Proceedings of the 2019 IEEE/ASME
International Conference on Advanced Intelligent Mechatronics
Hong Kong, China, July 8-12, 2019

An Optimal Motion Planning Method of 7-DOF Robotic Arm for Upper Limb Movement Assistance

Zemin Liu, Qingsong Ai, Yaojie Liu, Jie Zuo, Xiong Zhang, Wei Meng, *Member, IEEE*, Shane Xie, *Senior* Member*, IEEE*

Abstract**—Assistive robotic arm is crucial alternative resource for people disabled or injured in the upper limbs, which enable them to complete basic living tasks independently. Thus, an extremely accurate motion planning for robotic arm needs to be applied to improve assistive performance. Rapidly-Exploring Random Tree Star (RRT*) is one of the most representative methods, however, this method has great limitations due to the tedious iteration process while planning. In this study, the potentials guide sampling based-on RRT* (PGS-RRT*) approach is introduced through combination with artificial potential fields (APF) as an expansion of RRT* algorithm. A revision of repulsive potential force's formula in APF has been applied into sampling process of RRT*. The samples during motion planning is gathered through the optimization of potentials formulations. Specifically, the basic potential function give each sample an offset oriented to goal. Experiments in 2D and 3D environments and simulations on KUKA LBR iiwa 7 prove that the PGS-RRT* method is able to find an optimal path in a short time, which highlights the application prospect on robots with a number of degree of freedom (DOF) in patient's daily life assistance.**

I. Introduction

Recently, an increasing number of people have got injured in their upper limbs due to various internal and external causes, such as improper exercise, traffic accidents, strokes, etc. resulting in inconvenience of upper limbs movement [1]. Assistive rehabilitation robots are of great significance to help people with disabilities or injuries in the upper limbs to complete basic living tasks. The 7-DOF robotic arm is competent to be an assistive robot to work in complex environments because its redundancy can satisfy the demands of the task with more joint combinations [2, 3]. However, it's a fatal issue to figure out the obstacle avoidance and motion planning in 3D space for robot arms with a number of degree of freedom [4].

Sampled based algorithms are designed for solving the path planning problem faster, which is occurring by sampling the configuration[5]. Probabilistic RoadMap (PRM) and Rapidly-exploring Random Tree (RRT) are the most typical sampling-based motion planning algorithms [6, 7]. PRM is a multiple-query planner which assumes that many solutions could be obtain for a same situation, when it involves high dimensional configuration, PRM would make the roadmaps become very complex and it won't be suitable for robot arm motion planning [8, 9]. As a single-query planner, RRT could work normally in high dimensional space, one of the drawbacks of RRT algorithm is that it randomly samples around the workspace, specifically it cannot guarantee an optimal path [10].

In order to break through the limitations of RRT algorithm[11], many algorithms have emerged in recent years. RRT-connect utilizes a simple greedy algorithm that tries to connect two trees, one from initial configuration and the other from goal[12, 13]. Transition-based RRT (T-RRT) constructs a continuous cost spaces and make the planner follow valleys and saddle points to search low cost solution paths [14]. RRT* is a variant which includes the rewiring process, it renews the exploring tree when new nodes are added and eventually tends the optimal solution. Nevertheless, the effectiveness of the solution is limited by a great deal of iterations and large memory [10, 15]. A revised gaussian distribution sampling scheme is added to RRT* to bias the sample points to improve the convergence rate[16]. Potential Guided Directionalized -RRT*(PGD-RRT*) leads random samples with artificial potential fields(APF) to search goal quickly compared to original RRT* but it cannot guarantee convergence to optimal path [17]. Potential Function Based-RRT* (P-RRT*) also combines APF Algorithm with RRT*. However it only preserves the quadratic variation and discard the samples closed to obstacle [18, 19].

As concluded from the previous studies, there are still some problems to accomplish the motion planning of robotic arm. In this paper, an potentials guide sampling based-on RRT*(PGS-RRT*) algorithm through modification of the attractive and repulsive potential functions in basic APF and application of this modification into the sampling process of RRT* has been proposed, which leads the samples to get close to goal point. After modification, the samples scattering around previous begin to approach towards goal, and the exploring tree obtain a clear target to grow branches, otherwise a great deal of useless branches would be generated to cause large consumption of memory and time and make the complication of path. The rest of the paper is arranged as

*Research funded by National Natural Science Foundation of China under grant 51705381 and Excellent Dissertation Cultivation Funds of Wuhan University of Technology (2018-YS-054), also supported by the UK EPSRC Standard Research Scheme under grant EP/S019219/1.

