International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

Optimization of SLAM Gmapping based on Simulation

Werede Gunaza Teame[1], Prof. Wang Zhongmin[2], Dr. Yanan Yu[3]
[1 2] School of Mechanical Engineering,
[3]School of information technology and Engineering
Tianjin University of Technology and Education
Tianjin 300222, China

***Abstract*-This paper presents the Optimization of Simultaneous Localization and Mapping (SLAM) Gmapping algorithm, and compare the results of experiments based on the optimized parameters. From the known types of SLAM, we analyze Gmapping, and used the dataset and the ground truth standard map to carry out the experimental results in the simulation. Using the dataset result as a reference, and optimizing the parameters, we have different simulation results. We conducted The optimization and testing experiments in two ways, at first optimizing parameters separately, and second optimizing more than one at a time. This enables us to conclude the preferable way of optimizing parameters which led the close map result to dataset map or ground truth.**

Keywords-SLAM, Gmapping, Dataset, Ground truth, Algorithm

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is one of the most widely researched, under-researched field, especially in recent decades. And most definitely will be studied more to the extent. Conceptually SLAM is basically the two-operation process. For a mobile robot placed in unknown location, building and output the map of the environment, which is mapping, and while doing this, the robot localizes its own location which we call this Localization. Which is estimating the robot's location [1]. So, SLAM is Building the map of unknown environment and localizing of the robot simultaneously. While performing this operation the sensors fitted in the robot enables to visualize the environment and create the map [2]. indeed, there have been faced a lot of problems on this area which led and paves a way for brilliant scholars to do their best for the best way on how to solve the problems.

As a result, a lot of methodologies, approaches and developments have been seen significantly since the concept was introduced. Especially the tap root area of SLAM, Algorithm.

Different Algorithms have been put under research and we have seen results confirmed based on new types of algorithm. Although all invented SLAM algorithms share the same ultimate goal, but they

have their own features. In addition to this most SLAM problems, based on algorithm, use Baye's rule to solve mapping problems [3]. So far Different algorithm methods have been invented like Gmapping, Hector SLAM, and Karto SLAM. In this paper we conducted Gmapping algorithm to optimize and compare each result interms of its progress relative to the dataset map. We took the standard dataset and ground truth map the parameters from the source code, the C++ program.[1] and optimize the parameters in different ways, in which are described in this paper, and save accordingly. Based on the optimization the new results are compared with standard dataset map. And identified the progress and effects each parameter has on the map. And further conclusions have been given. We use Ubuntu 16.04 to undergo the optimization experiment.

II. RELATED WORK

In the past decades interests has been increasing astoundingly and exponentially in SLAM algorithm. Since a lot of SLAM problems are there, different solutions and methods have been astonishingly introduced on solving those problems and exciting progress have been seen. Basically, related to this paper we focused on Gmapping Algorithm. Optimizing SLAM algorithm has vital role on solving SLAM problem. Shortly sensors are fitted on the robot, and the sensors feed the SLAM algorithm every information including odometry, LIDAR datas, which in return, generates the surrounding map. And localize its own location. It is here where the heart of SLAM problem is occurred. And the solutions can be given either comparing different types of algorithms or optimizing parameters. It can be in a real environment or simulation model. It is highly believed and seen that, for better mapping and localization results, detecting the algorithm function and optimizing the parameters play a vital role. Classifying each parameter's function, and their effect on the map relative to the dataset and ground truth, enables to improve robot's performance on to solve the SLAM problems. Ruoxi Wu, Leizheng Shu, Xin Zhao [4], introduce navigation precision based on Rao-Blackwellized particle filter (RBPF). Factors and errors that affect the accuracy were listed out briefly, and applied firefly algorithm technicality. A grand result has been showed based on the simulation. providing a bright solution to the problems of navigation error, and the Gmapping algorithm shows accuracy in addition to solving the unsteadiness of positions. The vital actor on improving the Gmapping algorithm was the firefly algorithm. In real indoor environment, tests have been also conducted based on SLAM algorithm. In an analysis [5], the SLAM algorithms were applied in the same experimental environment by using crawler-based robot. And concludes that grid maps can be constructed with high-precision in addition to other necessary results.

