2024 10th International Conference on Control, Decision and Information Technologies
CoDIT 2024 | Valletta, Malta / July 01-04, 2024 Technically co-sponsored by IEEE & IFAC

2024 10th International Conference on Control, Decision and Information Technologies (CoDIT) | 979-8-3503-7397-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CoDIT62066.2024.10708298

Comparison of RRT, APF, and PSO-Based RRT-APF (PS-RRT-APF) for Collision-Free Trajectory Planning in Robotic Welding

Ozan Kaya[1] and Lars Tingelstad[1]

Abstract— **In robotic applications, trajectory planning is pivotal, particularly in navigating rapidly changing and dynamic environments. This paper presents a comparison of Rapidly Exploring Random Trees (RRT) and Artificial Potential Fields (APF), and Particle swarm Optimization (PSO) based RRT-APF for collision-free trajectory planning in robotic welding scenarios. The proposed methodology addresses a multifaceted set of challenges, emphasizing critical factors such as joint acceleration, obstacle avoidance, and singularity considerations. Robotic welding scenarios pose intricate challenges, demanding a multifaceted approach to ensure both precision and safety. By merging the exploration capabilities of RRT with the obstacle avoidance prowess of APF, our approach not only mitigates the drawbacks associated with local minima problems in APF but also enhances the overall efficiency of RRT. This combination enables rapid exploration and obstacle avoidance, guided by a bias force to optimize trajectory planning for enhanced robotic performance. Furthermore, the methodology extends beyond collision-free trajectories, considering key constraints to provide a holistic solution. Addressing joint, and workspace limits, proposed PS-RRT-APF approach proves collision-free trajectories while addressing specific challenges such as obstacle avoidance, joint velocities, and motion limits. Through this innovative approach, we seek to enhance the overall performance of robotic path planning, ensuring efficiency, and safety in dynamic environments.**

I. INTRODUCTION

In robotic applications, the precision and efficiency of trajectory planning play a pivotal role [1], particularly in dynamically changing environments. Robotic welding, as a representative scenario, demands rapid and optimal path planning to achieve safe movement in the welding process. Traditional trajectory planning methods often struggle with the complexities posed by dynamic environments, obstacles, and the intricate nature of robotic welding scenarios.

With the growing demand for robotics in manufacturing, robotic welding is a well-known application. As industries increasingly rely on robotic systems for welding tasks, the efficiency of trajectory planning becomes a key determinant of overall system performance. The motivation further lies in the limitations of existing trajectory planning methods, which often struggle to adapt to the intricate dynamic environment [2] and constraints of application scenarios [3]. For this purpose, various path-planning approaches, encompassing a spectrum of methodologies, have been explored. Search-based algorithms, exemplified by A* [4], Voronoi [5], and learning-based algorithms like reinforcement learning [6], introduce diverse strategies to navigate complex environments. Evolutionary-based algorithms, such as ant colony [7], PSO [8], along with sample-based algorithms like RRT, and local planning algorithms like APF [9], represent a diverse range of methods. However, each approach is accompanied by its own set of limitations. For instance, search-based algorithms like A* may struggle in scenarios with dynamic and unpredictable obstacles due to their reliance on predefined maps. Learning-based algorithms, particularly reinforcement learning, may face challenges in scaling to high-dimensional state spaces or require extensive training data for optimal performance. Evolutionary-based algorithms, such as ant colony and PSO, could exhibit limitations regarding convergence speed and adaptability to dynamic environments. Sample-based algorithms like RRT might face issues in scenarios where the vast exploration space leads to sub-optimal paths. Local planning algorithms, like APF, may encounter difficulties achieving global optimality, often getting stuck in local minima.

