TEKNOFEST

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

ROBOTAKSİ – BİNEK OTONOM ARAÇ YARIŞMASI

KRİTİK TASARIM RAPORU

(HAZIR ARAÇ KATEGORİSİ)

TAKIM ADI: AREN

TAKIM ID: 424064

İÇERİK

1. Takım Organizasyonu 3
2. Ön Tasarım Raporu Değerlendirmesi 5
3. Hazır Araç Özellikleri ve Analizi 5
4. Sensörler 7
5. Araç Kontrol Ünitesi 10
6. Araç Dinamiği Modelleme/Tanılama ve Kontrolü 11
8. Güvenlik Önlemleri 18
9. Simülasyon 19
10. Sistem Entegrasyonu 22
11. Test ve Doğrulama 22
12. Referanslar 26

1. Takım Organizasyonu

Aren Robotaksi takımı 2021 yılında Düzce Üniversitesi Teknoloji ve Bilişim Topluluğu tarafından kurulmuştur. Takımımız Düzce ve Gebze Teknik Üniversitesinden farklı mühendislik bölümlerinden oluştuğu için multidisipliner bir ekip yapısına sahibiz. Takım olarak Teknofest'in düzenlemiş olduğu farklı kategorilere de katılım sağlamaktayız, 9 mühendislik öğrencisi olarak hedefimiz yazılım teknolojileri alanında tecrübe edinmek ve bu alanda ülkemizi temsil etmektir.

Çalışmalarımızı kolaylaştırmak adına takım organizasyonunu 4 alt gruba ayırdık. Bunlar aşağıdaki şemada da gözüktüğü üzere; nesne tespiti, şerit takibi, simülasyon ve sürüş kontrolüdür.

Şekil 1

İş Planlaması (İP)			Aylar							
İP No	**İP Adı/Tanımı**	**Kim(ler) Tarafından Yapılacağı**	**Aralık**	**Ocak**	**Şubat**	**Mart**	**Nisan**	**Mayıs**	**Haziran**	**Temmuz**
1	Yarışma ile ilgili literatür araştırılması	Tüm üyeler	✓	✓						
2	Otonom sürüş için gerekli olan program ve yazılımların öğrenilmesi	Tüm üyeler	✓	✓	✓	✓				
3	Simülasyon ortamının öğrenilmesi	Üye 8 Üye 9		✓	✓	✓				
4	Simülasyon ortamının parkura göre hazırlanması	Üye 8 Üye 9			✓	✓	✓			
5	Trafik levhalarının ve ışıklarına ait verilerin toplanması	Üye 1 Üye 2		✓	✓	✓				
6	Veri setlerinin düzenlenip etiketlenmesi	Üye 1 Üye 2			✓	✓	✓			
7	Etiketlenen verilerin derin öğrenme yöntemiyle eğitilmesi	Üye 1 Üye 2						✓	✓	
8	Trafik levhalarının ve ışıklarının algılanması için algoritmaların oluşturulması	Üye 1 Üye 2						✓	✓	✓
9	Şerit takibi algoritmasının oluşturulması	Üye 3 Üye 4				✓	✓	✓		
10	Şerit ve nesne tespitinden gelecek bilgiler ile karar yapısı algoritmasının oluşturulması	Üye 1 Üye 2 Üye 3 Üye 4						✓	✓	✓
11	Otonom sürüş algoritmasının oluşturulması	Üye 5 Üye 6 Üye 7			✓	✓	✓			
12	Park ve durak algoritmalarının oluşturulması	Üye 5 Üye 6 Üye 7					✓	✓		
13	Algoritmaların simülasyon ortamına aktarılması ve test edilmesi	Üye 5 Üye 6 Üye 7				✓	✓	✓		
14	Algoritmaların hazır araca aktarılması ve test edilmesi	Üye 5 Üye 6 Üye 7								✓
15	Kullanılan algoritmaların geliştirilmesi	Tüm üyeler	✓	✓	✓	✓	✓	✓	✓	✓
16	Hata tespitleri ve onarım çalışmaları	Tüm üyeler	✓	✓	✓	✓	✓	✓	✓	✓

Şekil 2

2. Ön Tasarım Raporu Değerlendirmesi

Aren takımı olarak, Ön tasarım raporu öncesinde yapmış olduğumuz araştırmalar ve ilgili akademisyenlerden almış olduğumuz tavsiyeler sonucunda başlangıçta doğru adımlar atarak, belirtmiş olduğumuz program ve yazılım algoritmalarında değişikliğe gidilmemiştir.

Ön tasarım raporumuzu ve geri bildirimleri titizlikle inceledik. Yazılımsal olarak eksik olduğumuzu tespit ettik ve bu alandaki eksikliklerimizi gidermeye yönelik çalışmalar başlattık ve aramıza bilgisayar mühendisi bir arkadaşımızı da dahil ettik. Özellikle şerit takibindeki sürüş mantığımızı tamamen yenileyerek optimum sonucu elde etmek için algoritmalarımızda da geliştirmeye gittik.

Simülasyon olarak Gazebo'da bilgimizi ne yazık ki tam olarak aktaramadığımız için ön tasarım raporundan ciddi bir puan kaybı yaşadık. Simülasyondaki eksiğimizi kapatmak adına çözüm yolu olarak farklı simülasyon ortamlarını denemeye karar verdik ve kullanışlılık açısından bize en çok uygun simülasyon ortamının Unity olduğuna karar verdik. Şartnamede ve daha sonrasından bizlere verilen tabela konumlarına göre haritamızı tekrardan güncelleyerek oluşturduk. Kodlarımızı farklı yöntemlerle entegre edip simüle etmeyi başardık.

Nesne tespiti sistemimizde önceki raporumuzda YOLO üzerinden yapacağımızı bahsetmiştik ve sonuçlardan bazı fotoğraflarla desteklemiştik. Nesne tespitimizin fotoğraflarda hızlı fakat test sırasında araç üzerindeki kameradan gelen video görüntüsünde yavaş olduğunu fark ederek iyileştirmeye gittik. Ayrıca bazı açılardan iki farklı tabelanın görüntüsü de gözükmektedir. ZED kameranın sağlamış olduğu derinlik özelliği sayesinde sadece araca yakın olan tabelanın tespiti hedeflenmiştir. Bu değişiklik sayesinde birden çok tabelanın algılanmasından doğan hataların önüne geçilmiştir.

Verilen ÖTR geri bildirimde "Sensör füzyonu için de yeterli yaklaşımlar sunulmamıştır." ve "Projeyi ifade eden bir yazılım mimarisi sunulmamıştır." uyarıları dikkate alınmış, üzerine çalışıp eksiklerimiz telafi edilmiştir.

3. Hazır Araç Özellikleri ve Analizi

Araç Teknofest ekipleri tarafından verilmektedir. Modeli Tragger T-Car' dır. Teknik özellikleri olarak motor modeli 7,4 KW Schaubmuller olup 125 NM toru vardır. 48v çalışma gerilimine sahiptir. Trojan T145 Plus bataryasına sahiptir. Araç iki kişi için tasarlanmıştır. Maksimum 24 Km/H hız ile gidebilmektedir. Şasisi 6000 Serisi olup yüksek Mukavemetli Uçak Sınıfı Alüminyum ile şasi kaplanmıştır. Bağımsız Teleskopik Amörtisörü önde, Helezyon Yaylı Sabit Aks/ Teleskobik Amörtisörler arka tarafındaki süspansiyon sistemidir. Lastikleri 10" Duro modelidir. İki çeşit freni vardır. Bunlar; park freni ve servis frenidir. Park freni Mekanik/ayak pedalı, servis freni ise Hidrolik/Kampana modelidir. Dört çeşit sensörü bulundurmaktadır. Bu sensörler ayrıca ele alınmıştır. Velodyne Puch Hi-Res 16 Lidar modeli, Zed2 Stereo kameralı ve otonom kontrolör olarak Nvidia Xavier bulunmaktadır. Lidar ve kamera aracın ön kısmında, Xavier ise arka kısımda bulunmaktadır. Araç haberleşme protokolü olarak CAN kullanılmaktadır.