Doris M. Turnage take the simulation results of the three types of laser-based SLAM algorithm, Gmapping, coreSLAM, and HectorSLAM and compare based on their

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

performance [6]. from the analysis while performing according to the input, the implementation aspect was different, particle filter was used for Gmapping. While HectorSLAM and coreSLAM uses scan matching. Interms of the ground truth map topologies, the algorithm has to be laser-based data in the simulation. For more accurate map topology, changing few parameters was enabled from the simulation. In this paper the Gmapping simulation map result has higher intensity than the standard map. In general, the three algorithms perform interchangeably in different aspects. For the map comparison, Hausdorff Distance was calculated. The analysis made by Rauf Yagfarov, Mikhail Ivanou and Ilya Afanasyev, use the compare and analyze the SLAM libraries, Gmapping, Google cartographer, and Hector SLAM in accordance to the ground truth. The experiment used metrics of average distance to the nearest neighbor (ADNN). The comparison subject was to construct accurate map in accordance to ground truth [7]. the investigation claims Gmapping has close enough to the constructed maps generated by Google cartographer. This paper also claims the comparison is being suitable even for the three-dimensional in addition to the two-dimensional maps.

The other big issue rises during SLAM problem analysis, which is the map consistency. During our experiment, we have a look over the map consistency after optimization the source code. Since we generate a lot of experiments, by changing the parameters based on the two ways we define later, we gave firm attention on whether the map consistency varies as per the optimization or not, so that we can indicate later on the real environment, one can have autonomous navigation result. Indeed, there can be different analysis on the map consistency. Mladen Mazuran, Gian Diago Tipaldi, Luciano Spinello, Wolfram Bugard, Cyril Stachniss analyzed two basic computation on the consistency of map in SLAM based on 50 maps [8]. They introduced quite different from the mostly used technique graph-based SLAM paradigm, which rely on computing automatically. After pertaining tests statistically, since this enables to tuning the parameters, then after, the different results were recorded and introduce this method to successful map consistency.

Other SLAM algorithms also have been introduced by different researchers on different solving problems. For indicating landmarks on the generated map, R. Lemus, S. Diaz, C. Gutierrez, D. Rodriguez and F. Escobar [9], analyzed with scanning laser range finder and radiofrequency identification technology (RFID). Generated on the virtual map, and relative to the SLAM-R generated map based on the cycle closure technique, it is possible to generate 2D maps in indoor environment and improvements showed.so it is confirmed that introducing and solving or giving firm analysis, optimization, and developments on the algorithms gives and paves a way to solve different SLAM problems. Optimizing algorithms since the time from the very beginning researchers gave different methods was based on sensors, like laser or sonar, interms of deep calculations the likes of Kalman filters, Rao-Blackwellized particle filters and so on [10]. Different other approaches have ben also tested. Frank Dellaert also introduced the Factor graphs smoothing techniques [11].

OPTIMIZATION of GMAPPING

Till this day, different experiments and comparison and evaluation has been held between the recognized SLAM algorithm [3][4][5][6][7]. probability is one thing in common they share which enables the robot output impervious to failure in some conditions, and decrease measurement robustness [3]. For the simulation experiment we analysis Gmapping, which is the most used so far since the first time proposed by Gisetti 2006 [12]. And is open source SLAM algorithm in 2007 [5]. For the problem of positioning and mapping, it used Rao-Blackwellised particle filter (RBPF) [5], which is a version of particle filter [13][18][21]. it is derived from Monte Carlo algorithm, while solving its SLAM and use laser-based SLAM for the map result [12]. as is known SLAM problem or map robustness rises due to a lot of factors including hypothesis space, where being large of the map space, and the learning maps which is a “chicken-and-egg” problem [14]. Which is directly a problem related to odometry. This led to localization problem. So, the factors that decide SLAM problem including the environment cycles, resemblance of different positions, the environment’s being wide and so on. To overcome the problems, it is mandatory to refer definition the algorithms. Of the Gmapping, it should rely on the algorithm technicality to calculate and generate the map. For a map m, since the path is already known, according to the dataset or ground truth, with the robot’s poses, path of the robot $x_{1:t}$, at a time t, the distribution according to set of all measurements, $z_{1:t}$

$$p(\boldsymbol{m}_i \mid z_{1:t}, x_{1:t}) \quad (1)$$

While including the robot’s controls, u1:t we’ll have:

$$p(x_{0:t}, m \mid z_{1:t}, u_{1:t}) \quad (2)$$

Distribution robot’s path map observation control

Further calculations follow here. initial pose of the robot, and estimation of the entire path of the robot (full SLAM), as well as seeking the updated recent pose (online SLAM) respectively and finally in a given or known environment as well as known trajectory.

