

**T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**TEKERLEK İÇİ MOTORLU ELEKTRİKLİ ARAÇLARDA ELEKTRONİK
DİFERANSİYEL SİSTEMİN GERÇEKLEŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Merve YILDIRIM

(111113101)

Tezin Enstitüye Verildiği Tarih : 8 Nisan 2015

Tezin Savunulduğu Tarih : 22 Nisan 2015

Tez Danışmanı : Prof. Dr. Hasan KÜRÜM (F.Ü)

Diğer Jüri Üyeleri : Yrd. Doç. Dr. Mehmet ÖZDEMİR (F.Ü)

Yrd. Doç. Dr. Zeki OMAÇ (T.Ü)

NİSAN-2015

(Bu tez çalışması 113M090 nolu TÜBİTAK projesi tarafından desteklenmiştir.)

ÖNSÖZ

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren danışman hocam Sayın Prof. Dr. Hasan KÜRÜM'e en içten teşekkürlerimi sunarım.

Tez çalışmam süresince desteklerini hiç esirgemeyen, donanımın kurulmasında ve tezin oluşumunda çok büyük emeği olan Elektrik Elektronik Mühendisi Sayın Cihangir DERSE'ye, cihazların temin edildiği REKARMA Makina Sanayi Tic. A.Ş.'ye ve kullanımıyla ilgili gereken yardımı sağlayan, değerli fikirleriyle çalışmama büyük katkıda bulunan Sayın Oktay KARAKAYA'ya teşekkür eder, şükranlarımı sunarım.

Elektronik diferansiyel sistemin tasarımında sağladığı önemli katkılarından ötürü Sayın Yrd. Doç. Dr. Eyyüp ÖKSÜZTEPE'ye ve Sayın Arş. Gör. Dr. Burak TANYERİ'ye ve yazılım konusunda verdiği büyük destekler için Sayın Yüce KÜRÜM'e çok teşekkür ederim.

113M090 numaralı TÜBİTAK Projesi'nin sağladığı finansal desteklerden dolayı teşekkür ederim.

Merve YILDIRIM
ELAZIĞ-2015

İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	VI
SUMMARY	VIII
ŞEKİLLER LİSTESİ	X
TABLOLAR LİSTESİ	XIII
SEMBOLLER LİSTESİ	XIV
KISALTMALAR.....	XV
1. GİRİŞ.....	1
1.1. Genel Bilgi.....	1
1.2. Yapılan Çalışmalar	2
1.3. Tezin İçeriği.....	9
2. ELEKTRİKLİ ARAÇLAR.....	11
2.1. Giriş	11
2.2. Elektrikli Araçların Tarihçesi	13
2.3. Elektrikli Araçların Çalışma Prensipleri.....	15
2.3.1. Piller	15
2.3.2. Elektrik Motoru	16
2.3.3. Akü Şarj Ünitesi	16
2.3.4. Hız Kontrol Ünitesi	16
2.4. Araçlarda Elektronik Gelişimi.....	17
2.4.1. Gövde, Konfor ve Kolaylık Sistemleri	18
2.4.2. Güvenlik ve Korunma Sistemleri	18
2.4.3. Elektrik-Elektronik Enerji Sistemleri	19
2.4.4. Bilişim ve Eğlence.....	19
2.4.5. Güç ve Aktarım Sistemleri	19
2.5. Araçlarda Elektronik Sistemler	20
2.6. Sensör Teknolojisi	20

3. ARAÇLARDA VE ENDÜSTRİDE AĞ SİSTEMLERİ.....	22
3.1. Giriş	22
3.2. Motor Yönetim Sistemleri	22
3.3. Elektronik Kontrol Ünitelerinde Bilgi İşleme	23
3.4. Veri İletim Yöntemleri	23
3.5. Haberleşme Protokolleri	24
3.5.1. Açık Sistem Haberleşme Protokolleri	25
3.5.2. Kapalı Sistem Haberleşme Protokolleri	27
3.5.3. Diğer Veri Yolu Sistemleri.....	27
3.6. Elektronik Kontrol Üniteleri Arasında Haberleşme	28
3.6.1. CAN-BUS'ın Yapısal Özellikleri	29
3.6.1.1. CAN-BUS Bilgisi	30
3.6.1.2. CAN Topolojisi	40
3.6.1.3. CAN-BUS Çalışma Mantığı.....	41
3.6.1.4. CAN Mesaj Çerçeveleri.....	42
3.6.1.5. Bit Gönderme İşlemi (Arbitration).....	42
3.6.2. Yerel Bağlantı Ağı (LIN)	47
3.6.3. Medya Odaklı Sistemleri Taşıma Ağı (MOST)	48
3.7. Otomobillerde CAN-BUS Tercih Edilmesinin Nedenleri.....	50
4. TEKERLEK İÇİ MOTORLU ELEKTRİKLİ ARAÇLARDA ELEKTRONİK DİFERANSİYEL SİSTEM TASARIMI	52
4.1. Giriş	52
4.2. Ackermann-Jeantand Modeli.....	52
4.3. Elektronik Diferansiyel Sistemin Modellenmesi.....	55
4.4. CoDeSys Yazılımı	56
4.4.1. CoDeSys Fonksiyonları.....	56
4.4.1.1. CoDeSys'te Proje Oluşturma	56
4.4.1.2. Projenin Test Edilmesi.....	56
4.4.1.3. Hata Onarma.....	57
4.4.1.4. Ek Çevrimiçi Fonksiyonlar.....	57
4.4.2. Proje Bileşenleri	57
4.4.2.1. Proje (Project).....	57
4.4.2.2. Program Organizasyon Birimi (POU)	57

4.4.2.3. Fonksiyon (Function)	57
4.4.2.4. Fonksiyon Bloğu (Function Block)	58
4.4.2.5. Program	58
4.4.2.6. PLC_PRG	59
4.4.3. Kaynaklar (Resources)	59
4.4.4. Kütüphaneler (Libraries)	60
4.4.5. Veri Tipleri (Data Types)	60
4.4.6. Görüntüleme (Visualization)	60
4.4.7. CoDeSys Program Dilleri	60
4.4.7.1. Metin Dilleri	60
4.4.7.2. Grafiksel Diller	60
4.4.8. Program Değişkenlerini Gözlemleme	63
4.4.9. Simülasyon	64
4.4.10. CoDeSys Standardı	64
4.4.11. Veri Tipleri	64
4.5. HY-TTC60	65
4.6. PCAN-USB	68
4.7. Elektronik Diferansiyelin CoDeSys Yazılımı ile Gerçekleştirilmesi	69
4.7.1. CAN Hattından Veri Alma Programı	71
4.7.2. CAN Hattından Veri Gönderme Programı	73
4.8. Alınan ve Gönderilen Mesajları Görüntüleme	74
4.9. Deney Düzeneği	75
5. SONUÇLAR.....	85
KAYNAKLAR.....	87
ÖZGEÇMİŞ	93

ÖZET

Günümüzde ulaşım sektörünün genişlemesi bir takım sorunları beraberinde getirmiştir. Bu sorunların başında, çevre kirliliğinin artışı ve fosil yakıtların zamanla tükenmeye başlaması gelmektedir. Zararlı gazların yaydığı emisyonları ortadan kaldırmak, yakıt tasarrufu sağlayarak petrol ithal edilen ülkelere bağımlılığı azaltmak ve hava kirliliğini önlemek amacıyla elektrikli araçlar kullanılmaya başlanmıştır. 2000'li yılların ortalarında, sürücü ve batarya teknolojisindeki gelişmeler, verimli elektrik motorlarının kullanımı ve güvenli sürüşün önem kazanmasıyla elektrikli araçların kullanımı yaygınlaşmıştır. Bunların yanında, bataryaların ek masrafları ve şarj istasyonlarının azlığı gibi bazı olumsuz tarafları da vardır.

Elektrikli araçlarda kaygan ve eğimli yollarda, güç ve momenti çekiş tekerleklerine eşit şekilde dağıtmak için diferansiyel sistemler kullanılır. Bir elektrikli araç viraja girdiğinde, diferansiyel dişlileri devreye girerek virajın dışında kalan tekerleğin hızlı, içinde kalan tekerleğin ise yavaş dönmesini sağlarlar. Diferansiyel sistemin kullanılmadığı durumda, bir tekerlek hızlanırken diğerinin yavaşlaması için frenleme yapılacağından tekerlekler kayacaktır. Bu da güvenli sürüşün sağlanamamasına, yakıt tüketiminin artmasına ve lastiklerin kısa zamanda aşınmasına sebep olur.

Bu çalışmada, aktarım organlarının kullanımını ortadan kaldırmak için motorlar tekerlek içlerine yerleştirilerek bir elektronik diferansiyel sistem gerçekleştirilmiştir. Böylece aktarım organlarından dolayı oluşan mekanik kayıplar, dişlilerin bakım ve onarım masrafları gibi dezavantajlar da ortadan kalkmıştır. Genellikle arka tekerlekler için yapılmış çalışmalar yerine, ön tekerlekler için bir elektronik diferansiyel sistem tasarlanmıştır. Direksiyon açısına ve aracın hızına bağlı olarak tekerleklerin dönmesi gereken hız değerleri Ackermann-Jeantand modelinden alınan matematiksel denklemlerden hesaplanmıştır. Bu denklemler kullanılarak MATLAB/Simulink'te modellenmesi yapılmıştır. Daha sonra, CoDeSys V2.3 yazılımı kullanılarak sistemin tasarımı gerçekleştirilmiştir. Hesaplanan hız bilgilerini motor sürücülerine göndermek için açılımı "Kontrol Alan Ağı" olan "Controller Area Network (CAN-BUS)" protokolü kullanılmıştır. Bu protokol, hızlı, esnek, güvenilir, eşit mesaj erişimli, kablo kullanımının az olması ve dolayısıyla maliyetin düşük ve araç ağırlığının az olması gibi önemli

avantajlarından dolayı tercih edilmiştir. Elektronik diferansiyelin tasarımını doğrulamak için direksiyon açısını 1°'den 15°'ye kadar birer derece aralıklarla değiştirerek motorların hız bilgisi deneysel olarak alınmıştır. CAN-BUS'tan iletilen ön tekerlek içi motorların hız bilgileri, deneysel sonuçlarla ve Simulink modelinden alınan hız değerleriyle kıyaslanarak elde edilen sonuçlar doğrulanmıştır.

Anahtar Kelimeler: Elektrikli Araç, Elektronik Diferansiyel, Kontrol Alan Ağı.

SUMMARY

IMPLEMENTATION OF ELECTRONIC DIFFERENTIAL SYSTEM OF IN-WHEEL ELECTRIC VEHICLE

Nowadays, expansion of transportation sector causes some issues. The most important ones are increasing of environment pollution and depletion of fossil fuels. Therefore, Electric Vehicles (EVs) are used to eliminate the emission emitted from the harmful gases, reducing dependence on countries imported oil by providing fuel-saving, and prevention of air pollution. In the middle of 2000s, developments in driver and battery technology, using of efficient electric motors, and becoming the importance of safe driving increase using of EVs. On the contrary, there are also disadvantages such as additional costs of batteries and lack of charging stations.

Differential systems for EVs are used in slippery and sloping roads to distribute power and torque equally to the traction wheels. Once an EV enters to the cornering, the wheel which is outside of the cornering is provided to rotate fast and the wheel which is inside of the cornering is provided to rotate slowly by differential gears. In case of not using the differential gears, when a wheel accelerates, the wheels slip due to braking to decelerate the other one. This also causes unsafe driving, increasing of fuel consumption and tyre wear.

In this study, an electronic differential system of in-wheel EV has been realized for eliminating the use of powertrains. In this way, some drawbacks such as mechanical losses, maintenance and repair costs of gears which are caused by the powertrains are also eliminated. An electronic differential system for front wheels instead of studies generally realized for rear wheels has been designed. According to the change of the steering angle and speed of EV, required speed values of the wheels have been estimated by using mathematical equations obtained from Ackermann-Jeantand model. By using these equations, MATLAB/Simulink modelling has been realized. Then, an electronic differential system has been designed by CoDeSys V2.3 Software. "Controller Area Network (CAN-BUS)" protocol has been used for transmitting estimated speed values to the motor drivers. This protocol has been preferred for a lot of main advantages such as being fast, flexible, and reliable, equal message access, using fewer cables thus, low cost and low vehicle weight. In order to verify the design of the electronic differential system,

speed values have been taken as experimental by changing from 1° to 15° with a degree range. The speed values of front in-wheel motors received by CAN-BUS have been verified by comparing experimental results and motor speed values obtained from Simulink model.

Keywords: Electric Vehicle, Electronic Differential, Controller Area Network (CAN-BUS).

ŞEKİLLER LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1. Elektrikli aracın yapısı.	12
Şekil 2.2. 1904 yılında bir Alman elektrikli arabası.....	14
Şekil 2.3. 1913 yılında Thomas Edison ve bir Detroit elektrik arabası.....	14
Şekil 2.4. 2010 yılında piyasaya çıkan Mitsubishi MiEV.	14
Şekil 2.5. 2008 yılında piyasaya çıkan Tesla Roadster	14
Şekil 2.6. Elektrikli aracın çalışma sistemi	15
Şekil 2.7. 2008 model Chevrolet Volt.	16
Şekil 2.8. Bir otomobilde güç aktarım organları.	17
Şekil 2.9. ABS sisteminin pazar payının genişlemesi.	18
Şekil 2.10. Araç üzerindeki sensör ve kontrol birimleri.....	21
Şekil 3.1. EKÜ'nün iç yapısı.....	23
Şekil 3.2. Araç haberleşme ağ sistemleri.....	29
Şekil 3.3. Haberleşme sistemlerinin kullanım yerleri.....	29
Şekil 3.4. CAN hattında veri iletimi.	30
Şekil 3.5. Kontrol üniteli CAN-BUS ve hat sonu dirençleri.	31
Şekil 3.6. CAN-BUS sisteminin çalışma sırası.	32
Şekil 3.7. CAN-BUS Katmanları.	32
Şekil 3.8. CANOpen haberleşmesi.....	34
Şekil 3.9. OSI ağ modelinde CAN ve CANopen standartları.	35
Şekil 3.10. 11 bitlik CAN tanımlayıcısı.	39
Şekil 3.11. CAN protokolünde kullanılan topolojiler.	41
Şekil 3.12. Tipik bir otomobil CAN bağlantı şekli.	41
Şekil 3.13. Standart CAN-BUS veri çerçeve yapısı.	42
Şekil 3.14. Genişletilmiş CAN-BUS veri çerçeve yapısı.	42
Şekil 3.15. Standart istek çerçevesi.	45
Şekil 3.16. Genişletilmiş istek çerçevesi.	45
Şekil 3.17. Hata ve taşma çerçeve yapısı.....	46
Şekil 3.18. Bit dolgu işlemi	46
Şekil 3.19. Bit gönderme işlemi.	47

Şekil 3.20. LIN ana kontrol ünitesi.....	48
Şekil 3.21. MOST hattında kullanılan sistemler.....	49
Şekil 3.22. MOST kontrol ünitesinin yapısı.	49
Şekil 4.1. Düşük hızda sürüş eğiminin Ackermann-Jeantand modeli	52
Şekil 4.2. Bir araçtan alınan boyutlar	54
Şekil 4.3. Direksiyon açısı ve aracın hızına göre tekerlek içi motorların hızlarını hesaplamak için gerçekleştirilen Simulink model	55
Şekil 4.4. Elektronik diferansiyel sistemin Simulink modeli	55
Şekil 4.5. IL dilinde bir fonksiyon bloğu örneği	58
Şekil 4.6. Bir program örneği.	59
Şekil 4.7. SFC'de bir ağ örneği.....	62
Şekil 4.8. FBD örneği.....	62
Şekil 4.9. CFC örneği	62
Şekil 4.10. Bağlantılar ve sargılardan oluşan LD'de bir ağ örneği.	63
Şekil 4.11. PLC_PRG'de işaret edilen değerlerin gözlemlenmesi	64
Şekil 4.12. HY-TTC 60 modülü	66
Şekil 4.13. CAN ara yüzü.....	66
Şekil 4.14. HY-TTC 60'ın blok diyagramı.	67
Şekil 4.15. PCAN-USB	68
Şekil 4.16. PCAN-USB pinleri.....	68
Şekil 4.17. PCAN-USB bağlantı şeması.	68
Şekil 4.18. CoDeSys'te gerçekleştirilen elektronik diferansiyel sistemin akış diyagramı ..	69
Şekil 4.19. Hedef ayarları (Target Settings)	70
Şekil 4.20. Yeni bir POU oluşturma.....	70
Şekil 4.21. PLC konfigürasyon ayarları	71
Şekil 4.22. CAN alma/CAN gönderme programı oluşturma	71
Şekil 4.23. PLC konfigürasyonunda alıcı tamponun tanımlanması	73
Şekil 4.24. Motor kontrolü için CoDeSys'te gerçekleştirilen simülasyon.....	74
Şekil 4.25. Sürücünün "eds" dosyasını yükleme	75
Şekil 4.26. PDO ve SDO ayarları	76
Şekil 4.27. Değişkenlere isim atama	76
Şekil 4.28. Motorun sürücüye bağlanma şeması	76
Şekil 4.29. Bir motor sürücüsünün CAN-BUS hattına bağlantı blok diyagramı	76

Şekil 4.30. Sürücülerin CAN hattına bağlantı şeması.	78
Şekil 4.31. Enkoder	78
Şekil 4.32. Ultrasonik sensör	78
Şekil 4.33. Deney düzeneği.	79
Şekil 4.34. CoDeSys programında hesaplanan değerler.....	80
Şekil 4.35. PCAN-View programından alınan ve gönderilen mesajlar.....	81
Şekil 4.36. Ön iç tekerlek hızının CoDeSys, deneysel çalışma ve Simulink sonuçlarının karşılaştırması.....	82
Şekil 4.37. Ön dış tekerlek hızının CoDeSys, deneysel çalışma ve Simulink sonuçlarının karşılaştırması.....	83
Şekil 4.38. Ön iç tekerlek hızının CoDeSys, deneysel çalışma ve Simulink sonuçlarının karşılaştırması (Direksiyon açısı ters yön için)	84
Şekil 4.39. Ön dış tekerlek hızının CoDeSys, deneysel çalışma ve Simulink sonuçlarının karşılaştırması (Direksiyon açısı ters yön için)	84

TABLÖLAR LİSTESİ

	<u>Sayfa No</u>
Tablo 3.1. Açık ve kapalı haberleşme sistemleri	25
Tablo 3.2. Açık sistem haberleşme protokollerinin karşılaştırması.	27
Tablo 3.3. Seri haberleşme protokollerinin pozitif ve negatif yönleri	28
Tablo 3.4. Bir SDO'nun temel yapısı.....	35
Tablo 3.5. Yeni bir SDO yapısı	36
Tablo 3.6. Acil durum mesajları.....	37
Tablo 3.7. NMT mesajı.....	37
Tablo 3.8. NMT slave'in cevap mesajı	37
Tablo 3.9. Bit değerleri ve durumları	38
Tablo 3.10. NMT mesaj formatı	38
Tablo 3.11. CS değerleri.....	38
Tablo 3.12. Başlatma mesajı.....	38
Tablo 3.13. CAN tanımlayıcılarının ataması	39
Tablo 3.14. PDO mesaj örneği.	39
Tablo 3.15. İletim mesafelerine göre haberleşme hızları	42
Tablo 3.16. CAN-BUS protokolünün çerçeve alanları ve değerleri.....	45
Tablo 4.1. Integer veri tipleri.....	68
Tablo 4.2. HY-TTC 60 bağlantı pinleri.	68
Tablo 4.3. CoDeSys'ten alınan sonuçların deneysel ve MATLAB/Simulink sonuçlarıyla karşılaştırması.....	82
Tablo 4.4. CoDeSys'ten alınan sonuçların deneysel ve MATLAB/Simulink sonuçlarıyla karşılaştırması (Direksiyon açısı ters yön için)	79

SEMBOLLER LİSTESİ

$\delta_{\text{öi}}$	: Ön iç tekerlek açısı
$\delta_{\text{öd}}$	: Ön dış tekerlek açısı
L	: Ön tekerlek ile arka tekerlek arasındaki mesafe
K	: King pim arasındaki mesafe
ρ	: Direksiyon açısı
$R_{\text{öi}}$	: Ön iç tekerleğin viraj yarıçapı
$R_{\text{öd}}$	: Ön dış tekerleğin viraj yarıçapı
R_{ai}	: Arka iç tekerleğin viraj yarıçapı
R_{ad}	: Arka dış tekerleğin viraj yarıçapı
d	: Arka iki tekerlek arasındaki mesafe
$w_{\text{öi}}$	: Ön iç tekerleğin açısal hızı
$w_{\text{öd}}$	: Ön dış tekerleğin açısal hızı
w_{ai}	: Arka iç tekerleğin açısal hızı
w_{ad}	: Arka dış tekerleğin açısal hızı
R_{merkez}	: Virajın merkeze olan yarıçapı
r	: Tekerlek yarıçapı
V	: Aracın hızı
L_a	: Arka tekerlek ile ağırlık merkezi arasındaki mesafe
n	: Devir sayısı
p	: Kutup sayısı
f	: Frekans

KISALTMALAR

CAN	: Controller Area Network
EV	: Electric Vehicle
HEV	: Hybrid Electric Vehicle
CiA	: CAN in Automation
CSMA/CD	: Carrier Sense Multiple Access with Collision Detection
MCU	: Micro Controller Unit
TTCAN	: Time-Triggered CAN
DA	: Doğru Akım
CARB	: California Air Resources Board
ABS	: Anti-lock Braking System
ASP	: Automotic Stability Program
ACC	: Adaptive Cruise Control
EKÜ	: Elektronik Kontrol Ünitesi
ICE	: Internal Combustion Engine
FPGA	: Field Programmable Gate Array
UART	: Universal Asynchronous Receiver/Transmitter
CAL	: CAN Application Layer
CMS	: Content Management System
SDO	: Service Data Object
PDO	: Process Data Object
SYNC	: Synchronous
ASYNC	: Asynchronous
NMT	: Network Management Object
CS	: Command Specifier
ID	: Identifier
COB-ID	: Communication Object Identifier
SOF	: Start of Frame
RTR	: Remote Transmit Request
SRR	: Substitute Remote Request
IDE	: Identifier Extension
DLC	: Data Length Code
CRC	: Cyclic Redundancy Check
ACK	: Acknowledgement
EOF	: End of Frame
INT	: Intermission
LIN	: Local Interconnect Network
MOST	: Media Oriented Systems Transport
FOT	: Fiber Optical Transmitter
CoDeSys	: Controlled Development System
FBD	: Function Block Diagram
ST	: Structured Text
LD	: Ladder Diagram
POU	: Program Organization Unit
FUB	: Function Block
IL	: Instruction List
SFC	: Sequential Function Chart

CFC	: Continuous Function Chart Editor
PLC	: Programmable Logic Controller
EDS	: Electronic Data Sheet
ASR	: Anti Schlupf Regelung
ESP	: Electronic Stability Program
GPRS	: General Packet Radio Service
EPROM	: Erasable Programmable Read Only Memmory

1. GİRİŞ

1.1. Genel Bilgi

Günümüzde elektrikli araçların kullanılmaya başlanmasıyla çevre kirliliğine yol açan zararlı gazların etkisi ortadan kaldırılmıştır. Bununla birlikte elektrikli araçlar, sessiz çalışmaları, yüksek verimli olmaları, motorunun daha ucuz olması ve yüksek moment sağlamaları gibi önemli avantajlara sahiptir. Sürücü ve batarya teknolojisindeki gelişmeler ve güvenli sürüşün sağlanmasıyla birlikte bu avantajlar elektrikli araçların kullanımını yaygınlaştırmıştır.