- ## Teknik Verileri

Motor	7.4 KW Schaubmuller
Tork / Nm (Max)	125 NM
Çalışma Gerilimi / V	48 Volt
Batarya	Trojan T145 Plus
Oturma Kapasitesi	2,4,6,8
Max Araç Hızı Km/h	24
Şasi	6000 Serisi *Yüksek Mukavemetli Uçak Sınıfı Alüminyum*

- ## Süspansiyon Sistemi

Ön Süspansiyon	Bağımsız Teleskopik Amörtisör
Arka Süspansiyon	Helezon Yaylı Sabit Aks / Teleskobik Amörtisör
Lastikler	10" Duro

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

- ## Fren Sistemi

Park Freni	Mekanik/Ayak Pedallı
Servis Freni	Hidrolik/Kampana

- Araç Parçaları ve Modelleri

Oriental Motor	BLVM640N-GFS_ORIENTAL
Manyetik Kavrama	ABK 04 3.1 DIZAYN
Lineer Aktuatör	MD24A050-0100CNO2NNSD
Enkoder	8.F5888.5B2F.2123
Lidar	Velodyne Puch Hi-Res 16
IMU	https://www.sbg-systems.com/products/ellipse-micro-series/
Kamera	Zed2 Stereo
Otonom Kontrolör	Nvidia Xavier https://www.nvidia.com/tr-tr/autonomous-machines/embedded-systems/jetson-agx-xavier/
Araç Kontrolörü	Otomotiv Araç Kontrol Ünitesi 3 x Canbus 1 x GPS 1 X RS485 1x GSM Modülü

4. Sensörler

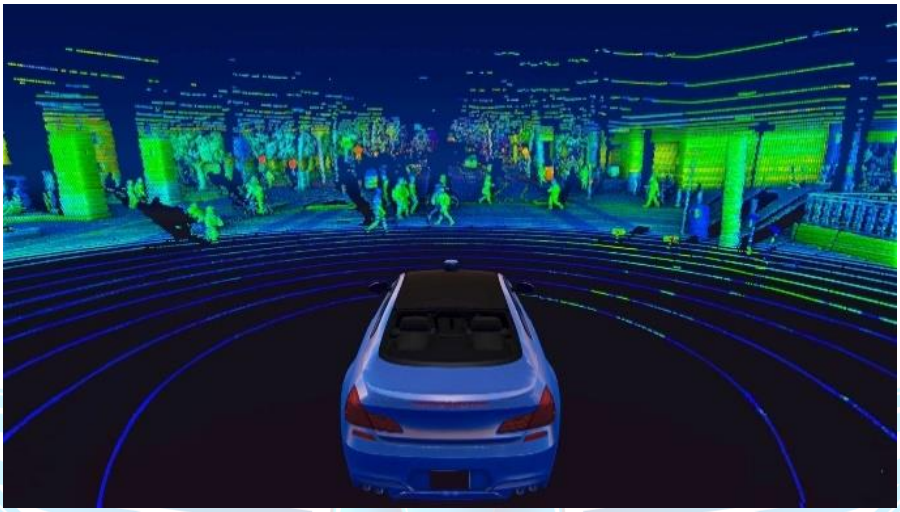
AREN takımı olarak Otonom araç projesinde ZED2 kamera, LİDAR, IMU ve GPS sensörleri kullanılmaktadır.
Sensörler, dört farklı şekilde ele alınacaktır.

4.1. Zed 2 Stereo Kamera

ZED Stereo kamera bize yüksek çözünürlük ve fps (frame per second) değerine sahiptir. Bu sayede yüksek kalitede üç boyutlu video yakalayabilmekteyiz. Ayrıca gölge veya hava şartlarının kötü olması gibi durumlarda kalitesiyle bizlere güzel bir imkân sunmaktadır. Farklı video modlarıyla görüntü alma özellikleri de mevcuttur.
Bu kamera şerit takip sistemimizi ve trafik levhalarının tespitinde bize yardımcı olmaktadır. Sağladığı görüntü ile bilgisayara net çözünürlüklü görseller ve videolar göndererek hazırlamış olduğumuz kodlarla bu verileri işleyebilmemizi ve aracın o verilerle hareketini sağlamaktadır.

4.2. LIDAR (Laser Imaging Detection and Ranging) Sensörü

LIDAR Sensörü lazer darbeleri kullanarak bir nesne veya yüzeyin uzaklığını anlayarak araca sinyal göndermektedir. Bu sayede hareketli veya sabit engelleri tespit ederek aracın durmasını ya da hızını düşürmesini sağlar. Bize engel tespitinde yardımcı olarak fayda sağlamaktadır. Çalışma prensibi Radar teknolojisine benzemektedir. Hem uzaklığı ölçer hem de nesneye çarpıp geri dönen yansımayla aradaki mesafeyi ölçmüş olur.



4.3. IMU (Inertial Measurement Unit)

IMU; ana işlemciye gönderilen açısal hız ve doğrusal ivme verisini tek bir modülde toplayan elektronik bir birimdir. Aracın hızını, dönme açısını ve dönme miktarı yardımıyla sistemin entegre olduğu kameranın elde ettiği görüntüler hassas şekilde koordinatlandırılmaktadır.

İvmeölçer, jiroskop ve manyetik alan ölçer olmak üzere üç temel bileşenden oluşmaktadır. İvmeölçer üç farklı analog sinyal üretir. Sensör sürekli olarak yer çekiminin etkisinde kalmaktadır. Bir, iki veya üç eksende ölçüm yapabilen türevleri vardır. IMU' da bulunan bir diğer sensör ise jiroskoptur. Buna dönüş ölçer de denilebilir. Tekerleğin ekseni etrafında hızlıca döndürülmesi ile oluşur. Tekerleğin etrafındaki çembere dik açıyla tutturulan başka bir çember vardır. Bu çemberlere dik açıyla tutturulmuş başka bir çember jiroskobu temsil eder. Jiroskobun temelde iki özelliği vardır. Yatay eksende dönmekte olan bir jiroskoba yatay eksende bir kuvvet uygulandığında eksen etrafında dönmeye başlar. Diğer özelliği ise jiroskobun dönüş eksenine sabit kalmasıdır.

4.4. GPS (Global Positioning System)

Genel itibarıyla anlık konum bilgisi sağlayan GPS, araçta bulunan önemli sensörlerdendir. Haritacılık sektöründe, Askeri amaçlı uygulamalarda, Deprem araştırma çalışmalarında, Deformasyon belirlemede, İnşaat sektöründe de kullanılan bu sensör, düzenli olarak kodlanmış bilgi yollayan bir uydu ağıdır ve uydularla arasındaki mesafeyi ölçerek Dünya üzerindeki kesin yeri tespit etmeyi mümkün kılar. IMU sensöründe bulunan ivmeölçer ve jiroskop ile sürekli olarak aracın pozisyonunu, yer değiştirmesini ve hızını ölçen bir sistemle birlikte kullanılır.

5. Araç Kontrol Ünitesi

Son zamanlarda gelişen teknolojiyle birlikte otonom araçlarda merkezi ağ sistemi önemli bir hal aldı. Aracın alt üniteleri merkezi ağ sistemi ile haberleşmektedir. Bizim için bu alt ünitelerin haberleşmesini ve fazlalık olan kablolardan kurtulmak için CAN Bus kullanılır. Aşağıda gördüğünüz görüntü canbus a aittir.

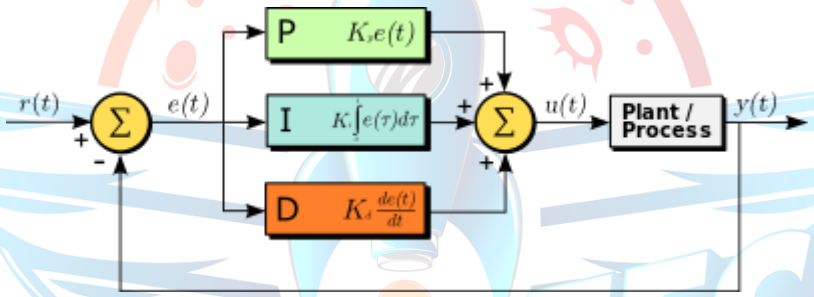
Canbus araç içerisindeki bir mesajı tüm ünitelere gönderir ve gerekli olan ünite mesajı alıp işler. Böylece araç içerisindeki tüm üniteler birbiriyle haberleşmiş olur. Sensörlerden alınan bilgilerin işlenmesi ve anlamdırılması için gömülü ve bilgisayar tabanlı platformların kullanılması mümkündür. Aracımızda bulunan Nvdia Jetson Xavier kiti ile yapmaktayız. Derin öğrenme, bilgisayar görüsü, GPU hesaplama, multimedya işleme gibi birçok konuda kullanılmaktadır.