$$p(m, x_{1:t} \mid z_{1:t}, u_{1:t}, x_0) \quad (4)$$
$$p(m, x_t \mid z_{1:t}, u_{1:t}, x_0) \quad (5)$$
$$p(x_{1:t}, \mid m, z_{1:t}, u_{1:t}, x_0) \quad (6)$$
$$p(m \mid z_{1:t}, x_{1:t}) \quad (7)$$

As is already defined, since the dataset path is known we use (1), and splitting the grid map with different grid cells ($m_1, m_2, m_3, \ldots\ldots, m$). So, we have

$$m = \sum_i \mathbf{m}_i \quad (8)$$

Having this grid cells, the distribution based on the estimation, and the distribution based on the posterior maps

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

will give us the following. We will not use (2) rather we use (1), because we had ignored the controls.

$$p(m \mid z_{1:t}, x_{1:t}) = p(m_i \mid z_{1:t}, x_{1:t}) \quad (9)$$

The analysis in [14], discussed the grid mapping algorithm (table 1) to the binary estimation problem the filter after using the log-odds calculation:

$$l_{t,i} = \log \frac{p(\mathbf{m}_i \mid z_{1:t}, x_{1:t})}{1 - p(\mathbf{m}_i \mid z_{1:t}, x_{1:t})} \quad (10)$$

$$p(\mathbf{m}_i \mid z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + \exp\{l_{t,i}\}} \quad (11)$$

The algorithm based on the robot's path, in a given path through the grid cells under the function inverse_sensor_model.

1: **Algorithm occupancy_grid_mapping**($\{l_{t-1,i}\}, x_t, z_t$):
2: for all cells m_i do
3: if m_i in perceptual field of z_t then
4: $l_{t,i} = l_{t-1,i} + \text{inverse\_sensor\_model}(m_i, x_t, z_t) - l_0$
5: else
6: $l_{t,i} = l_{t-1,i}$
7: endif
8: endfor
9: return $\{l_{t,i}\}$

Fig 1. Occupancy grid algorithm

For the grid maps based on the laser scan data Rao-Blackwellized particle filter (RBPF) is applied (15).

$$p(x_{1:t}, P \mid z_{1:t}, u_{1:t} - 1) \quad (12)$$

And with no odometry,

$$p(x_{1:t}, m \mid z_{1:t}) = p(m \mid x_{1:t}, z_{1:t}) \; p(x1{:}t \mid z_{1:t})$$

Particle filter analytical computation

Based on (11) which defines RBPF, the laser data feeding the path's measurements, particles get the landmarks or the poses. which are defined by the laser data. So now the laser scan with the observations or measurements, z_{t-1}, the pose is identified. Based on the algorithm, and from the distribution, p_{t-1} the map will be obtained. In every landmark and poses, the particles will be registered based on x_t, and z_t. [15]. The analysis [15] [16], claimed that RBPF as the better than Extended Kalman Filter (EKF) and other Kalman filter algorithms.

While the SLAM problem rises here in generating the map trajectory, RBPF algorithm will give a solution. Extending (8) it will use: (remind here including the controls)

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t} - 1) = p(m \mid z_{1:t}, u_{1:t} - 1) \cdot p(x_{1:t} \mid z_{1:t}, u_{1:t} - 1) \quad (14)$$

For detailed steps of the algorithm it has discussed in [4][16][17][18]. While we optimize the algorithm based on the source code, Gmapping provide to tune the parameters used by RBPF. A work by Giorgio Grisetti, Cyril Stachniss, Wolfram Burgard [19], claimed grid mapping with Rao-Blackwellized Particle Filter (RBPF) do the best on solving the SLAM problem by performing in each robot trajectory besides the map. and acquire accurate maps.

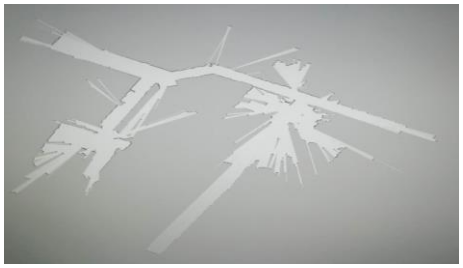
The general overlook on the working of the Gmapping SLAM, as defined shortly, we get the environment's information from LIDAR as well as the IMU feeds the attitude, the algorithm starts to occupy after getting the state of the robot from odometry, which updates the current state. While ongoing situations continue, either EKF or RBPF algorithms directly feed the position of the robot as well as the attitude. Interms of the RBPF, with discussed steps [4][16][17][18], after algorithm enable scan matching, the particle filters will be able to start the sampling. And while the odometry continue to update the current state, scan matching will resample and generate the map. Automatically the odometry will continue to feed the pose condition to the scan matching. Scan matching then update the information in the same cycle used as before.

