New Node Location Setting Using Random Sample Ratio In Occupancy Area On RRT Algorithm

2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED) | 978-1-6654-5389-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCED56140.2022.10010589

Putra Wisnu Agung Sucipto
Electrical Engineering Department
Universitas Islam "45"
Bekasi, Indonesia
wisnu@unismabekasi.ac.id

Seta Samsiana
Electrical Engineering Department
Universitas Islam "45"
Bekasi, Indonesia
seta_samsiana@unismabekasi.ac.id

Annisa Firasanti
Electrical Engineering Department
Universitas Islam "45"
Bekasi, Indonesia
annisa_firasanti@unismabekasi.ac.id

Eki Ahmad Zaki Hamidi
Electrical Engineering Department
UIN Sunan Gunung Djati
Bandung, Indonesia
ekiahmadzaki@uinsgd.ac.id

***Abstract*— The Rapidly Random Tree (RRT) algorithm works to grow trees gradually based on a random sampling process. Each growing branch needs to be arranged so that it can reach the target node. This arrangement is done by maintaining the distance between the edge knot and the parent knot so that they are always in the optimal direction and magnitude. This paper offers an alternative approach to adjust the position of the edge vertices in a branch based on the modeling occupancy area that can be occupied by random knots and the distance function which is formulated using the growth ratio of random samples that inhabit the target area and the area outside it. Based on the experimental results, the algorithm has succeeded in making the tree reach min = 0.5 at JI = 265 with PJ = 4.3 from the root node (1.5) and the target node (5.6). While the basic RRT algorithm has not been able to reach the target node at JI = 265 with the remaining distance to be reached is 2.7442 points.**

Keywords—Occupancy area, Sample Rate Ratio, Distance function

I. Introduction

RRT has gained a lot of the attention of the research community over the past few decades. Most researchers are interested in RRT because it has the ability to work in high-dimensional state space. This algorithm provides the opportunity to achieve a viable optimal solution based on the growth of randomness of the sampling process.

Since it is sampling-based, RRT has probability content in the solution. Thus, RRT can grow tree branches that are worthy or unworthy to develop. The development of these branches, of course, will also be on two possibilities, namely the feasible and not feasible path of growth.

Spatially, the growth and development of branches should be reviewed further. Branches that develop in a residential area that supports its growth, will leave a viable path seedling to reach the final node. On the other hand, the area of infertile branch dwelling will affect the growth of the new branch.

Some researchers have sought for techniques to grow new branch nodes. This study originated from the basic concept of edge node growth in RRT based on the safe distance of new edge growth or so-called edge node distance threshold against parental node [1][2][3][4]. The growth direction of this edge node corresponds to the direction of the parent node distance line and the random node raised in each iteration, so there is a possibility of developing away from the target node.

Furthermore, the edge node distance setting and its parents are set to use the minimum cost provision of a collection of the distances of neighboring parents in a hypersphere centered on the position of the random nodes. The edge node that is successfully laid will have the lowest cost to the tree root node [1][5][6][7]. However, the growth direction of a branch is also likely to grow away from the target node if there is a minimum cost among the collection of parental nodes in the hypersphere that is far from the target node.

Another way that can be used to adjust branch length is by using virtual forces. The edge distance to the parent node is arranged based on the repulsive and attractive forces in virtual forces [8][9][1]. The direction of the growth of the edge branch can reach the target node because of the existence of attractive forces around the target node. However, the emergence of a large distance from the resultant of this virtual style needs to get further attention.

Determination of the new edge location and values in the direction of branch growth is often also reviewed based on the setting of the distance function coefficient. For example, the setting of the target bias coefficient is based on the probability of transitions between branches. These coefficients are configured with target gravity variables, the ratio of the distance of the target node to the edge node and the ratio of the distance of the random node to the edge node [10] [11] as well as the complexity of the environment determined subjectively [16]. This provision does not rule out the possibility of using quadratic cost functions in Euclidian terminology and Generalized distance as a traditional traversal method [12] [13] [14] [15].

Setting the direction of branch growth from the root to the target and its magnitude is a crucial part. This growth will be better if the random node approaches the target node and vice versa. Research on the function of branch growth director generally chooses proximity between adjacent nodes and derivative aspects as a reference for formalization. There has been no formulation of restrictions on branch growth based on the portfolio of growth of random nodes that inhabit an area around the target and the outside area. So that it can be known the edge knot that must be maintained to always grow and the edge knot that needs to be limited to its extension of the parent node

This paper offers an alternative approach to setting up new nodes against parental nodes. The main contributions made in this paper related to the setting of the position of new nodes in branches lie in modeling the occupancy area of the target node side and the side of the root node that can be occupied by random samples, and the function of the distance of the parent node to the edge node based on the ratio of the number of random samples inhabiting the target area and the outside area.