	Jetson AGX Xavier Serisi		
	AGX Xavier 64GB	AGX Xavier	AGX Xavier Industrial
Yapay Zeka Performansı	32 TOPS		30 TOPS
GPU	512 NVIDIA CUDA çekirdekli ve 64 tensör çekirdekli NVIDIA Volta mimarisi		
CPU	8 çekirdekli NVIDIA Carmel Armv8.2 64 bit CPU 8 MB L2 + 4 MB L3		
DL Hızlandırıcı	2 adet NVDLA		
Görüş Hızlandırıcı	2 adet PVA		
Güvenlik Kümesi Altyapısı	-		Lockstep'te 2 adet Arm Cortex-R5
Bellek	64 GB 256 bit LPDDR4x 136,5 GB/sn	32 GB 256 bit LPDDR4x 136,5 GB/sn	32 GB 256 bit LPDDR4x (ECC desteği) 136,5 GB/sn

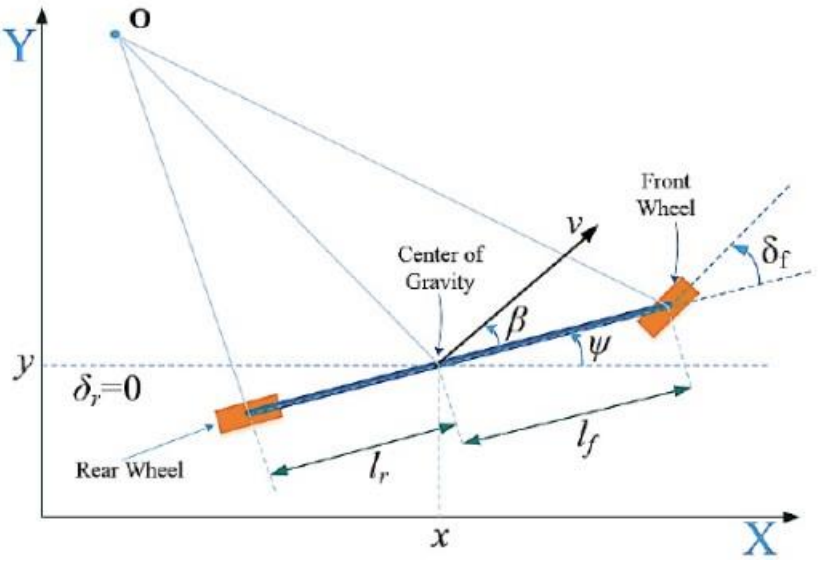
6. Araç Dinamiği Modelleme/Tanılama ve Kontrolü

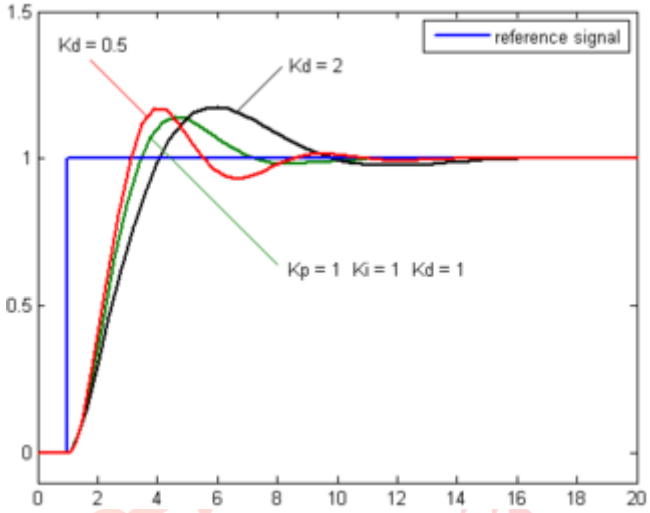
6.1. PID Kontrol

PID (oransal-integral-türevsel) denetleyici kontrol döngüsü yöntemi, elektronik cihazlar, mekanik cihazlar ve pnömatik sistemler gibi geniş bir kullanım yelpazesine sahip olan bir kontrol mekanizmasıdır. PID, geri besleme yolu ile girişe gönderilen sinyali giriş sinyaliyle karşılaştırıp elde edilen hatayı hesaplar. Bu hata, PID'ye gönderilir ve PID bu hatayı bir katsayı ile çarptıktan sonra çarpım işleminin sonucunun türevi ile integralini alır ve tekrardan çıkışa gönderir. Bu işlem hata minimum değere ulaşana kadar tekrarlanır.



PID, şerit takibi algoritmamızdan gelen "sola dön", "sağa dön" ve "düz git" komutlarının araç üzerinde dönüş açısının daha tutarlı bir şekilde olmasını sağlamaktadır.





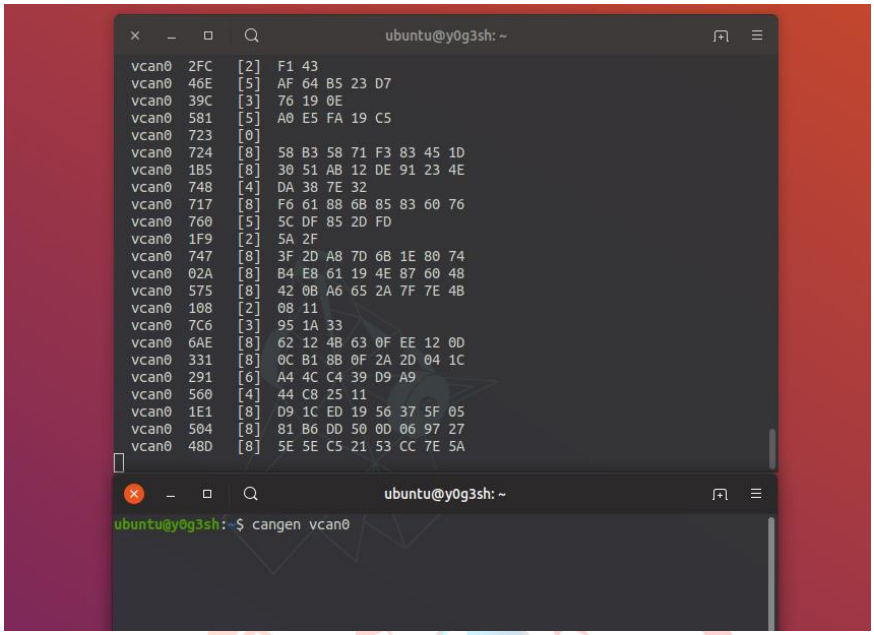
P, I, D parametrelerinin değişken değerleri Kp, Kd ve Ki olmak üzere üç adettir. Bu üç parametrenin en verimli anını belirleyebilmemiz için araç üzerinde test etmemiz gerekmektedir. Test sürecinde optimum sonuçları bularak şerit takibimizi sorunsuz bir şekilde gerçekleştirmeyi hedeflemekteyiz.

6.2. Can Bus Haberleşme Protokolü

Araç kontrol ünitesinde bahsetmiş olduğumuz Can bus ı aracın beyin düşüncelerine benzetebiliriz. Çünkü bu protokol sayesinde araç üzerinde yapmış olduğumuz mekanik ve elektriksel her fonksiyonu görmemizi ve aracın uzaktan kontrolünü yapabilmemizi sağlamaktadır.

Hazır aracımızda bulunan can bus verilerine terminale “candump can0” yazarak ulaşabiliyoruz. Karşımıza çıkan sonuç aşağıdaki resimde verilmiştir.

TEKNOFEST
HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ



Aracın hareketini sağlayabileceğimiz Can mesaj adı 0x560'dır. Test sürecinde araç üzerinde sinyal hali hazırda bulunmadığı için 4 bit bulunmaktaydı bu bitlerden ilki aracın kontağı, ikincisi direksiyon açısı, üçüncüsü aracın hareketini (ileri ve geri) sonuncu bit ise el frenini temsil etmektedir. Beşinci bit olarak sinyal geleceği söylenmiştir. Algoritmalarımızı bu varsayıma göre güncelleyeceğiz. Heksadesimal türden Can protokol haberleşmesi aşağıdaki tabloda gösterilmektededir.

