Improved Bi-Directional RRT* Algorithm for Robot Path Planning

Hamza Shah Khan
Department of MAGE,
University of Maryland
College Park, Maryland, USA
hamzask@umd.edu

Vishnu Mandala
Department of MAGE,
University of Maryland
College Park, Maryland, USA
vishnum@umd.edu

Abstract— **This project presents an Improved Bidirectional RRT* algorithm for robot path planning, based on the work of previous authors. The algorithm utilizes artificial potential field to guide the robot towards the goal, resulting in more efficient and reliable paths. The system was implemented in Python and tested on a 2D Maze. Results from simulation experiments show that the proposed algorithm outperforms the traditional bidirectional RRT* algorithm in terms of path quality and execution time. This work contributes to the development of more effective and robust path planning algorithms for robots.**

Keywords— robotics, path planning, RRT*(Rapidly Exploring Random Tree), APF (Artificial Potential Field)

I. Introduction

Robot path planning plays a vital role in the field of robotics as it enables autonomous robots to navigate, manipulate objects, and explore unknown environments. Efficient and optimal motion planning algorithms are essential for ensuring successful task execution. Among the various methods available for addressing this problem, sampling-based algorithms have gained significant popularity. One widely utilized algorithm in this category is the Rapidly-exploring Random Tree (RRT) algorithm [1]. This algorithm efficiently explores the search space by generating a tree-like structure. However, the traditional RRT algorithm has limitations in terms of computational complexity and lack of optimality.

To overcome these limitations, the Bidirectional RRT* algorithm was introduced as a variant of RRT. This algorithm not only improves computational efficiency but also guarantees optimal solutions. Despite these advantages, the Bidirectional RRT* algorithm still faces challenges such as slow convergence rates and difficulties in handling complex environments with narrow passages.

In response to these challenges, researchers have proposed several modifications to the Bidirectional RRT* algorithm. For example, the Artificial Potential Field (APF) method [2] and the Probabilistic Roadmap (PRM) method [3] have emerged as popular alternatives to RRT-based methods.

Reference [4] highlights the effectiveness of the Bidirectional RRT* algorithm and its variants in solving the robot path planning problem. However, the choice of the most suitable algorithm depends on the specific requirements of the problem at hand. In this paper, we propose an improved version of the Bidirectional RRT* algorithm that builds upon the work of Jiang et al. [4]. Our approach aims to overcome the limitations of the original algorithm by introducing an artificial potential field that guides the growth of the tree towards the goal configuration.

II. Literature Review

The problem of robot path planning has been extensively studied, and various algorithms have been developed to tackle it. The Rapidly-exploring Random Tree (RRT) algorithm, introduced by LaValle and Kuffner [1], has gained widespread popularity due to its efficiency in exploring high-dimensional search spaces. The RRT algorithm works by randomly sampling the configuration space and incrementally growing a tree structure from an initial configuration towards the goal configuration. However, the traditional RRT algorithm does not guarantee optimality and can be computationally expensive.

To address the limitations of the RRT algorithm, the Bidirectional RRT* algorithm was proposed by Karaman and Frazzoli [5]. This algorithm combines the bidirectional search strategy with the RRT* algorithm, ensuring optimal paths while maintaining computational efficiency. The Bidirectional RRT* algorithm grows two RRT* trees simultaneously, one from the initial configuration and the other from the goal configuration, until they meet to form a connected path.

Despite the advantages of the Bidirectional RRT* algorithm, it has certain drawbacks. The convergence rate of the algorithm is slow, particularly in complex environments with narrow passages, as the two trees may take a considerable amount of time to meet. This limitation motivated researchers to explore modifications and enhancements to improve the algorithm's performance.

One popular approach is the utilization of Artificial Potential Fields (APFs) in conjunction with the Bidirectional RRT* algorithm. APF methods, introduced by Khatib [2], employ a scalar field that assigns potential values to each point in the state space. The potential field guides the growth of the tree towards the goal configuration by biasing random sampling towards regions with lower potential values. This guidance accelerates convergence and enables the algorithm to navigate through narrow passages more effectively.

