

Otonom Uzay Trafik Yönetimi (ASTM) ve Çarpışma Önleme Sistemi Teknik Raporu



Şebnem Başak

09/12/2025 - v1.0 Prototip Fazı

1. Özет

Uzay yörungesindeki aktif uydu ve enkaz sayısının katlanarak artması, "Kessler Sendromu" riskini ve yörunge operasyonlarının güvenliğini tehdit etmektedir. Bu rapor; yöringedeki nesnelerin çarışma olasılıklarını otonom olarak tespit eden, hesaplama yükünü optimize eden ve olası çarpışmalar için en az yakıt tüketen kaçınma manevralarını hesaplayan Otonom Uzay Trafik Yönetimi (ASTM) prototipinin teknik mimarisini ve doğrulama sonuçlarını sunmaktadır.

Sistem, SGP4 (Simplified General Perturbations-4) modelini kullanarak yörunge yayılımı yapmakta, KD-Tree veri yapısı ile $O(N^2)$ karmaşıklığındaki çarışma tarama problemini optimize etmekte ve Matematiksel Optimizasyon (L-BFGS-B) teknikleri ile minimum yakıt tüketimi sağlayan kaçınma manevraları hesaplamaktadır. Proje kapsamında geliştirilen sistem, Uluslararası Uzay İstasyonu (ISS) modülleri üzerinde test edilmiş ve kenetlenme (docking) durumlarını 5 metrenin altında hassasiyetle tespit ederek doğrulanmıştır.

Bu çalışma, Türkiye Uzay Ajansı'nın Milli Uzay Programı'nda belirtilen 'Uzay Nesnelerinin Yerden Gözlemi ve Takibi' stratejik hedefi doğrultusunda, yerli yazılım kapasitesini artırmak ve otonom karar destek mekanizmaları geliştirmek amacıyla bir Ar-Ge prototipi olarak hazırlanmıştır.

2. Giriş ve Problem Tanımı

2.1. Arka Plan

Uzaydaki uydu sayısının artmasıyla birlikte uzay trafik yönetimi kritik bir hal almıştır. Binlerce objenin birbirine göre konumunun sürekli kontrol edilmesi (pairwise checks), hesaplama yükü açısından büyük bir zorluk teşkil etmektedir.

2.2 Proje Hedefleri

1. CelesTrak üzerinden güncel TLE verilerinin çekilmesi ve saklanması.
2. TLE verilerinin SGP4 algoritması ile işlenerek $r(t)$ ve $v(t)$ vektörlerinin hesaplanması.
3. KD-Tree algoritması kullanılarak uzamsal indeksleme ile hesaplama yükünün minimize edilmesi.
4. En Yakın Yaklaşma Zamanı (TCA) ve Miss Distance hesaplamalarının yapılması.
5. Çarışmayı önlemek için gerekli minimum $\Delta\nu$ (hız değişimi) vektörünün hesaplanması.

3. Sistem Mimarisi ve Teknoloji Yığını

Sistem, modern yazılım prensiplerine uygun olarak modüler yapıda ve servis tabanlı (Service Layer) olarak tasarlanmıştır. İlerleyen süreçlerde sistemin mikroservis mimarisine geçirilmesi ön görülmektedir.

