estimation and be consistent in long journey as well as can outperform in high speed navigation.

ACKNOWLEDGEMENT

The authors are grateful to Universiti Teknologi Malaysia for providing facilities to conduct this research and this work is supported by the Universiti Teknologi Malaysia Research Grant (04G42 and 16H47). Besides, the authors also thank to DF Automation & Robotics Sdn Bhd for supplying the test robotic platform.

REFERENCES

[1] 31 Million Robots Helping in Households Worldwide by 2019. [Online]. https://ifr.org/ifr-press-releases/news/31-million-robots-helping-in-households-worldwide-by-2019. Date accessed: 2017 November 25.

[2] M. S. Grewal, "Kalman Filtering," *Springer*, 2011.

[3] I. A Rekleitis, "Particle Filter Tutorial for Mobile Robot Localization," *Cent. Intell. Mach. McGill Univ. Tech. Rep. TR-CIM-04-02*, 2004.

[4] P. Del Moral, "Nonlinear Filtering : Interacting Particle Resolution," *Markov Process. Relat. Fields*, vol. 2, pp. 555–581, 1996

[5] W.Yu, J.Peng, X. Zhang, S.Li and Liu, W, "An Adaptive Unscented Particle Filter Algorithm through Relative Entropy for Mobile Robot Self-Localization," *Math. Probl. Eng*, 2013

[6] S.Thrun, W.Burgard and D. Fox, "Probabilistic Robotics," *MIT press*, 2005.

[7] D. Fox, "Adapting the Sample Size in Particle Filters through KLD-Sampling," *Int. J. Rob*, pp. 985–1003, 2003

[8] K.György , A. Kelemen and L.Dávid, "Unscented Kalman Filters and Particle Filter Methods for Nonlinear State Estimation," *Procedia Technol*, vol. 12 pp. 65–74, 2014.

[9] M.Montemerlo, S.Thrun, D.Koller and B. Wegbreit, "FastSLAM : A Factored Solution to the Simultaneous Localization and Mapping Problem," *Aaai/iaai*, pp. 593–598, 2002.

[10] J.Saarinen , H.Andreasson , T.Stoyanov and A. J. Lilienthal,"Normal Distributions Transform Monte-Carlo Localization (NDT-MCL)," *Intelligent Robots and Systems (IROS) 2013 IEEE/RSJ International Conference on*, pp. 382–389, 2013,.

[11] R.Valencia, J.Saarinen, H.Andreasson, J. Vallv, J.Andrade-cetto and A. J.Lilienthal, "Localization in Highly Dynamic Environments Using Dual-

Timescale NDT-MCL." *in Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp-3956–3962, 2014.

[12] R.Van Der Merwe, A.Doucet, N.De Freitas and E. A.Wan, "The Unscented Particle Filter". in *Advances in neuralinformation processing systems*, 2001.

[13] D.Fox, "KLD-Sampling : Adaptive Particle Filters," in *Advances in neural information processing systems*, pp- 713–720, 2002.

[14] X.Shao, B.Huang and J.M.Lee , "Constrained Bayesian State Estimation – A Comparative Study and a New Particle Filter Based Approach," *J. Process Control*, vol. 20, pp.143–157, 2010.

[15] H.Alkhatib, I.Neumann, H.Neuner and H.Kutterer, "Comparison of Sequential Monte Carlo Filtering with Kalman Filtering for Nonlinear State Estimation", in *1st International Conference on Machine Control Guidance*, pp. 1–11, 2008.

[16] E.Jung, H.Cho, J.Do, J.Kim and S.Kim, "Implementation of Laser Navigation System using Particle Filter," in *Control, Automation and Systems (ICCAS), 2011 11th International Conference* on, pp. 1636–1638, 2011.

[17] L.D.Alfonso, W.Lucia, P.Muraca and P.Pugliese, "Mobile Robot Localization via EKF and UKF : A Comparison Based on Real Data," *Rob. Auton. Syst*, vol. 74, pp.122–127, 2015.

[18] P. Biber and W. Straaer, "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, pp. 2743–2748, 2003.

[19] J.-SGutmann, W.Burgard, D.Fox and K.Konolige, "An Experimental Comparison of Localization Methods," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference* on, vol. 2, pp. 736–743, 1998.

[20] W. Yu, J. Peng, X. Zhang,S. Li and W. Liu, "An Adaptive Unscented Particle Filter Algorithm through Relative Entropy for Mobile Robot Self-Localization," *Math. Probl. Eng. 2013*, 2013.

[21] S. Dihua,Q. Hao, Z. Min, C. Senlin and Y. Liangyi, "Adaptive KLD sampling based Monte Carlo localization," *2018 Chinese Control Decis. Conf*, pp. 4154–4159, 20.