Kontak	0 Kontak Kapalı, 1 Kontak Açık
Direksiyon açısı	00 Tam Sol, FF Tam Sağ, 81 Düz
Hareket	00 Geri, FF İleri, 81 Hareketsiz
El freni	00 El Freni İnik, FF El Freni Çekik
Sinyal	0 Sinyal Kapalı,1 Sol Sinyal, 2 Sağ Sinyal

7. Otonom Sürüş Algoritmaları

7.1 Şerit Takibi

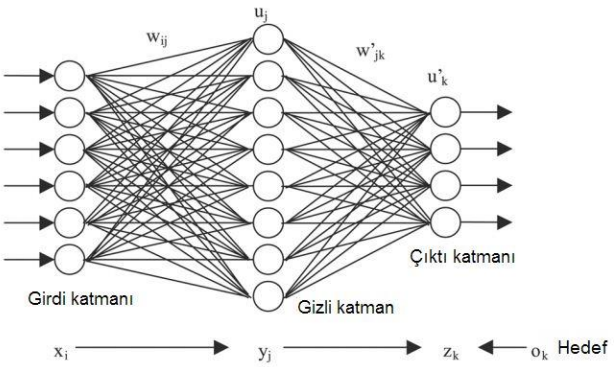
Simülasyonda ve test aşamalarında araçtan aldığımız videolarda uyguladığımız şerit takip sistemi başarılı bir şekilde çalışmaktadır. Sadece ufak bir revize oldu. Örnek parkurda ve simülasyonumuzda kullandığımız yarışma kapsamında tasarlanan otonom araçta şerit tespitinde başarılı olmak için araçtan görüntü almakta ZED2 kamera yarımcı olmaktadır. Yaptığımız revize ise bu kameralarda görünen şeridin ortalamasını alan üçgen şeklinde bir sistem hazırlayarak aracın şeridi doğru ve güvenli bir şekilde takip etmesini sağladık.Zed2 kamerası görüntüyü kaliteli bir şekilde alarak gölgelerde ya da aşırı yansımalarda kalitesiyle ön plana çıkarak bazı sıkıntı yaşanan durumlarda destek sağlamaktadır.

AREN takımı olarak Ön Tasarım Raporu'nda belirttiğimiz sistemi uyguladık ve OpenCV fonksiyonlarıyla şeritleri tespit ettik.

7. 2. Nesne Tespiti

7.2.1 Yapay Sinir Ağları

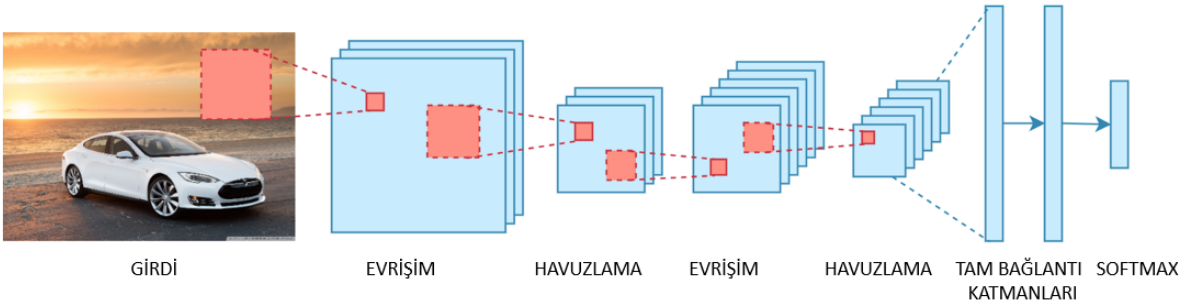
Yapay Sinir Ağları insan beyninin işleyişinden esinlenerek tasarlanmıştır. Bu nöron modelinde giriş ağırlıkları, bir çıkış ağırlığı, toplama ve aktivasyon fonksiyonu bulunmaktadır. Nörona gelen giriş işaretleri, giriş ağırlıkları ile çarpılarak toplama bloğunda bias ağırlığı ile toplanıp aktivasyon fonksiyonundan geçirilerek çıkış ağırlığı ile çarpılır ve böylece çıkış işareti elde edilir.



7.2.2 Konvolüsyonel Sinir Ağları

Konvolüsyonel Sinir Ağları (ConvNN) görüntü tanıma ve görüntü sınıflandırma alanlarında kendini kanıtlamış sinir ağlarının bir çeşididir. Yüz, nesne ve yol işaretlerini tanıma işlemlerinde oldukça başarılı olduğunu kanıtlamış ve robot, kendi kendini süren arabaların geliştirilmesinde büyük katkıda bulunmaktadır.

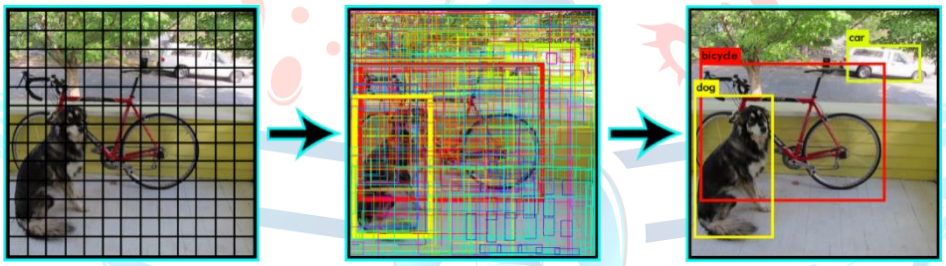
Konvolüsyonel Sinir Ağları ismini konvolüsyon operatöründen almaktadır. Konvolüsyonun temel amacı ise giriş görüntüsünün özelliklerini çıkarmaktır. Konvolüsyon giriş verisinin küçük karelerini kullanarak ve görüntü özelliklerini öğrenerek pikseller arasındaki uzaysal ilişkiyi korur. Herhangi bir Konvolüsyon operatörü, ReLU (Rectified Linear Units), alt örnekleme ve tam bağlantılı katman olarak 4 ana katmandan oluşur. Sinir ağlarını oluştururken bu katmanlar birden faza kez kullanılabilmektedir. Her konvolyüsyonel işlemden sonra ReLU aktivasyon fonksiyonu kullanılmaktadır. ReLU her bir piksel için uygulanır ve negatif değerleri taşıyan pikselleri sıfır ile değiştirir. ReLU yerine diğer hiperbolik tanjant ya da sigmoid aktivasyon fonksiyonları kullanılabilir, ama ReLU çoğu durumlarda daha iyi performans gösterdiği tespit edilmiştir. Alt örnekleme olarak bilinen katman her bir özellik haritasının boyutunu küçültmekle beraber onun önemli bilgilerini korur. Tam bağlantılı katmanın temel amacı giriş görüntünün konvolyüsyonel ve alt örnekleme işlemleri yardımıyla elde ettiğimiz üst düzey özelliklerini kullanarak ve eğitim veri setine dayanarak giriş görüntüsünü sınıflandırmak. Daha iyi performans sergilemek için genelde daha büyük sinir ağlarını eğitilir veya birden fazla modeli bir araya getirilir. YOLOv2'de büyük sinir ağlarını kullanmak yerine onları basitleştirerek ve daha sonra öğrenme yöntemi de kolay hale getirilmiştir.



7.3. YOLO Algoritması ve Uygulanması

Trafik levhalarının ve nesnelerin tespitinde kullandığımız YoloV4 algoritması bizlere yarımcı olmaktadır. Öncelikle toplayabildiğimiz kadar levha görsellerini Github Labell. mg üzerinden etiketledik. Etiketlenen görselleri YoloV4 algoritması ile eğiterek aracımıza entegre ettik. Bu sayede parkurda aracımız hareket ederken levhaları tespit ederek yönünü ve açısını ayarlamaktadır. Yaklaşık olarak 20.000 görsel test videomuzdan, internetteki görsellerden, sokaklardaki trafik işaretlerinden topladığımız görsellerle birleştirerek etiketleme ve eğitme işlemini gerçekleştirdik.