Elektrikli araçlarda, vites kutusu yardımıyla momentin ve hızın değiştirilerek tekerleklerle dağıtılması sağlanır. Böylece otomobil viraja girerken virajın yönüne göre dıştaki tekerleğin içteki tekerleğe göre daha hızlı dönmesi gerekmektedir. Bu işlem gerçekleştirilirken iç tekerleğin frenle hızı azaltılmaya çalışılacağı için iç tekerlek kaymaya başlayacaktır. Bu da yakıt ihtiyacını artırır ve lastiklerin kısa sürede aşınmasına neden olur. Bu sorunu çözmek amacıyla diferansiyel sistemler kullanılır. Genellikle araçlarda kullanılan mekanik diferansiyel sistemler yerine, aktarım organlarını ortadan kaldırmak için motorlar tekerlek içlerine yerleştirilerek elektronik diferansiyel sistemler tercih edilebilir.

Bu çalışmada, elektrikli araçların genellikle arka tekerlekleri için yapılmış elektronik diferansiyel sistemler yerine, ön tekerlekler için bir elektronik diferansiyel sistem tasarlanmıştır. Tasarlanan bu sistemin yazılımı CoDeSys V2.3 programı üzerinden gerçekleştirilmiştir. Elektrikli araçtan alınan direksiyon açısı ve hız bilgisine göre tekerlek içi motorların hız değerlerini hesaplamak için, Ackermann-Jeantand modeli kullanılmıştır. Ön tekerlekler için matematiksel denklemler bu modelden alınarak MATLAB/Simulink'te modellenmesi yapılmıştır. Daha sonra bu denklemler, CoDeSys yazılımına gömülerek hız değerleri hesaplanmış ve bu değerlerin motor sürücülerine gönderilmesi için açılımı "Kontrol Alan Ağı" anlamına gelen "Controller Area Network (CAN-BUS)" haberleşme protokolü kullanılmıştır.

CAN-BUS protokolü, 1987 yılında Robert Bosch Gmbh tarafından geliştirilmiş seri bir haberleşme protokolüdür. 1990'larda CAN özelliklerini belirleyen "CAN in Automation

(CiA)" grubu kurulmuştur. Otomotiv alanında yaygın olarak kullanılan bu protokol, zamanla sensörler, medikal uygulamalar, otomasyon sistemi gibi daha pek çok alanda kullanılmaya başlanmıştır. Ayrıca, ISO 11519 ve ISO 11898 standartlarını desteklemektedir. Bu standartlara göre CAN 2.0A düşük hızlı haberleşme, CAN 2.0B ise yüksek hızlı haberleşme sağlar.

CAN protokolünde mesaj önceliği olmakla birlikte, her mesajın eşit erişimli gönderilmesi sağlanır. Her mesajın bir tanımlayıcı numarası bulunur ve o adrese veri gönderilir. İletişim ortamına erişim yöntemi olarak "Carrier Sense Multiple Access with Collision Detection (CSMA/CD)" denilen bit öncelikli yapı kullanılır. Bu yapı ile mesajların çarpışmaması sağlanır ve hattın uzunluğu sınırlandırılır. 1 Mbit/s'ye kadar haberleşme hızına çıkılabilir. Böylece hızlı ve veri kaybı olmadan haberleşme yapmak mümkündür. CAN-BUS'ın en önemli avantajlarından biri olarak, CAN hattına pek çok cihaz bağlanarak tek hat üzerinden haberleşmeleri sağlanmaktadır. Böylece daha az kablo kullanımı hem aracın ağırlığını azaltır hem de maliyeti düşürür. Bunun yanı sıra, sisteme başka cihazlar eklenip çıkartılabilir ve bu durum diğer cihazları ve sistemin çalışmasını etkilemez. Bu üstün özelliklerinden dolayı elektrikli araçlarda yaygın olarak kullanılan CAN-BUS protokolü, bu çalışmada da tercih edilmiştir. Böylece direksiyon açısı bilgisini almak için kullanılan enkoder, aracın hız bilgisi için kullanılan bir ultrasonik sensör, iki adet asenkron motor, sürücüler ve işlemci modülün tek hat üzerinden haberleşmesi sağlanmıştır. Ayrıca, daha pek çok sensör ya da cihaz hatta bağlanabilir ve veri gönderimi gerçekleştirilebilir. Araç viraja girerken direksiyonun o andaki konumu ve hız bilgisiyle ilişkili olarak ön tekerlek içi motorların dönmesi gereken hız değerleri CAN hattı üzerinden motorlara gönderilmiş ve böylece motorların gereken hızda dönmesi sağlanmıştır. Ayrıca CoDeSys yazılımından alınan sonuçlar deneysel çalışmalarla da desteklenmiştir.

1.2. Yapılan Çalışmalar

Bu bölümde, ilk olarak CAN-BUS protokolü ve elektrikli araçlarda kullanımı, daha sonra elektrikli araçlarda bulunan diferansiyel sistemler ile ilgili yapılan çalışmalara yer verilmiştir.

1999 yılında M. Farsi ve arkadaşları, CAN-BUS'ın özellikleri, donanım ve yazılımda kullanımıyla ilgili detaylı bir çalışma yapmışlardır [1].

2002 yılında Lamia Chaari ve arkadaşları, elektrikli araç için CAN-BUS ağı ile elektronik kontrol üzerine çalışmışlardır [2]. CAN-BUS'ın getirdiği kablo boyutu ve uzunluğunun azalması ile düşük maliyet gibi avantajlarından bahsedilerek veri kontrolü yapılmıştır.

2006 yılında İlker Ünal'ın gerçekleştirdiği çalışmada, CAN üzerinden PIC 18F458 programlaması yapılmıştır [3]. Bunun için donanımı gerçekleştirilerek Visual Basic dili ile PIC programlanmıştır.

Bong-Ju Ahn ve arkadaşlarının yaptığı bir başka çalışma ise, hibrit elektrikli araçlar için bir yakıt hücresi uygulaması ve bunun için geliştirilen CAN-BUS ara yüzünü anlatmaktadır [4].

2008 yılında Ping Ran, Baoqiang Wang ve Wei Wang tarafından, seri haberleşme protokollerinden CAN-BUS ve RS-232 arasındaki dönüşüm yöntemi açıklanarak yeni bir haberleşme dönüştürücüsü ve hata bulma sistemi hakkında bilgiler verilmiştir [5].

Renjun Li ve arkadaşlarının yaptığı bir başka çalışmada, araçlar için bir CAN-BUS gözleme sistemi tasarlanmıştır [6]. Bu sistem CAN-BUS'a gönderilen ya da alınan CAN çerçevelerini görüntüleme imkanı verirken, aynı zamanda çevrim içi olmadan da değerlendirme yapabilmek için dosyalara verilerin kaydını tutar. Kullanıcılar, PC uygulamalı sistemin CAN kanal özelliklerini ve bazı gözlem modlarını ayarlayabilir, USB teknolojisiyle daha pratik şekilde CAN-BUS sistemini test edebilirler.

Frang Li, Lifang Wang ve Chenglin Liao'nun yaptığı bir çalışmada, MATLAB/Simulink tabanlı bir CAN-BUS haberleşme sistemi tasarlanmıştır [7]. Mesajların en kötü durum cevap süresini karşılayan CAN-BUS sisteminin performans indeksleri MATLAB'ta ve CANoe'de yapılan teorik hesaplamalar ve deneylerle bulunmuştur. Hesaplamalar ve deneysel sonuçlar karşılaştırıldığında MATLAB/Simulink'te tasarlanan CAN-BUS sisteminin verimli mesaj transferi yaptığı görülmüştür. Böylece CAN-BUS sistem tasarımı için bir test etme aracı MATLAB/Simulink'te tasarlanmıştır.

2009 yılında Xiao-feng Wan ve arkadaşları, endüstride gerçek zamanlı bir veri haberleşmesinde CAN-BUS teknolojisinin uygulaması üzerine bir çalışma yapmışlardır [8]. Gerçek zamanlı çalışma sistemi olan VxWorks ve CAN-BUS'ın karakteristikleri ve

işlevlerini anlatarak, VxWorks ve Micro Controller Unit (MCU) arasındaki veri aktarımını CAN-BUS ile gerçek zamanlı ve güvenilir şekilde gerçekleştirmesi ile ilgili bilgiler vermişlerdir.

Alper Karaduman tarafından CAN ve zaman tetiklemeli CAN anlamına gelen "Time-Triggered CAN (TTCAN)" sistemlerinin modellenmesi ve performans analizi üzerine bir çalışma yapılmıştır [9]. CAN'deki mesaj önceliğine göre yüksek öncelikli mesajlar düşük gecikmelerle iletilirken, düşük öncelikli mesajlar yüksek gecikmelerle iletilebilir. Bunu gidermek için TTCAN geliştirilerek performansı incelenmiştir.

Wang Jianfeng ve arkadaşlarının yaptığı bir çalışmada, CAN-BUS dağıtık kontrol üzerine elektrikli aracın gövde ağ tasarımı anlatılmıştır [10]. Elektrikli aracın gövde sisteminde donanımın basit, kablounun az kullanılması ve normal veri haberleşmesinin sağlanması istenir. Her bir düğüm belli alanı kontrol eder ve devrenin gövde düğümü temel devre kartı ve geliştirilmiş devre kartı içerir. Temel devre kartı, veri haberleşmesinin ihtiyaçlarını karşılarken, geliştirilmiş devre kartı gövdenin elektriksel olarak gelişmesini sağlar. Devreler bazı modüllere ayrılır. Her modül ayrı ayrı gömülü yazılım sistemi ile geliştirilir ve CAN haberleşme sistemleri test edilir.

İsmail Kara tarafından yapılan bir diğer çalışmada, CAN protokolü incelenerek bir sıcaklık kontrol sistemine uygulanmıştır [11]. Bunun için güvenilir ve hızlı olan zaman tetiklemeli bir işletim sistemi geliştirilmiş, sıcaklık kontrolü için ise PID kontrolör tasarlanmıştır.

2010 yılında Fan Xin ve Hu Chun, CAN-BUS tabanlı elektrikli aracın havalandırma kontrol ağı üzerine bir tasarım ve araştırma yapmışlardır [12]. STC12C5A60S2 mikroişlemcisi kullanılarak CAN-BUS ara yüzü tasarlanmıştır. Gerçek bir ağ sisteminin dijital simülasyonunu donanımsal olarak gerçekleştirmiş ve performansını test etmek için bir araca uygulamışlardır. Böylece gerçek zamanlı ve kararlı bir sistem olduğu görülmüştür.

Qiangsheng Ye, otomobil ağ sistemlerinde CAN-BUS ve Local Interconnect Network (LIN-BUS)'un uygulama ve araştırması üzerine bir çalışma gerçekleştirmiştir [13]. P87C591 mikroişlemcisi CAN kontrolörü olarak kullanılmıştır. Bu çalışmada düşük ve yüksek hızlı CAN-BUS ağlarının yazılımı ve donanımı gerçekleştirilmiş, aktif süspansiyon, motor, hava yastığı, Anti Schlupf Regelung (ASR), Anti-lock Brake System

(ABS), Electronic Stability Program (ESP) ve aktarım kontrolü gibi pek çok kontrol yüksek hızlı CAN-BUS ağı, havalandırma, cihaz, ışıklandırma, pencere, silecek ve kapı kontroller de düşük hızlı CAN-BUS ağı olarak tasarlanmıştır. Ağda veri paylaşımı ve ağ kontrol düğümünün yazılımı gerçekleştirilmiştir.

Li Ran ve arkadaşlarının elektrikli araçlar için CAN-BUS haberleşme ağının tasarımı üzerine yaptıkları çalışmada, master düğümü, aydınlatma, havalandırma, aletler ve kapılar için düğümler CAN-BUS ağında haberleşmektedir [14]. CAN-BUS donanım ara yüz tasarımı, SJA1000 kontrolörü, yüksek hızlı optokuplör 6N137, CAN-BUS sürücüsü 82C250 ve tasarım şemasını içerir. CAN-BUS ağı için yazılım tasarımları ise CAN başlatma birimi, mesaj gönderme, mesaj alma ve kesme servis birimi gibi birimleri kapsamaktadır. Tasarlanan bu sistem yeni araçlara uygulanarak, doğru, kararlı ve güvenilir bir sistem olduğu görülmüştür.

Metin Öztürk hidrojen hibrit otobüslerde CAN-BUS sistemleri ve uygulaması üzerine bir çalışma gerçekleştirmiştir [15]. Bu çalışmada, 3 adet sensör kullanılarak CAN-BUS protokolü ile motor sıcaklık bilgisi, otobüsün kapalı ya da açık olma durumu ve geçen yolcu sayısı bilgileri okunmuştur. Hidrojen hibrit otobüsün üretilmeden önce CAN-BUS haberleşme sisteminin tasarımı için farklı araçlara uygulanan farklı sistemler de incelenmiştir.

Bir başka çalışmada Erhan Dinçer tarafından, CAN-BUS ile dağıtık kontrol uygulaması gerçekleştirilmiştir [16]. Dağıtık sistem olarak dört katlı model asansör kullanılmış, her bir modülde PIC16F877 mikrodenetleyici bulunan altı kontrol modülü tasarlanmıştır. CAN haberleşmesi ise MCP2515 CAN-SPI ve MCP2551 CAN çevirici ile sağlanarak dağıtık kontrolü yapılmıştır.

Fatih Yardım tarafından bir tren eğitim simülatörünün donanımı CAN protokolü ile haberleştirilmiştir [17].

2011 yılında R. Nejat Tuncay ve arkadaşları tekerlek içi motora sahip elektrikli araç uygulamaları için bir elektrik sürücü sistemi tasarlamışlardır [18]. Bunun için 15 kW'lık Fırçasız Doğru Akım (DA) motorlar tasarlanarak üretilmiş ve tekerlek içine yerleştirilmiştir. Direksiyon açısına bağlı bir elektronik kontrol tekniği geliştirilmiştir. Aracın Elektronik Kontrol Ünitesi (EKÜ) ve sürücünün kontrol sistemi arasında bir CAN-

BUS haberleşmesi kurulmuştur. Arka tekerlekler için tasarlanan bu sistem deneysel olarak test edilmiştir.

2012 yılında Utku Civelek tarafından yapılan çalışmada, CAN-BUS ile araç kalibrasyonu, hata teşhisi ve testi için bir yazılım geliştirilmiştir [19]. Bu yazılım bir test düzeneğini ile doğrulanmıştır.

2014 yılında Joshua R. Henderson, James M. Conrad ve Craig Pavlich'in gerçekleştirdiği çalışmada arazi araçları için CAN-BUS kullanımı ele alınmıştır [20]. Gelecekteki gelişimi için sağlam bir alt yapı oluşturmak ve kablo ihtiyacını azaltmak için CAN-BUS haberleşme ağı ile aracın frenleme sistemi kontrol edilmiştir.

Muhammed Yakup Yılmaz tarafından yol araçları için bir CAN hattı tanım sistematığı gerçekleştirilmiş ve tasarlanmıştır [21]. Buna göre CAN hattından alınan veriler General Packet Radio Service (GPRS) üzerinden uzak kontrol merkezine gönderilerek tasarlanan hata ve veri yönetim sistemi ile analizi yapılmıştır. Ayrıca, araçtan alınan verilerin iletimini sağlayacak bir veri tabanı ve sunucusu tasarlanmıştır.

Elektrikli araçlarda bulunan diferansiyel sistemler üzerine yapılan çalışmalar:

2000 yılında Ju-Sang Lee ve arkadaşlarının yaptığı bir çalışmada, elektrikli araçlarda elektrik diferansiyel sistem için bir sinir ağı modeli tanımlanmıştır [22]. Bir araç eğimli yolda giderken sarsıntıyı önlemek için içindeki tekerleğin hızı dıştaki tekerlekten farklı olmalıdır. Her tekerlek bağımsız itme kuvvetine sahip olduğundan, vites değiştirmek için bir elektrik diferansiyel sistem gereklidir. Aracın hızı ve direksiyon açısıyla ilişkili olan diferansiyel sistemin lineer olmayan davranışını analiz etmek güç olduğundan sinir ağı modeli geliştirilmiştir. Bu modelle, direksiyon açısının ve aracın değişken hız değerlerine göre içindeki ve dıştaki tekerleğin hız değerleri bulunmuştur.

2003 yılında J. T. Economou ve arkadaşları ön tekerlekler için Ackermann modeli tanımlamışlardır [23]. Model, geleneksel dinamik ve kinematik denklemler ile viraja girerken ön ve arka tekerleklerin kuvvet karakteristikleri için Sugeno bulanık modelini kullanan bir hibrit yaklaşım ile geliştirilmiştir.

2004 yılında Sinclair Gair ve arkadaşlarının yaptığı bir çalışmada, doğrudan sürürlü elektrikli araçlar için kayan mod kontrolörlü 4 anahtar ve 3 faz inverter kullanarak arka tekerlekler için bir elektronik diferansiyel sistem tasarlanmıştır [24].

2008 yılında Abdelhakim Haddoun ve arkadaşları, bir elektrikli aracın bağımsız iki arka tekerleğindeki sürücülerinin simulasyonunu, analizini ve modellemesini yapmışlardır [25]. Simülasyon sonuçları 37 kW'lık iki asenkron motor üzerinde test edilerek önerilen yaklaşımdan istenilen sonuçlar elde edilmiştir.

2009 yılında Manu Jain ve Sheldon S. Williamson'ın gerçekleştirdiği çalışmada, iki ve dört tekerlekli araçlarda tekerlek içi motor sürücülerini için çeşitli motorlar ve kontrol stratejilerinden bahsedilmiştir [26].

Y. E. Zhao, J. W. Zhang ve X. Q. Guan, iki tekerlek içi motordan oluşan bir elektrikli araç için bulanık mantık kontrol yöntemini kullanarak yeni bir elektronik diferansiyel sistem geliştirmişlerdir [27]. Bulanık mantık, lineer olmayan sistem için her tekerleğin kayma oranını hesaplamada kullanılmıştır. Önerilen kontrol yöntemi, Matlab/Simulink'ten alınan sonuçlarla doğrulanmıştır.

2010 yılında Yong Zhou ve arkadaşları tekerlek içi motorlardan oluşan elektrikli araçlar için elektronik diferansiyelin kontrol stratejisi üzerine bir çalışma yapmışlardır [28]. Ackermann-Jeantand modeli ve dört teker direksiyon sistemini kullanarak dört tekerleğin hızları ayrı ayrı hesaplanmıştır. Aracın sabit hızda gitme ya da hızlanma ve yavaşlama durumlarına göre çalışma şartları araştırılarak, 4 tekerlek içi motordan oluşan elektrikli araç için elektronik diferansiyel sistemin kontrol stratejisi geliştirilmiştir.

Rafael C. B. Sampaio ve arkadaşları, 4×4 hibrit seyahat araçlarının şehir içi hız sınırlarında arkadan çekişli kontrol için sağlamlık kontrolü üzerine çalışmışlardır [29]. Bu çalışmanın amacı, arka tekerlekler için bir elektromekanik kontrol sistemi tasarlamaktır.

2011 yılında Chih-Hsien Yu ve arkadaşları, doğrudan sürüşlü elektrikli araçlar için elektronik stabilite kontrolü üzerine çalışmışlar ve arka iki tekerleğin analizini, modellemesini ve simulasyonunu gerçekleştirmişlerdir [30].

Li Zhai ve Shouwan Dong, dört tane tekerlek içi motor için elektronik diferansiyel hız ve direksiyon açısı kontrolü üzerine bir yaklaşım önermişlerdir [31]. Sinir ağı PID elektronik diferansiyelden oluşan moment ve hız kontrol stratejisi, dört tekerleğin hızını hesaplamak için kullanılmıştır. Farklı referans hız ve direksiyon açısı değerlerine göre alınan simülasyon sonuçlarıyla aracın düşük hızda kararlılığı geliştirebilmektedir.

S. You ve arkadaşları, elektrikli araçlarda bulunan elektronik diferansiyelin kullanımı için basit bir hız kontrolü yapmışlardır [32]. Önerilen bu kontrol yaklaşımının geçici ve sürekli durum performanslarının mevcut hız kontrollerinden daha iyi olduğu görülmüştür.

2012 yılında Rafael Coronel Bueno Sampaio ve arkadaşları, yapay zeka, PID ve H_{∞} gibi çeşitli kontrolörleri kullanan, arkadan çekişli yeni bir elektronik diferansiyel sistem tasarlamışlardır [33]. Simülasyonunu gerçekleştirerek deneysel sonuçları elde etmişlerdir.

2013 yılında Xiaodong Wu, Min Xu ve Lei Wang, dört tekerlekli elektrikli araçlarda Ackermann modeli yerine basit yapılı diferansiyel hız yönlendirme modelini tercih etmişlerdir [34]. Fakat bu yöntemde, diferansiyel hız kontrolü zordur. Bu çalışmada aracın kinematik model analiziyle tekerlek parametreleri ve dönme davranışı arasındaki ilişki araştırılmıştır.

Azeddine Draou, elektronik diferansiyelin tekerlek içine veya tek bir dişliyle tekerleklerle yerleşen verimli, hafif ve küçük motorların kullanımı gibi getirdiği avantajlarından ötürü iki ve dört tekerlekli araçlar için bir diferansiyel sistem önermiştir [35]. Bunun için senkron permanent magnet motorlar kullanılmıştır. Önerilen yaklaşım her bir tekerlekteki motorlar için doğrudan moment kontrolü gerektirir. Yolun eğimine ve yapısına göre farklı simülasyonlar gerçekleştirilmiş ve alınan sonuçlara göre aracın eğimli yolda istikrarlı şekilde ilerlediği görülmüştür.

Hakan Kahveci ve arkadaşları, bir elektrikli aracın arka iki tekerlekleri için bulanık mantık hız kontrolü kullanarak bir elektronik diferansiyel sistem önermişlerdir [36]. Arka tekerleklerde iki adet fırçasız DA motorlar kullanılmıştır. Elektronik diferansiyel sistem tasarımında Ackermann-Jeantand modeli tercih edilmiştir. Önerilen yaklaşımın, düz yol ve eğimli yol duruma göre simülasyonları Matlab/Simulink'te gerçekleştirilmiştir. Alınan sonuçlara göre sistemin iyi olduğu görülmüştür.

2014 yılında Balasharmila Rao Munusamy ve Joerg Dieter Weigl, çift motor elektrik sürüş sisteminde Ackermann geometrisine bağlı olarak arka tekerlekler için moment vektörü tasarlamışlardır [37]. Böylece arka iki tekerleğin hızları elektronik diferansiyel sistem ile direksiyon açısı ve aracın hızına göre elde edilerek deneysel sonuçlarla doğrulanmıştır.

Xiaodong Wu ve arkadaşlarının yaptığı çalışmada, dört tekerlekli elektrikli aracın hız kontrolü gerçekleştirilmiştir [38]. Ackermann-Jeantand modeline göre hız kontrolüyle elektronik diferansiyel için tekerleklerin kinematik modeli tanımlanmıştır. Bu yöntem, lastiğin aşınmasını azaltmış ve kararlılığı arttırmıştır. Sistemin simülasyonu yapılarak deneysel sonuçlarla verimliliği kanıtlanmıştır.

