2017 3rd International Conference on Control, Automation and Robotics

A Revised Gaussian Distribution Sampling Scheme Based on RRT* Algorithms in Robot Motion Planning

Li Yuncheng
Beijing Key Lab of Computational Intelligence and Intelligent System
Beijing University of Technology, BJUT
Beijing, China
e-mail: Richard.lee.lyc@gmail.com

Shao Jie
Beijing Key Lab of Computational Intelligence and Intelligent System
Beijing University of Technology, BJUT
Beijing, China
e-mail: shaolongjie@sina.com

Abstract—**This paper introduces the basic Rapidly-Exploring Random Tree (RRT) and its basic modification Rapidly-Exploring Random Tree star (RRT*), which is not only the extension of RRT, but also a widely applied algorithm because of the properties of asymptotically optimal path regardless of any obstacles, whereas the limitation to achieve optimal path has a slow convergence rate. As a result, it costs too much memory and time due to a large number of iterations, so we propose a method that should change the sampling scheme from random distribution sampling to Gaussian distribution sampling to overcome this limitation. In order to apply the improved algorithm in robot arms or manipulators motion planning, we extend the RRT* to simulate in higher dimensional spaces, the planner is implemented in 3D workspace. Finally we also revise the Gaussian distribution to suit the practical environment.**

Keywords-RRT*; motion planning; Gaussian distribution

I. INTRODUCTION

Motion planning for robot arms and manipulators with a number of degrees of freedom (DoF) has to complete collision-free trajectory through efficient approaches. The most important part of the approaches is the planning algorithm. With the development of robot motion planning, sampling-based algorithms draw extraordinary amount of attention. Arguably, the most widely-used two kinds of sampling are Probabilistic Roadmaps (PRM) [1], [2] and Rapidly-Exploring Random Tree (RRT) [3] based on planning alogrithm. RRTs [4], [5] are more suitable for high-dimensional spaces and nonlinear dynamic environments. Moreover, the RRTs are employed and demonstrated on various experimental robotic platforms [6], [7].

With the application of these algorithms, the quality of the solution is more important. We may concentrate on the solution paths of less cost such as whether it is the most optimality plan, the shortest length, the shortest time consuming, the lowest runtime, the most reasonable gaps to obstacles [8]. Most of the researchers do some relevant work depend on heuristics searching. Kuwata proposed a case in 2009 that computation time with heuristics could improve the efficiency because that a feasible path is found quickly [9]. Baris Akgun and Mike Stilman proposed two sampling heuristics in 2011 that improve RRT*, one is sampling bias to accelerate cost decrease in the spaces, the other is a simple node-rejection criteria to increase efficiency [10].

As mentioned above, no matter what variants of this algorithm, the baseline RRT play an critical leading role, as well as RRT*, which was introduced by Sertac Karaman and Emilio Frazzoli in 2010 [11], who went through introducing path cost and optimality. However, RRT*s perform so many iterations from initial state to every state in the planning domain. Not only the weakness, there exists some other constraints like its slow rate of convergence in finding the optimal path solution and its requirements of high memory. Considering all the inefficiency above, P-RRT* was proposed that incorporate APF (Artificial potential fields) into RRT* to speed up convergence rate toward a solution. [12]

In my paper, we propose a method that utilize the Gaussian distribution with the mean value in the target and adjustable standard deviation to the sampling. This new adding model is not only simplified and concise, but also effective to enhance convergence rate, we can see in the simulation section IV.

The remainder of this paper is organized as follows. Section II addresses the related work about RRT*, Sect.III applies the Gaussian distribution sampling compared to random sampling. Sect.IV provides experimental evidence to the effectiveness of the theoretical algorithms. Sect.V introduces the conclusions and puts forward some future work in the research domain.

II. BASIC RRT*

RRT was developed to solve the nonholonomy and dynamic constrains (Kuffner and LaValle, 2000). RRT algorithm is a typical algorithm based on tree structure, it is suitable to solve the kinematics and path planning problems due to its executable curve simulated through kinematics and dynamics in pairs of nodes, instead of arbitrary geometry line. The critical point is to bias toward the space, connect pairs of nearby nodes to the tree and generate to explore iteratively. RRT* is a variant of RRT that has the asymptotic optimality property, a concept introduced in RRT* is near neighbors within a certain radius of a node. Nodes found using Euclidean distance as the cost metric are closer to x_{rand} . The details are shown in Algorithm RRT*, another concept is the

Rewire model to calculate the most saving cost of path to cut off the original redundant edges.

