

Yoda Otonom Yazılım

Takım Adı: İtü Güneş Arabası Ekibi

Otonom sürüş algoritmaları

Bu raporda kontrol algoritmaları (PID, MPC, Pure Pursuit, Stanley vb.) çok detaylı geçilmiyor. “Planlayıcı & Kontrolcü” alt başlığı var ama algoritmanın hangi tip olduğu spesifik olarak raporda görünmüyor

Pure Pursuit: basit, geometrik tabanlı bir yol takip algoritmasıdır; sapma açısı + ileri bakma uzaklığı ile çalışır. Hızlı, kolay uygulanabilir

PID: klasik kontrol; hız / direksiyon kontrolü gibi düşük düzey kontrol görevlerinde yaygın. Avantajı basit olması, dezavantajı ayarlarının hassas olması ve kısıtlı dinamik davranışlarda yetersiz kalmasıdır

Stanley: lateral hata + yönelim hatasını minimize etmeye çalışır (örneğin otonom araçlarda popüler

Uzun düzgün yolda iz takibi için Pure Pursuit ya da Stanley yeterli olabilir.

Duraklama / kalkış senaryosunda hız kontrolü için PID + dur-kalk mantığı yaygın

Algılama (Nesne Tespiti) algoritmaları:

Levha tespiti için **YOLOv5** modeli kullanılmıştır. Kamera görüntüsünü alıp YOLO’ya verirler; tespit edilen levhaların koordinatları alınır

Tespit edilen levhanın derinliği (mesafe) bilgisi, kamera ile elde edilen point cloud verisinden çıkarılıyor — yani LiDAR ya da stereo kamera tarafından sağlanan nokta bulutu verisinden derinlik çıkarımıyla.

Kamera node’u, ZED Stereo Kameranın Python API’si kullanılarak ROS2 içinde çalıştırılıyor.

Ayrıca, kamera haritalama denemesi olarak RTAB-MAP denenmiş ama hatası yüksek olduğu için vazgeçilmiş.

Kullanılan modelin neden tercih edildiğini (örneğin hız, donanım uyumu, doğruluk oranı vb.) belirtiniz.

YOLO tipi modeller genelde hızlıdır, gerçek zamanlı çalışmaya uygundur.

Nesne tespiti konusunda iyi doğruluk / hız dengesi sunar.

Yapay zekâ yaklaşımları:

Bu raporda doğrudan reinforcement learning (RL) temelli bir yaklaşımdan söz edilmiyor.

Bazı takımlar RL yaklaşımı kullanabiliyor . Bu raporda bu tür bir uygulama yok, belki zaman ya da kaynak kısıtlılığı sebebiyle.

Sensörler:

ZED 2 Stereo Kamera: Levha tespiti ve levhaların uzaklıklarının ölçülmesi için. Kamera görüntüsü YOLO'ya giriyor; levhanın koordinatları alınıp point cloud verisi üzerinden derinlik çıkarılıyor.

Velodyne Puck Lite (16 kanallı 3D LiDAR): Haritalama (NDT eşleştirme) ve lokalizasyonda kullanılıyor.

Simülasyon ortamı:

Bu raporda simülasyon ortamı detayları fazla yer almamış ya da raporun bu kısımları daha düşük seviyede yazılmış olabilir

TAKIM ADI: EVA Otonom

Kullanılan algoritmalar

- Raporun “Yazılım Mimarisi” kısmında takımın **rotayı takip etme, adalı/adasız kavşak dönüşleri, park etme** gibi görevler için algoritmalar geliştirdiği belirtiliyor.
- Örneğin “Sürüş Algoritmaları Ekibi; otonom sürüş için gerekli rotayı takip etme ... kavşak dönüşleri ve park etme algoritmalarını oluşturmakla görevlidir.”
- Ancak raporda spesifik olarak “Pure Pursuit”, “Stanley”, “PID”, “MPC” gibi klasik kontrol algoritmalarından hangilerinin kullanıldığı açıkça listelenmemiş görünüyor

Algılama (Nesne Tespiti) algoritmaları:

Raporun ön tasarım raporu değerlendirme kısmında, “2D LiDAR ile engel tespiti yönteminin geliştirilmesine karar verilmiştir.”

Park etme senaryosunda yavaş hız-kontrol algoritmaları (örneğin PID) tercih edilmiş olabilir — ancak doğrulanmamış.

İşlemci olarak güçlü bir kit (NVIDIA Jetson AGX Xavier) kullanılmış; bu, daha karmaşık algoritmalar (örneğin MPC) için uygun altyapı olabilir.

Rapor detaylarında hangi algoritmanın hangi görevde kullanıldığı net değil; rapora bu boşlukların doldurulması veya takım ile görüşülmesi önerilebilir

Park etme senaryosunda yavaş hız-kontrol algoritmaları (örneğin PID) tercih edilmiş olabilir — ancak doğrulanmamış.

Hangi model ya da kütüphanenin (örneğin YOLO, OpenCV, TensorRT) kullanıldığı açık şekilde belirtilmemiş.

