Otonom Araçların Bileşenleri ve Yarışma Raporları İncelemesi

1. **Robotaksi Binek Otonom Araç Yarışması Teknik Şartname İncelemesi**

Yarışma genel anlamda ülkedeki otonom araç teknolojilerini geliştirmeyi ve tasarlanan kontrol sistemlerini mümkün olan en üst seviyeye taşımayı amaçlamaktadır. Otonom araçların tanımlanan senaryolarda insana benzer davranışlar göstermesine önem verilir. Yarışma 3 farklı günde gerçekleştirilen 3 turdan oluşur, tamamlanacak görevlerle ilgili bilgiler JSON dosyaları ile yarışmacılara aktarılır. Görevler süresince araçların karşısına statik veya dinamik olacak şekilde engeller çıkar, aracın bu engellere karşı doğru tepkiyi vermesi beklenir. Sürüş için gerekli trafik levhaları alana yerleştirilmiştir.

Değerlendirme sürecinde takımlardan öncelikle Teknik Yeterlilik Formu ve Kritik Tasarım Raporu, sonrasında ise Araç Test Videosu, Simülasyon Raporu ve Sunum talep edilmektedir. Kritik Tasarım Raporu aşamasında raporlar şablona uygun olarak oluşturulmalıdır; raporunun uygun olmadığı tespit edilen takımlar yarışmadan elenir, kalanlardan yarışma denetim kurulunca belirlenen bazı takımlara maddi destek sağlanır bazıları ise maddi destek olmadan ilerler. Bir sonraki aşamada takımlardan istenen Araç Test Videosu, Simülasyon Raporu ve Sunum'un KTR ile uyum içinde olması oldukça önemlidir.

Araç Test Videosunda araçların güvenilir bir şekilde hareket ettiği, otonom olarak istenilen yönde ilerleyebildiği teyit edilir. Video, aracı hem dışarıdan gören hem de simülasyon koltuğu ve direksiyonu kaydeden iki farklı açıdan aynı anda çekilen görüntüler barındırmalıdır. Sunumların paylaşılan şablona ve daha önce oluşturulan belgelere bağlı kalarak oluşturulması beklenir. Grafik, parça fotoğrafı gibi görsellere sunumda yer verilmelidir, isteğe bağlı olarak sunum videosu da eklenebilir. Simülasyon Raporunda otonom araçların yarışma gerekliliklerini sağladıkları ve görevleri başarılı bir şekilde gerçekleştirme yetkinliklerinin bulunduğu takımlar tarafınca doğrulanmış olur. Simülasyon ortamı Unity, Gazebo, Unreal Engine gibi bir benzetim programı ile oluşturulur. Simülasyonda geliştirilen yazılımların gerçek araç üzerinde uygulanabilir olması istenmektedir. Simülasyon ortamı tasarlanırken algılama, planlama, navigasyon gibi temel otonom sistem algoritmalarının test edilebilir olması göz önünde bulundurulmalıdır. Elde edilen veriler ve araç performansına dair bir video gerçekleştirilen görevleri içeren bir sunum ile birlikte hakem heyetine teslim edilmelidir. Simülasyon sürecinde yazılım teknolojisi, kontrol, konumlandırma, karar verme, drive-by-wire sistemi ve gerçek araca uyarlanabilirlik gibi kriterlere dikkat ederek değerlendirme yapılır.

Final aşamasına geçen ekiplerin verilen tüm görevleri tamamlayacak şekilde hazırlanması önerilir. Bölgede ölçümü kolaylaştırmak için işaretleyiciler kullanılır. Yarışma boyunca ışık ve trafik kurallarına uyum, park görevinin doğru bir şekilde tamamlanması, dinamik ve statik engellerden sakınabilmek, görev noktalarına ulaşabilmek gibi kategoriler aracılığıyla puanlama yapılır.