Z. Liu is with School of Information Engineering, Wuhan University of Technology, 122 Luoshi Road, Wuhan, China (zeminliu@whut.edu.cn).

Q. Ai is with the School of Information Engineering, Wuhan University of Technology, 122 Luoshi Road, Wuhan, China (qingsongai@whut.edu.cn).

Y. Liu is with School of Information Engineering, Wuhan University of Technology, 122 Luoshi Road, Wuhan, China (yaojieliu@whut.edu.cn).

J. Zuo is with the School of Information Engineering, Wuhan University of Technology, 122 Luoshi Road, Wuhan, China (zuojie@whut.edu.cn).

X. Zhang is with School of Information Engineering, Wuhan University of Technology, 122 Luoshi Road, Wuhan (xueyelongyin@whut.edu.cn).

W. Meng is with the School of Information Engineering, Wuhan University of Technology, 122 Luoshi Road, Wuhan, China and the School of Electronic and Electrical Engineering, University of Leeds, Leeds, LS2 9JT, UK (email: weimeng@whut.edu.cn, w.meng@leeds.ac.uk).

S. Xie is with the School of Electronic and Electrical Engineering, University of Leeds, Leeds, LS2 9JT, UK (email: s.q.xie@leeds.ac.uk).

978-1-7281-2493-3/19/$31.00 ©2019 IEEE

follows: Section II introduces the basic RRT* and its work procedures, and Section III revises the attractive as well as repulsive potentials and applies it to RRT* to get new PGS-RRT* algorithm. Section IV provides stimulation and experiment results of PGS-RRT* and RRT* in 2D, 3D as well as application on KUKA LBR iiwa 7 robot arm. Section V concludes the paper and outlooks the future work.

II. Basic Rapidly-exploring Random Tree Star Algorithm

This section introduces basic Rapidly-exploring Random Tree Star (RRT*) algorithm which guarantees asymptotic optimality, i.e. if time is adequate, an optimal path could be always obtained through RRT* [20-22]. Compared to original RRT algorithm, RRT* has added two procedures called "*Near vertices*" and "*Rewire*". "*Near vertices*" means searching a number of nearest nodes within a certain radius of a node. "*Rewire*" calculates the cost of path from different nodes obtained in "*Near vertices*" to the new node so as to achieve lower cost path [23]. The details are shown in Fig. 1.

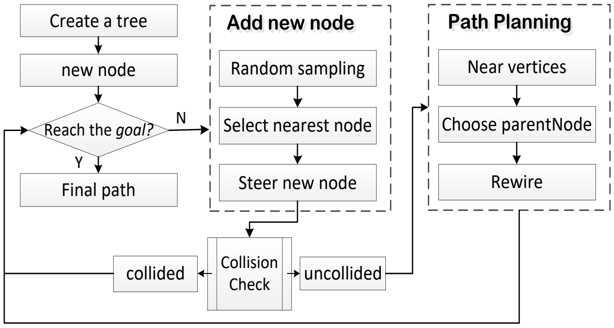


Figure 1. Program diagram of RRT*

RRT* create a exploring tree called T contained vertices and edges represented as V, E, and n means the number of iteration. During the main loop，firstly a random point x_{rand} would be attained in obstacle-free region X_{free}, then select the node $x_{nearest}$ in exploring tree T closest to x_{rand} and the tree generates a small step α to acquire new node x_{new} along the direction from $x_{nearest}$ to x_{rand}. Next, we need to check if this new node would break into obstacle region. If it's safe, search a set of node $X_{near} \subset T$ within a circle with a radius of r centered at x_{new}. The next step is calculate the cost from initial node to x_{new} passing all nodes $x_{near} \in X_{near}$ and select the smallest one as the parent node, as well as insert this node to exploring tree, then rewire new node to its parent node. If the tree reach goal or the iterations is ran out, the loop will stop. Finally we could obtain a path from the goal point to its parent to parent until the initial node, a near optimal path appears.

According to the theorem proofed by Karaman et.al[10], RRT* has the feature of asymptotic optimality which means the solution converge to the global optimum. However the precondition to optimality is that the time should be adequate and the running memory should be enough. There is urgent need to find an approach to increase convergence speed.

