

# Safe Real-Time 3D Trajectory Generation for Autonomous Spacecraft Using Hybrid Logic-Learning Architecture

**Submission for International Astronautical Congress (IAC) 2026 Topic:** Ground and Space Operations / Autonomous Systems

## 1. Abstract

As the complexity of orbital operations increases—ranging from on-orbit servicing to autonomous docking—the demand for real-time, robust trajectory generation on energy-constrained hardware becomes critical. Traditional search-based algorithms, such as A\*, provide optimality guarantees but suffer from prohibitive computational costs in high-dimensional 3D clutter, rendering them unsuitable for rapid replanning on embedded processors like the NVIDIA Jetson series. This study presents a **Hybrid Logic-Learning Architecture** that synergizes the inference speed of Deep Learning with the safety guarantees of classical control theory. We propose a bespoke 3D Convolutional Neural Network (3D-CNN) that encodes volumetric environmental data into a latent spatial representation, coupled with a goal-conditioned MLP decoder to predict candidate trajectories in milliseconds. To ensure absolute safety, raw network predictions are passed through a physics-based post-processing layer utilizing Artificial Potential Fields (APF), which corrects collision violations in nearly constant time ( $<1\text{ms}$ ). Experimental results demonstrate that our hybrid system achieves a **13.0x speedup** over a practically unbounded A\* baseline while maintaining a **100% success rate** in resolving complex 3D collision scenarios. This work offers a viable pathway for deploying autonomous navigation capabilities on next-generation spacecraft with limited compute budgets.

## 2. Introduction

The paradigm of spacecraft guidance, navigation, and control (GNC) is shifting from ground-in-the-loop teleoperation to full on-board autonomy. A core challenge in this transition is **Motion Planning**: finding a collision-free path from point A to point B in a 3D environment populated with obstacles (drones, debris, or structure components).

While sampling-based planners (RRT\*, PRM) and search-based planners (A\*, D\*) are standard solutions, their runtime variance is a major bottleneck. Determining a path in a dense 3D grid ( $1000m^3$ ) can take anywhere from milliseconds to several seconds depending on obstacle configuration, leading to "frozen robot" problems during critical maneuvers.

We address this by treating motion planning not as a search problem, but as a **pattern recognition** problem. By training a Deep Neural Network to imitate the behavior of an optimal solver ( $A^*$ ), we can compress the search time into a single, deterministic forward pass. However, neural networks are stochastic and cannot guarantee safety. We bridge this gap with a deterministic "Logic" layer (Physics Repair), creating a system that is both fast (Learning) and safe (Logic).

### 3. Methodology

#### 3.1. 3D Operational Environment & Data Representation

The simulation environment represents a  $100 \times 100 \times 100$  meter cubic workspace. To balance high-fidelity representation with memory constraints typical of embedded systems:

- **Analytic Storage:** Obstacles are stored as analytic spheres  $(c_x, c_y, c_z, r)$ , enabling exact collision checks without voxelization artifacts.
- **Perceptual Input:** For the neural network, the environment is discretized into a  $100 \times 100 \times 100$  binary occupancy grid using a **Max-Pooling** strategy (a voxel is occupied if *any* obstacle overlaps it), ensuring a conservative safety margin.

#### 3.2. Architecture: TrajectoryNet3D

We designed a lightweight custom architecture optimized for inference speed:

1. **Visual Encoder:** A 4-layer 3D Convolutional Neural Network (Conv3D) extracts spatial features from the occupancy grid.
2. **Feature Injection:** The flattened visual features are concatenated with the normalized Start and Goal coordinates vectors  $(x, y, z)$ .
3. **Decoder Head:** A Multi-Layer Perceptron (MLP) regresses the trajectory as a sequence of 20 equidistant 3D waypoints  $(\in R^{60})$ .