The motive for doing this research is significant in optimization of Gmapping algorithm. Most researches showed algorithm optimization based on the three types of SLAM algorithm, which is not enough to give precise direction on the parameters. So, this research will give the parameters and their effects in Gmapping algorithm. The other thing that make this study from other optimizations is, the methods used during the optimization. We can classify as optimization based on the parameters separately and see the results, and the second method is optimizing the parameters collectively. This will pave a way to understand the effects of every parameter and their function. And create a motive to optimize researchers in a real environment.

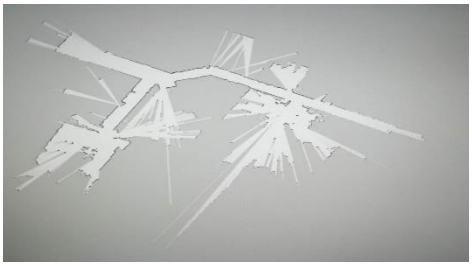
III. RESULTS AND DISCUSSION

In this paper, we analyze Gmapping and optimize the parameters in the source code, and undergo experiment to see the result which optimization is close matching relative to the dataset or ground truth. The source code is available in. we use two methods of the optimization. At first, we use optimizing the parameters separately and see all the results and the second optimizing the parameters collectively.and we will see the results below.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020



Map 1. Ground truth map (touch stone)



Map 2. Gmapping map using logged data

Fig 2. Maps showing the ground truth and the map we used as default value

From The above two maps. Map 1 indicates the ground truth map which is our touch stone. This is the standard map of the dataset, which is the collection of datasets gained from the map, that is the map described by the ground truth. In other words, the ground truth represents the information of the map. While Map 2 is Gmapping map built by using logged data which we refer as default values [21]. It is generated by running the dataset based on the algorithm, which is the first map estimate mentioned above. So, there is an error compared with the touchstone, which means that experiments are needed to have to get close map to the ground truth map.

Parameter explanation

The parameters are the key for our optimization. We use 10 of the parameters for the first method, and 12 of the parameters for the second method. Their function is as follow [22][23]:

Sigma: this is the parameter used by the greedy end point matching. It is standard deviation for the scan matching process.

Kernel size: the kernel which seeks the correspondence, or for the scan matching process.

lstep: this is initial search or optimization step for the scan matching process, mainly for translation (linear).

astep: this is initial search or optimization step for the scan matching process, mainly for rotation (angular).

Srr: the linear function of odometry error in linear (x and Y).

Srt: the rotational or angular function of odometry error in linear (theta).

str: the linear function of odometry error in angular or rotational (x and Y).

Stt: the rotational function of odometry error in rotation.

Linear update: the measurement, or the length where the robot moves or rotates, to process a new scan.

Resample threshold: The resampling threshold based on the particles resampled.

Particles: the number of particles, which defines the robot's possible trajectory.

Map update interval: this is the time the robot updates the map based on its scanning.

We take this parameter because they are editable in the source code. Indeed, except the map update interval, the rest are parameters used by Gmapping itself. The reason why we include the map update interval is, to take the time between successive recalculations of maps. Based on the optimization process, our experiment's result is included below.

1.optimizing the parameters separately

Here we tried to optimize selected parameters as per their function, separately by neglecting the rest as default. Out of the active parameters we perform and use editable parameters and appropriate of them, and evaluation of results were carried out relative to the touchstone, and compared with the dataset. The reason behind to choose the parameters are because they are editable, which means we can provide new values and run the program. During the optimization, what was our reference for changing the parameters? Is the main question. What we did during the optimization is according to two procedures. The first one is optimizing the parameters uniformly. The values of the parameters were changed in an equity manner which is uniform optimization. The second thing is, using reference. We take the time (in seconds), between two successive recalculations of the map, which is the map update interval. This is the parameter used by the Gmapping algorithm itself. We use This parameter as reference and the default value, 5.so we multiply each parameter by the map update interval.

$$optimization\ parameter(OP) = map\ update\ interval \times default\ parameter$$

As a result, we have showed in Table 1, what the values looks like.

