2026 12th International Conference on Control, Decision and Information Technologies (CoDIT) | 979-8-3195-2077-7/26/$31.00 ©2026 IEEE | DOI: 10.1109/CoDIT70676.2026.11630880

# Survivability-Oriented Kinodynamic Path Planning for F-16 Aircraft Over DEM-Based Terrain With Dynamic Threats

Bengisu Yücel
*Dept. of Computer Engineering*
*Hacettepe University*
Ankara, Türkiye
yucelbengisu@gmail.com

Mehmet Önder Efe
*Dept. of Computer Engineering*
*Hacettepe University*
Ankara, Türkiye
onderefe@gmail.com

***Abstract*—In the modern combat environment, within the enhancement of defense technology, fighter aircrafts are required to find paths considering dynamic and complicated threats. Enhanced air defense systems and derived threat profiles need to be considered for near real-time state alignment to plan the path for fighter aircraft. However, not only collision avoidance but also aircraft constraints must be reconsidered for safety. This paper demonstrates a kinodynamic path planning and guidance simulation for fighter aircraft over DEM-based real terrain with moving threats. Two planners are introduced: the *Kinematic Guidance Planner (K-GNP)*, which enforces minimum turn-radius feasibility, and the *Tactical Guidance Planner (T-GNP)*, which further reduces risk and control effort using threat-aware and smoothness-based costs. The proposed methods and the other planners (RRT*, PSO, A*, Dijkstra) performances are compared. All comparisons were tested in a common 'Feasibility Engine' and physics-based environment for a fair comparison using Monte Carlo simulations with wind and sensor noise. T-GNP achieves up to 80% reduction in tracking error compared to stochastic planners, with 100% mission success under Monte Carlo disturbances and significantly fewer control saturation events.**

***Index Terms*—Kinodynamic Path Planning, Fighter Aircraft, Guidance, Dynamic Threats, Monte Carlo Simulation.**

## I. Introduction

Advanced fighter jets now rely on threat awareness systems to suggest maneuvers in real time. The pilot still holds decision making authority. The capability to autonomously generate collision flyable trajectories becomes critical, for Unmanned Aerial Systems (UAS) and next generation fighter aircraft. Traditional mission planning typically defines routes during the flight phase based on static intelligence.

A significant challenge in current guidance systems is the discrepancy between geometric path planning and aircraft dynamics. Many algorithms treat the aircraft as a point mass, generating "optimal" geometric paths that violate the kinematic constraints of the platform, such as maximum turn rate or bank angle. This paper addresses this gap by proposing a framework that prioritizes "flyability" alongside optimality. The application can be used for UAS, embedded training and simulation systems for Computer-Generated-Forces (CGF) design. Seeing that the application can make the training more realistic by adding opponents that change. The application can help the pilot become more adaptable. The application can support training, for hard threat scenarios. The simulation can run times and repeat training. The fighter jet can practice moves. The fighter jet can refine tactics. The fighter jet can help develop aerial moves. The fighter jet can boost the pilot training. The fighter jet can improve defense plans in situations.

The problem of path planning in threat environments is studied a lot. Finding that the problem of path planning, in threat environments is usually split into deterministic graph-based methods, probabilistic sampling-based methods and bio-inspired heuristics. The high dimensionality of the flight envelope affects the problem of path planning in threat environments. Because of the dimensionality of the flight envelope many researchers use meta-heuristic algorithms. Researchers use heuristic algorithms to search for solutions. Genetic Algorithms (GA) are heuristic algorithms. Ji et al. Adapt Genetic Algorithms (GA) to add search strategies. The adaptation shows that modeling threat unpredictability helps real-time avoidance. The modeling threat unpredictability improves real-time avoidance [1], [2], [3]. Liu used the Artificial Immune Algorithms to solve the multi-evader scenarios [4]. Kamyar and Taheri made approaches that mix Differential Evolution with Neurofuzzy controllers to deal with the 3D terrain constraints [5]. Many researchers use the Particle Swarm Optimization (PSO) to balance the fuel efficiency and the survival probability [6], [7].

