Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 231 (2024) 703–708

Procedia Computer Science

www.elsevier.com/locate/procedia

Workshop on Intelligent Systems Based on Internet of Things Technology (IoT)
November 7-9, 2023, Almaty, Kazakhstan

Narrowed Regions-based Bidirectional Path Planning Using RRT-Connect for Single Aircraft Missions

Doğan Candemir[a,*], İbrahim Kök[b], Suat Özdemir[a]

[a]Department of Computer Engineering, Hacettepe University, Ankara 06530, Turkey
[b]Department of Computer Engineering, Pamukkale University, Denizli 20020, Turkey

Abstract

Fighter aircraft, the emblem of modern military aviation, have decisively shaped the landscape of contemporary warfare. These machines are purpose-built and equipped to carry out a variety of missions. Mission planning is a common practice, usually done before the aircraft takes off. However, the volatile nature of warfare poses a major challenge to the precise execution of pre-planned missions. If the established flight path needs to be adjusted during a mission, a new path must be created and the mission completed using the newly derived flight path. In this study, we present a novel approach to path planning using a modified RRT connect algorithm. By considering the nearest nodes in two trees starting from T_{init} and T_goal, we use a constrained sampling strategy within a bounded environment. This iterative process creates a tree between the current and target locations and a path is extracted using the A* algorithm. The proposed method aims to omit samples in regions where the passage of the aircraft is infeasible or costly, resulting in a smoother trajectory. Experiments in different scenarios have shown that the method consistently delivers smooth routes. This research demonstrates the potential of our approach to improve the adaptability and performance of mission-critical aircraft in dynamic environments.

Keywords: Path planning, aircraft, robotics, RRT, mission planning

1. Introduction

Fighter aircraft, often considered the epitome of military aviation, have been instrumental in shaping the dynamics of modern warfare [1]. These machines are specifically designed and equipped to carry out a wide range of missions such as air-to-air, air-to-ground and reconnaissance missions. It is common practice to plan all missions before the

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: dogan.candemir@tai.com.tr

1877-0509
10.1016/j.procs.2023.12.158

aircraft takes off, formulate a comprehensive plan, and determine a specific path based on mission objectives and operational requirements [2]. However, the dynamic nature of warfare poses significant challenges in the precise execution of pre-planned missions. Combat environments are characterized by uncertainty, rapidly changing conditions, and unpredictable actions by the enemy, necessitating adjustments to the original mission plan [3]. Real-time decision-making and flexibility are critical, as commanders and pilots must be able to make adjustments in the field, change targets, redirect resources, and change flight paths in response to changing circumstances. The limited availability of timely and accurate information can impede informed decision-making and requires commanders to process and analyze incoming data, assess situational developments, and consider changing factors such as enemy movements, weather conditions, and resource availability [4]. Pilots, on the other hand, must interpret and implement real-time instructions while managing operational requirements safely. The need for rapid decision-making and adaptation places tremendous pressure on commanders and pilots, who must carefully balance mission objectives with operational constraints and risk management [5]. So, there is a need for decision support systems that enable dynamic mission planning in modern fighters. If the existing path must be changed during the mission, a new path can be generated, and the mission carried out with the newly generated path.

Several path planning methods have been developed for various fields of application. Grid-based and sampling-based algorithms are commonly used algorithms. Among the grid-based path planning algorithms, A*, Dijkstra, and D* Lite algorithms are widely used. In static environments, grid-based algorithms can be beneficial. However, it is common to use sampling-based methods in dynamic and unknown environments. Probabilistic Roadmap (PRM)[6] and Rapidly-exploring Random Trees (RRT)[7] are the most common sampling-based algorithms. RRT*[8] and RRT-Connect[9] are other tree-based sampling algorithms. RRT* enables a shorter path from the starting configuration to the end configuration than RRT. In the RRT-Connect algorithm, sampling is performed from both the starting and end configurations to create two separate trees, which are then merged when both close each other sufficiently. Despite improved grid-based and sampling-based methods, they still have limitations and shortcomings. Grid-based methods can provide fast results in a known environment. However, in a dynamic environment, pre-determined waypoints may end up in threat zones. Moreover, when a new graph is generated using the Voronoi diagram, positions the aircraft cannot pass through may be produced. Sampling-based methods adapt to dynamics but can become time-consuming in large problem domains, as they sample unnecessary regions. An improved method should address these issues efficiently.

