

**ELEKTRİKLİ BİR ARAÇ İÇİN ELEKTRONİK
DİFERANSİYEL SİMÜLATÖRÜN
MODELLENMESİ VE GERÇEK ZAMANLI
UYGULANMASI**

YÜKSEK LİSANS TEZİ

İbrahim ÇELİK

DANIŞMAN

Dr. Öğr. Üyesi Güray SONUGÜR

MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI

Haziran 2019

Bu tez çalışması 18.FEN.BIL.52 numaralı proje ile BAPK tarafından desteklenmiştir.

AFYON KOCATEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ELEKTRİKLİ BİR ARAÇ İÇİN ELEKTRONİK DİFERANSİYEL
SİMÜLATÖRÜN MODELLENMESİ
VE GERÇEK ZAMANLI UYGULANMASI

İbrahim ÇELİK

DANIŞMAN
Dr. Öğr. Üyesi Güray SONUGÜR

MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI

Haziran 2019

TEZ ONAY SAYFASI

İbrahim ÇELİK tarafından hazırlanan “Elektrikli Bir Araç İçin Elektronik Diferansiyel Simülatörünün Modellenmesi ve Gerçek Zamanlı Uygulanması” adlı tez çalışması lisansüstü eğitim ve öğretim yönetmeliğinin ilgili maddeleri uyarınca 14/06/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü **Makine Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman :Dr. Öğr. Üyesi Güray SONUGÜR

Başkan : Prof. Dr. Hüseyin BAYRAKÇEKEN
Afyon Kocatepe Üniversitesi, Teknoloji
Fakültesi,

Üye : Dr. Öğr. Üyesi Güray SONUGÜR
Afyon Kocatepe Üniversitesi, Teknoloji Fakültesi,

Üye : Dr. Öğr. Üyesi Barış GÖKÇE
Necmettin Erbakan Üniversitesi, Mühendislik ve
Mimarlık Fakültesi,

İmza

İmza

İmza

İmza

Afyon Kocatepe Üniversitesi
Fen Bilimleri Enstitüsü Yönetim Kurulu’nun
...../...../..... tarih ve
..... sayılı kararıyla onaylanmıştır.

.....
Enstitü Müdürü
Prof. Dr. İbrahim EROL

BİLİMSEL ETİK BİLDİRİM SAYFASI
Afyon Kocatepe Üniversitesi

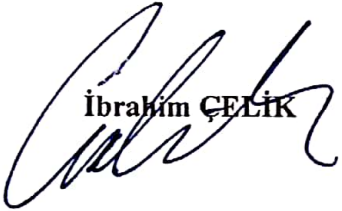
Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- Görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- Başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- Atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

Beyan ederim.

14/06/2019

İbrahim ÇELİK



ÖZET

Yüksek Lisans Tezi

ELEKTRİKLİ BİR ARAÇ İÇİN ELEKTRONİK DİFERANSİYEL
SİMÜLATÖRÜNÜN MODELLENMESİ VE GERÇEK ZAMANLI UYGULANMASI

İbrahim ÇELİK

Afyon Kocatepe Üniversitesi

Fen Bilimleri Enstitüsü

Makine Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Güray SONUGÜR

Ulaşım sektörünü kullanan insan kaynağının gittikçe artmasından dolayı birçok farklı problem meydana gelmektedir. Bu problemlerin önde gelenleri, ulaşımdan oluşan çevre kirliliği ve dünya üzerindeki fosil yakıtların hızlı bir şekilde tükenmesini söyleyebiliriz. Çevreye yayılan zararlı gazları tamamen ortadan kaldırmak, fosil yakıt tüketimini azaltmak ve hava kirliliğine neden olan birkaç maddeyi ortadan kaldırmak amacıyla elektrikli araçlara yönelim başlamıştır. 1930 'larda elektrikli araç yerine içten yanmalı motorlu araçlara geçiş başlamıştı. Çünkü elektrikli araçlar kullanıcı istekleri karşılayamamaktaydı. Fakat günümüzde tasarruflu ve bir o kadar da güçlü elektrik makinalarının bulunmuş olması nedeniyle kullanıcı istekleri karşılayabilmektedir. Bu nedenle elektrikli araçlar piyasaya daha kararlı bir şekilde giriş yapmaktadır. Elektrikli araçlar için sorun teşkil eden durumlar ise toplam menzil ve batarya şarj süresidir. Bu tez çalışmasında toplam menzili artırmaya yönelik bir çalışma yapılmıştır. Elektrikli aracın toplam menzili, araç ağırlığı ile ters orantılıdır. Tek motorlu bir elektrik aracının ağırlığı azaltmak için mekanik diferansiyel çıkarılıp matematiksel modellemesi yapılmıştır. Bu matematiksel model yardımıyla iki elektrik motoruna mekanik diferansiyelin yaptığı iş yaptırılmaya çalışılmıştır. Öncelikli olarak sistem MATLAB/Simulink ortamında kontrol edildikten sonra gerçekleştirilmesi yapılarak sistemin stabil çalışması doğrulanmıştır.

2019, xi + 104 sayfa

Anahtar Kelimeler: Elektronik diferansiyel, Elektrikli araç, Simulink, Modelleme

ABSTRACT

M.Sc. Thesis

SIMULATION MODELLING AND REAL-TIME APPLICATION OF ELECTRONIC DIFFERENTIAL FOR AN ELECTRICAL VEHICLE

İbrahim ÇELİK

Afyon Kocatepe University

Graduate School of Natural and Applied Sciences

Department of Mechanical Engineering

Supervisor: Asst. Prof. Güray SONUGÜR

Many different problems are occurred due to the increasing numbers of person using the transportation sector. The important of these problems are the rapid consumption of fossil fuels on earth and environmental pollution occurred transportation. Electric vehicles are preferred in order to eliminate the harmful gases emitted to the environment, to reduce fossil fuel consumption and to eliminate a few reasons that cause air pollution. In the 1930s, internal combustion engine vehicles were preferred instead of electric vehicles, so the electric vehicles were not used for a long time. Because electric vehicles could not meet user requests. However, due to the high efficiency of electrical machines and developments in battery technology, it can meet the demands of the user. For this reason, electric vehicles are decidedly entering the market. The problems for electric vehicles is the total range and battery charge time. In this thesis, a study has been conducted to increase the total range. The total range of the electric vehicle is inversely proportional to the vehicle weight. In order to reduce the weight of a single-motor electric vehicle, the mechanical differential is removed and created mathematically model of it. With the help of this mathematical model, it has been tried to perform the task of mechanical differential by two electric motors. Initially, after the system was controlled in MATLAB / Simulink, it is observed that the system stably operates.

2019, xi + 104 pages

Keywords: Electronic differential, Electric vehicle, Simulink, Modeling

TEŐEKKÜR

Tüm alıŐma hayatım ve öncesinde sürekli yanımda olan ve maddi-manevi desteklerini hiç bir zaman esirgemeyen başta sevgili eşim Sultan MURAT ELİK olmak üzere Nurettin ELİK, Nurcan ELİK ve Abubekir ELİK' e sonsuz teşekkürlerimi sunarım.

Bu araştırma boyunca değerli görüş ve yardımlarıyla beni yönlendiren ve değerli bilgilerini paylaşan danışman hocam Dr. Öğr. Üyesi Güray SONUGÜR' e en içten teşekkürlerimi sunarım.

Değerli görüş ve geri dönüşlerini hiçbir zaman esirgemeyen, yazılım konusunda verdiği büyük destekle her zaman yanımda olan Dr. Öğr. Üyesi Barış GÖKÇE' ye, mekanik kısımlar konusunda Prof. Dr. Hüseyin BAYRAKÇEKEN ve Dr. Öğr. Üyesi İbrahim YAVUZ hocalarıma teşekkür eder, şükranlarımı sunarım.

Tezimin gerçekleşmesinde 18.FEN.BIL.52 numaralı proje ile maddi destek sağlayan BAPK' na teşekkür ederim.

İbrahim ELİK
AFYONKARAHİSAR, 2019

İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER DİZİNİ.....	iv
KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ.....	x
RESİMLER DİZİNİ	xi
1. GİRİŞ.....	1
1.1 Elektrikli Araç Teknolojileri	2
1.1.1 Hibrit Elektrikli Araçlar	2
1.1.1.1 Paralel Hibrit Elektrikli Araçlar	3
1.1.1.2 Seri Hibrit Elektrikli Araçlar	3
1.1.1.3 Karma Hibrit Elektrikli Araçlar	4
1.1.1 Tamamen (%100) Elektrikli Araçlar	5
1.2 Tez İçeriği	8
2. LİTERATÜR BİLGİLERİ	9
2.1 İki Tekerlekten Tahrikli Araçlar	9
2.2 Dört Tekerlekten Tahrikli Araçlar	11
3. MATERYAL ve METOT	13
3.1 Araç Mekanik Sistemi.....	13
3.2 Elektronik Diferansiyel Sistemi	19
3.2.1 Bilgisayar Ortamında Matematiksel Modellenmesi	22
3.2.1.1 Elektronik Diferansiyel Modeli	22
3.2.1.2 PID Kontrolör Tasarımı.....	24
3.2.1.3 Doğru Akım Elektrik Motoru Modellenmesi	25
3.2.1.4 Elektrikli Araç Modeli.....	29
3.2.2 Elektronik Diferansiyel Sistem Simülatörünün Oluşturulması	32
3.2.2.1 Mikroişlemci.....	32
3.2.2.2 Enkoder.....	35

3.2.2.3 Simülâtör	38
3.2.3 Elektronik Diferansiyel Sisteminin Araç Üzerine Uygulanması	39
3.2.3.1 Eclipse Programı	40
3.2.3.2 Sistem Tasarımı	42
4. BULGULAR	45
4.1 Matematiksel Modelleme Bulguları.....	45
4.2 Gerçek Zamanlı Uygulama Bulguları	46
4.2.1 Asfalt Zemin	47
4.2.1.1 Asfalt zeminde 5° sağı yönelim için alınan veriler	47
4.2.1.2 Asfalt zeminde 15° sağı yönelim için alınan veriler	49
4.2.1.3 Asfalt zeminde 25° sağı yönelim için alınan veriler	51
4.2.2 Islak Zemin	53
4.2.2.1 Islak zeminde 5° sağı yönelim için alınan veriler	54
4.2.2.2 Islak zeminde 15° sağı yönelim için alınan veriler	55
4.2.2.1 Islak zeminde 25° sağı yönelim için alınan veriler	57
4.2.3 Zemin 1 Şartlarında.....	59
4.2.3.1 Zemin 1 Şartlarında 5° sağı yönelim için alınan veriler.....	60
4.2.3.2 Zemin 1 Şartlarında 15° sağı yönelim için alınan veriler.....	61
4.2.3.3 Zemin 1 Şartlarında 25° sağı yönelim için alınan veriler.....	63
4.2.4 Zemin 2 Şartlarında.....	65
4.2.4.1 Zemin 2 Şartlarında 5° sağı yönelim için alınan veriler.....	66
4.2.4.2 Zemin 2 Şartlarında 15° sağı yönelim için alınan veriler.....	67
4.2.4.3 Zemin 2 Şartlarında 25° sağı yönelim için alınan veriler.....	69
5. TARTIŞMA ve SONUÇ	72
6. KAYNAKLAR.....	75
ÖZGEÇMİŞ.....	80

KISALTMALAR DİZİNİ

Kısaltmalar

AJP	Ackermann – Jeantand Prensibi
ARM	Acorn RISC Machine
EDS	Elektronik Diferansiyel Sistemi
EMA	Elektrik Motorlu Araçlar
HEA	Hibrit Elektrikli Araç
İYM	İçten Yanmalı Motor
PID	Proportional-Integral-Derivative
PMSM	Permanent Magnet Synchronous Motor
PWM	Pulse Width Modulation
SI	Système International d'Unités
VGA	Video Graphics Array

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1. Paralel hibrit araç sistemi.	3
Şekil 1.2 Seri hibrit araç sistemi.....	4
Şekil 1.3 Seri-paralel hibrit araç sistemi.	4
Şekil 3.1 İçten yanmalı motorlu bir aracın temel yapısı (Bacon 1972).....	13
Şekil 3.2 İçten yanmalı motora sahip Porsche 911 modelinin motor tork ve güç değerleri (İnt.Kyn.7).....	14
Şekil 3.3 Elektrik motorlarının hız-tork grafiği (Quriroga <i>et al.</i> 2015).	15
Şekil 3.4 Tek elektrik motorlu elektrikli aracın basit yapısı.	15
Şekil 3.5 İki elektrik motorlu elektrikli aracın basit yapısı.	16
Şekil 3.6 Ackermann diferansiyel modeli (Nejh and Trabelsi 2015).....	19
Şekil 3.7 Elektronik diferansiyel bloğu.	22
Şekil 3.8 Oluşturulan elektronik diferansiyel bloğu kullanıcı ara yüzü.	23
Şekil 3.9 Oluşturulan elektronik diferansiyel bloğu içeriği.	23
Şekil 3.10 Araç hızınının 20 km/h olduğu durum için arka sağ ve sol tekerleklerin hızlarının(dev/dk) model çıktısı.	24
Şekil 3.11 Geri beslemeli bir kontrol sistem yapısı.....	24
Şekil 3.12 PID kontrolörünün içyapısı.	25
Şekil 3.13 Doğru akım motoru elektriks eşdeğer devresi (Bharatiraja <i>et al.</i> 2016). ...	26
Şekil 3.14 Oluşturulan doğru akım motoru model ara yüzü.	28
Şekil 3.15 Oluşturulan doğru akım motoru model bloğu.	29
Şekil 3.16 DC motor matematiksel modeli.	29
Şekil 3.17 Tasarlanan elektrikli araç model ara yüzü.	30
Şekil 3.18 Oluşturulan elektrikli araç modeli.....	30
Şekil 3.19 Oluşturulan sürücü ve kontrolör modeli.	31
Şekil 3.20 Oluşturulan araç modeli.	31
Şekil 3.21 Mikroişlemci basit yapısı.	33
Şekil 3.22 ARM mirarisine sahip STM32F407VG mikroişlemcisinin C portunun 11. pininin alabileceği görevler.....	34
Şekil 3.23 Araç üzerinde kullanılan mikroişlemci için kullanılan pinler.....	35
Şekil 3.24 Optik teknolojisi kullanılan artımlı bir döner enkoderin teknik yapısı (İnt.Kyn.10).....	35

Şekil 3.25 Üç farklı şekilde kodlanmış mutlak enkoder kod diski.....	36
Şekil 3.26 Artımsal enkoder kod diski.	37
Şekil 3.27 Devre alt (a) ve üst (b) katmanı.....	39
Şekil 3.28 Proje oluştururken mikroişlemci seçme ara yüzü.	40
Şekil 3.29 Proje oluşturulduktan sonra açılan boş sayfa.	41
Şekil 3.30 Mikroişlemci pinlerinin görev belirlendiği kod kısmı.	41
Şekil 3.31 Araç için kullanılan devre çizim (a) ve gerçekleştirilmesi (b).	42
Şekil 3.32 EDS'in temel akış şeması.....	43
Şekil 4.1 Sol arka tekerleklerin referans hız ve model hızı.....	45
Şekil 4.2 Sağ arka tekerleklerin referans hız ve model hızı.	46
Şekil 4.3 Asfalt zeminde 5° sağa yönelimde sol tekerlek hız ve akım grafiği.	47
Şekil 4.4 Asfalt zeminde 5° sağa yönelimde sağ tekerlek hız ve akım grafiği.....	48
Şekil 4.5 Asfalt zeminde 15° sağa yönelimde sol tekerlek hız ve akım grafiği.	50
Şekil 4.6 Asfalt zeminde 15° sağa yönelimde sağ tekerlek hız ve akım grafiği.....	50
Şekil 4.7 Asfalt zeminde 25° sağa yönelimde sol tekerlek hız ve akım grafiği.	52
Şekil 4.8 Asfalt zeminde 25° sağa yönelimde sağ tekerlek hız ve akım grafiği.....	52
Şekil 4.9 Islak zeminde 5° sağa yönelimde sol tekerlek hız ve akım grafiği.	54
Şekil 4.10 Islak zeminde 5° sağa yönelimde sağ tekerlek hız ve akım grafiği.....	54
Şekil 4.11 Islak zeminde 15° sağa yönelimde sol tekerlek hız ve akım grafiği.	56
Şekil 4.12 Islak zeminde 15° sağa yönelimde sağ tekerlek hız ve akım grafiği.....	56
Şekil 4.13 Islak zeminde 25° sağa yönelimde sol tekerlek hız ve akım grafiği.	58
Şekil 4.14 Islak zeminde 25° sağa yönelimde sağ tekerlek hız ve akım grafiği.....	58
Şekil 4.15 Zemin 1 şartlarında 5° sağa yönelimde sol tekerlek hız ve akım grafiği.	60
Şekil 4.16 Zemin 1 şartlarında 5° sağa yönelimde sağ tekerlek hız ve akım grafiği.	60
Şekil 4.17 Zemin 1 şartlarında 15° sağa yönelimde sol tekerlek hız ve akım grafiği. ...	62
Şekil 4.18 Zemin 1 şartlarında 15° sağa yönelimde sağ tekerlek hız ve akım grafiği. ..	62
Şekil 4.19 Zemin 1 şartlarında 25° sağa yönelimde sol tekerlek hız ve akım grafiği. ...	64
Şekil 4.20 Zemin 1 şartlarında 25° sağa yönelimde sağ tekerlek hız ve akım grafiği. ..	64
Şekil 4.21 Zemin 2 şartlarında 5° sağa yönelimde sol tekerlek hız ve akım grafiği.	66
Şekil 4.22 Zemin 2 şartlarında 5° sağa yönelimde sağ tekerlek hız ve akım grafiği.	66
Şekil 4.23 Zemin 2 şartlarında 15° sağa yönelimde sol tekerlek hız ve akım grafiği. ...	68
Şekil 4.24 Zemin 2 şartlarında 15° sağa yönelimde sağ tekerlek hız ve akım grafiği. ..	68

Şekil 4.25 Zemin 2 şartlarında 25° sağı yönelimde sol tekerlek hız ve akım grafiğı. ...	70
Şekil 4.26 Zemin 2 şartlarında 25° sağı yönelimde sağı tekerlek hız ve akım grafiğı. ..	70
Şekil 5.1 Farklı zeminlerde 5° direksiyon açısıyla alınan akım verileri grafiğı.	72
Şekil 5.2 Farklı zeminlerde 15° direksiyon açısıyla alınan akım verileri grafiğı.	73
Şekil 5.3 Farklı zeminlerde 25° direksiyon açısıyla alınan akım verileri grafiğı.	73

ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1 Porsche 911 vites dönüşüm oranları (İnt.Kyn.8).	14
Çizelge 3.2 Doğru akım motoru parametreleri.....	28
Çizelge 4.1 Elektrikli araç prototipinin test edildiği yol şartlarının yaklaşık sürtünme katsayıları.....	46
Çizelge 4.2 Sol ve sağ tekerleklerin asfalt zeminde 5° sağa yönelimdeki gerçek zamanlı verileri.....	48
Çizelge 4.3 Sol ve sağ tekerleklerin asfalt zeminde 15° sağa yönelimdeki gerçek zamanlı verileri.....	51
Çizelge 4.4 Sol ve sağ tekerleklerin asfalt zeminde 25° sağa yönelimdeki gerçek zamanlı verileri.....	53
Çizelge 4.5 Sol ve sağ tekerleklerin ıslak zeminde 5° sağa yönelimdeki gerçek zamanlı verileri.....	55
Çizelge 4.6 Sol ve sağ tekerleklerin ıslak zeminde 15° sağa yönelimdeki gerçek zamanlı verileri.....	57
Çizelge 4.7 Sol ve sağ tekerleklerin ıslak zeminde 25° sağa yönelimdeki gerçek zamanlı verileri.....	59
Çizelge 4.8 Sol ve sağ tekerleklerin zemin 1 şartlarında 5° sağa yönelimdeki gerçek zamanlı verileri.	61
Çizelge 4.9 Sol ve sağ tekerleklerin zemin 1 şartlarında 15° sağa yönelimdeki gerçek zamanlı verileri.	63
Çizelge 4.10 Sol ve sağ tekerleklerin zemin 1 şartlarında 25° sağa yönelimdeki gerçek zamanlı verileri.	65
Çizelge 4.11 Sol ve sağ tekerleklerin zemin 2 şartlarında 5° sağa yönelimdeki gerçek zamanlı verileri.	67
Çizelge 4.12 Sol ve sağ tekerleklerin zemin 2 şartlarında 15° sağa yönelimdeki gerçek zamanlı verileri.	69
Çizelge 4.13 Sol ve sağ tekerleklerin zemin 2 şartlarında 25° sağa yönelimdeki gerçek zamanlı verileri.	71

RESİMLER DİZİNİ

	Sayfa
Resim 1.1 İzsiz mekanik-elektrikli trolleybüs.	2
Resim 1.2 II. Abdülhamid 'e ait 1888 yapımı elektrikli araç (İnt.Kyn.1).....	5
Resim 1.3 1904 yılında Alman yapımı elektrikli araç (a) (İnt.Kyn.2), 1913 yılında Thomas Edison ve Baker firmasının elektrikli aracı (b) (İnt.Kyn.3).....	6
Resim 1.4 1910 yapımı mekanik kollu Ford Touring aracı (İnt.Kyn.4).	6
Resim 1.5 2019 yapımı Tesla Model 3 elektrikli araç (İnt.Kyn.5).	7
Resim 3.1 Araç üzerinde kullanılan DC elektrik motoru.....	17
Resim 3.2 Araç üzerinde kullanılan tekerlek.	17
Resim 3.3 Oluşturulan araçtaki tahrik sistemleri.	18
Resim 3.4 İki elektrik motorlu tez kapsamında geliştirilen elektrikli araç prototipi.	18
Resim 3.5 Tez kapsamında oluşturulan elektrikli araç.	21
Resim 3.6 Çalışma kapsamında geliştirilen simülatör ekran görüntüsü.	32
Resim 3.7 Direksiyon çizim (a) ve 3D yazıcı çıktısı (b).	38
Resim 3.8 Devre kartının son hali.	39
Resim 3.9 Dual DC motor 100A sürücü kartı.	42

1.GİRİŞ

Son yıllarda petrol kaynaklarının yoksunluđuna çekilen ilgi ve içten yanmalı motorların doğaya verdiği zararlar gündeme geldikçe hem doğanın temizliđi hem de petrol kaynaklarının tüketiminin azalması için çođunlukla farklı kaynaklar(elektrik, hidrojen, bor vs. gibi) tüketen araçların üretimine yönlenmeye başlanılmıştır. Bu kaynakların başında, kolay ulaşılabilirliđi ve kullanılabilirliđi ile elektrik yer almaktadır. Bu durumda elektrikli araçların gelişmesini tetiklemiştir.

Elektrik motorlu araçlar (EMA), içten yanmalı motorlu araçlarla kıyaslandığında kontrol açısından daha kompleks yapıda olsalar da temizlik, bakım ve onarım gibi başlıca ihtiyaçlarda çok daha rahat bir kullanım sunmaktadırlar. Bu bakımlar hem mali hem de vakit açısından oldukça sıkıntı içerebilmektedir. Fakat EMA'lar da uzun periyotlar sonrasında uzman kişiler tarafından bakım yapıldığında eđer hasarlı bir kısım yok ise temizlenerek bakım yapılması olasıdır.

EMA 'lar birkaç farklı türde meydana getirilebilir; İçten yanmalı motorlu bir araç üzerinde

- Sadece tahrik sistemi deđiştirilerek,
- Tahrik sisteminin devamında bulunan aktarım organları da(vites kutusu, kardan mili ve diferansiyel gibi) çıkarılarak

elde edilebilir. İkinci seçenek kontrol açısından daha zorlu algoritmalara neden olsa da araç üzerinde ađırlığı artıracak vites kutusu, kardan mili ve diferansiyel kutusu gibi çođu parçadan aracın kurtulmasına da neden olacaktır. Bu nedenle hem araç ađırlığı bakımından elektrikli aracın daha hafif olurken hem de aynı batarya ile daha uzun mesafeler kat etmesine imkân sağlayacaktır. Çıkarılan bu parçaların yaptığı işler iyice saptanıp bu işlerin yazılımsal olarak yapılması gerekmektedir. Bu tez çalışmasında çıkarılan mekanik diferansiyel sisteminin yaptığı işler tamamıyla saptanıp modellenmiş ve elektrik motorlarının bu senaryoya uyması sağlanmıştır. Diđer bir deyişle Ackerman modeli yazılımsal olarak ifade edilmiştir.

1.1 Elektrikli Araç Teknolojileri

Elektrikli araçlar birkaç başlık altında incelenebilir fakat bunların en önemlileri hibrit elektrikli araçlar ve %100 Elektrikli araçlardır.

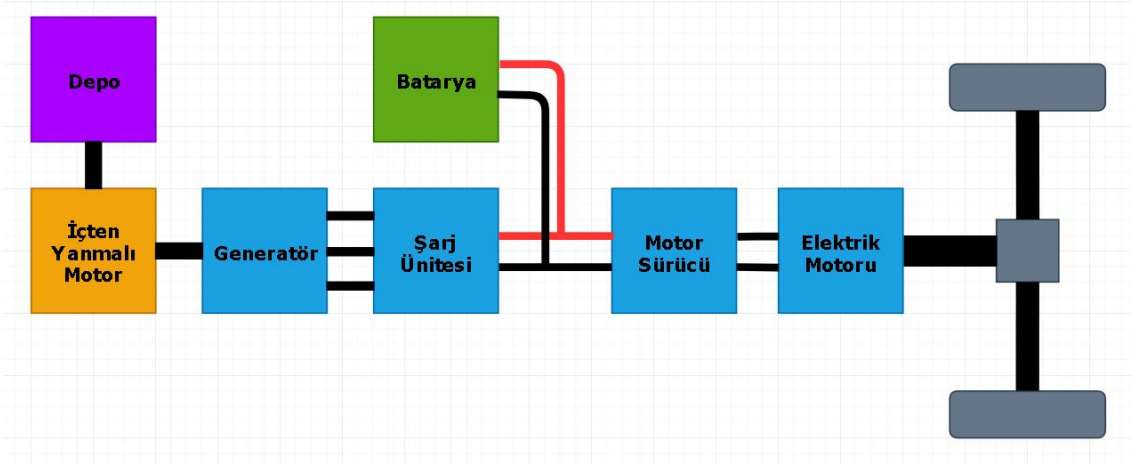
1.1.1 Hibrit Elektrikli Araçlar

Hibrit elektrikli araçlar(HEA), çok sayıda tahrik elemanına sahiptir. Dizel-elektrikli araçlar bu sınıfa dâhil edilmezler çünkü elektriksiz tahrik sistemi ek bir hareket sağlayıcısı değil de doğrudan tahrik için kullanılmıştır. Hibrit kara araçlarının en eski türlerinden biri 1933 'de Sovyet Sosyalist Cumhuriyetleri Birliği tarafından Moskova 'da kullanılmak üzere üretilen izsiz mekanik-elektrikli trolleybüslere dir. İzsiz mekanik trolleybüs resim 1.1' de gösterilmiştir. Trolleybüslere, ya doğrudan tahrik elemanı olarak ya da bağımsız olarak elektrik enerjisini üretebilmek için genellikle İçten Yanmalı Motor(İYM) barındırılırdı. Bu tarz bir güç aktarma sistemi, trolleybüsün yukarısındaki kablolar ve engeller etrafında manevra yapmasına olanak sağlamıştır. Güç aktarma sistemleri, kimyasal, güneş, nükleer enerji gibi kaynakları doğrudan kullanabilir ya da onları tahrik elemanı için uygun hale getirebilir. Hibrit güç aktarma organları birçok farklı yapıda meydana gelebilir.



Resim 1.1 İzsiz mekanik-elektrikli trolleybüs.

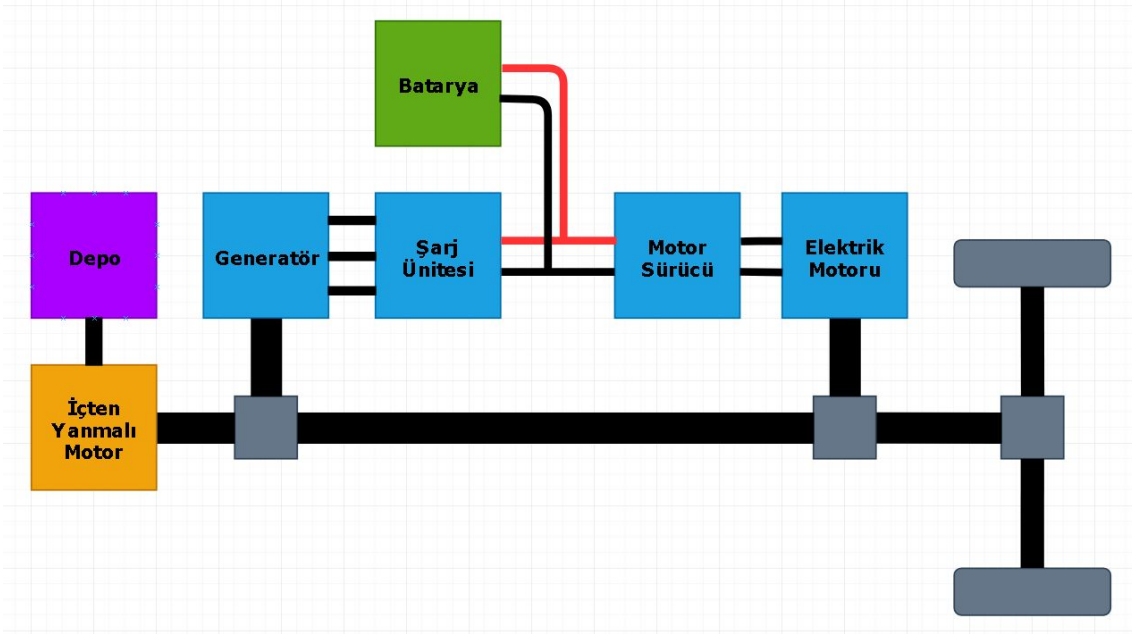
HEA 'lar çalışma algoritmalarına göre 3 farklı kısımda incelenebilir. Bunlar; paralel, seri ve karma hibrit elektrikli araçlardır.



Şekil 1.2 Seri hibrit araç sistemi.

1.1.1.3 Karma Hibrit Elektrikli Araçlar

İYM 'lardan mekanik veya elektrikli tahrik edilebilen tekerleklere giden güç aktarım sistemlerine izin veren, güç ayırma sistemler içeren bir hibrit elektrikli araç sistemidir. Seri paralel hibrit elektrikli araç sisteminin basit yapısı şekil 3.3' de gösterilmiştir. Ana prensip sürücü tarafından talep edilen gücü birincil kaynaktan sağlanan güçten ayırmaktır.



Şekil 1.3 Seri-paralel hibrit araç sistemi.

İYM 'ler düşük devirlerde aracın hareket için talep ettiği gücü sağlayamamaktadırlar dolayısıyla ilk hızlanma esnasında düşük devirlerde ve hatta durgun halde gerekli tork ihtiyacını karşılayabilecek bir motora ihtiyaç duyulmaktadır. Elektrik motorları ise bu prensibe tamamen uymaktadırlar. Seri-Paralel hibrit sistemlerde bu soruna ihtiyaç duyulduğundan ortaya çıkmıştır. Düşük devirlerde elektrik motoru araç tahrikini sağlarken yüksek hızlara çıkıldığında İYM 'a bağlı iletim organları aktif olup aracın talep ettiği güç İYM 'den sağlanır.

1.1.1 Tamamen (%100) Elektrikli Araçlar

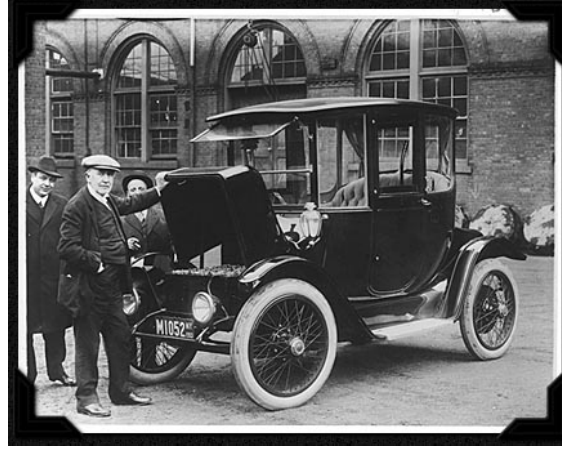
Tamamen elektrik tüketimiyle çalışan çevre dostu elektrikli araçlar düşündüğümüzden çok daha eskidirler. Türkiye 'de ilk elektrikli araç 1888 yılında İngiliz menşei bir firmadan II. Abdülhamid tarafından sipariş edilmiştir. II. Abdülhamid tarafından siparişi verilen elektrikli araç resim 3.2'de gösterilmiştir. Sipariş edilen 1 beygir gücündeki araç, sultan tarafından oldukça beğenildiğinden dolayı tasarımcı mühendisler ödüllendirilmiş ve dünyaca ünlenmişlerdir.



Resim 1.2 II. Abdülhamid 'e ait 1888 yapımı elektrikli araç (İnt.Kyn.1).

Elektrikli araçlar 19. yy 'ın ortalarından 20. yy 'ın başlarına kadar oldukça benimsenmişlerdi. 1904 yılı Alman yapımı elektrikli araç, aşağıda resim 1.3-a' da, 1913

yılında yapılmış Thomas Edison ve Baker firmasının elektrikli aracı ise resim 1.3-b’ de gösterilmiştir. Bunun sebebi ise o yıllarda elektrikli araçlar kullanım kolaylığı ve rahatlık açısından İYM ‘lu araçlara göre daha üstün olmasıydı.



(a)

(b)

Resim 1.3 1904 yılında Alman yapımı elektrikli araç (a) (İnt.Kyn.2), 1913 yılında Thomas Edison ve Baker firmasının elektrikli aracı (b) (İnt.Kyn.3).

Elektrikli araçların en büyük tercih sebeplerinden birisi; İYM ‘lu bir aracı çalıştırmak için araç önündeki mekanik kolun çevrilmesi gerekliydi. 1910 yapımı mekanik kollu Ford Touring aracı resim 1.4’ de gösterilmiştir.



Resim 1.4 1910 yapımı mekanik kollu Ford Touring aracı (İnt.Kyn.4).

Bu olay özellikle bayanlar tarafından oldukça zor oluyordu ve elektrikli araç tercih ediliyordu. Daha sonra ki yıllarda İYM ‘lu araçlardaki devrim niteliğindeki gelişmelerin kanımca en büyüğü araçları çalıştırmak için kullanılan mekanik kolun ortadan kaldırılıp bunun yerine bir elektrik motoru bırakılmasıdır. Bu gelişim ve akabinde meydana gelen Ford Motors gibi büyük şirketlerin seri olarak petrol yakıtlı araçlar üretmesi, petrol yakıtlı araçların elektrikli araçlarla neredeyse aynı fiyata kadar düşmesine hatta ve hatta ucuz olmasına neden olmuştur. Bu olayların ardından 1920 ‘lerin sonlarına doğru çoğu ülke piyasasından elektrikli araçlar silinmeye başlamıştır.

Günümüzde ise petrol tabanlı yakıtların çevreye verdikleri zararlar, güç elektroniğindeki devasa sayılabilecek gelişmelerin ardından elektrik motor sürücülerinin yüksek verimde çalışması ve batarya teknolojisindeki büyük ilerlemelerden sonra tekrar elektrikli araçlara geri dönüş başlamıştır. 2019 yapımı Tesla Model 3 elektrikli araç resim 1.5’ da gösterilmiştir.