II. Research Method

A. Algoritma RRT Basic

Trees grown algorithms are developed by several operating processes. First, the algorithm ran the initialization of *the root*node q_{start} and the target node, q_{goal} in the *T* tree. Then, the branch is grown by placing the edge knot, $q_{new,}$ at the end of the twig. This operation is preceded by the generation of a random sample, $q_{rand,}$ by the *RandomSample()* function. The laying *of* q_{new} is calculated based on the proximity of q_{rand} to the parent knot, $q_{parrent}$ adjacent to it. Using the NearestNeighborfunction () is generated *a node* $q_{parrent}$ that is located from the $_{new}$ $q$with a certain value. After successfully obtaining $q_{parrent}$ then the new $q_{is\ is}$ added to the tree branch through the AddNewNode() function. The sequence of this process repeats continuously until the distance q_{new} and $q$goal$_{reaches}$ a certain minimum value, ρ_{min}. The details of this algorithm are organized from [14][16] and described in the following algorithm.

Algoritma 1 Algoritma RRT Basic

T ← InitTree(q_{start}, q_{goal});
for i = 1 **to** n **do**
  q_{rand} ← RandomSample(i);
  $q_{parrent}$ ← NearestNeighbor(q_{rand});
  q_{new} ← Extend(q_{rand}, $q_{parrent}$, ε);
  AddNewNode(T, q_{new});
  if Distance(q_{new}, q_{goal}) < ρ_{min} **then**
    return T;
  end if
end for

B. New Node Mounting in Branch

The operation of generating a new node as the end of the branch in the AddNewNode() function in algorithm 1, is accompanied by the determination of the direction of the new node and the length of the parent node's distance to the new node. Therefore, in stochastic planners such as RRT, it contains randomness due to the existence of sampling process-based operations, potentially producing random twig pathways that limit convergence towards optimal solutions.

Spatially, the space of empty areas that can be inhabited by random samples as a reference for branch growth and development can be divided into two parts. The first part is called the habitable area around the target node and the habitable area outside the target area. The area around the target node is the area between the root node and the target node represented in the form of a rectangle formed from the *coordinate relationships* q_{start} and q_{goal} in the coordinate field. While the area other than this rectangular area is defined as the area outside it, as described in figure 1.

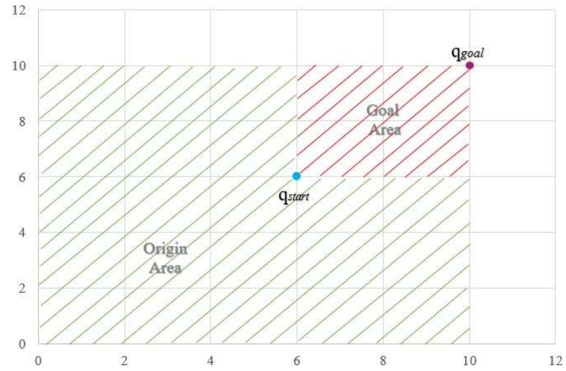

Fig. 1. Illustration of Division between Origin and Target Areas

When the random node has been generated and has occupied one of the two areas, its existence are detected by (1)

$$f(q_s;q_r;q_g) = \begin{cases} 1, & if\ q_s < q_r < q_g \\ 0, & otherwise \end{cases} \quad (1)$$

where :
q_s = root node,
q_r = random node, and
q_g = target node.

Number of random samples inhabiting both areas of origin is explained by expression (2)

$$O_a\left(f(q_s;q_r;q_g)\right) = \sum_{f(q_s;q_r;q_g)=1} f(q_s;q_r;q_g) \quad (2)$$

where :
O_a = Number of samples in origin area.

While those who inhabit the target area are expressed in (3)

$$T_a\left(f(q_s;q_r;q_g)\right) = \sum_{f(q_s;q_r;q_g)=0} f(q_s;q_r;q_g) \quad (3)$$

where :
T_a = Number of samples in the target area.

The growth of the number of random samples in each area needs to be noted as one of the considerations in the mechanism of extension of the new branch path. The more random samples are in the area of origin as a consequence of the area of occupancy, the slower arrival of random samples in the target area. The occupancy area around the target which is much narrower than the original area will cause less habitable empty space for random samples around the target.

This condition brings benefits on the other hand. This difference in conditions can be used as a reference in making the reciprocation function extend or suppress the growth rate of branches. Fewer and fewer random samples in the target area can be used to trigger a slowdown in branch extension in the area of origin. Conversely, the number of random samples in the target area that is less than the area of origin can also trigger the extension of branches in the area of origin.