1.3. Tezin İçeriği

Tezin birinci bölümünde, elektrikli araçlardaki elektronik diferansiyel sistemler ile CAN-BUS protokolünün yapısı ve özellikleri kısaca anlatılmıştır. Bu tezde gerçekleştirilen çalışma özetlenerek CAN-BUS protokolü ve elektrikli araçlardaki diferansiyel sistemler üzerine yapılan çalışmalardan bahsedilmiştir.

İkinci bölümünde, elektrikli araçlar, yapısı, tarihi gelişimi, çalışma prensipleri ve araçlarda elektroniğin gelişimi hakkında bilgiler verilmiştir.

Üçüncü bölümde, otomobillerde ve endüstride kullanılan ağ sistemleri anlatılmıştır. Tüm kapalı ve açık sistem haberleşme protokolleri ve bu çalışmada kullanılan CAN-BUS protokolü, yapısı, özellikleri, çalışma mantığı ve kullanım avantajları detaylı olarak açıklanmıştır.

Dördüncü bölümde, elektrikli aracın ön tekerlek içi motorları için tasarlanan elektronik diferansiyel sistem anlatılmıştır. Araç hızı ve direksiyon açısına göre ön tekerlek içi motorların hız değerleri Ackermann geometrisinden alınan denklemlere göre hesaplanmıştır. Bu denklemler MATLAB/Simulink'te modellenerek tasarlanacak sistemin sonuçları elde edilmiştir. Oluşturulan bu sistemin yazılımı CoDeSys V2.3 programı üzerinden yapılmıştır. Elektrikli aracın ön tekerlekleri için iki adet asenkron motor ve sürücüleri, bir işlemci, haberleşmeyi sağlayan CAN-BUS protokolü, hız bilgisi için ultrasonik sensör ve direksiyonun açısı bilgisini almak için ise enkoder kullanılmıştır. CAN-BUS protokolü üzerinden direksiyonun açısı ve gaz pedalından alınan hız bilgisi ile ön tekerleklerinin dönmesi gereken hız değerleri hesaplanmış ve bu hız bilgileri motorlara gönderilmiştir. Daha sonra direksiyon açısı 1°'den 15°'ye kadar birer derece aralıklarla değiştirilerek hız değerleri bir takometre yardımıyla deneysel olarak alınmıştır. CoDeSys sonuçları, Matlab/Simulink sonuçları ve deneysel sonuçlar karşılaştırılarak sonuçlar doğrulanmıştır. CoDeSys yazılımı, kullanılan cihazlar ve tasarlanan sistem bu bölümde kapsamlı şekilde açıklanmıştır.

Beşinci bölümde, değişik direksiyon açılarına ve aracın hız değerlerine göre hesaplanan ön tekerlek hızlarıyla ilgili alınan sonuçlar değerlendirilerek ileriki aşamalarda yapılabilecek çalışmalar üzerine öneriler verilmiştir.

2. ELEKTRİKLİ ARAÇLAR

2.1. Giriş

Son yıllarda ulaşım sektörünün giderek artması, çevre kirliliği ve fosil yakıtların zamanla tükenmesi gibi olumsuz sonuçları beraberinde getirmiştir. Bu kirliliği ortadan kaldırmak ve zararlı emisyonları azaltmak için elektrikli araçlar kullanılmaya başlanmıştır. Bunların yanı sıra, sürücü teknolojisinde ve batarya sistemlerindeki gelişmeler ile güvenli sürüş gibi pek çok etken de elektrikli otomobillere olan ilgiyi arttırmıştır [39, 40]. Elektrikli araçlar çok uzun yıllardan beri gündemde olup ancak şarj edilebilmesi için gerekli alt yapının olmaması, bu nedenle uzun mesafelerde kullanılamaması ve hızlarının çok düşük olması gibi olumsuzluklarından ötürü yaygın olarak kullanılamıyordu [41]. Daha sonra 1990'lı yıllarda çalışmaların artması ve 2000'li yıllarda hibrit elektrikli araçların geliştirilmesiyle elektrikli araçların kullanımı artmaya başlamıştır [42]. Elektrikli araçlar hibrit, bataryalı ve yakıt hücresi elektrikli araçlar olmak üzere üçe ayrılırlar. Hibrit araçlar, hem elektrik motoru hem de içten yanmalı motordan oluşan araçlardır. Bataryalı elektrikli araçlar ise, enerjisini akü bankından sağlayan ve şarj etme işlemi dışarıdan yapılan en verimli elektrikli araç türüdür [41].

Elektrikli araçların içten yanmalı motorlu araçlara göre üstün özellikleri şunlardır [42]:

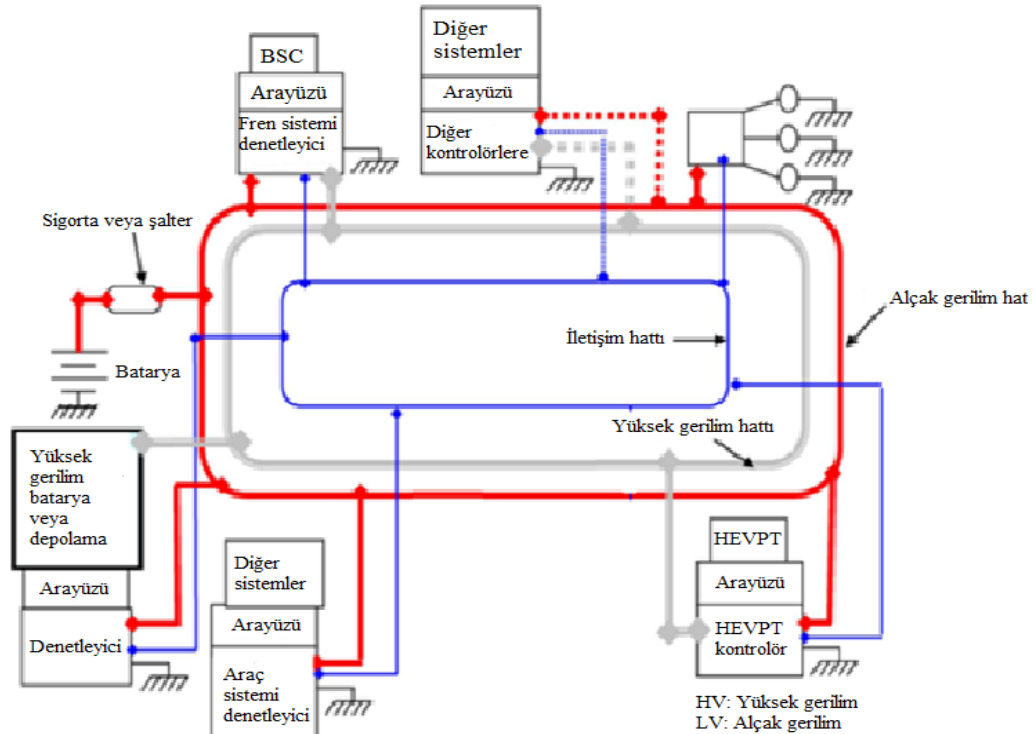
- Çok daha sessiz çalışırlar.
- Zararlı emisyonlar üretmediklerinden çevreye zarar vermezler.
- Motor ve vitesle ilgili ek donanımlar gerektirmediklerinden daha az elemana sahiptirler.
- Verimleri oldukça yüksektir.
- Zararlı gazlar yaymadıkları için bakım masrafları azdır.
- Güneş panelleri aracın çeşitli noktalarına yerleştirilerek şarj edilebilirler.
- Yüksek moment sağladıklarından araç daha kısa zamanda hızlanır.
- Motoru daha ucuz ve uzun ömürlüdür.
- İçten yanmalı motorlarla birlikte Hybrid olarak çalışabilirler.

Bunun yanında elektrikli araçların içten yanmalı motorlu araçlara göre bazı olumsuz yönleri de vardır. Bunlar şöyle sıralanabilir:

- Bataryaların yüksek maliyetinden ötürü elektrikli otomobil fiyatlarının yüksek olması,
- Bataryaların ömürlerinin kısa ve ağırlığının fazla olması,
- Suya ve neme karşı etkili koruma ve yalıtım yapma zorunluluğunun olması,
- Bataryaların iklim değişikliklerinden etkilenmesi,
- Elektrikli araçlar şarj edildiğinde daha az mesafe gidilebilmesi, şarj istasyonlarının az ve şarj sürelerinin uzun olması nedeniyle uzun mesafelerde kullanımının sıkıntılı olmasıdır.

Elektrikli aracın genel yapısı Şekil 2.1'de gösterilmektedir [42].

Elektrikli araçlarda, aküler şarj edilerek elektrik enerjisini depolar ve elektrikli motor bu enerjiyi kullanarak aracı hareket ettirir. Diğer makineler gibi elektrikli motorların da sargı direnci, eddy, histeresiz ve sürtünmeden dolayı oluşan kayıplarından ötürü verimleri % 100 değildir. İyi bir soğutma sistemiyle motorun verimi artırılarak, daha sessiz çalışması sağlanabilir [43].



Şekil 2.1. Elektrikli aracın yapısı [42].

2.2. Elektrikli Araçların Tarihçesi

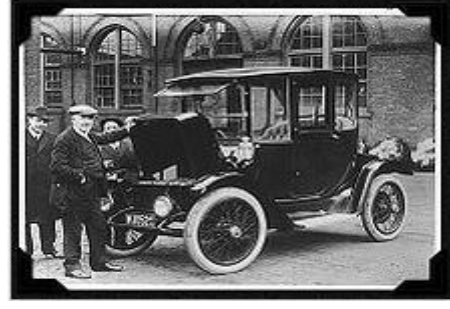
Türkiye'de ilk elektrikli araç, 1888 yılında II. Abdülhamit tarafından Messrs Immisch & Co adındaki şirkete sipariş edilmiştir. Bu şirkette çalışan Magnus Volk ve Moritz Immisch adındaki iki mühendis tarafından Abdülhamit Han için özel olarak tasarlatılmıştır. Bu otomobilin tasarımında ön kısımda tek bir büyük tekerlek yerine birbirine yakın iki küçük tekerlek kullanılmış ve Immisch tarafından patenti alınmış 20 A, 48 V, 1 beygir gücünde bir motora sahiptir. II. Abdülhamit bu araçtan çok memnun kalarak mühendisleri ödüllendirmiş ve bu olay o dönemin teknik dergilerinde de yer almıştır [44].

1880-1900'lü yıllarda elektrikli araçlar içten yanmalı motorlu araçlarla rekabet içerisindeydi. 29 Nisan 1899'da Camille Jenatton tarafından aracı Jamais Contente ile 100 km/saat'lik rekoru 106 km/saat ile kırarak elektrikli otomobillerin hız ve mesafe üstünlükleri de kanıtlanmıştır. Elektrikli araçların önemli bir sorunu olan şarj etme konusuna getirilen çözümlerden biri Hartford Electric Light Şirketi tarafından geliştirilmiş, ilk kez elektrikli kamyonlarda kullanılan değiştirilebilir batarya hizmetidir. Araç kullanıcıları araçlarını bataryasız şekilde General Electric Şirketinden, elektriği de Hartford Electric'ten değiştirilebilir bataryalar aracılığıyla satın alıyorlardı. Böylece mil başına şarj ücreti ve kamyon depolama ve bakım masraflarını karşılayan bir hizmet ücreti ödüyorlardı. Bu hizmetle, 1910 ile 1924 yılları arasında 6 milyon milden fazla bir ulaşım sağlanmıştır. 1917 yılında Chicago'da Milburn Light elektrik otomobilleri için de bataryasız araç satın alma hizmetleri verilmiştir [44]. 1897'de Fildelfiya Elektrikli Taşıma ve Vagon Şirketi tarafından yapılan New York şehri taksi filosu ilk ticari uygulamadır. 20. yüzyıl başlarında ABD'de Anthony Electric, Baker, Columbia, Anderson, Fritchie, Studebaker, Riker, Milburn ve daha pek çok kişi tarafından elektrikli araçlar üretilmiştir [44].

Elektrikli araçlar 1900'lü yılların başında içten yanmalı motorlu araçlara göre daha yavaş olmalarına rağmen, daha sessiz çalışması, ekonomik olması ve çevreyi kirletmemesi gibi olumlu yönlerinden ötürü tercih edilirlerdi. Ayrıca, elektrikli araçların vites değiştirme gibi bir problemleri de yoktur. Bunun yanı sıra, motoru çalıştırmak için elle kurulan bir kola ihtiyaç olmadığından kullanım kolaylığı sağlamaktadır [44].



Şekil 2.2. 1904 yılında bir Alman elektrikli arabası [44].



Şekil 2.3. 1913 yılında Thomas Edison ve bir Detroit elektrik arabası [44].

1970 ve 80'lerde petrol rezervlerinin azalmaya başlaması sebebiyle elektrikli otomobiller tekrar gündeme gelmiştir. 1990'larda, California Air Resources Board (CARB) çevreyi kirletmeyen, daha yakıt verimli araçlara yönelik bir çalışma başlatmıştır. Bunun üzerine, CryslerTEVan, Ford Ranger EV pickup truck, GM EV1 ve S10 EV pickup, Honda EV plus hatchback, Nissan lityum-iyon bataryalı Altra EV minivagon ve Toyota RAV4 EV gibi pek çok araç geliştirilmiştir [44].

2000'lerin sonlarında, fazla yakıt tüketimini azaltan daha küçük, hibrit ve elektrikli arabaların kullanımını arttırmaya yönelik çalışmalar hız kazanmıştır. 2004 yılında Kaliforniyalı otomobil üreticisi Tesla Motors tarafından Tesla Roadster geliştirilmiş ve 2008 yılında müşteriye sunulmuştur. 2010 yılı Nisan ayında Mitsubishi MiEV'in, Aralık ayında da Nissan Leaf'in Japonya ve Amerika olmak üzere çeşitli ülkelerde satışı başlamıştır. 2012 yılı Temmuz ayı itibarıyla, REVAi, Buddy, Citroen C1 EV, Transit Connect Electric, Mercedes-Benz Vito E-cell, Tazzari Zero, Smart ED, Wheego Whip LiFe, Mia Electric, BYD e6, Ford Focus Electric, BMW ActiveE, Coda, Renault Fluence Z.E., Tesla Model S., Honda Fit EV gibi pek çok elektrikli otomobil satışa sunulmaya başlanmıştır [44].



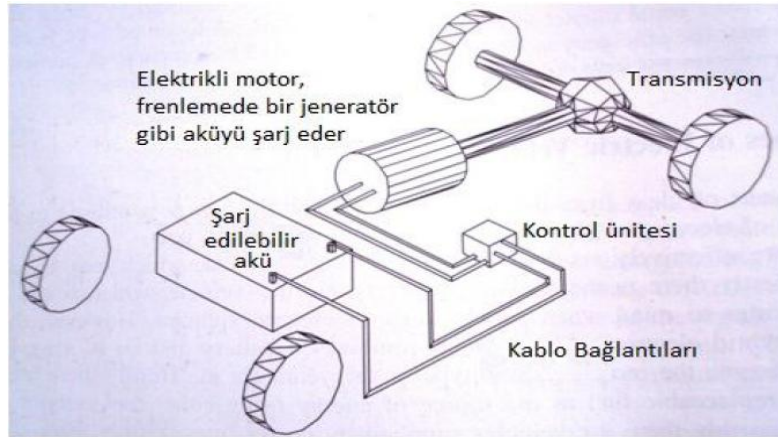
Şekil 2.4. 2010 yılında piyasaya çıkan Mitsubishi MiEV [44].



Şekil 2.5. 2008 yılında piyasaya çıkan Tesla Roadster [44].

2.3. Elektrikli Araçların Çalışma Prensibi

Elektrikli araçlarda içten yanmalı motorlar gibi vites kutusu, soğutma sistemi, debriyaj, egzoz vb. yapılar bulunmadığından çalışması çok daha basittir. Şekil 2.6'da görüldüğü gibi bir kontrol ünitesiyle, aküler motor için gerekli olan enerjiyi sağlarlar. Bu enerjiyi kullanan elektrik motoru ise direk diferansiyele bağlanarak tekerleklere dönme hareketini iletir ve tekerleklerin dönmesi bu şekilde sağlanır [43]. Bu çalışmada ise, motorlar tekerlek içlerine yerleştirilerek aktarım organlarının kullanımı ortadan kalkmıştır.



Şekil 2.6. Elektrikli aracın çalışma sistemi [45].

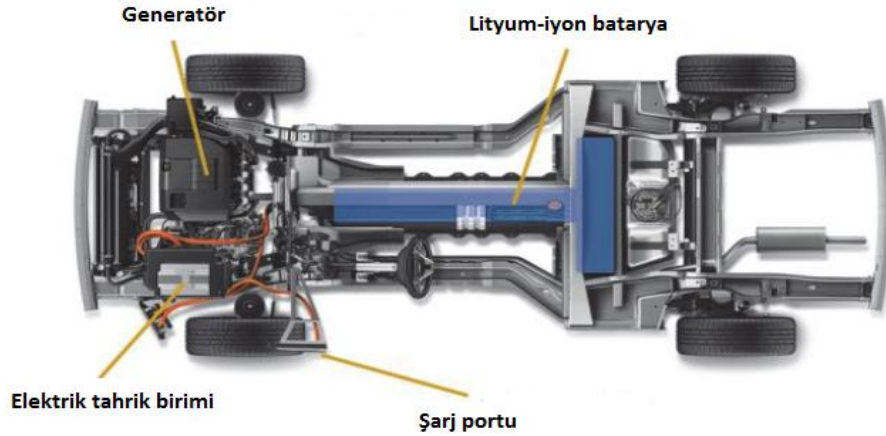
Elektrikli otomobillerin içten yanmalı motorlardan farklı olan kısımları piller, elektrik motoru, akü şarj ünitesi ve hız kontrol ünitesidir.

2.3.1. Piller

Kimyasal enerjiyi elektrik enerjisine dönüştürerek bünyesinde depolayan cihazlara pil ya da akü adı verilir. Bu dönüşüm tek yönlü ise pil primer yapıda şarj edilemez, her iki yönde olabiliyorsa yani elektrik enerjisi tekrar kimyasal enerjiye çevrilebiliyorsa sekonder yapıdadır. Primer yapı bir pil, şarj edilemezken sekonder yapıdaysa şarj edilebilir özelliğe sahiptir [46].

Bütün aküler yapısal olarak birbirine benzer ve bir grup elektrokimyasal hücreden oluşur. Her hücre de pozitif ve negatif elektrot ile bir ayıraçtan (seperatör) meydana gelir. Aynı kimyasal yapıya, ölçülere ve elektrik kapasitelerine sahip piller bir araya gelerek bataryayı oluşturur. Radyo, oyuncak, el feneri, cep telefonu gibi cihazlarda piller tek tek kullanılırken, kameralar, telsiz telefonlar, matkap, tornavida gibi cihazlarda ve

elektromobillerde batarya kullanılmaktadır [46]. Bataryaların enerji verimi, kullanım ömrü ve şarj hızı gibi özelliklerinin iyi olması istenir [47]. Elektrikli araçlarda kullanılan batarya teknolojileri, lityum-iyon (388 V, 360 Ah), kurşun asitli (12 V, 170 Ah), demir-lityum (30 kWh) ve sodyum sülfat bataryalarıdır [48]. Lityum-iyon batarya kullanılan bir Chevrolet Volt kasası Şekil 2.7'de görülmektedir:



Şekil 2.7. 2008 model Chevrolet Volt [48].

2.3.2. Elektrik Motoru

Elektrikli araçlarda kullanılan motorların hata toleranslı çalışması, yüksek verim ile düşük akustik gürültüye sahip olması ve esnek sürüş kontrolü gibi özelliklerinin olması istenmektedir [49]. Bu çalışmada etiket değerleri 2.2 kW, 4 kutuplu, 1420 dev/dakika olan iki adet asenkron motor kullanılmıştır.

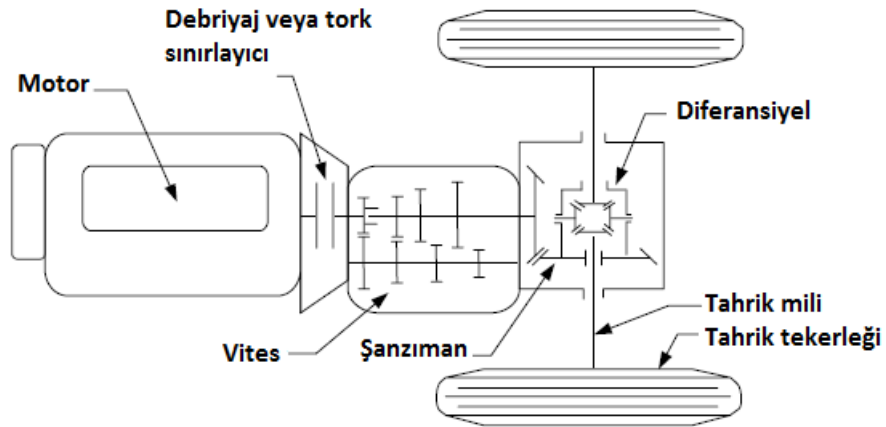
2.3.3. Akü Şarj Ünitesi

Şarj ünitesi şebeke elektriğini pillerin ihtiyacı olan DA gerilim seviyesine dönüştüren elemandır. Bu şekilde aracın pilleri şarj edilir. Eğer şebeke elektriğiyle şarj edilmek istenirse uzun bir sürede piller dolar. Bunun için hızlı şarj üniteleri geliştirilmiştir [46].

2.3.4. Hız Kontrol Ünitesi

Hız kontrol üniteleri motorun devrini ve yönünü ayarlarlar. İçten yanmalı motordaki gaz pedalının yerine elektrikli otomobillerde ayarlı bir direnç kullanılır. Bu direnç ile motor devri ayarlanır [46]. Bir otomobilde güç aktarma organları Şekil 2.8'de gösterilmiştir. Bunlar, içten yanmalı motor ya da elektrik motoru, debriyaj, diferansiyel, vites kutusu, şanzıman, tahrik mili ve tekerleğidir. Motorun moment ve hız bilgisi, debriyaj, vites

kutusu, şanzıman, diferansiyel ve tahrik mili ile tahrik edilen tekerleklere gönderilir. Debriyaj, manüel vites değiştirmede kullanılırken, otomatik viteste ise tork sınırlayıcı, manüel vitesteki debriyaj görevi gören hidrokinematik bir birimdir [50]. Elektrikli araçlarda vites kutusuna, tekerlek içi motorlu elektrikli araçlarda ise hem vites kutusuna hem de aktarma organlarına ihtiyaç yoktur. Bu çalışmada motorlar tekerlek içlerine yerleştirilerek aktarım organlarının kullanımı ortadan kalkmış, böylece aktarım organlarıyla kullanılan dişli sistemlerinin olmamasından ötürü bakım ve onarım masrafları da azalmıştır.



Şekil 2.8. Bir otomobilde güç aktarım organları [50].