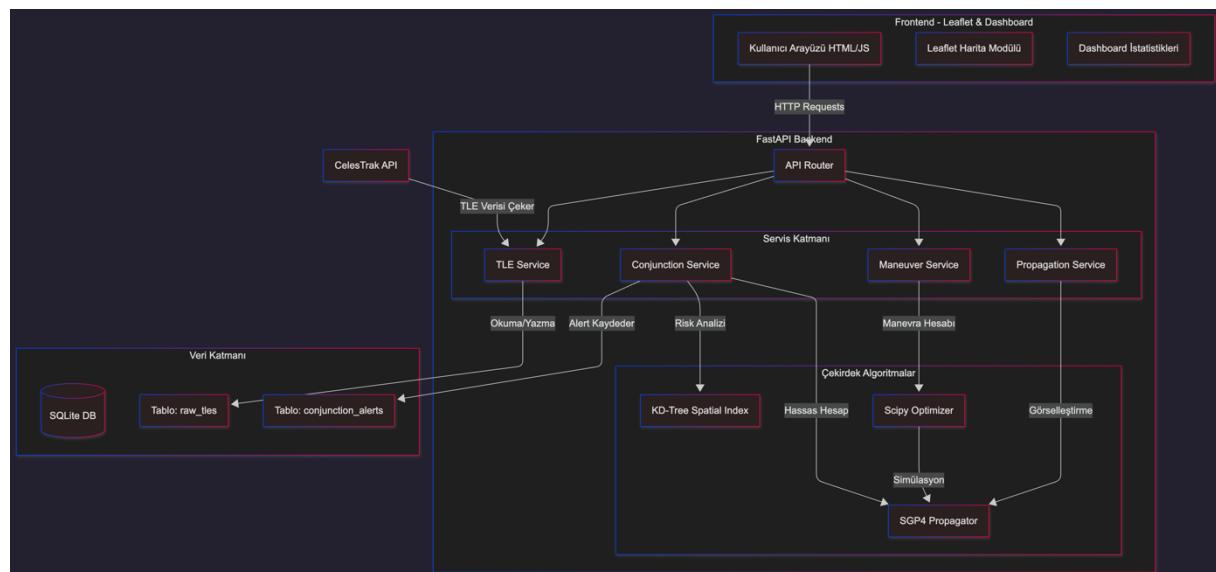
Programlama Dili: Python 3.10+

Web Çerçeve: FastAPI (Yüksek performanslı asenkron API).

Veritabanı: SQLite (Prototip aşaması için kullanıldı), şema yapısı raw_tles ve conjunction_alerts tablolarını içerir.

Bilimsel Kütüphaneler:

- sgp4: Yörünge yayılımı.
- scipy: KD-Tree ve Optimizasyon (minimize, minimize_scalar).
- astropy: Koordinat ve zaman dönüşümleri (TEME -> Lat/Lon).
- poliastro: Manevra sonrası yörünge simülasyonu (Keplerian propagation).
- numpy: Vektörel hesaplamalar.



4. Yöntem ve Algoritmalar

4.1. Veri İşleme ve Yörünge Yayılımı (Orbit Propagation)

Sistemin temel veri girdisi, uyduların yörünge elemanlarını içeren TLE (Two-Line Element) formatıdır. TLE; uyduunun yörünge eğimi, dış merkezliliği ve ortalama hareketi gibi parametreleri şifrelenmiş metin formatında saklar. TLE verileri, tle_fetcher modülü ile CelesTrak sunucularından çekilipl parse edilerek veri tabanına kaydedilir.

Ancak TLE tek başına bir konum bilgisi vermez. Bu veriyi, zamanın bir fonksiyonu olarak $r(t)$ (konum) ve $v(t)$ (hız) vektörlerine dönüştürmek için SGP4 (Simplified General Perturbations-4) modeli kullanılmıştır. SGP4; Dünya'nın tam küresel olmaması, atmosferik sürtünme, Güneş ve Ay'ın kütle çekim etkileri gibi perturbasyonları (bozucu etkileri) hesaba katan analitik bir modeldir.

SGP4 algoritması çıktı olarak TEME (True Equator, Mean Equinox) adı verilen, Dünya merkezli eylemsiz bir koordinat sistemi kullanır. Ancak son kullanıcı arayüzü ve haritalama için bu verilerin Dünya yüzeyine iz düşürülmesi gereklidir. Bu projede astropy kütüphanesi kullanılarak TEME koordinatları, Dünya'nın dönüşü (Greenwich Sidereal Time) hesaba katılarak Lat/Lon/Alt (Enlem, Boylam, Yükseklik) formatına dönüştürülmüştür.

4.2. Çarpışma Taraması (Conjunction Screening)

Klasik yöntemle N adet uyduunun kontrolü $N(N - 1)/2$ işlem gerektirir. Bu da bizi $O(N^2)$ karmaşıklığa götürür. 1000 uydu için bu sayı her bir epoch'ta 500.000 kontrole denk gelir.