Resim 1.5 2019 yapımı Tesla Model 3 elektrikli araç (İnt.Kyn.5).

Elektrikli araçla olan geri dönüş gelişimlere de yola açmış oldu. Standart tip içten yanmalı motorlu bir aracın motorunun eksi yönlerinin giderilmesi için bir elektrik motoru kullanılan hibrit sistemler ortaya çıkmıştır (Lam and Louey 2006, Moreno *et al.* 2006). Bu sistemlerin, içten yanmalı motorlu araçlara göre hem emisyon değerleri hem de kullanım maliyetleri düşüktür (Sioshansi and Denholm 2009, Gao 2011, Hall and Lutsey

2018).

Bu sürecin ardından araçlar artık tamamıyla elektrik enerjisi ile hareket eder olmuş ve bu sayede doğaya saldıkları CO₂ gazının miktarı sıfıra inmiştir. Bunun yanı sıra ulaşım masraflarını da oldukça düşmüştür (Vliet *et al.* 2011, İnt.Kyn.6).

1.2 Tez İçeriği

Tezin birinci bölümünde, içten yanmalı araçlardan, elektrikli araçlara geçiş ve bu bağlamda yapılabilecek unsurlara kısaca değinilmiştir. Elektronik diferansiyel kavramı kısaca açıklanmıştır.

İkinci kısımda, konuyla ilgili literatürde yapılan çalışmalar detaylı bir biçimde verilmiştir. Literatür taraması, iki tekerlekten ve dört tekerlekten tahrikli araçlar başlıklarına alınarak daha detaylı şekilde incelenmiştir.

Üçüncü kısımda, 1930' lardan günümüze kadar olan elektrikli araçların tarihi anlatılmıştır. Paralel hibrit, Seri hibrit gibi elektrikli araç teknolojiler ve tamamen elektrikli araçlar açıklanmıştır. Elektronik diferansiyel sisteminin bilgisayar ortamında modellenmesi açıklanmıştır. PID kontrol, ARM mikroişlemci gibi yapılar açıklanmıştır. Sistemin Simulink modelinin ardından araç üzerine uygulanması detaylı şekilde anlatılmıştır.

Dördüncü kısımda, Elde edilen sistemin modelleme ve gerçek sistem verileri açıklanarak verilmiştir.

Beşinci kısımda, tezin sonucunda elde edilen veriler değerlendirilerek ve gelecek çalışmalarda yapılabilecek çalışmalar hakkında bilgiler ve öneriler verilmiştir.

2. LİTERATÜR BİLGİLERİ

Elektronik diferansiyel sistemi üzerinde farklı bakış açılarına sahip birkaç literatür çalışması bulunmaktadır. Literatür bilgileri iki temel başlık altında toplanmıştır. Bunlar; iki tekerlekten tahrikli elektrikli araçlar ve dört tekerlekten tahrikli elektrikli araçlardır.

2.1 İki Tekerlekten Tahrikli Araçlar

Codesys yazılım paketi kullanılarak, AJP (Ackermann – Jeantand Prensibi) den yararlanılarak elde edilen model yardımıyla ön tekerlek denklemleri çıkartılmıştır. Taşıt hızı ve Elektrikli taşıtın direksiyon açısındaki değişim baz alınarak modelden elde edilen hızlarla simülasyon sonuçları doğrulanmıştır (Draou 2013). Direksiyon açısı 1° ‘den 15° ‘ye kadar değiştirilerek model ve Codesys programı eşleşmiş ve sistemin çalışmasının uygunluğu görülmüştür (Yıldırım vd. 2016).

FPGA (Field Programmable Gate Array) tabanlı bir kontrol algoritmasıyla, parametrik tanıma ve AC (Alternative Current) Motor kullanılarak EDS ‘ye farklı bir yaklaşım getirilmiştir (De Castro *et al.* 2007).

Mikroişlemci olarak TMS320F2812 kullanılarak iki adet indüksiyon motorunun bağımsız halde kontrolü sağlanmıştır. Çalışmadaki indüksiyon motorları 28 V_{rms} değerinde çalışabilecek şekilde yeniden sarılmıştır. Uygulanan eşit tork diferansiyel kontrolü, aracın düz ileri ve dönme yörüngelerinde, tekerleklerin hiçbirinin şasiyi zorlamadan hareket ettiğini gözlenmiştir (Magallán *et al.* 2008). Yapılan diğer bir çalışmada hem elektrik diferansiyel hem de asenkron motorların kontrolünü içeren tek bir yonga geliştirilmiştir. Deneysel sonuçlar, NDEC (the New European Driving Cycle) test döngüsünü kullanarak, güç aktarım organındaki iki adet 0.9kW’ lık endüksiyon motorları ile prototip çalışmasını yapılmıştır (Haddoun *et al.* 2010).

İki adet fırçasız DC motor kullanılarak torkun eşit şekilde bölünmesi Arduino kontrolör ile sağlanmıştır (Clavero-Ordóñez *et al.* 2018). Yapılan diğer bir çalışmada, 4 anahtarlı 3 fazlı bir invertör için bir elektronik diferansiyel ve kayar mod kontrol cihazı tasarımı

önerilmiştir. Tahrik sistemi, iki ayrı akü dizisinden çalıştırılan ve iki bağımsız arka tekerlekten çekiş motorunu doğrudan kontrol eden, iki invertör içeren bir test aracı üzerinde gerçekleştirilmiştir. Sonuçlar, hem elektronik diferansiyel hem de kayar mod kontrolörünün tatmin edici bir şekilde çalıştığını göstermiştir (Gair *et al.* 2004).

Bir araç kavisli yol şeridi boyunca hareket ettiğinde, aracın titreşmesini önlemek için iç tekerleğin hızı dış tekerleğin hızından farklı olmalıdır. Bu elektrikli aracın her tekerleğinin bağımsız bir sürüş kuvveti olduğu için bir elektrik diferansiyel sistemi gerekir. Ancak, diferansiyel sistemin doğrusal olmayan davranışlarını araç hızı ve direksiyon açısı ile araç yapısı açısından analiz etmek zor olmuştur. Bu nedenle, ilişkileri öğrenmek için bir sinir ağı kullanılmıştır. Yapay sinir ağı modelini gerçekleştirmek için çeşitli hızlarda ve direksiyon açılarında deneysel bir elektrikli araç kullanılarak, iç ve dış tekerlek için hız verileri elde edilmiştir. Bu bilgilerle, diferansiyel sistem doğrusal olmayan ilişkilerin bir sinir ağı modeli kullanılarak kontrol edilebilir hale gelmiştir. Performansı analiz edilmiş ve doğrusal olmayan özelliklere sahip olduğu saptanmıştır. Elektrikli araç tahrikçilerinin kontrolünde daha fazla sürüş konforu ve güvenliği sağlamak için bu sistemin faydalı olacağı beklenmektedir (Ju-Sang *et al.* 2000).

Araç dengesini sağlamak amacıyla arkadan itişli elektrikli araç için EDS tasarlanmıştır. Sistemde her bir tekerleğe doğrudan bağlı motor için tork kontrolüne dayalı bir kontrol algoritması önerilmiştir. Bu tasarımda iki adet PMSM kullanılmıştır. Simülasyon, farklı yol senaryolarında gerçekleştirilmiştir. Sistemin bu senaryolarda daha iyi dinamik ve statik performans sağladığı ve tekerleklerin hızını yüksek hassasiyetle kontrol ettiği gözlenmiştir (Hartani, M. and Miloudi 2010). Yapılan diğer bir çalışmada, eğri bir yolda iyi araç stabilitesini sağlayacak bir elektronik diferansiyel önerilmiştir. Önerilen çekiş sistemi, iki sürüş tekerleği tahrikini sağlayan iki PMSM' den oluşmaktadır. Önerilen kontrol yapısı, her bir tekerlek motoru için bulanık mantık kontrolör ile doğrudan tork ayarlanması yapılmıştır. Yapılan çeşitli simülasyon senaryoları sonucunda kavisli bir yolda iyi araç stabilitesi gözlenmiştir (Hartani *et al.* 2009).

Elektrikli araçlar günümüz teknolojisinde motor sporlarında yaygın olarak kullanılmamaktadır. Motor sporlarında başlangıç adımı olarak kullanılan go-kart,

üzerinde İYM barındırmaktadır. Yapılan çalışmada çevre kirliliğini minimum düzeye düşürmek amacıyla go-kartların yerine e-kart araçlar önerilmiştir. Bu nedenle elektrik motorlarının iletmesi gereken gücü ve batarya kapasitelerini belirlemek için farklı yaş kategorilerindeki go-kart yarışlarının teknik parametreleri analiz edilmiştir. Bu analizler sonucunda günümüz batarya teknolojisi çocuk ve genç yaş için e-kart oluşturmayı mümkün kılarken yetişkin kullanıcılar için henüz yeterli düzeyde olmadığı görülmüştür. Çalışmalar sonucunda e-kartlar, go-kartların oluşturduğu gürültü düzeylerinin yalnızca altıda biri kadar gürültü oluşturmaktadırlar. Bununla birlikte, arka tekerleklerin tork ve güçlerinin bağımsız olarak kontrol edildiği için 10 kw'lık bir e-kart, 12 kw'lık bir go-kart ile aynı performansa sahip olduğu gözlenmiştir (Wlodzimierz *et al.* 2018).

2.2 Dört Tekerlekten Tahrikli Araçlar

Dört tekerlekten tahrikli bir araç için karmaşık bir mekanik şanzıman sisteminin tasarım ve üretimiyle uğraşılması yerine her bir tekerleğe birbirinden bağımsız elektrik motorları bağlanarak gerekli tork ve hızın kazandırılması amaçlanmıştır. Bahsi geçen sistem öncelikle Matlab/Simulink ortamında çeşitli kontrol algoritmalarıyla simülasyon üzerinden kontrol edilmeye çalışılmış ve sistemin en uygun dinamik performansı gösterdiği algoritma kullanılmıştır. Bu sistemin sonucu olarak artık ihtiyaç duyulmayan mekanik parçaların kullanılmaması ile araç kütlesi azaldığı ve verim arttığı gözlenmiştir (Zhou *et al.* 2010).

EMA 'larda aracın viraj alma ve riskli manevralar sırasında esnekliği artırmak için tekerleklerin ihtiyaç duyduğu torku doğrudan iletme oldukça fazla avantaj sağlar. Anlık talep edilen hız değerlerini her tekerleğin ayrı ayrı hesaplanması için Ackerman prensibi kullanılmıştır. Çalışmada önerilen EDS, yan kayma açısını minimize etmek için hem ön tekerleklerde hem de arka tekerleklerin direksiyon açısını kazandığı, karmaşık direksiyon sistemine sahip olan araçlar için önerilmiştir. Çeşitli ortamlarda simülasyonları yapılarak, sistemdeki dört tekerlek aynı anda viraj eğrisine teğet hale geldiklerinden, virajı dönmek için gerekli direksiyon açısının azaldığı ve bunun sonucu olarak; araç dönme performansını arttırdığı görülmüştür (Moazen and Sabahi 2014). Dört tekerleğinde direksiyon açısından etkilendiği karmaşık bir direksiyon yapısına sahip araçların EDS' i

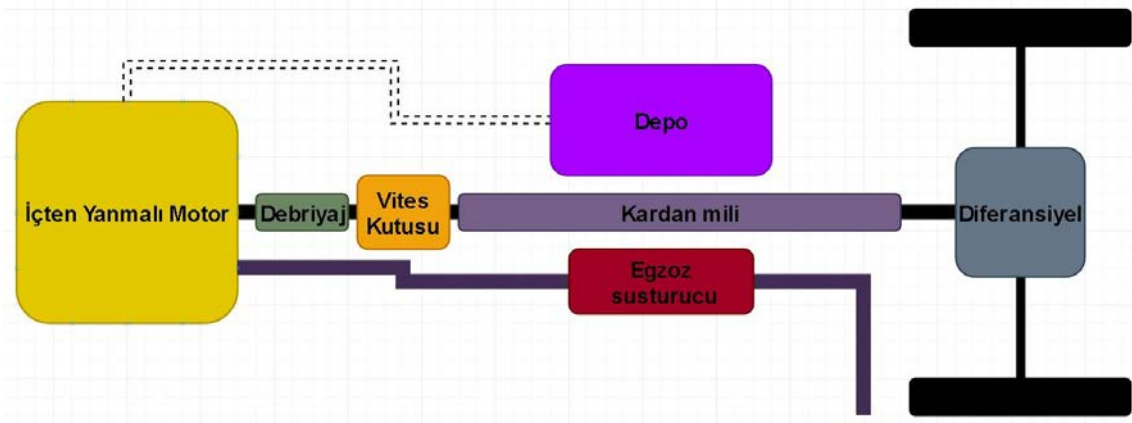
için yapılan diğeri bir çalıřmada ise her bir tekerleğın tork kontrolü için yapay sinir ağı temelli bir PID kontrolör kullanılmıřtır. Önerilen stratejinin direksiyon manevra ve dengesini düşük hızlarda iyileřtirdiğı gözlenmiřtir (Zhai and Dong 2011).

3. MATERYAL ve METOT

Tezin bu kısmında araç mekanik sistemi ve elektronik diferansiyel sistemi başlıklar altında detaylı anlatılmıştır.

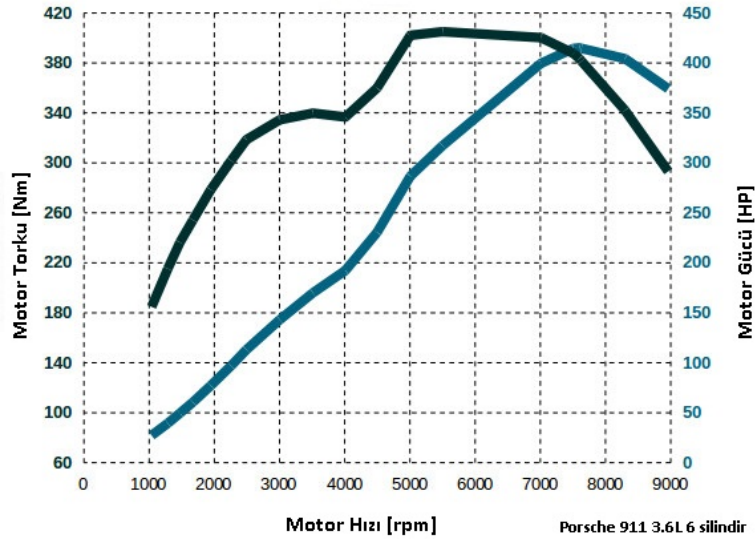
3.1 Araç Mekanik Sistemi

Teknolojik ürünlerin tümü kullanım amaçlarını tamamen karşılayabilmek için sürekli gelişim halindedirler. İYM' li araçlar uzun yıllardan beri kullanıcılara hizmet etmektedir. Şekil 3.1' de İYM' li bir aracın basit bir şekilde temel yapısı verilmiştir. İYM' li araçlarda motor, doğrudan hareket organı olan tekerleklere bağlanmamaktadır. Bunun temel sebebi tekerleklere iletilecek olan gücün ayarlanmasını sağlayacak debriyaj, vites kutusu gibi aktarım organlarına ihtiyaç duymasıdır. Doğrudan İYM kaldırılıp tahrik organı olarak elektrik motoru bırakıldığında sistemde debriyaj, vites kutusu gibi organlara ihtiyaç kalmayacaktır. Fakat toplam menzili artırmak için elektrikli aracın genel yapısında değişiklikler yaparak aktarım organı iki adet motora çıkarıldığında kardan mili ve diferansiyel kutusu gibi ağır mekanik parçalarda araçtan ayrılacaktır. Böylece elektrikli aracın toplam ağırlığı oldukça azalacaktır.



Şekil 3.1 İçten yanmalı motorlu bir aracın temel yapısı (Bacon 1972).

İçten yanmalı motora sahip bir aracın bu şekilde farklı aktarım organlarının kullanmasının temel sebeplerinden biri düşük devirlerde ürettiği gücün aracın hareketi için yeterli olmamasıdır. İçten yanmalı motora sahip Porsche 911 3.6L 6 silindirli modelinin motor tork ve güç değerleri şekil 3.2' de gösterilmiştir. Grafikten anlaşılacağı üzere motor 1000 dev/dk hıza sahip iken motor gücü yalnızca 180Nm civarlarındadır.



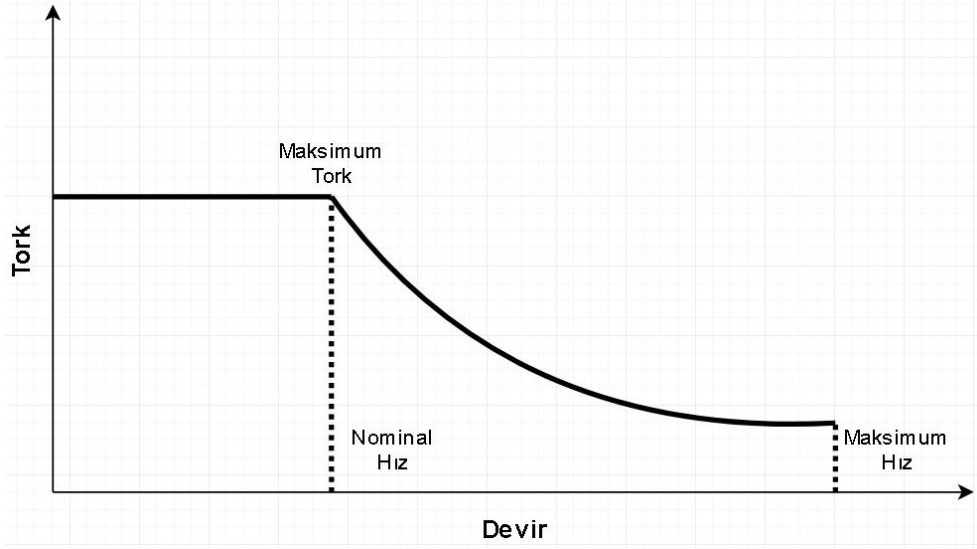
Şekil 3.2 İçten yanmalı motora sahip Porsche 911 modelinin motor tork ve güç değerleri (İnt.Kyn.7).

Düşük devirlerdeki motor güç değeri vites kutusuyla belirli oranlarda arttırılarak aracın talep ettiği güç sağlanmış olur. Çizelge 3.1’ den de anlaşıldığı gibi araca ilk hareket verilirken(birinci viteste iken) vites kutusundaki dönüşüm ile tekere iletilen güç, anlık motor gücünün 3.91 katı olur ve böylelikle araç harekete başlar. Araç hızlandıkça talep ettiği güç azalacağından dolayı dönüşüm oranları da o düzeyde düşmektedir.

Çizelge 3.1 Porsche 911 vites dönüşüm oranları (İnt.Kyn.8).

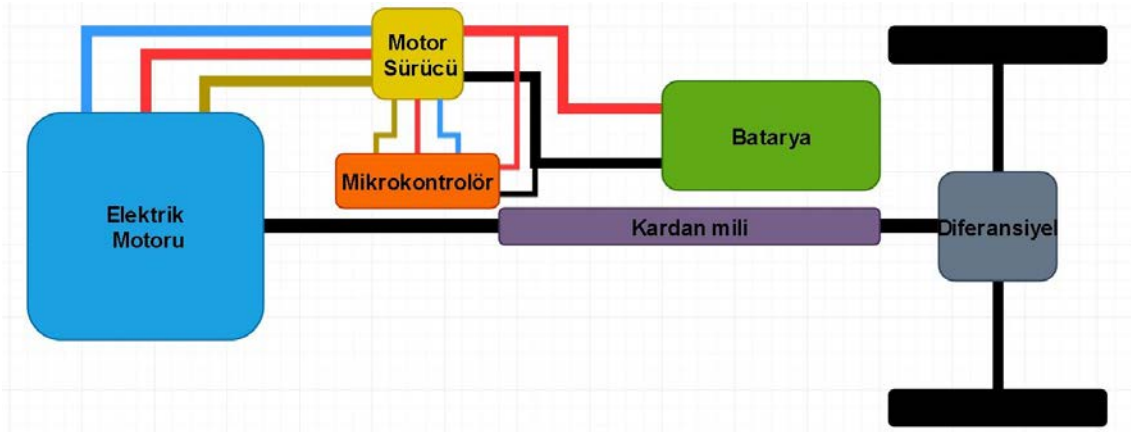
Vites	Dönüşüm oranı
1. vites	3.91
2. vites	2.29
3. vites	1.58
4. vites	1.18
5. vites	0.94
6. vites	0.79
7. vites	0.62
Geri vites	3.55

Elektrikli araçlar düşük devirlerde de maksimum torku üretebildikleri için araçlarda kullanım açısından vites kutusu, debriyaj gibi aktarma organlarına ihtiyaç duymazlar. Elektrik motorlarının hız-tork grafiği şekil 3.3’ te gösterilmiştir. Bu nedenle İYM’ lu araçlar zamanla yerini elektrikli araçlara bırakacak gibi görünmektedir.



Şekil 3.3 Elektrik motorlarının hız-tork grafiği (Quriroga *et al.* 2015).

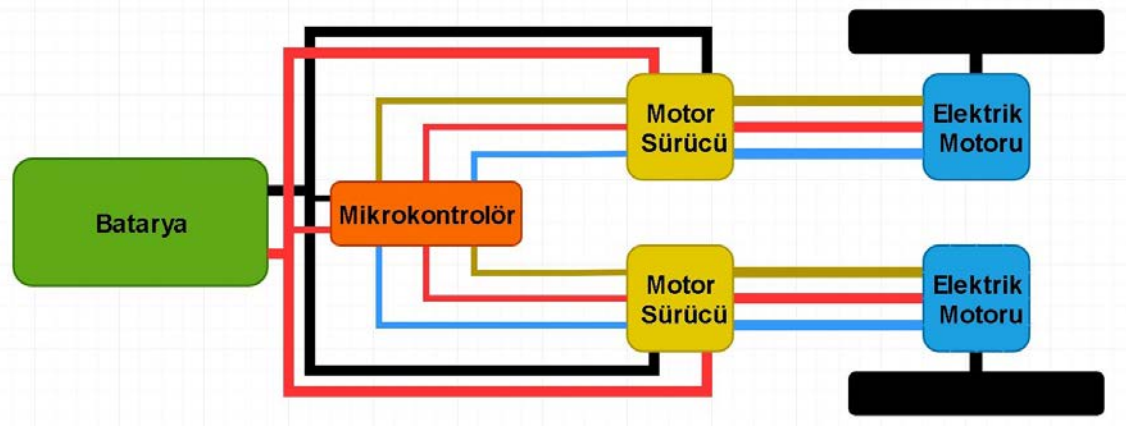
Elektrikli araçlar çevre kirliliğini açısından İYM' lu araçlara göre çok avantajlıdır çünkü egzoz emisyon oranları sıfırdır yani diğer bir deyişle çevreye zararlı herhangi bir gaz salınımı yapmazlar. Bu nedenle egzoz borusu ve susturucusu gibi sistemler elektrikli araçlarda bulunmazlar. Şekil 3.4' te tek motorlu elektrikli araçların temel yapısı verilmektedir.



Şekil 3.4 Tek elektrik motorlu elektrikli aracın basit yapısı.

İYM' li araçlara göre daha düşük maliyetle ulaşım sağlamaktadırlar fakat İYM' araçlarda yakıt deposunun dolma süresi 5-10 dk arasında iken elektrikli araçlarda batarya dolma süresi elektrik sisteminin gücüne bağlı olarak 1-2 saat civarlarına çıkabilmektedir. Bu nedenle elektrikli araçlar üzerinde bir takım değişiklikler yapılarak, tamamen dolu batarya ile alınan yolun artırılmasına yönelik çalışmalar yapılmaktadır. Bu tez

çalışmasında ise toplam menzilde önemli rolü olan araç ağırlığının düşürülmesi için aktarım organı olarak kullanılan kardan mili ve diferansiyel sisteminin çıkarılmasına yönelik çalışma yapılmıştır. Bu durumda elektrikli araç, aktarım organlarından dolayı meydana gelen kayıplardan da kurtulmuş olmaktadır. Bahsi geçen aktarım organlarının çıkarılmış halindeki elektrikli aracın temel yapısı şekil 3.5’ de verilmektedir.



Şekil 3.5 İki elektrik motorlu elektrikli aracın basit yapısı.

İki motorlu elektrikli araçlar, diferansiyel sisteminin çıkarılmasıyla aynı görevi yapacak öncekine göre daha küçük iki elektrik motoruyla yapmaktadır. Böylelikle sistem, hem mekanik sistemler üzerinde ısıya dönüşecek kayıplardan kurtulmuş olur hem de öncekine göre toplam ağırlık olarak azalmış olur.

Araç mekanik tasarımı olarak prototip bir araç düşünüldüğünden engelli aracı üzerinden değişiklik yapılmıştır. Araç tahrik sistemi olarak kullanılan iki adet 420 Wattlık 24 V doğru akım motoru şekil 3.1’ de gösterilmiştir. Doğru akım motorları, mıknatıslarla oluşturulmuş sabit manyetik alan içerisindeki rotor sargılarından akım geçirilmesi sonucu oluşan ikinci bir manyetik alanın birbirleri çekmesi veya itmesi sonucu hareket eden elektrik makineleridir. Motora bağlı redüktör millerine toplam 840 watt güç uygulanmaktadır. Elektrik motorları 1:10 dönüştürme oranı bulunan bir redüktöre sonrasında ise redüktör çıkış mili doğrudan tekerleklere bağlanmıştır. Böylelikle elektrikli araç tekerleğindeki toplam güç motorların ürettiği toplam gücün on katı olmaktadır.



Resim 3.1 Araç üzerinde kullanılan DC elektrik motoru.

DC motor özellikleri;

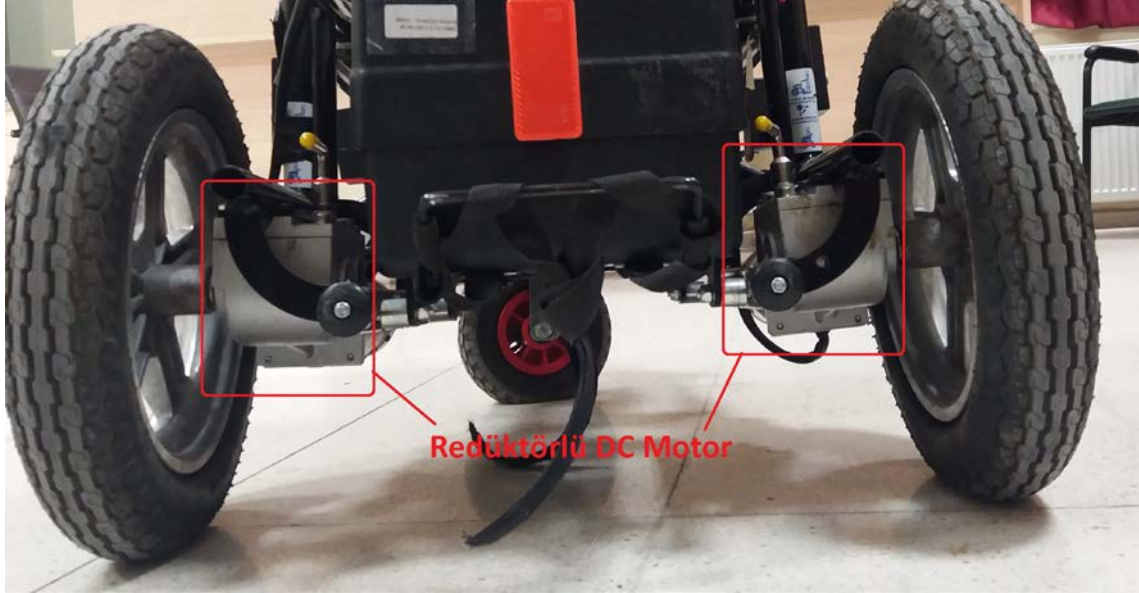
- 420Watt
- 4200 RPM
- 24 V DC

Araç tekerlekleri, araç yükünü taşıması, EDS' in yönlendirilmesini stabil yapabilmesi ve motor gücünün ileri-geri yönde düzgün bir şekilde aktarma yapabilmesi için tekerlek çapı 370mm seçilmiştir. Araç tekerleği resim 3.2'de gösterilmiştir.



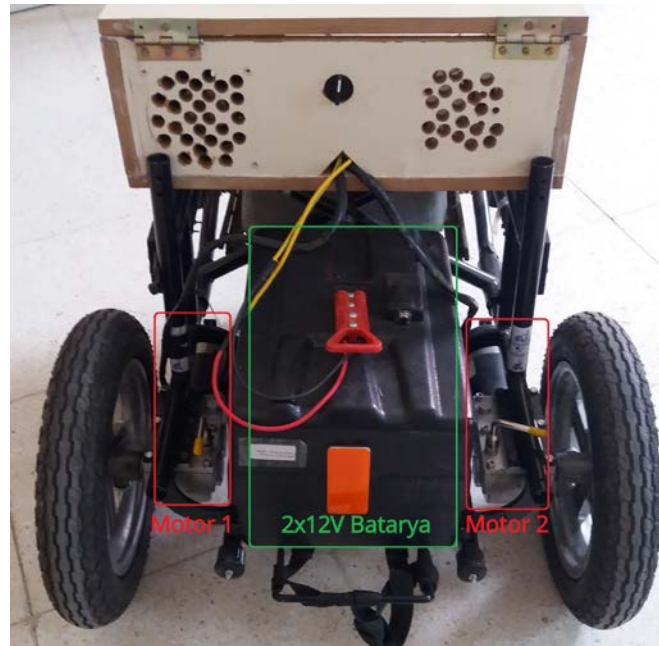
Resim 3.2 Araç üzerinde kullanılan tekerlek.

EDS, oluşturabilmesi için kullanılan iki adet 1:10 dönüştürme oranına sahip redüktörlü doğru akım motoru kullanılmıştır. Oluşturulan sistem resim 3.3' de gösterilmiştir.



Resim 3.3 Oluşturulan araçtaki tahrik sistemleri.

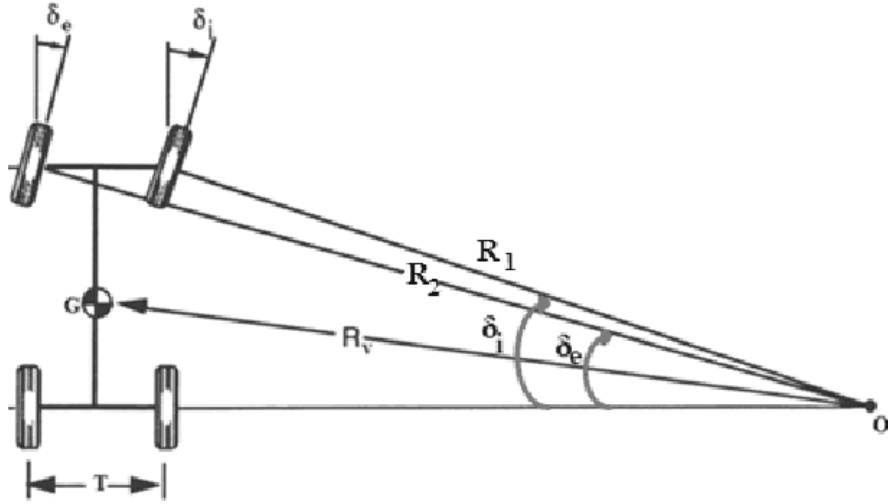
Sistemin genel yapısı, doğru akım motorlarına bağlı olduğu şase üzerinde ki bataryadan oluşmaktadır. Tez kapsamında oluşturulan elektrikli araç prototipinin son hali resim 3.4' de gösterilmiştir.



Resim 3.4 İki elektrik motorlu tez kapsamında geliştirilen elektrikli araç prototipi.

3.2 Elektronik Diferansiyel Sistemi

Diferansiyel sisteminin iki temel görevinden biri motor tarafından üretilen gücü 90° çevirerek sağ ve sol tekerleklere aktarmak diğeri ise tekerleklerin dönme yarıçapına bağlı olarak farklı hızlarda dönmesini sağlamaktır. Birinci temel görev için EDS' de motor tarafından üretilen gücü 90° çevirmeye gerek kalmamaktadır. Bunun nedeni ise aktarma organı olmadan her iki tekerleğe tahrik verecek olan doğru akım motorlarının doğrudan bağlanmasıdır. İkinci temel görev için dönme esnasında, dönme yönüne göre iç tekerleklerin alacağı yol, dış tekerleklere göre daha az olduğundan dolayı daha yavaş dönmeleri gerekmektedir. Bu problemi çözebilmek için de Ackermann matematiksel diferansiyel modeli yazılımsal olarak ifade edilerek kullanılmıştır. Diferansiyel sistemlerindeki matematiği basit bir şekilde ele alan Ackerman diferansiyel modeli şekil 3.6' da gösterilmiştir. Bu işlemde standart tip arkadan çekişli araçlar için tekerleklerin bu karmaşık devir ayarlama işlemini mekanik diferansiyel sistemi tarafından yapılmaktadır. Tez kapsamında oluşturulan sistemde ise direksiyon açısı ve araç hız bilgisi elektronik ortama aktarılarak matematiksel denklemlerden elde edilen sonuçlara göre EDS tarafından yapılmaktadır.



Şekil 3.6 Ackermann diferansiyel modeli (Nejh and Trabelsi 2015).

Şekil 3.6 'da verilen Ackermann modeli baz alınarak dönüş yönüne göre iç ve dış tekerleklerin açıları hesaplanmıştır.

Buna göre ön iç tekerlek dönme açısı($\delta_{\ddot{o}i}$);

$$\delta_{\ddot{o}i} = \arctan\left[\frac{\ell * \tan(\rho)}{\ell - ((k / 2) * \tan(\rho))}\right] \quad (3.1)$$

Aynı hesaplama şekliyle devam edecek olursak ön dış tekerlek dönme açısı($\delta_{\ddot{o}d}$);

$$\delta_{\ddot{o}d} = \arctan\left[\frac{\ell * \tan(\rho)}{\ell + ((k / 2) * \tan(\rho))}\right] \quad (3.2)$$

Hesaplamalarda kullanılan başlıca değişkenler;

- ℓ : Dingil mesafesi(m),
 k : King-Pim mesafesi(m),
 ρ : Direksiyon açısıdır($^{\circ}$).

$$R_1 = \frac{\ell}{\sin(\delta_{\ddot{o}i})} \quad (3.3)$$

$$R_2 = \frac{\ell}{\sin(\delta_{\ddot{o}d})} \quad (3.4)$$

$$R_{ai} = \frac{\ell}{\tan(\rho)} - \frac{T}{2} \quad (3.5)$$

$$R_{ad} = \frac{\ell}{\tan(\rho)} + \frac{T}{2} \quad (3.6)$$

- R_1 : Ön iç tekerlek dönme merkezine uzaklığı(m),
 R_2 : Ön dış tekerlek dönme merkezine uzaklığı(m),
 R_{ai} : Arka iç tekerlek dönme merkezine uzaklığı(m),
 R_{ad} : Arka dış tekerlek dönme merkezine uzaklığı(m),
 T : Aks mesafesidir(m).

Tekerleklerin açısal hızları;

$$w_{\ddot{o}i} = \frac{v * R_1}{(R_v) * r} \quad (3.7)$$

$$w_{\ddot{o}d} = \frac{v * R_2}{(R_v) * r} \quad (3.8)$$

$$w_{ai} = \frac{v * R_{ai}}{(R_v) * r} \quad (3.9)$$

$$w_{ad} = \frac{v * R_{ad}}{(R_v) * r} \quad (3.10)$$

$$R_v = \sqrt{(R_{ai} + (T / 2))^2 + (\ell_g)^2} \quad (3.11)$$

- $w_{\ddot{o}i}$:Ön iç tekerleğin açısai hızı(rad/sn),
 $w_{\ddot{o}d}$:Ön dış tekerleğin açısai hızı(rad/sn),
 w_{ai} :Arka iç tekerleğin açısai hızı(rad/sn),
 w_{ad} :Arka dış tekerleğin açısai hızı(rad/sn),
 R_v :Araç ağırlık merkezinin dönme merkezine olan uzaklığı(m),
 r :Araç tekerleği yarıçapı(m),
 v :Araç hızı(m/sn),
 ℓ_g :Araç ağırlık merkezinin aks mesafesine dik uzaklığıdır(m).