Yapay Zekâ Yaklaşımları

Raporda açıkça “Reinforcement Learning” (Pekiştirmeli Öğrenme) ya da “Machine Learning” modelleri ayrı ayrı belirtilmemiş görünüyor

Eğer takım ileri düzeydeyse, yol tahmini, nesne sınıflandırma gibi alanlarda makine öğrenmesi modelleri (örneğin CNN, SVM) kullanmış olabilir.

Sensörler

Kamera: ZED 2 Stereo Kamera.

LiDAR: RPLiDAR S1 2D LiDAR.

Enkoder: Direksiyon sistemi açısını ölçmek için EP50S8 mutlak enkoder.

Elektronik mimaride farklı voltaj regülatörleri, motor sürücüler, step motor gibi bileşenler yer almakta

GPS ya da IMU sensörlerinden bahsedilmiyor görünmekte — bu alanda bir eksik olabilir ya da kullanılmıyor olabilir

Simülasyon Ortamı

Raporun başında “Simülasyon Ekibi; sensörler, görüntü işleme, araç kontrol sistemi ve karar algoritmalarının test edilebileceği gerçekçi bir simülasyon ortamı hazırlamakla görevlidir.

Ancak raporda hangi simülasyon platformu (Gazebo, Unity, özel geliştirilmiş vs.) kullanıldığı açıkça belirtilmemiş.

Çıkarımlar:

- Simülasyon ortamı varsa, algoritmaların sahada karşılaşabileceği değişkenleri önceden test etme avantajı var.
- Dezavantaj: Simülasyon gerçek dünyada karşılaşılabilecek tüm belirsizlikleri tam yansıtmayabilir; gerçek araç testlerine geçilmesi şart.

Bu takım raporunda güçlü yönler:

- Donanım mimarisi açıkça belirtilmiş; güçlü bir ana bilgisayar (Jetson AGX Xavier) seçilmiş.
- Sensör seçimleri (stereo kamera, 2D LiDAR) ve kontrol sistem mimarisi ayrıntılı.

Geliştirilebilecek yönler / dikkat edilmesi gerekenler:

- Algoritma kısmında hangi spesifik kontrol algoritmalarının kullanıldığı, performans karşılaştırmaları, seçilme nedenleri daha açık olabilirdi.

TAKIM ADI: MİLAT

Otonom Sürüş Algoritmaları

Kullanılan Algoritmalar:

- Araç, otonom sürüşünü gerçekleştirmek için **Pure Pursuit** ve **PID kontrol** algoritmalarını kullanıyor.
- **Karar mekanizması**, ROS (Robot Operating System) üzerinden yapılandırılmış.
- Park etme, kavşaktan dönme ve durma gibi görevler için özel alt algoritmalar geliştirilmiş.

Senaryolar ve Uygulama:

- **Pure Pursuit:** Özellikle şerit takibi ve rota üzerindeki doğrusal hareketlerde tercih edilmiş.

- **PID:** Hız ve yön kontrolü gibi sürekli hata düzeltme gerektiren durumlarda kullanılmış.

Algılama (Nesne Tespiti) Algoritmaları

Kullanılan Yöntemler:

- **YOLOv4 mimarisi** ile nesne tanıma yapılıyor (tabela, yayalar, araçlar vs).
- Görüntü işleme için **OpenCV** kütüphanesi de kullanılmış.
- • **YOLOv4** hızlı ve gömülü sistemlerde çalışmaya uygun.
- • Takım, gerçek zamanlılık ve donanım uyumluluğu açısından YOLO'yu seçmiş.
- • **OpenCV**, temel işaret tespiti ve şekil analizi gibi işlemler için kullanılmış (örneğin şerit takibi).

Yapay Zeka Yaklaşımları

Kullanılan Yöntemler:

- Pekiştirmeli öğrenme doğrudan kullanılmasa da, **YOLO gibi derin öğrenme tabanlı modeller** uygulanmış.
- Ayrıca görüntü sınıflandırma ve engel türü belirlemede **CNN mimarisi** temelli çözümler geliştirilmiş.

Sensörler

Kullanılan Sensörler:

- **Kamera:** Görüntü alma, şerit ve nesne tespiti.
- **LiDAR:** 3D çevre haritalama ve engel tespiti.
- **IMU:** Yalpalama, hızlanma, yön bilgisi.
- **GPS:** Global konumlama.
- **Encoder:** Tekerlek hızı ölçümü.

Entegrasyon:

- Tüm sensör verileri ROS çatısı altında zaman senkronizasyonu ile birleştirilmiş.

- Bu sayede karar mekanizması gerçek zamanlı ve senkron veri ile çalışıyor.

Simülasyon Ortamı

Kullanılan Ortamlar:

- **Gazebo Classic** simülatörü kullanılıyor.
- Gerçek fizik motoru ve sensör taklitleri sayesinde test süreçleri modelleniyor.

Amaçlar:

- Algoritma testleri, görev senaryosu simülasyonları ve yapay zeka modeli denemeleri için kullanılmış.
- Kamera ve LiDAR gibi sensörler simülasyon ortamında fiziksel benzerlikte modellenmiş.