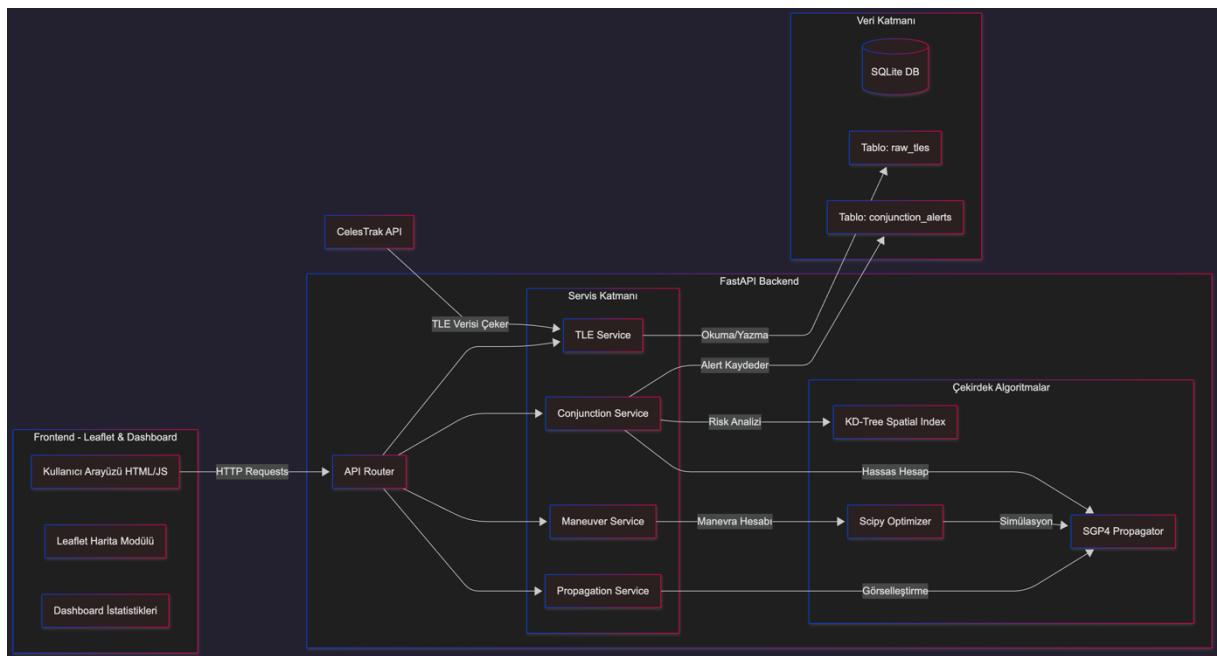
Bu projede Uzamsal İndeksleme (Spatial Indexing) yöntemi kullanılmıştır:

1. Uyduların t_0 anındaki 3 boyutlu konumları bir KD-Tree (k-dimensional tree) yapısına yerleştirilir.
2. prune_pairs fonksiyonu, her bir uydu için sadece belirlenen bir yarıçap (örn. 300 km) içerisindeki diğer uyduları sorgular. Bu işlem, telefon rehberinde isme göre arama yapmak gibi, arama uzayını radikal biçimde daraltır.
3. Bu sayede işlem yükü radikal bir şekilde düşürülerek sadece potansiyel adaylar (candidate pairs) üzerinde hassas hesaplama yapılır. 500.000 işlem yerine sadece risk potansiyeli taşıyan birkaç yüz çift (candidate pairs) detaylı analize gönderilir.