Given that convergence towards the target node is a top priority and the fact that the growth of the number of samples in both areas can be adjusted in ratio, this condition provides room to implement the reciprocity paradigm in calculating the distance of the parent node to the edge node. With this paradigm, the difference in the length of the edge node distance in both areas gets the right coefficient. In other words, if the growth ratio of the area of origin is greater than the target area, this ratio can be fed as a coefficient of distance function for the target area that can trigger branch based on the probability of randomness of the sample in the occupancy space. Conversely, because the area on the target side is much narrower than the area of origin, the extension has the potential to reduce the growth ratio of random samples in the target area, it can be fed to the area of origin as a reference for shortening of twigs around the edge knot. The growth ratio of random samples in the area of origin is defined by equation (4).

$$\Delta O_a = \frac{O_a}{O_a + T_a} \tag{4}$$

where :

ΔO_a = Random sample ratio in the origin area

while random sample ratio in the target area is defined by (5)

$$\Delta T_a = \frac{T_a}{O_a + T_a} \tag{5}$$

where :

ΔT_a = Random sample ratio in the target area

Thus, the distance of parent and edge node in each occupancy area is defined by

$$d_{pn} = \begin{cases} \Delta T_a, & if\ f\left(q_s; q_r; q_g\right) = 1 \\ \Delta O_a\ x\ d_{PR}, & if\ f\left(q_s; q_r; q_g\right) = 0 \end{cases} \tag{6}$$

where :

d_{pn} = Distance between parent and edge node
d_{PR} = Distance between q_r and q_p
q_p = Parent node.

Thus, the location of edge node in tree branch are

$$q_n^x = (q_r - d_{pn}) * \cos\left(\theta_{PR}\right) \tag{7}$$

$$q_n^y = (q_r - d_{pn}) * \sin\left(\theta_{PR}\right) \tag{8}$$

where :

q_n^x = Edge node coordinate in X axis
q_n^y = Edge node coordinate in Y axis
θ_{PR} =Angle between parent and random node

III. RESULTS AND DISCUSSION

The proposed methods and the basic RRT algorithm are compared based on several performance criteria indicators; path length (PJ), number of iterations (JI) and number of twigs (JR) formed and damped. The Comparison are implemented using Matlab runs on Laptop PCs with Intel i7-9750 @ 2.6 GHz (12 CPUs) processor and 9 MB of RAM.

These two algorithms are compared based on PJ, JI, and JR. The q_{start} coordinates are (1.5) marked with a yellow circle and the q_{goal} coordinates are (5,6) marked red stars. In addition, the stop condition parameters are set when the edge node approaches the target at a distance of at least ρmin = 0.5.

The experiment result, as seen in figures 2 and 3, shows that the proposed function has successfully made tree branches reach ρmin = 0.5 in JI = 265 with PJ = 4.3, with the number of branches formed is 43 branches. Meanwhile, the other algorithms have not been able to reach the target node even in the 265th iteration. The remaining distance that still needs to be reached is 2.7442 points. The PJ formed is 1.4 with the number of branches formed being 73 branches. Thus, the proposed algorithm is vastly superior to the basic RRT algorithm.

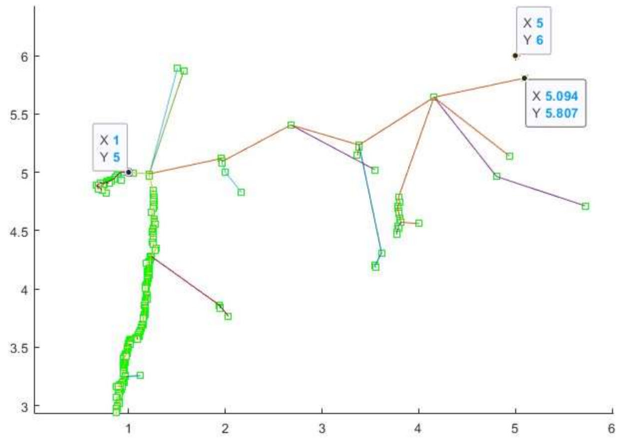

Fig. 1. Fig. 2. Trees with RRT Algorithms using Random Sample Growth Ratios as a Reference for Distance between Parent and Edge node.