2.4. Araçlarda Elektroniğin Gelişimi

Araçlarda elektroniğin kullanımı 1930'ların başında ilk araç radyolarının kullanımı ile başlamıştır. 1950'lerde düşük maliyetli transistörün gelişiminden sonra elektronik mühendisleri ciddi şekilde araçlarda elektroniği kullanmaya başlamışlardır [51]. Bunun nedeni yakıt tasarrufu, daha az hava kirliliği, daha güvenli ve konforlu sürüştür [52]. Otomobillerde elektroniğin kullanımı pek çok yenilik getirmiştir. Direksiyon hareketi, fren pedalı gibi mekanik işlemleri gerçekleştiren komutların elektrik sinyali şeklinde gönderilmesini sağlayan x-by-wire teknolojisi de bu yeniliklerden biridir. Bu teknoloji ile mekanik hareket komutu bir algılayıcı ile belirlenerek tel üzerinden elektriksel sinyal ile işlemi gerçekleştiren aktüatör sistemine gönderilir. Otomotiv teknolojisi beş kısımda incelenebilir [52]:

- Gövde, Konfor ve Kolaylık Sistemleri
- Güvenlik ve Korunma Sistemleri
- Elektrik-Elektronik Enerji Sistemleri

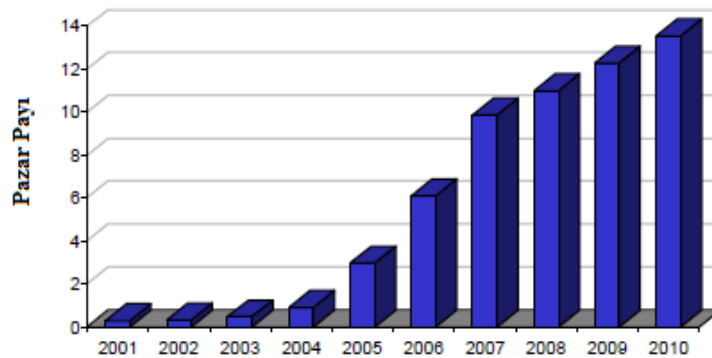
- Bilişim Sistemleri
- Güç ve Aktarım Sistemleri

2.4.1. Gvde, Konfor ve Kolaylık Sistemleri

Son zamanlarda otomobillerde konfor ve kolaylık sistemleri olduka n plana ıkmıřtır. Bu nedenle, arata pek ok algılayıcı ve elektromekanik hareket sistemleri kullanılmaktadır. Lks bir arata, marř motoru, alternatr, benzin pompası gibi aracın alıřması iin gerekli ana iřlemleri yerine getiren ve koltuk, kapı camı, n panel, anten, tepe gneřlięi gibi dięer iřlemler iin kullanılan yaklaşık 100 adet elektrik motoru ya da aktatr bulunur. Isıtma, soęutma ve havalandırma sistemlerinin tm elektronik olarak kontrol edilir.

2.4.2. Gvenlik ve Korunma Sistemleri

Gvenlik ve korunma sistemleri otomobil teknolojisinin en nemli konularından biridir. Bu alanda yapılan alıřmalar, hava yastıkları, lastik basın algılama sistemleri ile kaza uyarı ve nleme sistemleri zerinedir. Ayrıca gnmzde aralarda kullanılan ABS, Automotic Stability Program (ASP) ve Adaptive Cruise Control (ACC) de gvenli srř zerine geliřtirilen nemli uygulamalardandır. ABS sistemleri, fren sırasında tekerleęin kaymalarını nlemek ve ani frenlemelerde freni aıp kapatarak aracın kısa mesafelerde durmasını saęlamaktadır. ABS sisteminin yıllara gre pazar payındaki artıřları řekil 2.9'da gsterilmiřtir [52]:



řekil 2.9. ABS sisteminin pazar payının geniřlemesi [52].

ACC sistemi ile ndeki arala olan mesafenin ayarlanması ve arpıřmanın engellenmesi amalanır. Bu hedef iin araca bir elektronik sistem konularak elektromanyetik dalgalar aracılıęıyla ndeki arala ya da ona ait elektronik sistemle iletiřime geilir. Acil frenleme

ve çarpışma önleme sistemlerinde ise frenleme işlemi bir kontrolör tarafından yapılarak araç en yüksek ivme ile frenlenir [52].

2.4.3. Elektrik-Elektronik Enerji Sistemleri

Araçta konforun ve güvenlik sistemlerinin artışıyla enerji sarfiyatında da artışlar meydana gelmiştir. Direksiyon, fren, kapı, koltuk gibi parçaların hareketinde elektrik motorlarının kullanılması, ısıtma ve soğutma sistemleri, haberleşme donanımları, oto bilgisayarlar araçlarda güç ihtiyacını artırırlar [52].

2.4.4. Bilişim ve Eğlence

Otomobil teknolojisi gün geçtikçe ilerleyerek aracın iç ve dış sistemlerinin tek ağ üzerinden haberleşmesi sağlanmaktadır. Bunun için kullanılan veya kullanılması düşünülen veri yolları CAN, LIN, Media Oriented Systems Transport (MOST) Bus ve FlexRay'dir.

FlexRay, gelecekte kullanılması planlanan güvenli sürüş anlamında x-by-wire uygulamalarında tercih edilebilecek bir veri yoludur. BMW, DaimlerChrysler, Motorola, Philips, GM ve Bosch gibi şirketler bu veri yolunu geliştirmek için bir araya gelerek çalışmalar yapmaktadırlar [52].

2.4.5. Güç ve Aktarım Sistemleri

Bir araçta tahrik sistemi, içten yanmalı motor, bağlantı elemanları, diferansiyel, vites kutusu ve tekerleklerden oluşur. Tahrik sistemlerinde elektronikten oldukça yararlanılmaktadır. İçten yanmalı motorların kontrolü bir EKÜ tarafından yapılır. Yakıt-hava karışımı, motor hız ve gaz pedalı bilgisi alınarak elektronik olarak ayarlanır. Hedeflenen teknolojilerden biri de giriş ve çıkış subaplarının eksantrik mil ile değil, her subabın ayrı ayrı kontrol edilmesini sağlamak için elektromekanik hareket sisteminin kullanılmasıdır. Diğer bir yenilik, shift-by-wire denilen vites değiştirmenin mekanik olarak değil konum bilgisinin vites kolundan alınarak elektriksel sinyal ile yapılmasıdır. Güç ve aktarım sistemlerine göre araçların sınıflandırılması aşağıdaki gibidir [52]:

- Klasik içten yanmalı motor ile tahrik edilen araçlar (Internal Combustion Engine, ICE)
 - Motor türü benzin, dizel ve H motoru olanlar
 - Transmisyon türü: Manüel ve otomatik vites

- Tümü elektrikli araçlar
 - Akü beslemeli
 - Tekerlekleri diferansiyel üzerinden tahrik edilen ve tekerlekleri doğrudan tahrik edilen yakıt pili beslemeliler
- Hybrid elektrikli araçlar
 - Seri Hybrid
 - Paralel Hybrid
 - Seri-paralel Hybrid
 - Karışık Hybrid

Klasik içten yanmalı motorlarda mekanik enerji içten yanmalı motorlar tarafından üretilerek vites kutusu, transmisyon mili ve diferansiyel üzerinden tekerleklerle taşınır. Hidrojen yanmalı motorlar, yakıt olarak hidrojen kullanırlar ve hala araştırılmaktadırlar. Tümü elektrikli araçlarda tekerlek tahrik gücü elektrik makinaları tarafından sağlanırken, Hybrid elektrikli araçlarda ise, hem elektrik motoru hem de içten yanmalı motorla sağlanmaktadır [52].

2.5. Araçlarda Elektronik Sistemler

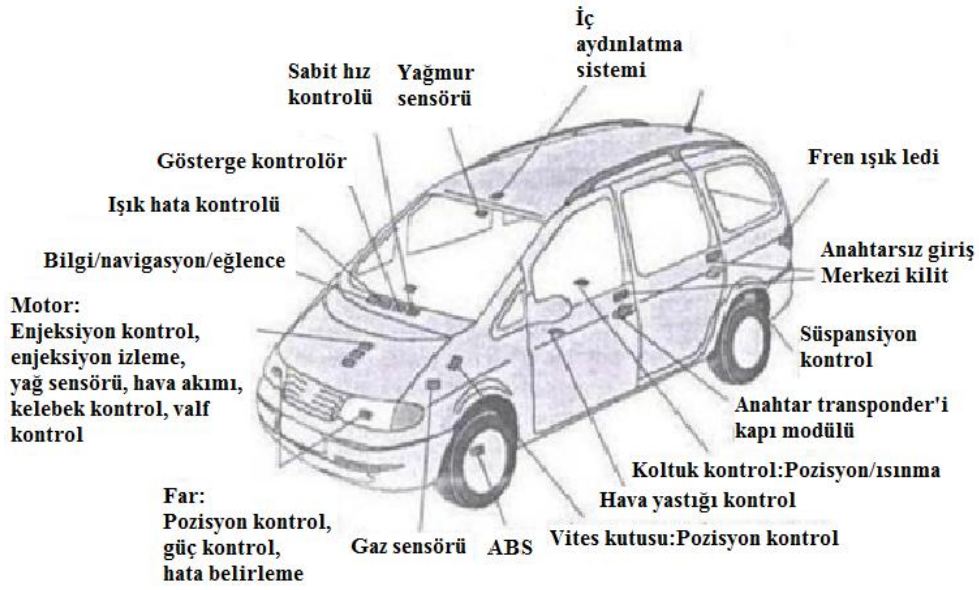
Daha fazla güvenlik ihtiyacı, konfor ve eğlence ihtiyacı, daha az yakıt kullanımı ve kirlilik gibi nedenlerle araçlarda elektronik sistemlerin kullanımı artmıştır. Elektronik sistemlerin çoğu sensörler ve kontrol birimlerinden oluşur. Bu elektronik sistemler, elektronik güç devreleri, motor sistemleri ve mekanik hareket sistemleri gibi araçlarda bulunan bütün bu donanımın birbiriyle ve ana işlemci ile haberleşmesi istenir [15].

2.6. Sensör Teknolojisi

İleri teknoloji araçlarda çok sayıda sensör yer almaktadır. Bu sensörlerin CAN hattına bağlanarak haberleşmesi sağlanır [52]. Şekil 2.10'da araçlarda kullanılan bazı sensörler görülmektedir [15]. Bunlar:

- Lastik basınç sensörü,
- Sıcaklık belirleme sensörü,
- Nem sensörü,
- Yük/ağırlık sensörü,

- Yağ basıncı sensörü,
- Tekerlek hızı sensörü,
- Yakıt düzeyi sensörü,
- Mesafe sensörü,
- İvmelenme sensörü,
- Yakıt deposu basınç sensörü,
- Soğutma suyu seviye ve sıcaklık sensörüdür [52].



Şekil 2.10. Araç üzerindeki sensör ve kontrol birimleri [15].

3. ARAÇLARDA VE ENDÜSTRİDE AĞ SİSTEMLERİ

3.1. Giriş

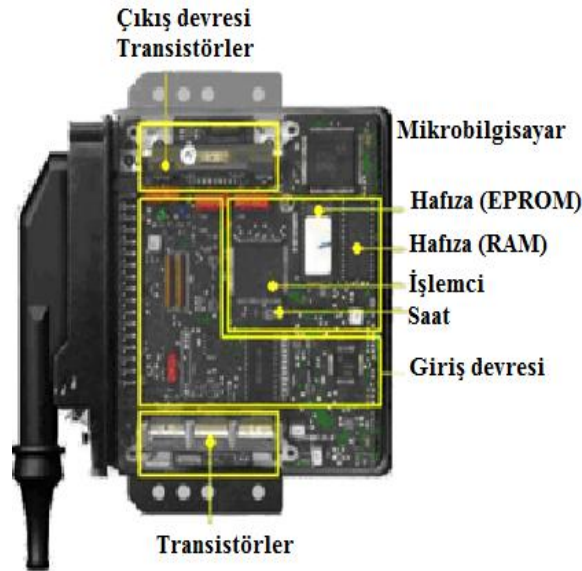
Elektronik teknolojinin son yıllarda hızla gelişmesi, otomotiv sektörünü önemli ölçüde etkilemiş ve akıllı araç teknolojisinin dünya genelinde hızla ilerlemesini sağlamıştır [53, 54]. En büyük gelişmelerden biri yüksek performanslı motorlar, yakıt tüketiminin azalması, daha az zararlı atık miktarı, üstün güvenlik gibi olumlu etkilere yol açan motor kontrol ünitesidir. Yeni araçlardaki elektronik motor yönetim sistemleri verimi önemli oranda arttırmaktadır. Elektronik kontrol ünitesi sayesinde yakıt tasarrufu sağlanarak frenler, konfor ve güvenlik sistemi ile de veri alışverişi yapılabilmektedir [53].

3.2. Motor Yönetim Sistemleri

Motor yönetim sistemleri daha az yakıt tüketimi ve yüksek motor gücü sağlar. Motor yönetim sistemlerinin en önemli bileşeni EKÜ olup gelen sensör sinyalleri burada işlenerek alınan kumanda komutları alt sistemlere gönderilir. İç yapısı Şekil 3.1'de görülmektedir. EKÜ, sıcaklık, basınç, hız, devir sayısı gibi pek çok sensörden gelen bilgileri kontrol eder. Sensörlerden gelen analog bilgiler sayısala dönüştürülerek EKÜ'ye iletilir. Bu değerler EKÜ tarafından algılanır, silinip programlanabilir bellek olan Erasable Programmable Read Only Memmory (EPROM) içerisindeki haritadan ilgili değerleri bularak ateşleme zamanlaması, turbo basıncı, optimum yakıt miktarı, emisyon değerleri gibi değişkenleri hesaplar. Bu bilgileri sayısaldan analoga çevirerek ilgili yerlere sinyal olarak gönderir. Tüm işlemler saniyenin binde biri kadar kısa sürelerde gerçekleşir. Sensörler görev yapmazsa, EKÜ yedek bilgileri kullanır ve herhangi bir arızada ikaz numarasıyla sürücüyü uyarır [53]. Motorlarda EKÜ kullanılmasının avantajları şunlardır [53]:

- Daha az yakıt tüketimi,
- İlk çalıştırma ve ısınma kolaylığı,
- Kolay arıza tespiti,
- Egzoz gazları emisyonunun düşürülmesi,
- Rakımdan etkilenmemesi,
- Ayar gerektirmemesi,

- Vuruntu kontrolü,
- Daha iyi karışım ayarı,
- Motorun özgül gücünün artırılması,
- Rölanti kontrolü ve adaptasyonu olup tek dezavantajı pahalı sistemler olmalarıdır [53].



Şekil 3.1. EKÜ'nün iç yapısı [53].

3.3. Elektronik Kontrol Ünitelerinde Bilgi İşleme

Otomotiv alanında elektronik kontrol üniteleri olarak genellikle mikrodenetleyiciler tercih edilmektedir [55]. Ancak bu projede farklı olarak HY TTC-60 modülü kullanılmıştır.

3.4. Veri İletim Yöntemleri

Dijital bilgilerin bir noktadan başka bir noktaya aktarılmasına veri iletimi adı verilir. Veri iletim sistemleri, bilgisayarlar, bilgisayarlar ile terminaller veya bilgisayarlar ile alıcılar arasında veri iletimini gerçekleştirirler. Ses, görüntü gibi bazı analog bilgiler de dijitalle dönüştürülerek iletebilir. Dijital sinyaller paralel ve seri olmak üzere iki şekilde iletilirler [55].

Veri iletişiminde önemli olan etkenler şunlardır [56]:

- Doğru Hedef (Delivery): Verinin doğru hedefe ulaşmasıdır.

- Doğruluk (Accuracy): Verinin kaynağından çıktığı şekliyle iletimidir.
- Zaman (Timeliness): Gerçek zamanlı iletişimde büyük ölçüde önemli olan, verinin zamanında hedefine ulaşmasıdır.
- Gecikme Değişimi (Jitter): Paketlerin hedefe ulaşma süresindeki değişimdir.

Veri iletişim sistemi beş elemandan oluşmaktadır [56]:

- Mesaj (Message): Ses, görüntü, sayı, metin, resim gibi çeşitli şekillerde iletilen bilgidir.
- Gönderici (Sender): Pc, workstation, video kamera gibi veri gönderen cihazlardır.
- Alıcı (Receiver): Pc, workstation, televizyon gibi veriyi alan cihazlardır.
- İletim Ortamı (Transmission Medium): Fiberoptik kablo, radyo dalgaları gibi verinin gönderen ve alan cihaz arasında iletilmesini sağlayan fiziksel yoldur.
- Protokol (Protocol): Veri iletişimini başlatan, yöneten ve sonlandıran elemandır.

3.5. Haberleşme Protokolleri

Protokol, bir sistemdeki tüm elemanların birbirleriyle haberleştiği ortak konuşma dilidir. Alıcı-verici arasında hızlı, güvenli ve anlaşılır bir veri iletişimi gerçekleştirilebilmesi için bazı etkenler kullanılan yazılım ve donanıma göre belirlenmelidir. Alıcılar ve vericiler belirli bir protokol üzerinden haberleşecek şekilde üretilmektedir [56].

Dünyadaki otomasyon sistemlerinde yaklaşık % 80'lik bir pazar payına sahip olan "Alansal Veri Yolu (Fieldbus)", fabrika içerisinde ve dışında kullanılan tüm protokollerin genel adıdır. Bu protokol, fiziksel katman, haberleşme çatısı ve kullanıcı katmandan meydana gelmektedir [56]. Açık ve kapalı sistem haberleşme protokolleri olmak üzere iki çeşit haberleşme protokolü vardır. Bunlar Tablo 3.1'de gösterilmiştir:

Tablo 3.1. Açık ve kapalı haberleşme sistemleri [57].

PROTOKOL	TANIMLAMA	ÜRETİCİ FİRMA
Açık Mimari	Interbus-S Profibus Canopen Devicenet As-ara yüzü FIP CAN LON	Phoenix Pno/Siemens CIA Allen-Bradley ASI Verein/Siemens AEG-Schneider Automation Intel/Motorolla/Phillips Echelon/Motorolla
Özgün (Spesifik) Mimari	Beckhoff I/O Melsec Suconet K ET 100 CS 31 Remote I/O Modnet	Beckhoff Mitshubishi Klockner Moeller Siemens ABB Alen Bradley AEG

3.5.1. Açık Sistem Haberleşme Protokolleri

Marka önemi olmadan, aynı amaçla kullanılan tüm cihazların birbirleriyle haberleşebilmesidir. Bunun için, Omron firmasının Profibus ürünleri ile Phonix firmasının Profibus ürünleri birbirleriyle rahatça haberleşebilirler. Açık sistem haberleşme protokolleri şunlardır [56]:

- **Interbus-S Protokolü:** Bu bus sistemi, Phoenix Contact tarafından geliştirilmiş ve Almanya'da oldukça yaygın olan açık mimarili ve DIN normlarına göre standartlaştırılmıştır. Veri iletişimi çift yönlüdür ve sistem kapalı halka topolojisi ile haberleşir [56]. Ağda hata tespiti sınırlı ölçüde yapılabilir.
- **Profibus Protokolü:** Açılımı "Process Field Bus" olan Profibus, geniş kapsamlı üretim otomasyonu için özellikle otomobil teknolojisinde kullanılmak üzere tasarlanmıştır. İlk defa German Department of Education and Research tarafından 1989 yılında geliştirilmiş ve daha sonra Siemens tarafından kullanılmıştır. EN 50170, EN 50254 ve IEC 61158 standartları üzerine kurulmuştur. Farklı üreticilerin cihazları arasında özel bir ara birime gerek olmadan haberleşme sağlayarak yüksek hız gerektiren uygulamalarda veya karmaşık haberleşme sistemlerinde yaygın olarak kullanılırlar [56]. Herhangi bir arıza anında diğer cihazlar etkilenmeden hatalı cihazlar hattan çıkartılabilir ve sistem kontrol edilebilir [57].

- **Modbus Protokolü:** 1979 yılında Modicon tarafından geliştirilmiş master/slave ilişkisine sahip bir seri iletişim protokolüdür. Bir master cihaz üzerinden haberleşme başlar ve maksimum 247 slave aynı ağ üzerinden haberleşebilir [56]. En yaygın kullanılan çeşitleri Modbus RTU ve TCP/IP'dir.
- **CAN-BUS Protokolü:** Genellikle otomotiv ve medikal alanda kullanılan Bosch tarafından geliştirilmiş seri bir haberleşme protokolüdür [19]. Multimaster çalışmaya imkan verir. Yani tüm CAN noktaları data iletebilir ve bazıları aynı anda istekte bulunabilirler. Mesaj önceliğine göre işlem yapar [56]. Araçlarda kullanılan kablo sayısını azaltarak aracın ağırlığını hafifleten, hızlı, hata tespiti kolay olduğundan güvenilir bir protokoldür. Bu bölümde CAN-BUS protokolü detaylı anlatılacaktır.
- **DeviceNet Protokolü:** Amerikan şirketi olan Allen-Bradley tarafından geliştirilmiş, CAN'ın uygulama katmanında yer alan bir protokoldür. Akıllı sensör ve aktüatörler için tasarlanmıştır. Bununla limit anahtar, fotoelektrik sensör, barkod okuyucu ve motor starterleri gibi cihazlara bağlanarak PC veya PLC gibi aletlerle haberleşmesi sağlanabilir [56]. Ayrıca düşük maliyetli ve sağlam bir haberleşme sistemidir.
- **AS-i Ara yüzü (Aktüatör Sensör):** AS-i ara yüzü, 11 farklı firma tarafından oluşturulmuş bir araştırma grubu tarafından geliştirilmiştir. Basit, güvenilir ve ucuz bir sistemdir. 1 master ve maksimum 31 slave'den oluşur. Özel veri dönüştürücüleri ile Profibus DP sinyalleri AS-i formatına dönüştürülebilir ve algılayıcılardan gelen sinyaller işlenerek aktüatörleri kumanda eden valfler kontrol edilebilir [56].

Açık sistem haberleşme protokollerinin karşılaştırması detaylı olarak Tablo 3.2'de verilmiştir:

Tablo 3.2. Açık sistem haberleşme protokollerinin karşılaştırması [57].

	AS-ARA YÜZÜ	PROFIBUS-DP	INTERBUS-S	CAN
Üretici Firma	Siemens	Siemens	Phoenix Contact	Intel/Phillips Motorolla
Kullanıcı Organizasyonu	AS- International Association	Profibus User Organization	Interbus-S Club	CiA (CAN in Automation)
Standart	-	EN 50170	DIN 19258	ISO/DIS 11519- 1 ISO/DIS 11898
Sertifikasyon	Var	Var	Var	Var
Mimari Tipi (Açık/Özgün)	Açık	Açık	Açık	Açık
Topoloji	Ağaç	Lineer	Halka	Lineer
Erişim Kontrolü	Master/Slave (Tek Master)	Master/Slave (İşaret Geçirme)	DSR	Master/Slave (Çoklu master)
Veri Formatı	4 bit data 4 bit parametre	1-16 byte data	Max. 512 byte	Max 8 byte data
Slave Sayısı	32	32	256	32
Slave Sayısı (Yükselticili)	32	122	256	2032
Veri Aktarım Hızı	167 kbit/s	12000 kbit/s	500 kbit/s	1000 kbit/s
Kablo Uzunluğu	100 m	100 m	12800 m	40 m
Voltaj Seviyesi	4 V data, 24 V/2 A güç	RS485	RS485	Diferansiyel

3.5.2. Kapalı Sistem Haberleşme Protokolleri

Mitsubishi, Melsec, Omron Phoenix Contact Interbus-S Compobus gibi her üreticinin sadece kendi ürünleri arasında haberleşmesini sağlayan protokollerdir [56].