Yarış pistinde sadece belirtilen zaman içinde veya ön izinle test sürüşü yapılmasına izin verilir. Bir ekibin başka bir ekibe yazılımsal veya donanımsal herhangi bir etkide bulunduğu tespit edilirse bahsi geçen ekip doğrudan yarışmadan diskalifiye edilir. Yarışmacıların kendilerine tanımlanan süre içerisinde aracı yeniden başlatma hakkı bulunmaktadır fakat bu eylem ciddi bir puan kaybına sebep olabilir.

2. KOÜ-MEKATRONOM Takımı Kritik Tasarım Raporu İncelemesi

2.1. Ön Değerlendirme Raporu Hakkında

KOÜ-MEKATRONOM ekibi araçlarını tasarlarken otonom, elektronik ve mekanik olmak üzere 3 gruba ayrılmıştır. İncelenen rapora göre önceki yıllarda katıldıkları yarışmalar sonucu yaptıkları değerlendirme ile simülasyon alanında yeniliklere gitme kararı vermişlerdir. Öncesinde Tensorflow ile çalışırken daha iyi bir verim almak üzere Pytorch üzerinde çalışan YOLOv5x'i denerler. Bu programı seçme sebepleri; öğrenme hızının daha gelişmiş olması ve ayrılan video bellek miktarının daha yüksek olmasıdır. Geçen senelerde yaşadıkları simülasyon sunum ve videosundaki düşük FPS problemini bu şekilde çözmeyi hedeflemişlerdir.

2.2. Yazılım Mimarisi

2.2.1.Şerit Takibi: Ekip şerit takibini gerçekleştirmek için OpenCV yardımı ile Python üzerinden yazdıkları kodu kullanarak ön çizgilerin takip edilmesi ve bu veri ile aracın dönüş yönünün belirlenmesini sağlamıştır. Kameradan alınan görüntü adım adım olmak üzere HSV'ye dönüştürülür, beyaz hariç renkler resimden çıkarılarak çizgiler belirlenir, görüntü üzerinde belli bir alan seçilir, hough dönüşümü sayesinde alandaki şeritler tespit edilip renklendirilir, bu şeritlerin eğimleri ve kesişimleri incelenerek aracın nerde dönüş yapacağı hesaplanır, sayısal sonuçlara göre araç yönlendirilir.

2.2.2.Tabela Tanıma: Görüntüleri ızgara sistemine yerleştirip nesne algılamayı sağlayan YOLO isimli bir algoritma kullanılır, hızı ve doğruluğu nedeniyle bu algoritma seçilmiştir. Farklı modelleri içinden hesaplama ve tahmin açısından en yüksek başarıya sahip YOLOv5x tercih edilmiştir.

2.2.3.Yön Belirleme: Araç tabelalara göre dönüş sağlarken şeritsiz yollarla da karşılaşıldığı için yön tayini yapabilmek için HMC5883L 3 Eksen Pusula Sensörü kullanmaya karar verilmiştir.

2.2.4.Durak Algoritması: Yolcu indirip bindirmek için durak tabelası ile araç arası mesafe ölçülür, öncesinde bu iş için lidar kullanılırken bu sefer kameralarının derinlik özelliği kullanılmıştır. Yarışma öncesinde belirlenen mesafelere göre araç durak cebine girer, buradaki işlemler otomatik gerçekleşir, 30 saniye bekledikten sonra çıkışını yaparak şerit takip algoritmasına geri döner.