Oluşturulan yazılım için araca bağıli parametreler resim 3.5 'de görülen tez için oluşturulmuş araçtan alınmıştır.



Resim 3.5 Tez kapsamında oluşturulan elektrikli araç.

Dingil mesafesi	:	0.740 m
King-Pim mesafesi	:	0.4 m
Aks mesafesi	:	0.65 m
Tekerlek yarıçapı	:	0.195 m

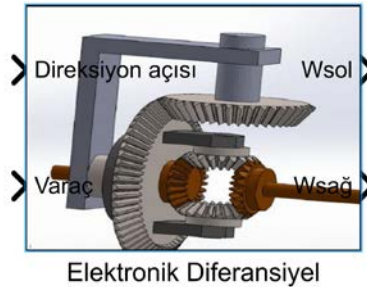
Ağırlık merkezi ile aks mesafesi arasındaki dik uzaklık (ℓ_g), dingil mesafesinin yarısı alınmıştır.

3.2.1 Bilgisayar Ortamında Matematiksel Modellenmesi

Sistemin bilgisayar modeli çıkarılması için öncelikle parça parça halinde sistemdeki elemanların modellenmesi gerekmektedir. Elektrikli araçta alt sistemler olarak elektronik diferansiyel sistemi ve elektrik motorlarının modellenmesi gerekmektedir.

3.2.1.1 Elektronik Diferansiyel Modeli

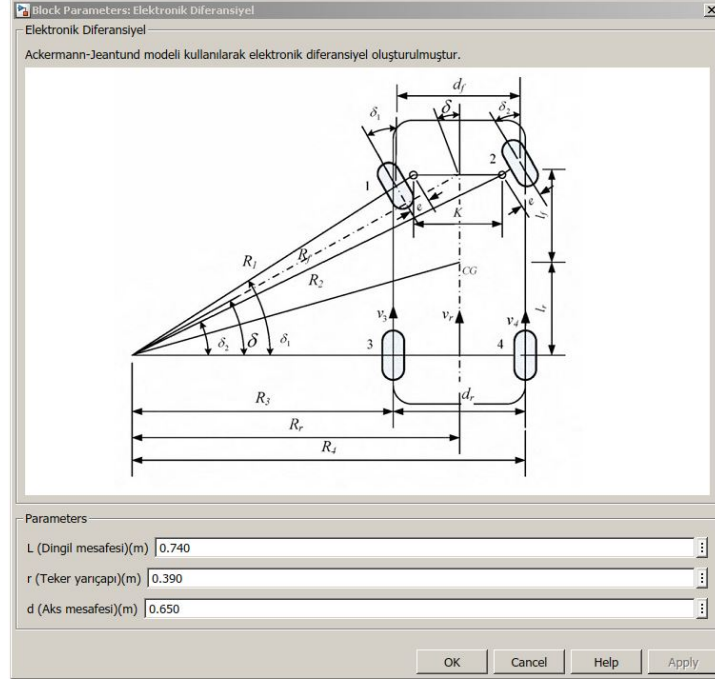
Önceki başlıkta anlatılan denklemlerin tamamıyla mekanik diferansiyelin matematiksel tanımı yapılmıştır. Bu başlık altında ise bu denklemler kullanılarak bilgisayar ortamında elektronik olarak sistemin modellenmesi yapılmıştır. Araç hızı ve direksiyon açısındaki değişimlerin, tekerleklerin açısal hızları üzerindeki etkileri yukarıdaki denklemler sayesinde ortaya koyularak, şekil 3.7 'de oluşturulan sistemin MATLAB/Simulink 'te hazırlanan elektronik diferansiyel bloğu gösterilmiştir.



Şekil 3.7 Elektronik diferansiyel bloğu.

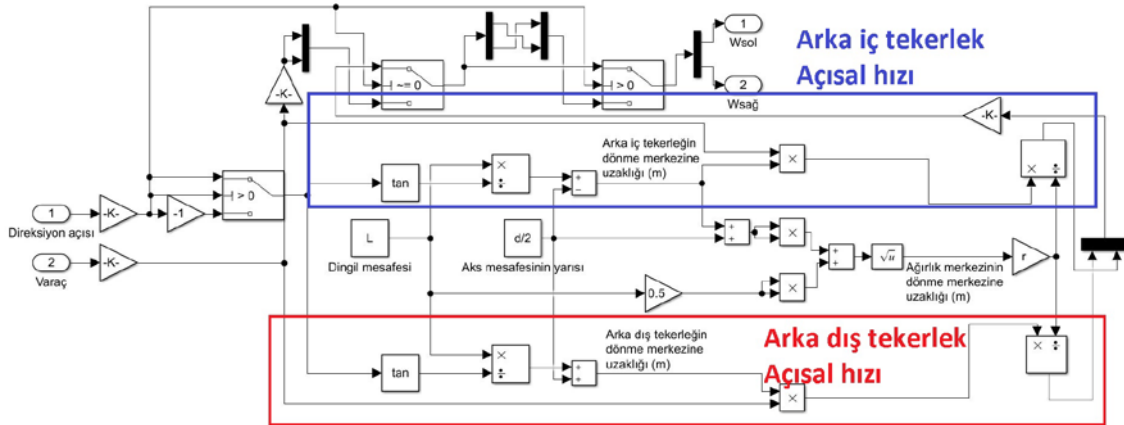
Tez kapsamında oluşturulacak araç arkadan çekişli olacağından dolayı hesaplamalar arka sol ve arka sağ tekerlek üzerinden yapılmıştır. Oluşturulan blok için kullanıcı ara yüzü

hazırlanmış ve yapılacak parametre değişikliklerinin rahatlıkla yapılması sağlanmıştır. Elektronik diferansiyel bloğu kullanıcı ara yüzü şekil 3.8’ de gösterilmiştir.



Şekil 3.8 Oluşturulan elektronik diferansiyel bloğu kullanıcı ara yüzü.

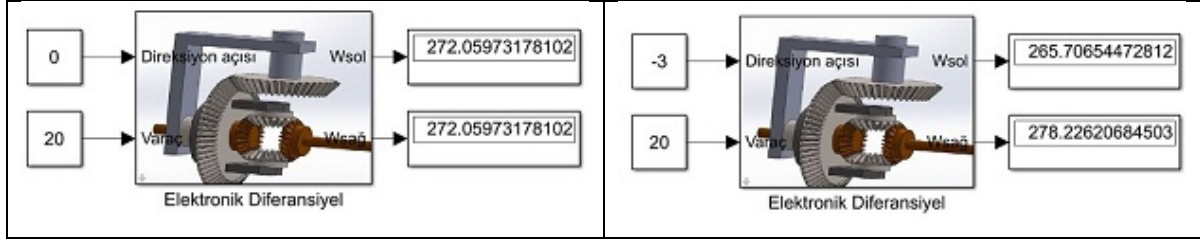
Blok ara yüzündeki parametrelere göre prensibin uygulandığı elektronik diferansiyel bloğunun iç yapısında, arka iç ve dış tekerleklerin açısal hızlarının hesaplandığı kısım şekil 3.9’ da gösterilmiştir.



Şekil 3.9 Oluşturulan elektronik diferansiyel bloğu içeriği.

Örneğin; Elektrikli araç 20 km/h(272.05 dev/dk) hızla hareket ederken 3° sola döndüğünü

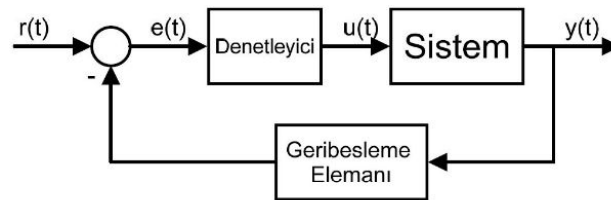
varsayalım, bu durumdaki arka tekerleklerin hız değerleri şekil 3.10’da verilmiştir. Buna göre araç stabil bir şekilde virajı dönebilmesi için sağ arka tekerlek 278.22 dev/dk hızıyla dönmesi gerekirken, sol arka tekerlek ise 265.70 dev/dk ile dönmesi gerekmektedir. Viraja girilirken dengenin sağlanabilmesi açısından iç tekerleğin açısal hızı azalırken, dış tekerleklerin açısal hızı artmaktadır.



Şekil 3.10 Araç hızının 20 km/h olduğu durum için arka sağ ve sol tekerleklerin hızlarının(dev/dk) model çıktısı.

3.2.1.2 PID Kontrolör Tasarımı

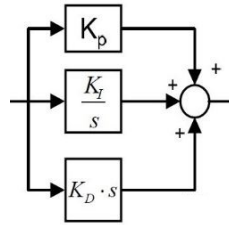
Endüstriyel alanlarda hız, konum ve akım gibi parametreleri kontrol etmek, sistemin stabil çalışması açısından oldukça önem arz etmektedir. Bu tip kontroller farklı birçok kontrol algoritmasıyla yapılabilmektedir fakat bu kontrol algoritmalarından en çok kullanılanı oransal-integral-türevsel denetleyici kontrol döngü yöntemi yani PID kontroldür. Bu kontrol metotlarından en çok kullanılanı oransal-integral-türevsel denetleyici kontrol döngü yöntemi yani PID kontroldür. PID denetleyicisi belli bir periyotla hata değerini, yani sistemde kontrol etmek istediğiniz parametre için referans değeri ile aynı parametrenin mevcut haldeki durumunun farkına bakarak kendi parametrelerine uygun bir şekilde kontrolör denetimi için çıkış sinyali üretmektedir. Üretilen çıkış sinyali ile sistem kontrolü sağlanmış olur. Şekil 3.11’ de geri beslemeli bir kontrol sisteminin yapısı gösterilmektedir.



Şekil 3.11 Geri beslemeli bir kontrol sistemi yapısı.

Yapısının basitliği itibariyle, ayarlanması gereken değişken sayısının az olması ve fiziksel olarak hayata geçirmenin kolay yapılması nedeniyle çok fazla tercih edilmektedir. Belirtildiği gibi PID kontrol yönteminde üç ayrı parametre içermektedir. Oransal katsayısı, integral katsayısı ve türev katsayısı parametreleri mevcuttur.

PID kontrolcüsü sistemin kontrol edilmek istenen parametresinin geri besleme ile olması istenilen değer ile karşılaştırır ve farkından bir hata değeri elde etmektedir. Bu hata değerini, kontrolör en aza hatta sıfıra indirmek için bir çıkış sinyali üretmektedir. Bu şekilde sistem belli periyotla aynı işi yaparak hata değerini hesaplar ve sistemi istenilen değere yakınsamaya devam eder. Bu kontrol tipi daha çok doğrusal sistemlere uygulanmaktadır. Şekil 3.12’ de PID kontrolörünün içyapısı gösterilmektedir.



Şekil 3.12 PID kontrolörünün içyapısı.

Bir sistemin laplace dönüşümü için integral fonksiyonunun $1/s$, türev fonksiyonunun ise s ile çarpılması gerekmektedir.

$$\frac{U(s)}{E(s)} = \left(K_p + \frac{K_i}{s} + K_d \cdot s \right) \quad (3.12)$$

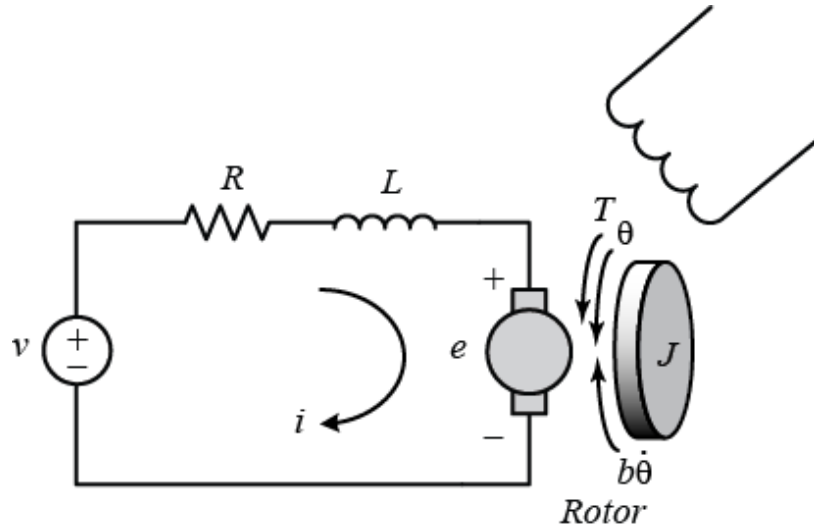
K_p , K_i ve K_d kazanç katsayılarının olduğu fonksiyonu s domeninde yazılırsa denklem 3.12 elde edilmektedir.

3.2.1.3 Doğru Akım Elektrik Motoru Modellenmesi

DC motorlar, evlerde, taşıtlarda, trenlerde ve daha birçok elektrik sisteminde olduğu gibi çoğu kontrol sistemi için de önemli bir elektrik makinesidir. Elektrik motorlarının ve

sistemin geri kalanının iyi bir şekilde kontrol edilebilmesi için matematiksel modelin ne kadar önemli olduğu iyi bilinmektedir. Bir DC motor için makine davranışını iyi bir doğrulukla temsil eden birçok model vardır. Bununla birlikte, modelin parametreleri de önemlidir çünkü matematiksel model, modelde doğru parametreler kullanılmadan o sistem için doğru bir çıktı sağlayamaz.

Kontrol sistemlerinde yaygın olarak kullanılan tahrik elemanlarından biri de doğru akım motorlarıdır. Aşağıdaki şekil 3.13 'de doğru akım motoru için elektriksel eşdeğer devresi gösterilmiştir (Emhemed and Mamat 2012).



Şekil 3.13 Doğru akım motoru elektriksel eşdeğer devresi (Bharatiraja *et al.* 2016).

Doğru akım motorunun matematiksel modellemesinde kullanılacak parametreler;

J	: Rotorun atalet momenti	kg.m^2
b	: Mekanik sistemin sönümlenme oranı	N.m.sn
R	: Elektrik direnci	ohm
L	: Elektrik endüktansı	mH
V	: Giriş gerilimi	V
T	: Tork	N
i	: Armatür akımı	A
K_t	: Armatür sabiti	-

K_e : Motor sabiti -
 $\dot{\theta}(t)$: Rotor hızı rad/sn

Tork, aşağıdaki şekilde akımla orantılıdır.

$$T = K_t * i \quad (3.13)$$

Elektro motor kuvveti ise aşağıdaki denklemde de görüldüğü gibi dönme hızıyla orantılıdır.

$$e = K_e * \dot{\theta}(t) \quad (3.14)$$

Bizimde kullandığımız Système International d'Unités (SI) birim standartlarına göre armatür sabiti(K_t) ve motor sabiti(K_e) birbirine eşittir. Termodinamiğin birinci kanunu ve Kirchhoff akım yasası baz alınarak şekil 3.9 incelendiğinde denklem 3.15 ve 3.16' da ortaya çıkmaktadır.

$$J.\ddot{\theta}(t) + b.\dot{\theta}(t) = K_t.i(t) \quad (3.15)$$

$$L.\frac{di(t)}{dt} + R.i(t) = V(t) - K_e.\dot{\theta}(t) \quad (3.16)$$

Yukarıdaki denklemleri lablace dönüşüm formatında yazarsak;

$$(Js + b).\theta(s) = K.I(s) \quad (3.17)$$

$$(Ls + R).I(s) = V(s) - K.\theta(s) \quad (3.18)$$

3.17 numaralı denklemde I(s) yalnız bırakılıp 3.18 numaralı denklemdeki yerine yazılırsa aşağıdaki denklem elde edilir.

$$\frac{(Ls + R)(Js + b).\theta(s)}{K} = V(s) - K.\theta(s) \quad (3.19)$$

Sistemin kontrol edilebilmesi için transfer fonksiyonunun çıkış/giriş formunda yazılması gerekmektedir. Bu nedenle sistem girişi gerilim, çıkış ise rotor hızı olduğundan denklem $\frac{\theta(s)}{V(s)}$ formuna dönüşmüştür. Sonuç olarak oluşan denklem;

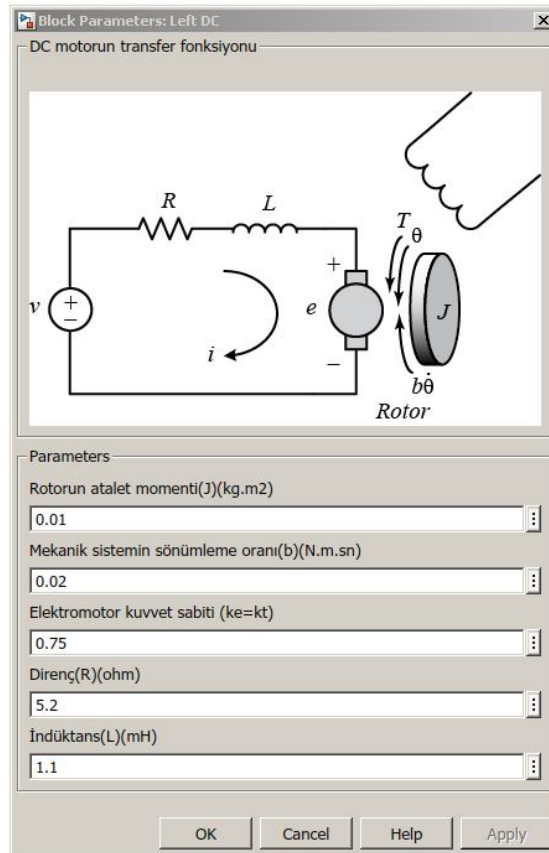
$$\frac{\theta(s)}{V(s)} = \frac{K}{(Ls + R)(Js + b) + K^2} \quad (3.20)$$

Kullanılacak doğru akım motorunun parametreleri çizelge 3.1 'de verilmiştir.

Çizelge 3.2 Doğru akım motoru parametreleri.

Parametre	Değer	Birim
J	0.01	kg.m ²
b	0.02	N.m.sn
R	5.2	ohm
L	1.1	mH
K	0.75	

Yukarıdaki denklemler kullanılarak araç modelinde kullanılmak üzere doğru akım motorunun modellenmesi yapılmıştır. Şekil 3.14 'te oluşturulan blok ara yüzü ve şekil 3.15' te blok görseli gösterilmektedir.

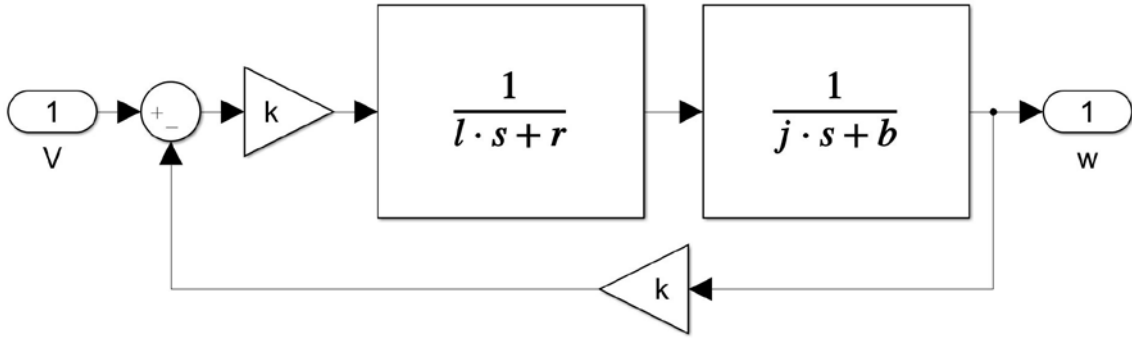


Şekil 3.14 Oluşturulan doğru akım motoru model ara yüzü.



Şekil 3.15 Oluşturulan doğru akım motoru model bloğu.

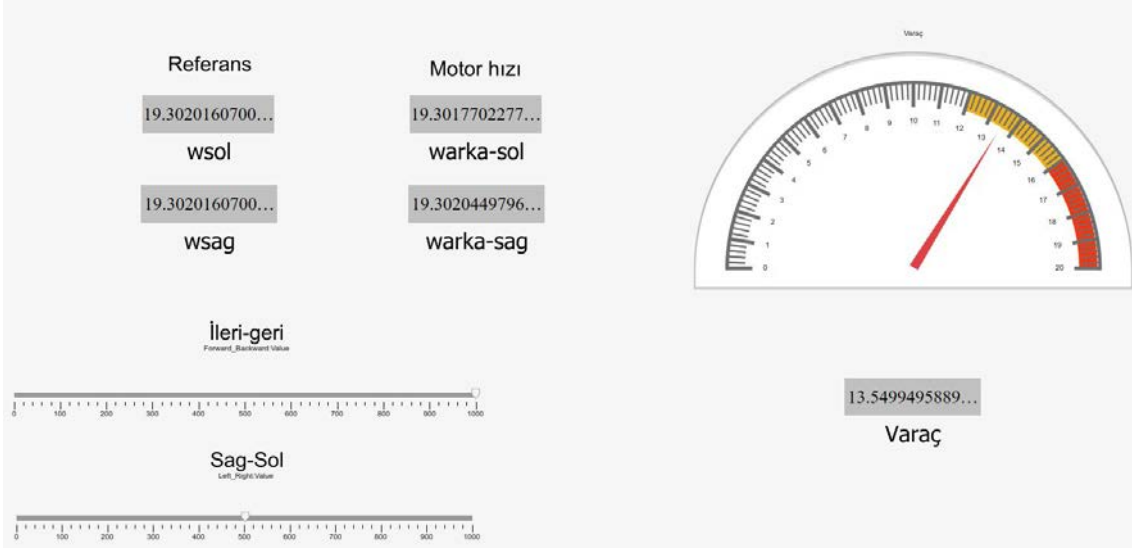
Oluşturulan DC motor bloğu giriş olarak aldığı gerilime karşılık rotor hızının değerini vermektedir. Şekil 3.16 'da blok içerisindeki denklemlerin transfer fonksiyonu halindeki biçimi gösterilmiştir.



Şekil 3.16 DC motor matematiksel modeli.

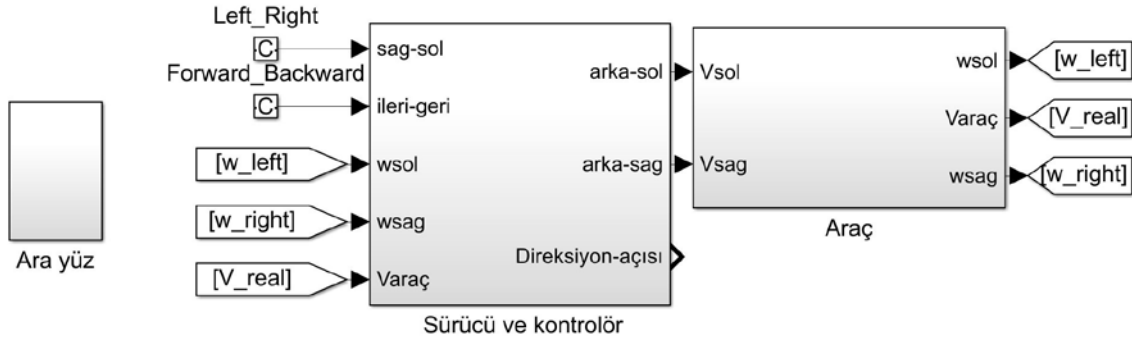
3.2.1.4 Elektrikli Araç Modeli

Bilgisayar ortamında oluşturulan model, sistemin eksikliklerini görüp, sistem gerçekleştirileceği sırada bu tecrübeden faydalanabilmek amacıyla yapılmaktadır. Böylelikle üretilecek herhangi bir sistemin ilk yapım maliyeti ciddi oranda düşmektedir. Tez kapsamında yapılacak elektrikli araç, arka iki tekerlekten tahrikli bir elektrikli araç prototipi olarak planlanmıştır. Bu nedenle oluşturulacak modelde elektronik diferansiyel sisteminden elde edilen hız verilerine göre araç üzerindeki iki elektrik motoru kontrollü bir şekilde hareketlendirilmektedir. Tez kapsamında oluşturulacak aracın matematiksel modeli elde edilerek, MATLAB/Simulink ortamında verilerin kullanıcı tarafından gözlenebilmesi için oluşturulan elektrikli araç prototip ara yüzü şekil 3.17' de gösterilmektedir.



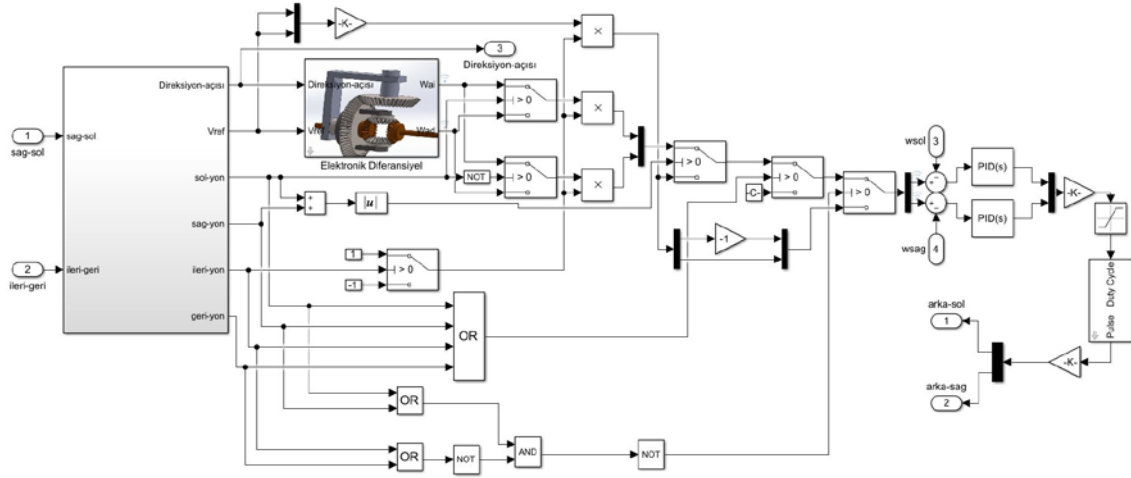
Şekil 3.17 Tasarlanan elektrikli araç model ara yüzü.

Elektrikli araç, elektrik motor sürücüsü ve elektrik motorları olmak üzere iki temel kısımdan oluşmaktadır. Şekil 3.17 'deki ara yüz sürücünün elektrikli aracı bilgisayar ortamında kontrol edebilmesi ve doğru akım motorlarının verilerini görebilmesi için düzenlenmiştir. Arka planda çalışan elektrikli araç modelinin tamamı şekil 3.18 'de gösterilmiştir.



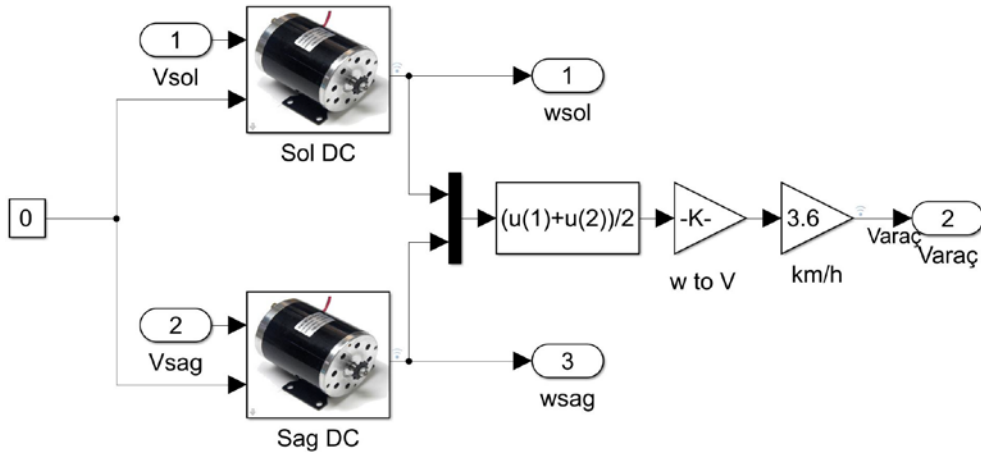
Şekil 3.18 Oluşturulan elektrikli araç modeli.

Direksiyon ve gaz sisteminden alınan veriler Ackerman prensibine göre elde edilen denklemlerde kullanılarak elektrikli aracın sağ ve sol tekerleklerinin referans hızları elde edilmektedir. Sürücü ve kontrolör bloğu elde edilen referans değer verilerini gerçek değerleri ile karşılaştırılarak motora olmasını istediği hıza ulaşması veya gerekli hıza düşmesi için motor gerilimini düzenlenmektedir. Sürücü ve kontrolör modeli şekil 3.19' da gösterilmiştir.



Şekil 3.19 Oluşturulan sürücü ve kontrolör modeli.

Sistemlerden alınan direksiyon ve gaz pedalı bilgisi dikkate alınarak oluşturulan sinyaller gerçek değerler ile kıyaslandıktan ve PID kontrolöre verildikten sonra çıkış olarak motora verilecek PWM sinyalinin doluluk oranı elde edilmektedir ve bu doluluk oranı baz alınarak oluşturulan tetikleme sinyali ile motor gerilimi oluşturulmaktadır. Şekil 3.20 'de Arka sağ ve sol tekerleklere bağlı doğru akım motorunun matematiksel modelleri yer almaktadır. Sistemde ki parametreler ve sistem geliştikçe araç modeli de kompleksleşmeye başlamaktadır.

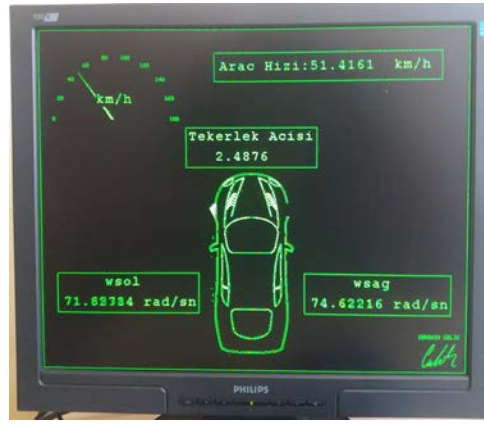


Şekil 3.20 Oluşturulan araç modeli.

Bilgisayar ortamında oluşturulan modelin sorunsuz çalışması nedeniyle aynı sistem simülatör olarak adlandırılan sisteme benzetilmiştir.

3.2.2 Elektronik Diferansiyel Sistem Simülâtörünün Oluşturulması

Elektronik diferansiyel sisteminin bir direksiyon ve gaz pedalıyla simülâtörü gerçekleştirilmiştir. Bu simülâtör sistemi için STMicroelectronics firmasının STM32F405RG mikroişlemcisi kullanılarak gaz pedalı ve direksiyondan alınan veriler anlamlandırılmıştır. Mikroişlemci üzerinde yapılan hesaplamalar sonucu elde edilen veriler, Video Graphics Array(VGA) portu kullanılarak 800x600 boyutundaki bir ekrana gönderilmektedir. Simülâtör ekran görüntüsü resim 3.6' da gösterilmiştir.



Resim 3.6 Çalışma kapsamında geliştirilen simülâtör ekran görüntüsü.

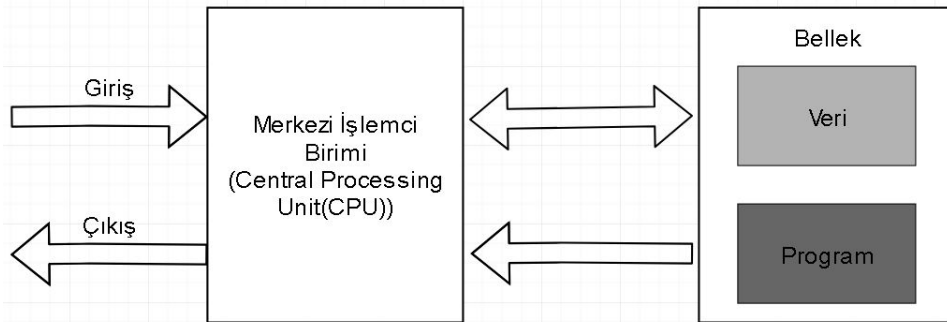
3.2.2.1 Mikroişlemci

Mikroişlemci, ana işlem biriminin özelliklerinin tamamını bir tek yarı iletken devresinde ortaya koyan programlanabilir bir sayısal elektronik bileşendir. Kişisel bilgisayarlardaki genel amaçlı mikroişlemciler, hesaplama, metin düzenleme, multimedya görüntüleme ve internet üzerinden iletişim için kullanılmaktadır. Birçok mikroişlemci, cihazlardan otomobillere, cep telefonlarına ve endüstriyel işlem kontrolüne kadar sayısız nesne üzerinde dijital kontrol sağlayan yerleşik sistemlerin bir parçası olmuştur.

1948'de transistörün ortaya çıkmasından beri elektronikteki en heyecan verici yenilik olmuştur. Bu harika cihaz sadece dijital elektronik alanında devrim niteliğinde değil, aynı zamanda insan yaşamının hemen her alanına girmeye başlamıştır. Transistörün ardından yarı iletkenlerin toplu halde bir araya getirildiği mikroişlemciler ortaya çıkmaya

başlamıştır. "Mikroişlemci" teriminin ilk kullanımı, 1968'de açıklanan Sistem 21 adlı mikroişlemciyi, küçük bilgisayar sistemlerinde kullanılan özel tümleşik devre olarak tanımlayan Viatron Bilgisayar Sistemleri olmuştur(İnt.Kyn.9). Mikroişlemcilerin kullanım alanları, çok karmaşık proses kontrol cihazlarından ve kontrol ekipmanlarından basit oyun makinelerine ve hatta oyunculara kadar uzanmaktadır.

Mikroişlemci, merkezi bir işlem biriminin yapması gereken işleri tek bir entegre devre veya birden fazla entegre devreleri içeren bir bilgisayar işlemcisidir. Mikroişlemciler yapılması gereken işlem için komutu bellekten okur, kodunu çözer ve işlemi yapmaktadır. Verilen talimatlara göre verileri istenildiği haliyle aritmetiksel ve mantıksal şekilde işlemektedir. Veriler giriş birimlerinden veya bellekten alınıp gerekli işlemlerden geçirilir ve sonunda belleğe ve başka bir cihaz için çıkış birimine gönderilir. Mikroişlemci basit olarak bir program kısmı, veri kısmı ve giriş-çıkışlardan oluşmaktadır. Mikroişlemci basit yapısı şekil 3.21' de gösterilmiştir.



Şekil 3.21 Mikroişlemci basit yapısı.