III. Potentials Guide Sampling Based-on Rapidly-exploring Random Tree Star

Although basic RRT* algorithm possess ability to find the optimal path to goal point, it's confronted with a severe issue that a large number of iterations and memory will be cost and high time will be consumed during this algorithm. Because the sampling process is random, the sample points scatter around workspace to lead exploring tree to grow along the direction of different samples, which cause the tree to expand a great deal of "redundant" branches which occupy large memory. Therefore we hope a method that could gather the samples around goal point, at the same time the tree would not be hindered by obstacles. Based on this surmise, we revise the basic formula of Artificial Potential Fields(APF) and utilize the notion of APF to guide the sampling process of RRT* to form a new algorithm called Potentials Guide Sampling based-on Rapidly-exploring Random Tree Star (PGS-RRT*).

A. Potentials Guide Sampling

APF is used to settle motion planning of low dimension robots because it may suffer local minimum problems i.e. it may be trapped in a local minimum and can't reach goal point[24, 25]. Before apply APF to sampling process, we should make some modifications. Firstly, the attractive potential of traditional APF is composed of quadratic component and conical component as seen in equation (1):

$$U_{att}(X)=\begin{cases}\lambda_1\rho^2\left(X,X_g\right),\ \rho\left(X,X_g\right)>\rho^* \\ \lambda_1\left[\rho\left(X,X_g\right)\rho^*-\left(\rho^*\right)^2\right],\rho\left(X,X_g\right)\le\rho^*\end{cases} \tag{1}$$

There is no need to keep two components like APF because overshooting wouldn't have been a burden, the application object of algorithm become sample from robot and the sample is allowed to surpass goal point. The attractive potential function can be expressed as :

$$U_{att}(X)=\lambda_1\rho^2\left(X,X_g\right) \tag{2}$$

In Eq. (2), $\rho\left(X,X_g\right)$ represents the distance of sample to goal, λ_1 is the attractive scaling factors scaling the magnitude of attractive potential. From this formula, we can draw a conclusion that the potential energy reduces as the distance from sample to goal decreases. So the lowest potential in the whole workspace is goal point. The attractive force could be attained through the negated gradient of attractive potential in equation (3).

$$F_{att}\left(X\right)=-\nabla U_{att}(X)=-2\lambda_1\rho\left(X,X_g\right)\frac{d\rho\left(X,X_g\right)}{dX} \tag{3}$$

The variation of attractive force is similar to $U_{att}(X)$, the closer the sample to goal, the force smaller. It means that the sample point farther from goal need to move a larger distance along the direction of goal. Thus, the samples will approach close to goal rather than spread in the whole space.

In workspace, this force is enough to satisfy the assume in preface of this section, but if there are many obstacles in workspace, samples may guide the tree trapped in obstacle region and can't turn back to find new path, the situation become complex. In basic APF, the repulsive potential is represented as:

$$U_{rep}(X)=\begin{cases}\lambda_2\left(\dfrac{1}{\rho(X,X_0)}-\dfrac{1}{\rho_0}\right)^2, \rho(X,X_0)\le\rho_0\\ 0, \rho(X,X_0)>\rho_0\end{cases} \tag{4}$$

This traditional method can't solve the local minimum problem well, so we lead into a variable the distance of sample to goal to Eq. (4) and we can get a new repulsive potential:

$$U_{rep}(X)=\begin{cases}\lambda_2\left(\dfrac{1}{\rho(X,X_0)}-\dfrac{1}{\rho_0}\right)^2\rho^2(X,X_g), \rho(X,X_0)\le\rho_0\\ 0, \rho(X,X_0)>\rho_0\end{cases} \tag{5}$$

In Eq. (5), $\rho(X,X_0)$ is the distance between sample and obstacle, ρ_0 is the influence radius of obstacle on the sample, and λ_2 represents the repulsive scaling factors scaling the magnitude of repulsive potential. Similarly, the repulsive force could be obtained through the negated gradient of repulsive potential:

$$F_{rep}(X)=-\nabla U_{rep}(X)=\begin{cases}F_{rep1}(X)+F_{rep2}(X), \rho(X,X_0)\le\rho_0\\ 0, \rho(X,X_0)>\rho_0\end{cases} \tag{6}$$

$$F_{rep1}(X)=-2\lambda_2\left(\frac{1}{\rho(\mathrm{X},X_0)}-\frac{1}{\rho_0}\right)\frac{\rho^2(\mathrm{X},X_g)}{\rho^2(\mathrm{X},X_0)}\frac{\partial\rho(\mathrm{X},X_0)}{\partial X} \tag{7}$$

$$F_{rep2}=2\lambda_2\left(\frac{1}{\rho(\mathrm{X},X_0)}-\frac{1}{\rho_0}\right)^2\rho(\mathrm{X},X_g)\frac{\partial\rho(\mathrm{X},X_g)}{\partial X} \tag{8}$$

From equation (5) we can find that obstacle have an effect region, only in this area is sample under the influence of repulsive potential, beyond the area repulsive potential becomes zero. Besides, as sample gets close to obstacle the potential energy will increase sharply. The formulas following explain the repulsive force sample suffers in effect region. After leading into the distance between sample and goal, the repulsive is split into two parts. Fig. 2 show the detail situations of sample's force-suffering in effect area.