3.5.3. Diğer Veri Yolu Sistemleri

Otomotiv sektörü son yıllarda hızlı geliştiğinden otomotiv endüstrisi, haberleşme, medikal cihazlar ve endüstriyel otomasyon gibi pek çok uygulamada, "Alanda Programlanabilir Kapı Dizileri" anlamına gelen "Field Programmable Gate Array (FPGA)" kullanılarak tasarlanan dijital modüller kullanılmıştır. SPI, I²C, Universal Asynchronous Receiver/Transmitter (UART), CAN, USB gibi pek çok seri haberleşme protokolleri vardır [58]. CAN protokolü dışında en yaygın kullanılan protokol UART protokolüdür. Bunlardan en çok kullanılanları RS232 ve RS485'tir. UART protokolü, noktadan noktaya haberleşme yapar ve düşük maliyetli olduğu için yaygın olarak kullanılırlar [11]. Kart ile

bilgisayar arasında standart seri kanal ara yüzü oluşturur [59]. Bu protokolün olumsuz özellikleri ise; byte tabanlı haberleşme olduğundan aynı anda birden çok byte gönderilemez. Ayrıca hata kontrolü bulunmadığından bunun için ayrı yazılım kullanılır. Bu da sistemi yavaşlatır [11].

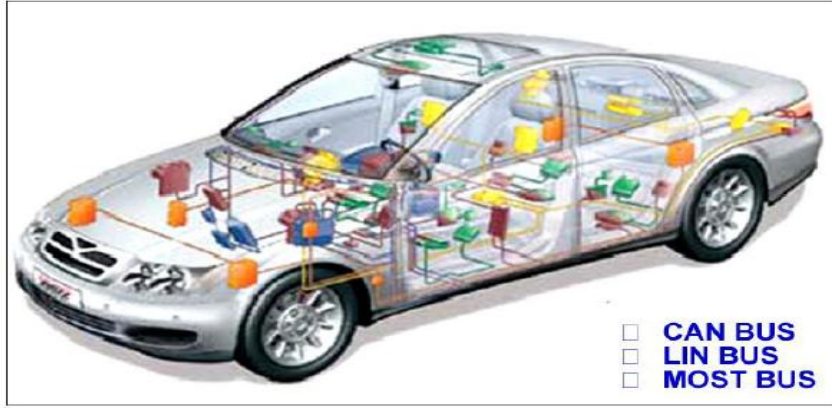
1970'lerin sonlarında otomotiv endüstrisi bazı sorunlarla karşılaşmıştır. UART protokolü, noktadan noktaya haberleşme yaptığı için kablo kullanımının artması, dolayısıyla aracın ağırlığının ve maliyetin artmasına neden olmuştur. Bunun üzerine bu olumsuzlukları ortadan kaldıran CAN protokolü geliştirilmiştir [11]. CAN protokolü, çoklu master özelliğiyle otomotiv ve endüstriyel uygulamalarda elektronik kontrol modülleri, sensörler ve aktuatörleri bağlamada özellikle tercih edilirler [58]. Seri iletişim protokolleri karşılaştırmalı olarak Tablo 3.3'te gösterilmiştir [60].

Tablo 3.3. Seri haberleşme protokollerinin pozitif ve negatif yönleri [60].

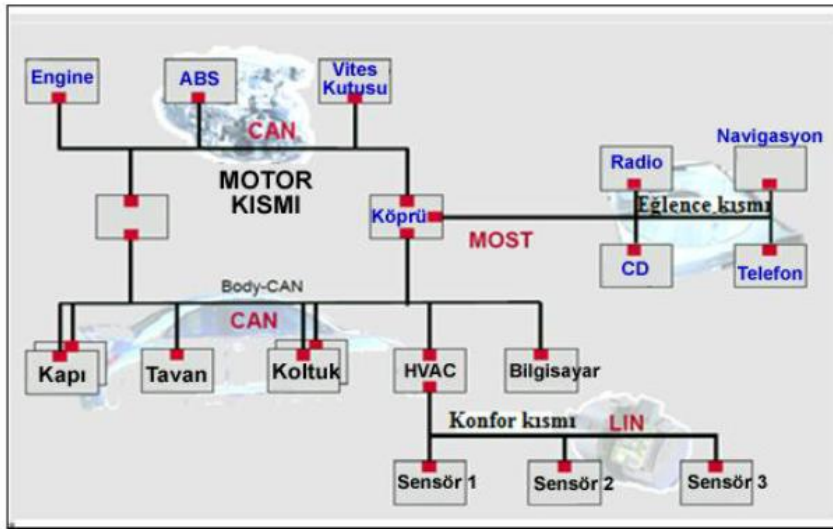
BUS	Pozitif yönleri	Negatif yönleri
UART	İyi bilinirler. Maliyet bakımından iyi ve basittir.	Sınırlı işlevsellik, Noktadan noktaya haberleşme yapar.
CAN	Güvenli ve hızlıdır.	Kompleks, otomotiv amaçlı kullanılır.
USB	Hızlı, basit ve düşük maliyetlidir.	Güçlü master ve özel sürücüler gerektirir.
SPI	Hızlı, evrensel olarak kabul edilmiş ve düşük maliyetlidir.	Belirlenmiş bir standardı yoktur.
I²C	Basit, iyi bilinir, evrensel olarak kabul edilmiş, maliyet bakımından iyidir.	Sınırlı hıza sahiptir.

3.6. Elektronik Kontrol Üniteleri Arasında Haberleşme

Otomobillerde elektronik kontrol birimlerinin birbirleriyle uyumlu çalışmaları için her sensörden alınan bilgi tek ağ üzerinden iletilerek haberleşme gerçekleşir. Şekil 3.2'de görüldüğü gibi otomobillerde CAN, LIN ve MOST BUS olmak üzere üç çeşit haberleşme sistemi kullanılır. Şekil 3.3'te ise aracın motor kısmında kullanılan CAN, eğlence kısmında kullanılan MOST ve konfor kısmında kullanılan LIN haberleşme sisteminin kullanım yerleri görülmektedir [55].



Şekil 3.2. Araç haberleşme ağ sistemleri [55].



Şekil 3.3. Haberleşme sistemlerinin kullanım yerleri [55].

3.6.1. CAN-BUS'ın Yapısal Özellikleri

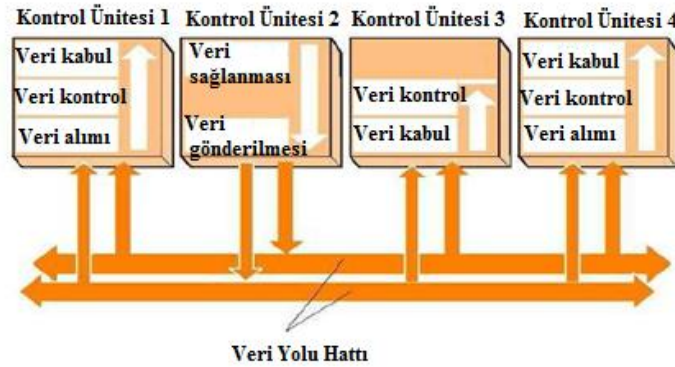
Günümüzde otomobillerde konforun artması ile elektronik parçaların sayısında artış görülmektedir. 1900'lü yıllarda araçlarda kontrol ünitelerinin sayısı 15 civarındayken 2000'li yıllarda 75'e kadar ulaşmıştır. Elektronik kontrol sistemlerinin artması haliyle daha uzun kabloların kullanılmasına ve ağırlığın artmasına neden olarak maliyeti arttırmış ve arıza tespitini de zorlaştırmıştır. Bu problemlere çözüm üretmek için 1987 yılında Bosch, Intel ile birlikte "Kontrol Alan Ağı" anlamına gelen CAN-BUS protokolünü geliştirmiştir. CAN 2.0A, düşük hızlı haberleşme olup ISO 11519 standardında ve CAN 2.0B haberleşmesi ise yüksek hızlı olup ISO 11898 standardında belirtilmiştir [55]. 1990'larda CiA kurulmuştur. Bu grup CAN özelliklerini belirleyerek gerçekleştirmiştir [11]. Böylece araçta bulunan ABS, ESP, enjeksiyon ve ateşleme sistemi gibi 20 ile 40 arasında araca

göre deęişkenlik gösteren elektronik kontrol sistemlerinin tek bir kablo üzerinden haberleşmesi sağlanmaktadır [55]. CAN protokolü bir çift kablo ile dięer veri hatlarından oluşur. Sistemde sadece bileşenler CAN-BUS üzerinden kontrol ünitelerine bağlanır. CAN-BUS, sensörler, üretim ve dokuma makineleri, asansörler, medikal cihazlar, makine kontrol birimleri, bina otomasyonu, elektrikli sandalyeler gibi pek çok uygulama alanlarında maksimum 1 Mbit/sn'lik bir hızda veri iletişimi yapmaktadır [55, 6, 13].

3.6.1.1. CAN-BUS Bilgisi

Şekil 3.4'te CAN hattından veri transferinin nasıl gerçekleştirildięi görölmektedir. Verinin sağlanması, gönderilmesi, alınması, kontrol edilmesi ve kabul edilmesi aşamalarından oluşur. CAN hattından elektronik kontrol ünitelerinin gönderdięi bilgiler bus bilgisi olarak adlandırılır. Her bus kullanıcısı, bilgiyi gönderme ve alma ünitesine sahiptir. Veriler gönderilirken tüm üniteler bilgiyi alır, sadece bir bus kullanıcısı bilgi aktarımı yapar.

Seri ve paralel bus bilgisi olmak üzere iki çeşit bilgi aktarımı vardır. Seri bus bilgi aktarımında, veriler ayrı ayrı iki kablo veya bir kablo ve araç şasesi üzerinden, önceden yükseklięi ve uzunluęu belirlenmiş gerilim sinyalleri olarak gönderilir. Paralel bus bilgi aktarımında ise, iki veya daha fazla kablo üzerinden devamlı bilgi alışverişi gerçekleşir [55].



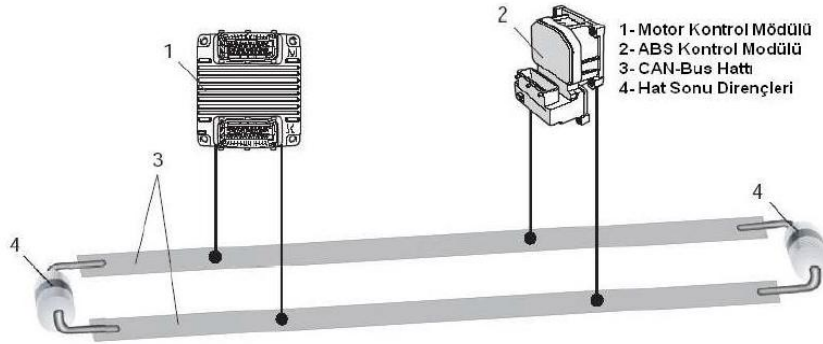
Şekil 3.4. CAN hattında veri iletimi [55].

CAN-BUS sistemi, kontrol üniteleri, alıcı/verici üniteler ve data-bus kablolarından oluşur:

- **CAN-BUS Kontrol Ünitesi:** Kumanda cihazı içindeki merkezi işlem biriminden gönderilmesi gerekli olan verileri alır. Bu verileri düzenler ve CAN alıcı/verici ünitesine gönderir. Bu veriler kontrol ünitesi içinde yaklaşık 200 kHz gibi yüksek frekanslı

gerilime dönüştürülür ve yaklaşık 5 V gibi küçük bir doğru akıma modüle edilir. CAN kontrol ünitesi, CAN alıcı/vericisinden de verileri alır, düzenler ve kumanda cihazındaki merkezi işlem birimine iletir [55].

- **CAN-BUS Alıcı/Verici Ünitesi:** CAN-BUS alıcı/verici ünitesi, bir verici (transmitter) ve bir alıcıdan (receiver) oluşur. CAN kontrol ünitesinden gelen veriler mikroişlemcinin anlayabileceği sinyallere dönüştürülür ve Databus kablolarına gönderilir. Databus kablolarından gelen veriler de CAN kontrol ünitesi için dönüştürülür [55].
- **CAN-BUS Databus Kabloları:** CAN-High ve CAN-Low olarak adlandırılan kablolar, elektromanyetik etkilere karşı dirençli olmasını sağlamak için saç örgüsü şeklinde örülü çift yönlü iletim kablolarıdır. Her iki kablodaki elektrik gerilimleri zıt yönlüdür. Bir kablodaki elektrik gerilimi yaklaşık 0 V ise diğer kablodaki gerilim 5 V ya da tam tersi olur. Bu kabloların kullanılma sebepleri şunlardır [56]:
- Kablo ağırlığı azaltılmış, arıza tespiti kolayca yapılarak güvenilirlik artırılmıştır.
- Veri kablolarından birinin kısa devre yaptığı ve kullanılmadığı durumda veriler diğer kablo üzerinden gönderilebilirler.
- Parazite karşı yüksek direnç sağlayabilirler.



Şekil 3.5. Kontrol üniteli CAN-BUS ve hat sonu dirençleri [15].

Şekil 3.5'te CAN-BUS hat sonu dirençleri ile kontrol modülleri görülmektedir. CAN kontrolü ve CAN alıcı-vericisi kontrol ünitesinin içindedir. CAN kontrolörden gönderilecek veriler mikroişlemciden CAN-BUS üzerinden gönderilebilir. CAN alıcı-vericiyle bilgiler uyarlanır ve iletilir. CAN kontrolörü bilgileri CAN alıcı-vericisinden alır ve bu bilgileri mikroişlemciye göndermeden önce uyarlar. CAN alıcı-vericisi CAN

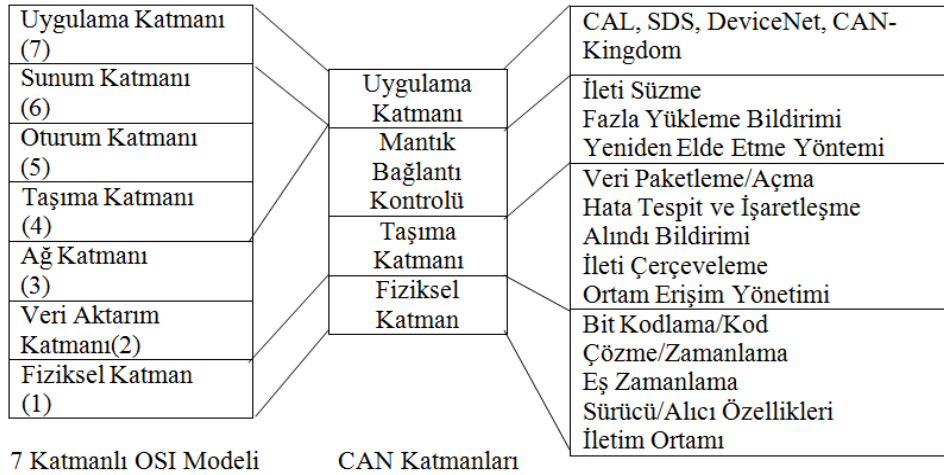
kontrolöründen gelen elektrik sinyallerini sayısala dönüştürerek CAN-BUS'a iletir. Sistemin çalışması Şekil 3.6'da gösterilmiştir [15].



Şekil 3.6. CAN-BUS sisteminin çalışma sırası [15].

CAN-BUS kablolarının kontrolünde iki yöntem kullanılır. İlk yöntemde, CAN-H ve CAN-L arasında direnç ölçümünde 60 Ω 'luk direnç kullanılır. Ölçümlerde sisteme gerilim verilmemelidir. Bu direncin kullanılmasının amacı, bilgilerin geri dönüşünü engellemektir. Böylece bus bilgileri karışmaz. İkinci olarak gerilim ölçümü yönteminde ise, CAN-H ve CAN-L ile şasi arasında ölçümler yapılır. Bu gerilimin değeri yaklaşık 2.5 V olmalıdır [55].

CAN-BUS, Şekil 3.7'de görüldüğü gibi ISO/OSI referansı üzerinden açıklanırsa, en üst katmandan görüntü ya da yazı şeklinde gelen veri, alt katmanlara doğru makine diline dönüşerek en son 1 ve 0'lardan oluşan elektrik sinyallerine dönüşür [56].



Şekil 3.7. CAN-BUS Katmanları [9].

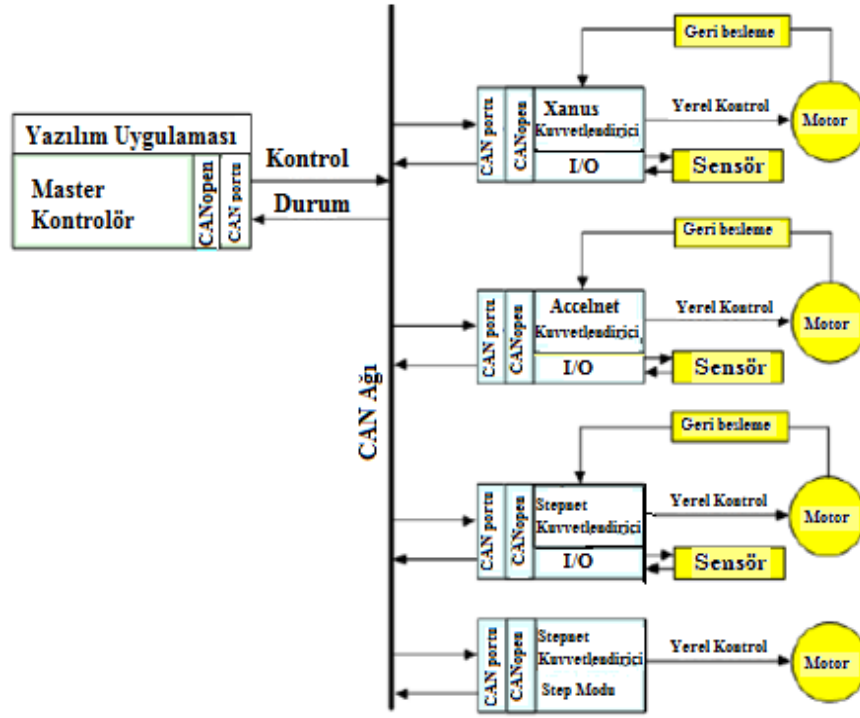
Nesne Katmanı (Object Layer), İletim Katmanı (Transfer Layer), Fiziksel Katman (Physical Layer) olmak üzere üç katmana ayrılır:

- **Nesne Katmanı:** Hangi mesajın transfer edileceğini tespit etme, iletim katmanında hangi mesajın alınacağına karar verme, donanımla ilgili uygulamaya ara yüz sağlama gibi işlemlerin gerçekleştirildiği katmandır [15].

- **İletim Katmanı:** Transfer protokolüdür. Yeni bir mesajı yollamadan önce iletim hattının boş olmasına bakar. İletim hattından veri alınmasından ve hata kontrolünden de sorumludur [15].
- **Fiziksel Katman:** CAN-BUS üzerinden haberleşen tüm sistem bileşenlerine ünite (node) denir. Bunları CAN-BUS sistemi ile haberleştirirsek her bir mikrodenetleyicili sistem bir üniteyi oluşturur. Bu üniteler haberleşirken tüm elektriksel kısım fiziksel katmanı oluşturur.

Uygulama katmanı, OSI referans modelinde uygulama katmanıya eşdeğer sayılan katmandır. Bu katmanda, temel ihtiyaçlar dışında kullanım kolaylığına yönelik CAL, CANOpen, PCAL, DeviceNet, SDS ve CAN Kingdom gibi uygulamalar geliştirilmiştir [9]:

- **CAN Application Layer (CAL) ve CANOpen:** CANOpen, Philips Medical Systems tarafından geliştirilmiş mevcut uygulamalarda kullanılan bir protokoldür. CAN'ın uygulama katmanında tanımlı olup, özellikleri CiA'da DS-301 numaralı döküman ile tanımlanmıştır [9]. CAN-Open, motor sürücüler gibi değişik tipteki endüstriyel cihazların haberleşmesinde kullanılır. Şekil 3.8'de görüldüğü gibi CAN hattına bağlı her cihaz master denetleyiciye ve diğer modüllere veri gönderebilir. Master denetleyici iletişim ağı boyunca cihazlara komutları gönderince durum bilgilerini de alır. CAN-Open CAN-BUS üzerinde seri iletişim yapısına sahip ve 1 Mbps hızda 25 m'ye kadar veri iletimi yapabilir. Çoklu master özelliğini kullanabilir ve 127 bağlantı noktasını destekler [56]. Şekil 3.8'de CANopen haberleşmesi görülmektedir:



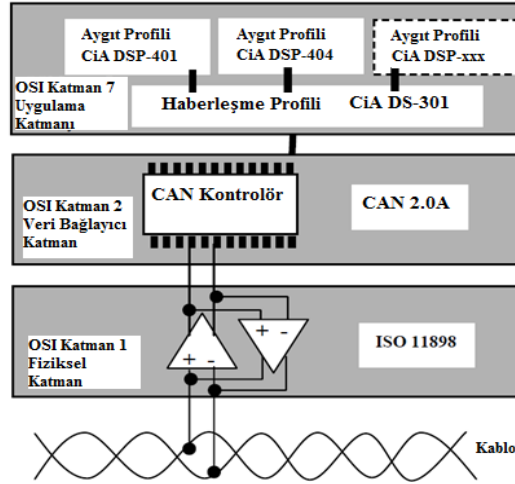
Şekil 3.8. CANOpen haberleşmesi [56].

CAN-BUS sadece katman 1 ve 2'yi tanımlar (ISO11898). CANopen ise mesaj çerçevesinin 11 bit tanımlayıcı ve 8 byte'lık verisini tanımlamak için gerekli yüksek seviyeden bir protokoldür. Farklı üreticilerin cihazlarının birlikte çalışmasını sağlayan CAN tabanlı endüstriyel otomasyon sistemleri kurarken haberleşme sistemi, cihaz işlevselliği ve sistem yönetimini standartlaştıran "uygulama katmanı ve profiller" gerektirir. "Uygulama katmanı" ağ üzerinde her aygıt için faydalı protokoller ve servis ayarları sağlar. Haberleşme profili, cihazlar arasında bilginin nasıl haberleştiğini tanımlar ve iletilen veriyi ve cihazları konfigüre eder [61].

CANopen tabanlı CAN ağları için daha yüksek katmanlı haberleşme protokollerinden biri olan CAL uygulama katmanı, Philips Medikal Sistemler tarafından geliştirilmiştir. CAL, CiA üretici grubu ve bağımsız CAN kullanıcıları tarafından kabul edilerek zamanla geliştirilmiştir [61].

CAL, tüm ağ yönetici servislerini ve mesaj protokollerini sağlar; ancak İçerik Yönetim Sistemleri olan Content Management System (CMS) nesnelerinin içeriklerini ve haberleşilen nesnelerin çeşidini tanımlamaz. Bunun için CANopen protokolü kullanılır. CANopen, CAL servislerinin ve haberleşme protokollerinin bir alt kümesini kullanarak CAL'nin en üstünde yapılandırılır. Açılımı cihaz nesne sözlüğü olan Device Object

Dictionary (OD), CANopen'da merkezi bir kavram olup CAL'da yer almaz. OD'de, 16 bitlik indeks kullanılarak her nesne adreslenir ve düzenli olarak gruplanır. Bilginin yapısının ayrı ayrı bileşenlerine erişebilmek için 8 bitlik bir alt indeks tanımlanır. Ağdaki her düğümün bir OD'si vardır. Bu OD, cihazı ve cihazın ağdaki davranışını tanımlayan tüm parametreleri içerir [61]. OSI ağ modelinde CAN ve CANopen standartları Şekil 3.9'da görülmektedir:



Şekil 3.9. OSI ağ modelinde CAN ve CANopen standartları [61].

CANopen protokolünde dört tip mesaj tanımlanır [61]:

1. Yönetici mesaj (Administrative message): Katman yönetimi, ağ yönetimi ve tanımlayıcı dağıtım hizmetlerinin yönetildiği kısımdır. Bir CAN ağında sadece bir tane master, bir veya daha fazla slave bulunur.