Parameter Specification

Table 1. optimization values of parameters

parameters	Default value	OP
sigma	0.05	0.25
Kernel size	1	5
astep	0.05	0.25
srr	0.1	0.5
srt	0.2	1
str	0.1	0.5
stt	0.2	1
Linear update	1.0	5
Resample threshold	0.5	2.5
particles	30	150

After the optimization, each result was saved, and the bring abouts was seen. The detailed map result is shown in Map 3~ Map5 below.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

Optimization result explanation

Based on the experiments, after each parameter was optimized, the map results and the necessary significant results m_count, Neff, and average scan matching score varies as per each experiment.

Explanations based on map: Results rely on the areas where we saw the changes. Having the map results, we have the definition as:

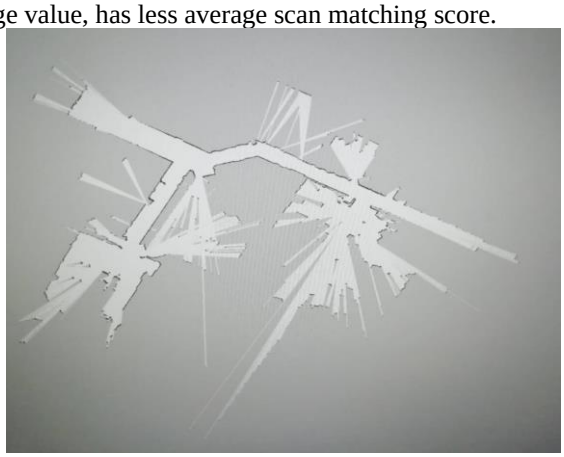
- ✓ Map results of parameters: sigma, srr, srt, str have close map results. And are far from the ground truth. Map 3 is shown as representative.
- ✓ Map results of parameters: srt, resample threshold have close result of the map. Map 4 is shown as representative.
- ✓ Map results kernel size, linear update, and particles have the same effects on the map. And show the best effects so far. These three parameters have different m_count from the other parameters. Map 5 is shown as representative.

Explanations based on m_count, Neff, and average scan matching score: we have also the comparison as shown below in Table 2, based on the m_count, Neff, and average scan matching score.

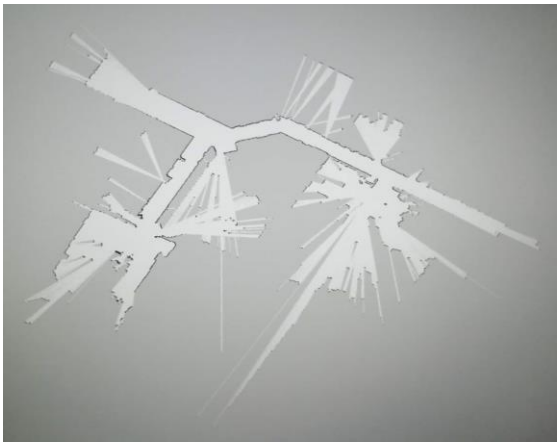
Since the optimization is processed uniformly, we didn't receive the same results interms of m_count, neff, and the average scan matching score.

1.we have the same m_count for all parameters except kernel size, linear update, and particles.

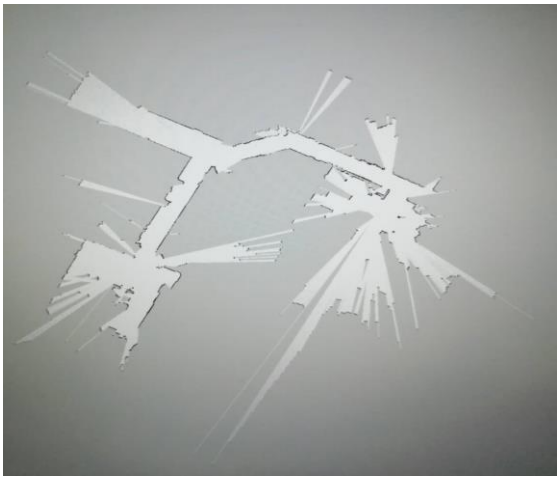
2. when the value of neff increase for each parameter, the average scan matching will increase accordingly, except particles. For the number of particles in the filter, with the large value, has less average scan matching score.