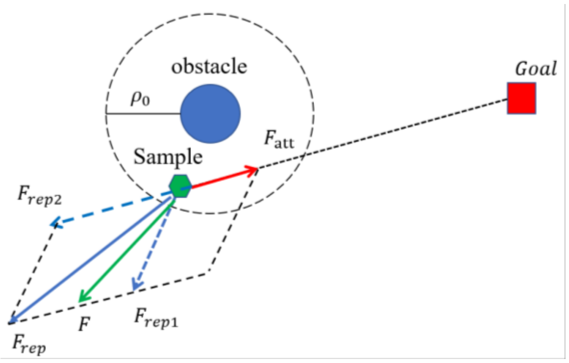


Figure 2. Force analysis of sample in obstacle effect area

From the force analysis of sample in Fig. 2, the repulsive force is composed of two components from obstacle and goal respectively, which insure the sample break away from effect region and guide the node along the direction bypass the obstacles at a larger angle. This modification could make sample out of obstacle's effect region and avoid local minimum problems efficiently.

B. Potentials Guide Sampling Based-on RRT*

After revision of attractive and repulsive potential functions in last section, we introduce the new functions into sampling process of RRT*. At the basis of original RRT* we add a procedure rightly after randomly sampling and form a new algorithm called Potentials Guide Sampling Based-on RRT* (PGS-RRT*). Fig. 3 describes the pseudocode of PGS-RRT*.

Algorithm I. Pseudocode of PGS-RRT*

```
$T=(V,E)$;
for $i=1, \dots, n$ do
    $x_{rand} \leftarrow Random\_State()$;
    $x_{prand} \leftarrow PGS(x_{rand})$ ;
    $x_{nearest} \leftarrow Nearest(T, x_{prand})$;
    $x_{new} = Steer(x_{nearest}, x_{prand})$;
    if $Collision\_Check(x_{new})$ then
        $X_{near} \leftarrow Near\_Vertices(T, x_{new})$;
        $x_{\min} \leftarrow Choose\_Parent(T, X_{near}, x_{new})$;
        $T \leftarrow InsertNode(T, x_{\min}, x_{new})$;
        $(V,E) \leftarrow Rewire((V,E), x_{new}, x_{\min})$;
return $T \leftarrow (V,E)$
```

Algorithm I show the process of PGS-RRT*, $x_{prand} \in X_{free}$ represents new samples that guided by potentials, i.e. new samples would deviate previous position and make a step along the direction of attractive potential decreasing as the random sample approaches get close to the goal region. In this new method, x_{prand} has taken place of x_{rand} to perform the following procedures. x_{prand} calculated through PGS function which is also called Potentials Guide Sampling Planning.

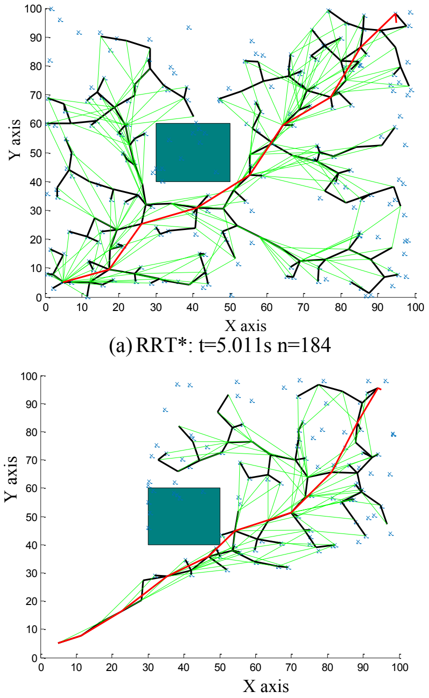
Algorithm II. Potential Guide Sampling(PGS) Planning

```
$U_{att} \leftarrow AttractivePotential(X_{goal}, x_{rand})$;
$\rho_{\min} \leftarrow DistanceObstacle(x_{rand}, X_{obs})$;
if $\rho_{\min} < \rho_0$ then
    $U_{rep} \leftarrow RepulsivePotential(X_{obs}, x_{rand}, X_{goal})$;
    $\overline{F} \leftarrow Compute(U_{att}, \lambda_1, U_{rep}, \lambda_2)$;
else
    $\overline{F} \leftarrow Compute(U_{att}, \lambda_1)$;
    $x_{prand} \leftarrow x_{rand} + \overline{F}$ ;
return $x_{prand}$ ;
```