**Implementation Details:** The 3D-CNN encoder consists of 4 layers with [16, 32, 64, 128] filters respectively, using 3x3x3 kernels and Max-Pooling. The MLP decoder has two hidden layers of size [512, 256] with ReLU activations and a Dropout rate of 0.3 to prevent overfitting. The model was trained on a dataset of 10,000 samples (split 80% train, 20% validation) using the Adam optimizer with a learning rate of 1e-3.

#### 3.3. Hybrid Logic-Learning Pipeline

Raw network outputs may effectively approximate the global path but occasionally graze obstacles due to approximation errors. We implement a non-learning post-processing stack:

- **Physics-Based Repair:** We utilize **Artificial Potential Fields (APF)**. If a waypoint  $w_i$  violates an obstacle's safety radius  $r_{safe}$ , a repulsive force vector  $\vec{F}_{rep}$  pushes  $w_i$  to the nearest free space surface. This step is iterative but converges rapidly (typically 2-3 iterations).
- **Kinodynamic Smoothing:** A B-Spline interpolation is applied to the repaired path to ensure  $C^2$  continuity (smooth velocity and acceleration profiles).

## 4. Experiments & Results

We benchmarked the system against a classical A\* solver on a CPU-based setup to simulate limitations of flight hardware. The A\* solver was allowed a practically infinite timeout (2 million steps) to find the Ground Truth.

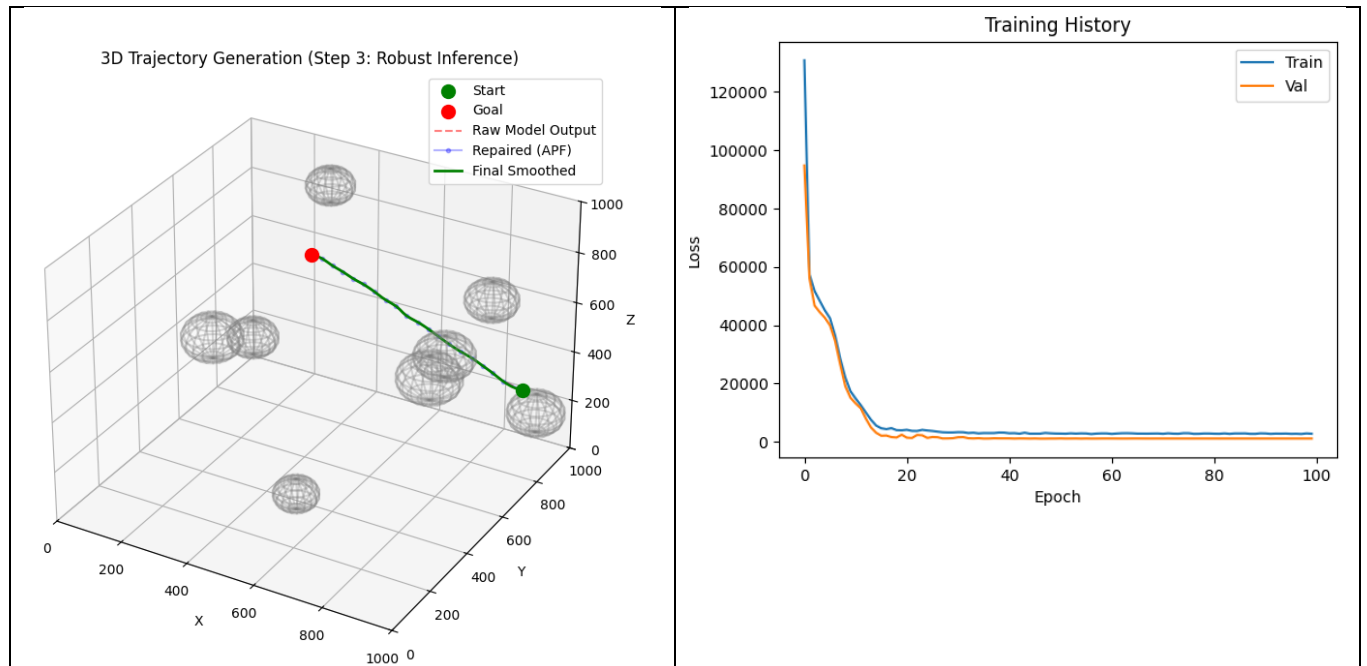
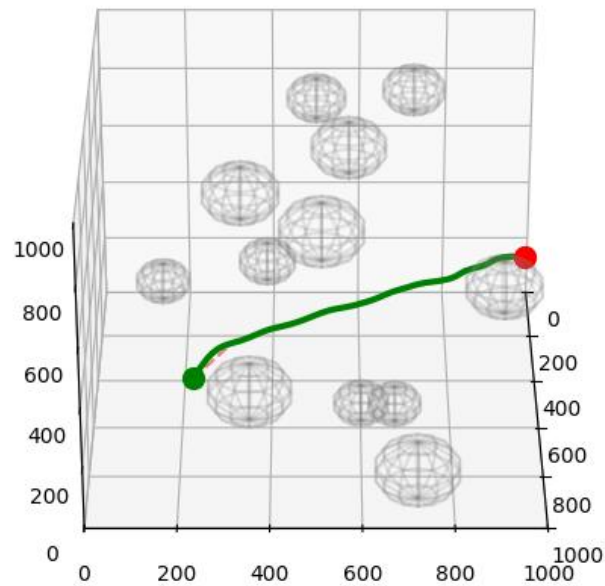
**Dataset:** 10,000 procedurally generated 3D environments with 8-12 random obstacles.