Heuristic methods give flexibility and deterministic algorithms give optimality guarantees. Researchers often use variants of A* algorithm because modified A* algorithm versions run fast. Zhang et al. and Zhao et al. Combined A* algorithm, with Dubins curves and dynamic window approaches to keep the paths kinematically feasible [9], [10], [15]. Chen et al. showed that focused D* can handle uncertainty and can update paths quickly [11]. Accurate threat modeling is critical. Ling-Xiao and Zhang et al. added the Radar Cross-Section

This work was supported by Turkish Aerospace Industries (TAI).

(RCS) minimization. Ling-Xiao and Zhang et al. emphasized that stealth is as vital, as avoidance [12], [9]. Voronoi diagrams are used for dynamic programming to map the threat boundaries and to improve the penetration trajectories, [13], [14]. There is still a gap in the research. Heuristic methods are usually too slow, for real-time decisions. The standard A* algorithm cannot deal with aircraft motion without a lot of changes. To fix this we suggest the Kinematic Guidance Planner (K-GNP). The Kinematic Guidance Planner (K-GNP) does not use a grid. The Kinematic Guidance Planner (K-GNP) uses motion pieces that come from a three-degree-of-freedom aircraft model. By putting the bank angle limits, inside the search loop K-GNP makes sure that each path segment is flyable. K-GNP avoids saturation.

Mission success inherently depends on both flyability and survivability. To address this, we propose the Tactical Guidance Planner (T-GNP), which utilizes a risk-driven cost function to balance threat exposure with trajectory smoothness. The primary contribution of this work is an experimentally validated, implementation-aware algorithmic framework rather than a theoretical theorem. Specifically, we formulate the fixed-wing tactical path-planning problem in a heading-augmented search space with explicit coordinated-turn constraints, introducing two planners: K-GNP and T-GNP. Furthermore, we provide a unified benchmarking environment to evaluate deterministic, stochastic, and learning-based approaches under identical terrain, threat, and disturbance conditions. Planners are rigorously assessed using tracking error, dynamic violations, and overall success rates, supported by extensive ablation analyses to demonstrate the practical value of the proposed risk-perception components.

This paper is organized as follows: Section II details the system modeling, including aircraft dynamics and environmental constraints. Section III introduces the proposed K-GNP and T-GNP architectures. The experimental setup and mission scenarios are outlined in Section IV, followed by the simulation results and discussion in Section V. Finally, the conclusions are presented.

## II. System Modeling and Problem Formulation

The system architecture has input parameters and constraint models. In the field of aviation, at the mission planning phase, the start point and the goal radii is given. The problem is given as regarding of the real terrain area with dynamics threats aircraft model must follow the suggested path to accomplish task. We define this operation as a non-linear optimal problem with many constraints both environmental and dynamics.

### A. 3- DoF Point Mass Aircraft Constraint Model

We use a 3-DOF Point-Mass Model to represent the aircraft. This model is enough to capture the important kinematic limits for our guidance tests. We assume that the inner-loop controllers are already handling the fast rotations, so we can directly send commands for the bank angle ($\phi$) and flight-path angle ($\gamma$). For the simulation, we use parameters from the standard F-16 model by Stevens and Lewis, [16]. We set

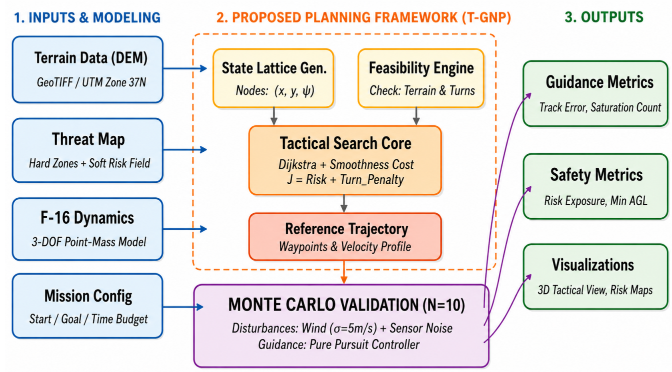


Fig. 1. The proposed planning architecture for survivability and kinodynamic constraints.

the mass $m = 12\,000\,\text{kg}$ and use standard drag values. Even though an F-16 can pull up to $9g$, we use a more conservative bank limit of $|\phi|_{max} = 60^\circ$ (which is about $2.0g$). This is to make sure the flight is smooth and safe while flying very close to the ground (nap-of-the-earth). The state vector is $\mathbf{x} = [x, y, h, \psi, V]^T$ (position, altitude, heading, and speed). The control inputs are $\mathbf{u} = [\delta_T, \phi, \gamma]^T$ (throttle, bank, and flight-path angle).