2.2.5.Sensörler: Araçta otonom sürüş için gerekli olan kamera, lidar ve ultrasonik sensörler; diğer sistemlere yardım etmesi için de sıcaklık sensörü, akım sensörü ve hall-effect sensörü kullanılır. Kamera olarak ROS ortamına tam uyumlu olan Intel® RealSense™ Depth Camera D435i tercih edilmiştir. Tabela tespitlerinde farklı açılardan yakalanan görüntülerde de başarı sağlayabilmek için kameranın derinlik algılama özelliğinden yararlanılmaktadır. Lidar olarak önceki sene kullanılan RPLİDAR S1 sadece 2 boyutlu tarama yapabildiği için değiştirilmesi düşünülmüş fakat bu sene lidara daha az görev düşeceği için aynısıyla devam edilmiştir. RPLİDAR S1 40 metre ölçüm menzili ve güneş ışığı altında da dışarıda çalışabilmesi sebebiyle de ekibi memnun etmiştir. Lidar aracın ön en uç kısmında konumlandırılır ve lidardan alınan veriler ROS ile ilgili yerlere yönlendirilir. Lidar aracın arkasını göremediği için ileri geri manevralar ve park etme görevinde kolaylık sağlaması için DFROBOT'un URM-37 isimli ultrasonik mesafe sensörü aracın arka kısmına yerleştirilmiştir. Hassas ölçüm yapabilmesi ve 5 metreye kadar keskin veriler elde edebilmesi sensörün tercih edilmesinde etkili olmuştur. Sıcaklık sensörü bataryaların sıcaklığını kontrol eder, One Wire protokolü ile tek hattan 20 pil sıcaklığını okuyabildiği için Texas Instruments firmasının ürünü olan DS18B20 sensörü kullanılır. Akım sensörü araçtaki ani akım artışlarına karşı aracı korumak, hall-effect sensörü ise aracın hızını ölçmek için yerleştirilir. Bu sensörler araç kontrol ünitesi olan Jetson AGX Xavier ile haberleşmektedir. Elektronik direksiyon, gaz pedalı ve fren kontrolü, PWM tabanlı mikrodenetleyici yazılımları kullanılarak sağlanır.

2.2.6.Mekatronom ROS Paketi: Ekip önceki senelerde farklı ROS paketleri kullanıp otonom kontrolcülerini değiştirdiklerinde paketleri tek tek yeniden kurmaları gerektiğini deneyimlemiştir. Bunun üzerine kendi ROS paketlerini oluşturmaya başlamış, paket içine çizgi, görüntü izleme algoritmaları ve araç modelini gerçek araca oldukça benzeten URDF dosyalarını eklemişlerdir.

2.2.7.Takometre: Parkurda rüzgar ve eğimli noktalar gibi dış etkenler dolayısıyla sabit hızını kaybeden araç için bir takometre tasarlanmıştır. Araç jantının karşısına konumlandırılmış hall-effect sensörü ile ölçülen araç hız bilgilerine göre takometre araç hızını sabitler.

2.2.8.Uzaktan Kumanda: Aracın kontrolünü uzaktan da sağlayabilmek ve acil müdahale edilmesi gereken bir durumda kullanılmak üzere bir uzaktan kumanda tasarlanmıştır. Kumandada bulunan algoritma takıma aittir.

3. VOLANS Takımı Kritik Tasarım Raporu İncelemesi

3.1. Ön Değerlendirme Raporu Hakkında

Volans takımının değerlendirme raporuna göre projede yazılımsal eksiklikler olduğu tespit edilmiş ve geliştirilmeye çalışılmıştır. Aynı zamanda raporda bilgilerin daha derinlikli olarak verilmesi gerektiğine karar verilmiştir. Ön tasarım raporunda YOLOv3 kullanmayı düşünürken yeni bir fikir değişikliği ile YOLOv5 kullanılmıştır. Ekip hazır araç kategorisinde yarıştığı ve sunulan araçta Velodyne Puck Hi-Res Lidar bulunduğu için simülasyon çalışma ortamında da Velodyne VLP-16 Puck Hi-Res Lidar kullanılmıştır. Araç kontrol ünitesinde gömülü PID algoritması ile hız kontrolü sağlanmaktadır. PID hata değerlerini hesaplayıp bu ihtimali en aza indirgemeye çalışan bir kontrol geri mekanizmasıdır. Araç aynı zamanda kablosuz kumanda kullanılarak CANBUS yardımı ile kontrol edilebilmektedir.