Algorithm RRT*

RRT*$((V,E),K)$

for $i=1,...,K$ do

  $x_{rand} \leftarrow Random\_state()$

  $X_{near} \leftarrow Near(V, x_{rand})$

  $(x_{\min}, \sigma_{\min}) \leftarrow Choose\_Parent(X_{near}, x_{rand})$

  if Collision_Free(σ) then

    $V \leftarrow V \cup \{x_{rand}\}$

    $E \leftarrow E \cup \{(x_{\min}, x_{rand})\}$

    $(V,E) \leftarrow \mathrm{Re}wire((V,E), X_{near}, x_{rand})$

$return T = (V,E)$

Choose_Parent(X_{near},x_{rand})

minCost $\leftarrow \infty$; $x_{\min} \leftarrow$ NULL; $\sigma_{\min} \leftarrow$ NULL

for $x_{near} \in X_{near}$ do

  $\sigma \leftarrow Steer(x_{near}, x_{rand})$

  if Cost(x_{near})+Cost(σ)<minCost then

    minCost $\leftarrow$ Cost(x_{near})+Cost(σ)

    $x_{\min} \leftarrow x_{near}; \sigma_{\min} \leftarrow \sigma$

return($x_{\min}, \sigma_{\min}$)

Rewire((V,E), X_{near},x_{rand})

for $x_{near} \in X_{near}$ do

  $\sigma \leftarrow$ Steer(x_{rand}, x_{near})

  if Cost(x_{rand})+Cost(σ)<Cost(x_{near}) then

    if Collision_Free(σ) then

      $x_{parent} \leftarrow Parent(x_{near})$

      $E \leftarrow E \backslash \{x_{parent}, x_{near}\}$

      $E \leftarrow E \cup \{x_{rand}, x_{near}\}$

$return(V,E)$

The algorithm's workflow is as follows, RRT* create a searching tree contained vertexs and edges represented as V , E , and K means the iterations. we firstly sample randomized, denoted as x_{rand} like in RRT algorithm, then we use the *Near* function as follow equation. The $\|x-x'\|$ is the distance of vertices in the tree, d is the dimension of the space, and γ is a constant, n denote as the iteration times

$$Near(V,x) = \{x' \in V : \|x-x'\| \le \gamma (\frac{\log n}{n})^{1/d}\} \quad (1)$$

Next, we call a function Choose_Parent to find an alternative for a parent node to x_{rand} . Choose_Parent's inputs are the set X_{near} and x_{rand} , and the outputs are $x_{\min}$ and $\sigma_{\min}$. In this function, through all the nodes x_{near} in set X_{near}, we compare the cost between x_{near} and x_{rand} to find the minimum cost , then denoted as $x_{\min}$, $\sigma_{\min}$ is the minimum path cost between $x_{\min}$ and x_{rand} . We use function Steer to return a path σ to connect x_{rand} and x_{near} .

Then we check the $\sigma_{\min}$ after $x_{\min}$ is found, if the path is obstacle free, add this randomized sampling node x_{rand} to the tree and connect $x_{\min}$ to x_{rand} , which are the current parent nodes with vertex and edge.

However, edges are not the shortest path cost from the root of the tree to the vertex, so we need function Rewire to remove the "redundant" edges, we need to make sure the minimum cost path from the basic root to improve the efficiency.

Then using the function Rewire to connect x_{rand} with each remaining node in the set X_{near} of near nodes. If the path through x_{near} has lower cost than the current path, then the x_{near} is rewired to x_{rand} by connecting this two nodes, and delete the edge of its current path.

III. GOAL BIAS SCHEM OF GAUSSIAN DISTRIBUTION

RRTs and RRT*s seek most saving cost growth of the explored area when x_{rand} has generated in the unexplored area. Whereas we can replace the $Random\_state()$ by another improved planner. As we know, A reasonable sampling nodes should be distributed as there are seldom sampling nodes in the non-obstacle region or non-goal region, and there are large sampling nodes near obstacle and goal region to enhance searching convergence rate. Based on this problem, we give up keeping to generate nodes randomly with the global searching characteristic. Instead, sampling property should fall into local optimum, this paper adopts the standard deviation variable of Gaussian distribution to replace the original random distribution. We propose Gaussian Distribution $N(x;\mu,\sigma)$, with the mean value μ and standard deviation σ . The probability density function is

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

The expected μ determines the distribution location, the standard deviation σ determines the range of distribution or the level of focus. Probability density curve can be seen in Fig. 1, when the standard deviation of Gaussian distribution is small, the probability density near the mean value relative to other places is more precipitous, and when the Gaussian Distribution standard deviation increases gradually, the probability density curve tends to smooth, and observed much like random distribution. Inspired by this characteristic, this paper will improve the original RRT random sampling distribution, we finally change the random distribution with the standard deviation variables of Gaussian Distribution.