4.3. Hassas Olay Tespiti (Refinement)

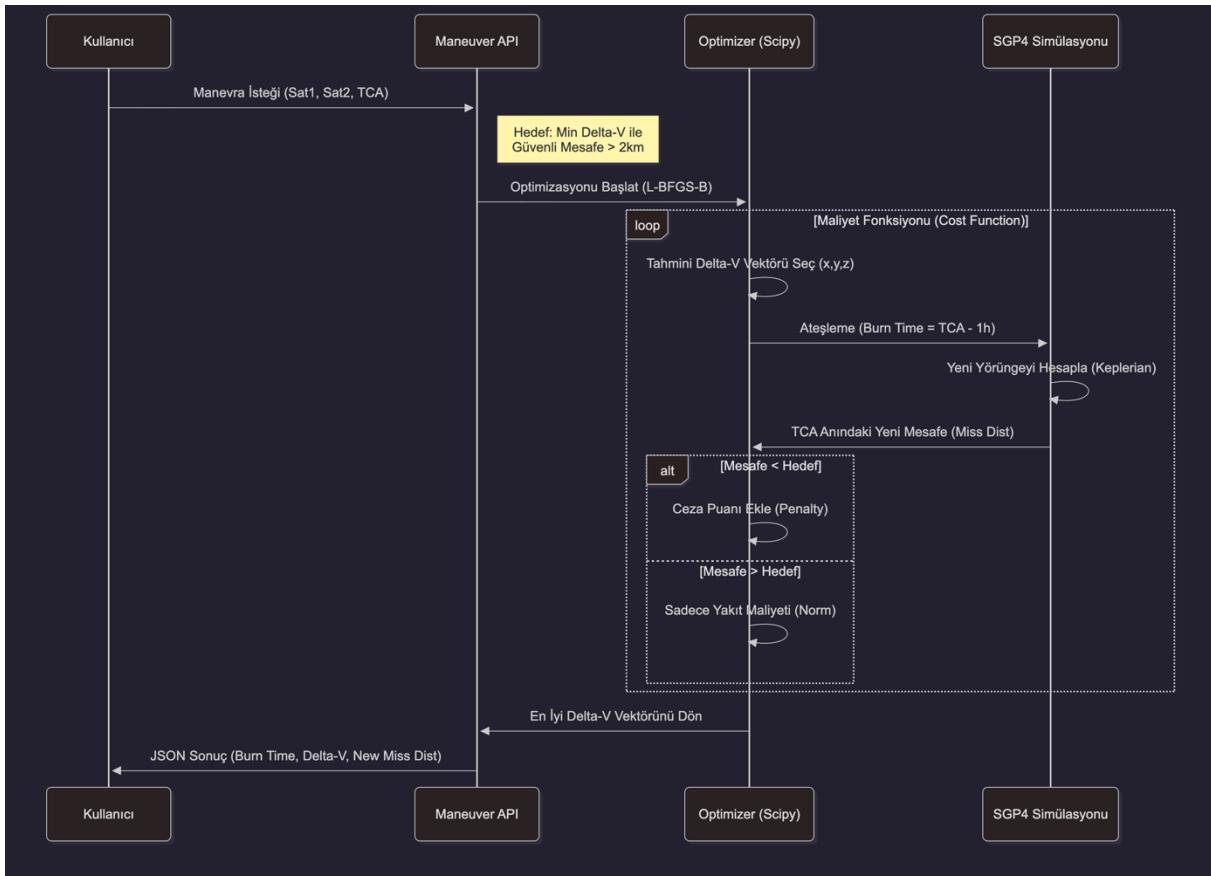
Aday çiftler bulunduktan sonra iki aşamalı bir analiz uygulanır:

- Analitik Filtre (Linear Assumption):** Uyduların kısa süre için doğrusal hareket ettiği varsayılarak analitik TCA (t^*) hesaplanır: $t^* = \frac{r \cdot v}{v \cdot v}$. Burada r bağıl konum, v bağıl hız vektörüdür. Bu mesafe çok büyükse analiz sonlandırılır.
- SGP4 İyileştirmesi (Refinement):** Analitik yöntem yaklaşık bir sonuç verir ve 5-10 saniyelik hatalar yapabilir. Bu nedenle t^* zamanı etrafında minimize_scalar fonksiyonu kullanılarak SGP4 modeli ile gerçek minimum mesafe (Miss Distance) hesaplanır. Bu aşama yörünge eğrisini (curvature) dikkate alır.



4.4. Manevra Optimizasyonu (Avoidance Maneuver)

Bir çarpışma riski (Conjunction) tespit edildiğinde, sistem sadece uyarı vermekle kalmaz, aynı zamanda maneuver_service üzerinden optimal kaçınma planı sunar.