*1) Equations of Motion:* The aircraft dynamics and environmental effects are defined by the following set of equations:

$$\dot{x} = V\cos\gamma\cos\psi + W_x, \tag{1}$$

$$\dot{y} = V\cos\gamma\sin\psi + W_y, \tag{2}$$

$$\dot{h} = V\sin\gamma, \tag{3}$$

$$\dot{\psi} = \frac{g\tan(\phi + \eta_\phi)}{V}, \tag{4}$$

$$\dot{V} = \frac{T(\delta_T) - k_{drag}V^2}{m}. \tag{5}$$

In these equations, $W_x$ and $W_y$ are stochastic wind gusts that cause drift. We also added Gaussian noise $\eta_\phi \sim \mathcal{N}(0, \sigma_\phi^2)$ to the bank angle command to simulate real-world sensor errors during Monte Carlo tests. Drag is calculated as $k_{drag}V^2$ with $C_{D0} \approx 0.025$. Equation (4) is very important because it defines the minimum turning radius $R_{\min} = V^2/(g\tan\phi_{\max})$. This tells us whether a path is actually flyable or not.

### B. Environmental Modeling Specifications

To build simulation environment this paper demonstrates, real terrain file and added some synthetic and predefined threat zones for realistic mission pre-planning.

*1) Terrain Data (DEM):* The high-resolution Digital Elevation Models (DEM) from the Eastern Black Sea region, Türkiye is used. The terrain has boundaries, UTM coordinates is between (585500, 5054000) - (587200, 5055050). Threat center locations are relocated in (586500, 5054500).

- **Data Format:** We use GeoTIFF data projected into UTM Zone 37N. This allows us to calculate distances in meters easily.

- **Safety:** For safety and realistic perspective to avoid ground collision there is a hard constraint for safe altitude:

$$h(t) \geq H_{DEM}(x(t), y(t)) + h_{AGL}, \tag{6}$$

where, $H_{DEM}$ is the highest point of the mountain and $h_{AGL}$ is our safety margin (set to 100 m).

*2) Surface Threat Constraint Model:* Surface threats models, SAM sites or radars are modeled on the map. The effects area is demonstrates as a sphere form. Let $\mathbf{c}_i(t)$ be the center of the threat. We use two zones for each threat:

*a) Risk Potential Field:* Instead of just saying "inside" or "outside," we use a smooth potential field $J_{risk}$. This helps the optimizer see which paths are safer. The cost depends on the distance $d(\mathbf{p}, t)$ from the threat.

- **Hard Kill Zone ($R_{hard}$):** This is the "no-go" zone. If the plane enters this radius, the risk is infinite because the $P_{kill}$ is almost 1.0.

$$C_{hard}(\mathbf{p}) = \infty \quad \text{if } d(\mathbf{p}, t) \leq R_{hard} \tag{7}$$

- **Soft Risk Zone ($R_{soft}$):** An outer identification zone ($R_{hard} < d < R_{soft}$). To prevent sharp gradients, the risk cost is modeled using a cubic polynomial decay:

$$J_{risk}(\mathbf{p}) = \sum_{i=1}^{N_{threat}} \lambda_i \cdot \left[ \max\left(0, \frac{R_{soft} - d(\mathbf{p}, t)}{R_{soft} - R_{hard}}\right) \right]^3 \tag{8}$$

where $N_{threat}$ denotes the total number of active adversarial units in the theater. The parameter $\lambda_i \in [0.5, 1.0]$ represents the severity weight of the $i$-th threat (e.g., $\lambda = 1.0$ for SAMs, $\lambda = 0.5$ for surveillance radars), scaled by a global cost factor ($w_{risk} \approx 5000$).