3.2. Yazılım Mimarisi

3.2.1.Şerit Takibi: Şerit takibi OpenCV kütüphanesinden yararlanarak Zed 2 kamerası aracılığıyla yapılmıştır. Ekibin bu kamerayı seçme sebeplerinin içinde kameranın en başarılı derinlik kameralarından biri olması ve araca 120 derecelik bir görüş açısı sağlaması bulunmaktadır. Bu sayede görüntüler en hızlı şekilde optimum olarak araca iletilir. Ayrıca kameranın çoklu sensörlerle kullanılabilirliği sayesinde de GPS, IMU gibi sensörlerle iş birliği yapılmasını kolaylaştırmıştır. Kameradan elde edilen görüntüler gri maskeleme yaparak sadeleştirilmiş, beyaz maskeleme ile koyu renklerden arındırılmış, Gaussian blur filtresi ile bulanıklaştırılmış, Canny Edge Dedector kullanarak kenarları çizilmiştir. Sonrasında Region of Interest yardımı ile görüntünün yalnızca şeritleri gören alanı seçilmiştir. Kullanılan kodlarla kesikli şeritler birleştirilmiş ve belirlenmiştir.

3.2.2.Nesne Tanımlama: Nesne tanımlama algoritması nesne uzaklık tespiti ile birlikte kullanılarak kamera açısına birden çok tabela girebileceği sebebiyle yakın olan tabelayı algılayacak şekilde tasarlanmıştır. Tespit edilen levha ile araç arası uzaklık derinlik kamerası ile ölçülür ve her levhanın algılanması için gereken uzaklık farklı belirtilmiştir. İstenilen uzaklık sağlanınca levhaların doğruluk değeri okunmaya başlar, bir problem saptanmazsa algoritma çalışmaya başlayacaktır. Tanımlamanın ilk aşamasında kameradan veya internet üzerinden alınan görseller ile veri girişi sağlanmıştır. Tabela ve levha tespiti için yapılan araştırma sonucu YOLOv5 ve Pytourch'un daha uygun olduğu belirlenmiş ve görseller matrislerine ayrılarak YOLOv5 mimarisi ile tabela ve işaretler etiketlenmiş, ayrıca birden çok kez eğitilmiştir. Kameranın yanı sıra LİDAR sensörü de aracın engellerle arasındaki mesafeyi korumasını sağlar, Zed 2 kameranın algılayamadığı engelleri algılayarak aracın kaza yapmasını önlemiştir.

3.2.3.Park ve Durak Algoritması: Aracın park edebilmesi için sağlanan veriler LİDAR ve Zed2 kameradan elde edilir, verilerin çıktısı ise hız, fren ve hareket yönü direksiyon açıları olarak alınır. Araçta CANBUS iki kablo hattı üzerinden sağlanan bir haberleşme ağı bulunmaktadır. Sensörlerden alınan veriler değerlendirilip Can protokolü ile dönütler sağlanır. Can protokolünü diğerlerinden ayıran en önemli özelliği öncelikli ve 2 yönlü geçişlere sahip olmasıdır. Zed 2 ile durak tabelası algılandıktan sonra araç sağa doğru hareket eder, bekleme süresi kadar durakta bekler ve sola dönerek duraktan çıkarak normal düzenine döner.

3.2.4.Güvenlik Önlemleri: Araç ile ilgili alınan genel güvenlik önlemleri şu şekildedir:

-Araç dönüşlerinde yaşanabilecek hız problemleri için dönüşlerdeki hız, aracın düz giderken kullandığı sabit hızının yarısı olarak belirlenmiştir.

-Araç şerit takibi yaparken bir sorun yaşayıp şerit göremez ise ani fren yaparak durur ve şerit bulana kadar ilerlemeyi durdurur.

-Araçta bulunan Zed 2 kamerası ile nesneler arasındaki uzaklık 1 metrenin altına düşerse araç ani fren yapar ve tamamen durur.

-Araçta beklenmedik bir durum fark edilirse ön ve arka kısımlarında yer alan güç kesme butonları sayesinde durdurulur.