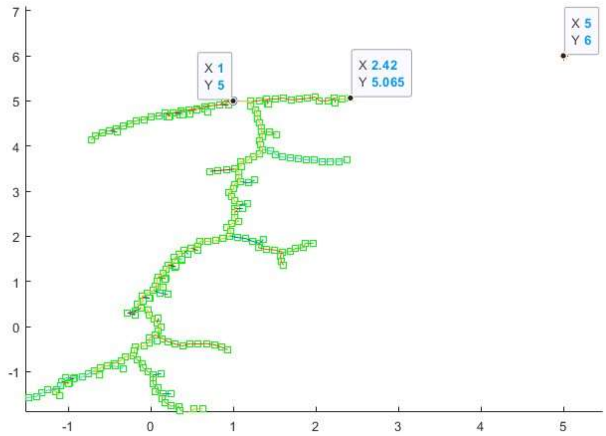

Fig. 3. Trees with RRT Algorithms using Threshold distance between Parental And Edge node

IV. CONCLUSION

This study succeeded in making the function coefficient of the distance function of the parent's node to the edge node, in order to achieve convergence of the motion direction of tree branches in the RRT. The distance function can trigger the acceleration of branch growth towards the target node. In addition, the setting of occupancy areas can also be a strategy for the extension and restriction of branch growth.

As for future works, it is necessary to make additional settings related to the potential change in the branch direction movement that approaches the target node but has not implemented an optimal restriction of branch length yet.

REFERENCES

[1] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, "PQ-RRT*: An improved path planning algorithm for mobile robots," *Expert Syst. Appl.*, vol. 152, p. 113425, 2020.

[2] F. Baberg and P. Ogren, "Formation obstacle avoidance using RRT and constraint based programming," *SSRR 2017 - 15th IEEE Int. Symp. Safety, Secur. Rescue Robot. Conf.*, pp. 1–6, 2017.

[3] Y. J. Kim, J. H. Wang, S. Y. Park, J. Y. Lee, J. J. Kim, and J. J. Lee, "A RRT-based collision-free and occlusion-free path planning method for a 7DOF manipulator," *2014 IEEE Int. Conf. Mechatronics Autom. IEEE ICMA 2014*, pp. 1017–1021, 2014.

[4] J. Janos, V. Vonasek, and R. Penicka, "Multi-Goal Path Planning Using Multiple Random Trees," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 4201–4208, 2021.

[5] H. Lee, H. Kim, and H. J. Kim, "Planning and Control for Collision-Free Cooperative Aerial Transportation," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 1, pp. 189–201, 2018.

[6] X. Wang, J. Wei, X. Zhou, Z. Xia, and X. Gu, "Dual-Objective Collision-Free Path Optimization of," vol. 6, no. 4, pp. 6353–6360, 2021.

[7] D. Connell and H. M. La, "Dynamic path planning and replanning for mobile robots using RRT," *2017 IEEE Int. Conf. Syst. Man, Cybern. SMC 2017*, vol. 2017-Janua, pp. 1429–1434, 2017.

[8] A. H. Qureshi and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning," *Auton. Robots*, vol. 40, no. 6, pp. 1079–1093, 2016.

[9] J. W. Woo, J. Y. An, M. G. Cho, and C. J. Kim, "Integration of path planning, trajectory generation and trajectory tracking control for aircraft mission autonomy," *Aerosp. Sci. Technol.*, vol. 118, p. 107014, 2021.

[10] W. Gong, "Probabilistic model based path planning," *Phys. A Stat. Mech. its Appl.*, vol. 568, p. 125718, 2021.

[11] L. Ye, J. Duan, Z. Yang, X. Zou, M. Chen, and S. Zhang, "Collision-free motion planning for the litchi-picking robot," *Comput. Electron. Agric.*, vol. 185, no. 483, p. 106151, 2021.

[12] Y. Zhou, E. Zhang, H. Guo, Y. Fang, and H. Li, "Lifting path planning of mobile cranes based on an improved RRT algorithm," *Adv. Eng. Informatics*, vol. 50, no. June, p. 101376, 2021.

[13] D. Gao, J. Luo, W. Ma, and B. Englot, "Online feedback motion planning for spacecraft obstacle avoidance using positively invariant sets," *Adv. Sp. Res.*, vol. 65, no. 10, pp. 2424–2434, 2020.

[14] X. Wang, X. Luo, B. Han, Y. Chen, G. Liang, and K. Zheng, "Collision-free path planning method for robots based on an improved rapidly-exploring random tree algorithm," *Appl. Sci.*, vol. 10, no. 4, 2020.

[15] J. Qi, H. Yang, and H. Sun, "MOD-RRT*: A Sampling-Based Algorithm for Robot Path Planning in Dynamic Environment," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7244–7251, 2021.

[16] C. Yuan, G. Liu, W. Zhang, and X. Pan, "An efficient RRT cache method in dynamic environments for path planning," *Rob. Auton. Syst.*, vol. 131, p. 103595, 2020.