Map. 3 the sigma parameter result



Map 4. Resample threshold parameter result



Map 5. Particles parameter result

Fig 3. Representative 3 Maps showing after each optimization the parameters separately

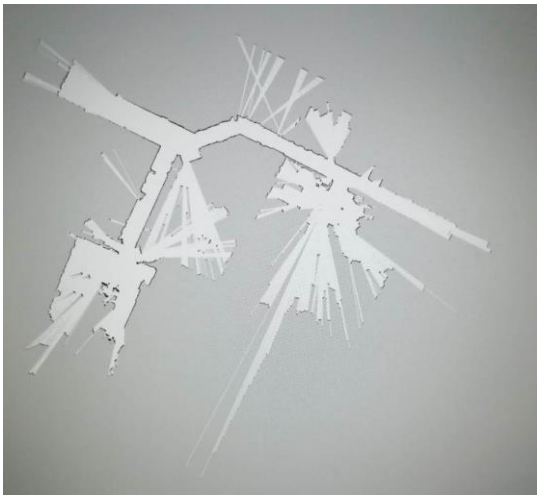
Having this map result, we have also the comparison as shown below Table 2, based on the m_count, Neff, and average scan matching score.

(2) optimizing the parameters collectively

In this experiment we undergo the optimization, by taking the parameters collectively. Indeed, this is the better optimization to get the best result to match the standard dataset map. most focusing on the parameters: matching of endpoints, optimal step size of translation and rotation, and indeed on the odometer detail and so on. We undertook four experiments.in the last two experiments we also include the map update interval, which showed us the best results. the values are shown in the table below Table 3.

By optimizing the parameters in each experiment, we have the map results below.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

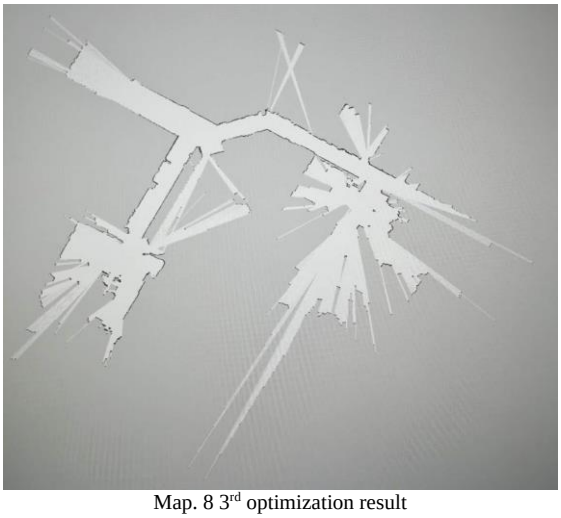


Map 6. 1st optimization result

The first experiment was optimizing the parameters which includes sigma, lstep, astep, srr, srt, str, stt, linear update, angular update, resample threshold. We focus only the mentioned parameters to see parameters which feed the scan matching (sigma, lstep, astep), the effects of motion model parameters (srr, srt, str, stt), and the parameters which enable the robot to process new measurements (linear update, angular update).

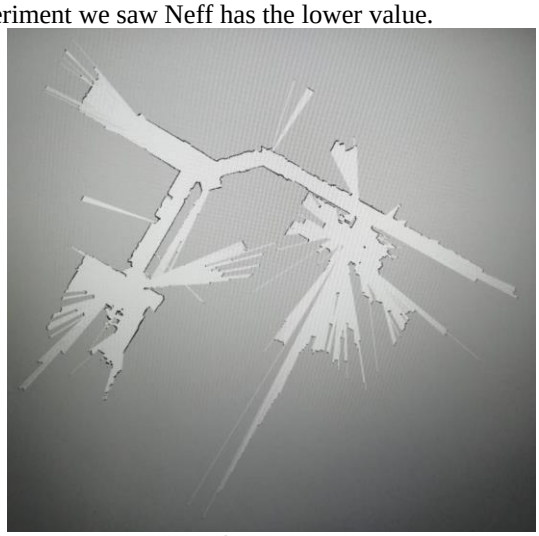
Map 7. 2nd optimization result

This selected optimization result is based on the parameters sigma, lstep, astep, srr, srt, str, stt, linear update, particles and resample threshold. The reason for this result more appropriate and better than experiment 1 is the addition of particles in the optimization.



Map. 8 3rd optimization result

Here appears with high resemblance to the ground truth map, which includes kernel size, lstep, motion model parameters (srr, srt, str, stt), linear update, map update interval and particles. We focus on the motion model parameters with the particles, which is the better optimization so far. Studies show that Neff has the least value over time [19]. for Neff to regain maximum value, resampling plays vital role. In this experiment we saw Neff has the lower value.