Algorithm II shows the process of PGS function which is set up through the attractive and repulsive force formulas in last section, This procedure firstly constructs attractive potential in equation (2) and obtains the minimum distance $\rho_{\min}$ between the sample point and the obstacle. Then compare the distance $\rho_{\min}$ and the effect radius of obstacles ρ_0 , if $\rho_{\min}$ is smaller, combine equation (5) to set up repulsive potential, the next step is to figure out the total force seen in fig. 2, otherwise directly calculate the attractive force. Finally, according to the magnitude of the force, add an corresponding offset to x_{rand} and figure out x_{prand} . By means of PGS function, an offset added to sampling point oriented to goal point, so that lots of scattering samples could be avoided, otherwise the growth direction of the tree would deviate the optimal path and the planning time would increase greatly.

IV. Experiment Results and Analysis

In this section, we simulate RRT* and PGS-RRT* on MATLAB to compare which is better in motion planning. In order to confirm the performance of PGS-RRT*, the parameters and the configuration space remain the same, but since the sampling-based algorithm shows diverse results because of its randomness, each experiment need to run 50 times and take the average. The attractive scaling factors λ_1 is 0.35, and the repulsive scaling factors λ_2 is $0.3\rho_0$. The experiments can be divided into three groups: (1) in 2D workspace of four different size; (2) in 3D workspace; (3) applied into robotic arm KUKA LBR iiwa 7 in V-rep.

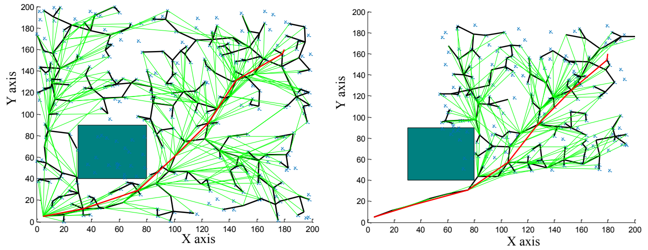


(a) RRT*: t=5.011s n=184

(b) PGS-RRT*: t=1.293s n=92

Figure 3. Near-optimal path conducted by RRT* and PGS-RRT* in a 2D space, the searching space is a square 100*100, the initial position is at (5, 5), the goal position at (95, 95), the deep green rectangle is obstacle, small marker '×' represents sampling point, and the black line describes the rapidly-growing exploring tree, the green is the lower cost path after *Rewire* procedure, and the red path represents the final route.

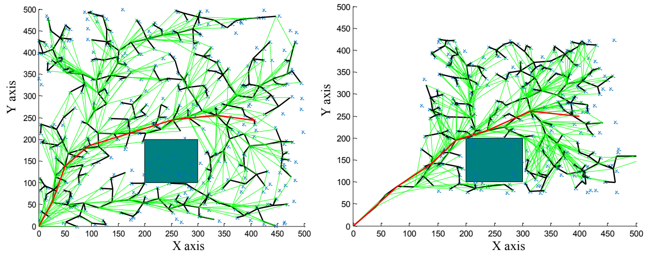
Fig. 3 shows the planning of these two methods. It can be seen that both the proposed PGS-RRT* and RRT* were able to obtain a path from initial point to goal in red line, but RRT* obviously needs more iterations and running time. Furthermore, Fig.3(a) indicates that the sampling points of RRT* are spread throughout the whole space which results in the nodes of exploring tree expanding around, contrarily in Fig.3(b) the sampling points of PGS-RRT* are close to goal point because of the influence of potential and the path is more smooth. The trajectory of robotic arm used to be smooth curve so that the movements will be more flexible, if the angle of trajectory is too large, the robotic arm would get injured because there’s not enough time to adjust velocity.