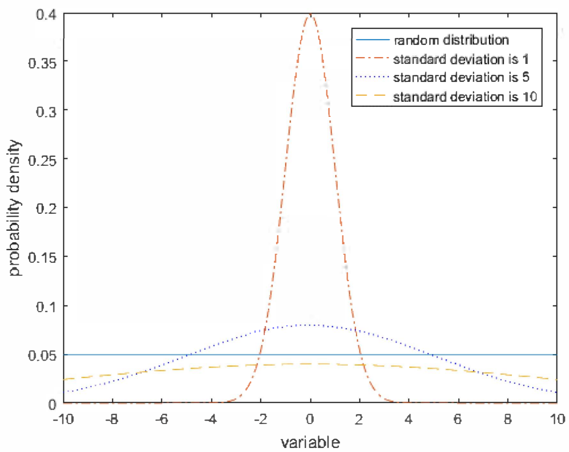
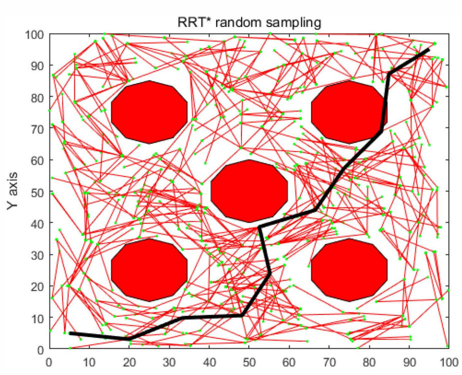


Figure 1. The distribution $N(x;\mu,\delta)$ with μ =0 and various values of σ

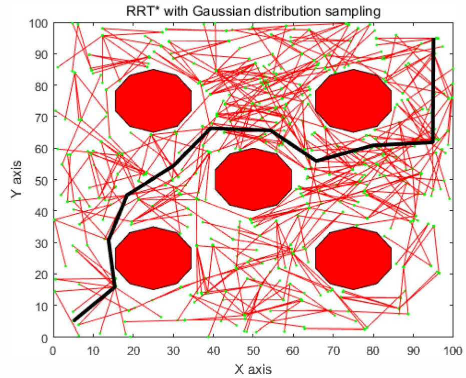
We set the mean value of the Gaussian Distribution almost close to the target position, when the standard deviation is small relatively, the sampling nodes are distributed near the goal region, which will rapidly lead the RRT to approach the target position. However, when there are large amounts of obstacles and complicated environment, this searching tree would like to be attracted by the local optimal points, as a result, it leads a more time cost or cannot achieve the planning in a limited time. On the contrary, when the standard deviation is increasing, the sampling nodes distributed normally and uniformly, the effect behaves more same as the original RRT random sampling method. The detailed experiments will show in next section.

IV. SIMULATION AND RESULTS

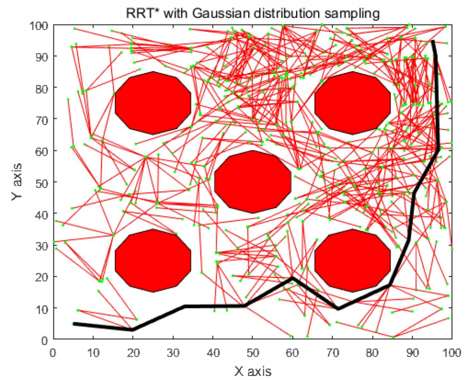
In Sect.II, we introduce the asymptotic optimality RRT*, in Sect.III we present a Gaussian Distribution standard deviation variable sampling method to replace the original random sampling in RRT* to enhance the convergence speed. The new sampling method is an alternative approach to refine the original path in global area of the environment. We test this differences with RRT* in 2D and 3D workspaces.