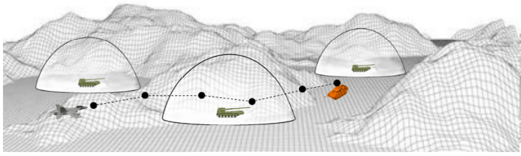
Many existing studies[10, 11, 12, 13] primarily focus on sampling the entire configuration space within an environment. However, a notable limitation among these studies is the omission of the vehicle's speed, which is a crucial factor in real-time path planning scenarios. Moreover, the specific context of path planning for fighter aircraft remains a significant research gap. To address these issues, this study proposes a novel approach that shifts the emphasis from an exhaustive exploration of the entire configuration space to a more tailored strategy. By incorporating vehicle-specific characteristics and constraining the sampling process to a smaller feasible area that accommodates the aircraft's maneuverability, this approach aims to enhance the effectiveness of path planning for fighter aircraft. Specifically, this study proposes an RRT-Connect-based approach as a solution to the specified problem. The method involves creating two trees, with the starting and ending points serving as the root nodes for each tree. For each tree, the nearest node to the other tree is determined. Subsequently, new nodes are incorporated into the trees through sampling, performed within a smaller area centered around the selected node, as opposed to considering the entire environment. This iterative process continues until the nearest nodes of the two trees are sufficiently close to each other. Finally, the trees are merged to form the ultimate tree representing the optimized path. This approach offers several advantages, including improved efficiency and effectiveness in path planning, making it a promising technique for addressing the problem at hand. The main contributions of this paper as follows.

- We propose a novel bidirectional path planning approach based on narrowed regions using RRT Connect. The proposed approach restricts the sampling to regions of interest regardless of the size of the problem domain. This targeted sampling optimizes the search process by increasing the probability of computing the shortest path between the start and end points and offers computational advantages.
- We provide shorter planning time for Single Aircraft Missions with the proposed approach by looking at a limited number of search regions.

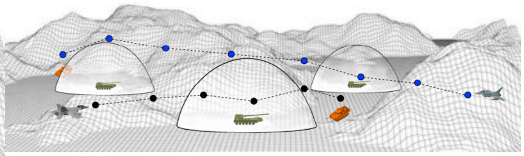
The remainder of the paper is organized as follows. Assumptions and problem descriptions are explained in Section II. Section III includes the proposed approach and algorithm. Section IV consists of the experimental results and evaluations. Finally, Section V concludes this paper.

2. Assumptions

Military missions are typically categorized into two types: single-fighter and multi-fighter missions. These approaches play vital roles in military strategies and tactics. In single-fighter missions, a lone aircraft is deployed to engage enemy targets. In contrast, multi-fighter operations involve the coordination of multiple aircraft to achieve a common objective. Single-fighter missions prioritize individual pilot skills, situational awareness, and adaptability. The aircraft operates independently, utilizing its speed, maneuverability, and weaponry to neutralize enemy threats. Figure 1(a) illustrates a scenario of a single-aircraft mission. The terrain includes air defense systems and a tank that the aircraft must destroy. The hemispherical areas represent the coverage of the air defense systems. Flying through these areas increases the risk of being shot down, so the mission should be executed outside them. The required flight path to reach the tank and the route between these points are indicated by dashed lines.