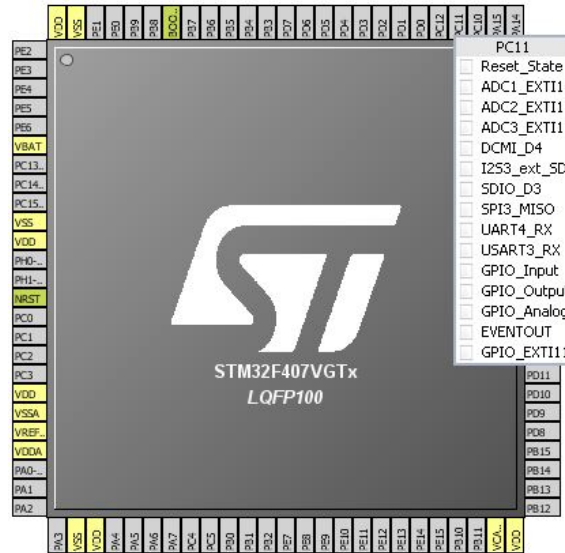
Tez kapsamında ARM (Acorn indirgenmiş komut takımıyla hesaplama makinesi)(Acorn RISC(Reduced Instruction Set Computer) Machine) mimarili mikroişlemci olarak adlandırılan STM32F405RG ve araç üzerinde de STM32F103 işlemciler kullanılmıştır. ARM firması aslında mikroişlemci üretmeyen fakat ARM mimarisinde mikroişlemciler tasarlayan ve bu tasarımları ST, TI(Texas Instrument), Samsung, Apple, Toshiba, HP(Hewlett-Packard) gibi birçok firmaya lisansını veren bir firmadır. Daha henüz firma kurulmadan 5 yıl önce kendilerine özgün mimari tasarlayıp mikroişlemci alanına yeni bir boyut kazandırmış ve 1990 'da Acorn, Apple ve VLSI Technology firmalarıyla birlikte ortaklaşa firma kurmuşlardır. ARM mimarisine sahip mikroişlemcilerin, diğer

mikroişlemcilerde göre avantajlarını

- Daha düşük güç tüketimi
- Daha yüksek performans
- Daha uygun fiyat,

şeklinde sıralayabiliriz. Bunlar ve bunlar gibi diğer göz alıcı özellikleri, ARM mimarisine sahip mikroişlemcileri, günümüzde özellikle yüksek hıza gereksinim duyan taşınabilir ürünler ve bununla birlikte birçok teknolojik alanda kullanılmaya devam etmesine neden olmuştur. Popülarlığı her geçen gün artan bu mikroişlemci türü, başta teknolojik aletlerde satış rekorları kıran Apple firmasının hemen hemen tüm ürünlerinde, Nintendo gibi oyun konsolu firmasında, çoğu cep telefonu üzerinde ve daha birçok ev aletinde kullanılmaktadır.

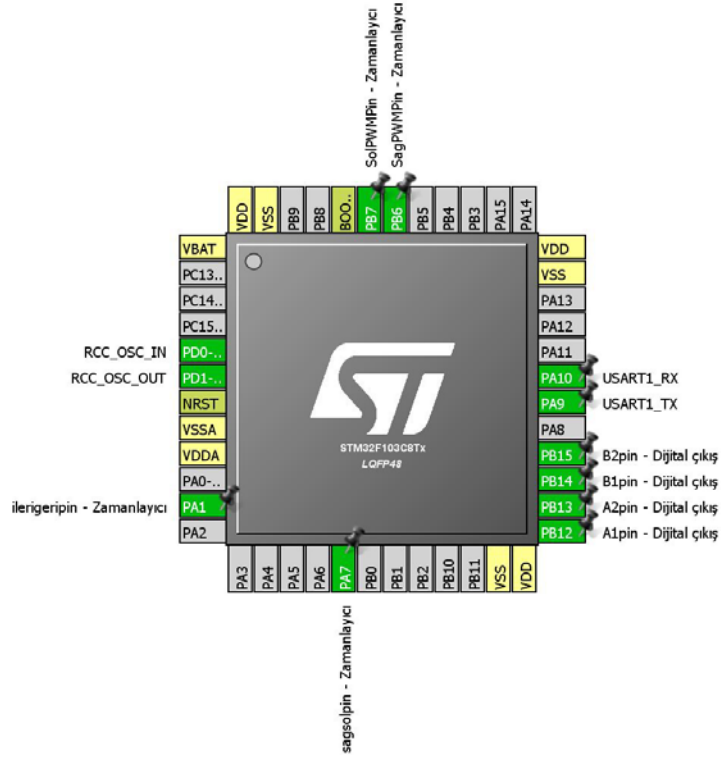
ARM mimarisine sahip bir mikroişlemci üzerinde çalışma yapılacaksa, yapılması gereken ilk işlem mikroişlemci pinlerinin konfigürasyonunu yapılacak olan işe göre düzenlemesini sağlamaktır. Çünkü bu mimarideki mikroişlemcilerde bir pinin birden fazla işlevi vardır ve sizin yapılacak olan işleme göre bu işlevlerden bir tanesini pine uyarlamamız gerekmektedir. ARM mirarisine sahip STM32F407VG mikroişlemcisinin C portunun 11. pininin alabileceği görevler şekil 3.22’ de gösterilmiştir.



Şekil 3.22 ARM mirarisine sahip STM32F407VG mikroişlemcisinin C portunun 11. pininin alabileceği görevler.

Tez kapsamında oluşturulan araç üzerindeki mikroişlemcide kullanılan pinler şekil 3.23

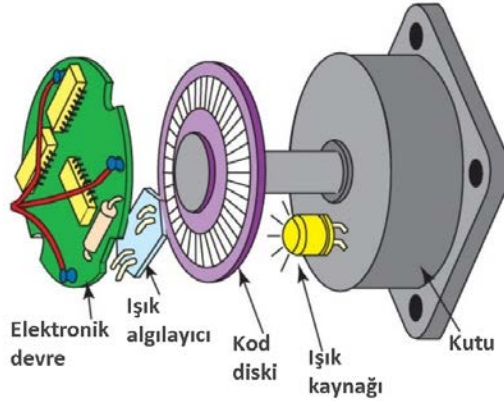
'de gösterilmektedir.



Şekil 3.23 Araç üzerinde kullanılan mikroişlemci için kullanılan pinler.

3.2.2.2 Enkoder

Enkoder, bağlı olduğu kısmın hareketini bir tür kontrol cihazı tarafından okunabilen bir elektrik sinyaline dönüştürmektedir. Enkoder pozisyonu, sayımı, hızı veya yönü belirlemek için kullanılabilir. Bir geri bildirim sinyali gönderir. Bir kontrol cihazı bu bilgiyi belirli bir fonksiyon için veri göndermek için kullanabilir. Optik teknolojisi kullanılan artımlı bir döner enkoderin teknik yapısı şekil 3.24'da gösterilmiştir.



Şekil 3.24 Optik teknolojisi kullanılan artımlı bir döner enkoderin teknik yapısı (İnt.Kyn.10).

Enkoder, pozisyon ve hız gibi bilgileri elde etmek için mekanik, dirençli, manyetik ve optik gibi birçok farklı teknolojik yapıdadır. Bunlar içerisinde en yaygın olanı optik enkoderdir. Işık kaynağı olarak adlandırılan led tarafından yayılan ışık demeti mat çizgilerle yapılandırılmış kod diskinden geçmektedir. Enkoder mili döndükçe ledten gelen ışık demeti kod diskindeki çizgiler nedeniyle kesilmektedir. Bu olay ise ışık algılayıcısı tarafından darbe sinyali elde edilmesine sebep olmaktadır. Bu darbe sinyalleri elektronik devre tarafından düzenlenerek mikroişlemci tarafından rahatlıkla algılanabilecek hale getirilmektedir. Böylelikle hız veya pozisyon bilgisi elde edilmiş olunur. Enkoder başlığı altında sayabileceğimiz iki temel tür vardır. İlki mutlak tip enkoder, diğeri ise artımsal tip enkoderlerdir.

Mutlak Tip Enkoder

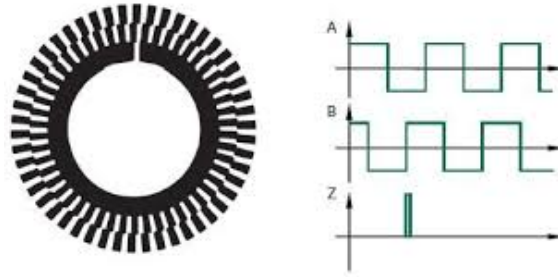
Mutlak enkoderler, diğeri tip enkoderlere göre içyapı olarak oldukça farklı türdedirler. Binary bir ölçüm sistemi kullanarak, ölçüm aralığının ve açının her konumunu, bir cam ve ya plastik bir disk üzerindeki belirli kodlarla tasarlanmış boşluklarla tanımlar. Bu kodlar, kod diski üzerinde farklı izlerin içindeki aydınlık ve karanlık bölgeler şeklinde gösterilir. Bu aydınlık ve karanlık yüzeyler mutlak bir sayısal değeri işaret etmektedir. Bu nedenle pozisyon değeri enerji kesilse daha son konum her zaman bilinmektedir. Bunun yanı sıra sinyallerdeki parazit gürültüler ve beslenme gerilimi kesilmesinden dolayı oluşan sürekli geçersiz değerler elde edilmesi mümkün değildir. Sistem kapalıyken yapılan değişimler bile sistem açıkken ölçüle bilinmektedir. Üç farklı şekilde kodlanmış mutlak enkoder kod diskileri şekil 3.25’ te gösterilmiştir.



Şekil 3.25 Üç farklı şekilde kodlanmış mutlak enkoder kod diski.

Artımsal Tip Enkoder

Bu enkoder tipi, mutlak enkoderde de olduğu gibi bir cam veya plastik disk üzerinde kare dalga oluşturarak bir desen şeklinde tasarlanmıştır. Şeffaf veya saydam olmayan çizgi deseninin durumları ışık algılayıcı birim tarafından darbelere dönüştürülmektedir. Örneğin; algılayıcı saydam kısmında ışığı algıladığında 5V çıkış verirken, saydam olmayan kısımdaki karanlık kısım için 0V gerilim vermektedir. Sinyallerin analizi bir değerlendirme ünitesinde, her gelen darbe sinyali ile yukarı veya aşağı sayılarak konum hesaplanmaktadır. Doğal olarak bu hesaplama biçimi, enerji her kesilip yeniden geldiğinde sistemin sıfırdan başlamasına neden olmaktadır. Artımsal enkoder kod diski şekil 3.26' da gösterilmiştir.



Şekil 3.26 Artımsal enkoder kod diski.

Artımsal enkoderler disk üzerindeki boşluk sayısına göre 512 / 1024 / 2048 / 4096 darbelik artımsal enkoder şeklinde isimlendirilirler. Bu sayılar tam tur olarak adlandırılan 360° derecenin sayısal karşılığını vermektedir. Yani 2048 darbelik enkoder için her sayılan bir darbe $\frac{360}{2048}$ değerinin karşılığı olan 0.1757° ye karşılık gelmektedir.

Şekilde gösterilen kod diskinden de anlaşılacağı gibi artımsal enkoder, konum ve yön gibi bilgileri hesaplamak için üç adet dijital sinyal kullanmaktadır. Bu sinyallerden ikisi A ve B olarak adlandırılan konum sinyalleridir. Bu iki sinyal arasında 90° faz farkı bulunmaktadır. Bunun temel nedeni yön bilgisinin elde edilmesinden kaynaklanmaktadır. Örneğin; her iki sinyalinde 0 olduğu konumda iken, ilk olarak A:1-B:0 oluyorsa mil saat yönünde, A:0-B:1 oluyorsa mil saatin tersi yönde dönüyor demektir. Konum bilgisi ise gelen bu sinyallerin teker teker saymakla elde edilmektedir. Örneğin; 124 adet sinyal sayıldığını varsayalım ve elimizdeki enkoder 1024 darbelik bir enkoder olsun. O halde

enkoder milinin döndüğü açı değeri $\frac{360}{1024} * 124$ şeklinde hesaplanıp 43.5937° ye karşılık gelmektedir.

Artımsal enkoder, diğer tip enkoderler ile kıyaslandığında ucuz olması avantajının yanı sıra dezavantajları da fazladır. Sinyal arızası nedeniyle ölçülmemiş her bir darbe yanlış sinyale sebebiyet vermektedir. Ayrıca konum bilgisinin önemli olduğu noktalarda, enerji kesilmesinden kaynaklanan hatalar sonrasında artımsal enkoder sıfırdan başlarken mutlak enkoder kaldığı konumdan devam eder. Bu da oldukça ciddi problemlere sebep olabilmektedir.

3.2.2.3 Simülasyon

Simülasyon için direksiyon SolidWORKS programında çizilip üç boyutlu yazıcı tarafından oluşturulmuştur. Direksiyon çizim ve 3D yazıcı çıktısı resim 3.7’ de gösterilmiştir. Direksiyon enkoder miline sıkı geçme şeklinde montaj yapılarak direksiyon açısı elde edilmektedir.



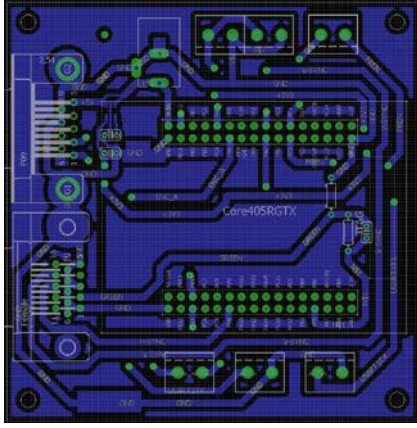
(a)



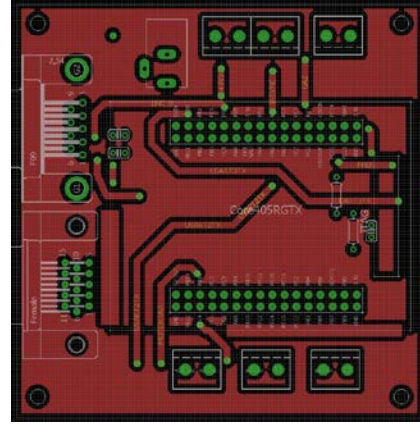
(b)

Resim 3.7 Direksiyon çizim (a) ve 3D yazıcı çıktısı (b).

Tez kapsamında STM32F405RG mikroişlemcisinin kullanıldığı bir devre kartı oluşturulmuştur. Devre alt ve üst katmanı çizimleri şekil 3.27’ de gösterilmiştir. Bu devre kartı üzerinde ekrana görüntü aktarmak için bir VGA çıkışı, direksiyon açısını tespit etmek için kullanılan enkoder için bir RS-232 portu ve gaz pedalı gibi diğer giriş çıkışlardan oluşan portlar mevcuttur.



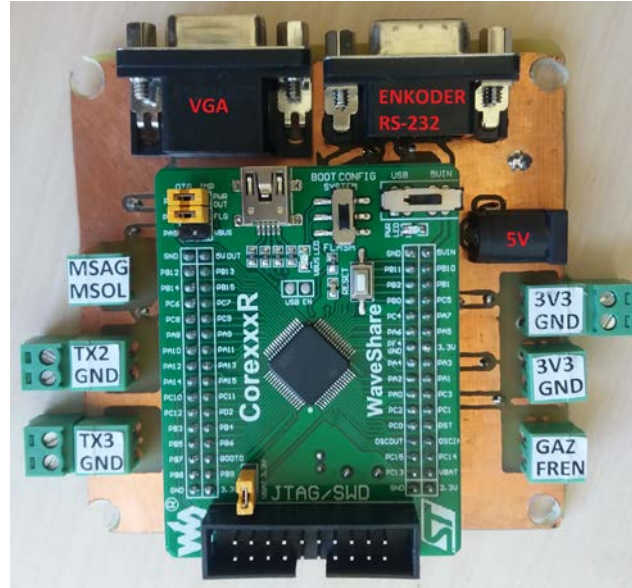
(a)



(b)

Şekil 3.27 Devre alt (a) ve üst (b) katmanı.

Tez kapsamında elektronik diferansiyel sisteminin rahatlıkla anlaşılabilmesi açısından oluşturulan simülör kısmı için oluşturulan elektronik devre resim 3.8' de gösterilmektedir.



Resim 3.8 Devre kartının son hali.

3.2.3 Elektronik Diferansiyel Sisteminin Araç Üzerine Uygulanması

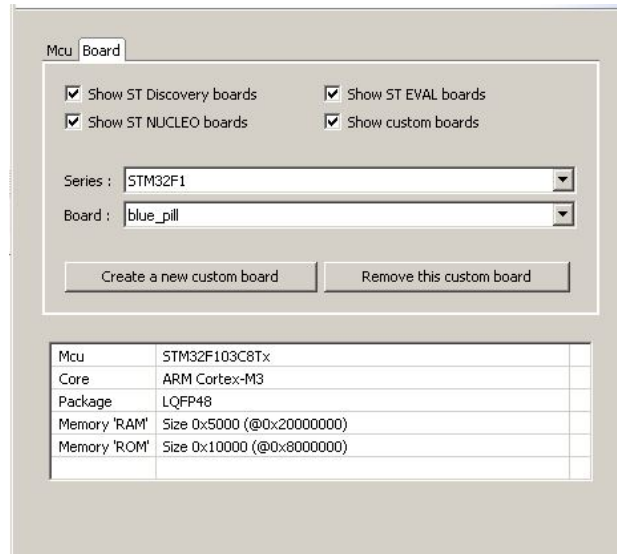
Bu tez çalışmasında, STM32F103 işlemcisi kullanılarak Eclipse programı vasıtasıyla EDS tasarlanmıştır. Bu sistem için araç hızı ve direksiyon açısındaki değişimler dikkate alınarak, aracın arka tekerleklerinin açısız hızları AJP 'ye göre matematiksel olarak STM

mikroişlemcisi üzerinde hesaplanmaktadır. Hesaplanan hız bilgilerine göre program içerisinde PWM sinyalleri üretilerek motor sürücülerine gönderilip motorların istenilen hızlarla dönmesi sağlanmıştır.

3.2.3.1 Eclipse Programı

Tez kapsamında EDS tasarlanırken Eclipse(System Workbench for STM32) yazılımı kullanılmıştır. Eclipse, IBM firması tarafından, Object Technology International(OTI) ile birlikte geliştirdikleri bir proje olarak 2001 'de duyurulan açık kaynak kodlu bir yazılım geliştirme aracıdır(İnt.Kyn.11). Öncelikli olarak sadece Java için geliştirilmiş olsa da şuanda diğer birçok dil içinde derleyici bir ortam sunmaktadır. Tez kapsamındaki proje için C dilini tercih edilmiştir.

Proje oluşturmak için öncelikle hangi dilde yazım yapacağımızı belirtmeli ve sonrasında geliştirici platform olarak kullanacağınız platform olan STM32 platformu seçilir. Platform seçme işlemi ve proje ismi belirlemenin ardından geliştirmek için kullanacağınız mikroişlemcinin gerekli dosyalarını proje klasörüne kopyalaması için mikroişlemciyi seçmek gerekmektedir. Proje oluştururken mikroişlemci seçme ara yüzü şekil 3.28' de gösterilmiştir.



Şekil 3.28 Proje oluştururken mikroişlemci seçme ara yüzü.

İşlemler tamamlandıktan sonra karşımıza algoritmamızı oluşturmamız için boş bir kod sayfası açılacaktır. Proje oluşturulduktan sonra açılan boş sayfa şekil 3.29’ da gösterilmiştir.



```

main.c
3 * @file main.c
10
11
12 #include "stm32f10x.h"
13
14
15 int main(void)
16 {
17
18     for(;;);
19 }
20

```

Şekil 3.29 Proje oluşturulduktan sonra açılan boş sayfa.

Mikroişlemcilerde, yapılacak herhangi bir işlem için programlamaya başlamadan önce yapmamız gereken işlemler vardır. Bunlardan en önemlisi mikroişlemcide yapılacak olan işlemlerin sıralanıp, bu işlemlere en uygun pin konfigürasyonunun yapılmasıdır. Bunun temel sebebi ise daha önceki kısımlarda anlatıldığı gibi mikroişlemciler için her pinin ayrı ayrı özelliklerinin olmasıdır. Şekil 3.20’ de tez kapsamında yapılacak araç için kullanılacak pinlerin STM32F103 işlemcisi üzerinde gösterilmektedir. Mikroişlemci pin konfigürasyonu belirlenmesinin ardından pinlere görevlerinin verilmesi gerekmektedir. Mikroişlemci pinlerinin görev belirlendiği kod kısmı şekil 3.30’ da gösterilmiştir.

```

void GpioC(void){
    GPIO_InitTypeDef gpioconf;

    gpioconf.GPIO_Pin = ilerigeripin|sagsolpin;
    gpioconf.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    gpioconf.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &gpioconf);

    gpioconf.GPIO_Pin = usarttx;
    gpioconf.GPIO_Mode = GPIO_Mode_AF_PP;
    gpioconf.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &gpioconf);

    gpioconf.GPIO_Pin=SagPWMPin|SolPWMPin;
    gpioconf.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_Init(GPIOB, &gpioconf);

    gpioconf.GPIO_Pin=A1pin|A2pin|B1pin|B2pin;
    gpioconf.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_Init(GPIOB, &gpioconf);

    GPIO_ResetBits(GPIOB, A1pin|B1pin|A2pin|B2pin);
}

TIM_ICInitTypeDef timicconf;
timicconf.TIM_Channel=TIM_Channel_2;
timicconf.TIM_ICPrescaler=TIM_ICPSC_DIV1;
timicconf.TIM_ICPolarity=TIM_ICPolarity_Rising;
timicconf.TIM_ICSelection=TIM_ICSelection_DirectTI;
timicconf.TIM_ICFilter=0;

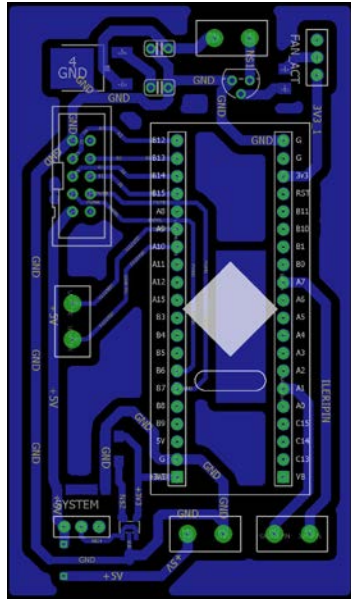
TIM_ICInitTypeDef timicconf;
timicconf.TIM_Channel=TIM_Channel_2;
timicconf.TIM_ICPrescaler=TIM_ICPSC_DIV1;
timicconf.TIM_ICPolarity=TIM_ICPolarity_Rising;
timicconf.TIM_ICSelection=TIM_ICSelection_DirectTI;
timicconf.TIM_ICFilter=0;

```

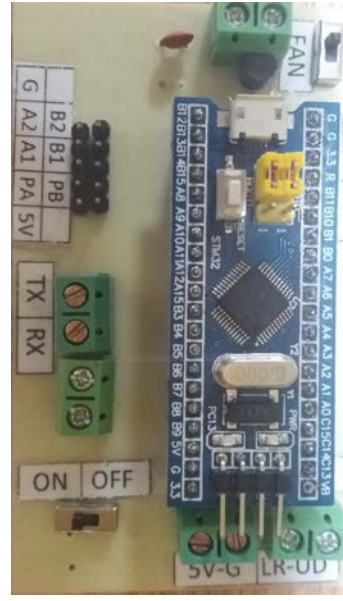
Şekil 3.30 Mikroişlemci pinlerinin görev belirlendiği kod kısmı.

3.2.3.2 Sistem Tasarımı

Sistemin yazılım aşamasının tamamlanmasının ardından, sistemin fiziksel olarak pin bağlantılarının yapılmasını kolaylaştırmak ve bağlantıları daha düzgün yapılabilmek için elektronik devre tasarlanmıştır. Aracı kontrol etmek için kullanılan devre çizimi şekil 3.31-a' da, gerçekleştirilmesi ise şekil 3.31-b' de gösterilmiştir. Elektronik devre ile mikroişlemci ve kullanılan devre elemanlarının bağlantıları daha kompakt hale gelmiştir.



(a)



(b)

Şekil 3.31 Araç için kullanılan devre çizim (a) ve gerçekleştirilmesi (b).

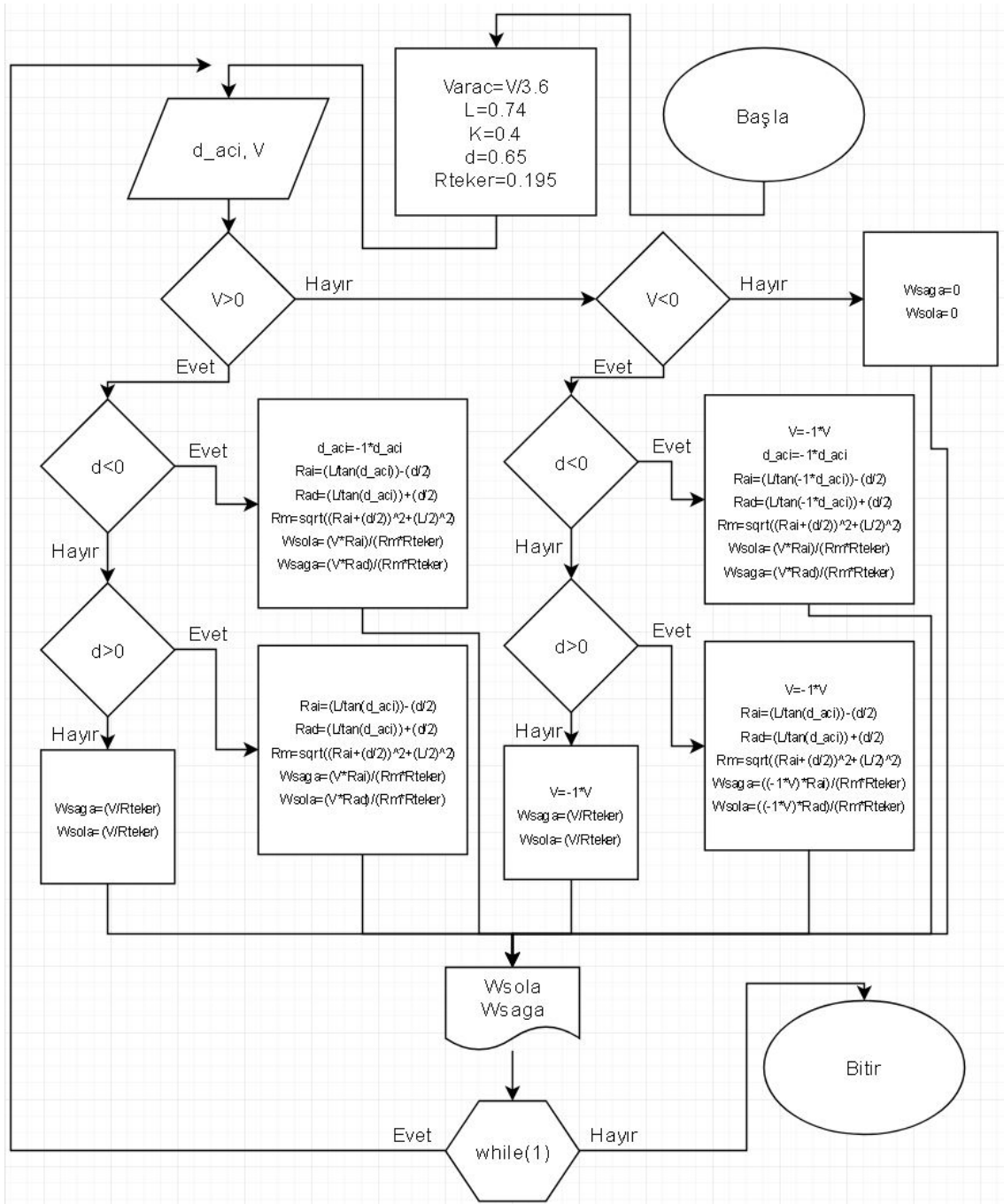
Sistemde iki motorun birden kontrol edilebilmesi için 100 A lik dual doğru akım motor sürücüsü kullanılmıştır. Dual DC motor 100A sürücü kartı resim 3.9' da gösterilmiştir.



Resim 3.9 Dual DC motor 100A sürücü kartı.

Sistemde seri haberleşme kullanılması için RX-TX pin konnektörleri tasarlanan elektronik kart üzerine bırakılmıştır. Bu konnektörler ile telemetri bağlantısı yapılarak elektrikli aracın yaklaşık olarak 2 km'ye kadar uzaktan kontrolü sağlanmaktadır.

Sistemin çalışma mantığına dair temel akış şeması şekil 3.32'de gösterilmektedir.



Şekil 3.32 EDS'in temel akış şeması.

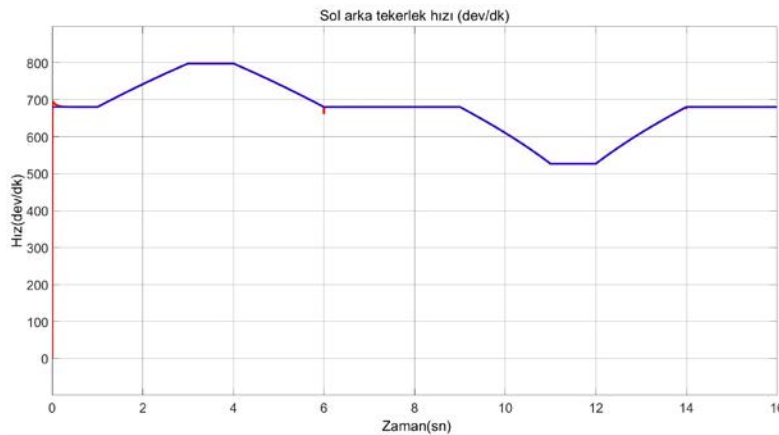
Temel akış şemasından da anlaşıldığı gibi sistem kapalı çevrim kontrol metoduyla direksiyon açısı ve araç hızına bağlı olarak kontrol edilmektedir. EDS, direksiyon açısı ve araç hızına göre tekerleklerin açisal hızlarını hesaplamakta ve motorların olması gereken hızlarda dönmesini sağlamaktadır. Araç hareketi durdurulana kadar kontrol süreci devam etmektedir. Ayrıca dışarıdan gelecek herhangi bir bozucu etkiye göre gerekli kontrolü sağlamaktadır.

4. BULGULAR

Tez çalışma temel olarak matematiksel modelleme bulguları ve gerçekleştirme bulguları olarak iki ana başlıktan oluşmaktadır. Matematiksel modelleme aşamasında EDS ve araç birlikte modellenerek MATLAB/Simulink ortamında çalışması gözlenmiştir. Gerçekleştirme aşamasında ise EDS yazılımsal olarak mikroişlemciye gömülmüş ve araç üzerine bağlanmıştır. Araç farklı sürtünme katsayısına sahip yollar zeminler üzerinde test edilmiş ve gerçek zamanlı test verileri araç üzerindeki hafıza kartına kaydedilmiştir. Modelleme ve gerçekleştirme aşamaları aşağıdaki başlıklarda detaylandırılmıştır.

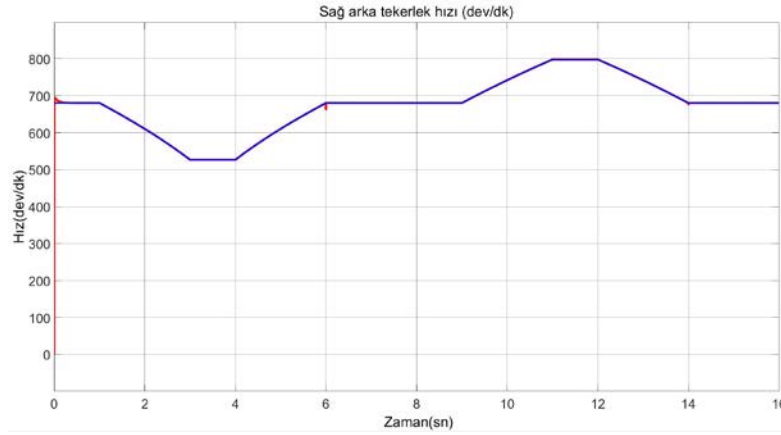
4.1 Matematiksel Modelleme Bulguları

Matematiksel modelleme, mühendislik açısından sistemin daha oluşturulmadan önce oluşum sonrasında meydana çıkabilecek maddi kayıpları önlemek için çok önemli bir buluş olmuştur. Sistemler tamamıyla matematiksel olarak ifade edilemeseler de önemli olan çoğu parametre bu modellerden elde edilmektedir. Diğer bir avantajı ise daha henüz ortada herhangi bir sistem yok iken bile sisteme en uygun kontrol algoritmasının seçilmesine olanak tanımaktadır. Tez çalışmasında oluşturulacak EDS için, “Bilgisayar Ortamında Matematiksel Modellenmesi” başlığı altında detaylı şekilde modellemeye yer verilmiştir. Matematiksel model aşamasında EDS’ in standart tip mekanik diferansiyel sistemi gibi istenilen hızlara karşı doğrudan tepki verip veremeyeceği gözlenmek istenmiştir. Sol tekerleklerin referans hız ve model hızı şekil 4.1’de gösterilmiştir.



Şekil 4.1 Sol arka tekerleklerin referans hız ve model hızı.

Sağ tekerleklerin referans hız ve model hızı şekil 4.2’de gösterilmiştir.



Şekil 4.2 Sağ arka tekerleklerin referans hız ve model hızı.

Oluşturulan model, bilgisayar ortamında modellenen elektrikli araç modeline bağlanarak 100 km/h hızla hareket halindeyken öncelikle 25° sağa doğru daha sonrasında ise 25° sola doğru yönelimde tekerleklerin dönmesi gereken referans hızları elektronik diferansiyel modelinden alınarak, PID kontrolör yardımıyla gerçek hızları referansa yakınsanmıştır. Şekil 4.1’ de sol ve şekil 4.2’ de sağ tekerleğin verilen referansa sağladıkları uyum görülmektedir.

4.2 Gerçek Zamanlı Uygulama Bulguları

Tez kapsamında oluşturulan elektrikli araç prototipi sürtünme katsayısı farklı olan çeşitli yol şartlarında, 5°, 15° ve 25° direksiyon açılarıyla dairesel yol yörüngesinde denenmiş ve sonuçlar gözlenmiştir. Yol şartlarının yaklaşık sürtünme katsayıları Çizelge 4.1’de verilmiştir.

Çizelge 4.1 Elektrikli araç prototipinin test edildiği yol şartlarının yaklaşık sürtünme katsayıları.

Zemin	Yaklaşık sürtünme katsayısı
Asfalt	0.9 - 1.05
Islak zemin	0.75 - 0.85
Zemin 1	0.55 – 0.70
Zemin 2	0.35 – 0.50

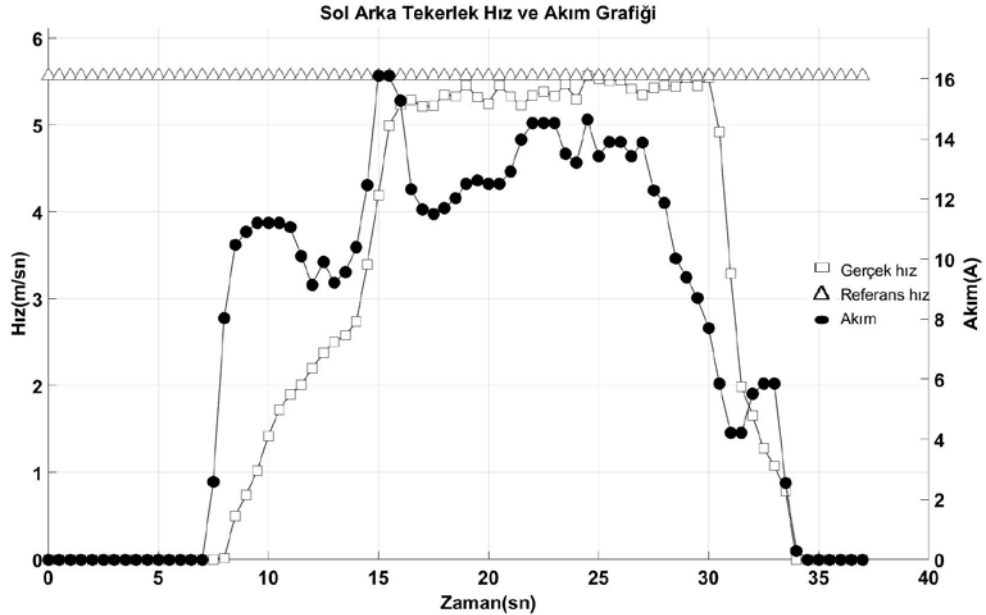
Elektrikli araç prototipi dört farklı sürtünme katsayısına sahip yol şartlarında test edilmiştir. Tekerleklerin referans hızı, gerçek hızı ve elektrikli aracın tahriki için kullanılan doğru akım motorlarının akım sonuçları dört farklı yol şartı ve farklı direksiyon açıları için ayrı ayrı alt başlıklar halinde sunulmuştur.