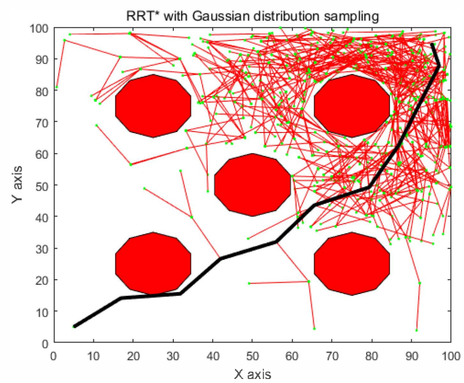
(a) Random sampling



(b) σ=80



(c) σ=50



(d) σ=20

Figure 2. random sampling contrast with Gaussian distribution sampling in 2D space, x_{init} is the initial position at (5, 5), the goal position at (95, 95), the red circles represent obstacles, the red lines are the growth lines between nodes, the black line represent final path, the searching space is a square 100*100. The image(a) using no goal bias scheme, a goal bias of σ =80, 50, 20 are using the goal bias scheme of deviation variable of Gaussian distribution. Planning requires 14.87s, 13.02s, 10.26s 8.43s, respectively.

Through the improvement, the Gaussian Distribution reduces the time consuming and obtains better solutions compared to non-biased search. Random distribution sampling and standard deviation variable distribution is shown in the following Fig. 2.

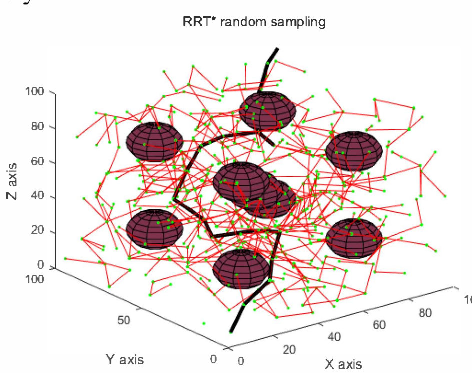
In Fig. 3 (b) to (d), when standard deviation σ is decreasing, the sampling nodes are sampled near the goal region, that the searching nodes approaching unexplored spaces is decreasing, as a result, the convergence rate would be increased owing to the standard deviation variable method.

Considering the robot arms or manipulators' work space, we also expand the searching space into 3D workspace, the test contents in Fig. 3 are the same as above in 2D workspace, we also could obtain the same results.

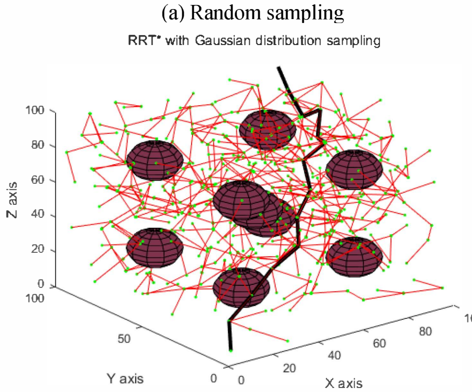
Through the improvement of sampling nodes shown in Fig. 2 and Fig. 3, we should pay attention to one aspect, the Gaussian Distribution probability density curve distribute symmetric along the mean value and extend infinitely, whereas in the practical application, the searched workspace is always bounded area and asymmetrical along the goal region. Considering this problem, we revise the Gaussian Distribution curve to adapt to this problem, the revised curve is shown in Fig. 4. The calculated model is as follows:

$$f'(x \mid \mu,\sigma)=\begin{cases}0, x \le x_{lower} \\ \dfrac{f(x \mid \mu,\sigma)}{\phi(x_{upper})-\phi(x_{lower})}, x_{lower} < x < x_{upper} \\ 0, x \ge x_{upper}\end{cases} \quad (3)$$

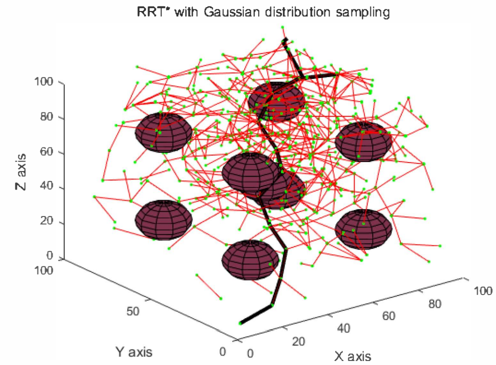
In this revised expression, x is variable, $f(x \mid \mu,\sigma)$ is unrevised Gaussian Distribution probability density function, $f'(x \mid \mu,\sigma)$ is the revised probability density, $\phi(x_{upper})$ and $\phi(x_{lower})$ is correspond to the upper and lower boundary cumulative probability of unrevised Gaussian distribution, respectively