(a) Single-aircraft mission scenario



(b) Multi-aircraft mission scenario

Fig. 1: Missions in military operations

On the other hand, multi-fighter missions focus on teamwork, coordination, and leveraging the strengths of multiple aircraft. These missions aim to harness the collective capabilities of the aircraft, enhance combat effectiveness, and increase survivability. Tasks within multi-fighter missions can range from escorting friendly aircraft to executing complex attack maneuvers and engaging multiple enemy targets simultaneously. Figure 1(b) illustrates a scenario with two jets and two tanks, emphasizing the coordinated movement of each aircraft towards their designated targets. Flight paths are determined based on threats and mutual coordination.

3. Proposed Approach

This section formally defines our dynamic path-planning approach for single-aircraft missions in complex 3D environments. Our method aims to improve path planning for fighter aircraft missions by creating two trees, rooted at the start and goal positions (T_{init} and T_{goal}). It employs a narrowed sampling strategy guided by the nearest nodes in both trees, focusing exploration on restricted regions. This iterative process generates a connecting tree between the start and goal points. By combining bidirectional Rapidly-Exploring Random Trees (RRT) with narrowed sampling, the algorithm effectively explores regions near the current node, enhancing path planning in constrained aircraft movement scenarios. The optimal trajectory is determined using the A* algorithm. Simulation results across diverse scenarios consistently demonstrate the algorithm's ability to generate smooth and adaptable paths, promising improved mission planning and execution for fighter aircraft in dynamic environments.

Sample: The "sampling" algorithm is designed to perform a random sampling within a narrowed area, denoted S, based on the given initial state and the all threat information contained in $Threats$. This process generates a random configuration by successively calculating the x, y and z coordinates, resulting in the creation of $node_{\text{rand}}$. Then, the algorithm evaluates the proximity of $node_{\text{rand}}$ to each threat by measuring their respective distances. To ensure safety, the algorithm compares these distances with the detection radius of each threat. If $node_{\text{rand}}$ falls within the coverage

Algorithm 1: Narrowed-Regions Based RRT

```
Data: Initial Configuration T_init(x, y, z), Goal Configuration T_goal(x, y, z), Threats Threats, Step Size M
Result: Trajectory
Tree_init ← ∅;
Tree_goal ← ∅;
Trajectory ← ∅;
E ← CreateEnvironment(T_init, T_goal);
Set T_init as root of Tree_init;
Set T_goal as root of Tree_goal;
node_init ← Nearest(Tree_init, T_goal);
node_goal ← Nearest(Tree_goal, T_init);
flag ← init;
while dist(node_init, node_goal) > N do
    if flag == init then
        S_init ← GetSamplingRegion(node_init);
        node_rand ← Sample(S_init, Threats);
        TakeAStep(Tree_init, node_rand, M, Threats);
        node_init ← Nearest(Tree_init, node_goal);
        TakeAStep(Tree_goal, node_init, M, Threats);
        node_goal ← Nearest(Tree_goal, node_init);
        flag ← goal;
    else
        S_goal ← GetSamplingRegion(node_goal);
        node_rand ← Sample(S_goal, Threats);
        TakeAStep(Tree_goal, node_rand, M, Threats);
        node_goal ← Nearest(Tree_goal, node_init);
        TakeAStep(Tree_init, node_goal, M, Threats);
        node_init ← Nearest(Tree_init, node_goal);
        flag ← init;
Tree ← Tree_init ∪ Tree_goal;
Trajectory ← A_star(T_init, T_goal, Tree);
```

Algorithm 2: *Sample(S, Threats)*

```
node_rand ← ∅;
isCollisionOccured ← true;
while isCollisionOccured do
    x ← S(x) + (S(x) − S(R)) · random_number;
    y ← S(y) + (S(y) − S(R)) · random_number;
    z ← S(z) + (S(z) − S(R)) · random_number;
    node_rand ← (x, y, z);
    isCollisionOccured ← false;
    for each threat threat in Threats do
        if norm(threat(x, y, z) - node_rand) ≤ threat(R) then
            isCollisionOccured ← true;
            break;
return node_rand;
```