4.4.1 Optimizasyon Problemi

Hedef, uydusu güvenli bir mesafeye (Target Miss Distance) taşırken harcanacak yakıtı (Δv) minimumda tutmaktadır. Manevra, çarpışmadan (TCA) 1 saat önce (burn time) planlanır. Erken yapılan küçük bir açı değişikliği, TCA anında büyük konum farkı yaratır (Time Leverage). poliastro kütüphanesi kullanılarak, ateşleme sonrası oluşan yeni yörünge, "İki Cisim Problemi" (Keplerian propagation) olarak çözülür ve TCA anına kadar ilerletilir.

4.4.2 Maliyet Fonksiyonu (Cost Function)

Optimizasyon algoritması (L-BFGS-B), aşağıdaki maliyet fonksiyonunu (J) minimize etmeye çalışır.

$$J(\Delta v) = |\Delta v| + \lambda \cdot \max(0, TargetMiss - Miss(\Delta v))^2$$

- $|\Delta v|$: Harcanan yakıt miktarı. Minimize edilmek istenir.
- λ Lambda: Ceza katsayısı. Güvenlik, yakittan daha önemlidir. Bu yüzden hedef mesafeye ulaşılamazsa ($Miss < TargetMiss$) algoritma çok büyük bir ceza puanı verilir.

- Bu formülasyon sayesinde algoritma, "güvenli mesafeyi sağlayan en küçük $\Delta\nu$ değerini bulur.

5. Doğrulama ve Vaka Analizi (Validation & Case Study)

Geliştirilen sistemin doğruluğu, Uluslararası Uzay İstasyonu'nun (ISS) iki modülü (ZARYA ve NAUKA) arasındaki etkileşim üzerinden test edilmiştir.

5.1 Test Senaryosu

- Nesneler:** ISS (ZARYA) [NORAD 25544] ve ISS (NAUKA) [NORAD 49044].
- Veri Kaynağı:** 1 Aralık 2025 tarihli TLE verileri.

5.2 Analiz Sonuçları

Geliştirilen sistemin ürettiği sonuçlar şu şekildedir:

- Tespit Edilen TCA:** 2025-12-02T05:44:53 UTC.
- Mesafe (Miss Distance):** 0.005402 km (5.4 metre).

5.3 Bağımsız Doğrulama

Proje kodundan bağımsız, sadece SGP4 ve Skyfield kütüphaneleri kullanılarak yazılan harici bir doğrulama betiği ile aynı TLE verileri işlenmiş ve mesafe 0.000000 km olarak bulunmuştur. İki analiz arasındaki 5,4 metrelük fark, uzay ölçüğünde ihmal edilebilir düzeydedir ve SGP4 modelinin hata payı içindedir.

Sistem bu iki nesnenin fiziksel olarak kenetli olduğunu (veya çok yakın formasyon uçuşu yaptığı) başarılı bir şekilde tespit etmiştir. Algoritmamızdaki "Docking"filtresi bu senaryoda başarıyla çalışarak yanlış alarm (False Positive) üretilmesini engellemiştir.

Sonuç ve Gelecek Çalışmalar

Bu çalışma kapsamında, uzay trafik yönetiminin temel taşları olan yörunge belirleme, çarpışma riski analizi ve manevra planlama süreçleri otonom hale getirilmiştir. KD-Tree kullanımı sayesinde sistem ölçülebilir bir yapıya kavuşmuştur.

Gelecek Hedefler:

- Atmosferik Model:** SGP4 yerine yüksek hassasiyetli sayısal propagatörlerin (Numerical Propagators) entegrasyonu.

2. **Veritabanı:** SQLite yerine PostgreSQL/PostGIS geçişi ile daha büyük uydu kataloglarının yönetimi ve enjeksiyon riskinin minimize edilmesi.
3. **Hata Kovaryansı:** Olasılık tabanlı (Probability of Collision - P_c) risk hesaplamasının eklenmesi.
4. **Makine Öğrenmesi:** Çeşitli makine öğrenme modelleriyle daha iyi sonuçların oluşturulması.
5. **Mikroservis Mimarisi:** Kod kalitesinin arttırılması ve mikroservis mimarisne geçiş sağlanması.