(a) RRT*: t=10.308s n=268

(b) PGS-RRT*: t=4.210s n=159

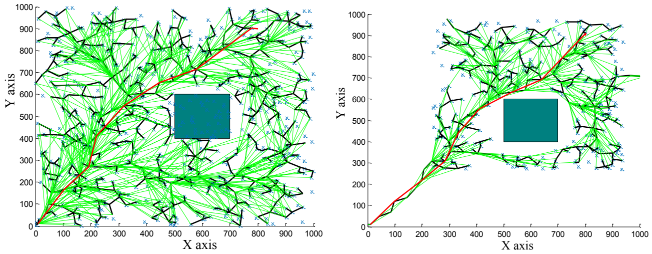
Figure 4. Near-optimal path conducted by RRT* and PGS-RRT* in a 2D space 200*200, the initial position is at (5, 5), the goal position at (180, 180)



(a) RRT*: t=17.904s n=379

(b) PGS-RRT*: t=9.621s n=292

Figure 5. Near-optimal path conducted by RRT* and PRS-RRT* in a 2D space 500*500, the initial position is at (0, 0), the goal position at (400, 250)



(a) RRT*: t=53.783s n=667

(b) PGS-RRT*: t=16.682s n=394

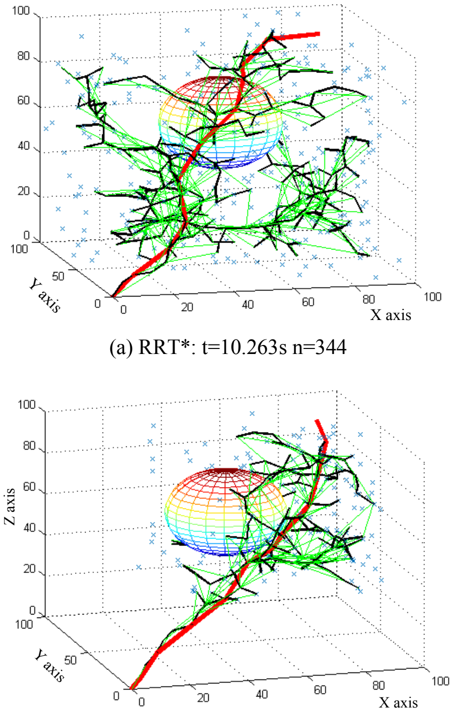
Figure 6. Near-optimal path conducted by RRT* and PRS-RRT* in a 2D space 1000*1000, the initial is position at (0, 0), the goal position at (800, 900)

In order to protrude the superiority of PGS-RRT*, we increase the size of workspace to 200*200, 500*500, 1000*1000 and the experimental results are shown in Fig. 4, 5 and 6. Table I summaries the statistical data of iterations and running time. We can find that the average time and iterations all become larger with the space get bigger in both methods, furthermore, in the four different space, the performance of PGS-RRT* is all obvious superior to RRT* in terms of the average time and average iterations.

TABLE I. The iterations and running time of 4 workspace

Environment	Methods	*Average Time(s)*	*Average iterations*
2D(100*100)	RRT*	5.011	184
	PGS-RRT*	1.293	92
2D(200*200)	RRT*	10.308	268
	PGS-RRT*	4.210	159
2D(500*500)	RRT*	17.904	379
	PGS-RRT*	9.621	292
2D(1000*1000)	RRT*	53.783	667
	PGS-RRT*	16.682	394

Fig. 4, 5 and 6 show the comparison of RRT* and PGS-RRT* in searching path in larger workspace, we can easily observe that the graph becomes more and more complicated as space gets larger, but it's obvious that the samples PGS-RRT* needs are less than RRT* in a same level. Similar to Fig. 3, samples in Fig. 4(a), 5(a) and 6(a) are full of space and exploring tree grows everywhere, contrarily samples in (b)s have shifted a distance pointing to goal. Thus the target area of tree's growing is more specific which would lead the nodes to steer in shorter path rather than randomly in the whole space. Table I reveals the iterations and running time of these two methods in four workspace quantitatively.



(a) RRT*: t=10.263s n=344

(b) PGS-RRT*: t=4.313s n=218

Figure 7. Near-optimal path conducted by RRT* and PGS-RRT* in a 3D space, the searching space is a square 100*100*100, the initial position is at (0, 0, 0), the goal position at (90, 90, 90), the sphere represents the obstacle, small marker 'x' represents sampling point, and the black line describes the rapidly-growing exploring tree, the green is the lower cost path after Rewire procedure, and the red path represents the final route.

We can see that when the workspace is small, the difference of consuming time between these two methods is no more than 10 second, but when the space becomes 1000*1000, the running time of RRT*(53.783s) is close to 1 minute, which is far more than PGS-RRT*(16.684s). This phenomenon is enough to indicate the high efficiency of PGS-RRT*.

Considering that the working space of the robot arm is generally in 3D space, we extend this algorithm into 3D space 1000*100*100. The results are shown in Fig. 7. According to average of 50 experiments, iterations and running time required for PGS-RRT* (t=4.313s n=218) is less than the RRT*(t=10.263s n=344) algorithm, and the path in (b) is smoother. The samples' distribution is similar to 2D space , the samples in PGS-RRT* have been offset along the target direction to a certain extent, the potential of goal is lowest and the potential of obstacle is high, so the exploring tree could grow down the direction of potentials decreasing to goal in a smooth manner.