In practical applications, researchers have proposed combining multiple path-planning algorithms to mitigate these limitations. For instance, a combination of A* and Voronoi field in [10] has been suggested to address the shortcomings of individual algorithms and enhance adaptability to various scenarios. Despite the strides made in path planning research, ongoing efforts are essential to develop hybrid approaches that can overcome the diverse challenges posed by real-world environments. In [11], a path planning method for obstacle avoidance of a robot arm based on the improved potential field method is proposed. In addition, this combination can be employed to quickly avoid from the obstacles for UAV real-time path planning [12]. However, the shortest path is not guaranteed by the combination of RRT and APF. Therefore, an optimization algorithm may be fruitfully employed to achieve a shorter path and validate the constraints.

This research aims to optimize trajectory planning for robotic welding applications by integrating RRT [13] and APF [14] within a constrained-based PSO. We aim to achieve collision-free trajectories while addressing specific challenges such as obstacle avoidance, joint velocities, and motion limits. Through this innovative approach, we seek to enhance the overall performance of robotic welding systems, ensuring precision, efficiency, and safety in dynamic environments.

The remainder of this paper is structured as follows: Section II provides the constraints in robotic welding applications. Section III outlines the trajectory planning, specifically focusing on RRT, APF, RRT-APF, and constrained-based optimization, detailing the integration of RRT and

[1]The authors are with the Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway ozan.kaya@ntnu.no

APF within the proposed constrained framework. Section IV presents the results and discussions, followed by a conclusion summarizing the key findings and outlining avenues for future research.

II. Constraints in Robotic Welding

Trajectories in robotic applications need to be considered with the limitation of the process, which is caused by the environment and robot arm.

A. Environmental constraints

A comprehensive description of obstacles and the workspace in multitasking robotic operations is crucial to achieving a collision-free path. To define an obstacle, maintaining a safe distance from the obstacle's surface is essential to prevent potential collisions caused by sudden changes in the dynamic environment. Therefore, the produced trajectory is expected to be far enough from obstacles. Introducing an artificial layer (AL) around the obstacle is beneficial for generating random nodes through RRT and applying repulsive forces with APF. Figure 1 illustrates the placement of obstacles in the workspace and their corresponding layers.

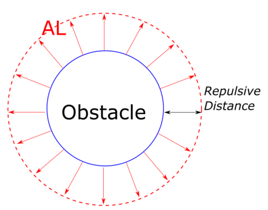

Fig. 1. Artificial layer (AL) on the obstacles in the workspace.

In addition to these considerations, limiting the working area where the end effector is freely moved is essential to decrease the possibility of collisions. In this study, the workspace limits are defined for all axes: [0, 0.1 m]. This workspace is fixed into the YASKAWA workpiece positioner MT1-100.

B. Robotic constraints

Robots have several constraints because of the configuration, joint limits (such as motion limits, maximum velocity, and accelerations), and singularities. These should be considered for trajectory planning. In this experiment, the Yaskawa Motoman GP25-12 industrial robot arm is used as given in Fig. 2. Its joint limits are given in Table I.

TABLE I
The joint limits of YASKAWA GP25-12

	Joints					
	S	*L*	*U*	*R*	*B*	*T*
Motion range	±180	$+155/-105$	$+160/-86$	±200	±150	±455
Maximum speed	210	210	220	435	435	700

To compute the joint positions related to the generated path, a geometric inverse kinematic (IK) solution is used. In the IK solution, the first three joints are assigned for the position of the end effector, whereas the others are assigned for the rotation of the end effector. Assuming no changes in the rotation of the end-effector, the positions of the first three joints are computed for constraint-based trajectory planning. This is executed to minimize the overall motion sum of the joints.

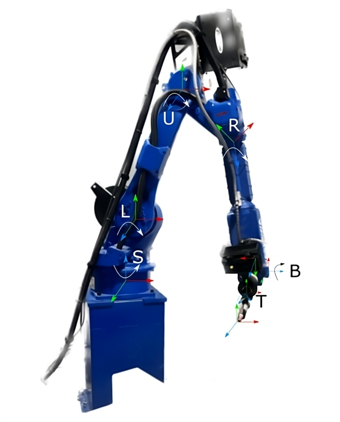

Fig. 2. The Yaskawa Motoman GP25-12 industrial robot arm.