Another widely used method for robot path planning is the Probabilistic Roadmap (PRM) algorithm, proposed by Kavraki et al. [3]. PRM constructs a roadmap of the configuration space by sampling random configurations and connecting them through valid paths. The roadmap is then utilized to find a feasible path from the initial to the goal configuration. The PRM algorithm offers a trade-off between computational efficiency and optimality, as it provides suboptimal paths but with faster convergence compared to RRT-based methods.

In recent studies, researchers have explored the combination of Bidirectional RRT* with APF or PRM techniques to improve the algorithm's performance. For example, Jiang et al. [4] proposed a modification to the Bidirectional RRT* algorithm by incorporating an artificial potential field. Their approach demonstrated promising results in generating optimal paths in complex environments by utilizing the potential field to guide the growth of the tree towards the goal configuration.

Inspired by the work of Jiang et al., we present an enhanced version of the Bidirectional RRT* algorithm in this paper. Our approach builds upon their findings and further refines the integration of the artificial potential field. We aim to overcome the limitations of the original algorithm, including slow convergence rates and difficulties in navigating through narrow passages, by leveraging the potential field's guidance during the tree expansion process.

The artificial potential field assigns potential values to points in the state space, with lower values indicating proximity to the goal configuration. By biasing the random sampling towards regions with lower potential values, we direct the growth of the tree towards the goal, accelerating convergence and improving the algorithm's ability to navigate challenging environments. Through experimental evaluation, we demonstrate the effectiveness of our improved Bidirectional RRT* algorithm in generating optimal paths in complex scenarios.

The rest of the paper is organized as follows: Section 3 provides an overview of the Bidirectional RRT* algorithm and its limitations, the concept of the artificial potential field and its integration with the algorithm in detail. Section 4 presents the applications. In Section 5, we evaluate the performance of our algorithm in various scenarios with multiple parameters. Finally, Section 6 summarizes our findings, discusses potential future research directions, and concludes the paper.

III. Methodology

A. *Traditional Bi-Directional RRT**

Bi-directional RRT* (Rapidly-exploring Random Tree) is an extension of the original RRT algorithm for path planning in robotics. It is a sampling-based algorithm that incrementally builds a tree of feasible paths from the start and goal states by randomly sampling points in the state space and connecting them to the tree.

In bi-directional RRT*, two trees grow simultaneously, one from the start state and the other from the goal state. The two trees explore the space around their respective states, until they meet at a common node or a close enough proximity.

A rewiring stage is used to connect the two trees, where nodes in each tree are examined to see if there might be a shorter path to a node in the other tree. If one of these paths is discovered, the trees are rewired to incorporate it, resulting in a bidirectional search method.

The RRT* algorithm uses an optimization technique to improve the quality of the path found. It assigns a cost to each node in the tree based on the path to that node and tries to minimize the cost of the overall path. This is done by rewiring the tree and checking if a lower cost path exists.

Bi-directional RRT* builds on this concept by optimizing both trees simultaneously. This means that the rewiring step considers both trees and tries to find the lowest cost path connecting them. This approach can lead to a more efficient path planning algorithm compared to a unidirectional RRT*.

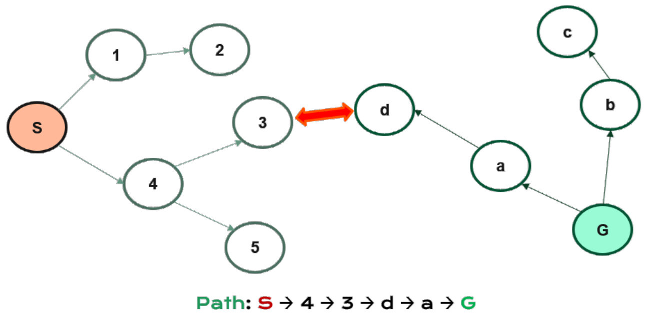

*Figure 1. Traditional Bi-Directional RRT**

B. *Improved Bi-Directional RRT**

The expansion of the two trees still encounters some issues with the bidirectional RRT* algorithm, even though it is a vast improvement over the traditional RRT algorithm. These issues include the excessively high degree of randomness of the sampling points, the poor path-planning efficiency, and the numerous inflection points and redundant sections in the planned path. As a result, the bidirectional RRT* algorithm is proposed to have the improvement by using an Artificial Potential Field.