4.2.1 Asfalt Zemin

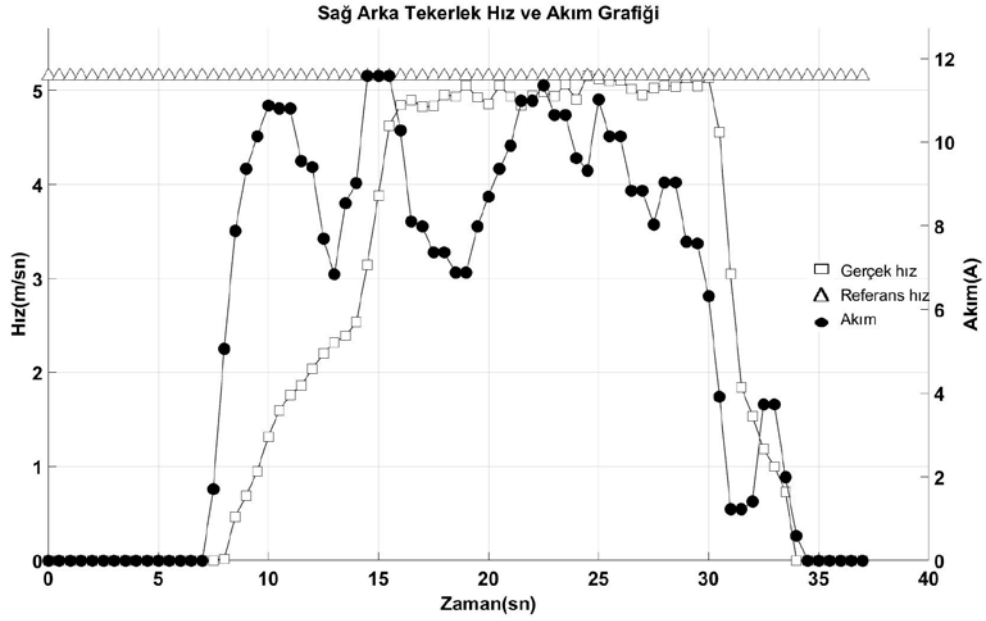
Elektrikli araç prototipi asfalt zemin üzerinde sırasıyla 5° , 15° ve 25° sağa doğru yönelimle tekerleklerin hızları, tekerleklere bağlı motor akımları ve referans hızları verilecektir. Asfalt zemin üzerinde sürtünme katsayısı çizelge 4.1’de de verildiği gibi yaklaşık 1 civarındadır.

4.2.1.1 Asfalt zeminde 5° sağa yönelim için alınan veriler

Şekil 4.3’ te sol ve şekil 4.4’de sağ tekerlek için 5° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.3 Asfalt zeminde 5° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.4 Asfalt zeminde 5° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 5° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.2' de ise aynı veriler tablo halinde sunulmuştur.

Çizelge 4.2 Sol ve sağ tekerleklerin asfalt zeminde 5° sağa yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
5.2306	5.5635	12.8	6.3645	4.8435	5.1518	12.8	6.3652
5.2875	5.5635	9.47	5.2199	4.8962	5.1518	9.47	5.2204
5.2109	5.5635	11.84	6.7666	4.8252	5.1518	11.84	6.7686
5.2207	5.5635	11.47	6.5662	4.8344	5.1518	11.47	6.5654
5.349	5.5635	11.54	4.0101	4.9532	5.1518	11.54	4.0095
5.3284	5.5635	12.72	4.4122	4.934	5.1518	12.72	4.4143
5.4585	5.5635	12.5	1.9236	5.0545	5.1518	12.5	1.925
5.3181	5.5635	12.5	4.6144	4.9245	5.1518	12.5	4.6157
5.2472	5.5635	13.32	6.028	4.8589	5.1518	13.32	6.0281
5.4549	5.5635	11.84	1.9909	5.0512	5.1518	11.84	1.9916
5.3284	5.5635	12.5	4.4122	4.934	5.1518	12.5	4.4143
5.224	5.5635	15.61	6.4989	4.8374	5.1518	15.61	6.4994
5.3421	5.5635	15.46	4.1444	4.9468	5.1518	15.46	4.1441
5.3804	5.5635	13.61	3.4031	4.9822	5.1518	13.61	3.4041
5.3318	5.5635	13.02	4.3456	4.9372	5.1518	13.02	4.3466
5.4693	5.5635	17.02	1.7223	5.0646	5.1518	17.02	1.7218
5.2943	5.5635	13.39	5.0847	4.9025	5.1518	13.39	5.0852

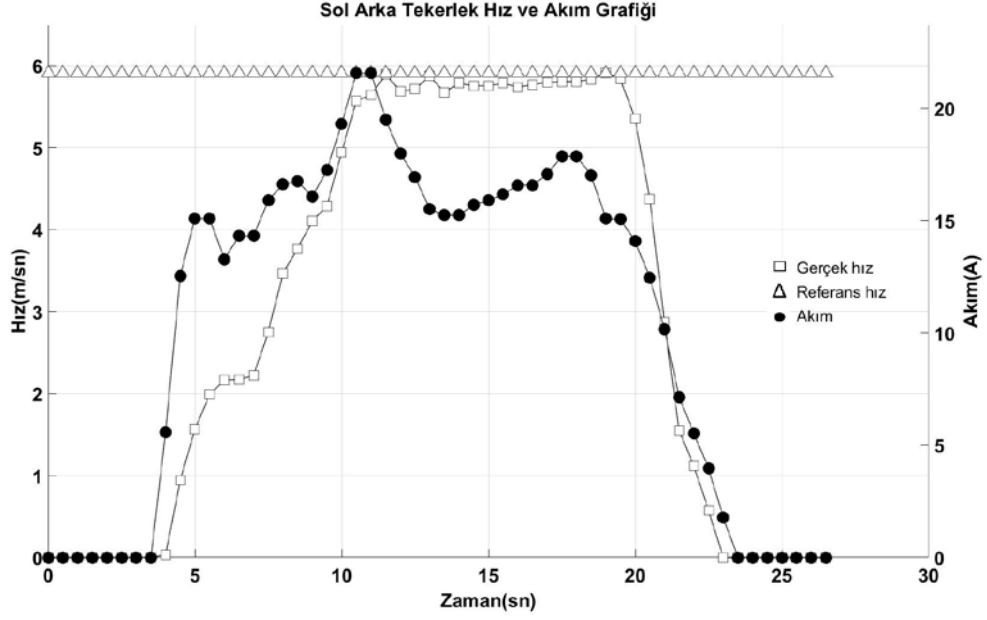
Çizelge 4.2 (Devam) Sol ve sağ tekerleklerin asfalt zeminde 5° sağa yönelimdeki gerçek zamanlı verileri

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
5.5688	5.5635	12.8	0.0952	5.1566	5.1518	12.8	0.0931
5.5315	5.5635	15.91	0.5785	5.1222	5.1518	15.91	0.5779
5.5021	5.5635	13.46	1.1159	5.0949	5.1518	13.46	1.1168
5.5168	5.5635	14.35	0.8465	5.1085	5.1518	14.35	0.8476
5.4191	5.5635	11.17	2.6646	5.0181	5.1518	11.17	2.6644
5.349	5.5635	13.39	4.0101	4.9532	5.1518	13.39	4.0095
5.4263	5.5635	15.39	2.5284	5.0247	5.1518	15.39	2.5295
5.4585	5.5635	9.69	1.9236	5.0545	5.1518	9.69	1.925
5.4406	5.5635	10.36	2.2589	5.0379	5.1518	10.36	2.2609
5.5426	5.5635	9.1	0.3771	5.1325	5.1518	9.1	0.376
5.4477	5.5635	7.1	2.1257	5.0446	5.1518	7.1	2.125
5.5426	5.5635	8.29	0.3771	5.1325	5.1518	8.29	0.376

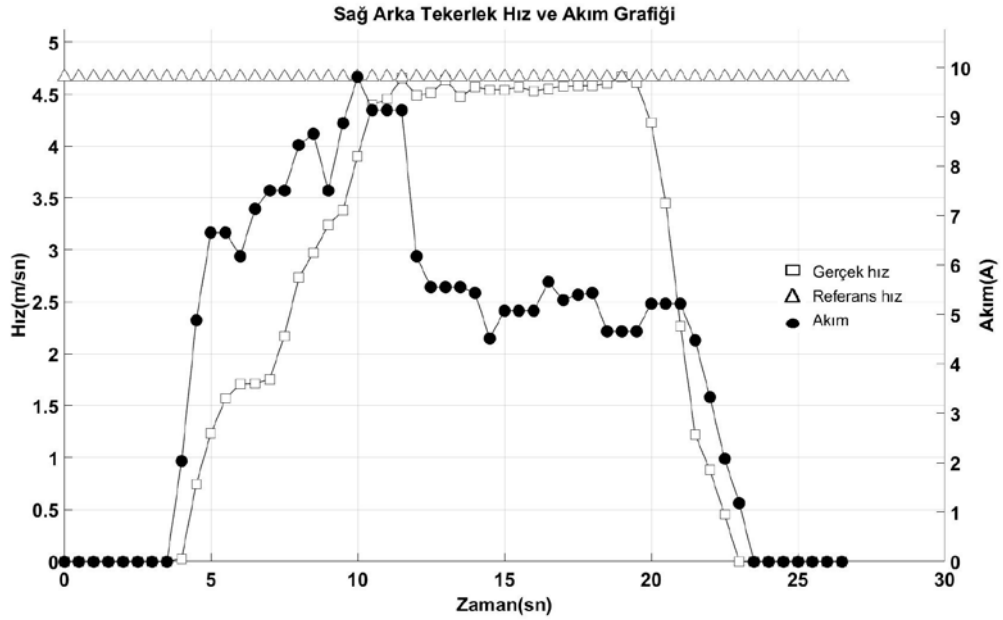
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %3.32 iken sağ tekerleğin ise %3.32 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 5.56 m/sn iken %3.32' lik bir kaymayı $\frac{5.56 \cdot 3.32}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.184 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 5.15 m/sn iken %3.32' lik bir kaymayı $\frac{5.15 \cdot 3.32}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.170 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.1.2 Asfalt zeminde 15° sağa yönelim için alınan veriler

Şekil 4.5' te sol ve şekil 4.6'de sağ tekerlek için 15° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.5 Asfalt zeminde 15° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.6 Asfalt zeminde 15° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 15° sağa dönüş için gerçek hız ve motor akımları

verilmiştir. Çizelge 4.3’ de ise aynı veriler tablo halinde sunulmuştur.

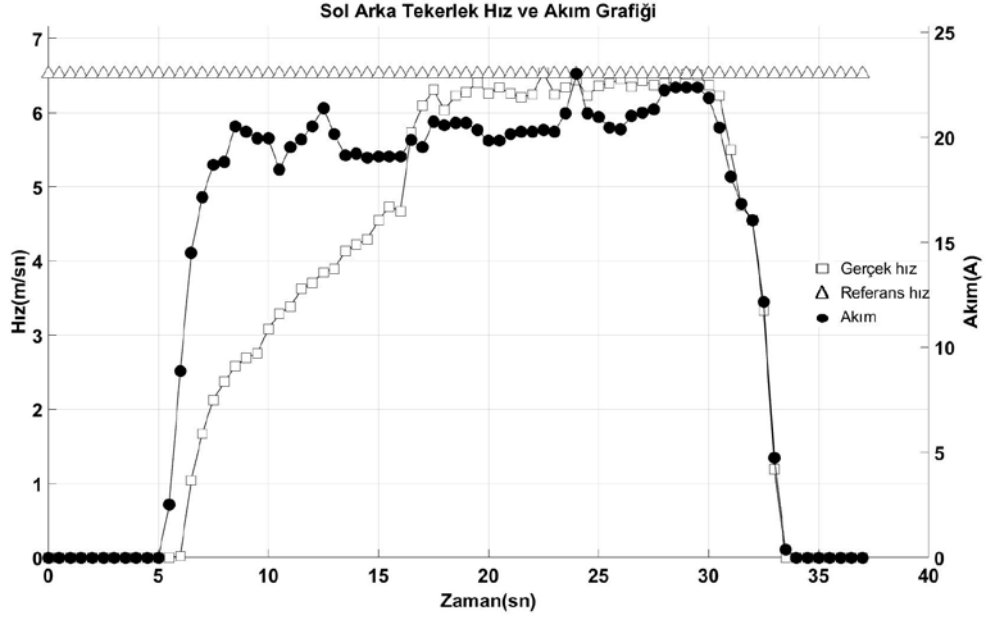
Çizelge 4.3 Sol ve sağ tekerleklerin asfalt zeminde 15° sağa yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
5.5685	5.9076	12.8	6.0896	4.4521	4.6636	12.8	4.7506
5.6397	5.9076	9.47	4.7503	4.6523	4.6636	9.47	0.2429
5.8933	5.9076	11.84	0.2426	4.4886	4.6636	11.84	3.8988
5.686	5.9076	11.47	3.8973	4.5098	4.6636	11.47	3.4104
5.7128	5.9076	11.54	3.4099	4.6354	4.6636	11.54	0.6084
5.8719	5.9076	12.72	0.608	4.4781	4.6636	12.72	4.1424
5.6727	5.9076	12.5	4.1409	4.5663	4.6636	12.5	2.1308
5.7843	5.9076	12.5	2.1316	4.5419	4.6636	12.5	2.6795
5.7535	5.9076	13.32	2.6784	4.5419	4.6636	13.32	2.6795
5.7535	5.9076	11.84	2.6784	4.5663	4.6636	11.84	2.1308
5.7843	5.9076	12.5	2.1316	4.5285	4.6636	12.5	2.9833
5.7364	5.9076	15.61	2.9845	4.5527	4.6636	15.61	2.4359
5.7671	5.9076	15.46	2.4362	4.5744	4.6636	15.46	1.95
5.7947	5.9076	13.61	1.9483	4.5827	4.6636	13.61	1.7653
5.8051	5.9076	13.02	1.7657	4.5827	4.6636	13.02	1.7653
5.8051	5.9076	17.02	1.7657	4.602	4.6636	17.02	1.3385
5.8295	5.9076	13.39	1.3397	4.6664	4.6636	13.39	0.06
5.9112	5.9076	12.8	0.0609	4.6075	4.6636	12.8	1.2176
5.8365	5.9076	15.91	1.2182	4.2237	4.6636	15.91	10.415

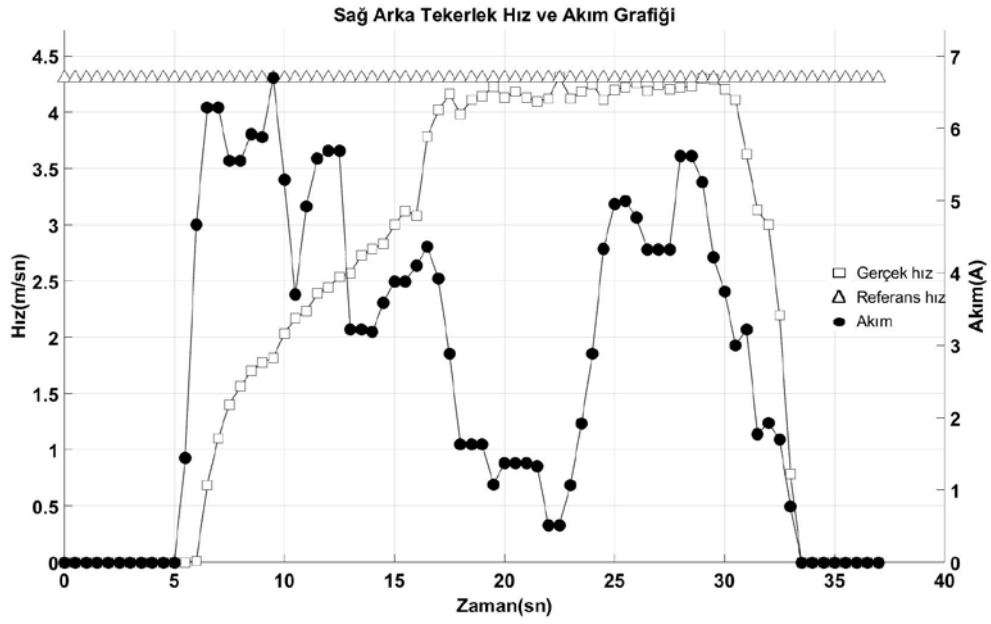
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %2.43 iken sağ tekerleğin ise %2.66 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 5.90 m/sn iken %2.43’ lik bir kaymayı $\frac{5.90 \cdot 2.43}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.143 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 4.66 m/sn iken %2.66’ lik bir kaymayı $\frac{4.66 \cdot 2.66}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.123 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.1.3 Asfalt zeminde 25° sağa yönelim için alınan veriler

Şekil 4.7’ de sol ve şekil 4.8’de sağ tekerlek için 25° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.7 Asfalt zeminde 25° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.8 Asfalt zeminde 25° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 25° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.4' de ise aynı veriler tablo halinde sunulmuştur.

Çizelge 4.4 Sol ve sağ tekerleklerin asfalt zeminde 25° sağa yönelimdeki gerçek zamanlı verileri

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
6.2477	6.5204	12.8	4.3648	4.1237	4.3036	12.8	4.3626
6.54	6.5204	9.47	0.2997	4.3166	4.3036	9.47	0.3012
6.2443	6.5204	11.84	4.4216	4.1214	4.3036	11.84	4.4208
6.3364	6.5204	11.47	2.9039	4.1822	4.3036	11.47	2.9028
6.4348	6.5204	11.54	1.3303	4.2472	4.3036	11.54	1.3279
6.2242	6.5204	12.72	4.7588	4.1082	4.3036	12.72	4.7563
6.3607	6.5204	12.5	2.5107	4.1983	4.3036	12.5	2.5082
6.3958	6.5204	12.5	1.9482	4.2214	4.3036	12.5	1.9472
6.4491	6.5204	13.32	1.1056	4.2566	4.3036	13.32	1.1042
6.3503	6.5204	11.84	2.6786	4.1914	4.3036	11.84	2.6769
6.4312	6.5204	12.5	1.387	4.2448	4.3036	12.5	1.3852
6.3677	6.5204	15.61	2.398	4.2029	4.3036	15.61	2.396
6.3958	6.5204	15.46	1.9482	4.2214	4.3036	15.46	1.9472
6.4135	6.5204	13.61	1.6668	4.2331	4.3036	13.61	1.6654
6.5143	6.5204	13.02	0.0936	4.2996	4.3036	13.02	0.093
6.507	6.5204	17.02	0.2059	4.2948	4.3036	17.02	0.2049
6.3747	6.5204	13.39	2.2856	4.2075	4.3036	13.39	2.284

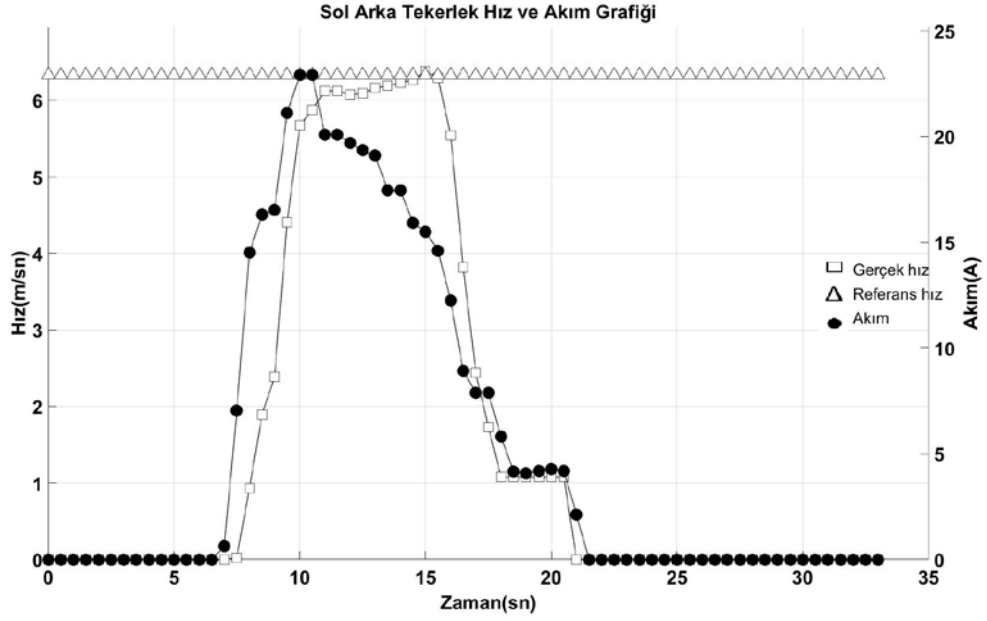
Şekil ve çizelgeler yorumlandığında sol tekerlek içi kayma ortalama %2.13 iken sağ tekerleğin ise %2.13 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 6.52 m/sn iken %2.13' lik bir kaymayı $\frac{6.52*2.13}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.138 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 4.30 m/sn iken %2.13' lik bir kaymayı $\frac{4.30*2.13}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.091 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.2 Islak Zemin

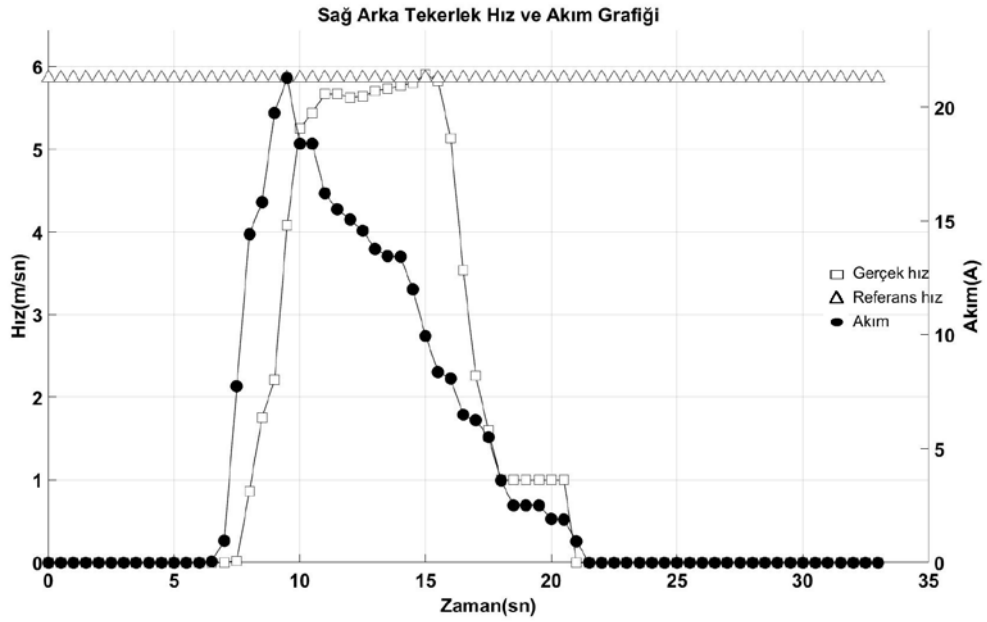
Elektrikli araç prototipi ıslak zemin üzerinde sırasıyla 5°, 15° ve 25° sağa doğru yönelimle tekerleklerin hızları, tekerleklere bağlı motor akımları ve referans hızları verilecektir.

4.2.2.1 Islak zeminde 5° sağı yönelim için alınan veriler

Şekil 4.9' te sol ve şekil 4.10'de sağ tekerlek için 5° sağı dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.9 Islak zeminde 5° sağı yönelimde sol tekerlek hız ve akım grafiğı.



Şekil 4.10 Islak zeminde 5° sağı yönelimde sağ tekerlek hız ve akım grafiğı.

Yukarıda sağ ve sol tekerleğin 5° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.5' de ise aynı veriler tablo halinde sunulmuştur.

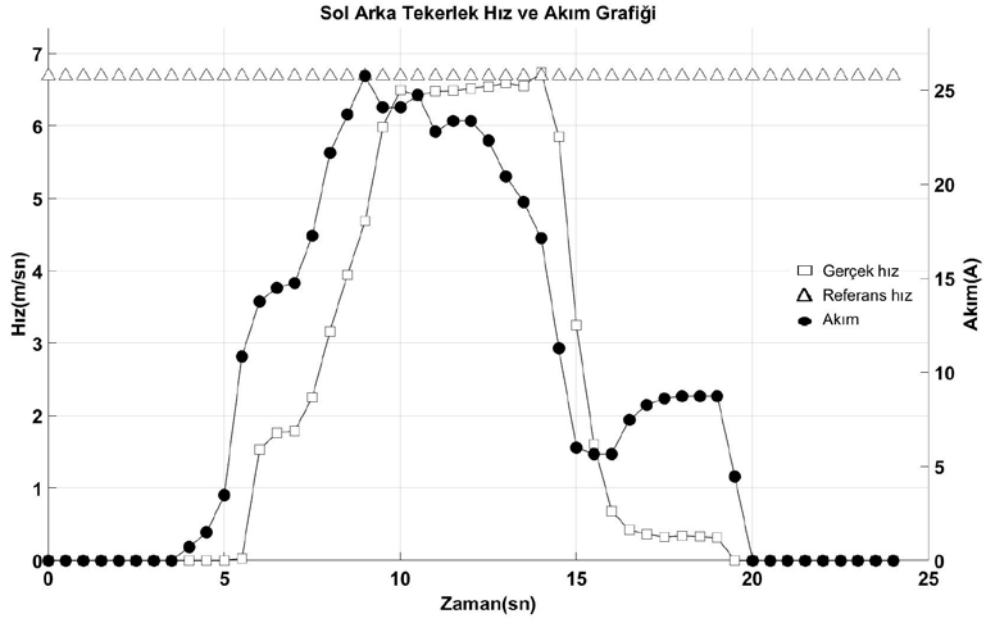
Çizelge 4.5 Sol ve sağ tekerleklerin ıslak zeminde 5° sağa yönelimdeki gerçek zamanlı verileri

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
5.8775	6.3297	12.8	7.6937	5.4425	5.8613	12.8	7.695
6.1257	6.3297	9.47	3.3302	5.6723	5.8613	9.47	3.332
6.1211	6.3297	11.84	3.4079	5.6681	5.8613	11.84	3.4085
6.0761	6.3297	11.47	4.1737	5.6265	5.8613	11.47	4.1731
6.0896	6.3297	11.54	3.9428	5.6389	5.8613	11.54	3.944
6.1622	6.3297	12.72	2.7182	5.7061	5.8613	12.72	2.7199
6.1898	6.3297	12.5	2.2602	5.7318	5.8613	12.5	2.2593
6.2318	6.3297	12.5	1.571	5.7706	5.8613	12.5	1.5718
6.2696	6.3297	13.32	0.9586	5.8056	5.8613	13.32	0.9594
6.3809	6.3297	11.84	0.8024	5.9087	5.8613	11.84	0.8022
6.2935	6.3297	12.5	0.5752	5.8277	5.8613	12.5	0.5766

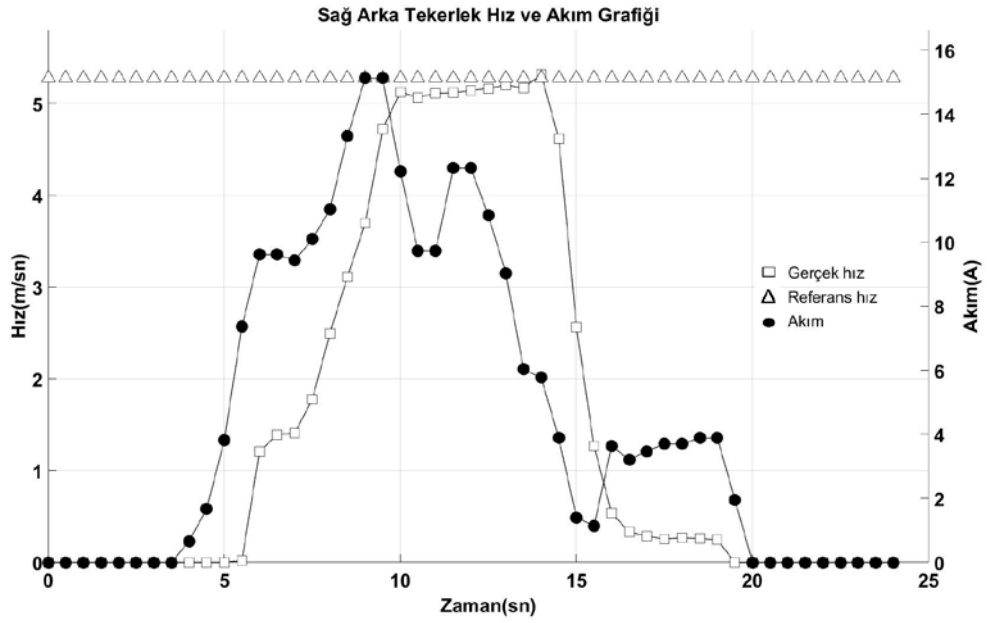
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %2.85 iken sağ tekerleğin ise %2.85 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 6.32 m/sn iken %2.85' lik bir kaymayı $\frac{6.32 \cdot 2.85}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.180 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 5.86 m/sn iken %2.85' lik bir kaymayı $\frac{5.86 \cdot 2.85}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.167 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.2.2 Islak zeminde 15° sağa yönelim için alınan veriler

Şekil 4.11' te sol ve şekil 4.12' de sağ tekerlek için 15° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.11 Islak zeminde 15° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.12 Islak zeminde 15° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 15° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.6' de ise aynı veriler tablo halinde sunulmuştur.

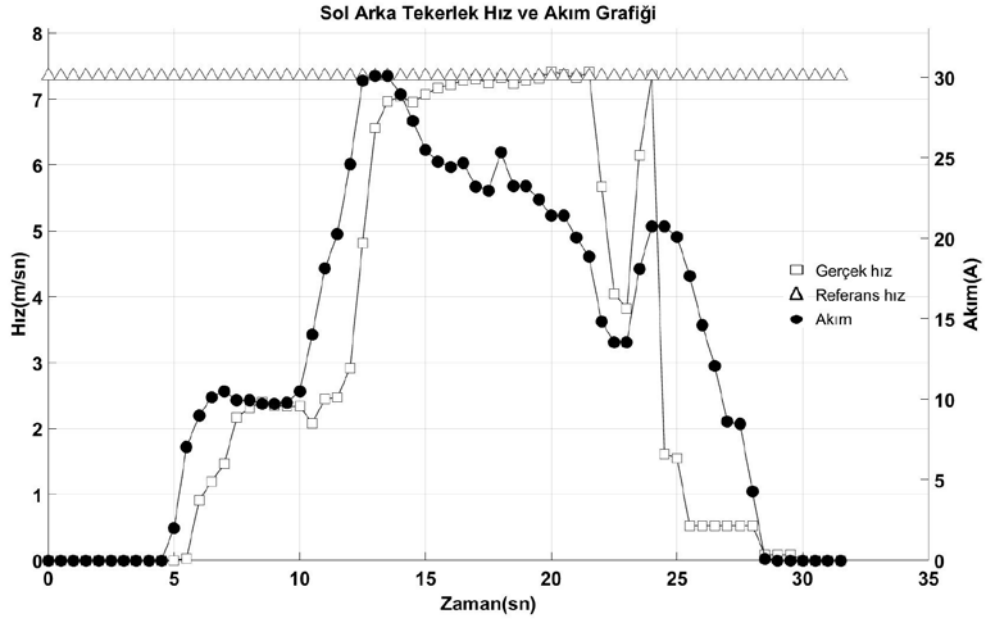
Çizelge 4.6 Sol ve sağ tekerleklerin ıslak zeminde 15° sağa yönelimdeki gerçek zamanlı verileri

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
6.4929	6.6861	12.8	2.9756	5.1256	5.2781	12.8	2.9753
6.4198	6.6861	9.47	4.1481	5.0679	5.2781	9.47	4.1477
6.4755	6.6861	11.84	3.2523	5.1119	5.2781	11.84	3.2512
6.4842	6.6861	11.47	3.1137	5.1187	5.2781	11.47	3.1141
6.5147	6.6861	11.54	2.631	5.1428	5.2781	11.54	2.6309
6.5454	6.6861	12.72	2.1496	5.1671	5.2781	12.72	2.1482
6.5899	6.6861	12.5	1.4598	5.2022	5.2781	12.5	1.459
6.5498	6.6861	12.5	2.081	5.1706	5.2781	12.5	2.0791
6.7363	6.6861	13.32	0.7452	5.3178	5.2781	13.32	0.7465

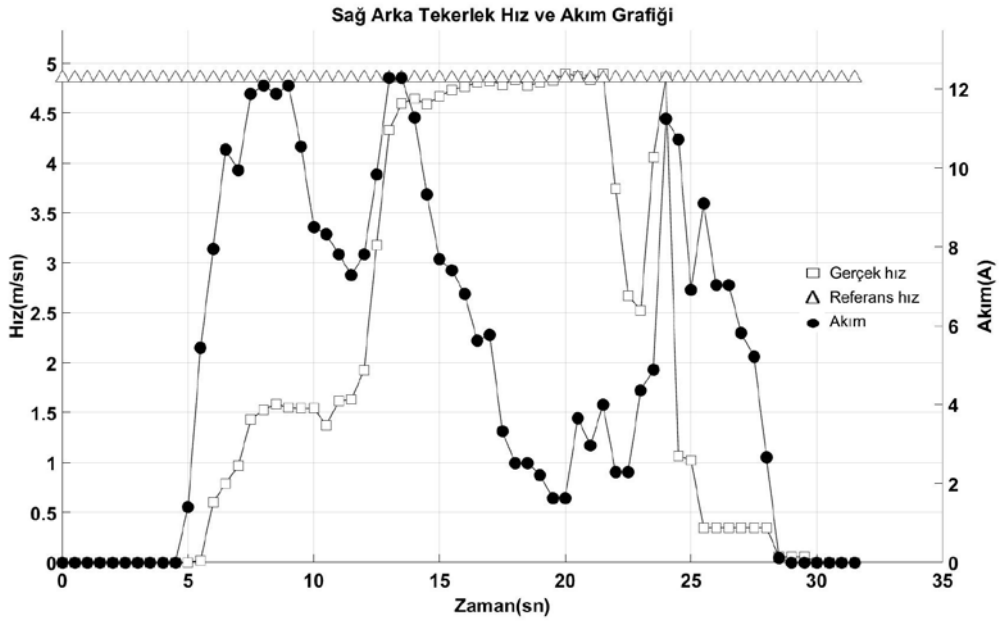
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %2.50 iken sağ tekerleğin ise %2.50 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 6.68 m/sn iken %2.50' lik bir kaymayı $\frac{6.68*2.50}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.167 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 5.27 m/sn iken %2.50' lik bir kaymayı $\frac{5.27*2.50}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.131 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.2.1 Islak zeminde 25° sağa yönelim için alınan veriler

Şekil 4.13' te sol ve şekil 4.14'de sağ tekerlek için 25° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.13 Islak zeminde 25° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.14 Islak zeminde 25° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 25° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.7' de ise aynı veriler tablo halinde sunulmuştur.