III. Trajectory Planning

Trajectory planning is a fundamental aspect of robotic navigation and manipulation, often conceptualized as a search within a configuration space C. In this space, denoted by $x \in C$, each configuration represents the position and orientation of geometrically intricate bodies within 3D environment. The configuration space is divided into C_{free}, representing configurations where the robotic bodies do not collide with static or dynamic obstacles. The obstacles are fully modeled within the workspace. Utilizing a collision detection algorithm, any given configuration $x \in C$ can be tested to ascertain whether it belongs to the collision-free subset C_{free}.

Our path planning strategy employs three key algorithms: Rapidly Exploring Random Trees (RRT), Artificial Potential Fields (APF), and a synergistic combination of both. RRT excels in exploration, navigating the configuration space dynamically, while APF focuses on obstacle avoidance. By integrating these algorithms within a unified framework, we seek to harness the strengths of both approaches for optimal path planning in robotic applications. The path planning task entails computing a continuous path from an initial configuration x_{init} to a goal configuration x_{goal}. This involves navigating the complex configuration space, leveraging RRT's exploration capabilities, mitigating collisions with APF, and achieving a seamless trajectory from the starting point to the desired destination.

A. RRT

The Rapidly-exploring Random Tree (RRT) was introduced in [15] as a versatile data structure and sampling strategy for efficiently exploring high-dimensional workspace. It may include constraints from obstacles and the dynamics of the manipulator. The basis of the RRT lies in guiding the exploration towards the unexplored regions of the workspace.

The fundamental RRT construction algorithm is expressed as in Algorithm 1.

Algorithm 1 RRT path planning algorithm

Input: Map M; Initial Position x_{init}; Goal x_{goal};Step size Δx; Goal region Q_{goal}
Output: Path G;
while $dist(x_{new}, x_{goal}) \leq Q_{goal}$ **do**
  $x_{rand} \longleftarrow Sample(M)$
  $x_{near} \longleftarrow Near(x_{near}, G)$
  $x_{new} \longleftarrow Steer(x_{near}, x_{new}, \Delta x)$
  if collisionCheck(x_{new}, M) **then**
    $G.addNode(X_{new})$
  end if
end while
return G

The algorithm proceeds iteratively, with each step extending the path by introducing a new position by a randomly selected configuration. The Near function identifies the nearest existing position to the sample configuration, x_{near}. The Steer function attempts to move x_{near} toward x_{new} with a fixed incremental distance Δx. Then, CollisionCheck function is applied to x_{new}. This can be executed efficiently. Two possible outcomes are considered: "Collision free," where x_{new} is directly added to the path; and "Collision," where the proposed x_{new} is rejected if it does not lie in the collision-free space free C_{free}. The exploration stops when the distance between x_{new} and x_{goal} is smaller than Q_{goal}.

RRT is useful for rapidly exploring the map and obstacle avoidance. However, it may be stuck in case a random node is not converged to x_{goal}, which x_{new} is not inside of the goal region (Q_{goal}). For this reason, RRT is generally combined with another search algorithm preferred for bias force. In this study, we utilized APF for achieving the bias force of the newly created nodes to reach the target position.

B. APF

The Artificial Potential Field (APF) algorithm is a path-planning technique widely employed in robotics and autonomous systems to navigate through complex environments. Inspired by concepts from physics, the APF algorithm models the robot's motion within a virtual field of attractive force by the target and repulsive forces by obstacles. This approach aims to guide the robot towards its goal while avoiding obstacles in [16]. The fundamental APF construction algorithm is expressed as in Algorithm 2.