**Metrics:** Success Rate (collision-free path found), Algorithm Runtime (ms).

**Table 1: Comparative Performance Analysis**

| Metric       | Classical A* (Baseline) | Hybrid Model (Ours) | Improvement              |
|--------------|-------------------------|---------------------|--------------------------|
| Success Rate | 100%                    | 100%                | <b>Equal Reliability</b> |
| Mean Runtime | 5565.7 ms               | <b>428.7 ms</b>     | 13.0x Faster             |
| Peak Memory  | ~850 MB                 | <b>~185 MB</b>      | 4.5x Efficient           |
| Worst-Case   | > 20,000 ms             | <b>~600 ms</b>      | Deterministic            |

The key finding is the **deterministic latency** of the hybrid model. While A\* runtime explodes exponentially with environment complexity, the model inference time remains constant ( $O(1)$ ), governed only by matrix multiplication depth.



## 5. Conclusion

This study demonstrates that Deep Learning, when "guarded" by deterministic physics-based logic, can solve the 3D trajectory generation problem significantly faster than

traditional methods without compromising safety. The achieved **13x speedup** enables high-frequency replanning (approx. 2Hz) on low-power hardware, paving the way for truly autonomous GNC systems in cluttered orbital environments.

## 6. References

1. Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics*.
2. Karaman, S., & Frazzoli, E. (2011). "Sampling-based algorithms for optimal motion planning". *The International Journal of Robotics Research*.
3. Qureshi, A. H., et al. (2019). "Motion Planning Networks". *International Conference on Robotics and Automation (ICRA)*.
4. Rubí, B., et al. (2020). "Deep Learning-based Path Planning for Multirotor Aerial Vehicles in Unknown Environments". *International Conference on Unmanned Aircraft Systems (ICUAS)*.
5. Khatib, O. (1986). "Real-time obstacle avoidance for manipulators and mobile robots". *The International Journal of Robotics Research* (APF Referansı).