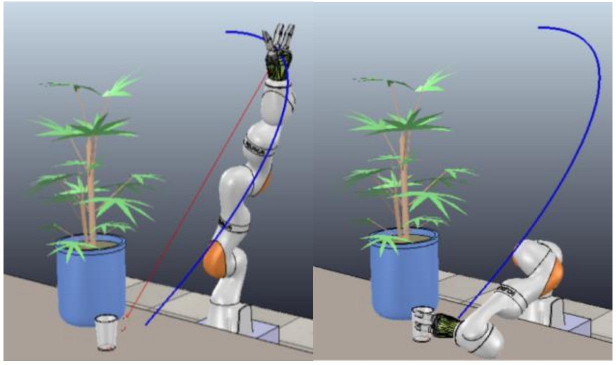


Figure 8. running time = 12.11s, total time =18.17s
RRT* planning experiment on robot arm (KUKA LBR iiwa 7 R800) in V-rep, the task is to plan a route from upright state(initial position) to the cup(goal position) on the table for robot arm, the plant represents obstacle, the blue line describes the final route

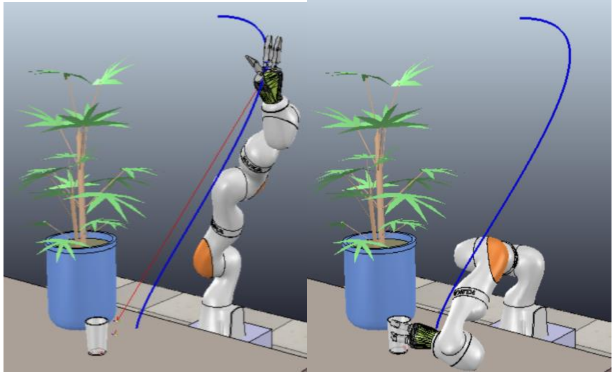


Figure 9. running time = 3.86s total time =11.51s
PGS-RRT* planning experiment on robot arm (KUKA LBR iiwa 7 R800) in V-rep

The algorithm needs to be applied to the assistive rehabilitation robot to help people with disabilities or injuries in the upper limbs accomplish simple daily tasks. Thus, we set up a simulation scene on V-rep and use KUKA LBR iiwa 7 to conduct a simple movement, planning a path from the upright state to the cup on the table as seen in Fig.8. Dummy is a tool as a reference frame in V-rep environment, so we can obtain the space coordinates of the robot arm, cup and plant directly, which represent the initial position, goal position and obstacle. Similar to previous simulation, our task is to plan a route from initial position to goal position for robot arm. Hence there is feasibility for us to put this method PGS-RRT* into motion planning of robotic arm in V-rep.

Fig. 8 shows that path calculated using RRT*, which is represented by the blue line, the planning time is 12.11s, the total time (including the arm moving time) is 18.17s; Fig. 9 of PGS-RRT* indicates the planning time is 3.86s and the total time is 11.51s. It's not hard to observe that the path in Fig.8 is a little longer than Fig.9, thus the moving time in RRT* is a little bit more than PGS-RRT*. while the planning time of former takes about three times as much as the latter. Obviously it's because too many scattering samples make the path complicated and bring too many useless nodes and branches to the exploring tree. From the experimental results analysis, we can draw a conclusion that both of PGS-RRT* and RRT* are able to find a feasible path for 7-DOF robotic arm, but the efficiency of PGS-RRT* is much higher than RRT*.

V. Conclusion

In this study, we revised the function expressions of Artificial Potential Fields, and applied it to the sampling process in RRT*, we call the new algorithm PGS-RRT*. It guides sampling points to move close to the goal and make samples within the scope of obstacles to deviate away under the influence of potentials. The exploring tree would grow orienting the coordinate of goal rather than scatter around the workspace. This algorithm is suitable to be applied to high-dimensional spaces as well as helpful to decrease iterations and running time efficiently. This paper compares experiments in 2D and 3D environments, and applies this algorithm to 7-DoF robotic arm, indicating that the planning speed has been greatly improved.

In the future, the experiment will be carried out on the C++ platform to build more complex scenes. The algorithm of this paper is going to be applied to assistive rehabilitation robot, and integrated system will be constructed and more complex test experiments will be designed.