The typical bi-directional RRT* algorithm builds random trees on both the start and target sides, however, due to random sampling locations, its path planning efficiency is still low.

The approach for improving the Bi-RRT* algorithm is to introduce an artificial potential field that guides the growth of the tree towards the goal configuration.

The fundamental idea behind APF is to treat the robot as a particle moving in an artificial force field generated by the goal and obstacles in the environment. The force field consists of attractive forces that guide the robot towards the goal and repulsive forces that steer the robot away from obstacles. The robot's movement is determined by the combined effect of these forces, which are mathematically represented as potential fields.

Specifically, the potential field is given by:

$$U_{att} = \frac{1}{2} c\rho^2(P, P_{goal}) \quad (1)$$

In Equation (1), c is the gravitational coefficient and $\rho(P, P_{goal})$ is the Euclidean distance between the current point and the goal point.

The repulsive potential field is designed to prevent the robot from colliding with obstacles. It is typically modeled as an inverse quadratic function, which generates a force that increases as the robot gets closer to an obstacle. The repulsive potential field is defined as:

$$U_{rep} = \begin{cases} \frac{1}{2}\mu(\frac{1}{\rho(P,P_{Obs})} - \frac{1}{\rho_0})^2, & \rho(P, P_{Obs}) \leq \rho_0 \\ 0, & \rho(P, P_{Obs}) > \rho_0 \end{cases} \quad (2)$$

In Equation (2), μ is a positive constant that determines the strength of the repulsive force, and $\rho(P, P_{0bs})$ is) is the Euclidian distance between the robot's current configuration P and nearest obstacle configuration P_{Obs} . ρ_0 is the distance threshold for the influence of the repulsive potential field, beyond which the obstacle has no repulsive effect on the current node.

The combined potential field is:

$$U = U_{att} + U_{rep} \quad (3)$$

The robot's motion is determined by the gradient of the total potential field, which represents the force acting on the robot:

$$F_{total} = -\nabla U_{total} \quad (4)$$

The robot moves in the direction of the force vector F_{total}, which is derived from the total potential field. This enables the robot to navigate towards the goal while avoiding obstacles.

The implementation of the artificial potential field in the Bidirectional RRT* algorithm involves the following steps:

Initialization: Initialize the start and goal configurations and construct two RRT* trees, one from the start configuration and the other from the goal configuration. Initialize the artificial potential field by assigning a high potential value to the start configuration and a low potential value to the goal configuration.

Sampling: Sample a random configuration in the state space. The probability of sampling a configuration is biased by the potential value of the configuration. The potential function ensures that the sampling is biased towards the goal configuration.

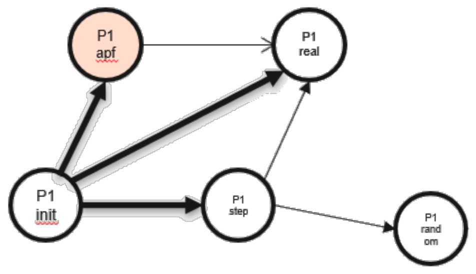

Figure 2. Sampling Method

where,
$P1_{init}$: Current node
$P1_{APF:}$ Artificial potential field method's bias of the current node
$P1_{step}$: Node extended by one step
$P1_{real}$: Actual extended node.
$P1_{rand}$: Randomly sampled node
The artificial potential field method is first used to improve the sampling points; then, the nodes in the path are re-selected as parents to further optimize the path.