# Uzay Aracı Otonomisi İçin Hibrit Mantık-Öğrenme Mimarisi ile Güvenli ve Gerçek Zamanlı 3D Yörünge Üretimi

## Uluslararası Astronotik Kongresi (IAC) 2026 Raporu

### 1. Özet

Yörüngede bakım onarım hizmetlerinden otonom kenetlenmeye kadar uzanan uzay operasyonlarının karmaşıklığı arttıkça, enerji kısıtlı donanımlar üzerinde gerçek zamanlı ve güvenli yörünge planlama ihtiyacı kritik hale gelmektedir. A\* gibi geleneksel arama tabanlı algoritmalar matematiksel olarak en iyi (optimal) sonucu garanti etse de, karmaşık 3D engellerin bulunduğu ortamlarda hesaplama maliyetleri gömülü sistemler (örn. NVIDIA Jetson serisi) için çok yüksektir. Bu çalışma, Derin Öğrenme'nin hızı ile Klasik Kontrol Teorisinin güvenlik garantilerini birleştiren **Hibrit Mantık-Öğrenme Mimarisi** sunmaktadır. Hacimsel ortam verisini sıkıştırılmış bir uzamsal temsile dönüştüren özelleştirilmiş bir 3D Evrişimli Sinir Ağı (3D-CNN) ve hedefe yönelik yörünge tahminleyen bir MLP yapısı geliştirilmiştir. Mutlak güvenliği sağlamak adına, ağın ürettiği ham tahminler, Yapay Potansiyel Alanlar (APF) kullanan fizik tabanlı bir onarım katmanından geçirilir; bu katman çarpışma risklerini milisaniyeler içinde düzeltir. Deneysel sonuçlar, hibrit sistemimizin, pratikte sınırsız süre tanınan A\* taban hattına

kıyasla **13.0 kat hız artışı** sağladığını ve karmaşık çarpışma senaryolarında **%100 başarı oranını** koruduğunu göstermektedir. Bu çalışma, sınırlı işlem gücüne sahip yeni nesil uzay araçlarında otonom seyrüsefer yetenekleri için uygulanabilir bir çözüm sunmaktadır.

## 2. Giriş

Uzay aracı güdüm, seyrüsefer ve kontrol (GNC) paradigması, yer tabanlı tele-operasyondan tam araç içi otonomiye doğru kaymaktadır. Bu geçişteki temel zorluklardan biri **Hareket Planlama** (Motion Planning) sorunudur: Engellerle (drone'lar, uzay çöpleri veya yapısal bileşenler) dolu 3D bir uzayda A noktasından B noktasına çarpışmasız bir yol bulmak.

Örnekleme tabanlı (RRT\*) ve arama tabanlı (A\*) planlayıcılar standart çözümler olsa da, çalışma sürelerindeki belirsizlik (variance) büyük bir darboğazdır. Yoğun bir 3D ızgarada ( $1000m^3$ ) yol bulmak, engel konfigürasyonuna bağlı olarak milisaniyelerden saniyelere kadar değişebilir; bu da kritik manevralar sırasında aracın tepki verememesine ("frozen robot" sendromu) yol açar.

Biz bu sorunu bir arama problemi olarak değil, bir **örüntü tanıma** (pattern recognition) problemi olarak ele alıyoruz. Optimal bir çözücü (A\*) taklit edecek şekilde eğitilen bir Derin Sinir Ağı ile arama sürecini tek ve hızlı bir ileri beslemeli (forward pass) işleme indirgiyoruz. Ancak sinir ağları stokastik (olasılıksal) yapıdadır ve %100 güvenlik garanti edemezler. Bu boşluğu deterministik bir "Mantık" (Fiziksel Onarım) katmanı ile doldurarak hem hızlı (Öğrenme) hem de güvenli (Mantık) bir sistem oluşturuyoruz.