Çizelge 4.7 Sol ve sağ tekerleklerin ıslak zeminde 25° sağa yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
6.9681	7.3489	12.8	5.4649	4.5992	4.8505	12.8	5.464
7.0358	7.3489	9.47	4.4501	4.6438	4.8505	9.47	4.4511
6.9598	7.3489	11.84	5.5907	4.5937	4.8505	11.84	5.5903
7.07	7.3489	11.47	3.9448	4.6664	4.8505	11.47	3.9452
7.175	7.3489	11.54	2.4237	4.7357	4.8505	11.54	2.4241
7.2151	7.3489	12.72	1.8544	4.7622	4.8505	12.72	1.8542
7.2922	7.3489	12.5	0.7775	4.8131	4.8505	12.5	0.777
7.3106	7.3489	12.5	0.5239	4.8252	4.8505	12.5	0.5243
7.2467	7.3489	13.32	1.4103	4.783	4.8505	13.32	1.4112
7.3291	7.3489	11.84	0.2702	4.8374	4.8505	11.84	0.2708
7.2331	7.3489	12.5	1.601	4.7741	4.8505	12.5	1.6003
7.2877	7.3489	15.61	0.8398	4.8101	4.8505	15.61	0.8399
7.3198	7.3489	15.46	0.3976	4.8313	4.8505	15.46	0.3974
7.4181	7.3489	13.61	0.9329	4.8962	4.8505	13.61	0.9334
7.3757	7.3489	13.02	0.3634	4.8682	4.8505	13.02	0.3636
7.3244	7.3489	17.02	0.3345	4.8344	4.8505	17.02	0.333
7.4134	7.3489	13.39	0.87	4.8931	4.8505	13.39	0.8706

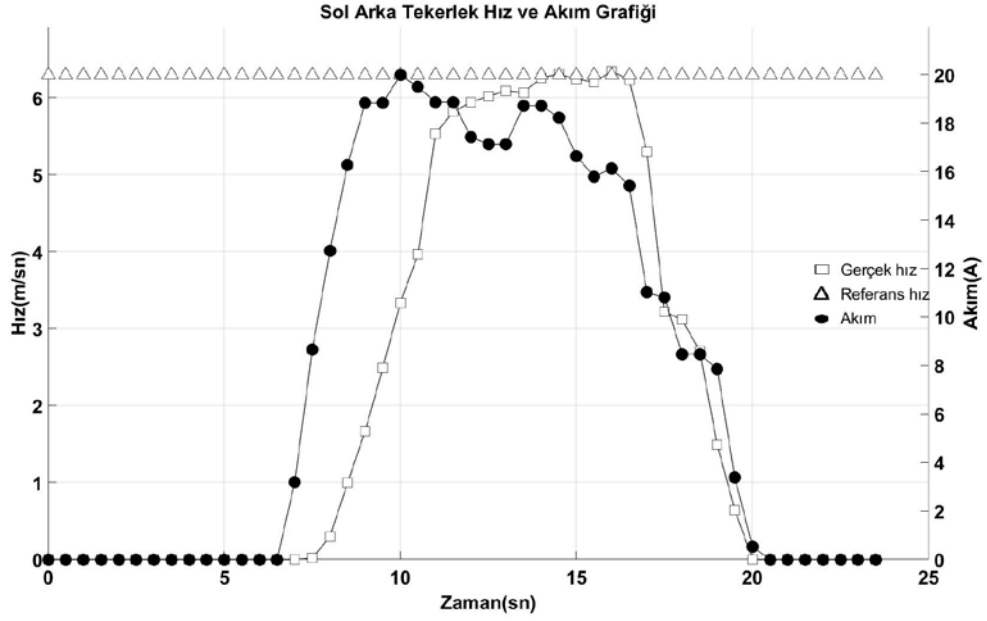
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %1.88 iken sağ tekerleğin ise %1.88 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 7.34 m/sn iken %1.88' lik bir kaymayı $\frac{7.34 \cdot 1.88}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.137 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 4.85 m/sn iken %1.88' lik bir kaymayı $\frac{4.85 \cdot 1.88}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.091 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.3 Zemin 1 Şartlarında

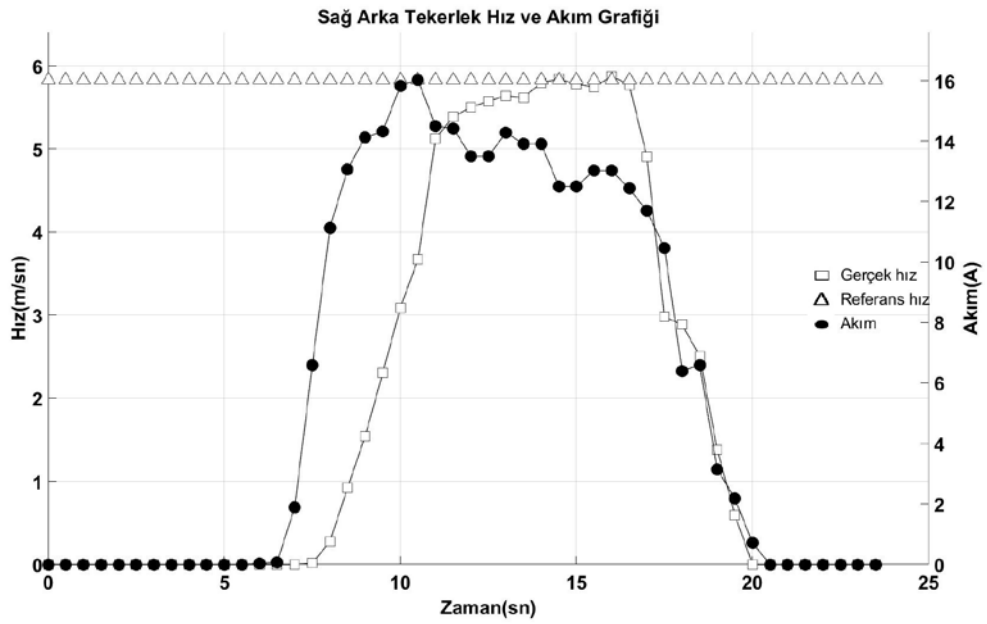
Elektrikli araç prototipi ıslak zemin üzerinde sırasıyla 5°, 15° ve 25° sağa doğru yönelimle tekerleklerin hızları, tekerleklere bağlı motor akımları ve referans hızları verilecektir.

4.2.3.1 Zemin 1 Şartlarında 5° sağı yönelim için alınan veriler

Şekil 4.15' te sol ve şekil 4.16'de sağ tekerlek için 5° sağı dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.15 Zemin 1 şartlarında 5° sağı yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.16 Zemin 1 şartlarında 5° sağı yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 5° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.8’ de ise aynı veriler tablo halinde sunulmuştur.

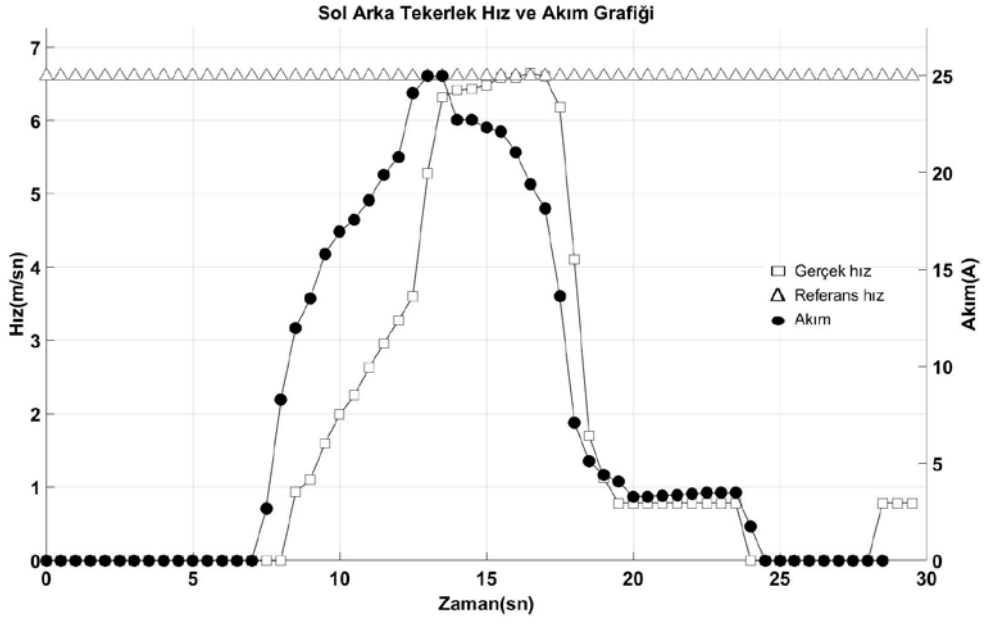
Çizelge 4.8 Sol ve sağ tekerleklerin zemin 1 şartlarında 5° sağa yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
6.0896	6.2928	12.8	3.3368	5.6389	5.8271	12.8	3.3375
6.0672	6.2928	9.47	3.7184	5.6182	5.8271	9.47	3.7183
6.2601	6.2928	11.84	0.5224	5.7968	5.8271	11.84	0.5227
6.3079	6.2928	11.47	0.2394	5.8411	5.8271	11.47	0.2397
6.2412	6.2928	11.54	0.8268	5.7793	5.8271	11.54	0.8271
6.2038	6.2928	12.72	1.4346	5.7447	5.8271	12.72	1.4344
6.3417	6.2928	12.5	0.7711	5.8724	5.8271	12.5	0.7714
6.2365	6.2928	12.5	0.9027	5.775	5.8271	12.5	0.9022

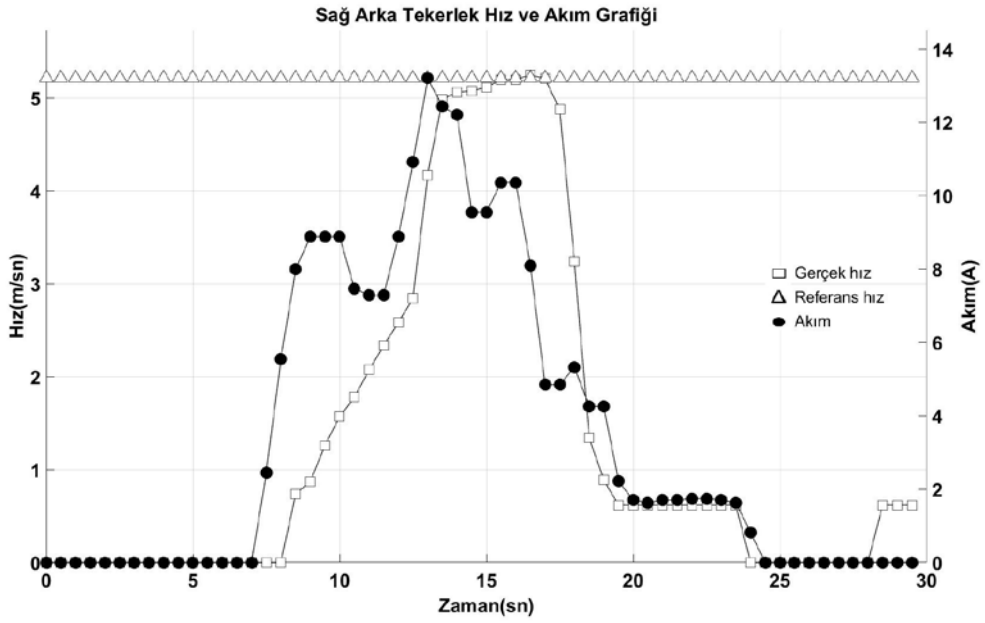
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %1.46 iken sağ tekerleğin ise %1.46 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 6.29 m/sn iken %1.46’ lik bir kaymayı $\frac{6.29 \cdot 1.46}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.091 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 5.82 m/sn iken %1.46’ lik bir kaymayı $\frac{5.82 \cdot 1.46}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.084 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.3.2 Zemin 1 Şartlarında 15° sağa yönelim için alınan veriler

Şekil 4.17’ te sol ve şekil 4.18’de sağ tekerlek için 15° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.17 Zemin 1 şartlarında 15° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.18 Zemin 1 şartlarında 15° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 15° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.9' de ise aynı veriler tablo halinde sunulmuştur.

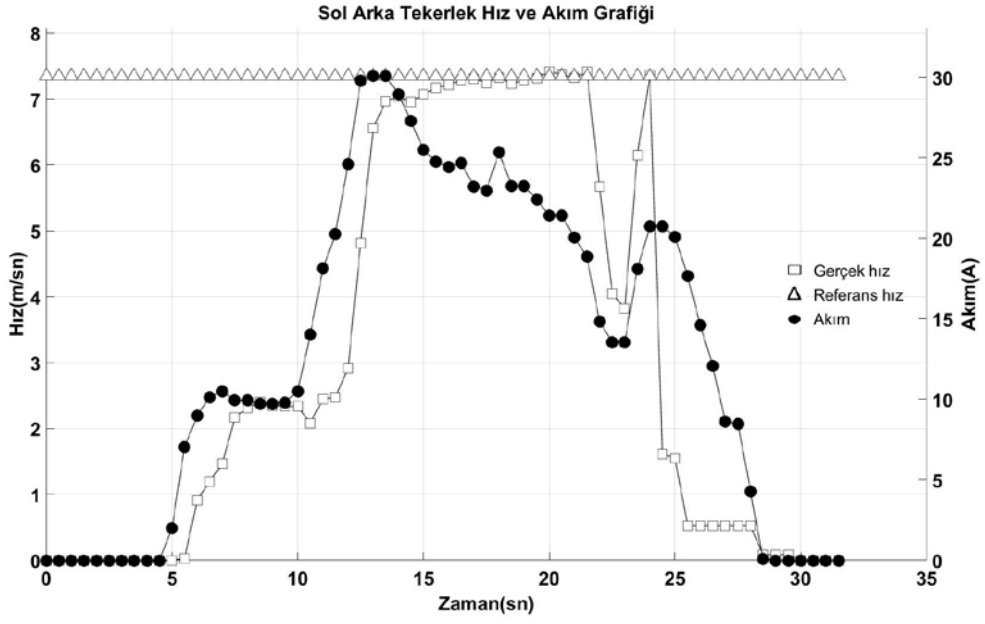
Çizelge 4.9 Sol ve sağ tekerleklerin zemin 1 şartlarında 15° sağa yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
6.3153	6.6033	12.8	4.5604	4.9854	5.2128	12.8	4.5613
6.4113	6.6033	9.47	2.9947	5.0612	5.2128	9.47	2.9953
6.4283	6.6033	11.84	2.7223	5.0746	5.2128	11.84	2.7234
6.4798	6.6033	11.47	1.9059	5.1153	5.2128	11.47	1.906
6.581	6.6033	11.54	0.3389	5.1951	5.2128	11.54	0.3407
6.5854	6.6033	12.72	0.2718	5.1987	5.2128	12.72	0.2712
6.6486	6.6033	12.5	0.6813	5.2485	5.2128	12.5	0.6802
6.5944	6.6033	12.5	0.135	5.2057	5.2128	12.5	0.1364

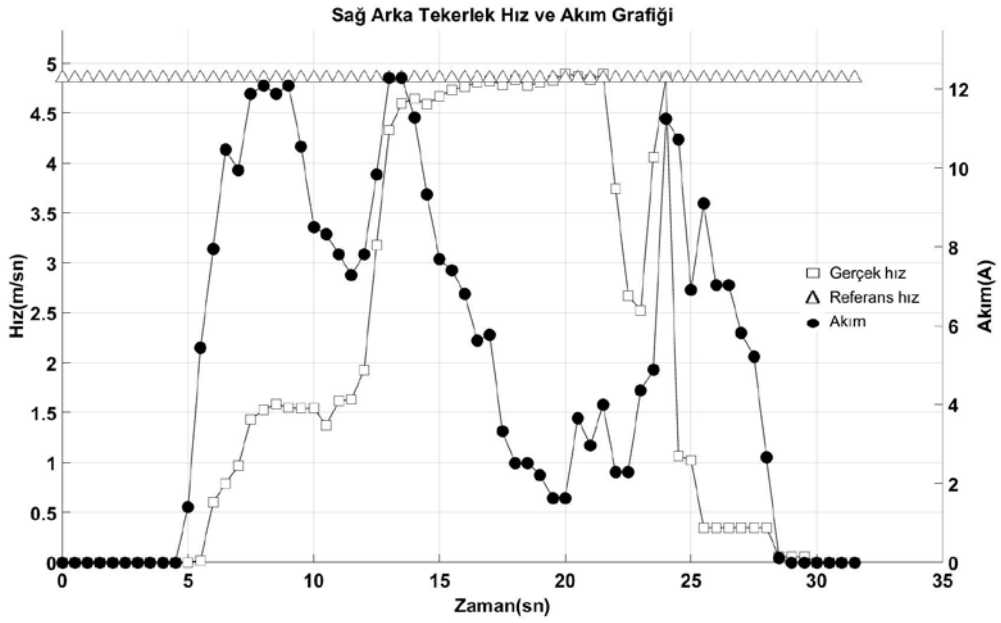
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %1.70 iken sağ tekerleğin ise %1.70 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 6.60 m/sn iken %1.70' lik bir kaymayı $\frac{6.60*1.70}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.112 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 5.21 m/sn iken %1.70' lik bir kaymayı $\frac{5.21*1.70}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.088 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.3.3 Zemin 1 Şartlarında 25° sağa yönelim için alınan veriler

Şekil 4.19' te sol ve şekil 4.20' de sağ tekerlek için 25° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.19 Zemin 1 şartlarında 25° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.20 Zemin 1 şartlarında 25° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 25° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.10' de ise aynı veriler tablo halinde sunulmuştur.

Çizelge 4.10 Sol ve sağ tekerleklerin zemin 1 şartlarında 25° sağa yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
7.3384	7.4672	12.8	1.7552	4.8435	4.9286	12.8	1.757
7.3244	7.4672	9.47	1.9496	4.8344	4.9286	9.47	1.9485
7.1928	7.4672	11.84	3.8149	4.7474	4.9286	11.84	3.8168
7.2421	7.4672	11.47	3.1082	4.78	4.9286	11.47	3.1088
7.1794	7.4672	11.54	4.0087	4.7386	4.9286	11.54	4.0096
7.2785	7.4672	12.72	2.5926	4.804	4.9286	12.72	2.5937
7.2512	7.4672	12.5	2.9788	4.786	4.9286	12.5	2.9795
7.2331	7.4672	12.5	3.2365	4.7741	4.9286	12.5	3.2362
7.1485	7.4672	13.32	4.4583	4.7182	4.9286	13.32	4.4593
7.0787	7.4672	11.84	5.4883	4.6721	4.9286	11.84	5.49
7.1177	7.4672	12.5	4.9103	4.6979	4.9286	12.5	4.9107
7.1573	7.4672	15.61	4.3298	4.724	4.9286	15.61	4.3311
7.2196	7.4672	15.46	3.4296	4.7652	4.9286	15.46	3.429
7.2151	7.4672	13.61	3.4941	4.7622	4.9286	13.61	3.4942
7.2017	7.4672	13.02	3.6866	4.7533	4.9286	13.02	3.688
7.3106	7.4672	17.02	2.1421	4.8252	4.9286	17.02	2.1429
7.2467	7.4672	13.39	3.0428	4.783	4.9286	13.39	3.0441
7.2603	7.4672	12.8	2.8497	4.792	4.9286	12.8	2.8506

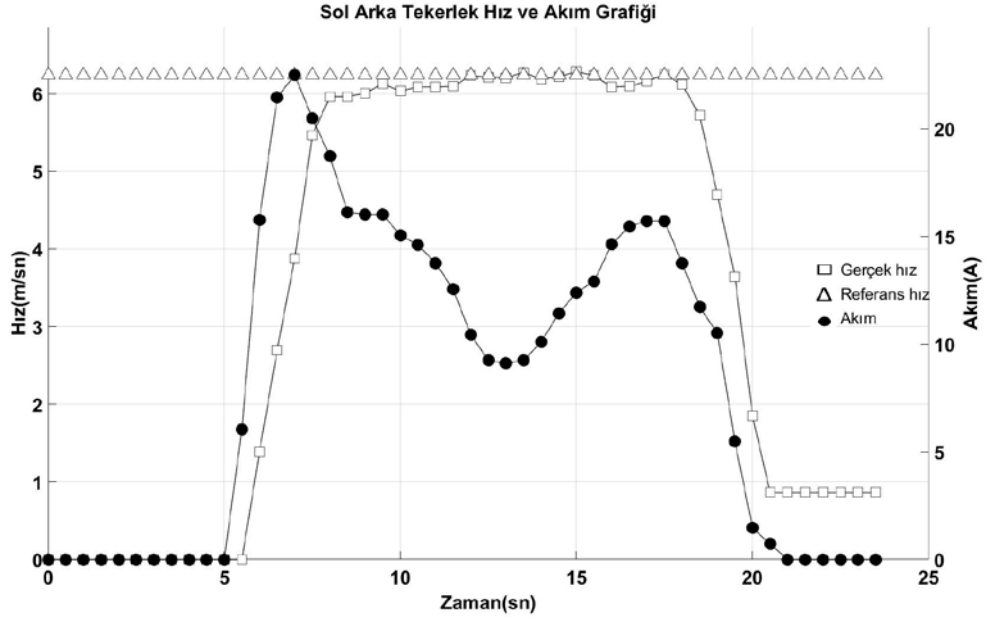
Şekil ve çizelgeler yorumlandığında sol tekerlek içi kayma ortalama %3.40 iken sağ tekerleğin ise %3.40 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 7.46 m/sn iken %3.40' lik bir kaymayı $\frac{7.46 \cdot 3.40}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.253 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 4.92 m/sn iken %3.40' lik bir kaymayı $\frac{4.92 \cdot 3.40}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.167 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.4 Zemin 2 Şartlarında

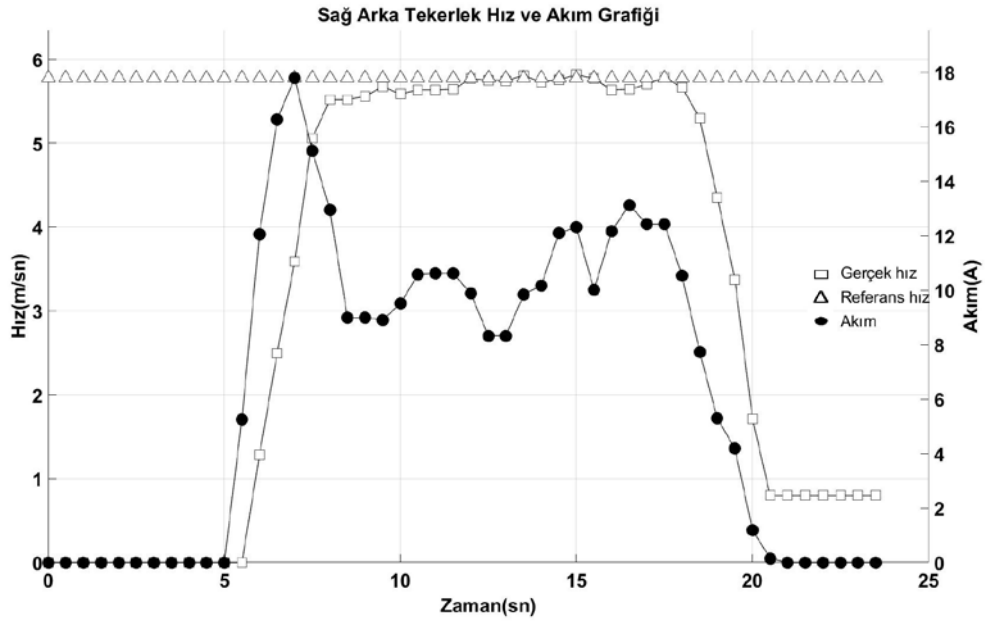
Elektrikli araç prototipi ıslak zemin üzerinde sırasıyla 5°, 15° ve 25° sağa doğru yönelimle tekerleklerin hızları, tekerleklere bağlı motor akımları ve referans hızları verilecektir.

4.2.4.1 Zemin 2 Şartlarında 5° sağa yönelim için alınan veriler

Şekil 4.21’ de sol ve şekil 4.22’de sağ tekerlek için 5° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.21 Zemin 2 şartlarında 5° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.22 Zemin 2 şartlarında 5° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 5° sağı dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.11’ de ise aynı veriler tablo halinde sunulmuştur.

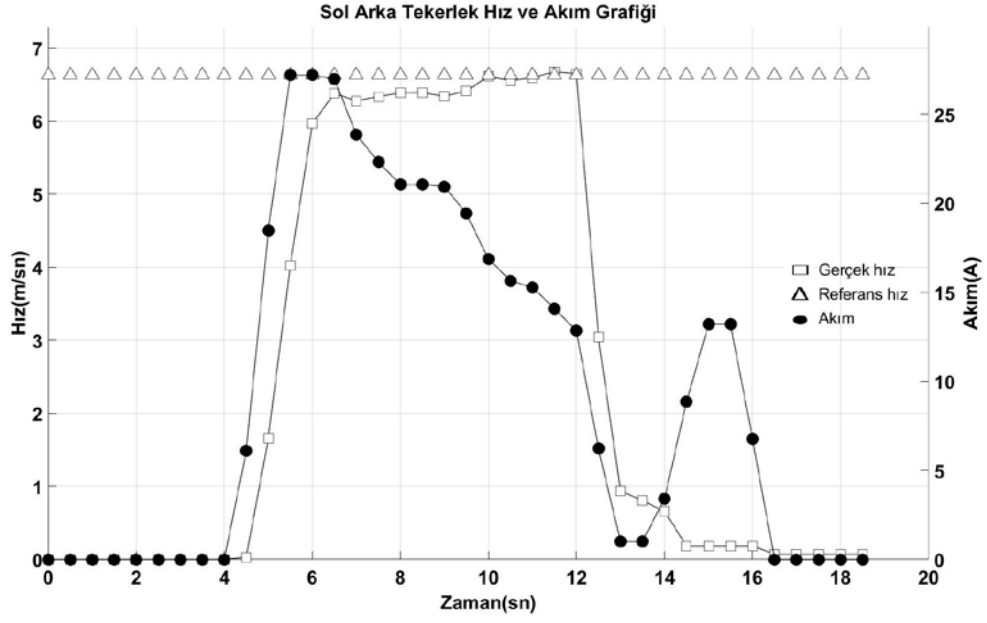
Çizelge 4.11 Sol ve sağ tekerleklerin zemin 2 şartlarında 5° sağı yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
6.2318	6.2382	12.8	0.1027	5.7706	5.7765	12.8	0.1022
6.2131	6.2382	9.47	0.404	5.7533	5.7765	9.47	0.4032
6.2038	6.2382	11.84	0.5545	5.7447	5.7765	11.84	0.5536
6.2744	6.2382	11.47	0.5769	5.81	5.7765	11.47	0.5766
6.1852	6.2382	11.54	0.8569	5.7275	5.7765	11.54	0.8555
6.2178	6.2382	12.72	0.3281	5.7576	5.7765	12.72	0.3283
6.2839	6.2382	12.5	0.7273	5.8189	5.7765	12.5	0.7287
6.2318	6.2382	12.5	0.1027	5.7706	5.7765	12.5	0.1022
6.0851	6.2382	13.32	2.516	5.6348	5.7765	13.32	2.5147
6.094	6.2382	11.84	2.3663	5.6431	5.7765	11.84	2.3639
6.1576	6.2382	12.5	1.309	5.7019	5.7765	12.5	1.3083
6.2459	6.2382	15.61	0.1233	5.7837	5.7765	15.61	0.1245
6.1211	6.2382	15.46	1.9131	5.6681	5.7765	15.46	1.9125

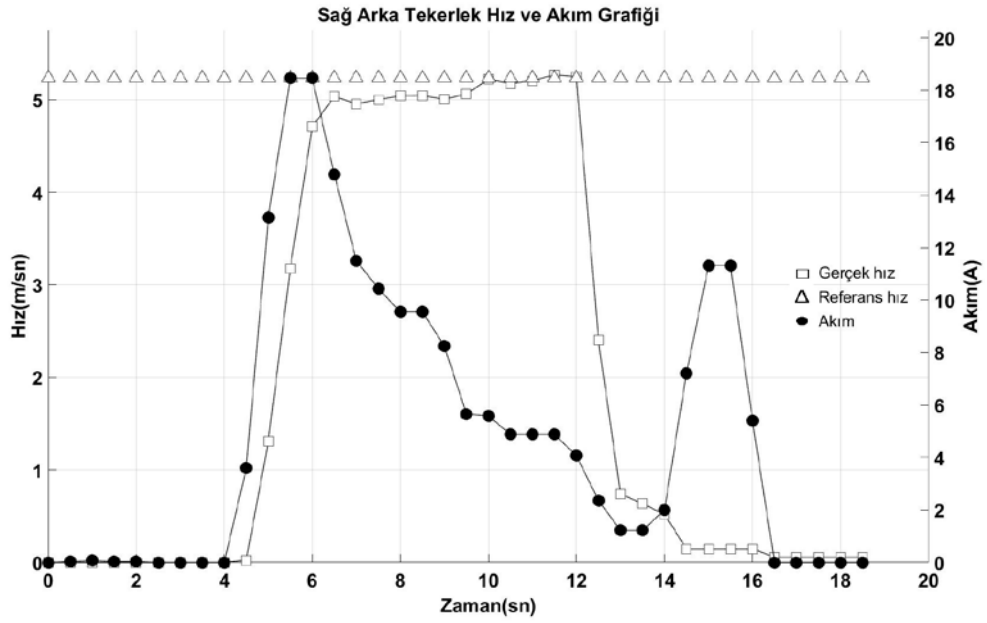
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %0.91 iken sağ tekerleğin ise %0.91 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 6.23 m/sn iken %0.91’ lik bir kaymayı $\frac{6.23 \cdot 0.91}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.056 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 5.77 m/sn iken %0.91’ lik bir kaymayı $\frac{5.77 \cdot 0.91}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.052 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.4.2 Zemin 2 Şartlarında 15° sağı yönelim için alınan veriler

Şekil 4.23’ de sol ve şekil 4.24’te sağ tekerlek için 15° sağı dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.23 Zemin 2 şartlarında 15° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.24 Zemin 2 şartlarında 15° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 15° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.12’ de ise aynı veriler tablo halinde sunulmuştur.

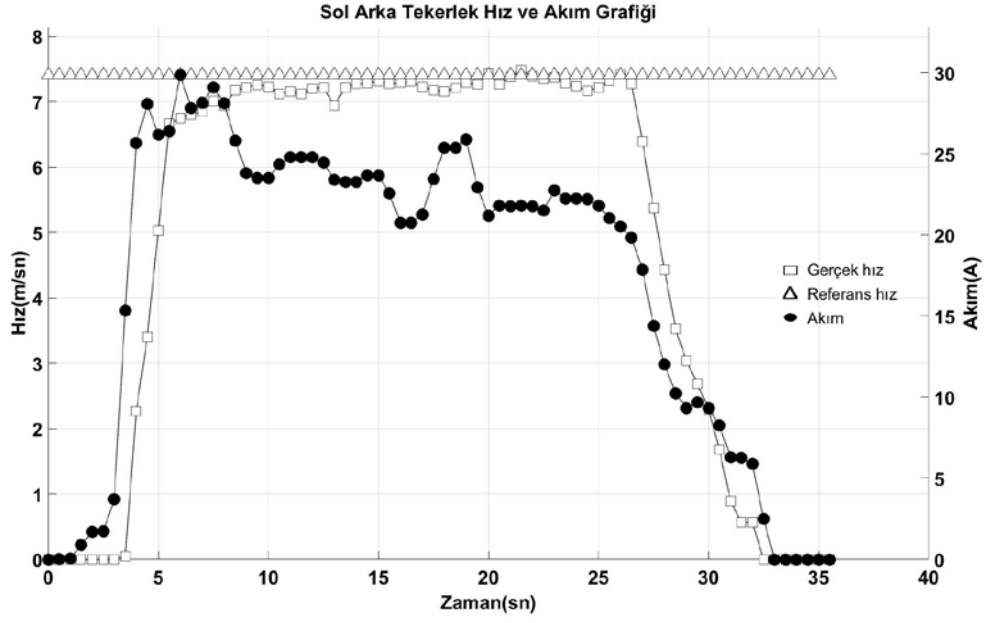
Çizelge 4.12 Sol ve sağ tekerleklerin zemin 2 şartlarında 15° sağa yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
6.3359	6.6292	12.8	4.6292	5.0017	5.2332	12.8	4.6284
6.386	6.6292	9.47	3.8083	5.0412	5.2332	9.47	3.8086
6.3902	6.6292	11.84	3.7401	5.0446	5.2332	11.84	3.7387
6.3401	6.6292	11.47	4.5599	5.005	5.2332	11.47	4.5594
6.4113	6.6292	11.54	3.3987	5.0612	5.2332	11.54	3.3984
6.6124	6.6292	12.72	0.2541	5.2199	5.2332	12.72	0.2548
6.5543	6.6292	12.5	1.1428	5.1741	5.2332	12.5	1.1422
6.5944	6.6292	12.5	0.5277	5.2057	5.2332	12.5	0.5283
6.6761	6.6292	13.32	0.7025	5.2702	5.2332	13.32	0.7021
6.6486	6.6292	11.84	0.2918	5.2485	5.2332	11.84	0.2915

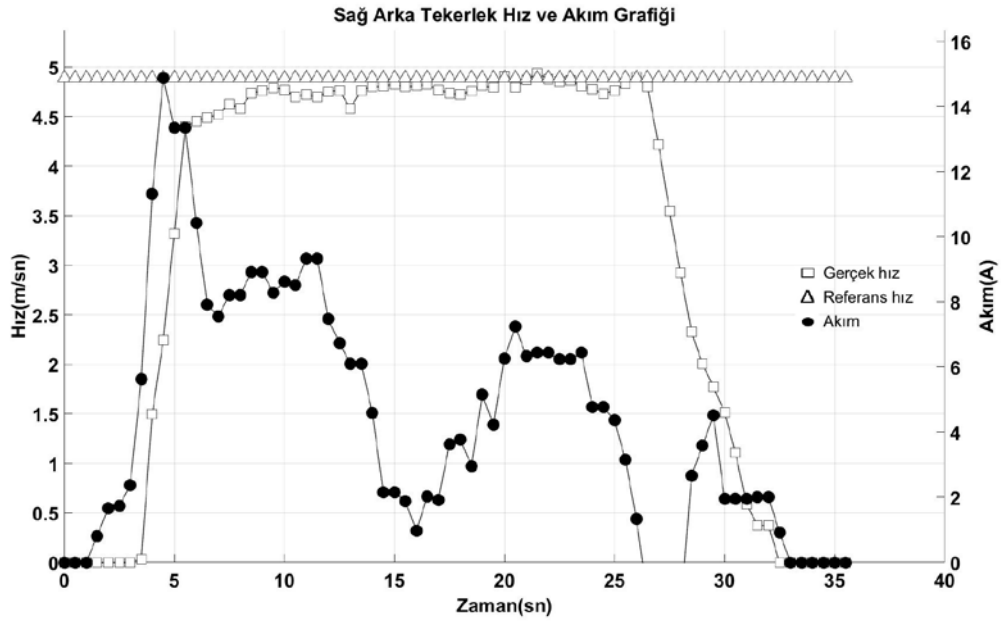
Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %2.30 iken sağ tekerleğin ise %2.30 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 6.62 m/sn iken %2.30’ lik bir kaymayı $\frac{6.62*2.30}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.152 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 5.23 m/sn iken %2.30’ lik bir kaymayı $\frac{5.23*2.30}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.120 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

4.2.4.3 Zemin 2 Şartlarında 25° sağa yönelim için alınan veriler

Şekil 4.25’ te sol ve şekil 4.26’ de sağ tekerlek için 25° sağa dönüşte elde edilen sonuç verileri ayrı ayrı verilmiştir.