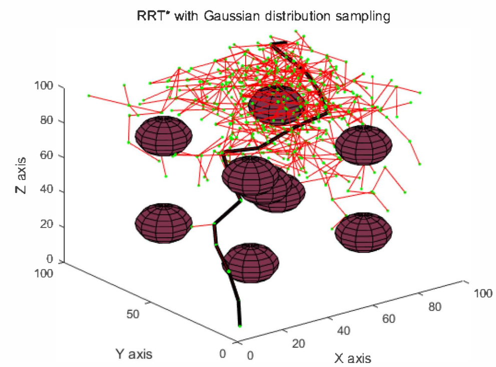
(a) Random sampling



(b) σ=80



(c) σ=50



(d) σ=20

Figure 3. random sampling contrast with Gaussian distribution sampling in 3D space, x_{init} is the initial position at (5, 5,5), the goal position at (95, 95, 95), the purple sphere globe represent obstacles, the red lines are the growth lines between nodes,the black line represent final path, the searching space is a square 100*100*100. The image(a) using no goal bias scheme, a goal bias of σ =80, 50, 20 are using the goal bias scheme of deviation variable of Gaussian distribution. Planning requires 17.89s, 16.42s, 11.14s 10.33s, respectively.

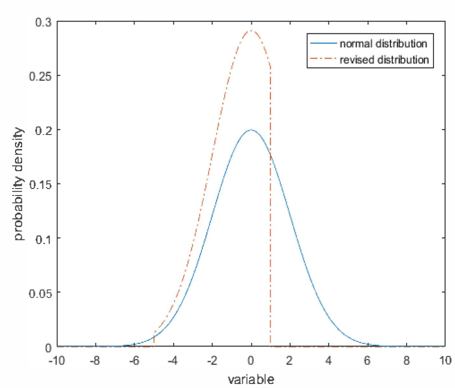


Figure 4. A revised Gaussian distribution to adapt practical sampling

In Fig. 4, when the x go beyond the upper and lower boundary, the probability density is zero, when in the inner boundary, the revised probability density will suit the practical environment to sampling.

V. CONCLUSION

Basic RRT* gain a asymptotic optimality but is not a memory efficient algorithm and its slow convergence rate, high memory and costly time will destroy the efficiency, an improved method that add a Gaussian distribution sampling scheme to bias to the goal region could improve the convergence rate, for better practical sampling, we revised the original Gaussian distribution curve to suit the practical environment. This improved RRT* sampling scheme also could apply in RRT*s and RRTs adaptively. Finally we do a series simulation in 2/3D workspace. There are also many other method to improve the efficiency both in local heuristic growing scheme and in random sampling, in the future proceedings, we could apply this scheme to the online application because of the improvement of the convergence rate.

REFERENCES

[1] Kavraki, p. Sevestka, J. Claude Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration space,” in IEEE ICRA, 1996, pp. 566-580

[2] L. E. Kavraki, M. N. Kolountzakis, and J. C. Latombe. “Analysis of probabilistic roadmaps for path planning.” IEEE Transactions on Robotics and Automation, 14(1):166-171,1998

[3] J. Kuffner and S. M. LaValle. “RRT-connect: An efficient approach to single-query path planning.” In Proceedings of the IEEE International Conference on Robotics and Automation, 2000

[4] S.M. LaValle and J. J. Kuffner. “Randomized kinodynamic planning.” International Journal of Robotics Research, 20(5): 378-400, May 2001

[5] S.M. LaValle. Planning Algorithms. Cambridge University Press, 2006

[6] Teller, M.R. Walter, M. Antone, A. Correa, R. Davis, L. Fletcher, E. Frazzoli, J. Glass, J.P. How, A.S. Huang, J. Jeon, S. Karaman, B. Luders, N. Roy, and T. Sainath. “A voice-commandable robotic forklift working alongside humans in minimally-prepared outdoor environments.” In Proceedings of the IEEE/RSJ International Conference on Robotics and Automation, 2010

[7] Shkolnik, M. Levashov, I.R. Manchester, and R. Tedrake. “Bounding on rough terrain with the LittleDog robot.” Submitted for publication, 2011.

[8] NikolausVahrenkamp, TamimAsfour and RudigerDillmann Efficient Motion and Grasp Planning for Humanoid Robots

[9] Kuwata,J.Teo, G. Fiore, S. Karaman, E Frazzoli, and J.P. How. ”Real-time motion planning with applications to autonomous urban driving.”IEEE Transaction on Control Systems,17(5):1105-1118, 2009

[10] BarisAkgun and Mike Stilman “Sampling Heuristics fot Optimal Motion Planning in High Dimensions” in IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011

[11] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning.” Int. J. Rob. Res, vol. 30, pp. 846-894, June 2011.

[12] Ahmed Hussian Qureshi, Yasar Ayaz “Potential function based sampling heuristic for optimal path planning” Journal Auton Robot(2016)40:1079-1093