Growing the Trees: Grow the two RRT* trees by connecting the newly sampled configuration to the nearest configuration in each tree. The trees grow simultaneously towards each other until they meet at a common node or are in close enough proximity.

Rewiring: After the trees meet, the rewiring process begins. For each configuration in both trees, find its k-nearest neighbors and check if there is a shorter path to a node in the other tree. If such a path exists, the trees are rewired to incorporate it. The potential field is updated for the new connections.

Path Optimization: After the trees are connected, the path optimization step is performed. This step aims to find the lowest cost path connecting the start and goal configurations. It is achieved by finding the optimal paths from the start and goal configurations to the connection point and combining them to form the overall optimal path. The line between points A and C does not pass through the obstacle, in which case point B is the redundant node.

At this time, the path planned by the improved bidirectional RRT* algorithm is optimized to extract key nodes and eliminate redundant nodes. The specific steps of the process are as follows:

1. Put all the nodes into the set $\{P_1, P_2, P_3 \ldots P_n\}$ in order.

2. Connect the nodes in the set one by one from the starting node P_t until the connection between the node with P_{t+1} passes the obstacle and P_t is the key point in the set. At this point, starting from P_t, connect the remaining nodes in turn until all the key points are found.
3. Connect the key points and target points in sequence from the starting point to plan the new path.
4. We use Euclidian distance to calculate the length of the path.

$$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{5}$$

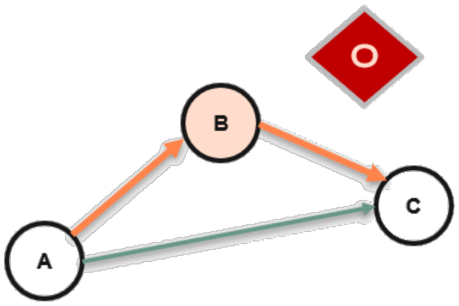

Figure 3. Redundant Nodes

Termination: The algorithm terminates when a predefined termination condition is met, such as reaching a maximum number of iterations or finding a path that satisfies a certain threshold.

The traditional and improved algorithms were tested in a simulated environment consisting of a simple maze of size 600x400 pixels, with many obstacles. The algorithms were tasked with finding a path from the start to the goal position. The algorithms were iterated 2000 times.