Şekil 4.25 Zemin 2 şartlarında 25° sağa yönelimde sol tekerlek hız ve akım grafiği.



Şekil 4.26 Zemin 2 şartlarında 25° sağa yönelimde sağ tekerlek hız ve akım grafiği.

Yukarıda sağ ve sol tekerleğin 25° sağa dönüş için gerçek hız ve motor akımları verilmiştir. Çizelge 4.13’ de ise aynı veriler tablo halinde sunulmuştur.

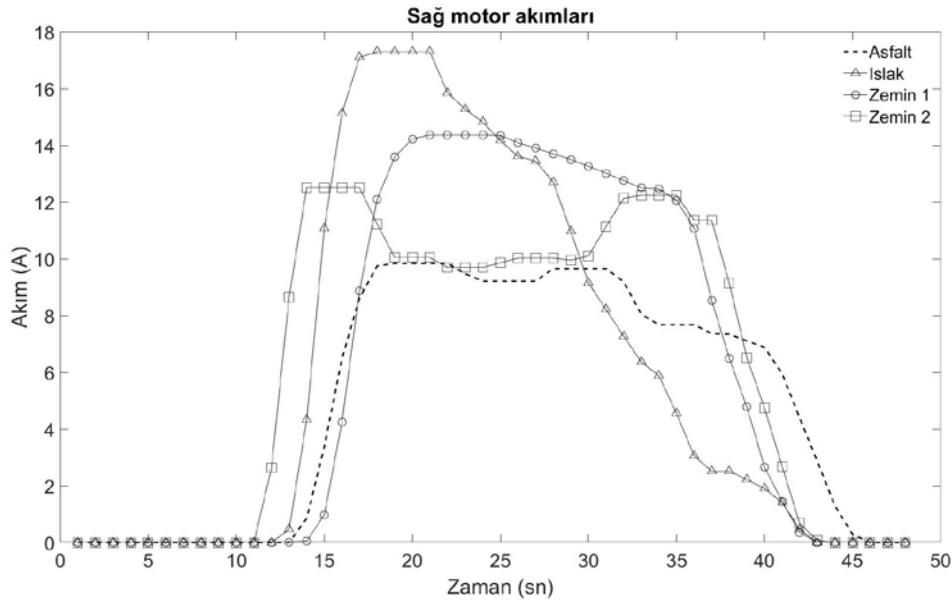
Çizelge 4.13 Sol ve sağ tekerleklerin zemin 2 şartlarında 25° sağa yönelimdeki gerçek zamanlı verileri.

Sol tekerlek				Sağ tekerlek			
Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)	Gerçek hız (m/sn)	Referans hız (m/sn)	Akım (A)	Kayma (%)
7.2648	7.4075	12.8	1.9643	4.795	4.8892	12.8	1.9645
7.4372	7.4075	9.47	0.3993	4.9087	4.8892	9.47	0.3973
7.2694	7.4075	11.84	1.8997	4.798	4.8892	11.84	1.9008
7.3804	7.4075	11.47	0.3672	4.8713	4.8892	11.47	0.3675
7.4803	7.4075	11.54	0.9732	4.9372	4.8892	11.54	0.9722
7.3945	7.4075	12.72	0.1758	4.8806	4.8892	12.72	0.1762
7.3523	7.4075	12.5	0.7508	4.8528	4.8892	12.5	0.7501
7.371	7.4075	12.5	0.4952	4.8651	4.8892	12.5	0.4954
7.2831	7.4075	13.32	1.7081	4.8071	4.8892	13.32	1.7079
7.2376	7.4075	11.84	2.3475	4.7771	4.8892	11.84	2.3466
7.1661	7.4075	12.5	3.3686	4.7298	4.8892	12.5	3.3701
7.2196	7.4075	15.61	2.6026	4.7652	4.8892	15.61	2.6022
7.3198	7.4075	15.46	1.1981	4.8313	4.8892	15.46	1.1984
7.4229	7.4075	13.61	0.2075	4.8993	4.8892	13.61	0.2062

Şekil ve çizelgeler yorumlandığında sol tekerlek için kayma ortalama %1.31 iken sağ tekerleğin ise %1.31 olduğu görülmüştür. Sol tekerlek için referans hız değerimiz 7.40 m/sn iken %1.31’ lik bir kaymayı $\frac{7.40 \cdot 1.31}{100}$ formülden hesaplandığında; sol tekerlek için kayma değeri 0.096 m/sn olmaktadır. Sağ tekerlek için referans hız değerimiz 4.88 m/sn iken %1.31’ lik bir kaymayı $\frac{4.88 \cdot 1.31}{100}$ formülden hesaplandığında; sağ tekerlek için kayma değeri 0.063 m/sn olmaktadır. Sol tekerlek dönme merkezine göre dıştaki tekerlek olduğundan dolayı kaymanın sağ tekerleğe göre fazla çıkması stabil dönmeyi desteklemektedir.

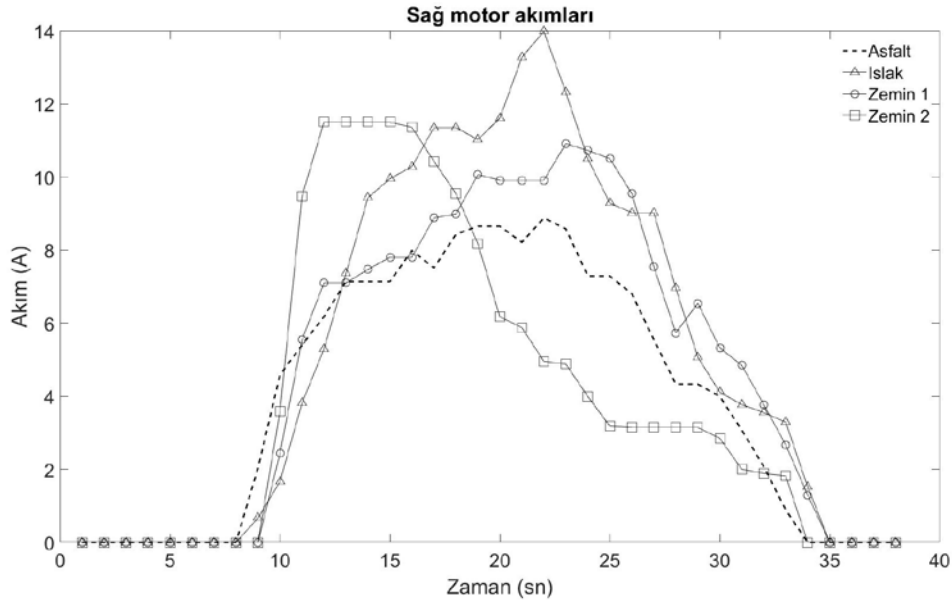
5. TARTIŞMA ve SONUÇ

Literatürdeki çalışmalar incelendiğinde elektronik diferansiyel sistemi için genellikle simülasyon üzerinde durulmuştur. Bu tez kapsamında elektronik diferansiyelin MATLAB/Simulink ortamında modellenmesinin yanı sıra bir elektrikli araç tasarlanıp elektronik diferansiyelin gerçek zamanlı uygulaması gerçekleştirilmiştir. Sistem yazılımı STMicroelectronics firmasının geliştirmiş olduğu STM32F103 mikroişlemcisi üzerine kodlanmıştır. Kod derleme ve yükleme programı olarak Eclipse programının temel alındığı System Workbench for STM32 yazılımı kullanılmıştır. Mekanik diferansiyelin matematiksel olarak ifade edilmesinde Ackermann-Jeantand modeli baz alınarak hem simulink ortamından hem de gerçek araç üzerinde elektronik diferansiyel sistemi modellenmiştir. Belirli sınırlar içerisinde, aracın gireceği viraj için direksiyon açısına göre tekerleklerin dönme hızları hesaplanıp, olması gereken hızlarda dönmeleri için kontrol sağlanmıştır. Bu nedenle mekanik diferansiyel yerine iki farklı elektrik motoruyla stabil bir kontrol sağlanmış olmaktadır. Sistemde, doğru akım elektrik motorları, elektrikli aracın arka iki tekerleğine kayıp oluşturacak herhangi bir aktarma organı kullanılmadan doğrudan bağlanmıştır. Bu nedenle kayıp minimuma indirilmeye çalışılmıştır. Tez kapsamında yapılan elektrikli araç prototipi dört farklı yol şartında direksiyon açısı sabit 5° girilerek alınan akım verileri şekil 4.27' de gösterilmiştir.

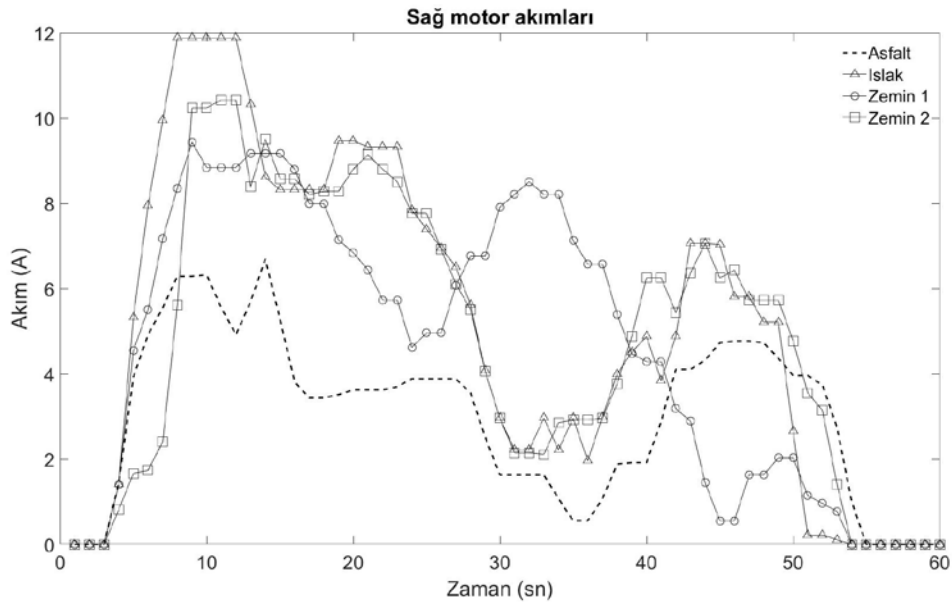


Şekil 5.1 Farklı zeminlerde 5° direksiyon açısıyla alınan akım verileri grafiği.

Direksiyon açısı telemetri ile kumandadan gelerek alınmış ve matematiksel modele girdi olarak verilip referans hız değerleri elde edilmiştir. Bu hız değerlerine göre sistem ARM işlemcisi üzerinden kontrolü sağlanmıştır. Elektrikli araç istenilen direksiyon açısı ve hız değerine göre kontrol edilmiştir. Tez kapsamında yapılan elektrikli araç prototipi dört farklı yol şartında direksiyon açısı sabit 15° girilerek alınan akım verileri şekil 5.2’ de ve direksiyon açısı sabit 25° girilerek alınan akım verileri ise şekil 5.3’ da gösterilmiştir.



Şekil 5.2 Farklı zeminlerde 15° direksiyon açısıyla alınan akım verileri grafiği.



Şekil 5.3 Farklı zeminlerde 25° direksiyon açısıyla alınan akım verileri grafiği.

Yapılan testler sonrasında zemin srtnme katsayısı ne kadar yksek yani ara ne kadar iyi bir Őekilde yere tutunuyorsa, elektrikli ara tahriki iin kullanılan dođru akım motorları bir o kadar dzgn akım ekmektedirler. Bu da elektronik iin sorun teŐkil etmemektedir. Fakat diđer zemin Őartlarında srtnme katsayısı dŐtke akımda bir o kadar kararsız olmaktadır. Gelecek alıŐmalarda ara mekanik olarak daha iyi bir Őekilde geliŐtirilerek ara mekaniđinden kaynaklanan akım dzensizlikleri giderile bilinir. Bu da aracı hem konfor hem de elektronik anlamda olduka iyiye gtrecektir. Daha sonrasında standart bir elektrikli ara ile karŐılaŐtırılıp, kullanıcıya daha konforlu viraj deneyimi sunabilmek ve viraj gvenliđini arttırmaya ynelik alıŐmalar yapılabilir.

6. KAYNAKLAR

- Bacon, R. H. (1972). The Complete Car and Basic Structure. R. H. Bacon içinde, The Car (Engine and Structure). New York, The Macmillan Press:170-177.
- Bharatiraja, C., Munda, J., Vaghasia, I., Valiveti, R. and Manasa, P. (2016). Low cost Real Time Centralized Speed Control of DC Motor Using Lab View -NI USB 6008. *International Journal of Power Electronics and Drive System (IJPEDS)*, **7**:656-664.
- Clavero-Ordóñez, L., Fernández-Ramos, J. and Gago-Calderón, A. (2018). Electronic differential system for Light Electric Vehicles with two inwheel motors. *Renewable Energy and Power Quality Journal*, **16**:325-329.
- De Castro, R. P., Oliveira, H. S., Soares, J. R., Cerqueira, N. M. and Araujo, R. E. (2007). A new FPGA based control system for electrical propulsion with electronic differential. European Conference on Power Electronics and Applications. Aalborg, Denmark, IEEE:1-10.
- Draou, A. (2013). Electronic differential speed control for two in-wheels motors drive vehicle. 4th International Conference on Power Engineering, Energy and Electrical Drives. Istanbul, IEEE:764-769.
- Emhemed, A. A. and Mamat, R. B. (2012). Modelling and Simulation for Industrial DC Motor Using Intelligent Control. International Symposium on Robotics and Intelligent Sensors. *Procedia Engineering*, **41**:420-425.
- Gair, S., Cruden, A., McDonald, J. and Hredzak, B. (2004). Electronic differential with sliding mode controller for a direct wheel drive electric vehicle. Proceedings of the IEEE International Conference on Mechatronics. IEEE:98-103.
- Gao, L. (2011). Well-to-Wheels Analysis of Energy Use and Greenhouse Gas Emissions

for Alternative Fuels. *International Journal of Applied Science and Technology*, **6**:1-8.

Haddoun, A., Benbouzid, M. E., Diallo, D., Abdessemed, R., Ghouili, J. and Srairi, K. (2010). Design and implementation of an electric differential for traction application. Vehicle Power and Propulsion Conference, VPPC 2010. IEEE:1-6.

Hall, D. and Lutsey, N. (2018). Effects of battery manufacturing on electric vehicle life-cycle greenhouse gas emissions. The International Council On Clean Transportation:1-12.

Hartani, K., Bourahla, M., Miloud, Y. and Sekour, M. (2009). Electronic differential with direct torque fuzzy control for vehicle propulsion system. *Turkish Journal of Electrical Engineering and Computer Sciences*, **17**:21-38.

Hartani, K., Miloud, Y. and Miloudi, A. (2010). Electric Vehicle Stability with Rear Electronic Differential Traction. International Symposium on Environment Friendly Energies in Electrical Applications. Gardaya-Cezayir:1-5.

Ju-Sang, L., Young-Jae, R., Young-Cheol, L., Freere, P., Tae-Gon, K., Seok-Jun, S., and Eui-Sun, K. (2000). A neural network model of electric differential system for electric vehicle. Industrial Electronics Society 26th Annual Conference of the IEEE:83-88.

Lam, L. T., and Louey, R. (2006). Development of ultra-battery for hybrid-electric vehicle applications. *Journal of Power Sources*, **158**:1140-1148.

Magallán, G. A., De Angelo, C. H., Bisheimer, G., and Garcia, G. (2008). A neighborhood electric vehicle with electronic differential traction control. IECON Proceedings (Industrial Electronics Conference). IEEE:2757-2763.

Moazen, M. and Sabahi, M. (2014). Electric Differential for an Electric Vehicle with

Four Independent Driven Motors and Four Wheels Steering Ability Using Improved Fictitious Master Synchronization Strategy. *Journal of Operation and Automation in Power Engineering*, **2**:141-150.

Moreno, J., Ortuzar, M. E. and Dixon, J. W. (2006). Energy-Management System for a Hybrid Electric Vehicle, Using Ultracapacitors and Neural Networks. *IEEE Transactions On Industrial Electronics*, **53**: 614-623.

Nejh, A. and Trabelsi, H. (2015). Design of PMSMs Topologies Based on FEA for In-Wheel Motor Hybrid Propulsion Application. 12th International Multi-Conference on Systems, Signals and Devices. IEEE:1-5.

Quriroga, L. A., Benítez-Read, J. S., López-Callejas, R. and Segovia, A. (2015). New Relation to Improve the Speed and Torque Characteristics of Induction Motors. *Revista Facultad de Ingeniería*, **74**:37-49.

Sioshansi, R., and Denholm, P. (2009). Emissions Impacts and Benefits of Plug-In Hybrid Electric Vehicles and Vehicle-to-Grid Services. *Environmental Science & Technology*, **43**:1199-1204.

Vliet, O. v., Brouwer, A. S., Kuramochi, T., Broek, M. v. and Faaij, A. (2011). Energy use, cost and CO2 emissions of electric cars. *Journal of Power Sources*, **196**: 2298-2310.

Wlodzimierz, G., Przemyslaw, K. and Madziara, S. (2018). Vehicle Dynamics and Green Electronic Differential for eKart. *Green Electronics*, **10**:105-123.

Yıldırım, M., Öksüztepe, E., Tanyeri, B. ve Kürüm, H. (2016). Electronic differential system for an electric vehicle with in-wheel motor. 2015 9th International Conference on Electrical and Electronics Engineering (ELECO). Bursa: IEEE:1048-1052.

Zhai, L. and Dong, S. (2011). Electronic Differential Speed Steering Control for Four In-wheel Motors Independent Drive Vehicle. World Congress on Intelligent Control and Automation. Tapei-Tayvan: IEEE:780-783.

Zhou, Y., Li, S., Zhou, X. and Fang, Z. (2010). The control strategy of electronic differential for EV with four in-wheel motors. 2010 Chinese Control and Decision Conference. Xuzhou, China: IEEE:4190-4195.

İnternet Kaynakları

- 1) <https://www.dunyabulteni.net/images/resize/100/656x400/haberler/250x190/2012/06/08/elektrikli-araba.jpg>, 21.01.2019
- 2) <http://www.hayretedeceksin.com/wp-content/uploads/2018/04/elektrikaraba1.jpg>, 21.01.2019
- 3) <https://upload.wikimedia.org/wikipedia/commons/thumb/8/8a/EdisonElectricCar1913.jpg/220px-EdisonElectricCar1913.jpg>, 21.01.2019
- 4) <https://www.american-automobiles.com/Ford/1910-Ford-Touring-Car.jpg>, 21.01.2019
- 5) <https://wallpapercave.com/wp/wp4318798.jpg>, 24.01.2019
- 6) <https://www.energuide.be/en/questions-answers/how-much-power-does-anelectric->, 29.01.2019
- 7) <https://x-engineer.org/automotive-engineering/internal-combustion-engines/performance/power-vs-torque/>, 15.02.2019
- 8) https://press.jp.porsche.com/presskits_until_2015/products/2015/911Carrera/text/technical_data/CarreraS/TD_911_Carrera_S_EN.pdf, 20.02.2019
- 9) <http://bitsavers.org/pdf/viatron/ViatronSystem21Brochure.pdf>, 02.02.2019
- 10) <https://otomasyonadair.com/wp-content/uploads/2018/04/enkoder3-300x277.png>, 10.02.2019
- 11) <https://cahitsoyman.blogspot.com/2015/06/eclipse-nedir-ve-nasil-kullanilir.html>, 10.02.2019

ÖZGEÇMİŞ

Adı Soyadı : İbrahim ÇELİK
Doğum Yeri ve Tarihi : AĞRI - 01.08.1992
Yabancı Dili : İngilizce
İletişim (Telefon/e-posta) : 0553 351 90 01 / ibrahimcelik@aku.edu.tr

Eğitim Durumu (Kurum ve Yıl)

Lise : Ağrı Dağı Anadolu Lisesi, (2005-2009)
Lisans : Fırat Üniversitesi, Mühendislik Fakültesi,
Mekatronik Mühendisliği Bölümü, (2010-2014)
Yüksek Lisans : Afyon Kocatepe Üniversitesi, Fen Bilimleri
Enstitüsü, Makine Mühendisliği Anabilim Dalı,
(2017-2019)

Çalıştığı Kurum/Kurumlar ve Yıl : Afyon Kocatepe Üniversitesi (2017-Devam ediyor)

EKLER

EK-1 Simülâtör Kodları

main.c

```
#include "stm32f405rg.h"
/*
 * Include all needed standard peripheral driver headers.
 */
#include "stm32f4xx_rcc.h"
#include "stm32f4xx_gpio.h"
#include "stm32f4xx_dma.h"
#include "misc.h"

#include "sys.h"
#include "video.h"
#include "gdi.h"

#define PLLM          8
#define PLLN          288
#define PLLP          1
#define PLLQ          7

/*
 * Define a structure for inializing the LED port lines.
 */

extern void gosterge (void);

void RCC_Configuration(void)
{
    /* TIM1, GPIOA, GPIOB, GPIOE and AFIO clocks enable */
    /* Original code initialized AFIO; what is it? */

    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_DMA2
RCC_AHB1Periph_GPIOA |
                           RCC_AHB1Periph_GPIOB, ENABLE);
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1
RCC_APB2Periph_TIM1 |
                           RCC_APB2Periph_SPI1, ENABLE);

    //RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE); //
    debug only for LEDs
}

void Clock_Conf(void) {
    ErrorStatus HSE_Status;
    RCC_HSEConfig(RCC_HSE_ON);
    HSE_Status = RCC_WaitForHSEStartUp();
    if (HSE_Status==SUCCESS) {
        RCC_HCLKConfig(RCC_SYSCLK_Div1);
        RCC_PCLK1Config(RCC_HCLK_Div4);
        RCC_PCLK2Config(RCC_HCLK_Div2);
        RCC_PLLConfig(RCC_PLLSource_HSE, PLLM, PLLN, PLLP, PLLQ);
        RCC_PLLCmd( ENABLE);
    }
}
```


EK-1(Devam) Simülâtör Kodları

```
        while (RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET)
            {}
        RCC_SYSClkConfig(RCC_SYSClkSource_PLLCLK);
        while (RCC_GetSYSClkSource() != 0x08)
            {}
    }
    else{
        while(1){}
    }
}
void GPIO_Configuration(void)
{
}

int main(void)
{
    //SystemInit();
    Clock_Conf();
    RCC_Configuration();
    GPIO_Configuration();

    vidInit();
    // sysInitSystemTimer();

    gosterge ();

    return 0;
}
```

gdi.c

```
#include "stm32f405rg.h"

#include "gdi.h"
#include "mth.h"
#include "video.h"

#ifndef NULL
#define NULL 0
#endif

extern u8          fb[VID_VSIZE][VID_HSIZE+2];
extern const u8   gdiSystemFont[];
extern const u8   gdiSystemFont2[];

const u8          gdiCloseBm[] = {    0x7f, 0xC0,
                                       0x7f,
0xC0,
                                       0x7f,
0xC0,
                                       0x7f,
0xC0,
                                       0x40,
0x40,
```

EK-1(Devam) Simülâtör Kodları

```
0xC0, 0x7f,
0xC0, 0x7f,
0xC0, 0x7f,
}; 0x7f, 0xC0

u16 strLen(pu8 str) {
int i = 0;
while (*str != 0) {
++i;
++str;
}
return(i);
}

void gdiCopyRect(PGDI_RECT rc1, PGDI_RECT rc2) {
rc1->x = rc2->x;
rc1->y = rc2->y;
rc1->w = rc2->w;
rc1->h = rc2->h;
}

void gdiBitBlt(PGDI_RECT prc, i16 x, i16 y, i16 w, i16 h, pu8 bm, u16
rop) {
u16 i, xz, xb, xt;
u32 wb; // Width in bytes
u32 r; // Start X position in bits
(relative to x)
u32 k;
u32 d;
u32 ofs;
u8 c;
pu8 fbPtr; // Pointer to the Frame
Buffer Bit-Band area
pu8 fbBak;
u8 fb1;
u32 fb2;
u32 rp;
pu8 bmPtr; // Pointer to the bitmap
bits

// Calculate clipping region
if (prc != NULL) {
x = prc->x + x;
y = prc->y + y;
}

// Get total bitmap width in bytes
wb = (u32) w >> 3;
```

EK-1(Devam) Simülâtör Kodları

```
        if ((wb << 3) < (u32) w) ++wb;

//    Get starting bit inside the first byte

    d = (u32) x >> 3;
    r = ((u32) x - (d << 3));

//    Clip X

    if (prc == NULL) {
        if ((x + w) >= VID_PIXELS_X ) {
            xt = VID_PIXELS_X - x;
        } else {
            xt = w;
        }
    } else {
        if ((x + w) >= (x + prc->w)) {
            xt = prc->w - x;
        } else {
            xt = w;
        }
    }

//    Draw bits

    for (i = 0; i < h; i++) {

//    Clip Y

        if ((i + y) > (VID_VSIZE - 1)) return;

//    Get offset to frame buffer in bit-banding mode

        ofs = (((u32) x >> 3)) + ((u32) (y + i) * VID_HSIZE_R);
        k = (u32) (&fb - 0x20000000);
        k += ofs;
        fbPtr = (pu8) (0x22000000 + (k * 32) + ((7 - r) * 4));
        fbBak = (pu8) (0x22000000 + (k * 32) + 28);

//    Get offset to bitmap bits

        bmPtr = bm + ((u32) i * wb);
        xz = w;

        xb = 0;
        for (xz = 0; xz < xt; xz++) {
            fb1 = ((u32) fbPtr) & 0x000000E0;
            if (xb++ == 0) {
                c = *bmPtr;
                ++bmPtr;
            }
            xb &= 0x07;
            (c & 0x80) ? (rp = 1) : (rp = 0);
            switch(rop) {
                case GDI_ROP_COPY:      *fbPtr = rp;
                case GDI_ROP_XOR:      *fbPtr ^= rp;
                case GDI_ROP_AND:      *fbPtr &= rp;
            }
        }
    }
}

break;
break;
```

EK-1(Devam) Simülâtör Kodları

```
break;
                                case GDI_ROP_OR: *fbPtr |= rp;
break;
                                }
                                fbPtr -= 4;
                                fb2 = ((u32) fbPtr) & 0x000000E0;
                                if (fb1 != fb2) {
                                    fbPtr = fbBak + 32;
                                    fbBak = fbPtr;
                                }
                                c <<= 1;
                            }
                    }
}

void gdiPoint(PGDI_RECT rc, u16 x, u16 y, u16 rop) {
u16        w, r;
u8         m;

// Test for point outside display area

if (x >= VID_PIXELS_X || y >= VID_PIXELS_Y) return;

w = x >> 3;
r = x - (w << 3);

// Prepare mask

m = (0x80 >> r);

switch(rop) {
    case GDI_ROP_COPY:          fb[y][w] |= m;
                                break;
    case GDI_ROP_XOR:          fb[y][w] ^= m;
                                break;
    case GDI_ROP_AND:          fb[y][w] &= m;
                                break;
}
}

void gdiLine(PGDI_RECT prc, i16 x1, i16 y1, i16 x2, i16 y2, u16 rop)
{
i16        dx, dy, i, e;
i16        incx, incy, incl, inc2;
i16        x, y;

dx = x2 - x1;
dy = y2 - y1;

if(dx < 0) dx = -dx;
if(dy < 0) dy = -dy;
incx = 1;
if(x2 < x1) incx = -1;
incy = 1;
if(y2 < y1) incy = -1;
x=x1;
```

EK-1(Devam) Simülâtör Kodları

```
y=y1;

if (dx > dy) {
    gdiPoint(prc, x, y, rop);
    e = 2*dy - dx;
    incl = 2 * ( dy -dx);
    inc2 = 2 * dy;
    for (i = 0; i < dx; i++) {
        if (e >= 0) {
            y += incy;
            e += incl;
        }
        else {
            e += inc2;
        }
        x += incx;
        gdiPoint(prc, x, y, rop);
    }
} else {
    gdiPoint(prc, x, y, rop);
    e = 2 * dx - dy;
    incl = 2 * (dx - dy);
    inc2 = 2 * dx;
    for(i = 0; i < dy; i++) {
        if (e >= 0) {
            x += incx;
            e += incl;
        } else {
            e += inc2;
        }
        y += incy;
        gdiPoint(prc, x, y, rop);
    }
}

}

void gdiRectangle(i16 x0, i16 y0, i16 x1, i16 y1, u16 rop) {

    gdiLine(NULL,x0,y0,x1,y0,rop);
    gdiLine(NULL,x0,y1,x1,y1,rop);
    gdiLine(NULL,x0,y0,x0,y1,rop);
    gdiLine(NULL,x1,y0,x1,y1,rop);
}

void gdiRectangleEx(PGDI_RECT rc, u16 rop) {

    gdiRectangle(rc->x, rc->y, rc->x + rc->w, rc->y + rc->h,rop);
}

void gdiCircle(u16 x, u16 y, u16 r, u16 rop) {

i32      x1, y1;
u16      a;

    for (a = 0; a < 360; a++) {
        x1 = r * mthCos(a);
        y1 = r * mthSin(a);
    }
}
```

EK-1(Devam) Simülâtör Kodları

```
        gdiPoint(NULL, (x1 / 10000) + x, (y1 / 10000) + y, rop);
    }
}

void gdiDrawText(PGDI_RECT prc, pu8 ptext, u16 style, u16 rop) {
    u16      l, i, pos, xp;
    u8       c;
    pu8      ptx;

    l = strlen(ptext) * GDI_SYSFONT_WIDTH;
    switch(style) {
        case GDI_WINCAPTION_RIGHT:          if (l < prc->w) {
                                                prc->x += (prc->w - l);
                                                }
                                                break;
        case GDI_WINCAPTION_CENTER:        if (l < prc->w) {
                                                prc->x += ((prc->w - l) / 2);
                                                }
                                                break;
    }
    l = strlen(ptext);
    xp = 1; //prc->x;
    for (i = 0; i < l; i++) {
        c = *(ptext++);
        if (c >= GDI_SYSFONT_OFFSET) {
            pos = (u16) (c - GDI_SYSFONT_OFFSET) *
GDI_SYSFONT_BYTEWIDTH * GDI_SYSFONT_HEIGHT;
            ptx = ((pu8) gdiSystemFont) + pos;
            gdiBitBlt(prc, xp, 0, GDI_SYSFONT_WIDTH,
GDI_SYSFONT_HEIGHT, ptx, rop);
            xp += GDI_SYSFONT_WIDTH;
            if (xp >= ((prc->x + prc->w) - GDI_SYSFONT_WIDTH))
return;
        }
    }
}

void gdiDrawText24(PGDI_RECT prc, pu8 ptext, u16 style, u16 rop) {
    u16      l, i, pos, xp;
    u8       c;
    pu8      ptx;

    l = strlen(ptext) * GDI_SYSFONT_WIDTH24;
    switch(style) {
        case GDI_WINCAPTION_RIGHT:          if (l < prc->w) {
                                                prc->x += (prc->w - l);
                                                }
                                                break;
        case GDI_WINCAPTION_CENTER:        if (l < prc->w) {
                                                prc->x += ((prc->w - l) / 2);
                                                }
                                                break;
    }
}
```

EK-1(Devam) Simülâtör Kodları

```
    l = strlen(ptext);
    xp = 1; //prc->x;
    for (i = 0; i < l; i++) {
        c = *(ptext++);
        if (c >= GDI_SYSFONT_OFFSET24) {
            pos = (u16) (c - GDI_SYSFONT_OFFSET24) *
GDI_SYSFONT_BYTEWIDTH24 * GDI_SYSFONT_HEIGHT24;
            ptx = ((pu8) gdiSystemFont2) + pos;
            gdiBitBlt(prc, xp, 0, GDI_SYSFONT_WIDTH24,
GDI_SYSFONT_HEIGHT24, ptx, rop);
            xp += GDI_SYSFONT_WIDTH24;
            if (xp >= ((prc->x + prc->w) - GDI_SYSFONT_WIDTH24))
return;
        }
    }
}

void gdiDrawTextEx(i16 x, i16 y, pu8 ptext, u16 rop) {

u16    l, i, pos, xp;
u8     c;
pu8    ptx;

    l = strlen(ptext);
    xp = x;
    for (i = 0; i < l; i++) {
        c = *(ptext++);
        if (c >= GDI_SYSFONT_OFFSET) {
            pos = (u16) (c - GDI_SYSFONT_OFFSET) *
GDI_SYSFONT_BYTEWIDTH * GDI_SYSFONT_HEIGHT;
            ptx = ((pu8) gdiSystemFont) + pos;
            gdiBitBlt(NULL, xp, y, GDI_SYSFONT_WIDTH,
GDI_SYSFONT_HEIGHT, ptx, rop);
            xp += GDI_SYSFONT_WIDTH;
            if (xp >= VID_PIXELS_X) return;
        }
    }
}

void gdiDrawTextEx24(i16 x, i16 y, pu8 ptext, u16 rop) {

    u16    l, i, pos, xp;
    u8     c;
    pu8    ptx;

    l = strlen(ptext);
    xp = x;
    for (i = 0; i < l; i++) {
        c = *(ptext++);
        if (c >= GDI_SYSFONT_OFFSET24) {
            pos = (u16) (c - GDI_SYSFONT_OFFSET24) *
GDI_SYSFONT_BYTEWIDTH24 * GDI_SYSFONT_HEIGHT24;
            ptx = ((pu8) gdiSystemFont2) + pos;
            gdiBitBlt(NULL, xp, y, GDI_SYSFONT_WIDTH24,
GDI_SYSFONT_HEIGHT24, ptx, rop);
            xp += GDI_SYSFONT_WIDTH24;
            if (xp >= VID_PIXELS_X) return;
        }
    }
}
```

EK-1(Devam) Simülâtör Kodları

```
void gdiDrawWindow(PGDI_WINDOW pwin) {  
  
    i16          i;  
    GDI_RECT     rc, rt;  
  
    gdiCopyRect(&rc, &pwin->rc);  
    if (pwin->style & GDI_WINCAPTION) {  
        gdiCopyRect(&rt, &pwin->rc);  
        rt.h = rt.y + 11;  
        rt.x += 2;  
        rt.y += 1;  
        rc.h += 10;  
        for (i = 0; i < 11; i++) {  
            gdiLine(NULL, rc.x, rc.y + i, rc.x + rc.w, rc.y + i,  
GDI_ROP_COPY);  
        }  
        if (pwin->style & GDI_WINCLOSEICON) {  
            gdiBitBlt(&rc, rc.w - 9, 1, 10, 9, (pu8) gdiCloseBm,  
GDI_ROP_COPY);  
            rt.w -= 11;  
        } else {  
            rt.w -= 1;  
        }  
        gdiDrawText(&rt, pwin->caption, pwin->style &  
GDI_WINCAPTION_MASK, GDI_ROP_XOR);  
    }  
    gdiRectangleEx(&rc, GDI_ROP_COPY);  
}
```

gosterge.c

```
#include "stm32f405rg.h"  
#include "sys.h"  
#include "video.h"  
#include "gdi.h"  
#include "string.h"  
#include <stdlib.h>  
#include "aracresim.h"  
extern const u16 arac_x[];  
extern const u16 arac_y[];  
// extern volatile u32      sysTicks;  
  
const u8  deBadBoys1[] = { 0x00, 0x00,          0x00, 0x00,  
                           0x07, 0xc0,          0x07, 0xc0,  
                           0x1f, 0xf0,          0x1f, 0xf0,  
                           0x3f, 0xf8,          0x3f, 0xf8,  
                           0x33, 0x98,          0x33, 0x98,  
                           0x3f, 0xf8,          0x3f, 0xf8,  
                           0x0c, 0x60,          0x0c, 0x60,  
                           0x1b, 0xb0,          0x1b, 0xb0,  
                           0x30, 0x18,          0x30, 0x18,  
                           0x00, 0x00,          0x00, 0x00,  
                           0x00, 0x00 };  
  
const u8  deBadBoys2[] = { 0x00, 0x00,
```