YOLOV4-Tiny algoritmasını kullanmamızdaki asıl amaç hızlı olmasıdır. Tek bir seferde resmi nöral ağından geçirerek resimdeki tüm nesnelerin sınıfını ve koordinatlarını tahmin edebilmektedir. Yani bu tahmin işleminin temeli, nesne tespitini tek bir regresyon problemi olarak ele almalarında yatıyor. Bunu yapmak için ilk önce girdi resmini SxS'lik ızgaralara bölüyor. Bu ızgaralar 3x3 5x5 19x19 gibi olabilir.



Her bir ızgara kendi içinde, alanda nesnenin olup olmadığını, varsa orta noktasının içinde olup olmadığını, orta noktası da içindeyse uzunluğunu, yüksekliğini ve hangi sınıftan olduğunu bulmakla sorumludur. YOLO her ızgara için ayrı bir tahmin vektörü oluşturur. Bunların her birinin içinde:

GÜVEN SKORU: Bu skor modelin içinde geçerli ızgara içinde nesne bulunup bulunmadığından ne kadar emin olduğunu gösterir. (0 ise kesinlikle yok 1 ise kesinlikle var)
Eğer nesne olduğunu düşünürse de bu nesnenin gerçekten o nesne olup olmadığından ve etrafındaki kutunun koordinatlarından ne kadar emin olduğunu gösterir.

7.4. Park Algoritması

Aracımız Zed2 kamera ile eğittiğimiz tabelalardan biri olan park tabelasını algılar. Sonra belli bir açıyla kendisini ayarlayarak park tabelasının orta noktasına doğru direksiyon manevrasıyla döner. Araç park yapılacak alana dönüşünü tamamladıktan sonra Lidar ile belirli bir mesafeye kadar yaklaşarak parkını tamamlamış olur.

7.5. Hareket

7.5.1 Hareketin Hiyerarşisi

Aracın hareket algoritması en temelinde 2 farklı başlık altında incelendi ve şekillendi. Bunlardan ilki nesne tespiti sayesinde canlı kamera görüntüsünden çıkardığımız tabela imgeleridir Algoritmamızda hiyerarşik olarak en üst katmanda yer alıyor. İkinci katmanda ise şerit takibi sayesinde aracın sağ veya sola yönlenmesi yer alıyor.

7.5.2 Tabela algılama

Bu katmanın en üstte yer almasının sebebi; bu katmanda olacak olan değişikliğin, aracın yakın gelecekteki hareketinin majör sebebi olmasıyla paraleldir. Tabela algılamalarının en doğru şekilde çalıştığından emin olmak adına tabelaların minimum %70 oranda algılanması gerekmektedir. Bu regülasyonun, aracın özgürlüğünü kısıtladığı düşünülebilirse bile tam da böyle değildir. Bu güvenlik önlemi sayesinde istenmeyen durumların önüne geçilebilmiştir. Son olarak bu sayede algoritmamız, yapması ve yapmaması (Sola dönülmez tabelası vs.) gerektiği şeyleri algılayıp buna göre rota belirleyebiliyor. Bunu yaparken kodda “alert” değişkeni ile belirtilen uyaranlarla, farklı durumların gerekliliklerini barındıran senaryolarda optimum verim elde edilmeye çalışıldı.

7.5.3 Aracın Şeridi Takibi

Başlık 1 de belirtildiği gibi araç, üstündeki kameralarla iki beyaz çizgiyi algılayıp kendini arada konumlandırmaya çalışıyor. Bu konumlandırma sırasında aracın lastiklerinin aşırı dönmesini engellemek ve daha stabil bir hareketi sağlamak adına halihazırda kullanılan PID teknolojilerinden de ilham alarak buna oranla ilkel ama etkili olan; lastiklerin kısa sürede aşırı dönmesini engelleyen bir algoritma tasarladık. Bu algoritma sayesinde aracın hareketinde aşırı dalgalılık önlenmiş oldu ve yumuşak geçişler sağlanabildi. 6.5.1 de bahsedildiği gibi şerit takibi hiyerarşide 2. Sırada bulunuyor. Şerit takibinin çok sık yapılıyor olması gerekliliğini bildiğimizden dolayı şerit takip kodunu oldukça yalın ve optimize tutmamız gerektiğini düşündük ve buna uygun hareket ettik.

8. Güvenlik Önlemleri

Otonom araç seyir halinde ilerlerken herhangi bir aksilik veya arıza durumunda hızını ayarlamalı ya da herhangi bir bölgeye giderek durmalıdır. Aracın şeridini kaybetmesi durumu, levhaların net olmama veya aracın belli bir nedenden dolayı levhaları algılayamaması durumu, araçtaki sensörlerin bozulması durumu ve haberleşme sistemlerindeki bozulmalar aracın acil önlem alması için başlıca örneklerdir.

8.1 Şerit Tespiti ile Güvenlik Önlemleri

Şeritlerin algılanamaması veya şeritten herhangi bir nedenden dolayı çıkması durumunda aracın hızını trafiğe uygun bir şekilde azaltması gerekmektedir. Parkurda yaşanabilecek bu durum karşısında Lidar sisteminin nesnelere belli bir mesafede dur mesajını aracın algılayıp durması ve bariyerlere çarpmaması beklenir.

8.2 Nesne Tespiti ve Nesnelerin Uzaklık Hesabı ile Güvenlik Önlemleri

Araç üzerindeki sensörlerden elde edilen verilerin yazılımsal unsurlar sayesinde tanımı yapılmaktadır. Tanımı yapılamayan ya da sensörlerden yeterli bilgi alınamadığı durumlarda araca hızı yavaşlatma ve araca durdur komutu yazılımsal olarak gönderilir. Daha ciddi durumlarda ise araçta iki tane olmak üzere acil durum butonlarına basılarak veya yine aracın kumandasından aracın elektriğini keserek araç durdurulur.

8.3 Sensörlerin Bozulması Durumunda Güvenlik Önlemi:

Otonom sürüş sırasında herhangi bir sensörün bozulması veya sensörlerin temiz olmamasından kaynaklı sınırlı görüş durumlarında CanBus ağ üzerinde daha güvenli bir bilgi alış verişi sağlamaya devam eder fakat bozulmuş sensörler daha çok tehlike oluşturmaktadır. Bu yüzden araca hızı yavaşlatma ve araca dur komutu gönderilir.

8.4 Haberleşme Altyapısının Güvenlik Önlemlerindeki Yeri

Gömülü sistemler, donanım ve yazılımın sıkı bir birliktelik içinde çalıştığı, başka çevre birimlerinin de olduğu, belirli bir amaç için tasarlanmış sistemlerdir.

Günümüzde birçok kontrol uygulaması birden çok kontrolcüye ihtiyaç duymaktadır. Bu kontrolcülerin birbirleri ile güvenli, hızlı ve düşük maliyetli haberleşmesi büyük önem taşımaktadır. Bu ihtiyaca en iyi cevap veren Can haberleşme protokolüdür. Can haberleşme protokolünün incelenmesi, veri yolu kullanarak bir mikro denetleyici ağ yapısının kurulması ve kurulan bu ağ yapısı üzerinde bulunan denetim sistemlerinin bir araya getirilerek bir bilgisayar tarafından kontrol edilmesi hedeflenmiştir. Bu sayede otonom araçtan eş zamanlı veri analizi elde edip araç güvenliği için sağlıklı veri elde etmiş olmaktayız.

9. Simülasyon

Simülasyon ortamımızı Gazebo ortamında gerçekleştirmiştik. Ayrıca Aren takımı olarak simülasyon ortamını kurmak için daha rahat çalışabileceğimiz Unity ortamında da örnek yarışma parkurunun bir simülasyonunu tasarladık. Bu oyun tarzı tasarımımızla beraber denemelerimizi yapmak daha verimli oldu.