## 3. Metodoloji

### 3.1. 3D Operasyonel Ortam

Simülasyon ortamımız  $100 \times 100 \times 100$  metrelik kübik bir çalışma alanını temsil eder. Gömülü sistemlerin bellek kısıtlarına uygunluk açısından:

- **Analitik Depolama:** Engeller analitik küreler  $(c_x, c_y, c_z, r)$  olarak saklanır, böylece voksel hataları olmadan kesin çarpışma kontrolü yapılır.
- **Algısal Girdi:** Sinir ağı için ortam, **Max-Pooling** stratejisi kullanılarak  $100 \times 100 \times 100$  boyutunda ikili (binary) bir doluluk ızgarasına indirgenir.

### 3.2. Mimari: TrajectoryNet3D

Çıkarım (inference) hızı için optimize edilmiş hafif bir mimari tasarlanmıştır:

1. **Görsel Kodlayıcı (Encoder):** 4 katmanlı 3D-CNN, doluluk ızgarasından uzamsal öznitelikleri çıkarır.

2. **Öznitelik Enjeksiyonu:** Düzleştirilmiş görsel öznitelikler, normalize edilmiş Başlangıç ve Hedef koordinat vektörleri ile birleştirilir.
3. **Çözücü (Decoder):** Bir MLP yapısı, yörüngeyi 20 adet eş aralıklı 3D nokta dizisi olarak tahmin eder.

### 3.3. Hibrit Mantık-Öğrenme Hattı

Ağın ham çıktıları global yolu doğru tahmin etse de bazen engellere çok yaklaşabilir. Öğrenme içermeyen (non-learning) bir son işleme hattı uygulanır:

- **Fizik Tabanlı Onarım: Yapay Potansiyel Alanlar (APF)** kullanılır. Eğer bir nokta  $w_i$  bir engelin güvenlik yarıçapını ihlal ederse, itici bir kuvvet vektörü  $\vec{F}_{rep}$  bu noktayı en yakın güvenli yüzeye iter. Bu işlem iteratiftir ancak çok hızlı yakınsar ( $<1ms$ ).
- **Kinodinamik Yumuşatma:** Onarılan yola B-Spline enterpolasyonu uygulanarak hız ve ivme profillerinin pürüzsüz (smooth) olması sağlanır.

## 4. Deneyler ve Sonuçlar

Sistem, uçuş donanımı sınırlamalarını simüle etmek amacıyla CPU tabanlı bir kurulumda klasik A\* çözücüsü ile kıyaslanmıştır. A\* çözücüsüne "Yer Gerçeği"ni (Ground Truth) bulabilmesi için pratikte sınırsız zaman (2 milyon adım) tanınmıştır.

Tablo 1: Karşılaştırmalı Performans Analizi

| Metrik          | Klasik A* (Baseline) | Hibrit Model (Bizimki) | İyileştirme         |
|-----------------|----------------------|------------------------|---------------------|
| Başarı Oranı    | %100                 | %100                   | Eşit Güvenilirlik   |
| Ortalama Süre   | 5565.7 ms            | <b>428.7 ms</b>        | 13.0 Kat Daha Hızlı |
| Maksimum Ram    | ~850 MB              | <b>~185 MB</b>         | 4.5x Verimli        |
| En Kötü Senaryo | > 20,000 ms          | <b>~600 ms</b>         | Deterministik Süre  |

En çarpıcı bulgu, hibrit modelin **deterministik gecikme süresidir**. A\* süresi ortam karmaşıklığıyla üssel olarak artarken, modelin çıkarım süresi sabittir ( $O(1)$ ) ve sadece matris çarpım derinliğine bağlıdır.

## 5. Sonuç

Bu çalışma; deterministik fizik tabanlı mantık katmanıyla "korunan" Derin Öğrenme yöntemlerinin, güvenlikten ödün vermeden geleneksel yöntemlerden çok daha hızlı çözüm üretebildiğini kanıtlamaktadır. Elde edilen **13 kat hız artışı**, düşük güçlü donanımlarda yüksek frekanslı (yaklaşık 2Hz) yeniden planlamaya olanak tanıyarak, karmaşık yörünge ortamlarında tam otonom sistemlerin önünü açmaktadır.