*b) Temporal Dynamics:* Threats in our simulation are not static; they move to simulate a real battlefield. The position $\mathbf{c}_i(t)$ changes over time:

- **Circular Scanning:** This represents a radar threat scanning or another plane orbiting a point.

$$\mathbf{c}_i(t) = \mathbf{c}_{base} + R_{orb} \begin{bmatrix} \cos(\omega t) \\ \sin(\omega t) \end{bmatrix} \tag{9}$$

- **Linear Sweeping:** This represents a mobile SAM launcher moving back and forth on a patrol line. The code uses a phase-based calculation to move the unit along the X-axis within its defined patrol length.

$$\mathbf{c}_i(t) = \mathbf{c}_{base} + \text{offset}(t) \cdot \hat{\mathbf{i}} \tag{10}$$

where, the offset moves between negative and positive half-lengths of the path.

### C. Optimization Objective

Our main goal is to find the best control sequence $\mathbf{u}(t)$ from start to finish ($T_f$). We want to minimize the total cost $J$:

$$J = \int_0^{T_f} \left( w_1 + w_2 J_{risk}(\mathbf{x}) + w_3 |\dot{\psi}|^2 \right) \mathrm{d}t, \tag{11}$$

While following the dynamics in Equations (1)-(5). The term $\|\dot{\psi}\|^2$ is added so the aircraft does not perform aggressive turns and keeping the trajectory smooth.

## III. Path Planning Architectures

To carefully evaluate the balance between geometric optimality and kinematic feasibility, a multi-level benchmarking test is used. A key part of this test is a shared same feasibility engine and constraints, which ensure that all algorithms—independent of their search strategy—use the same environmental constraints and the 50 m collision-detection resolution defined in Section II.

### A. Geometric Baselines (Graph-Based)

These deterministic methods discretize the environment into an 8-connected grid. To keep the computation manageable while preserving geometric optimality in Euclidean space, the following algorithms are used:

- **Dijkstra's Algorithm:** Performs an exhaustive uniform-cost search. It guarantees the shortest geometric path, but models the aircraft as a holonomic point and does not consider turn-radius constraints ($R_{\min}$).
- **A* Search:** Extends Dijkstra's algorithm by using a Euclidean distance heuristic ($h(n) = |\mathbf{p}n - \mathbf{p}\text{goal}|$). In our implementation, a heuristic weight of $w_h = 1.5$ is applied to improve search speed in large-scale terrains (greater than 50km).

### B. Stochastic Baselines (Sampling-Based)

These methods operate in continuous space and are used to evaluate the performance of randomized planners under a fixed time limit ($T_{\max} = 2.0$s).

- **RRT* (Rapidly-exploring Random Tree Star):** Constructs a probabilistic tree with asymptotic optimality. However, standard RRT* connects nodes using straight-line segments without enforcing curvature continuity, which often requires strong post-processing.
- **PSO (Particle Swarm Optimization):** Unlike standard PSO, which assumes an unconstrained search space, our implementation uses a *corridor-based* sampling strategy. The particles optimize a sequence of ordered waypoints that are strictly confined to a local corridor between the start and goal states, reducing convergence to infeasible local solutions.

### C. Proposed Methodology: Guidance-Aware Lattice Planner

We propose a state-lattice planner that embeds the F-16's kinematic limits directly into the graph generation process. The search is conducted in the directed state space $\mathbf{s} = (x, y, \psi)$, discretized with a spatial resolution of 200 m and an angular resolution of 10°.

*1) K-GNP (Kinematic Guidance Planner):* The baseline planner (F16-Basic) generates motion primitives based on the minimum turn radius, derived from the maximum sustained bank angle ($\phi_{\max} = 60°$) at cruise velocity:

$$R_{\min} = \frac{V_{cruise}^2}{g \cdot \tan(\phi_{\max})} \tag{12}$$

A transition between state $\mathbf{s}_i$ and $\mathbf{s}_j$ is considered valid only if the required heading change $|\Delta\psi|$ satisfies the curvature constraint imposed by $R_{\min}$ over the segment length.

*2) T-GNP (Tactical Guidance Planner):* The proposed tactical variant extends K-GNP by modifying the edge cost function to actively penalize sharp maneuvering. While K-GNP treats all feasible turns equally, T-GNP introduces a *Turn Penalty Heuristic* to minimize actuator saturation and "bang-bang" control behavior. The augmented cost function for a transition is defined as:

$$J(\mathbf{s}_i, \mathbf{s}_j) = J_{dist} + \lambda_{turn} \cdot \left(\frac{|\Delta\psi|}{\pi}\right) \cdot J_{dist} \tag{13}$$

where, $J_{dist}$ is the Euclidean segment length and $\lambda_{turn}$ is a tunable gain (set to $1.5$ in our experiments). This term scales the cost proportionally to the severity of the heading change relative to the segment length, guiding the search towards smoother, flyable trajectories that naturally align with the guidance system's capabilities.