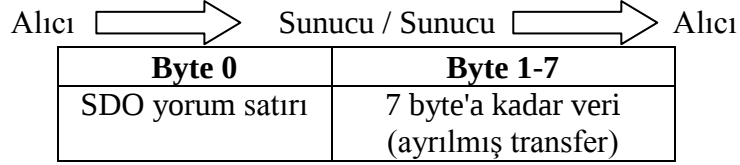
2. Hizmet veri nesnesi (Service Data Object, SDO): SDO, CAN mesajının birkaç byte'ını içeren indeks ve alt indeksini kullanarak bir nesnenin nesne sözlüğüne erişimini sağlar. Mesajı alan alıcı her zaman 8 byte içerir ve alıcının isteği göndericiden alınan cevapla kabul edilir. 2 tip SDO transferi vardır: Hızlandırılmış ve parça parça transfer. Hızlandırılmış transferde 4 byte uzunluğundaki bilgiler gönderilirken, parça parça transferde 4 byte'tan daha uzun veriler gönderilebilir. Bir SDO'nun temel yapısı Tablo 3.4'te gösterilmiştir:

Tablo 3.4. Bir SDO'nun temel yapısı

Alıcı		→	Sunucu / Sunucu	→	Alıcı
Byte 0	Byte 1-2		Byte 3		Byte 4-7
SDO yorum satırı	Nesne indeksi		Nesne alt indeksi		**

(** 4 byte'a kadar veri (genişletilmiş transfer) veya 4 byte'lık byte sayıcı (ayrılmış transfer) veya parametreye ilişkin blok transferdir.) Yeni bir SDO transfer mekanizması:

Tablo 3.5. Yeni bir SDO yapısı



3. İşlem veri nesnesi (Process Data Object, PDO): Gerçek zamanlı bilgiyi bir veya daha fazla alıcıya göndermek için kullanılır. Bilgi gönderimi 1 ile 8 byte arasında sınırlıdır. Her PDO nesne sözlüğündeki iki nesne ile tanımlanır:

- **PDO haberleşme parametresi:** PDO tarafından kullanılan "Haberleşme Nesne Tanımlayıcısı" anlamına gelen "Communication Object Identifier (COB-ID)", aktarım tipi, önleme zamanı ve zamanlayıcı periyodu bilgilerini içerir.
- **PDO eşleştirme parametresi:** PDO'da eşleştirilmiş nesne sözlüğündeki nesnelerin listesini içerir. Bir PDO'daki aktarım tipleri şunlardır:

a. Senkron (SYNC)

a.1. Periyodik olmayan senkron mesajları (acyclic): İletim, başka bir cihazdan gelen uzaktan iletim isteğiyle ön tetiklemeli olarak gerçekleşir.

a.2. Periyodik senkron mesajları (cyclic): İletim, her 1, 2 veya 240 mesaja kadar periyodik olarak tetiklenebilir.

b. Asenkron (ASYNC)

İletim, başka bir cihazdan gelen uzaktan kontrol isteğiyle (CAN kontrol çerçevesi ile) tetiklenir.

4. Önceden tanımlanan mesajlar veya özel fonksiyon nesneleri

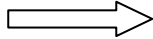
a. Senkronizasyon (SYNC): Sürücü uygulamalarında özellikle önemli olan ağdaki tüm işlemleri senkronize gerçekleştirmektir. Girişlerin gerçek değerleri ağda eş zamanlı olarak saklanır ve gerektiğinde iletilir. Çıkış değerleri önceki senkronizasyondan sonra alınan

mesajlara göre güncellenir. Bir senkron PDO'nun iletimi, senkron mesajının iletimine nazaran belirli bir zaman içinde gerçekleşir.

b. Zaman etkisi (Time stamp): Uygulama cihazları için ortak bir zaman çerçevesi sağlar.

c. Acil durum (Emergency): Cihazda oluşan bir hatada gönderilen mesajdır. Bu mesajlar 8 byte içerir ve Tablo 3.6'daki formatta gönderilir:

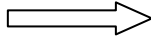
Tablo 3.6. Acil durum mesajları

Acil gönderici  Acil alıcı

COB-ID	Byte 0-1	Byte 2	Byte 3-7
0×080+Düğüm_ID	Acil Hata Kodu	Hata Kaydı (Nesne 0×1001)	Üretici özel hata alanı

d. Düğüm/Hayat koruma mesajı (Node/Life Guarding): Master-slave kavramıdır. Ağ yönetici nesne anlamına gelen "Network Management Object (NMT)"in ayrı ayrı düğümlerin durumunu gözlemlemesine düğüm koruma (node guarding) adı verilir. Düğümlerin de isteğe bağlı olarak NMT master'ının durumunu izlemesi işlemine hayat koruma (life guarding) denir. Bu işlem, NMT master'dan ilk düğüm kurtarma mesajını aldıktan sonra NMT slave'i üzerinde başlar. NMT master Tablo 3.7'de görülen veri byte'ları olmayan bir CAN istek çerçevesi gönderir:

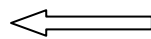
Tablo 3.7. NMT mesajı

NMT Master  NMT Slave

COB-ID
0×700+Düğüm_ID

NMT slave, master'dan gelen mesajı Tablo 3.8'deki gibi cevaplar:

Tablo 3.8. NMT slave'in cevap mesajı

NMT Master  NMT Slave

COB-ID	Byte 0
0×700+Düğüm_ID	Bit 7:Değiştirme biti, Bit 6-0: Durum

Burada verinin 7. biti, her düğüm koruma için 0 ile 1 arasında değişen anahtarlama bitidir. 0'dan 6. bite kadar değişen bitler ise düğümün durumunu gösterir. Bu değerler Tablo 3.9'da verilmiştir:

Tablo 3.9. Bit değerleri ve durumları

Değer	Durum
0	Başlatılıyor
1	Bağlanamadı*
2	Bağlanıyor*
3	Hazırlanıyor*
4	Durduruldu
5	Çalışma durumunda
127	Çalışma öncesi durumda

CAN ağında bir master vardır. Diğer düğümler slave'lerdir. NMT master tüm slave'leri kontrol edebilir. NMT mesaj formatı Tablo 3.10'da gösterilmiştir:

Tablo 3.10. NMT mesaj formatı

NMT Master	→	NMT Slave
COB-ID	Byte 0	Byte 1
0×000	CS	Düğüm ID'si

ID numarası 0 ile tüm slave'ler adreslenir. "Command Specifier (CS)" denilen komut belirleyici ise Tablo 3.11'de verilen değerleri alır.

Tablo 3.11. CS değerleri

Yorum belirteci (CS)	NMT Servis
1	Uzaktan başlatma düğümü
2	Uzaktan durdurma düğümü
128	Çalışma öncesi durumunu gir
129	Düğümü resetleme
130	Haberleşmeyi resetleme

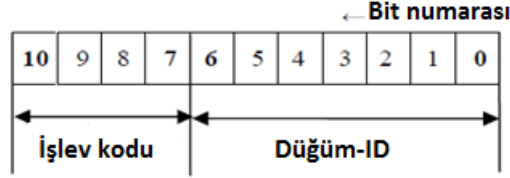
e. Başlatma mesajı (Boot up): Master-slave kavramı olan bu mesaj gönderilerek, NMT master'ın çalışma öncesi durumundan ilk durumuna geçişini gösterir. Bir NMT slave, ilk durumdan çalışma öncesi durumuna geçişini NMT master'a göstermek için Tablo 3.12'deki gibi bir uyandırma mesajı gönderir:

Tablo 3.12. Başlatma mesajı

NMT Master ← NMT Slave	
COB-ID	Byte 0
0×700+Düğüm_ID	0

Özetlenirse, SDO CANopen ağındaki cihazların konfigürasyonunda cihazlar arasında düşük öncelikli bilgi transferinde kullanılırken, PDO ise 8 byte'lık ya da daha az bilginin

hızlı transferinde kullanılır. Şekil 3.10'da görüldüğü gibi 11 bitlik CAN tanımlayıcısının 4 biti işlev kodu olan "function code", kalan 7 biti de ID numarasını gösterir. Tablo 3.13'te de CAN tanımlayıcılarının ataması detaylı olarak gösterilmiştir:



Şekil 3.10. 11 bitlik CAN tanımlayıcısı [61].

Tablo 3.13. CAN tanımlayıcılarının ataması [61].

Master/Slave haberleşme ayarı tanımlanmadan önce CANopen'ın yayınladığı nesneler			
Nesne	İşlev Kodu (ID bitleri 10-7)	COB-ID	OD indeksindeki haberleşme parametreleri
NMT Modül Kontrol	0000	000h	-
SYNC	0001	080h	1005h, 1006h, 1007h
Zaman Damgası	0010	100h	1012h, 1013h

Master/Slave haberleşme ayarı tanımlanmadan önce CANopen'ın eş düzey nesneleri			
Nesne	İşlev Kodu (ID bitleri 10-7)	COB-ID	OD indeksindeki haberleşme parametreleri
ACİL	0001	081h-0FFh	1024h, 1015h
PDO 1 (gönderen)	0011	181h-1FFh	1800h
PDO 1 (alan)	0100	201h-27Fh	1400h
PDO 2 (gönderen)	0101	281h-2FFh	1801h
PDO 2 (alan)	0110	301h-37Fh	1401h
PDO 3 (gönderen)	0111	381h-3FFh	1802h
PDO 3 (alan)	1000	401h-47Fh	1402h
PDO 4 (gönderen)	1001	481h-4FFh	1803h
PDO 4 (alan)	1010	501h-57Fh	1403h
SDO (gönderen/sunucu)	1011	581h-5FFh	1200h
SDO (alan/alıcı)	1100	601h-67Fh	1200h
NMT Hata Kontrolü	1110	701h-77Fh	1016h-1017h

Tablo 3.14'te görüldüğü gibi alt indeksi 2 olan bir PDO mesaj örneği [61]:

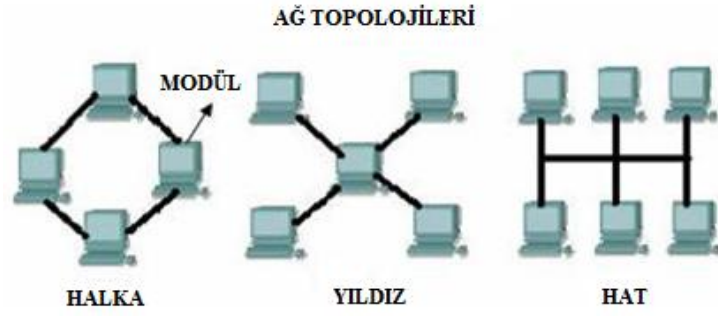
Tablo 3.14. PDO mesaj örneği.

PDO Üreticisi		PDO Alıcı veya Alıcıları	
COB-ID	Byte 0	Byte 1	Byte 2
0x280+Düğüm_ID	8 bitlik dijital giriş	16 bitlik analog girişin en düşük ağırlıklı biti (LSB)	16 bitlik analog girişin en yüksek ağırlıklı biti (MSB)

- **PCAL:** PCAL, Münih German Federal Armed Forces Üniversitesi tarafından geliştirilmiş bir CAL uygulamasıdır.
- **DeviceNet:** CiA onaylı, Rockwell//Allen-Bradley tarafından geliştirilmiş, genellikle endüstriyel otomasyonda kullanılan bir sistemdir.
- **SDS:** Honeywell tarafından geliştirilmiş, CiA onaylı bir uygulama katmanıdır. Genellikle makine kontrol uygulamalarında kullanılır.
- **CAN Kingdom:** İsveç Firması Kvaser AB tarafından geliştirilmiş, yine CiA onaylı bir uygulama katmanıdır [9].

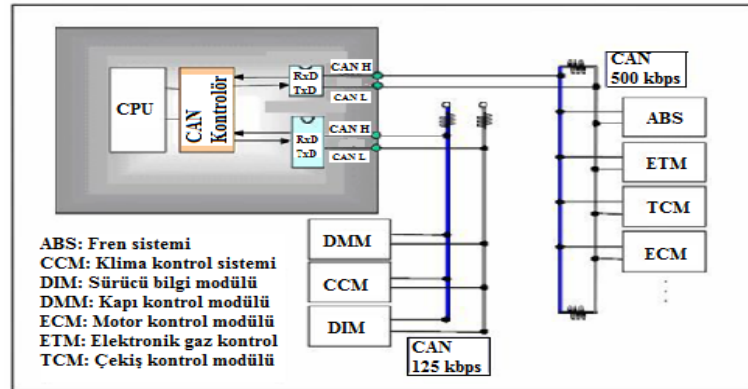
3.6.1.2. CAN Topolojisi

CAN topolojisi, ağ üzerindeki CAN modüllerini birbirine bağlayan kablonun yapısını ifade eder. CAN protokolünde Hat, Halka veya Yıldız topolojilerinden biri kullanılır. En yaygın kullanılan Hat topolojisidir. Bu topolojide, her modül ağ içerisinde tek bir seri kablo ile mesajlaşmayı sağlar [3]. Hatta bağlanan tüm cihazların mesaj önceliği önemlidir. Her üniteye bir ID numarası verilir. Her üniteye bir çevrim süresinde bir kez bilgi aktarımı yapılabilir. CAN ve Profibus protokolleri bu topolojiye örnektir [57]. Halka topolojisinde, bir bilgi bir üniteden diğer üniteye aktarılarak ilerler ve öncelik sırası önemlidir. Adresleme yapılmadığından her ünitenin adresi halka yapısı içerisindeki fiziksel konumu üzerinden belirlenir. Her çevrimde sadece bir komut iletilir. Hatta bağlı olan ünitelerden herhangi biri arızalanırsa tüm hat hata verecektir. Interbus-S protokolü bu topolojiye örnektir. Yıldız topolojisinde ise, bilgi bir merkezi kontrol sistemi tarafından kontrol edilir. Kablolamanın fazla olması en büyük dezavantajıdır. Herhangi bir arızada sadece arızanın verdiği hat ve cihaz çalışmaz. Diğer cihazlar aynı şekilde çalışmalarını sürdürürler. Ancak merkezi kontrol sistemi arızalanırsa tüm sistemin çalışması durur [57]. Şekil 3.11'de bu üç topoloji gösterilmektedir [15]. Bunların yanı sıra ağ yapılı ve ağaç topolojileri de vardır. Ağ topolojisinde, iki ünite birbirine bağlıdır ve bir arızada hatayı bulmak kolaydır. Ağaç topolojisi de hat topolojisinin daha gelişmiş halidir. Tek bir hat yerine bir bölge kontrol edilir ve her bir ünitenin bir adresi vardır. AS-Interface bu topolojiye örnektir [57].



Şekil 3.11. CAN protokolünde kullanılan topolojiler [15].

CAN haberleşme sisteminde kontroller iki kablo arasındaki potansiyel farka bakarak gelen veri hakkında karar verir. Baskın ve çekinik olmak üzere iki seviye vardır. Bu iki seviye aynı anda veri yolunda olabilirler. Gönderen kontrolcü, bu veri yolu seviyelerini değiştirerek alıcıya mesaj gönderir. Tüm düğümler mesaj gönderebilir ancak mesaj önceliği olan düğümün mesajı öncelikle iletilir [11]. Tipik bir CAN bağlantısı Şekil 3.12'de gösterilmektedir:



Şekil 3.12. Tipik bir otomobil CAN bağlantı şekli [11].

3.6.1.3. CAN-BUS Çalışma Mantığı

CAN-BUS sisteminde tüm düğümler iletim hattına eşit öncelikli veri yollama hakkına sahiptir. Bu da 'Multimaster çalışma' olarak adlandırılır. Eğer aynı anda veri gönderilirse çatışmalar olacağından iletim hattının boş olduğu anda veri gönderilir. Bir düğüm veri gönderirken, uzak mesafelerden dolayı başka bir düğüm mesajın hatta olduğunu fark edemeyip veri gönderirse çatışma (collision) gerçekleşir. Bu durumda iki düğüm de çatışmayı anlar ve veri göndermeyi bir süre bekletirler. Hat boş olduğunda verilerini tekrar gönderirler. CAN hattına gelen tüm veriler ilgili düğümler tarafından alınmaktadır [62].

CAN protokolü, mesajların eşit erişimli yollanmasını sağlarken, diğer yandan öncelikli mesajların daha erken iletilmesine de dikkat eder. Her mesajın bir ID numarası vardır ve mesajlar o ID'ye gönderilir. CAN, iletişim ortamına erişim yöntemi olarak bit öncelikli yapı olan "Carrier Sense Multiple Access with Collision Detection (CSMA/CD)" kullanarak gönderilen mesajların çarpışmamasını sağlar ve iletişim hattının uzunluğunu sınırlandırır [62, 63, 7]. Böylece düğümler, CAN hattına 1 Mbit/s veri iletim hızı ile 40 m ve 40 Kbit/s veri iletim hızıyla ise 1000 m'lik bir veri yolundan bağlanabilirler [62]. Tablo 3.15'te bus uzunluğuyla veri iletim hızının değişimi detaylı görülmektedir [16]:

Tablo 3.15. İletim mesafelerine göre haberleşme hızları [16].

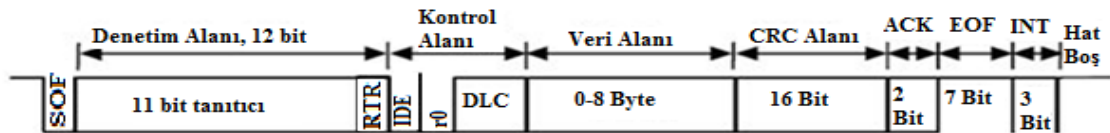
Bus Uzunluğu (m)	Maksimum Bit Oranı (bit/s)
40	1 Mbit/s
100	500 Kbit/s
200	250 Kbit/s
500	125 Kbit/s
1000	40 Kbit/s

3.6.1.4. CAN Mesaj Çerçeveleri

CAN protokolü dört mesaj tipinden oluşur. Bunlar: Veri çerçevesi (data frame), istek çerçevesi (remote frame), hata çerçevesi (error frame) ve taşma çerçevesidir (overload frame).

• Veri Çerçevesi (Data Frame)

Veriyle ilgili bilgilerin yer aldığı çerçevedir. Standart ve genişletilmiş veri çerçevesi olmak üzere iki çeşittir. Şekil 3.13'te standart CAN-BUS veri çerçeve yapısı ve Şekil 3.14'te genişletilmiş CAN-BUS veri çerçeve yapısı görülmektedir:



Şekil 3.13. Standart CAN-BUS veri çerçeve yapısı [62].



Şekil 3.14. Genişletilmiş CAN-BUS veri çerçeve yapısı [16].

Bu yapılarda ortak olan alanlar şunlardır:

- **Çerçeve Başlangıcı (Start of Frame, SOF):** Çerçeve başlangıç bitidir [64].
- **Mesaj ID Alanı (Arbitration Field):** CAN protokolünde veriler paketler halinde iletilir. CAN 2.0A ve CAN 2.0.B olmak üzere iki tip paketleme vardır. "Standart CAN" olarak adlandırılan CAN 2.0A, 12 bit tanımlayıcıya sahiptir. Bunun 11 biti mesajı tanımlamada kullanılırken, 1 biti de Remote Transmit Request (RTR) bitidir. "Geliştirilmiş CAN" olarak adlandırılan CAN 2.0B ise 32 bit tanımlayıcıya sahiptir. Bunun 29 biti mesaj tanımlamada kullanılır. 1 biti, belirleyici denilen Identifier (IDE) biti, 1 biti Substitute Remote Request (SRR) bitidir. 1 bit de RTR biti bulunur. Standart CAN'de $2^{11} = 2048$ mesaj tanımlanabilirken, geliştirilmiş CAN'de $2^{29} = 536.870.912$ mesaj tanımlanabilir. Böylece bu bilginin tutulduğu alana Mesaj ID Alanı (Arbitration Field) denir. Mesaj önceliği burada belirlenir. ID numarası düşük olan mesajlar yüksek önceliklidir.
- **Uzak Gönderim İsteği (RTR):** Bu bit, veri çerçevesi mi yoksa istek çerçevesi mi olduğunu gösterir [16]. Veri çerçevesinin uzaktaki bir cihazdan istekte bulunmasında kullanılır. Veri çerçevesi için değeri 0, istek çerçevesi için ise 1'dir.
- **Yedek Uzak İsteği (SRR):** SRR biti sadece genişletilmiş çerçeve yapısında kullanılır. RTR bitinin yedeğidir ve her zaman lojik 1 değerini alır.
- **Tanımlayıcı Uzatma (IDE):** Standart ya da genişletilmiş bir veri çerçevesi olup olmadığını gösterir. Standart çerçevede değeri 0, genişletilmiş çerçevede ise 1'dir.
- **r0:** Ayrılmış bittir, kullanılmamaktadır [62].
- **Veri Uzunluk Kodu (Data Length Code, DLC):** Veri uzunluk kodu olup, kontrol alanının son 4 bitlik kısmını oluşturur. Veri alanının kaç byte'tan oluştuğunu gösterir [62, 65].
- **Veri Alanı (Data Field):** Veri alanı en fazla 8 byte'tan oluşur. CAN bilgisinin taşındığı kısımdır [9].
- **Çevrimsel Hata Kontrolü (Cyclic Redundancy Check Field, CRC):** Açılımı "Çevrimsel Hata Kontrolü" anlamına gelen CRC, 15 bitlik alana sahip, hata denetlemede kullanılan bölümdür. CRC, verilerin kablodan bozulmadan iletebilmesi

için veriyi matematiksel işlemde geçirir. Bunun sonucu CRC kodunu verir. Alıcı kısmında veri alındıktan sonra aynı matematiksel işlemle verinin bozulmadan gelip gelmediğine bakılır. Sonuç, CRC kodu ile aynı olmalıdır. Olmadığı durumda veri tekrar gönderilir [15]. 15 bitlik CRC Dizi (Sequence) ve CRC Sınırlayıcı'dan (Delimiter) oluşur. Pakete ait CRC kodunu tutar. Üniteye gelen pakete cevap verilebilmesi için paketin doğruluğu tespit edilir. Bunun için paketin CRC değeri hesaplanarak, alıcı ünite paket ile birlikte gelen CRC değeri ile hesaplanan değeri karşılaştırır. Değerler birbirine eşitse alınan paket geçerlidir; değil ise hatayı belirten bir Hata Çerçevesi (Error Frame) gönderir. Bu mesajı tek bir ünite dahi gönderse veri tekrar yollanmalıdır [62].

- **Çevrimsel Hata Kontrol Sınırlandırıcı (CRC Delimiter):** Her zaman lojik 1 değerlikli bittir. İletim çerçevesi tarafından doğruluğu onaylandıktan sonra bu bitin değeri lojik 0 olur [9].
- **Alındı Bilgisi Alanı (Acknowledgement Field, ACK):** Bu bit de her zaman lojik 1 değerindedir. Mesaj iletilirken başla biti gönderilerek hatta gönderici belli olur. Daha sonra mesaj ID alanı, kontrol alanı, veri alanı, CRC alanı gönderilir. Alındı bilgisi alanında, bilginin alınıp alınmadığı tespit edilir. Eğer herhangi bir üniteden alındığı bilgisi gelmezse ACK hatası oluştuğunu belirten hata çerçevesi üretilir ve bilgi tekrar yollanır [62].
- **Çerçevenin Sonu (End of Frame, EOF):** Çerçeve sonudur. 7 tane lojik 1 değerini alan bitlerdir.
- **Ara Alan (Intermission Field, INT):** Çerçeveler arası boşluktur. Arka arkaya veri ya da istek çerçevesi gönderilirse bunları birbirinden ayırmak için kullanılan alandır. 3 biti lojik 1 değerindedir. İki çerçeve arasındaki boş sürede hattın lojik 1 olması gerekir ve bu süre çerçeveler arası boşluğu ifade eder. Hattın lojik 0 değerini alması yeni bir çerçevenin başladığı anlamına gelir [9].