To compute the attribute force, the quadratic potential field equation in [17] is used. The calculation of the attribute potential (U_{att}) and its gradient are given in (1) and (2).

$$U_{\text{att}} = \begin{cases} \alpha |dist(x, x_{goal})|^2 & \text{if } |(x - x_{\text{goal}})| \leq Q_r \\ \alpha (2Q_r |dist(x, x_{goal})| - Q_r^2) & \text{if } |(x - x_{\text{goal}})| > Q_r \end{cases} \tag{1}$$

$$-\nabla U_{\text{att}} = \begin{cases} -2\alpha \text{ dist}(x, x_{\text{goal}}) & \text{if } |(x - x_{\text{goal}})| \leq Q_r \\ -2\alpha Q_r \frac{\text{dist}(x, x_{\text{goal}})}{|dist(x, x_{\text{goal}})|} & \text{if } |(x - x_{\text{goal}})| > Q_r \end{cases} \tag{2}$$

Algorithm 2 APF path planning algorithm

Input: Obstacles positions x_{obs}; Initial Position x_{init}; Goal position x_{goal};Step size Δt; safety distance from obstacles Q_{safe}; Attractive potential gain α; Repulsive potential gain β
Output: Path G
while $x_{new} \neq x_{goal}$ **do**
  compute F_{att}
  for $i = 1$ to $n_{obstacles}$ **do**
    if $dist(x_{new}, x_{goal}) \leq Q_{safe}$ **then**
      compute F_{rep}
    else
      $F_{rep} = 0$
    end if
    sum(F_{rep})
  end for
  compute F_{total}
  compute x_{new}
end while
return G

Where Q_r represents the quadratic range and x is the current position. The attribute force is in (3).

$$F_{att} = -\nabla U_{att} \tag{3}$$

To compute the repulsive force given in (6), repulsive potential (U_{rep}) and its gradient is given in (4) and (5).

$$U_{rep} = \frac{1}{2}\beta \left(\frac{1}{dist(x, x_{obs})} - \frac{1}{Q_{safe}} \right)^2 \tag{4}$$

$$-\nabla U_{rep} = \beta (\frac{1}{dist(x, x_{obs})} - \frac{1}{Q_{safe}}) \frac{1}{dist(x, x_{obs})^2} \tag{5}$$

The shortest distance from the obstacle is calculated in (5) and (6).

$$F_{rep} = -\nabla U_{rep} \tag{6}$$

The total force is given in (7).

$$F_{total} = F_{att} + F_{rep} \tag{7}$$

To compute the next position, motion formulation is given in (8) and (9). It is assumed that the mass value is 1 kg.

$$V_{new} = V + \Delta t F_{total} \tag{8}$$

$$x_{new} = x + \Delta t V_{new} \tag{9}$$

Here, V represents the velocity, x is the current position, V_{new} is the updated velocity by APF, and x_{new} is the next position on the path.

The APF method offers straightforward and efficient path-planning solutions. Despite its effectiveness, APF-based local path planning encounters a significant challenge - the potential occurrence of local minima that can trap a robot before reaching its target. This issue arises because the robot can only perceive local information about obstacles, preventing it from anticipating and avoiding local minima beforehand.

C. RRT-APF

Consequently, a combination of RRT and APF is expected to be helpful in terms of overcoming their limitation, such as convergence problems, exploration of the map, and local minima. RRT is used for solving the problems on the local minima and exploration of the map, while APF is fruitful for convergence. For this purpose, APF generates potential force, which steers the RRT nodes to the goal position. As a result of APF, the RRT node position is updated and added to the path G. The fundamental RRT-APF construction algorithm is expressed as in Algorithm 3.

Algorithm 3 RRT-APF path planning algorithm

Input: Map M; Obstacles positions x_{obs}; Initial Position x_{init}; Goal x_{goal};Step size Δx; safety distance from obstacles Q_{safe}; Attractive potential gain α; Repulsive potential gain β

Output: Path G

```
while x_new ≤ Q_goal do
    x_rand ⟵ Sample(M)
    x_near ⟵ Near(x_near, G)
    x_steer ⟵ Steer(x_near, x_rand, Δx)
    x_new ⟵ APF(x_steer, Δx, x_obs)
    if collisionCheck(x_new, M) then
        G.addNode(x_new)
    end if
    Rewire(x_new, x_neigbor)
end while
return G
```

As a result of RRT-APF, random nodes reach the goal position faster than both RRT and APF.