## IV. Experimental Setup

To evaluate the proposed planning architectures, high-resolution Digital Elevation Models (DEM) of the Eastern Black Sea region are used. The test set includes three different operational scenarios, each designed to test specific guidance capabilities

TABLE I
Benchmark Configuration and Algorithm Parameters

| Parameter | Value |
|---|---|
| **I. Common Environmental Constraints (Fairness Baseline)** *(Applied strictly to ALL algorithms via Feasibility Engine)* | |
| Terrain Data Source | Eastern Black Sea DEM |
| Safety Altitude Margin ($h_{\text{AGL}}$) | 100 m |
| Feasibility Check Resolution | 50 m |
| Dynamic Threat Radius | Varies per scenario (Hard/Soft zones) |
| **II. Shared Aircraft Dynamics** | |
| Aircraft Mass ($m$) | 12 000 kg |
| Cruise Velocity ($V_{\text{cruise}}$) | 220 m/s |
| Max. Bank Angle ($\phi_{\max}$) | 60° |
| Min. Turn Radius ($R_{\min}$) | $\approx$ 2800 m |
| **III. Algorithm-Specific Tunings** | |
| *Deterministic (A*, Dijkstra):* | |
| Grid Resolution | 200 m |
| Heuristic Weight ($w_h$) | 1.5 (A* only) |
| *Stochastic (RRT*, PSO):* | |
| PSO Configuration | 150 particles, 8 waypoints |
| RRT* Step Length | 2500 m |
| RRT* Search Radius | 3000 m |

The benchmark environment applies the same global constraints to all planners, as detailed in Table I, common parameters—such as terrain safety margins ($h_{AGL}$) and aircraft dynamics ($R_{min}$)—are fixed to guarantee that no algorithm has an unfair advantage regarding physical limits. However, to evaluate each planner at its peak performance, we tuned the algorithm-specific parameters (e.g., particle count for PSO, step size for RRT*) individually. This approach ensures fairness in the problem definition while allowing each method to operate with its optimal settings.

### A. Mission Scenarios

Table II summarizes the mission plan scenarios for each scenario.

**Scenario 1: Base Scenario** The short range mission plan ($L \approx 40$ km) has three static threats. The main goal is to verifying that generated trajectories satisfy the minimum turn radius ($R > R_{min}$) constraints in an unconstrained environment.

**Scenario 2: High-Density Threat Environment** This scenario simulates a complex Integrated Air Defense System (IADS), where multiple SAM engagement zones overlap. It forces the planner to navigate narrow "safety corridors" constrained horizontally by threat boundaries and vertically by terrain features.

**Scenario 3: Long-Range Dynamic Mission** A strategic flight profile ($L \approx 140$ km) incorporating moving adversaries. This scenario demonstrates the flight regarding of moving threats to understand time-varying threats effects on planning algorithms during flight.

TABLE II
Operational Mission Scenarios

| Scenario | Objective |
|---|---|
| S1 (Base) | Verify kinematic feasibility in sparse environments. |
| S2 (Dense) | Navigate narrow corridors in high-density clutter. |
| S3 (Long) | Ensure resilience against dynamic threats over long range. |

### B. Performance Evaluation Metrics

To rigorously assess the trade-off between geometric optimality and operational feasibility, we report the following metrics:

*1) Computational Efficiency:*

- **CALC (s):** The wall-clock time required to generate the initial path.
- **CHECKS:** The total number of collision and feasibility queries performed ($N_{check}$).

*2) Operational Safety:*

- **RISK ($10^6$):** The cumulative threat exposure cost accumulated along the trajectory.
- **AGL (m):** The minimum Altitude Above Ground Level. A value below the safety margin (100 m) indicates a collision risk.
- **T.FAIL:** Threat Failure Count. The number of steps where the aircraft breached a hard-kill zone.