Algorithm 3: *TakeAStep(tree, node, M, Threats)*

```
node_nearest ← Nearest(tree, node);
node_new ← Nearest(tree, node);
if norm(node_nearest − node) > 0 then
    if norm(node_nearest − node) < M then
        node_new ← node;
        if leg node_nearest → node_new not intersected by any threat in Threats then
            add node_new to tree;
    else
        node_new ← node − node_new;
        node_new ← Nearest(tree, node) + (node_new / norm(node_new)) · M;
        if leg node_nearest → node_new not intersected by any threat in Threats then
            add node_new to tree;
return tree;
```

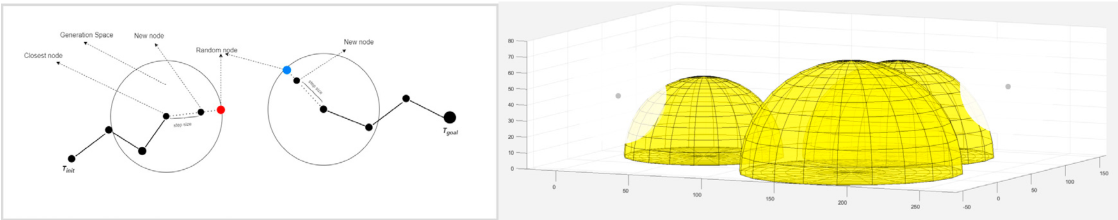
area of a threat, a new random configuration is calculated and the comparison process is repeated. This iterative process continues until a configuration is found that falls outside the coverage areas of all threats. It is worth noting that empirical experiments have consistently shown that this sampling method successfully prevents the algorithm from entering an infinite loop, which underlines its efficiency and reliability in practical applications.

$TakeAStep$: The algorithm works in the context of a tree structure, starting from T_{init} or T_{goal}, and considers a defined node within the environment E, together with the inputs M and $Threats$. The main function of this algorithm is to add a new node to the given tree. The process starts by identifying the $node_{nearest}$ configuration within the tree. Then the algorithm calculates the distance between $node_{nearest}$ and the input node. If the distance is less than M and the path between $node_{nearest}$ and $node_{new}$ does not intersect any threat zones, $node_{new}$ is determined and added to the tree. Conversely, the calculated distance is greater than M, the algorithm continues by advancing $node_{nearest}$ by the distance M in the direction of the input node, which leads to the derivation of $node_{new}$. Again, $node_{new}$ is only included in the tree if the path between $node_{nearest}$ and $node_{new}$ is free of threatening areas.

Our approach employs dynamic sampling to efficiently establish a route between T_{init} and the destination T_{goal} within environment E. Unlike traditional methods that sample the entire environment, we adapt the sampling region based on the aircraft's speed. Using the formula $x = vt$, where x represents the radius of a sphere centered on the aircraft, v is the aircraft's speed, and t is set to 1 second, we dynamically sample the aircraft's immediate surroundings. This real-time adjustment optimizes computational resources and expedites planning by focusing on relevant areas near the aircraft.

Inspired by RRT-Connect principles, our approach utilizes two distinct trees rooted at T_{init} and T_{goal}. The process iteratively selects the closest nodes from each tree, shaping their growth and forming a bridge between the start and target points. We systematically apply a predefined sampling strategy to these selected nodes, optimizing the search process. The algorithm converges toward an optimal trajectory by identifying the closest nodes and iteratively using the sampling approach. Figure 2(a) illustrates this process with two trees originating from T_{init} and T_{goal}. Nodes labeled as $node_{nearest}$ are the closest in their respective trees, and a sampling region is defined around them. Within this region, $node_{rand}$ is generated in the search space of both closest nodes. Moving from $node_{nearest}$ to $node_{rand}$ by a specified step size creates a new node, symbolizing progress in the trajectory space.