Furthermore, the path generated by RRT-APF avoids the obstacles and local minima caused by obstacles/environment. However, different paths can be executed for only a single environment since the RRT-APF still has randomness caused by the RRT structure of the algorithm.

D. PS-RRT-APF for constraint based trajectory planning

Our approach employs the Particle Swarm Optimization (PSO) algorithm to determine the optimal path. This algorithm is executed to iteratively refine a trajectory that adheres to the constraints outlined in Section II. The paths between the start and goal points are generated using the RRT-APF (Rapidly-exploring Random Trees with Adaptive Potential Fields) technique, each path represented by a particle.

The evaluation of particles is conducted based on their adherence to the desired constraints. Each particle is scrutinized, and its fitness is determined according to the specified criteria. The particle representing the shortest path length, the ”pbest” (personal best), is then identified. By computing the fitness function using the parameters discovered by IK model, the fitness function of each particle is determined. An essential feature of the PSO is the fitness function as in (10).

$$F = \sum_{t=2}^{T}\sum_{i=1}^{6} w_1(\theta_{i,t}-\theta_{i,t-1})^2 + \sum_{t=1}^{T}\sum_{i=1}^{6} w_2[max(0,\theta_{i,t}-\theta_{max,i}) + max(0,\theta_{min,i}-\theta_{i,t}) + max(0,\dot{\theta}_{i,t}-\dot{\theta}_{max,i})] \quad (10)$$

Where $\dot{\theta}_{i,t}$ and $\theta_{i,t}$ are the velocity and position of i-th joint in the t-th node of path length, w is weights.

The PSO algorithm facilitates the iterative improvement of particles through a dynamic process of position updates. This process continues until a particle, consequently, a path that satisfies the applicable constraints, is found. The iterative refinement allows that the algorithm converges towards a shorter solution, enhancing the efficiency of the search process.

The entire search algorithm is depicted in the flow chart illustrated in Fig. 3, offering a visual representation of how the PSO algorithm refines the paths generated by the RRT-APF approach. This integration of techniques aims to find a trajectory that not only navigates from the start to the goal but does so in a manner that satisfies the specific constraints outlined in Section II.

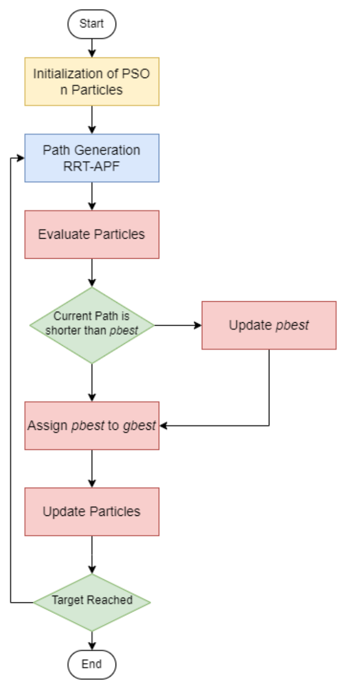

Fig. 3. Flow chart of the constrain based trajectory planning.

IV. The Results and Discussion

This section presents the results of constrained base trajectory planning using the RRT-APF algorithm. The evaluation focuses on trajectory quality and adherence to specified constraints. The performance of our approach is assessed through trajectory length, execution time, collision avoidance, and feasibility within defined constraints. The experiments were conducted in a simulated environment with obstacles. The YASKAWA Motoman GP25-12 robot arm and its inverse kinematic solution compute joint motion.

As a result of RRT-APF, the converging problem of the RRT on exploration is coped with adding bias force into nodes by using APF. For different case scenarios, RRT, APF, and RRT-APF are compared with execution time, collision, and node number. In the case scenarios, two spherical obstacles are used and their positions are changed. For different scenarios, over 1000 different environments have been created, and the data related to the path results in several of these environments is shared on GitHub. The four cases are displayed in Fig. 4. The results are given in Table II.