Map. 9 4th optimization result

Fig 4. Maps showing after optimization parameters collectively

The last discussed optimization is based on parameters sigma, lstep, astep, srr, srt, str, stt, linear update, particles, map update interval and resample threshold. This experiment registered the lower value of Neff as well as lower average scan matching score.

For all the above results, the values of each experiment are shown below in Table [2][3][4].

Table.2 effects of the optimization in each parameter

parameters	Default value	OP	m_count	neff	Average scan matching score
sigma	0.05	0.25	38	28.2732	961.734
Kernel size	1	5	16	4.23896	784.87
astep	0.05	0.25	38	18.1783	953.515
srr	0.1	0.5	38	17.7095	915.318

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

srt	0.2	1	38	14.6944	853.491
str	0.1	0.5	38	21.3017	936.548
stt	0.2	1	38	17.1335	769.493
Linear update	1.0	5	14	28.2693	943.341
Resample threshold	0.5	2.5	38	24.9151	925.188
particles	30	150	14	93.6089	636.611

Table. 3 optimized values of the parameters in each experiment

parameters	Default value	Optimized value			
		First exp't	Second exp't	Third exp't	Fourth exp't
sigma	0.05	0.04	0.09	0.05	0.1
Kernel size	1	1	1	3	1
lstep	0.05	0.03	0.09	3	0.1
astep	0.05	0.03	0.09	0.05	0.1
srr	0.1	0.2	0.5	0.01	0.7
srt	0.2	0.3	0.6	0.02	0.8
str	0.1	0.2	0.5	0.01	0.7
stt	0.2	0.3	0.6	0.02	0.8
Linear update	1.0	1.05	2	2	2.5
Angular update	0.5	0.8	0.5	0.5	0.5
Resample threshold	0.5	0.8	0.9	0.5	1.0
particles	30	30	20	100	20
Map update interval	5.0	5.0	5.0	0.05	0.05

Table 4. optimization effects on m_count, neff, and the average scan matching score

sequence	M_count	neff	Average scan matching score
1st optimization	36	16.0608	920.82
2nd optimization	18	5.00868	848.326
3rd optimization	29	99.9995	1077.96
4th optimization	20	1.11902	815.745

Based on the above the above experiments Adding the Map update interval shows better improvements in order getting the most out of our experiment. And the effects on of m_count, neff, and the average scan matching score is shown In Table 4.

IV. CONCLUSION AND FUTURE WORK

This paper has presented optimization of Simultaneous Localization and Mapping (SLAM) by conducting Gmapping algorithm. We undertook the optimization based on two ways.

1.optimizing the parameters separately

A. optimizing separately led to only a fraction of changes on the map relative to dataset map.

B. We may have a close match or result of the map when we compare each result, although we still optimize separately. for example, when we look at the result after optimizing the sigma value, in which it is important for the matching of endpoints, in some region we may have some resemblance with optimized value or result of kernel size, lstep or number of iterations for scan matching (iterations).

C. Whenever we are optimizing separately, it is not mandatory to have the same result or change. because we improve one parameter does not mean we get uniform development. even we may have unimaginable different results.

2.Optimizing parameters collectively

We tried to optimize by selecting some parameters collectively and compare with the touchstone. and conclude that we can decidedly say having optimizing collectively, can get a bring about result relative to the ground truth. Especially the parameters matching of endpoints, optimal step size of translation and rotation, and indeed on the odometer detail. The more we tried to optimize collectively according to our calculation, the better we have the map result of the dataset. Optimizing including the map update interval Map 3 and 4 shows the better result and closer to the ground truth.

We can thereby conclude and suggest that optimizing SLAM Gmapping algorithm collectively and including the map update interval gives the best results and will have better improvements if further researches held with parameters. Having this optimization, it is also possible to see in real indoor environment using robot, to be more practical. So, for the future real indoor environment is expected.

REFERENCES

[1] Hugh Durrant-Whyte, *Fellow, IEEE*, and Tim Bailey. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. IEEE Robotics & Automation Magazine, June 2006, issue 2, Vol. 13, **pp.** 99 – 110.

[2] Filatov Anton, Filatov Artyom, Krinkin Kirill, Chen Baian, Molodan Diana. 2D SLAM Quality Evaluation Methods. *IEEE*, 2017 21st Conference of Open Innovations Association (FRUCT), January 2018.