EK-1(Devam) Simülâtör Kodları

```
0x00, 0x00,
0x07, 0xc0,
0x1f, 0xf0,
0x3f, 0xf8,
0x33, 0x98,
0x3f, 0xf8,
0x0c, 0x60,
0x1b, 0xb0,
0x0c, 0x60,
0x00, 0x00,
0x00, 0x00 };

void gosterge (void) {
    //i16      x=0, y=0, x1=0, y1=0, i, flip;
    char text[50]={'\0'};

    vidClearScreen();
    gdiRectangle(0,0,(VID_PIXELS_X - 1),VID_VSIZE - 1,0);
    gdiRectangle(350,60,750,110,0);//araç hızı
    gdiRectangle(350,60,750,110,0);//sol arka teker hızı
    gdiRectangle(350,60,750,110,0);//sağ arka teker hızı
    sprintf(text,"ARAC HIZI:200 km/h");
    gdiDrawTextEx24(360, 75, (pu8)text, GDI_ROP_COPY);
    //vidClearScreen();
    for (int i=0;i<sizeof(arac_x);i++){
        if(arac_x[i]>160 &&
arac_x[i]<410)gdiPoint(NULL, arac_x[i]+70, arac_y[i]+50,GDI_ROP_COPY);
    }
    //gdiPoint(NULL,kadrancerceve_x,kadrancerceve_y,GDI_ROP_COPY);

    //gdiLine(NULL,x,y,x1,y1,GDI_ROP_COPY);
    gdiDrawTextEx(650, 550, (pu8)"IBRAHIM CELIK", GDI_ROP_COPY);
    gdiDrawTextEx(20, 150, (pu8)"0", GDI_ROP_COPY);
    //gdiDrawTextEx(25, 124, (pu8)"20", GDI_ROP_COPY);
    gdiDrawTextEx(34, 100, (pu8)"40", GDI_ROP_COPY);
    //gdiDrawTextEx(38, 90, (pu8)"60", GDI_ROP_COPY);
    gdiDrawTextEx(70, 63, (pu8)"80", GDI_ROP_COPY);
    //gdiDrawTextEx(10, 110, (pu8)"100", GDI_ROP_COPY);
    gdiDrawTextEx(120, 50, (pu8)"120", GDI_ROP_COPY);
    //gdiDrawTextEx(120, 50, (pu8)"120", GDI_ROP_COPY);
    gdiDrawTextEx(170, 63, (pu8)"160", GDI_ROP_COPY);
    gdiDrawTextEx(206, 100, (pu8)"200", GDI_ROP_COPY);
    gdiDrawTextEx(220, 150, (pu8)"320", GDI_ROP_COPY);

// Demo Point

    vidClearScreen();
    gdiRectangle(0,0,(VID_PIXELS_X - 1),VID_VSIZE - 1,0);
    gdiDrawTextEx(180, 40, (pu8)"IBRAHIM CELIK", GDI_ROP_COPY);
    gdiDrawTextEx(161, 55, (pu8)"MEKATRONIK MUHENDISLIGI",
GDI_ROP_COPY);
    gdiDrawTextEx(172, 70, (pu8)"POINT DEMO", GDI_ROP_COPY);
    sysDelayMs(3000);
// srand(sysTicks);
    for (i = 0; i < 5000; i++) {
        sysDelayMs(1);
    }
}
```

EK-1(Devam) Simülâtör Kodları

```
        x = rand(); x = x % (VID_PIXELS_X - 1);
        y = rand(); y = y % VID_VSIZE - 1;
        gdiPoint(NULL, x, y, GDI_ROP_COPY);
    }
    sysDelayMs(3000);

//    Demo Line

    vidClearScreen();
    gdiRectangle(0,0,(VID_PIXELS_X - 1),VID_VSIZE - 1,0);
    gdiDrawTextEx(180, 40, (pu8)"IBRAHIM CELIK", GDI_ROP_COPY);
    gdiDrawTextEx(161, 55, (pu8)"MEKATRONIK MUHENDISLIGI",
GDI_ROP_COPY);
    gdiDrawTextEx(175, 70, (pu8)"LINE DEMO", GDI_ROP_COPY);
    sysDelayMs(3000);
//    srand(sysTicks);
    for (i = 0; i < 100; i++) {
        sysDelayMs(50);
        x = rand(); x = x % (VID_PIXELS_X - 1);
        y = rand(); y = y % VID_VSIZE - 1;
        x1 = rand(); x1 = x1 % (VID_PIXELS_X - 1);
        y1 = rand(); y1 = y1 % VID_VSIZE - 1;
        gdiLine(NULL, x, y, x1, y1, GDI_ROP_COPY);
    }
    sysDelayMs(3000);

//    Demo Square

    vidClearScreen();
    gdiRectangle(0,0,(VID_PIXELS_X - 1),VID_VSIZE - 1,0);
    gdiDrawTextEx(180, 40, (pu8)"IBRAHIM CELIK", GDI_ROP_COPY);
    gdiDrawTextEx(161, 55, (pu8)"MEKATRONIK MUHENDISLIGI",
GDI_ROP_COPY);
    gdiDrawTextEx(157, 70, (pu8)"RECTANGLE DEMO", GDI_ROP_COPY);
    sysDelayMs(3000);
//    srand(sysTicks);
    for (i = 0; i < 100; i++) {
        sysDelayMs(50);
        x = rand(); x = x % (VID_PIXELS_X - 1);
        y = rand(); y = y % VID_VSIZE - 1;
        x1 = rand(); x1 = x1 % 100;
        y1 = rand(); y1 = y1 % 100;
        gdiRectangle(x, y, x1, y1, GDI_ROP_COPY);
    }
    sysDelayMs(3000);

//    Demo Circle

    vidClearScreen();
    gdiRectangle(0,0,(VID_PIXELS_X - 1),VID_VSIZE - 1,0);
    gdiDrawTextEx(180, 40, (pu8)"IBRAHIM CELIK", GDI_ROP_COPY);
    gdiDrawTextEx(161, 55, (pu8)"MEKATRONIK MUHENDISLIGI",
GDI_ROP_COPY);
    gdiDrawTextEx(165, 70, (pu8)"CIRCLE DEMO", GDI_ROP_COPY);
    sysDelayMs(3000);
//    srand(sysTicks);
    for (i = 0; i < 200; i++) {
        sysDelayMs(20);
        x = rand(); x = x % (VID_PIXELS_X - 1);
```

EK-1(Devam) Simülâtör Kodları

```
        y = rand(); y = y % VID_VSIZE - 1;
        x1 = rand(); x1 = x1 % 80;
        gdiCircle(x,y,x1,GDI_ROP_COPY);
    }
    sysDelayMs(3000);

//    Demo Bitmap

    vidClearScreen();
    gdiRectangle(0,0,(VID_PIXELS_X - 1),VID_VSIZE - 1,0);
    gdiDrawTextEx(180, 40, (pu8)"IBRAHIM CELIK", GDI_ROP_COPY);
    gdiDrawTextEx(161, 55, (pu8)"MEKATRONIK MUHENDISLIGI",
GDI_ROP_COPY);
    gdiDrawTextEx(165, 70, (pu8)"BITMAP DEMO", GDI_ROP_COPY);
    sysDelayMs(3000);

    x = 100;
    y = 130;
    flip = 0;
    for (i = 0; i < 100; i++) {
        sysDelayMs(150);
        if (flip) {
            flip = 0;
            gdiBitBlt(NULL,x,y,15,12, (pu8)
deBadBoys1,GDI_ROP_COPY);
        } else {
            flip = 1;
            gdiBitBlt(NULL,x,y,15,12, (pu8)
deBadBoys2,GDI_ROP_COPY);
        }
        x += 2;
    }
    sysDelayMs(3000);

    vidClearScreen();
    gdiRectangle(0,0,(VID_PIXELS_X - 1),VID_VSIZE - 1,0);
    gdiDrawTextEx(190, 40, (pu8)"IBRAHIM CELIK", GDI_ROP_COPY);
    gdiDrawTextEx(161, 55, (pu8)"MEKATRONIK MUHENDISLIGI",
GDI_ROP_COPY);
    gdiDrawTextEx(115, 75, (pu8)"ARM CORTEX DEVELOPMENT BOARDS",
GDI_ROP_COPY);

    gdiDrawTextEx(161, 90, (pu8)"IBRAHIM CELIK", GDI_ROP_COPY);
    gdiDrawTextEx(141, 120, (pu8)"FREE GRAPHIC ENGINE",
GDI_ROP_COPY);
    gdiDrawTextEx(161, 130, (pu8)"SOURCE CODE", GDI_ROP_COPY);

    while (1) {};

*/
}

static int                next;

int rand(void)
{
    next = next * 1103515245 + 12345;
    return (unsigned int)(next/65536) % 32768;
}
```

EK-1(Devam) Simülâtör Kodları

```
}

/* srand: set seed for rand() */
void srand(unsigned int seed)
{
    next = seed;
}
}
```

video.c

```
#include "stm32f405rg.h"

#include "stm32f4xx_rcc.h"
#include "stm32f4xx_gpio.h"
#include "stm32f4xx_dma.h"
#include "stm32f4xx_spi.h"
#include "stm32f4xx_tim.h"
#include "misc.h"

#include "video.h"

#define GREEN                LED4_PIN
#define ORANGE               LED3_PIN
#define RED                  LED5_PIN
#define BLUE                 LED6_PIN
#define ALL_LEDS             (GREEN | ORANGE | RED | BLUE)        //    all
leds
#define LEDS_GPIO_PORT      (GPIOB)

#define LED_ON(x)            GPIO_SetBits(LED_S_GPIO_PORT, x)
#define LED_OFF(x)          GPIO_ResetBits(LED_S_GPIO_PORT, x)

#define VIDEO_DMA            DMA2
#define DMA_STREAM           DMA2_Stream3
#define DMA_CHANNEL          DMA_Channel_3
#define DMA_STREAM_IRQ       DMA2_Stream3_IRQn
#define DMA_IT_TCIF          DMA_IT_TCIF0
#define DMA_STREAM_IRQHANDLER DMA2_Stream3_IRQHandler

#define VTOTAL               (VID_HSIZE+2)                        /* Total bytes to send
through SPI for one horizontal line */

u8    fb[VID_VSIZE][VTOTAL]  __attribute__((aligned(32))); /*    Frame
buffer */

static volatile u16 vline = 0;                                    /* The current
line being drawn */
static volatile u32 vflag = 0;                                    /* When 1, the
SPI DMA request can draw on the screen */
static volatile u32 vdraw = 0;                                    /*    Used    to
increment vline every 3 drawn lines */

void TIMER_Configuration(void)
{
    GPIO_InitTypeDef          GPIO_InitStructure;
    NVIC_InitTypeDef          nvic;
```

EK-1(Devam) Simülâtör Kodları

```
TIM_TimeBaseInitTypeDef      TIM_TimeBaseStructure = {0,};
TIM_OCInitTypeDef            TIM_OCInitStructure = {0,};
u32                           TimerPeriod = 0;
u16                           Channel1Pulse = 0;
u16                           Channel2Pulse = 0;
u16                           Channel3Pulse = 0;

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 | GPIO_Pin_8;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_Init(GPIOA, &GPIO_InitStructure);

/* Configure PA8 for Alternate Function 2 (TIM1) */
GPIO_PinAFConfig(GPIOA, GPIO_PinSource8, GPIO_AF_TIM1);
// TimerPeriod = 2048;
// Channel1Pulse = 144;          /* HSYNC */
// Channel2Pulse = 352;        /* HSYNC + BACK PORCH */

TimerPeriod = SystemCoreClock / 35156;          // Ticks
per whole line
Channel1Pulse = SystemCoreClock / 500000;      // HSYNC
Channel2Pulse = SystemCoreClock / 180000;      // HSYNC +
BACK PORCH

TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure); // always
init the time base structure first!

TIM_TimeBaseStructure.TIM_Prescaler = 0;
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseStructure.TIM_Period = TimerPeriod;
TIM_TimeBaseStructure.TIM_ClockDivision = TIM_CKD_DIV1;
TIM_TimeBaseStructure.TIM_RepetitionCounter = 0;
TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);

TIM_OCStructInit(&TIM_OCInitStructure);        // always init
the OC structure first!

TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM2; // PWM2 =
clear on compare match
TIM_OCInitStructure.TIM_Pulse = Channel1Pulse;
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_OutputNState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_Low;
TIM_OCInitStructure.TIM_OCNPolarity = TIM_OCPolarity_Low;
TIM_OCInitStructure.TIM_OCIdleState = TIM_OCIdleState_Reset;
TIM_OCInitStructure.TIM_OCNIIdleState = TIM_OCNIIdleState_Reset;
TIM_OC1Init(TIM1, &TIM_OCInitStructure);

TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM2; // was
TIM_OCMode_Inactive
TIM_OCInitStructure.TIM_Pulse = Channel2Pulse;
TIM_OC2Init(TIM1, &TIM_OCInitStructure);

TIM_OC1PreloadConfig(TIM1, TIM_OCPreload_Enable); // added
per stm32 tutorial
TIM_OC2PreloadConfig(TIM1, TIM_OCPreload_Enable); // added
per stm32 tutorial
```

EK-1(Devam) Simülâtör Kodları

```
TIM_ARRPreloadConfig(TIM1, ENABLE); //
added per STM32 forum

/* TIM1 counter enable and output enable */
TIM_CtrlPWMOutputs(TIM1, ENABLE);

/* Select TIM1 as Master */
TIM_SelectMasterSlaveMode(TIM1, TIM_MasterSlaveMode_Enable);
TIM_SelectOutputTrigger(TIM1, TIM_TRGOSource_Update);

/*
    Vertical timing
    -----

    Polarity of vertical sync pulse is positive.

                                Lines
    -----
    Visible area      600
    Front porch      1
    Sync pulse       2
    Back porch       22
    Whole frame      625
*/

/* VSYNC (TIM2_CH2) and VSYNC_BACKPORCH (TIM2_CH3) */
/* Channel 2 and 3 Configuration in PWM mode */
TIM_SelectSlaveMode(TIM2, TIM_SlaveMode_Gated);
TIM_SelectInputTrigger(TIM2, TIM_TS_ITR0);

TimerPeriod = 625;          /* Vertical lines */
Channel2Pulse = 2;         /* Sync pulse */
Channel3Pulse = 24;        /* Sync pulse + Back porch */

TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure); // always
init the time base first!

TIM_TimeBaseStructure.TIM_Prescaler = 0;
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseStructure.TIM_Period = TimerPeriod;
TIM_TimeBaseStructure.TIM_ClockDivision = 0;
TIM_TimeBaseStructure.TIM_RepetitionCounter = 0;
TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);

TIM_OCStructInit(&TIM_OCInitStructure); // always init
the OC structure first!

TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM2;
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_OutputNState = TIM_OutputNState_Enable;
TIM_OCInitStructure.TIM_Pulse = Channel2Pulse;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_Low;
TIM_OCInitStructure.TIM_OCNPolarity = TIM_OCNPolarity_Low;
TIM_OCInitStructure.TIM_OCIdleState = TIM_OCIdleState_Reset;
TIM_OCInitStructure.TIM_OCNIIdleState = TIM_OCIdleState_Reset;
TIM_OC2Init(TIM2, &TIM_OCInitStructure);
```

EK-1(Devam) Simülâtör Kodları

```
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_Inactive;
TIM_OCInitStructure.TIM_Pulse = Channel3Pulse;
TIM_OC3Init(TIM2, &TIM_OCInitStructure);

/* TIM2 counter enable and output enable */
TIM_CtrlPWMOutputs(TIM2, ENABLE);

/* Interrupt TIM2 */
nvic.NVIC_IRQChannel = TIM2_IRQn;
nvic.NVIC_IRQChannelPreemptionPriority = 1;
nvic.NVIC_IRQChannelSubPriority = 0;
nvic.NVIC_IRQChannelCmd = ENABLE;

NVIC_Init(&nvic);
TIM_ITConfig(TIM2, TIM_IT_CC3, ENABLE);

/* Interrupt TIM1 */
nvic.NVIC_IRQChannel = TIM1_CC_IRQn;
nvic.NVIC_IRQChannelPreemptionPriority = 1;
nvic.NVIC_IRQChannelSubPriority = 0;
nvic.NVIC_IRQChannelCmd = ENABLE;

NVIC_Init(&nvic);
TIM_ITConfig(TIM1, TIM_IT_CC2, ENABLE);

TIM_Cmd(TIM2, ENABLE);
TIM_Cmd(TIM1, ENABLE);

// while (1) ; // debug stop here so we can
confirm timer setup
}

void SPI_Configuration(void)
{
    NVIC_InitTypeDef          nvic;
    SPI_InitTypeDef           SPI_InitStructure;
    DMA_InitTypeDef           DMA_InitStructure;
    GPIO_InitTypeDef          GPIO_InitStructure;

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    /* Configure PB5 for Alternate Function 5 (SPI1) */
    GPIO_PinAFConfig(GPIOB, GPIO_PinSource5, GPIO_AF_SPI1);

    SPI_I2S_DeInit(SPI1);
    SPI_Cmd(SPI1, DISABLE);
    DMA_DeInit(DMA_STREAM);

    DMA_StructInit(&DMA_InitStructure);
    DMA_InitStructure.DMA_PeripheralBaseAddr = (uint32_t)&SPI1->DR;
    DMA_InitStructure.DMA_Memory0BaseAddr = (u32) &fb[0][0];
    DMA_InitStructure.DMA_DIR = DMA_DIR_MemoryToPeripheral;
    DMA_InitStructure.DMA_BufferSize = VTOTAL;
```

EK-1(Devam) Simülâtör Kodları

```
    DMA_InitStructure.DMA_PeripheralInc =
DMA_PeripheralInc_Disable;
    DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
    DMA_InitStructure.DMA_PeripheralDataSize =
DMA_PeripheralDataSize_Byte;
    DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_Byte;
    DMA_InitStructure.DMA_Mode = DMA_Mode_Normal;
    DMA_InitStructure.DMA_Priority = DMA_Priority_High;
    DMA_InitStructure.DMA_Channel = DMA_CHANNEL; // added
channel number
    DMA_InitStructure.DMA_FIFOMode = DMA_FIFOMode_Disable; //
added FIFO mode
    DMA_InitStructure.DMA_MemoryBurst = DMA_MemoryBurst_Single;
    DMA_InitStructure.DMA_PeripheralBurst = DMA_MemoryBurst_Single;
    DMA_Init(DMA_STREAM, &DMA_InitStructure);

    SPI_StructInit(&SPI_InitStructure);
    SPI_InitStructure.SPI_Direction = SPI_Direction_1Line_Tx;
    SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
    SPI_InitStructure.SPI_DataSize = SPI_DataSize_8b;
    SPI_InitStructure.SPI_CPOL = SPI_CPOL_Low;
    SPI_InitStructure.SPI_CPHA = SPI_CPHA_2Edge;
    SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
    SPI_InitStructure.SPI_BaudRatePrescaler =
SPI_BaudRatePrescaler_2; // use 4 for 400 pixels, 2 for 800 pixels
    SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB;
    SPI_InitStructure.SPI_CRCPolynomial = 7;
    SPI_Init(SPI1, &SPI_InitStructure);

    SPI_CalculateCRC(SPI1, DISABLE);

    nvic.NVIC_IRQChannel = DMA_STREAM_IRQ;
    nvic.NVIC_IRQChannelPreemptionPriority = 0;
    nvic.NVIC_IRQChannelSubPriority = 1;
    nvic.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&nvic);

    DMA_STREAM->CR &= ~DMA_SxCR_EN; // clear the EN
bit to disable the stream
    DMA_STREAM->NDTR = VTOTAL; // set number of
bytes to transfer
    DMA_STREAM->M0AR = (u32) &fb[0][0]; // set start of frame
buffer

    SPI_I2S_DMACmd(SPI1, SPI_I2S_DMAREq_Tx, ENABLE); // allow
Tx interrupt to generate DMA requests
    SPI_Cmd(SPI1, ENABLE);

// while (1) // debug send a stream
of chars out SPI
// {
//     SPI1->DR = 0x55;
//     while (SPI1->SR & SPI_FLAG_TXE == RESET) ;
// }

    DMA_ITConfig(DMA_STREAM, DMA_IT_TC, ENABLE);
// VIDEO_DMA->LIFCR = 0x003d0000; // hard-coded
clear of all DMA2 interrupt flags
    DMA_STREAM->CR |= DMA_SxCR_EN; // set the EN
```


EK-1(Devam) Simülâtör Kodları

```
bit to enable the stream
    LED_ON(GREEN); // debug
}

void TIM1_CC_IRQHandler(void)
{
    if (vflag)
    {
        // VIDEO_DMA->LIFCR = 0x003d0000; // hard-coded
        clear of all DMA2 interrupt flags
        DMA_STREAM->CR |= DMA_SxCR_EN; // set the EN
        bit to enable the stream
        // LED_ON(RED); // debug
        // LED_OFF(ORANGE); // debug
    }
    TIM1->SR &= ~TIM_IT_CC2; // 0xFFFB;
    //~TIM_IT_CC2;
}

void TIM2_IRQHandler(void)
{
    vflag = 1;
    TIM2->SR &= ~TIM_IT_CC3; // 0xFF7;
    //~TIM_IT_CC3;
}

void DMA_STREAM_IRQHANDLER(void)
{
    // VIDEO_DMA->LIFCR = 0x003d0000; // hard-coded
    clear of all DMA2 interrupt flags
    VIDEO_DMA->LIFCR = DMA_LIFCR_CTCIF3; // clear the transfer
    complete interrupt flag
    DMA_STREAM->CR &= ~DMA_SxCR_EN; // clear the EN
    bit to disable the stream
    DMA_STREAM->NDTR = VTOTAL; // reload the
    number of bytes to send
    // LED_OFF(RED); // debug
    // LED_ON(ORANGE); // debug

    vdraw++;

    if (vdraw == 1) // was 3
    {
        vdraw = 0;

        vline++;

        if (vline == VID_VSIZE)
        {
            vdraw = vline = vflag = 0;
            DMA_STREAM->M0AR = (u32) &fb[0][0];
        } else {
            DMA_STREAM->M0AR += VTOTAL;
        }
    }
}

void vidClearScreen(void)
{
    u16 x, y;
```

EK-1(Devam) Simülâtör Kodları

```
    for (y = 0; y < VID_VSIZE; y++)
    {
        for (x = 0; x < VTOTAL; x++)
        {
            fb[y][x] = 0;
        }
    }
}

void vidInit(void)
{
    GPIO_InitTypeDef      GPIO_InitStructure;

    GPIO_InitStructure.GPIO_Pin = ALL_LEDS;
    // select the pins to modify
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    // set the mode to output
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    // set the I/O speed to 100 MHz
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    // set the output type to push-pull
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;           //
set the pull-up to none
    GPIO_Init(LEDS_GPIO_PORT, &GPIO_InitStructure);
    // do the init
    LED_ON(BLUE);

    SPI_Configuration();
    TIMER_Configuration();
    vidClearScreen();
}
```

EK-2 Araç Kodları

main.c

```
#include "stm32f10x.h"
#include "stm32f10x_rcc.h"
#include "stm32f10x_gpio.h"
#include "stm32f10x_usart.h"
#include "misc.h"
#include <math.h>
#include <stdio.h>
//*****A_port*****
#define usarttx          GPIO_Pin_9
#define ilerigeripin    GPIO_Pin_1
#define sagsolpin       GPIO_Pin_7
//*****B_port*****
#define Alpin           GPIO_Pin_12 //B12
#define A2pin          GPIO_Pin_13 //B13
#define B1pin          GPIO_Pin_14 //B14
#define B2pin          GPIO_Pin_15 //B15
```

EK-2(Devam) Araç Kodları

```
#define SagPWMPin      GPIO_Pin_6 //B6 TIM4_CH1
#define SolPWMPin     GPIO_Pin_7 //B7 TIM4_CH2

volatile char buffer[15]= {'\0'};

volatile unsigned int ilerigeri;
volatile unsigned int sagsol;

volatile double arachizi;
volatile double d_acisi;
double L=0.740;           //Dingil mesafesi
double K=0.4;            //king pim mesafesi
double d=0.65;           //Arka iki teker arası mesafesi
double Rteker=0.390/2;   //teker yarıçapı
double wasol,wasag;
double pi=3.14159265;
double Rai=0,Rad=0,Rmerkez=0;

void SystemC(void) {
    ErrorStatus Hse;
    RCC_DeInit();
    RCC_HSEConfig(RCC_HSE_ON);
    Hse=RCC_WaitForHSEStartUp();
    if(Hse){
        RCC_HCLKConfig(RCC_SYSCLK_Div1);
        RCC_PCLK1Config(RCC_HCLK_Div2);
        RCC_PCLK2Config(RCC_HCLK_Div1);
        RCC_PLLConfig(RCC_PLLSource_HSE_Div1,RCC_PLLMul_9);
        RCC_PLLCmd(ENABLE);
        while(!RCC_GetFlagStatus(RCC_FLAG_PLLRDY)){
        }
        RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);
        while(RCC_GetSYSCLKSource() !=0x08){}
    }else{
        while(1){}
    }
}

void GpioC(void) {
    GPIO_InitTypeDef gpioconf;

    gpioconf.GPIO_Pin = ilerigeripin|sagsolpin;
    gpioconf.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    gpioconf.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &gpioconf);

    gpioconf.GPIO_Pin = usarttx;
    gpioconf.GPIO_Mode = GPIO_Mode_AF_PP;
    gpioconf.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &gpioconf);

    gpioconf.GPIO_Pin=SagPWMPin|SolPWMPin;
    gpioconf.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_Init(GPIOB, &gpioconf);

    gpioconf.GPIO_Pin=Alpin|A2pin|B1pin|B2pin;
    gpioconf.GPIO_Mode = GPIO_Mode_Out_PP;
    GPIO_Init(GPIOB, &gpioconf);
}
```

EK-2(Devam) Araç Kodları

```
        GPIO_ResetBits(GPIOB, Alpin|B1pin|A2pin|B2pin);
    }

void RccC(void) {
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1
|RCC_APB2Periph_GPIOA |RCC_APB2Periph_GPIOB, ENABLE);
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2
RCC_APB1Periph_TIM3, ENABLE);
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);
}

void UsartC(void) {
    USART_InitTypeDef usartconf;

    USART_StructInit(&usartconf);
    usartconf.USART_BaudRate=115200;
    usartconf.USART_Mode=USART_Mode_Tx;
    USART_Init(USART1, &usartconf);

    USART_Cmd(USART1, ENABLE);
}

void Timer4C(void) {
    TIM_TimeBaseInitTypeDef timconf;
    TIM_TimeBaseStructInit(&timconf);
    timconf.TIM_Prescaler=360-1;
    timconf.TIM_Period=100-1;
    TIM_TimeBaseInit(TIM4, &timconf);

    TIM_OCInitTypeDef tim4oc;
    tim4oc.TIM_OutputState=TIM_OutputState_Enable;
    tim4oc.TIM_OCMode=TIM_OCMode_PWM2;
    tim4oc.TIM_OCPolarity=TIM_OCPolarity_Low;

    tim4oc.TIM_Pulse=1-1;
    TIM_OC1Init(TIM4, &tim4oc);
    TIM_OC1PreloadConfig(TIM4, TIM_OCPreload_Enable);

    tim4oc.TIM_Pulse=1-1;
    TIM_OC2Init(TIM4, &tim4oc);
    TIM_OC2PreloadConfig(TIM4, TIM_OCPreload_Enable);

    TIM_Cmd(TIM4, ENABLE);
}

void Timer2C(void) {
    TIM_TimeBaseInitTypeDef timconf;
    TIM_TimeBaseStructInit(&timconf);
    timconf.TIM_Prescaler=36;
    TIM_TimeBaseInit(TIM2, &timconf);

    TIM_ICInitTypeDef timicconf;
    timicconf.TIM_Channel=TIM_Channel_2;
    timicconf.TIM_ICPrescaler=TIM_ICPSC_DIV1;
    timicconf.TIM_ICPolarity=TIM_ICPolarity_Rising;
    timicconf.TIM_ICSelection=TIM_ICSelection_DirectTI;
```

EK-2(Devam) Araç Kodları

```
    timicconf.TIM_ICFilter=0;

    //TIM_ICInit(TIM2,&timicconf);
    TIM_PWMConfig(TIM2, &timicconf);
    TIM_SelectInputTrigger(TIM2, TIM_TS_TI2FP2);
    TIM_SelectSlaveMode(TIM2, TIM_SlaveMode_Reset);
    TIM_SelectMasterSlaveMode(TIM2, TIM_MasterSlaveMode_Enable);

    TIM_ITConfig(TIM2,TIM_IT_CC2,ENABLE);

    TIM_Cmd(TIM2, ENABLE);

    NVIC_InitTypeDef timnvc;
    timnvc.NVIC_IRQChannel=TIM2_IRQn;
    timnvc.NVIC_IRQChannelCmd=ENABLE;
    timnvc.NVIC_IRQChannelPreemptionPriority=0;
    timnvc.NVIC_IRQChannelSubPriority=1;
    NVIC_Init(&timnvc);
}

void Timer3C(void) {
    TIM_TimeBaseInitTypeDef timconf;
    TIM_TimeBaseStructInit(&timconf);
    timconf.TIM_Prescaler=36;
    TIM_TimeBaseInit(TIM3,&timconf);

    TIM_ICInitTypeDef timicconf;
    timicconf.TIM_Channel=TIM_Channel_2;
    timicconf.TIM_ICPrescaler=TIM_ICPSC_DIV1;
    timicconf.TIM_ICPolarity=TIM_ICPolarity_Rising;
    timicconf.TIM_ICSelection=TIM_ICSelection_DirectTI;
    timicconf.TIM_ICFilter=0;

    //TIM_ICInit(TIM2,&timicconf);
    TIM_PWMConfig(TIM3, &timicconf);
    TIM_SelectInputTrigger(TIM3, TIM_TS_TI2FP2);
    TIM_SelectSlaveMode(TIM3, TIM_SlaveMode_Reset);
    TIM_SelectMasterSlaveMode(TIM3, TIM_MasterSlaveMode_Enable);

    TIM_ITConfig(TIM3,TIM_IT_CC2,ENABLE);

    TIM_Cmd(TIM3, ENABLE);

    NVIC_InitTypeDef timnvc;
    timnvc.NVIC_IRQChannel=TIM3_IRQn;
    timnvc.NVIC_IRQChannelCmd=ENABLE;
    timnvc.NVIC_IRQChannelPreemptionPriority=0;
    timnvc.NVIC_IRQChannelSubPriority=1;
    NVIC_Init(&timnvc);
}

void TIM2_IRQHandler(void) {
    TIM_ClearITPendingBit(TIM2, TIM_IT_CC2);
    ilerigeri = TIM_GetCapture1(TIM2);
    if(ilerigeri>=1900 && ilerigeri<=2872){
        arachizi=(0.0530*ilerigeri)-152.216;
    }else if(ilerigeri>=2973 && ilerigeri<=3950){
```

EK-2(Devam) Araç Kodları

```
        arachizi=(0.1060*ilerigeri)-315.138;
    }else {
        arachizi=0;
    }
}

void TIM3_IRQHandler(void){
    TIM_ClearITPendingBit(TIM3, TIM_IT_CC2);
    sagsol = TIM_GetCapture1(TIM3);
    if(sagsol>=1900 && sagsol<=2871){
        d_acisi=(0.0530*sagsol)-152.163;
    }else if(sagsol>=2972 && sagsol<=3950){
        d_acisi=(0.0530*sagsol)-157.516;
    }else {
        d_acisi=0;
    }
}

void UsartSend(const unsigned char *pbuffer,unsigned int blong){
    while(blong--){
        USART_SendData(USART1,*pbuffer++);
        while(!USART_GetFlagStatus(USART1,USART_FLAG_TC)){
        }
    }
}

double elek_dif_wai(double d_acisi,double arachizi){
    double wai;
    d_acisi=pi*(d_acisi/180);
    arachizi=(arachizi/6.8);
    Rai=(L/(tan(d_acisi)))-(d/2);
    Rad=(L/(tan(d_acisi)))+(d/2);
    Rmerkez=sqrt(pow((Rai+(d/2)),2)+pow((L/2),2));
    wai=(arachizi*Rai)/(Rmerkez*Rteker);
    return wai;
}

double elek_dif_wad(double d_acisi,double arachizi){
    double wad;
    d_acisi=pi*(d_acisi/180);
    arachizi=(arachizi/6.8);
    Rai=(L/(tan(d_acisi)))-(d/2);
    Rad=(L/(tan(d_acisi)))+(d/2);
    Rmerkez=sqrt(pow((Rai+(d/2)),2)+pow((L/2),2));
    wad=(arachizi*Rad)/(Rmerkez*Rteker);
    return wad;
}

void Hareket(double arachizi,double d_acisi){
    if(arachizi>0){ //ileri yön
        GPIO_SetBits(GPIOB, Alpin|B1pin);
        GPIO_ResetBits(GPIOB, A2pin|B2pin);
        if(d_acisi<0){
            TIM4->CCR1=(int)elek_dif_wad((-1)*d_acisi,
arachizi);
            TIM4->CCR2=(int)elek_dif_wai((-1)*d_acisi,
arachizi);
        }else if(d_acisi>0){
```

EK-2(Devam) Araç Kodları

```
        TIM4->CCR1=(int)elek_dif_wai(d_acisi, arachizi);
        TIM4->CCR2=(int)elek_dif_wad(d_acisi, arachizi);

    }else {
        TIM4->CCR1=(int) (arachizi*0.75);//sag
        TIM4->CCR2=(int) (arachizi*0.75);//sol
    }

}

}else if(arachizi<0){//geri yön
    GPIO_SetBits(GPIOB, A2pin|B2pin);
    GPIO_ResetBits (GPIOB, Alpin|B1pin);
    if(d_acisi<0) {
        TIM4->CCR1=(int)elek_dif_wad((-1)*d_acisi, (-
1)*arachizi);
        TIM4->CCR2=(int)elek_dif_wai((-1)*d_acisi, (-
1)*arachizi);

    }else if(d_acisi>0) {
        TIM4->CCR1=(int)elek_dif_wai(d_acisi, (-
1)*arachizi);
        TIM4->CCR2=(int)elek_dif_wad(d_acisi, (-
1)*arachizi);

    }else {
        TIM4->CCR1=(int) ((-1)*(arachizi*0.75));//sag
        TIM4->CCR2=(int) ((-1)*(arachizi*0.75));//sol
    }
}

}else {
    GPIO_ResetBits (GPIOB, Alpin|B1pin|A2pin|B2pin);
    TIM4->CCR1=(int)arachizi;//sag
    TIM4->CCR2=(int)arachizi;//sol
}

}

int main(void)
{
    const unsigned char mytext[]="Başladı...\n";
    SystemC();
    RccC();
    GpioC();
    UsartC();
    Timer2C();
    Timer3C();
    Timer4C();
    //USART_SendData (USART1, 0x000D);
    //UsartSend(mytext, sizeof(mytext));
    while(1) {
        //sprintf(buffer, "%d\t%d \n", (int)arachizi, (int)d_acisi);
        //showfloat(d_acisi, buffer);
        //UsartSend(buffer, sizeof(buffer));
        //USART_SendData (USART1, 0x000A);
        //TIM4->CCR1=50;
        //TIM4->CCR2=50;
        Hareket(arachizi, d_acisi);
    }
}
```