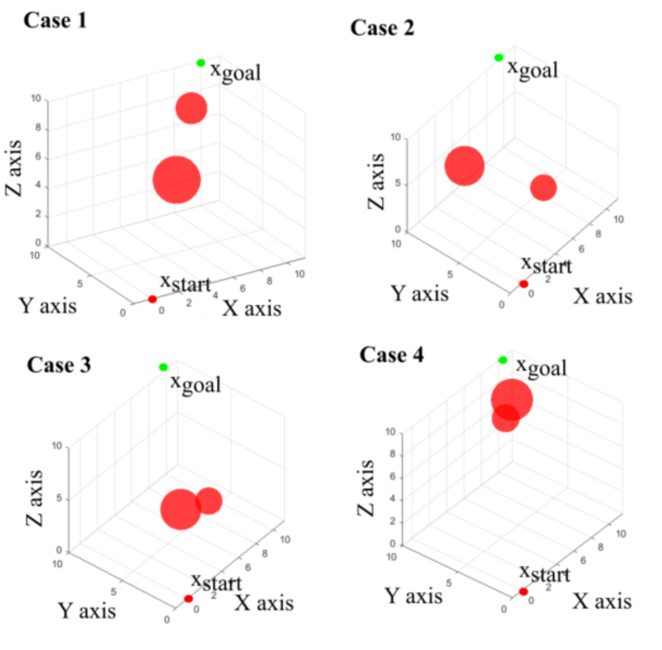

Fig. 4. The case scenarios.

TABLE II
THE COMPARISON RESULTS OF RRT, APF, AND RRT-APF FOR CASE SCENARIOS

		APF	*RRT*	*RRT-APF*
Case 1	execution time (s)	-	1.6214	0.7918
	collision	-	-	-
	node number	-	30	32
Case 2	execution time (s)	54.2	1.5132	0.7917
	collision	-	-	-
	node number	182	28	29
Case 3	execution time (s)	58	1.5347	0.7915
	collision	-	-	-
	node number	206	30	30
Case 4	execution time (s)	61	1.6452	0.7918
	collision	-	-	-
	node number	218	29	28

All algorithms are successful in collision free path. However, APF can not reach the target position in case 1 because of the local minimum problem caused by repulsive force in the same direction as the attractive force. This problem can be fixed by adding tangent force into the total force equation of APF. RRT and RRT-APF have shorter node number. Regarding execution time, RRT is better than APF, but it can be stuck in exploration. So, RRT-APF is the best performance for the execution time. The comparison results in [18] also support our achievements. It can be preferred in real-time applications.

Because of the randomness of RRT, several exploration trees might be executed. It affects that different path connections occur for each iteration. To find a shorter path suitable to the constraints, PSO executes the path generation several times. Each path performance related to trajectory length, obstacle avoidance, and joint motion is compared to achieve an optimal path. As a result of APF, RRT, and PS-RRT-APF, the obtained paths are shown in Fig. 5.

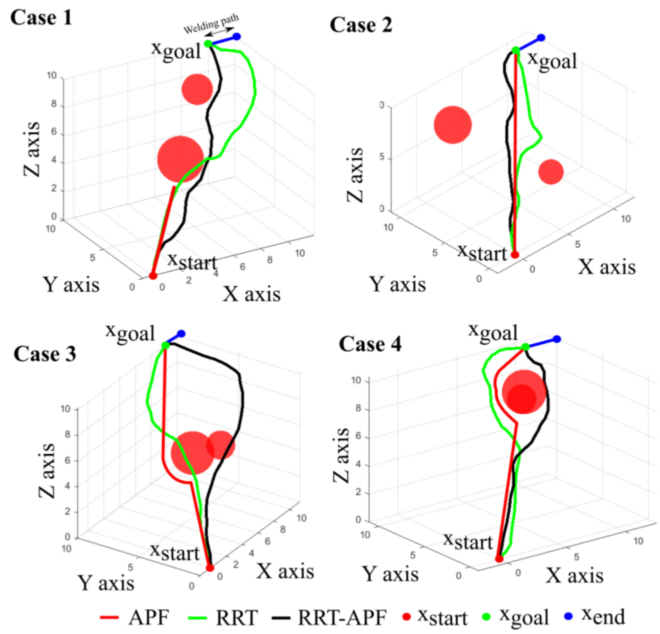

Fig. 5. The trajectories for case scenarios.