[3] Joao Machado Santos, David Portugal and Rui P. Rocha. An Evaluation of 2D SLAM Techniques Available in Robot Operating System. 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR) 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), January 2014.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 04, April-2020

[4] Ruoxi Wu, Leizheng Shu1, Xin Zhao. Optimization of the Grid Mapping Algorithm for Mobile Robots, IOP Conf. Series: Journal of Physics: 2019, Vol. 1237, Issue 2.
[5] Zhang Xuexi1, Lu Guokun, Fu Genping, Xu Dongliang, Liang Shiliu, SLAM Algorithm Analysis of Mobile Robot Based on Lidar, 2019 Chinese Control Conference (CCC), July 2019, pp. 4739-4745.
[6] Doris M. Turnage. SIMULATION RESULTS FOR LOCALIZATION AND MAPPING ALGORITHMS. 2016 Winter Simulation Conference, JANUARY 2017, PP. 3040-3051
[7] Rauf Yagfarov, Mikhail Ivanou and Ilya Afanasyev, *Member, IEEE*. Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth, 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), December 2018, pp. 1979-1983.
[8] Mladen Mazuran, Gian Diego Tipaldi, Luciano Spinello, Wolfram Burgard, Cyrill Stachniss. A Statistical Measure for Map Consistency in SLAM, 2014 IEEE International Conference on Robotics and Automation (ICRA), September 2014, conference location Hong Kong China.
[9] R. Lemus, S. Díaz, C. Gutiérrez, D. Rodríguez and F. Escobar. SLAM-R Algorithm of Simultaneous Localization and Mapping Using RFID for Obstacle Location and Recognition, Journal of applied research and technology, June 2014, Vol. 12, issue 3, pp. 55-559.
[10] Bruno Steux, Oussama El Hamzaoui. CoreSLAM: a SLAM Algorithm in less than 200 lines of C code, 2009.
[11] Frank Dellaert. Factor Graphs and GTSAM: A Hands-on Introduction, Georgia Institute of Technology, Technical Report number GT-RIM-CP&R-2012-002, September 2012.
[12] Seokju Lee, Girma Tewolde, Jongil Lim and Jaerock Kwon. 2D SLAM Solution for Low-Cost Mobile Robot based on Embedded Single Board Computer, The 2017 World Congress on Advances in Nano, Bio, Robotics and Energy (ANBRE17), September 2017, lisan(seoul), Korea.
[13] Johan Alexandersson och Olle Nordin. implementation of SLAM algorithms in a small-scale vehicle using model-based development, Master of Science Thesis in Datorteknik, Fordonssystem, 2017.
[14] Sebastian Thrun, Wolfram Burgard, Dieter Fox. Probabilistic Robotics, 1999-2000.
[15] B.L.E.A. Balasuriya, B.A.H. Chathuranga, B.H.M.D. Jayasundara3, N.R.A.C. Napagoda, S.P. Kumarawadu, D.P. Chandima6 and A.G.B.P. Jayasekara. Outdoor Robot Navigation Using Gmapping Based SLAM Algorithm, IEEE, 2016, pp. 403-408
[16] Wenjun Zhang, Qiao Zhang, Kai Sun, Sheng Guo. A Laser-SLAM Algorithm for Indoor Mobile Robot Mapping, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLI-B4, July 2016, pp. 351-355.
[17] Yassin Abdelrasoul, Abu Bakar Sayuti HM Saman, Patrick Sebastian. A Quantitative study of Tuning ROS Gmapping Parameters and Their Effect on Performing Indoor 2D SLAM, 2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA), February 2017,
[18] Zia Khan, Tucker Balch, and Frank Dellaert. A Rao-Blackwellized Particle Filter for Eigen Tracking, Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. July 2004, Vol 2, pp. 980-986.
[19] Giorgio Grisetti, Cyrill Stachniss, Wolfram Burgard. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters, IEEE Transactions on Robotics, Vol. 23, Issue 1, Feb. 2007.
[20] W.J. Kuo, S. H. Tseng, J. Y. Yu, and L. C. Fu, A hybrid approach to RBPF based SLAM with grid mapping enhanced by line matching," in 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2009, pp. 1523–1528.
[21] David Ryskalczyk, "How to build a map using logged data," *ros.wiki*, 14-Mar-2014.
[22] slam_gmapping 2008, Willow Garage, Inc. Author: Brian Gerkey, Modified by: Charles DuHadway
[23] http://wiki.ros.org/gmapping