3.2.5.Simülasyon: Performans iyileştirmeleri, güvenlik yamaları gibi güncellemelerde çalışma ortamının rahatlığı sebebiyle Ubuntu 20.04 versiyonu, Python 3 desteği ve Gazebo uyumu sebebiyle ise açık kaynaklı ROS noethic versiyonu kullanılmıştır. ROS kavramsal bir işletim sistemi olduğu için gerçek bir sistem olan LINUX üzerinde çalışmalar yapılmıştır. ROS'da kullanılan paketler, gerçek araca yüksek benzerlikte bir robot modeli tasarlanması için destek olur. CUDA tercih edilerek algoritmaların aynı anda uyumlu işlem yapabilmesi sağlanmıştır. ROS kütüphanesi, Teknofest tarafından verilen modelin uyumluluğu, gerçek dünyaya benzer gürültülü veri üretme özelliği ve sensör ekleme kolaylığı nedeniyle de simülasyon ortamı olarak Gazebo tercih edilmiştir. Gazebo açık ve kapalı ortamlar için kullanılan açık kaynak kodlu bir simülasyon ortamıdır. Gzserver somut işlemleri gerçekleştirir, gzclient ise kullanıcının isteklerini yerine getirip simülasyonun görsele aktarımını sağlar. Harita oluşturulması için Blender programı üzerinden yollar çizilmiştir. Simülasyon ortamının asıl yarışma ortamı ile birebir olması açısından hava şartları, araç boyutu gibi bileşenler tamamıyla gerçeğe uygun olarak tasarlanmıştır. Takıma gönderilen hazır araç simülasyona entegre edilerek çalıştırılabilir hale gelmiştir.

4. BEÜ OVAT Takımı Kritik Tasarım Raporu İncelemesi

4.1. Ön Değerlendirme Raporu Hakkında

Bülent Ecevit Üniversitesi Otomasyon ve Arge Takımı elektronik, mekanik ve yazılım ekibi olmak üzere 3 ekipten oluşur. Ön Tasarım Raporu değerlendirmesi sonucu genel olarak raporda verilen bilgilerin detaylandırılması, şema ve görsellerle desteklenmesi gerektiğine karar verilmiştir. Ayrıca aracın dış kabuğu ile ilgili değişikliklere gidilmiş ve gereksiz görülen bazı parçalar araçtan çıkarılmıştır.

4.2. Yazılım Mimarisi

4.2.1.Şerit Takibi: Şerit tespiti yapmak için Zed 2 Stereo kamera kullanarak alınan görüntüler öncelikle gri tona dönüştürülmüş, sonrasında siyah ve beyaz ayrımı yapılarak şeritlerin belirginleşmesi sağlanmıştır. Görüntüde şeritleri algılarken problem yaşatabilecek kirlilikler bulanıklaştırılarak şeritlerin algılanması kolaylaştırılmış, OpenCV kütüphanesinde bulunan Canny fonksiyonu sayesinde kenar tespit işlemi gerçekleştirilmiştir. Son olarak Region of Interest sistemi ile belirli bir alan seçilerek sadece şeritlere odaklanılmıştır. Tespit edilen şeritlerin takibi için yine OpenCV'de bulunan ve çizgilerin segmentlerini bulup işleyebilen HoughLineP fonksiyonu kullanılır. Bu aşamada şeritlerin eğimini hesaplamak için devreye "numpy" kütüphanesi içinde bulunan "polyfit" fonksiyonu, şeritlerin açısını hesaplamak içinse "math" kütüphanesi içindeki "atan" fonksiyonu girer. Tüm bu algoritmalar oluşturulduktan sonra araç başarılı bir şekilde şerit takibi gerçekleştirebilir.