When dealing with singularities, it is important to be able to adjust the path around them instead of going through them. In our experiments, the rotation of the end effector is fixed at Euler angles [90 0 0]. After obtaining the shortest path, a weighted average method as a smoothing algorithm is used. For the smoothed path, the first three joints' positions are shown in Fig. 6.

As seen in Fig. 6, the welding process consists of two phases: (1) end-effector motion with collision-free trajectory until reaching the starting point of welding (x_{goal}) and (2) end-effector following the welding line. At t=28 s, the end-effector reaches the starting point by RRT-APF. Then, the robot tracks the straight line in the welding surface.

The trajectories exhibit robust collision avoidance behavior, which is crucial for real-world applications. Visualizations of generated trajectories showcase the smoothness and adherence to specified constraints.

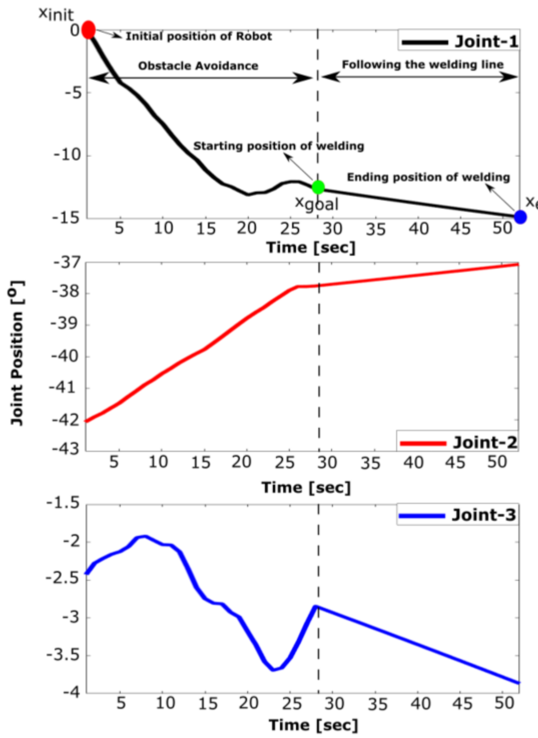

Fig. 6. The joints' position related to the trajectory generated by the PS-RRT-APF for case 1.

V. CONCLUSION

This paper presented the results of constrained base trajectory planning utilizing the PS-RRT-APF algorithm. The evaluation focused on trajectory quality and adherence to specified constraints, assessing performance through trajectory length, execution time, collision avoidance, and feasibility within defined constraints. The experiments, conducted in a simulated environment with obstacles using the YASKAWA Motoman GP25-12 robot arm, demonstrated the effectiveness of the RRT-APF algorithm.

The RRT-APF algorithm effectively addressed the converging problem of the RRT on exploration by incorporating bias force into nodes through APF. Comparative analyses involving RRT, APF, and RRT-APF in various case scenarios, considering execution time, collision, and node number, illustrated the superiority of RRT-APF, particularly in real-time applications. Despite the tangent force is able to resolve the local minima problem encountered by APF, RRT-APF is quite successful in local minima problems.

Additionally, the paper introduced the integration of PSO with RRT-APF to find shorter paths suitable for the constraints, considering trajectory length, obstacle avoidance, and joint motion. A weighted average method for smoothing the trajectories ensured robust collision avoidance behavior, vital for real-world applications.

The overall results indicated that the RRT-APF algorithm excels in generating efficient, collision-free trajectories while adhering to specified constraints. The computational time analysis emphasized its practicality for real-time applications. Despite these promising results, the paper acknowledges the need for further discussions on limitations and potential improvements, paving the way for future research in the constraint-based trajectory planning field.