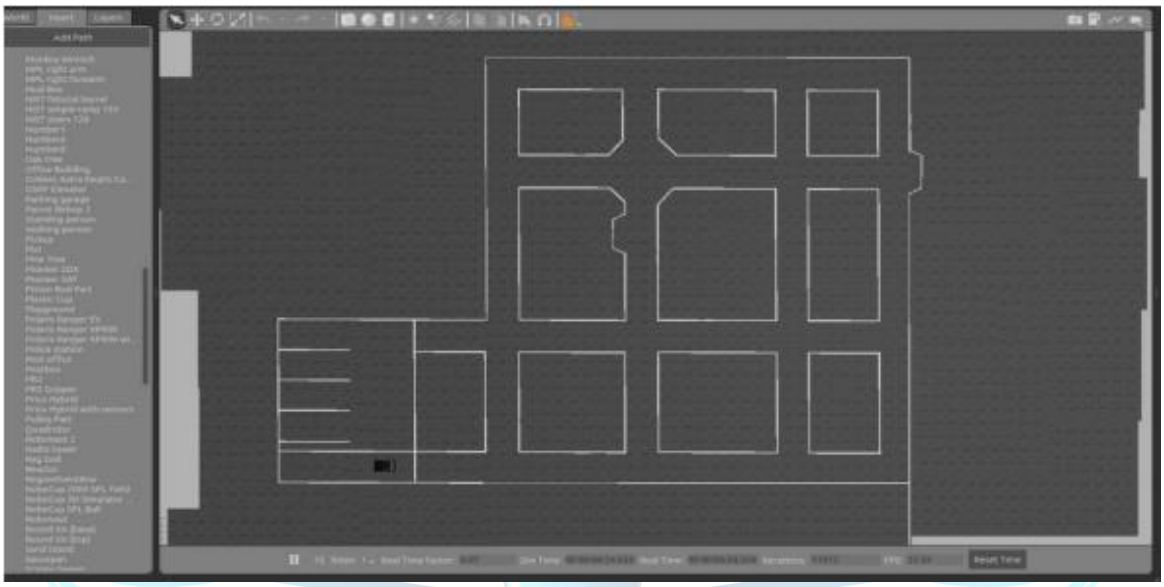
Teknofest' in yollamış olduğu Solid Works çizimini Blender ile düzenleyerek Unity' e uygun hale getirdik. Sonrasında testte başarılı sonuçlar aldığımız kodlarımızla birtakım denemeler gerçekleştirdik. Eksikliklerimizi de düzelterek kodlarımızı yarışmaya hazır hale getirdik. Aynı şekilde haritalarımızı da geliştirmek için Blender ve Sketchup kullandık. İşletim sistemi Ubuntu 18.04 dür. Çünkü açık kaynaklı bir yazılım olması ve kullanılan donanımın Nvidia Xavier, Ubuntu 18.04'ü desteklemesidir.

9.1 Gazebo

Gazebo simülasyon programı 2002 yılının sonlarına doğru California'da öğretim görevlisi Dr. Andrew Howard ile öğrencisi Nate Koenig iş birliğiyle başlanmıştır. Açık kaynak kodlu olarak Linux ortamında geliştirilmiştir. 2009 yılında kıdemli araştırma mühendisi Jhon Hsu tarafından Gazeboya, ROS (Robot Operating System) ve PR2 entegre edilmiştir. Bu entegreden sonra Gazebo programı daha fazla tercih edilen bir simülatör programı olmuştur. Apache 2,0 ile lisans alınmıştır. İlk başta dış ortama göre tasarlanması sebebiyle Gazebo ismi verilmiştir. Daha sonra kapalı ve açık mekânlar için geliştirilmiştir. Ücretsiz ve açık kaynak kodlu, robot simülasyon ortamıdır. Bir ve birden fazla robotun 3 boyutlu ortamda simüle edilebildiği bir programdır. V-REP gibi DART, Simbody, ODE, Bullet kütüphanelerinin kullanılması ile fiziksel ortamı işlemesi kolaylaştırılmıştır. Görüntüleme işlemi için ise yine açık kaynak kodlu olan OGRE kullanılmıştır. Sahneleme işleminde en iyi başarımı NVIDIA GPU'lar vermektedir. Gazebonun simülasyon programının yapısı ise diğer simülasyon programlarına göre daha işlevsel hale getirilmiştir. Farklı kütüphaneler kullanılarak her işlevin birbirinden bağımsız çalışması sağlanmıştır.

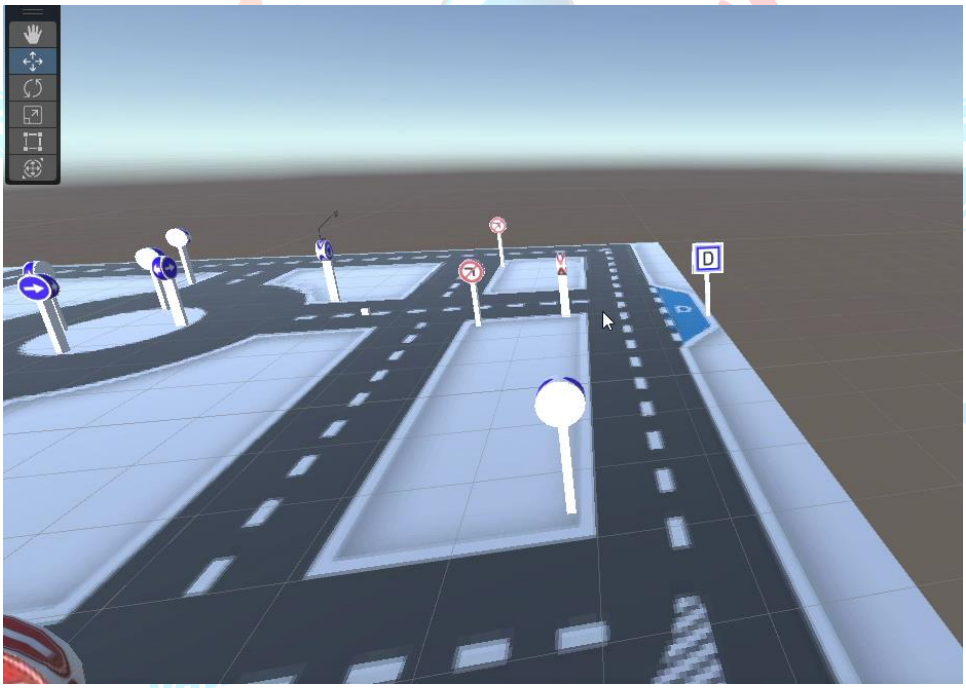
Gazebo genelinde sunucu-client mantığıyla çalışmaktadır. Gzserver (sunucu) somut işlevleri yaparken (robot ortamda dolaşırken), Gzclient (istemci) kullanıcının isteklerinin yerine getirilmesi ve simülasyonun görsele aktarılmasını sağlar. V-REP simülasyon programına göre Gazebonun kullanıcı arayüzü daha vasat kalmıştır. Fakat ROS'un entegre edilmesi ile Gazebonun bu eksikliği kapatılmıştır. Gazebo, geliştiricilere C++ gibi nesneye yönelik diller ile geliştirme yapma olanağı sağlamıştır. Kodların Linux işletim sisteminde derlenmesi sebebiyle geliştirmenin daha verimli olması için program geliştiricilerinin bu dilleri iyi bir şekilde kullanabilmelerinin önemi büyüktür. Gazebo programının yüklenmesiyle birlikte, Gazebonun online veri tabanından bir çok ortam (dünya), robot ve nesneye ulaşabiliriz. Veri tabanındaki çokça bulunan bu robotlar ROS ile beraber çalışabilme özelliğine sahiptir. Bu online veri tabanından Pioneer, iRobot gibi robotlara da erişim imkânımız vardır. Sadece veri tabındaki robotlarla sınırlı kalmayıp, kendi robot ve dünyamızı yaratma imkânı bize tanınmıştır. Eğer istenilirse model.config ve model.sdf dosyalarının içindeki XML kodlarını değiştirerek fiziksel robotlara istenilen sensör ve kamera ekleyebilme şansımız vardır. Oluşturulan bu modeller için işletim sisteminin SDF ve URDF formatlarını desteklemesi gerekmektedir. Diğer simülasyon programlarına nazaran Gazebo, sensör ekleme ve gerçekleşmesi konusunda daha fazla verimlidir. Sersörlerden alınan verilere gürültülerin dâhil edilmesi gerekmektedir. Gerçek dünyada gürültüsüz veriye ulaşılması imkânsızdır. Bu sebepten dolayı simülasyon ortamının gerçek dünyadaki gibi gürültülü veri üretmesi

sağlanarak ideal robot tasarımı gerçekleşmelidir. Gazebonun sensörleri sayesinde gürültülü veri oluşturmak mümkündür. Bütün bu özellikler sebebiyle ve ROS entegrasyonun olmasından dolayı Gazebo simülasyonu yazılım geliştiricileri tarafından daha çok tercih edilen bir simülasyon programı olmuştur.