4.2.2. Nesne Algılama: Tabela, levha, trafik ışığı gibi nesneleri algılamak için Zed 2 Stereo kameradan alınan görüntüler kullanılmaktadır. Nesne tespiti yapmak için YOLOv4 ve YOLOv4-tiny algoritmaları arasında muhakeme yapılmış, YOLOv4-tiny düşük donanımlı araç içi bilgisayarlarda yüksek FPS vermesine rağmen yüksek doğruluk değeri vermemesi sebebiyle YOLOv4 tercih edilmiştir. Bu algoritma görüntüyü tek seferde nöral ağdan geçirerek resimdeki tüm nesnelerin koordinatını tahmin edebildiği için diğer ağlara göre daha hızlıdır. YOLOv4 nesne tespitini sağlarken görseli ızgaralara ayırır ve her ızgara kendi içindeki nesneleri algılar, tespit edilen nesneler sınırlayıcı kutular (bounding-box) içerisine alınır ve işlem tamamlanır. Nesne algılama için veri seti hazırlanırken seçilen fotoğraflar MakeSense.ai kullanılarak etiketlenmiştir. Veriler hazırlanırken yüksek çözünürlükte YOLO'nun nesneleri daha iyi seçmesi sebebiyle görsellerin piksel boyutlarına oldukça dikkat edilmiştir.

4.2.3. Park Algoritması: Araç bitiş çizgisinin üzerinden geçtikten sonra park algoritması başlar ve levha tanıma sistemi ile park edilebilir levhasına doğru yönelir. Park alanındaki dubalara çarpmamak için araçta bulunan ultrasonik sensör ile duba arasında belli bir mesafe kalınca araç durur. Bu sensörün yanı sıra Zed 2 Stereo kamerasının derinlik algılama özelliği ile araç park edilebilir tabelasına belli bir uzaklığa

gelince durmaya programlanmıştır. Bu sayede park görevi için birden fazla veri sağlanarak hatasız bir park hedeflenmiştir.

4.2.4. Gömülü Yazılım: Aracın otonom şekilde ilerleyebilmesi için elektronik kartın içerisinde geliştirilen gömülü yazılım araç için beyin olarak nitelendirilebilir. Görüntü işleme bilgisayarına gelen komutlara göre hız ve fren verilerinin ilgili elektronik devrelere iletilmesini sağlar. Akım gerilim sensörü ile araç içi akım kontrolünü sağlamak ve oluşabilecek hasarların önüne geçmek, ultrasonik mesafe sensörü ile yaşanabilecek kazaları engellemek amaçlanmıştır.

4.2.5. Simülasyon: Simülasyon ortamı kısıtlanmadan gerçeğe en yakın olacak şekilde tasarlanmak üzere UNITY 3D oyun motoru tercih edilmiştir. Unity fiziksel kamera özellikleri ile gerçek dünyadaki kamera bilgileri içe aktarılabilir, Rigidbody özelliği ile gerçek fizik ölçmeleri verilebilir ve Raycast özelliği ile sensörler test edilebilir. EasyRoads3D v3 paketi ile pist yeri oluşturulmuş, 3D-Blender yardımı ile şartnamede talep edilen varlıklar modellenip yerleştirilmiştir.

5. Sonuç

Yapılan detaylı Kritik Tasarım Raporu değerlendirmeleri sonucu Robotaksi Otonom Araç Yarışması hakkında yazılım alanında birçok yeni bilgi elde edinip genel işleyişin nasıl olduğuna dair güçlü bir temel oluşturabildiğime inanıyorum. Başlıca yarışma şartnamesi ve 3 farklı takımın proje işleyişi incelemeleri, internet üzerinden yapılan araştırmalar ve kişisel çıkarımlar da dahil olmak üzere edinilen tüm veriler şeffaf bir şekilde rapora eklenmiştir.

6. Kaynaklar

https://cdn.teknofest.org/media/upload/userFormUpload/2025_Robotaksi_Binek_Otonom_Arac_Yarismasi_Teknik_Sartnamesi_u3y7t.pdf

https://cdn.t3kys.com/media/upload/userFormUpload/hscPbmRxYoacCzh277Ou0uZaMvvvcFSK.pdf

https://cdn.t3kys.com/media/upload/userFormUpload/WXzr3izr9HYaTCaaYa8L39IrH3z3DYdl.pdf

https://cdn.t3kys.com/media/upload/userFormUpload/J0KtHW1ER4u05FZBvdNTgMRaeAE0qHlM.pdf