ACKNOWLEDGMENT

The research presented in this paper was funded by the Norwegian Research Council under Project Number 295138, Robotic Welding of Aluminium Hulls.

REFERENCES

[1] H. Fang, S. Ong, and A. Nee, "Robot path planning optimization for welding complex joints," *The International Journal of Advanced Manufacturing Technology*, vol. 90, pp. 3829–3839, 2017.
[2] S. Hosseininejad and C. Dadkhah, "Mobile robot path planning in dynamic environment based on cuckoo optimization algorithm," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839575, 2019.
[3] H. Wang, Y. Huang, A. Khajepour, Y. Zhang, Y. Rasekhipour, and D. Cao, "Crash mitigation in motion planning for autonomous vehicles," *IEEE transactions on intelligent transportation systems*, vol. 20, no. 9, pp. 3313–3323, 2019.
[4] C. Wang, L. Wang, J. Qin, Z. Wu, L. Duan, Z. Li, M. Cao, X. Ou, X. Su, W. Li, *et al.*, "Path planning of automated guided vehicles based on improved a-star algorithm," in *2015 IEEE International Conference on Information and Automation*, pp. 2071–2076, IEEE, 2015.
[5] B. B. K. Ayawli, X. Mei, M. Shen, A. Y. Appiah, and F. Kyeremeh, "Mobile robot path planning in dynamic environment using voronoi diagram and computation geometry technique," *Ieee Access*, vol. 7, pp. 86026–86040, 2019.
[6] B. Li and Y. Wu, "Path planning for uav ground target tracking via deep reinforcement learning," *IEEE access*, vol. 8, pp. 29064–29074, 2020.
[7] L. Jianhua, Y. Jianguo, L. Huaping, G. Peng, and G. Meng, "Robot global path planning based on ant colony optimization with artificial potential field," *Nongye Jixie Xuebao/Transactions of the Chinese Society of Agricultural Machinery*, vol. 46, no. 9, 2015.
[8] E. Masehian and D. Sedighizadeh, "A multi-objective pso-based algorithm for robot path planning," in *2010 IEEE international conference on industrial technology*, pp. 465–470, IEEE, 2010.
[9] E. N. Sabudin, R. Omar, S. K. Debnath, and M. S. Sulong, "Efficient robotic path planning algorithm based on artificial potential field," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 4840–4849, 2021.
[10] J. Zhao, X. Ma, B. Yang, Y. Chen, Z. Zhou, and P. Xiao, "Global path planning of unmanned vehicle based on fusion of a star algorithm and voronoi field," *Journal of Intelligent and Connected Vehicles*, vol. 5, no. 3, pp. 250–259, 2022.
[11] X. Xia, T. Li, S. Sang, Y. Cheng, H. Ma, Q. Zhang, and K. Yang, "Path planning for obstacle avoidance of robot arm based on improved potential field method," *Sensors*, vol. 23, no. 7, p. 3754, 2023.
[12] L. Yafei, W. Anping, C. Qingyang, and W. Yujie, "An improved uav path planning method based on rrt-apf hybrid strategy," in *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*, pp. 81–86, 2020.
[13] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt," in *2011 IEEE international conference on robotics and automation*, pp. 1478–1483, IEEE, 2011.
[14] M. C. Lee and M. G. Park, "Artificial potential field based path planning for mobile robots using a virtual obstacle concept," in *Proceedings 2003 IEEE/ASME international conference on advanced intelligent mechatronics (AIM 2003)*, vol. 2, pp. 735–740, IEEE, 2003.
[15] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.
[16] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.
[17] J. R. Andrews and N. Hogan, "Impedance control as a framework for implementing obstacle avoidance in a manipulator," Master's thesis, M. I. T., Dept. of Mechanical Engineering, 1983.
[18] C. Huang, Y. Zhao, M. Zhang, and H. Yang, "Apso: An a*-pso hybrid algorithm for mobile robot path planning," *IEEE Access*, 2023.