9.2. Unity

En popüler platformlardan biri Unity Technologies tarafından 2005 yılında geliştirilen Unity 3D oyun geliştirmedir. Platformlar arası oyun tasarlama ve geliştirme yazılımı, kendisini en yüksek standartlara yükseltti. Birçok özelliği bir araya getirir ve hayal edebilebilecek hemen hemen her oyunu yapacak kadar esnek bir araçtır. Bir oyun motoru olarak kullanılan Unity, bir oyunun çalışmasını sağlayan en önemli yerleşik özelliklerin birçoğunu sağlayabilir. Bu 3B oluşturma, fizik, çarpışma ve algılama gibi şeyler anlamına gelir. Unity'nin adında 3D olsa da Unity 3D, 2D oyun geliştirme araçlarını da içermektedir.



Simülasyon videosu: https://youtu.be/8SZyZv1FC1g

10. Sistem Entegrasyonu

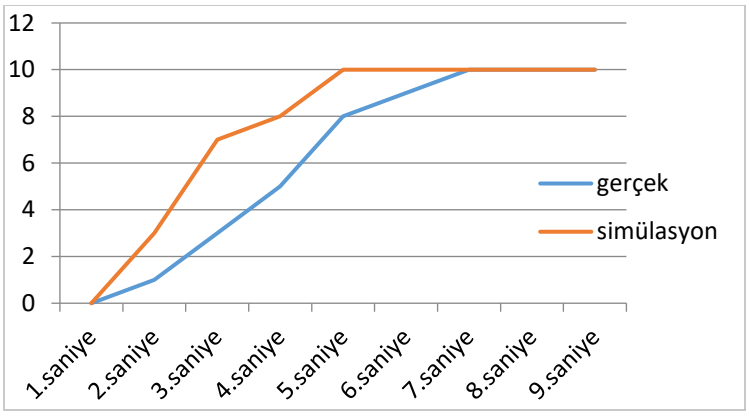
Araçta bulunan Zed2 kamerası ve Velodyne Lidar sistemi Nvidia Jetson TX2 kontrol kartına bağlıdır. Bu kamera ve radar sistemlerinin işleyebilmesi için Nvidia Xavier sisteminden araçla iletişim sağlayabildik. Nvidia Xavier ile aracın hızını, frenlemesini, açısını, levha-şerit algılamasını sağladık. Bu sistem aracın kontrol ünitesine bağlı olup gönderdiğimiz bilgiler aracın sağ ve sol motorlarını, direksiyon motorunu ve fren motorunu hareket ettirir. Sistemli bir şekilde bu hareketlenme devam eder.

Sensör bilgilerinin işlenmesi ve rota-hız bilgilerinin elde edilmesinden sonra bu bilgiler CANBUS aracılığıyla Araç Kontrol Sistemi'ne(AKS) aktarılmaktadır. AKS de bu bilgiler elektronik diferansiyele ve diğer bazı kontrol adımlarına göre işlenip daha sonra ilgili motorlara yine CANBUS üzerinden aktarılmaktadır.

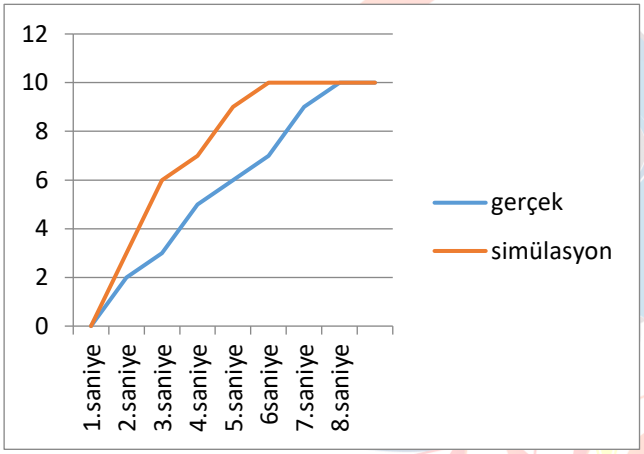
Araçta mevcut olan gaz ve fren pedalı devre dışında yer almaktadır. Bunlarla beraber hız göstergesini de yazılımsal olarak yönetip takip etmekteyiz. Aracın otonom olarak kontrol edilmesi ve yönlendirilmesiyle durumuna göre sensör bilgileriyle Nvidia Jetson TX2 kartı ile aracın hareket etmesi gereken rota bilgileri ve akabinde buna ilişkin açı bilgisi üretilmektedir. Bu açı bilgisi AKS sistemine aktarılmakta ve bu sistemde, kullanılan direksiyon kutusunun özelliğinden kaynaklanan fonksiyona göre tekerlekleri istenilen açıya getirmek için direksiyon milinin ne kadar dönmesi gerektiği hesap edilmekedir. Bu konum bilgisi de direksiyon milini döndürmekle görevli olan servo motora iletilmektedir.

11. Test ve Doğrulama

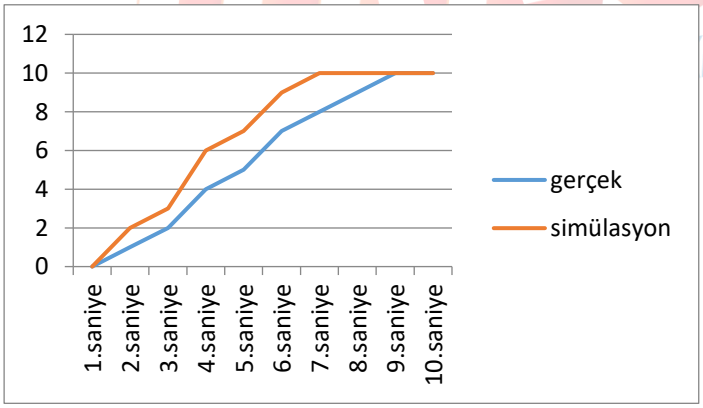
Araç için hazırlamış olduğumuz algoritmalarımızı simülasyon ortamında test etmiş ve beklenen sonuçları elde etmiş bulunmaktayız. Daha sonra teknofest otonom araç test günlerinde, aracımızda : nesne tanıma, Lidar ile otonom sürüş, şerit algılama, park senaryosu, uzaktan acil durum ve fren testlerini yaparak hatalarımızı giderdik. Bu şekilde algoritmalarımızı en sağlıklı çalışır duruma getirdik.
Kameradan alınan veriler ile eğitimini yapmış olduğumuz nesne tespiti algoritmamız ile farklı açılardan tabela okuma testi yapıldı. Yapılan testlerde %90 üzeri değerlerde algılama yapıldığını gözlemledik. Gerçek hava koşulları gereğince gölge ve yansıma gibi faktörler göz önüne alınarak gerekli ayarlamalar yapılmıştır. Eğer ki araç tabela okumayı yapamazsa araç hareketine devam etmek üzere programlanmıştır.
Araç hareket halindeyken önüne çıkan bir engel olduğu durumda kazaya sebebiyet vermemesi için geliştirdiğimiz acil durum fren sistemi testini yaptık ve başarı ile çalıştığını gözlemledik.
Test sürecinde aracın kamerasının algoritmadan veya yoldaki dış etkenlerden (yansıma, gölge düşmesi vb.) kaynaklanan şerit algılamama durumunu Lidar ile giderebileceğimizi tespit ettik. Yani kameranın şeritleri algılamasının yanı sıra Lidar ile şerit dışındaki dubaların algılanmasının aracın alanda sürüşe devam edebilmesine katkı sağladığını gözlemledik.
Şerit takibi testlerinde tespit ettiğimiz diğer önemli nokta ise virajlar. Aracın virajı kusursuz dönebilmesi için tekerleklerin en uygun dönme açısına sahip olması gerekir. Test aşamasında bir çok kez sürüşe çıkıp test ederek en sağlıklı tekerlek dönme açısını bulduk.
Geliştirdiğimiz park senaryomuzun testini gerçekleştirdik ve başarı ile çalıştığını gözlemledik. Aracımız kendine uygun park alnını seçerek düzgün bir şekilde parkını tamamladı.