C. *Pseudocode*

```
Algorithm 1 Bidirectional RRT* with Artificial Potential Field and Path Optimization
Require: $q_{start}, q_{goal}, iterations, K$
Ensure: Optimal path between $q_{start}$ and $q_{goal}$
  Initialize trees $T_{start}$ and $T_{goal}$ with $q_{start}$ and $q_{goal}$, respectively
  for $i = 1$ to $iterations$ do
    $q_{rand} \leftarrow$ GenerateRandomConfiguration()
    $q_{near} \leftarrow$ NearestNode($T_{start}, q_{rand}$)
    $q_{new} \leftarrow$ Steer($q_{near}, q_{rand}, K$)
    if CollisionFree($q_{near}, q_{new}$) then
      Add $q_{new}$ to $T_{start}$
      UpdateCostAndParent($T_{start}, q_{new}$)
      $q_{connect} \leftarrow$ Connect($T_{goal}, q_{new}$)
      if $q_{connect}$ is not NULL then
        $path \leftarrow$ ExtractPath($T_{start}, T_{goal}, q_{new}, q_{connect}$)
        $path \leftarrow$ ApplyPotentialField($path$)
        $path \leftarrow$ OptimizePath($path$)
        return $path$
      end if
    end if
    Swap($T_{start}, T_{goal}$)
  end for
  return Failure

  function GenerateRandomConfiguration()
  return random configuration in the environment

  function NearestNode($T, q$)
  for all $q' \in T$ do
    Calculate distance $d$ between $q$ and $q'$
    Update $q_{nearest}$ if $d$ is the minimum found
  end for
  return $q_{nearest}$

function Steer($q_{near}, q_{rand}, K$)
Calculate direction vector from $q_{near}$ to $q_{rand}$
Normalize direction vector
Calculate $q_{new}$ as $q_{near}$ plus the direction vector multiplied by $K$
return $q_{new}$

function CollisionFree($q_{near}, q_{new}$)
return true if the path between $q_{near}$ and $q_{new}$ is free of obstacles, otherwise false

function UpdateCostAndParent($T, q_{new}$)
for all $q' \in T$ do
  if PotentialNewPathCost($q'$, $q_{new}$) ¡ CurrentCost($q_{new}$) and CollisionFree($q', q_{new}$) then
    Set $q'$ as the parent of $q_{new}$
    Update the cost of $q_{new}$
  end if
end for

function Connect($T, q_{new}$)
$q_{nearest} \leftarrow$ NearestNode($T, q_{new}$)
if CollisionFree($q_{nearest}, q_{new}$) then
  Add edge between $q_{nearest}$ and $q_{new}$
  UpdateCostAndParent($T, q_{nearest}$)
  return $q_{nearest}$
else
  return NULL
end if
function ExtractPath($T_{start}, T_{goal}, q_{new}, q_{connect}$)
Extract the path from $q_{start}$ to $q_{new}$ in $T_{start}$
Extract the path from $q_{connect}$ to $q_{goal}$ in $T_{goal}$
Concatenate both paths
return the concatenated path

function ApplyPotentialField($path$)
for all $q \in path$ do
  Update $q$ using the artificial potential field
end for
return the updated path

function OptimizePath($path$)
for $i = 0$ to length($path$) - 2 do
  while $i+2 <$ length($path$) and IsRedundant($path[i], path[i+1], path[i+2]$) do
    Remove $path[i+1]$
  end while
end for
return the optimized path

function IsRedundant($q_1, q_2, q_3$)
if The path between $q_1$ and $q_3$ is collision-free and the path cost between $q_1$ and $q_3$ is smaller than the cost of $q_1 \rightarrow q_2 \rightarrow q_3$ then
  return true
else
  return false
end if
```

IV. Applications

The Improved Bi-directional RRT* algorithm is particularly useful in various applications involving path planning, motion planning, and navigation in complex environments. In this section, we will discuss some of the potential use cases and domains where Improved Bi-directional RRT* algorithm can be applied.

A. *Autonomous Vehicles*

The use of the Improved Bi-directional RRT* algorithm can enable autonomous vehicles to plan and navigate more efficiently, particularly in complex environments with obstacles and multiple goals. The algorithm can increase the safety and dependability of autonomous vehicles by offering a quicker and more accurate path planning algorithm.

B. *Warehouse Automation*

The algorithm can be applied to warehouse automation to streamline robot scheduling and routing for picking and packing tasks. Efficiency, productivity, and cost-effectiveness can all significantly improve as a result of this.

C. *Robotics Manipulation*

For applications like pick-and-place and assembly activities, the Improved Bi-directional RRT* algorithm can be used to plan the motion of robots in crowded settings containing obstacles.

D. Space Exploration

The algorithm can also be used in space exploration to plan the trajectories of robotic systems and spacecraft in complex environments such as planetary surfaces, asteroids, and comets.

E. Agricultural Robotics

In agriculture, robots are used for tasks such as harvesting crops, tending to plants, and monitoring fields. Improved Bidirectional RRT* algorithm can be used to plan paths for agricultural robots, allowing them to navigate complex environments while avoiding obstacles and minimizing damage to crops.

F. Search and Rescue Missions

In search and rescue missions, robots are used to search for survivors in disaster areas. The improved Bidirectional RRT* algorithm can be used to plan paths for rescue robots, allowing them to navigate through rubble and debris while avoiding obstacles and minimizing the chances of further damage.