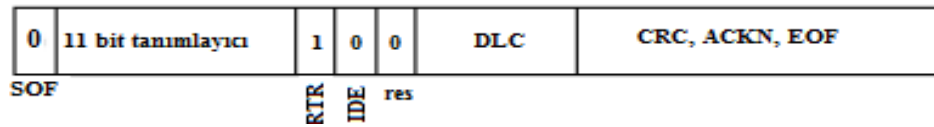
CAN-BUS protokolünün çerçeve alanları ve alabilecekleri değerler Tablo 3.16'da gösterilmiştir [66]:

Tablo 3.16. CAN-BUS protokolünün çerçeve alanları ve değerleri [66].

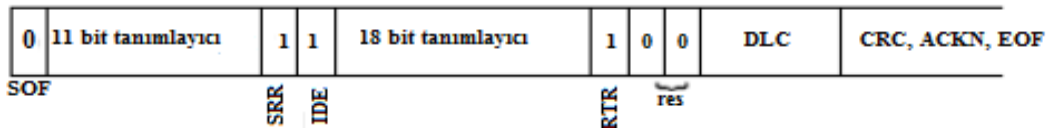
Çerçeve Alanı	Değer/Uzunluk	Açıklama
SOF	0	Her zaman lojik-0 değeri alır.
SRR	1	Sadece genişletilmiş protokolde kullanılır.
RTR	0/1	İstek mesajlarında lojik-1, veri mesajlarında lojik-0 alır.
R0, r1	00	Rezerve edilmiş alandır. Her zaman lojik 0 alırlar.
DLC	0000...1000	Veri uzunluk kodudur.
CRC	15 bit	Gönderici tarafından hesaplanan 15 bit kontrol alanıdır.
CRC Sınırlayıcı	1	CRC alanını sonlandırmak için kullanılır.
ACK	0/1	Gönderici lojik-1 olarak gönderir, alıcı lojik-0 yazar.
ACK Sınırlayıcı	1	Her zaman lojik-1 değerindedir.
EOF	1111111	Her zaman 7 adet lojik-1 alır.

- **İstek Çerçevesi (Remote Frame)**

İstek çerçevesi, bir cihazın başka bir cihazdan bilgi istemesi durumunda kullanılır. Bu da veri çerçevesindeki RTR biti ile ifade edilir. Eğer RTR biti 1 ise gönderilecek pakete "İstek Çerçevesi" (Remote Frame), 0 ise "Veri Çerçevesi" (Data Frame) adı verilir. Eğer bir ünite istek çerçevesi yolluyorsa veri alanı yok demektir. Çünkü istenen verinin bilgisi mesaj ID alanında yer alır [62]. Standart ve genişletilmiş istek çerçevesinin yapısı Şekil 3.15 ve Şekil 3.16'da gösterilmektedir:



Şekil 3.15. Standart istek çerçevesi [16].

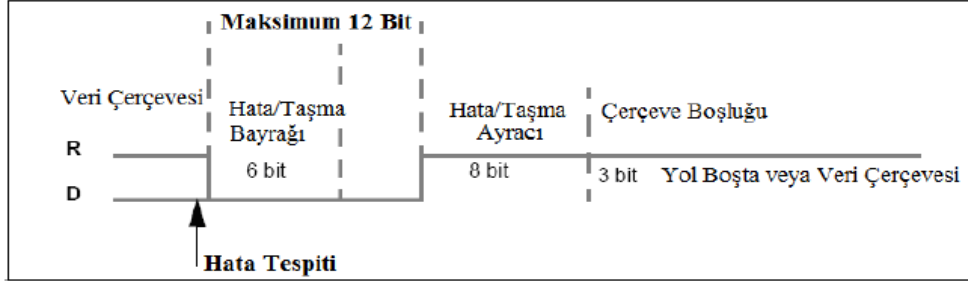


Şekil 3.16. Genişletilmiş istek çerçevesi [16].

- **Hata Çerçevesi (Error Frame)**

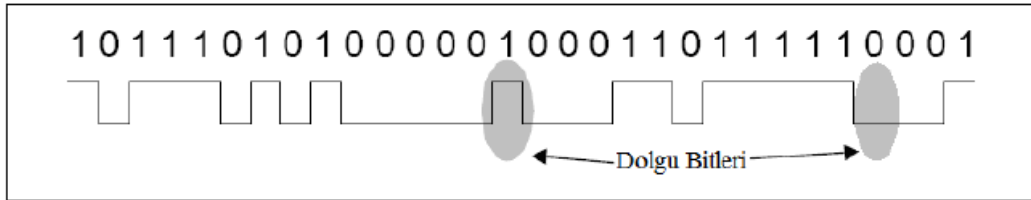
Veri çerçevesi ya da istek çerçevesi bilgi gönderirken veya alırken bir hata oluşursa ne tür bir hata olduğunu gösterir [62]. Şekil 3.17'de hata çerçevesinin yapısı görülür. Bir hata

çerçevesi içerisinde 6 veya 12 adet aynı lojik seviyeli bit ve 8 adet hata sınırlama biti bulunur. Aynı seviyeli her 6 bitlik grup hata bayrağı olarak ifade edilir. Gönderici ya da alıcı modül hata bayrağı gönderdiği zaman diğer modüller bit dolgu (bit stuffing) kuralının ihlal edildiğini anlar ve hepsi tekrar hata bayrağı gönderir.



Şekil 3.17. Hata ve taşma çerçeve yapısı [16].

Bit dolgu yönteminde, CAN veri ve istek çerçevesi içerisinde arka arkaya aynı lojik seviyeli 5 bit gönderildiğinde 6. bitine ters lojik seviyeli bit eklenir. Bunun nedeni örnekleme işlemi için çok sayıda kenar kullanarak çerçeve senkronizasyonunu sağlamaktır. Alıcı tarafından arka arkaya aynı seviyeli 5 bitin gözlenmesinden sonra 6. bitin silinmesiyle kaldırılır. Şekil 3.18'de bit dolgu işlemi gösterilmiştir [16]:



Şekil 3.18. Bit dolgu işlemi [16].

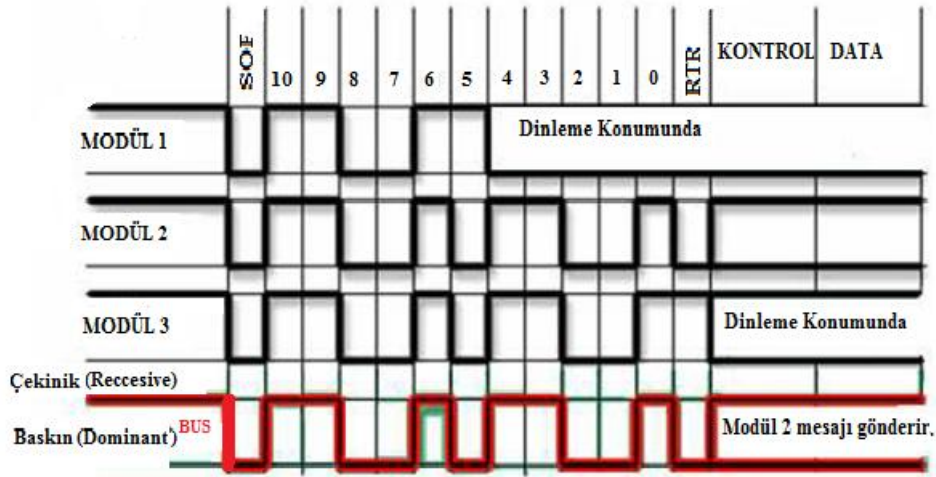
- **Taşma Çerçevesi (Overload Frame)**

Şekil 3.17'de gösterilen taşma çerçevesi, veriyi alacak ünitenin bir sonraki alacağı veri için dahili çalışma zamanının yetmemesi ve çerçeveler arasındaki alanda baskın bir bitin olması durumunda devreye giren çerçevedir [16].

3.6.1.5. Bit Gönderme İşlemi (Arbitration)

Bit seviyesinde bit gönderme işleminin gerçekleşmesi için lojik durumların baskın (dominant) ve çekinik (recessive) olarak tanımlanması ve veri gönderen modül tarafından gönderme girişimlerinin lojik durumlarının nasıl olduğunun izlenmesi gerekir. Lojik 0

seviyesi baskın, lojik 1 ise çekinik olarak CAN protokolünde tanımlanır. Bit gönderme işleminde her zaman baskın bit çekinik bitin önüne geçer. ID'si düşük olan mesaj yüksek öncelikli olacaktır. Aynı anda iki modül mesaj gönderirse öncelikle bus izlenir ve düşük öncelikli mesaj bir süre sonra çekinik bir bit göndermeyi dener. Bu durumda biti gönderen modül, bit gönderme işlemini kaybederek aniden mesaj gönderme işlemini durdurur. Gönderme işlemi bitene kadar yüksek öncelikli mesaj bus üzerinden istenilen yerlere gider. Bit gönderme işlemini kaybettikten sonra modül boş işlem periyodunu bekleyerek mesajını tekrar gönderir. Bu durum Şekil 3.19'da gösterilmiştir. Mesajını gönderen modül bit gönderme işlemini kaybederek aniden mesajını göndermeyi durduracaktır ve yüksek öncelikli mesaj, gönderme işlemi bitene kadar bus üzerinde istenen yerlere gider [15].

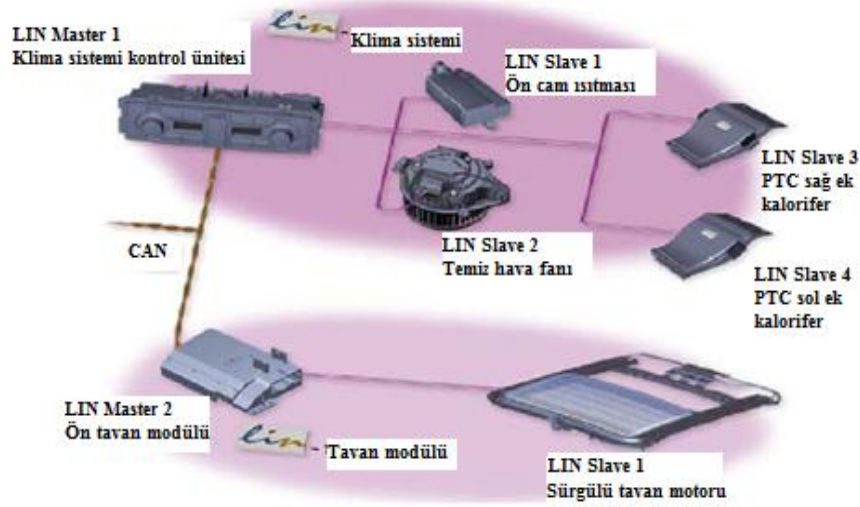


Şekil 3.19. Bit gönderme işlemi [15].

3.6.2. Yerel Bağlantı Ağı (LIN)

Günümüzde araçlarda elektrikli cam, klima, merkezi kilit, elektrikli açılır tavan (sunroof) gibi pek çok konfor donanımı mevcuttur. Eski yıllarda bu donanımların sayısı azken kablo bağlantısı kullanılıyordu. Son yıllarda arıza tespitinin zor olması ve kablo sayısının artmasıyla oluşan zorluklar sebebiyle 1998 yılında BMW, Daimler-Benz, Volkswagen, Volvo üreticisi olarak bilinen Motorola ve network uzmanı Volcano Communication Technologies, "Yerel Bağlantı Ağı" anlamına gelen "Local Interconnect Network (LIN)" adında bir ağ geliştirmiştir. LIN, daha düşük veri iletim kapasitesi gerektiren konfor donanımında kullanılır. Kontrol ünitelerinin tümü aynı yapı bölgesi içinde bulunur. Şekil 3.20'de LIN hattının örnek kontrol ünitesi görülür. Bunun görevleri şunlardır [53]:

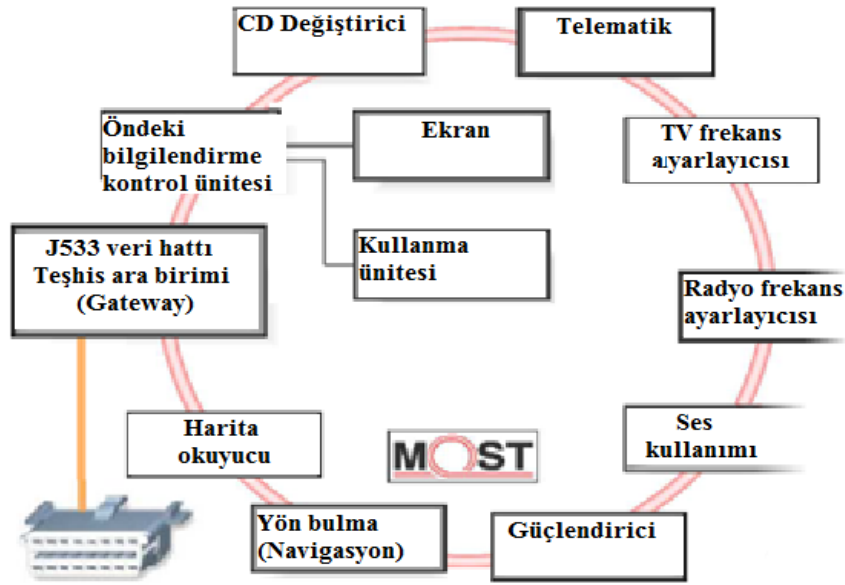
- Mesaj başlığını gönderir.
- Bölgesel LIN veri sisteminin LIN kontrol üniteleri ve CAN-çift yönlü veri hattı arasındaki dönüşüm fonksiyonlarını ve parça kontrol ünitelerinin arıza tespitini gerçekleştirir.
- LIN veri sistemindeki kontrol ünitesine bağlı tek CAN-çift yönlü veri hattıdır.
- Veri aktarım hızını ve veri transferini kontrol eder.
- Hatta ne zaman ve ne kadar aralıkla hangi mesajın gönderileceği yazılımın içindeki bir periyotla yapılır.



Şekil 3.20. LIN ana kontrol ünitesi [53].

3.6.3. Medya Odaklı Sistemleri Taşıma Ağı (MOST)

Araçlardaki radyo, televizyon, telefon, navigasyon gibi bilgilendirme ve eğlence donanımları için geliştirilmiş yüksek hızda veri iletimi yapan ağıdır. Açılımı "Medya Odaklı Sistemleri Taşıma" olan "Media Oriented Systems Transport (MOST)" ağı, adrese yönelik mesajların belirli bir alıcıya aktarılması anlamını taşır. 1998'de BMW, Daimler Benz, Becker ve OASIS Silicon Systems tarafından Most Cooperation kurulmuştur. Sonradan Audi de katılmış ve çalışmalar yapılmıştır. MOST hattı 21.2 Mbit/s'de bilgi aktarır. MOST kullanılan sistemler Şekil 3.21'de görülmektedir [53].

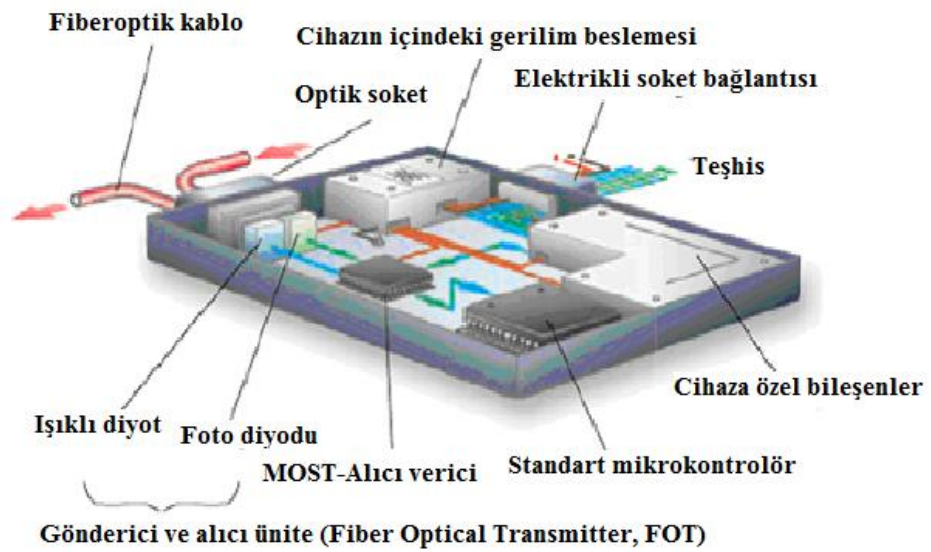


Şekil 3.21. MOST hattında kullanılan sistemler [53].

MOST hattının avantajları:

- Işık dalgaları yardımıyla veri aktarımı,
- Daha az kablo ağırlığı,
- Yüksek veri aktarım hızı,
- Kısa dalga uzunluğu ve elektromanyetik parazitli dalgalar üretmemesidir [53].

MOST kontrol ünitesinin yapısı Şekil 3.22'de detaylı görülmektedir:



Şekil 3.22. MOST kontrol ünitesinin yapısı [53].

3.7. Otomobillerde CAN-BUS Tercih Edilmesinin Nedenleri

- **Mesaj Önceliği:** CAN protokolü ile mesaj iletimi hızlı ve veri kaybı olmadan yapılabilir. İki veya daha fazla modül aynı anda mesaj gönderirse iki modül de bir bozulma olduğunu düşünerek mesajını göndermez. Bir süre sonra rastgele mesajını tekrar yayımlar ve hat boşsa mesaj iletilir. Ethernet kartlarında bu durum zaman kaybına neden olurken CAN protokolünde bu sıkıntı bit gönderme işlemi ile çözülür. İki veya daha fazla mesaj aynı anda gönderilirse daha yüksek öncelikli mesaj zaman ve veri kaybı olmadan iletilir. Bu durumda daha düşük öncelikli mesaj geri çekilerek hattaki mesajlar dinlenir. Öncelikli mesajlar iletilip hat boş kaldıktan sonra diğer mesajlar gönderilir [15].
- **Mesaj Yönlendirme:** CAN-BUS hattına her gönderilen mesajın bir ID'si vardır. Böylece mesaj filtreleme özelliğiyle hatta bağlı tüm birimler mesajın kendisine ait olup olmadığını anlarlar [15].
- **Çoğa Gönderim Yapılması (Multicast):** Mesajlar filtrelendikten sonra bir mesajı aynı anda birden çok birim olarak değerlendirebilir [15].
- **Veri Kararlılığı:** Mesajlar gönderildikten sonra aynı anda tüm birimler değerlendirerek mesajı kabul eder veya etmezler. Bu kabul edilme veya edilmemesi konuları ve kararlılık sistemi çoğa gönderim ve hata bulma sistemleriyle gerçekleştirilir [15].
- **Öncelikler:** ID numarası hattaki öncelik sırasını belirler. Düşük ID'li mesajlar yüksek önceliklidir [15].
- **Güvenilirlik:** CAN-BUS sisteminde oluşan hataları bulmak için gönderilecek bit seviyeleri ile hattaki bit seviyelerinin karşılaştırılması, CRC alanı, bit doldurma, mesajın bütünlük kontrolü, tüm global hataların belirlenmesi, her sensördeki yerel hataların belirlenmesi, mesajlarda beş adete kadar olan farklı hataların belirlenmesi, mesajlarda tekil sayı hatalarının ve uzunluğu 15 bitten az olan patlama (burst) hatalarının belirlenmesi gibi pek çok yöntem uygulanır. 4 farklı hata tespit mekanizması eş zamanlı çalışır [66]. Hata tespit edildiğinde bunu bulan birim hatayı işaretler ve bu mesajlar iptal edilir. Hatalı mesajdan en fazla 31 bit süre sonra tekrar yenisi gönderilir [15].

- **Haberleşme Hızı:** Ağ büyüklüğüne ve mesafelere göre haberleşme hızı değişirken, aynı ağda bulunan tüm düğümler aynı haberleşme hızına sahiptirler [17]. Eğer bir düğüm farklı bir haberleşme hızına sahipse hata mesajı sadece bu ağdaki düğüme gönderilir [11]. Ayrıca CAN-BUS protokolü ISO standartlarına göre en fazla 1 Mbit/s oranında iletişim yapabilirken, CAN-BUS aygıtları sayesinde iletişim hızı 5 Kbit/s ile 1 Mbit/s arasında ayarlanabilir [66].
- **Sistem Esnekliği:** Ağdaki düğümler belirgin bir ID'ye sahip değildirler. Bundan dolayı veri yoluna bir düğüm eklenip çıkartılması yazılımını, diğer düğümlerin donanımını ya da uygulama alanlarını değiştirmez. Bu da sistemin kolaylıkla genişletilmesine olanak verir [11].

Çok sayıda birimi tek hat üzerinden haberleştirerek kablo azlığı imkanı ve böylece daha düşük maliyet, hata tespitinin iyi olması sonucu güvenilir olması, saniyede 10000 mesaj iletimi, veri uzunlukları kısa olduğundan, diğer haberleşme sistemlerine göre gecikme süresinin düşük olması, hata bulma mekanizmalarının gelişmiş ve tepki sürelerinin hızlı olması gibi pek çok avantajları vardır [1, 67, 10]. Ayrıca maliyetleri düşük, hızlı, sağlam, esnek ve güvenilir bir haberleşme sistemidir [68, 69, 70, 71, 72, 5]. Bu protokol, en fazla 8 byte uzunluğuna sahip verileri haberleştirdiği için gerçek zamanlı haberleşme sağlanmaktadır [8, 14]. Bu pek çok önemli özelliğiyle CAN-BUS teknolojisi günümüzde araçlarda yaygın olarak kullanılmaktadır [15].

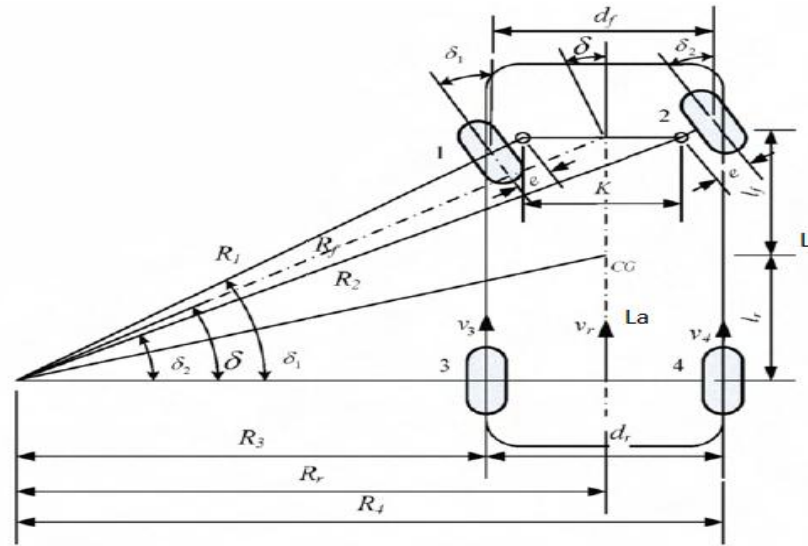
4. TEKERLEK İÇİ MOTORLU ELEKTRİKLİ ARAÇLARDA ELEKTRONİK DİFERANSİYEL SİSTEM TASARIMI

4.1. Giriş

Bu tez çalışmasında, HY-TTC 60 işlemcisi kullanılarak CoDeSys V2.3 yazılımı üzerinden elektronik diferansiyel sistem tasarlanmıştır. Bunun için direksiyon açısının ve araç hızının değişimine göre aracın ön tekerleklerinin hızları Ackermann-Jeantand Modeli'nden alınan matematiksel denklemlere göre CoDeSys V2.3 yazılımı ile hesaplanmıştır. Hesaplanan hız bilgileri CAN-BUS protokolü üzerinden ön tekerlek içi motorlara gönderilerek elektronik diferansiyel gerçekleştirilmiştir.