References

[1] A. Jain and C. C. Kemp, "EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots,* vol. 28, no. 1, pp. 45-64, Jan 2010.

[2] I. Kuhlemann, P. Jauer, F. Ernst, and A. Schweikard, "Robots with seven degrees of freedom: Is the additional DoF worth it?," in *International Conference on Control*, 2016.

[3] F. Cheng, J. Lee, A. Ajoudani, C. Zhou, and D. G. Caldwell, "RRT-based motion planning with sampling in Redundancy space for robots with anthropomorphic arms," in *IEEE International Conference on Advanced Intelligent Mechatronics*, 2017.

[4] R. Bordalba, L. Ros, J. M. Porta, and Ieee, "Randomized Kinodynamic Planning for Constrained Systems," in *2018 Ieee International Conference on Robotics and Automation*(IEEE International Conference on Robotics and Automation ICRA, 2018, pp. 7079-7086.

[5] H. Choset *et al.*, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. 2005.

[6] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[7] M. Du *et al.*, "RRT-based motion planning algorithm for intelligent vehicle in complex environments," vol. 37, no. 4, pp. 443-450, 2015.

[8] L. Kavraki, P. Svestka, and M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. Unknown Publisher, 1994.

[9] Y. Dong, E. Camci, and E. Kayacan, "Faster RRT-based Nonholonomic Path Planning in 2D Building Environments Using Skeleton-constrained Path Biasing," *Journal of Intelligent & Robotic Systems,* vol. 89, no. 3-4, pp. 387-401, Mar 2018.

[10] S. Karaman and E. Frazzoli, *Sampling-based algorithms for optimal motion planning*. 2011.

[11] Y. X. Shan, B. J. Li, Z. Jian, Z. Yue, and Ieee, "An Approach to Speed Up RRT*," in *2014 Ieee Intelligent Vehicles Symposium Proceedings*(IEEE Intelligent Vehicles Symposium, 2014, pp. 588-592.

[12] J. J. Kuffner Jr and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *ICRA*, 2000, vol. 2.

[13] L. Chen, Y. X. Shan, W. Tian, B. J. Li, and D. P. Cao, "A Fast and Efficient Double-Tree RRT*-Like Sampling-Based Planner Applying on Mobile Robotic Systems," *Ieee-Asme Transactions on Mechatronics,* vol. 23, no. 6, pp. 2568-2578, Dec 2018.

[14] L. Jaillet, J. Cortés, and T. Siméon, "Transition-based RRT for Path Planning in Continuous Cost Spaces," in *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2008.

[15] C. Goerzen, Z. Kong, and B. Mettler, "A Survey of Motion Planning Algorithms from the Perspective of Autonomous UAV Guidance," *Journal of Intelligent & Robotic Systems,* vol. 57, no. 1-4, pp. 65-100, Jan 2010.

[16] Y. Li and S. Jie, "A revised Gaussian distribution sampling scheme based on RRT* algorithms in robot motion planning," in *International Conference on Control*, 2017.

[17] A. H. Qureshi, K. F. Iqbal, S. M. Qamar, F. Islam, Y. Ayaz, and N. Muhammad, "Potential guided directional-RRT* for accelerated motion planning in cluttered environments," in *IEEE International Conference on Mechatronics & Automation*, 2013.

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

[19] A. H. Qureshi *et al.*, "Adaptive Potential guided directional-RRT," in *IEEE International Conference on Robotics & Biomimetics*, 2014.

[20] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. J. Teller, "Anytime Motion Planning using the RRT*," in *IEEE International Conference on Robotics & Automation*, 2011.

[21] A. T. Perez, S. Karaman, A. C. Shkolnik, E. Frazzoli, and M. R. Walter, "Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms," in *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2013.

[22] O. Salzman, D. Halperin, and Ieee, "Asymptotically near-optimal RRT for fast, high-quality, motion planning," in *2014 Ieee International Conference on Robotics and Automation*(IEEE International Conference on Robotics and Automation ICRA, 2014, pp. 4680-4685.

[23] M. Elbanhawi and M. Simic, "Sampling-Based Robot Motion Planning: A Review," *Ieee Access,* vol. 2, pp. 56-77, 2014 2014.

[24] P. Shi and J. N. J. A. M. R. Hua, "Mobile Robot Dynamic Path Planning Based on Artificial Potential Field Approach," vol. 490-495, pp. 994-998, 2012.

[25] L. Zhou and W. Li, "Adaptive Artificial Potential Field Approach for Obstacle Avoidance Path Planning," in *Seventh International Symposium on Computational Intelligence & Design*, 2015.