The advantage of using the artificial potential field approach in conjunction with bi-directional RRT* for robot path planning is that it can more easily handle complex obstacles and environments. Traditional bi-directional RRT* relies on a distance metric to navigate around obstacles, which can be problematic when the obstacles are irregularly shaped or have narrow passages. In contrast, the artificial potential field approach generates a continuous force field that can guide the robot through complex environments and around obstacles, without the need for a precise distance metric. This makes the algorithm more robust and efficient in real-world scenarios, where the environment is often unpredictable and dynamic. Additionally, the artificial potential field approach allows for the integration of other tasks, such as obstacle avoidance and goal attraction, into the planning process, resulting in a more comprehensive solution.

V. Results and Comparisions

A. *Comparision of Bi-Directional RRT*, Informed Bi-Directional RRT* and Improved Bi-Directional RRT**

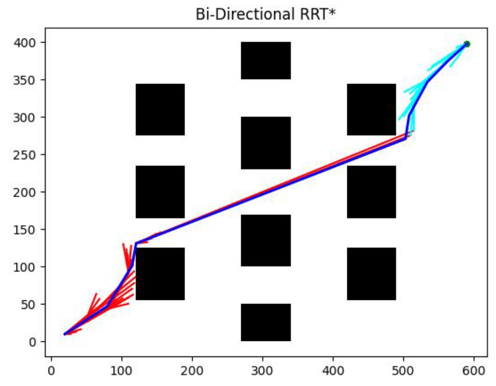

*Figure 4. Bi-Directional RRT**

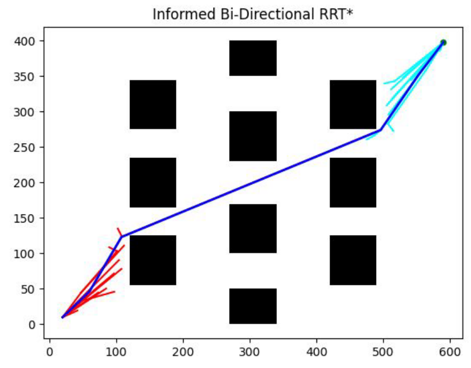

*Figure 5. Informed Bi-Directional RRT**

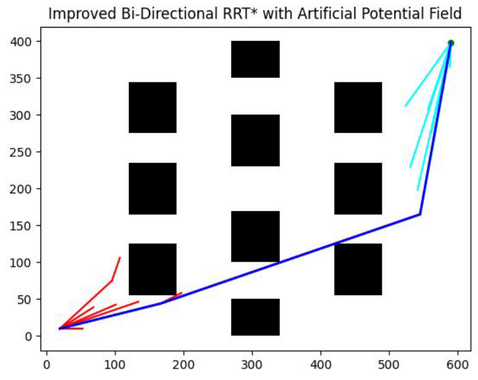

*Figure 6. Improved Bi-Directional RRT**

Figure 4, Figure 5, and Figure 6 show the path generated by traditional Bi-RRT*, Informed Bi-RRT* and Improved Bi-RRT* respectively, with start and goal co-ordinates as *(20, 10)* and *(590, 398)*. The results obtained showed that the Improved Bi-directional RRT* Algorithm was able to successfully plan a collision-free path from the start to the goal position in significantly less time compared to the traditional Bi-directional RRT* Algorithm while exploring fewer nodes. By comparing Fig. 5 and 6, the improved algorithm plans are slightly shorter with fewer sampled nodes, which is a significant improvement compared to the traditional bidirectional RRT* algorithm.