*3) Guidance Performance & Flyability:*

- **SUCC (%):** Mission success rate over $N = 10$ runs.
- **TRACK (m):** Mean Tracking Error. High values indicate the planner ignored kinematic constraints.
- **MAX.CTE (m):** Maximum Cross-Track Error. The peak deviation from the path (overshoot).
- **D.SAT:** Dynamic Saturation Count. The number of steps where the autopilot command exceeded limits ($|\phi| > 60°$). High saturation implies the path is "unflyable."

### *C. Stochastic Protocol*

To ensure fairness, deterministic planners (K-GNP, T-GNP) are subjected to a **Monte Carlo Simulation** ($N = 10$) with randomized wind gusts ($\sigma_{wind} = 5$ m/s) and sensor noise ($\sigma_{noise} = 0.5°$). Stochastic planners (PSO, RRT*) are executed 10 times to evaluate convergence reliability.

## V. Simulation Results and Discussion

In this section, we compare the results of our proposed algorithms (K-GNP and T-GNP) with standard methods. We used Monte Carlo simulations ($N = 10$ runs) to include the effects of random wind and sensor noise.

### *A. Scenario 1: Kinematic Validation Baseline*

This scenario was a basic test in a safe environment to check if the planners create flyable paths. The results are shown in Table III.

*1) Geometric vs. Kinematic Results:* As expected, **A*** and **Dijkstra** found the shortest paths ($\approx$ 39.3 km). However, Dijkstra was very slow (113.07 s) because it had to check too many points ($>$ 1.3 million) on the grid. This shows that standard Dijkstra is too slow for this problem.

*2) Problems with Stochastic Planners:* The sampling-based methods (PSO and RRT*) had problems with the short time limit (2.0 s). **PSO** only had an 80% success rate and a very high Mean Tracking Error of 628.65 m. This high error happens because the random waypoints create sharp turns that the autopilot cannot follow. **RRT*** also had a high Max Cross-Track Error (1019.2 m).

*3) Proposed Methods:* Both **K-GNP** and **T-GNP** achieved 100% success. In this open area, **K-GNP** had the lowest tracking error (28.95 m). This shows that using minimum turn radius ($R_{min}$) limits in the graph is enough for basic flight safety.

### *B. Scenario 2: High-Density Threat Environment*

In this scenario, the planners are evaluated in narrow corridors between threat regions. The quantitative results are reported in Table IV.

*1) Performance in Narrow Corridors:* In this challenging environment, the stochastic planners show reduced performance. The success rate of **PSO** decreases to 70%, and the Mean Tracking Error exceeds 500,m. This indicates that the algorithm frequently converges to local minima in highly cluttered environments.

*2) Precision of Lattice Planners:* **K-GNP** achieves strong performance with a Mean Tracking Error of 23.10,m, outperforming the geometric **A*** planner (37.02,m). Since K-GNP enforces kinematic feasibility during planning, the resulting trajectories remain closer to the center of the safe corridor. **T-GNP** also demonstrates robust behavior, achieving a 100% success rate.

### *C. Scenario 3: Long-Range Dynamic Mission*

This scenario has long range start to goal destination (52 km) with moving surface threats. The results in Table V are shown.

*1) Reduced Control Saturation:* Both **K-GNP** and **T-GNP** complete the mission successfully with a 100% success rate; however, their flight behaviors differ. **K-GNP** experiences 116 Dynamic Saturation ($D_{\text{sat}}$) events, indicating frequent saturation of the aircraft's bank angle limits.

In contrast, **T-GNP** reduces the number of saturation events to 48, corresponding to an approximate **58%** reduction. This demonstrates that the turn-penalty term in the cost function effectively discourages aggressive maneuvers.

On the other hand, **T-GNP** reduced these events to just 48. This is a reduction of about **58%**. This shows that the "Turn Penalty" in our cost function effectively prevents aggressive maneuvers.

*2) Improved Tracking Accuracy:* The smoother trajectories generated by **T-GNP** lead to improved guidance performance. **T-GNP** achieves the lowest Mean Tracking Error (20.54,m) among all evaluated methods, outperforming **K-GNP** (39.08,m) and **A*** (40.19,m). This result suggests that penalizing sharp turns during planning improves flight stability and tracking accuracy.