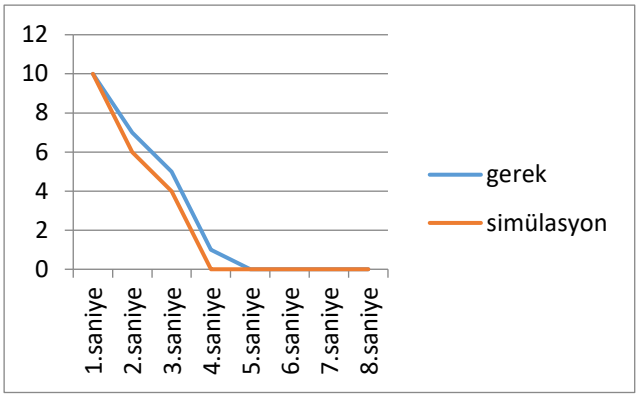
%100 basılınca hızlanma



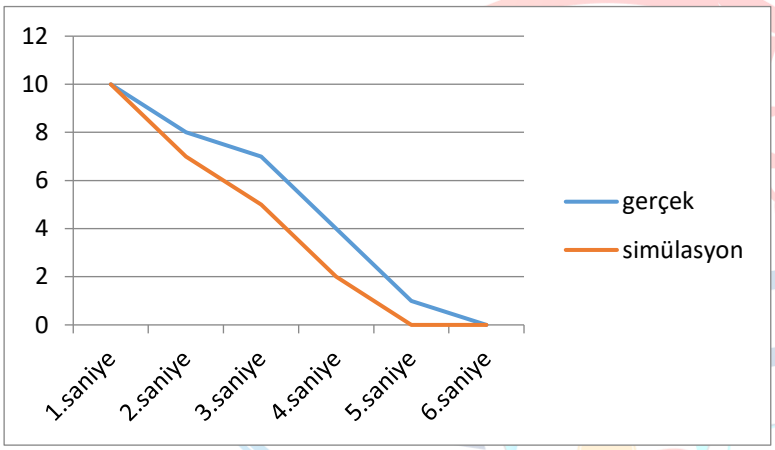
% 50 basılınca hızlanma grafiği



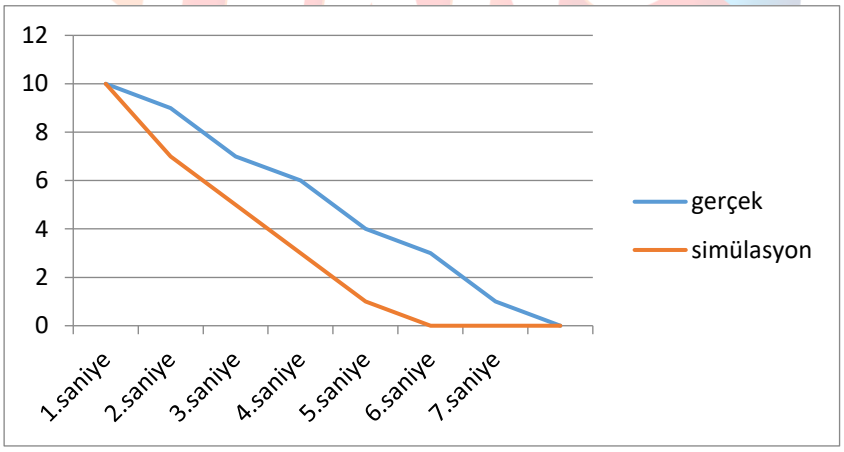
%25 basılınca hızlanma grafiği



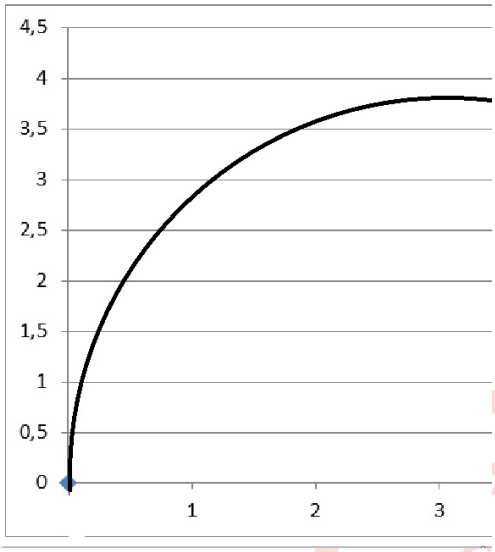
Fren pedalına %100 basılınca aracın yavaşlama grafiği



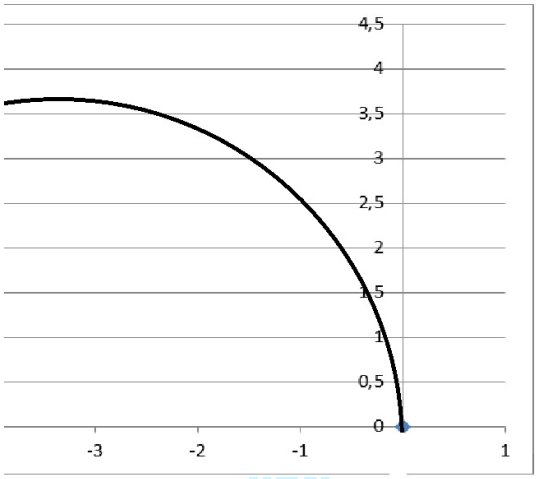
Frene %50 basınca yavaşlama grafiği



Frene %25 basınca yavaşlama grafiği



5km/s 0.3 radyan x y grafiği



5km/s -0.3 radyan x y grafiği

12. Referanslar

1. www.researchgate.net.Seda Kul Konvolusyonel Sinir Ağları adlı makalesi
2. www.wikipedia.com
3. https://medium.com/deep-learning-turkiye/yolo-algoritmasi
4. https://evrimagaci.org/lidar-teknolojisi-nedir-nasil-calisir-nerelerde-kullanilir-10214
5. Taraba, M., Adamec, J., Danko, M., & Drgona, P. (2018, May). Utilization of modern sensors in autonomous vehicles. In 2018 ELEKTRO (pp. 1-5). IEEE.
6. Sae International, "Taxonomy And Definitions For Terms Related To Driving Automation Systems For On-Road Motor Vehicles," 2018. [Online]. Available: Https://Www.Sae.Org/Standards/Content/J3016_201806/. [Accessed: 08-Jul-2019].
7. Martínez-Díaz, M., & Soriguera, F. (2018). Autonomous vehicles: theoretical and practical challenges. Transportation Research Procedia, 33, 275-282.
8. Schumacher, R. W. (1996). Smart Cars: 2000 or 2020? (No. 96C026). SAE Technical Paper.
9.]FOSSEN, T., Pettersen, K. Y., & Nijmeijer, H. (2017). Sensing And Contol For Autonomous Vehicles. Springer International PU.
10. Cheng, H. (2011). Autonomous intelligent vehicles: theory, algorithms, 28 and implementation. Springer Science & Business Media
11. Peker M. (Mayıs 2016). Otonom araçlarda kullanılan görsel seyir sistemleri için yeni bir (Yayınlanmamış Doktora Tezi) Niğde Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı.
12. ÖNAL N. (2020). (Determining The Position and Orientation of an Autonomous Vehicle From The Camera Images) (Yayınlanmamış Yüksek Lisans Tezi) Atatürk Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı.
13. PalamuT.(OCAK-2022)
    Otonom araçlar için gerçek zamanlı nesne tanıma ve uzaklık tahmini sistem tasarım ve uygulaması
    Real-time object recognition and distance estimated system design and implementation for autonomous vehicles (Yayınlanmamış Yüksek Lisans Tezi) Mersin Üniversitesi / Fen Bilimleri Enstitüsü / Bilgisayar Mühendisliği Ana Bilim Dalı
14. Kara I. (İstanbul, 2009) Can Haberleşme Protokolünün İncelenmesi ve Bir Sıcaklık Kontrol Sistemine Uygulanması (Yayınlanmış Yüksek Lisans Tezi) Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü
15. www.siberegitmen.com
16. Eylül Özer ve Burak İnner' in ROS Destekli Otonom Mobil Robot Geliştirilmesi adlı araştırma makalesi
17. https://cdn.teknofest.org/media/upload/userFormUpload/Robotaksi_Hazir_Arac_Bilgilendirme_Belgesi_dhqOM_z5DLP.pdf