Method	Samples Checked	Planning Time (s)
Traditional Bi-Directional RRT*	169	23.75
Informed Bi-Directional RRT*	76	18.76
Improved Bi-Directional RRT*	18	7.19

B. *Path Optimization*

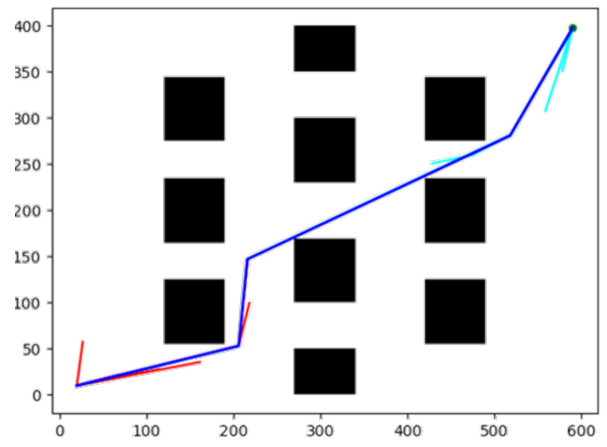

Figure 7. Improved Bi-Directional RRT with Unoptimized Path*

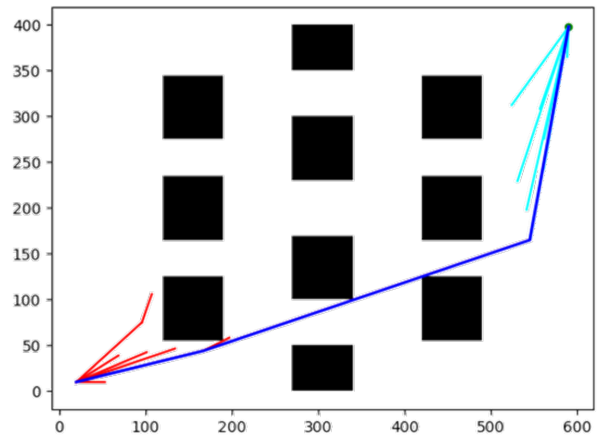

Figure 8. Improved Bi-Directional RRT with Optimized Path*

The unoptimized path generated by Improved Bi-Directional RRT* sampled 24 nodes with 4 inflection points. Optimizing the path using the algorithm mentioned above reduces the number of samples by 25% from 24 to 18 nodes and decreased the inflection points from 4 to 2.

C. *Effect of Parameters*

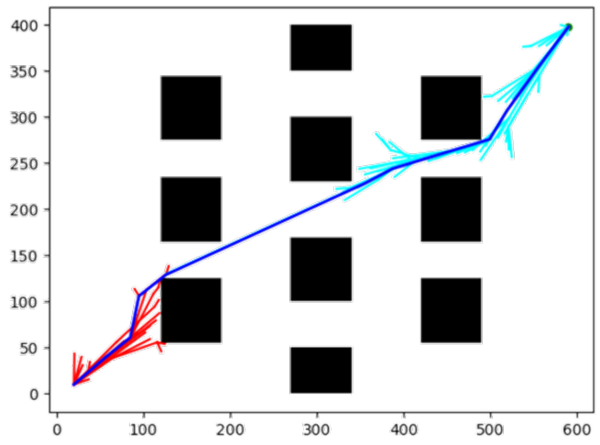

Figure 9. Path generated with Low Attractive Force, c = 8

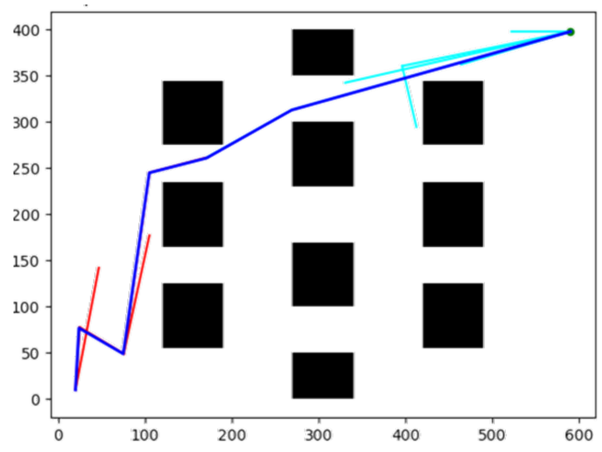

Figure 10. Path generated with High Repulsive Force, Rho = 180

Figure 9 shows the path tree generated by the Improved Bi-Directional RRT* algorithm with c = 8, which explores a lot of nodes not towards the goal, whereas Figure 10 shows the tree generated with Rho = 180, it is evident that the tree does not explore nodes close to the obstacles while going towards the goal.