## VI. Conclusion

The main contribution of this study is a practically validated algorithmic framework for tactical fixed-wing path planning. Simulation results demonstrate that minimizing Euclidean distance or time efficiency alone does not guarantee operational success. Standard planning baselines and stochastic methods (e.g., RRT* and PSO) often yield "unflyable" trajectories due to high control saturation ($D_{sat}$) or inconsistent performance in narrow terrain corridors. In contrast, the proposed T-GNP architecture embeds physical aircraft limitations ($R_{min}$) and smoothness objectives directly into the search graph, penalizing sharp heading changes to enhance survivability. Consequently, T-GNP achieved a 100% mission success rate across all scenarios and reduced control saturation events by approximately 58% compared to the basic kinematic planner.

Future work will expand the T-GNP framework to higher-fidelity 6-DOF aircraft dynamics, incorporating advanced constraints such as angle-of-attack (AoA) limits, G-load factors, and fuel-aware optimization. Additionally, it could be integrate this kinodynamically feasible planner into tactical environment simulations to develop high-fidelity Computer Generated Forces (CGF) for realistic pilot training and evaluate its compatibility with AI-based navigation algorithms.

TABLE III
GNC BENCHMARK RESULTS - SCENARIO 1: KINEMATIC VALIDATION BASELINE ($T_{budget} = 2.0$ S)

| Algorithm | Status | Succ (%) | Path (km) | Time (s) | Risk ($10^6$) | AGL (m) | D.SAT | Track (m) | Max CTE (m) | Calc (s) | Checks |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **A*** | Success | 100 | 39.29 | 156.05 | 0.00 | 655.8 | 33 | 33.93 | 171.7 | 6.10 | 67,801 |
| **Dijkstra** | Success | 100 | 39.66 | 153.57 | 0.00 | 660.1 | 108 | 49.61 | 339.1 | 113.07 | 1,316,265 |
| **RRT*** | Success | 90 | 42.53 | 171.64 | 0.03 | 701.7 | 344 | 187.29 | 1,019.2 | 2.01 | 11,685 |
| **PSO** | Success | 80 | 52.08 | 214.94 | 0.02 | 699.0 | 421 | 628.65 | 3,336.5 | 2.00 | 438 |
| **K-GNP** | **Success** | **100** | **39.38** | **155.72** | **0.00** | **703.1** | **39** | **28.95** | **156.0** | **20.09** | **9,043** |
| **T-GNP** | **Success** | **100** | 39.53 | 155.35 | 0.00 | 702.6 | 81 | 36.19 | 181.9 | 30.01 | 12,353 |

TABLE IV
GNC BENCHMARK RESULTS - SCENARIO 2: HIGH-DENSITY THREAT ENVIRONMENT ($T_{budget} = 3.0$ S)

| Algorithm | Status | Succ (%) | Path (km) | Time (s) | Risk ($10^6$) | AGL (m) | D.SAT | Track (m) | Max CTE (m) | Calc (s) | Checks |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **A*** | Success | 100 | 43.62 | 173.80 | 0.00 | 786.0 | 33 | 37.02 | 186.4 | 5.53 | 62,353 |
| **Dijkstra** | Success | 100 | 44.20 | 176.55 | 0.00 | 814.6 | 85 | 36.84 | 190.3 | 132.37 | 1,594,509 |
| **RRT*** | Success | 90 | 45.37 | 184.36 | 0.08 | 732.6 | 354 | 232.10 | 1,171.3 | 3.01 | 20,405 |
| **PSO** | Success | 70 | 56.43 | 239.89 | 0.10 | 728.4 | 439 | 512.04 | 3,174.9 | 3.00 | 677 |
| **K-GNP** | **Success** | **100** | **44.32** | **179.90** | **0.00** | **821.2** | **39** | **23.10** | **157.9** | **27.74** | **11,731** |
| **T-GNP** | **Success** | **100** | 44.49 | 179.05 | 0.00 | 823.8 | 57 | 30.56 | 169.7 | 38.05 | 15,469 |