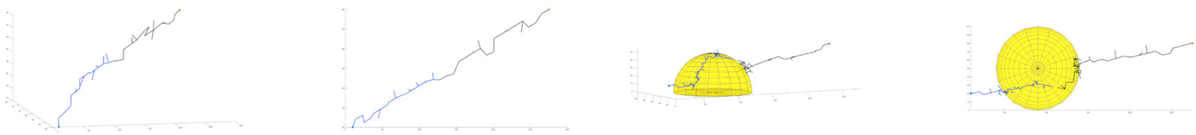
Figure 2(b) provides an illustrative example of environment 'E.' The aircraft's current position is marked in blue, and a red point indicates the desired destination. Yellow semi-spheres represent threat regions, assumed to be at a z-coordinate of 0, denoting ground level. The transparent 3D sphere in the scenario visually symbolizes the concept



(a) Sampling in E

(b) Sample Environment with sampling area around T_{init} and T_{goal}

Fig. 2: Sampling process and overall environment

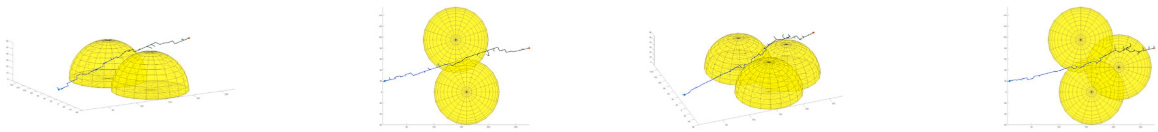


(a) No threat(3D)

(b) No threat(2D)

(c) One threat(3D)

(d) One threat(2D)



(e) Two threats(3D)

(f) Two threats(2D)

(g) Three threats(3D)

(h) Three threats(2D)

Fig. 3: Experimental Scenarios

Table 1: Experimental Results

T_{init}	T_{goal}	Threat 1	Threat 2	Threat 3	Execution Time
(10, 20, 15)	(275, 80, 50)	-	-	-	17 ms
(10, 20, 15)	(275, 80, 50)	(90, 50, 0, 50)	-	-	34 ms
(10, 20, 15)	(275, 80, 50)	(160, 0, 0, 60)	(140, 95, 0, 60)	-	53 ms
(10, 20, 15)	(275, 80, 50)	(160, 0, 0, 60)	(140, 95, 0, 60)	(210, 45, 0, 60)	58 ms

of a sampling area, focusing on specific environmental regions for analysis and planning. This demonstrates the role of adaptive sampling in path planning, enhancing computational efficiency, precision, and navigational pathfinding. Path planning algorithms like A*, Dijkstra, or similar methods can be employed for route planning, with this study utilizing the A* algorithm.

4. Experimental Results

The proposed approach was evaluated through simulations in various scenarios. The algorithms were developed and visualized with Matlab. Also, it was tested on an 11th Gen Intel Core i5 12 CPU 2.75GHz computer using C++ in different scenarios.

Figure 3 shows a series of results showing the $Tree_{init}$ and $Tree_{goal}$ generated from the initial T_{init} and T_{goal} configurations in different test environments in both 3D and 2D representations. Table 1 complements this visual data by providing detailed insights into the execution times for each scenario. Figure 4(a) and 4(b) show that in regions with no threats, the algorithm achieves results in an average time frame of 17 milliseconds. In the other three scenarios, however, the average execution times increase proportionally to the increased threat density in the environment, at 34, 53 and 58 milliseconds, respectively. It is important to emphasise that the speed with which the algorithm produces results in threat-free areas is to be expected, as there are no collisions. Conversely, of course, as the number of environmental threats increases, so does the execution time for the same reasons. Furthermore, variations in execution time were found when the same scenario was repeatedly executed, confirming that the results of the algorithm are not predetermined but adapt to specific conditions. In essence, the narrowed regions- based bidirectional RRT proves capable of generating routes that remain undetectable to adversary elements when deployed in real aircraft. Furthermore, a careful examination of the results shows that redundant sampling is effectively circumvented, supporting the assumption that the developed method provides the most efficient route in a dynamically changing warfare environment, taking into account the current threats.