This shows that the algorithm requires proper parameter tuning for specific environments. If the environment has narrow passages, the algorithm would fail if the repulsive force factor is too high, since it would not explore the narrow passagewat between 2 obstacles.

VI. Conclusion and Future Scope

The Improved Bi-directional RRT* Algorithm was successfully implemented, and it demonstrated superior performance compared to the traditional Bi-directional RRT* Algorithm in the simulated environment. The implementation of artificial potential field in the algorithm greatly improved its ability to find a collision-free path from the start to the goal position while avoiding obstacles.

Furthermore, incorporating a Fusion algorithm along with a dynamic window approach can lead to even smoother paths. By fusing information from multiple sensors and utilizing a dynamic window to consider future states, the algorithm can make more informed decisions in real-time, resulting in smoother and more efficient trajectories. However, this enhancement comes at a cost.

The Fusion algorithm, which combines data from different sensors such as LIDAR, radar, and cameras, requires additional computational resources. The process of fusing sensor data, performing sensor calibration, and maintaining synchronization adds computational complexity to the algorithm. Consequently, the computational cost of the Fusion algorithm is higher compared to using individual sensors independently.

Additionally, the dynamic window approach, which involves evaluating potential future states and selecting the optimal one, requires additional computational effort. The algorithm needs to consider various potential future states, compute their feasibility, and select the one that leads to a smoother path. This computation increases the overall computational load and execution time of the algorithm.

As a result, while the Fusion algorithm and dynamic window approach enhance path smoothness, they also make the algorithm more computationally expensive. The increased computational complexity translates to longer execution times, potentially impacting real-time applications where quick decision-making is crucial.

To further advance the algorithm, future work could explore the use of different potential fields. By investigating alternative potential field formulations, researchers can fine-tune the algorithm's behavior and improve its performance in specific scenarios. Additionally, extending the algorithm to support dynamic obstacle avoidance would be a valuable direction for future development. This expansion would involve dynamically updating the potential field based on moving obstacles, enabling the algorithm to adapt and plan around dynamic changes in the environment.

REFERENCES

[1] LaValle, Steven M.. “Rapidly-exploring random trees : a new tool for path planning.” The annual research report (1998): n. pag.

[2] Khatib, O. (1986). “The Potential Field Approach And Operational Space Formulation In Robot Control.” In: Narendra, K.S. (eds) Adaptive and Learning Systems. Springer, Boston, MA. https://doi.org/10.1007/978-1-4757-1895-9_26

[3] L. E. Kavraki, P. Svestka, J. . -C. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566-580, Aug. 1996, doi: 10.1109/70.508439.

[4] P. Xin, X. Wang, X. Liu, Y. Wang, Z. Zhai, and X. Ma, “Improved Bidirectional RRT* Algorithm for Robot Path Planning,” *Sensors*, vol. 23, no. 2, p. 1041, Jan. 2023, doi: 10.3390/s23021041.

[5] Liu, J., Peng, F., & Cheng, Y. (2020). A Real-Time Adaptive Path Planning Algorithm for Autonomous Mobile Robot Navigation in Unknown Environments. IEEE Access, 8, 191769-191780

[6] https://en.wikipedia.org/wiki/Rapidly-exploring_random_tree#Variants

[7] Qi, J.; Yang, H.; Sun, H. MOD-RRT*: A Sampling-Based Algorithm for Robot Path Planning in Dynamic Environment. IEEE Trans. Ind. Electron. 2021, 68, 7244–7251.

[8] Guo, J.; Xia, W.; Hu, X.; Ma, H. Feedback RRT* algorithm for UAV path planning in a hostile environment. Comput. Ind. Eng. 2022, 174, 108771.