TABLE V
GNC BENCHMARK RESULTS - SCENARIO 3: LONG-RANGE DYNAMIC MISSION ($T_{budget} = 5.0$ S)

| Algorithm | Status | Succ (%) | Path (km) | Time (s) | Risk ($10^6$) | AGL (m) | D.SAT | Track (m) | Max CTE (m) | Calc (s) | Checks |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **A*** | Success | 100 | 52.35 | 210.85 | 0.00 | 772.0 | 61 | 40.19 | 167.7 | 13.38 | 143,529 |
| **Dijkstra** | Success | 100 | 51.86 | 211.05 | 0.00 | 684.3 | 116 | 41.63 | 334.0 | 183.01 | 2,135,602 |
| **RRT*** | Success | 100 | 55.87 | 231.95 | 0.06 | 732.4 | 356 | 165.18 | 1,045.6 | 5.02 | 40,085 |
| **PSO** | Success | 100 | 53.62 | 217.80 | 0.01 | 699.1 | 192 | 96.97 | 730.8 | 5.00 | 1,074 |
| **K-GNP** | Success | 100 | 52.30 | 211.95 | 0.00 | 726.6 | 116 | 39.08 | 165.2 | 39.60 | 18,574 |
| **T-GNP** | **Success** | **100** | **52.59** | **217.85** | **0.00** | **785.8** | **48** | **20.54** | **172.2** | **55.56** | **24,652** |

## REFERENCES


[1] W. Ji, H. Xu, G. Wang, and X. Yang, "Path Planning under Dynamic Threat Environment," *Applied Mechanics and Materials*, vol. 543-547, pp. 1790–1794, 2014.

[2] W. Ji, C. Li, and Y. Zhang, "Aircraft trajectory planning based on threat sources in uncertain environments," in *2024 4th International Signal Processing, Communications and Engineering Management Conference (ISPCEM)*, 2024, pp. 693–697.

[3] F. Qi, W. Zhenyu, and B. Zhaohui, "Aircraft path planning based on GA in the moving threats environment," in *Proceedings of the 32nd Chinese Control Conference*, 2013, pp. 5107–5110.

[4] L. Liu, "Airplane Trajectory Planning Using Artificial Immune Algorithm," in *Lecture Notes in Electrical Engineering*, Springer, 2020, pp. 245–251.

[5] R. Kamyar and E. Taheri, "Aircraft Optimal Terrain/Threat-Based Trajectory Planning and Control," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 2, pp. 466–483, 2014.

[6] T. Turker, O. Sahingoz, and G. Yilmaz, "2D path planning for UAVs in radar threatening environment using simulated annealing algorithm," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 56–61.

[7] L. Swartzentruber, J. Foo, and E. Winer, "Multi-Objective UAV Path Planning with Refined Reconnaissance and Threat Formulations," in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, 2010.

[8] H. Chen, H. Wang, and L. Jiang, "Path planning of UAV based on cultural algorithm in dynamic environments," in *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2016, pp. 130–134.

[9] Z. Zhang, J. Jiang, J. Wu, and X. Zhu, "Efficient and optimal penetration path planning for stealth unmanned aerial vehicle using minimal radar cross-section tactics and modified A-Star algorithm," *ISA Transactions*, 2022.

[10] W. Zhao, Z. Meng, and P. W. Chung, "A heuristic particle swarm optimization with dynamic window approach for mobile robot path planning," *Applied Soft Computing*, 2022.

[11] X. Chen, D. Liu, and T. Tang, "Path Planning for UCAV in Dynamic and Uncertain Environments Based on Focused D* Algorithm," in *2011 Fourth International Symposium on Computational Intelligence and Design*, vol. 2, pp. 55–58, 2011.

[12] Lingxiao, W. (2009) 'Effective path planning method for low detectable aircraft *Journal of Systems Engineering and Electronics*, 20(4), pp. 784–789, 2009.

[13] S. Xiaoli, W. Xinmin, L. Yongcai, W. Changqing, and X. Cheng, "Optimization of Fighter Aircraft Evasive Trajectories for Radar Threats Avoidance," in *2007 IEEE International Conference on Control and Automation*, 2007, pp. 303–307.

[14] Q. Xiao, X. Gao, X. Fu, and H. Wang, "New Local Path Replanning Algorithm for Unmanned Combat Air Vehicle," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 1, pp. 4033–4037, 2006.

[15] P. Maini and P. B. Sujit, "Path planning for a UAV with kinematic constraints in the presence of polygonal obstacles," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 62–67.

[16] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2016.