Unlike other path-planning methods, our approach does not sample irrelevant points. Instead, we consider the target and potential threats in the environment and exclude points that the aircraft cannot fly through. This results in more efficient route planning compared to alternative methods. In each iteration, two points closest to each other are selected from the pre-determined sample points in the init and goal trees, facilitating the sampling process from the starting point to the destination. As a result, the route planning time is significantly reduced compared to other methods. Our future research will expand to include multi-aircraft missions using a unified algorithm for simultaneous path planning. Extensive testing in real-world operational environments is planned to validate our approach's practical applicability. This strategic shift underscores our commitment to advancing aviation mission planning, focusing on enhancing multi-aircraft mission coordination and execution capabilities.

5. Conclusions

In this paper, we introduce a novel approach to path planning for fighter aircraft missions, emphasizing adaptability in dynamic and volatile warfare scenarios. The proposed narrowed regions-based bidirectional path planning, coupled with the modified RRT Connect algorithm, showcases its potential to generate smoother trajectories and enhance mission performance. The contributions of this study lie in the innovative bidirectional path planning strategy, which optimizes search processes by focusing on regions of interest, and the notable reduction in planning time for single aircraft missions. These outcomes underscore the method's effectiveness in achieving efficient and successful mission execution, thereby holding promise for advancements in aircraft path planning techniques in dynamic environments.

References

[1] Robert L. Shaw (1985), "Fighter combat," *Tactics and Maneuvering; Naval Institute Press: Annapolis, MD, USA*.

[2] Nils-Hassan Quttineh and Torbjörn Larsson (2015), "Military aircraft mission planning: efficient model-based metaheuristic approaches," *Optimization Letters*, vol. 9, pp. 1625–1639.

[3] Janice H. Laurence (2011), "Military leadership and the complexity of combat and culture," *Military Psychology*, vol. 23, no. 5, pp. 489–501, *Taylor & Francis*.

[4] Don Harris (2017), "Decision making in aviation," *Routledge*.

[5] Robert E. Enck (2012), "The OODA loop," *Home Health Care Management & Practice*, vol. 24, no. 3, pp. 123–124, *Sage Publications Sage CA: Los Angeles, CA*.

[6] Lydia Kavraki and J-C Latombe (1994), "Randomized preprocessing of configuration for fast path planning," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pp. 2138–2145.

[7] Steven M. LaValle (1998), "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, *Department of Computer Science, Iowa State University*.

[8] Sertac Karaman and Emilio Frazzoli (2011), "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, *Sage Publications Sage UK: London, England*.

[9] James J. Kuffner and Steven M. LaValle (2000), "RRT-connect: An efficient approach to single-query path planning," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 2, pp. 995–1001.

[10] Nancy M. Amato, O. Burchan Bayazit, Lucia K. Dale, Christopher Jones, and Daniel Vallejo (1998), "Choosing good distance metrics and local planners for probabilistic roadmap methods," *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 1, pp. 630–637.

[11] Valérie Boor, Mark H. Overmars, and A. Frank Van Der Stappen (1999), "The Gaussian sampling strategy for probabilistic roadmap planners," *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 2, pp. 1018–1023.

[12] Robert Bohlin and Lydia E. Kavraki (2000), "Path planning using lazy PRM," *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, vol. 1, pp. 521–528.

[13] Fei Yan, Yan Zhuang, and Jizhong Xiao (2012), "3D PRM based real-time path planning for UAV in complex environment," *2012 IEEE international conference on robotics and biomimetics (ROBIO)*, pp. 1135–1140.