4.2. Ackermann-Jeantand Modeli

Direksiyon açısına ve aracın hız bilgisine göre elektrikli aracın ön tekerlek içi motor hızları Şekil 4.1'de gösterilen Ackermann-Jeantand Modeli'nden alınan matematiksel denklemlerden elde edilmiştir:



Şekil 4.1. Düşük hızda sürüş eğiminin Ackermann-Jeantand modeli [27].

Buna göre ön iç tekerlek açısı $\delta_{öi}$;

$$\delta_{öi} = \arctan \left[\frac{L * \tan(\rho)}{L - ((K/2) * \tan(\rho))} \right] \quad (4.1)$$

Ön dış tekerlek açısı $\delta_{öd}$;

$$\delta_{öd} = \arctan \left[\frac{L * \tan(\rho)}{L + ((K/2) * \tan(\rho))} \right] \quad (4.2)$$

Burada;

L: Ön tekerlek ile arka tekerlek arasındaki mesafe,

K: King Pim arasındaki mesafe,

ρ : Direksiyon açısıdır.

Açısal hızları bulmak için virajdaki yarıçaplar bulunur:

$$R_{öi} = \frac{L}{\sin(\delta_{öi})} \quad (4.3)$$

$$R_{öd} = \frac{L}{\sin(\delta_{öd})} \quad (4.4)$$

$$R_{ai} = \frac{L}{\tan(\rho)} - \frac{d}{2} \quad (4.5)$$

$$R_{ad} = \frac{L}{\tan(\rho)} + \frac{d}{2} \quad (4.6)$$

$R_{öi}$: Ön iç tekerleğin viraj yarıçapı,

$R_{öd}$: Ön dış tekerleğin viraj yarıçapı,

R_{ai} : Arka iç tekerleğin viraj yarıçapı,

R_{ad} : Arka dış tekerleğin viraj yarıçapı,

d: Arka iki tekerlek arasındaki mesafedir.

Tekerleklerin açısal hızları:

$$w_{öi} = \frac{V.R_{öi}}{(R_{merkez}).r} \quad (4.7)$$

$$w_{öd} = \frac{V.R_{öd}}{(R_{merkez}).r} \quad (4.8)$$

$$w_{ai} = \frac{V \cdot R_{ai}}{(R_{merkez}) \cdot r} \quad (4.9)$$

$$w_{ad} = \frac{V \cdot R_{ad}}{(R_{merkez}) \cdot r} \quad (4.10)$$

$$R_{merkez} = \sqrt{(R_{ai} + (d/2))^2 + (L_a)^2} \quad (4.11)$$

$w_{öi}$: Ön iç tekerleğin açısal hızı,

$w_{öd}$: Ön dış tekerleğin açısal hızı,

w_{ai} : Arka iç tekerleğin açısal hızı,

w_{ad} : Arka dış tekerleğin açısal hızı,

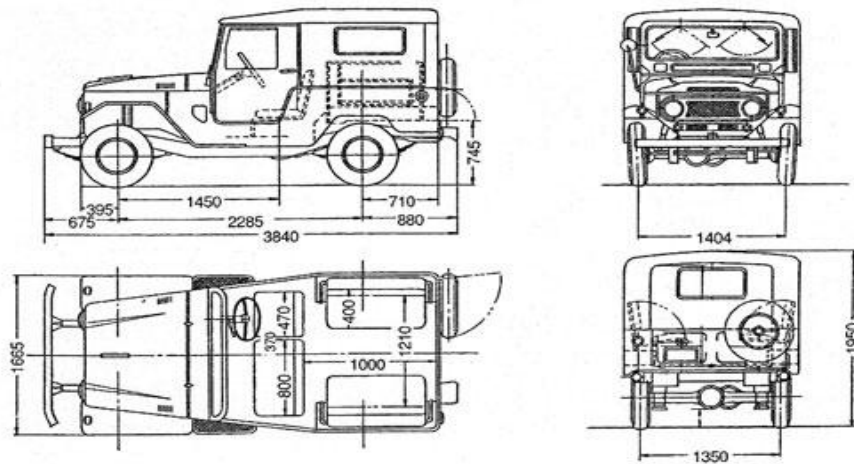
R_{merkez} : Virajın merkeze olan yarıçapı,

r : Tekerlek yarıçapı,

V : Aracın hızı,

L_a : Arka tekerlek ile ağırlık merkezi arasındaki mesafedir.

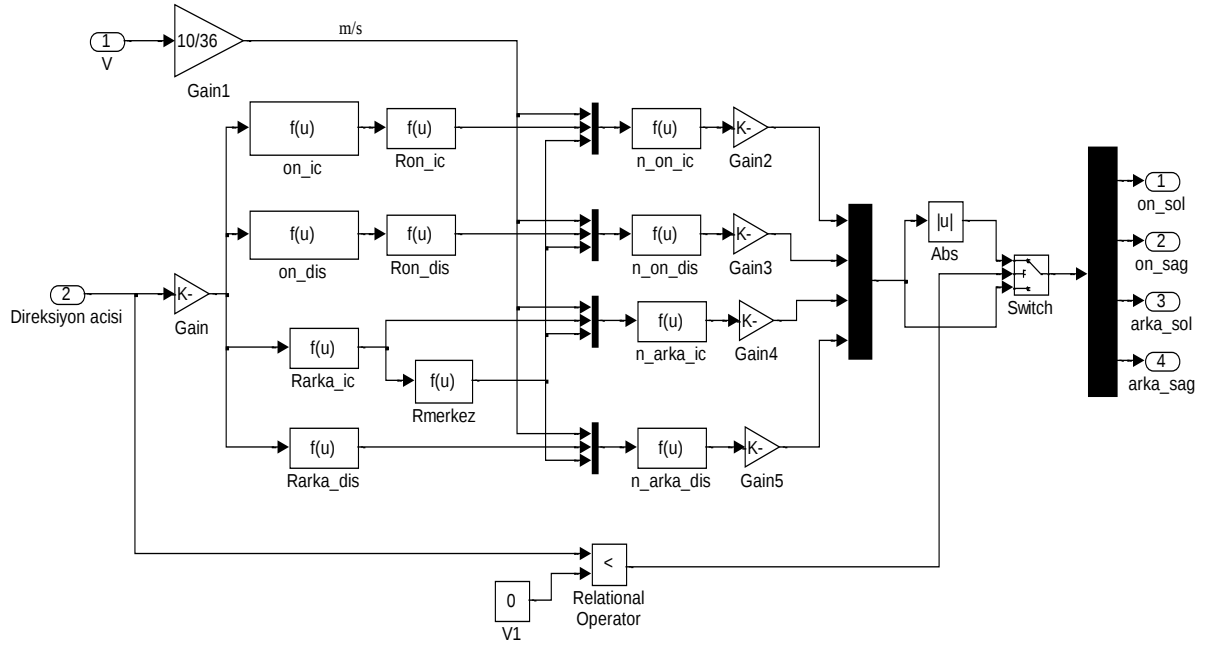
Oluşturulan yazılımda, L , L_a , d , r ve K değişkenleri, Şekil 4.2'de görülen araçtan alınarak programa sabit değer olarak girilmiştir.



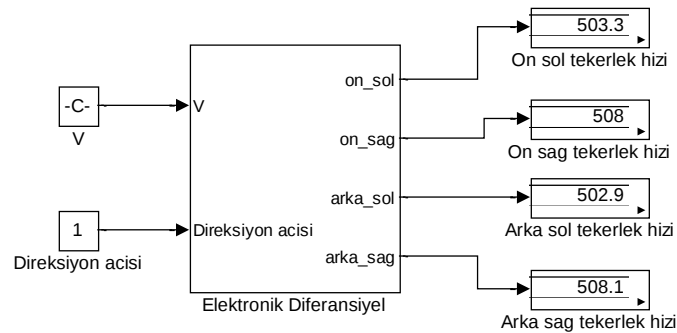
Şekil 4.2. Bir araçtan alınan boyutlar

4.3. Elektronik Diferansiyel Sistemin Modellenmesi

Direksiyon açısı ve araç hızının değişimine göre, ön tekerlek içi motorların hızlarını hesaplamak için Ackermann-Jeantand modelinden alınan denklemler kullanılarak Şekil 4.3'te görüldüğü gibi MATLAB/Simulink'te modellenmesi yapılmıştır. Örneğin; direksiyon açısı 1° ve araç 505.538 dev/dakika (200 km/saat) hızla giderken tekerlek içi motorların dönmesi gereken hız değerleri Şekil 4.4'te gösterilmiştir. Buna göre ön dış tekerlek 508 dev/dakika hızla giderken, ön iç tekerlek hızı 503.3 dev/dakika olmuştur. Böylece dıştaki tekerleğin hızı artarken, içteki tekerleğin hızı azalarak elektronik diferansiyel sistemin modellenmesi gerçekleştirilmiştir.



Şekil 4.3. Direksiyon açısı ve aracın hızına göre tekerlek içi motorların hızlarını hesaplamak için gerçekleştirilen Simulink model



Şekil 4.4. Elektronik diferansiyel sistemin Simulink modeli

4.4. CoDeSys Yazılımı

Bu tez çalışmasında elektronik diferansiyel sistemi tasarlarlarken, "Kontrollü Geliştirme Sistemleri" anlamına gelen "Controlled Development System" (CoDeSys) yazılımı kullanılmıştır. Bu yazılım özellikle "Programmable Logic Controller (PLC)" için geliştirilmiş olup, Structured Text (ST), Ladder Diagram (LD), Function Block Diagram (FBD) gibi pek çok PLC mantığı kullanılabilir [73]. CoDeSys, düzenleme, hata bulma ve hatayı düzeltme fonksiyonlarını kullanarak Visual C++ gibi ileri programlama dilleri üzerine kurulmuştur [74].

4.4.1. CoDeSys Fonksiyonları

4.4.1.1. CoDeSys'te Proje Oluşturma

"Program Organizasyon Birimi" anlamına gelen, yeni bir projede oluşturulmuş ilk "Program Organization Unit (POU)", otomatik olarak PLC_PRG olarak adlandırılır. Bir C programındaki ana fonksiyonla uyumlu işlem burada başlar ve diğer POU'lara aynı programlar, fonksiyon ve fonksiyon bloklarından ulaşılabilir. Oluşturulan tüm POU'lar PLC_PRG içerisinde çağırılır. Bir projede program organizasyon birimleri (POUs), data tipleri (data types), görüntü elemanları (visualizations), kaynaklar (resources) gibi çeşitli nesneleri içeren bir nesne düzenleyici (object organizer) yer almaktadır [74].

4.4.1.2. Projenin Test Edilmesi

Tüm hatalar giderildiği zaman simülasyon aktif edilir, PLC simülasyonuna giriş yapılır ve proje PLC'ye yüklenir. Böylece çevrimiçi olarak bağlanılmış olur. Daha sonra PLC konfigürasyon penceresi açılır ve doğru sıralama için proje test edilir. POU'lardaki yerel değişkenlerin değer sıralamasını da gözlemlemek mümkündür. İzleme ve alma yöneticisinde, incelenmek istenen değerlerin veri kayıtları yapılandırılabilir [74]. Çevrimiçi olduktan sonra Ctrl+F8 kısayol tuşuyla gerçek zamanlı olarak ilgili değerler görülebilir. Bu değerlere çift tıklanarak müdahale edilebilir ve yeni değer girilerek Ctrl+F7 ile yeni değer yüklenebilir [73].

4.4.1.3. Hata Onarma

Programda bir hata olduđu zaman kesme noktaları ayarlanabilir. Eğer bir kesmede işlem durursa, tüm proje değişkenlerinin değerleri incelenebilir, sırayla programın mantıksal doğruluđu kontrol edilebilir [74].

4.4.1.4. Ek Çevrimiçi Fonksiyonlar

İleri düzey hata onarma fonksiyonları mevcuttur. Belirli değerlerde program değişkenleri, girişler ve çıkışlar ayarlanabilir. Hangi program satırlarının çalıştığını kontrol etmek için akış kontrolü kullanılabilir. Hedef ayarları (Target Settings) aktif edilerek gerçek değişkenler izlenebilir. Ayrıca belirli bir hedef fonksiyon, PLC'den belirli bilgileri istemek için hizmet eden bir PLC alıcısıdır. Proje oluşturulduđu zaman donanıma yüklenebilir ve test edilebilir. Simülasyonda kullanıldığı gibi aynı çevrimiçi fonksiyonlar yer alacaktır [74].

4.4.2. Proje Bileşenleri

4.4.2.1. Proje (Project)

Bir proje, PLC programdaki nesnelerin tümünü içerir. Proje, adlandırılan bir dosyanın içine kaydedilir. Program organizasyon birimleri, data tipleri, görüntülemeler, kaynaklar ve kütüphaneler gibi nesneler bir projenin içerisinde yer almaktadır [74].

4.4.2.2. Program Organizasyon Birimi (POU)

Fonksiyonlar, fonksiyon blokları ve programlar POU'ları oluşturur. Her POU, bir açıklama ve gövde bölümü içerir. Gövde bölümü, IL, ST, SFC, FBD, LD veya CFC gibi IEC programlama dillerinden birinde yazılır. CoDeSys tüm IEC standart POU'ları destekler. Projede bu POU'lar kullanılmak istenirse, proje standard.lib kütüphanesini içermelidir [74].

4.4.2.3. Fonksiyon (Function)

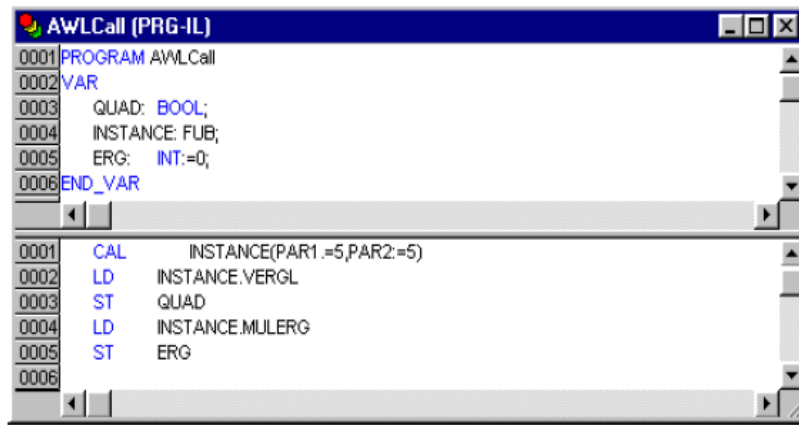
Fonksiyonlar, alanlar ve yapılar gibi birkaç elemandan oluşan bir POU'dur. Bir fonksiyonun isminden sonra iki nokta konularak veri tipi yazılır. Aşağıdaki örnekle fonksiyonun kullanımı gösterilmiştir:

FUNCTION Fct : INT

Daha sonra fonksiyona bir sonuç atanmalıdır. Yani fonksiyon ismi çıkış değişkeni olarak kullanılır. Bir fonksiyon açıklaması "FUNCTION" kelimesiyle başlar. IL programlama dilinde bir fonksiyon, aşama işlemler içerisinde veya bir basamağın içinde oluşturulabilir. ST dilinde bir fonksiyon, bir ifadede işlenen olarak kullanılabilir. SFC dilinde ise fonksiyon; sadece bir basamak ya da geçiş içerisinde yer alabilir.

4.4.2.4. Fonksiyon Bloğu (Function Block)

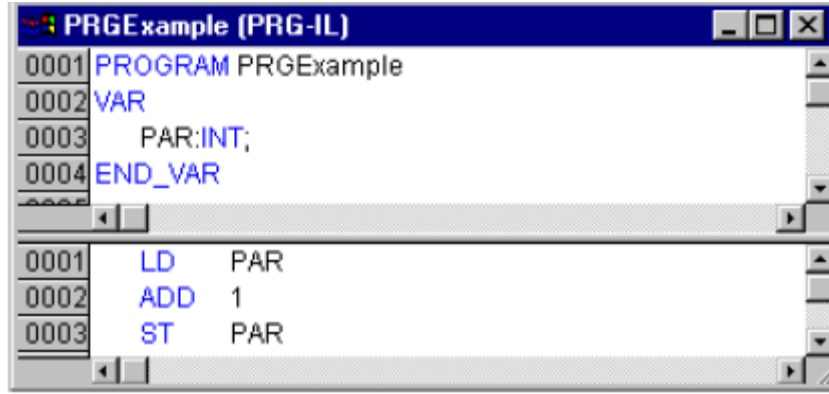
Fonksiyon bloğu (Function Block, FUB), işlem boyunca bir veya daha fazla değer sağlayan bir POU'dur. Fonksiyona göre, fonksiyon bloğu dönüş değeri sağlamaz. Fonksiyon bloğunun açıklama satırı "FUNCTION_BLOCK" ile başlar. Fonksiyon bloklarının ve programların açıklama kısımları örnek açıklamalarını içerebilir. Fonksiyonlarda ise; örnek açıklamalarına izin verilmez. Bir fonksiyon bloğu örneğinin ismi bir fonksiyon ya da fonksiyon bloğu için giriş olarak kullanılabilir. FUB çağırma ile ilgili "Örnek (Instance)" isimli IL dilinde bir FUB örneği Şekil 4.5'te gösterilmiştir. Çoğalma sonucu ERG değişkeni içerisine, kıyaslama sonucu da QUAD içine kaydedilir.



Şekil 4.5. IL dilinde bir fonksiyon bloğu örneği [74].

4.4.2.5. Program

Bir program çalışırken birkaç değer döndüren bir POU'dur. Programlar proje boyunca tanınır. Tüm değerler program sonradan çalıştırılacağı zamana kadar korunur. Programlar çağırılabilir ve bir fonksiyondaki bir program çağırısına izin verilmez [74]. IL dilinde bir program örneği Şekil 4.6'da gösterilmiştir:



Şekil 4.6. Bir program örneği [74].

Program açıklaması "PROGRAM" kelimesiyle başlar ve "END_PROGRAM" ile biter. Programı çağırırken giriş/çıkış parametreleri ayarlanmak istenirse parantez içinde programın isminden sonra parametrelere değerler atanarak IL ve ST dillerinde yapılabilir. (Giriş parametreleri ":@" ile çıkış parametreleri de "=>" ile atanır.)

4.4.2.6. PLC_PRG

PLC_PRG, önceden tanımlanmış bir POU'dur. Her proje bu programı içermelidir. Bu POU her kontrol döngüsünde bir kez çağırılır. Yeni bir proje oluşturulduktan sonra "Project-Object-Add" seçilerek program "PLC_PRG" olarak isimlendirilir [74].

4.4.3. Kaynaklar (Resources)

Projeyi konfigüre etmek ve oluşturmak için kaynaklara ihtiyaç vardır. Global değişkenler (Global Variables) proje veya ağ boyunca kullanılır. Projeye kütüphane eklemek için kütüphane yöneticisi "Library Manager", çevrimiçi hatta bağlanmak için "Log in", projede alarm konfigürasyonu için "Alarm Configuration", donanımı konfigüre etmek için "PLC Configuration", çalışmalar süresince programa yol göstermede "Task Configuration", değişken değerlerini görüntüleme ve hata değişken değerlerini ayarlama, izleme ve alma yöneticisi "Watch and Receipt Manager", hedef sistemin son bir konfigürasyonu gerekliyse ve seçimi için hedef ayarları "Target Settings", proje seçeneklerinin görüntüsü için ise çalışma sayfası olan "Workspace" kullanılır [74].

4.4.4. Kütüphaneler (Libraries)

Kullanıcı tarafından tanımlanan POU'lar, data tipleri ve global değişkenlerin kütüphaneleri projede kullanılabilir. Standard.lib ve util.lib kütüphaneleri programları oluşturan başlıca kütüphanelerdir [74].

4.4.5. Veri Tipleri (Data Types)

Standart veri tipleriyle kullanıcı kendi veri tiplerini tanımlayabilir; yapılar, sayım tipleri ve referanslar oluşturulabilirler [74].

4.4.6. Görüntüleme (Visualization)

Proje değerlerini görüntüleyebilmek için CoDeSys görüntüleme (visualization) seçeneği sunar. Görselleştirme yardımıyla geometrik elemanları çizilebilir, renkleri, metinleri, biçimleri değiştirilebilir. Görüntüleme, PLC için çalışma ara yüzü olarak kullanılabilir [74].

4.4.7. CoDeSys Program Dilleri

CoDeSys, IEC-61131 standardıyla tüm dilleri destekler [74].

4.4.7.1. Metin Dilleri

Instruction List (IL) ve Structured Text (ST) metin dillerini oluşturur.

4.4.7.2. Grafiksel Diller

Sequential Function Chart (SFC), Function Block Diagram (FBD) ve Ladder Diagram (LD) dilleri bu gruba girer. Bunlardan ayrı FBD'ye bağlı grafik Continuous Function Chart Editor (CFC) programlama dili de mevcuttur.

▪ Instruction List (IL)

Instruction List (IL), "Açıklama Listesi" anlamına gelir ve açıklamaları içerir. Her açıklama, yeni bir satır ve operatörlerden oluşur. Çalışmanın tipine göre bir veya daha fazla işlenen virgülle ayrılır. Bir açıklamanın önünde ":" ile belirtilmiş bir etiket olmalı ve açıklama satırın en sonunda yer almalıdır. Açıklamaların arasına boş satırlar koyulabilir [74]. Bir IL örneği;

LD 17

ST lint (*yorum*)

GE 5

JMPC next

LD idword

EQ istruct.sdword

STN test

next:

▪ **Structured Text (ST)**

"Yapısal Metin" anlamına gelen Structured Text (ST), "IF..THEN..ELSE" yapılarını veya "WHILE..DO" döngülerini içerir [74]. Bir ST örneği;

IF deger<7 THEN

WHILE deger<8 DO

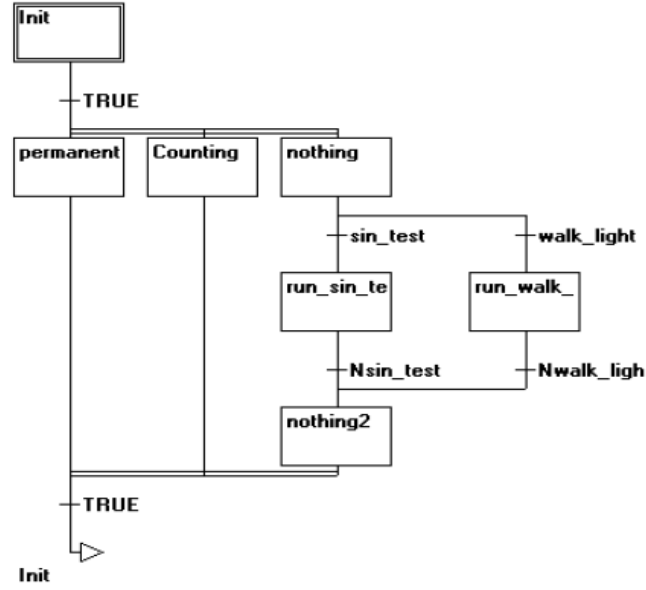
deger:=deger+1;

END_WHILE;

END_IF;

▪ **Sequential Function Chart (SFC)**

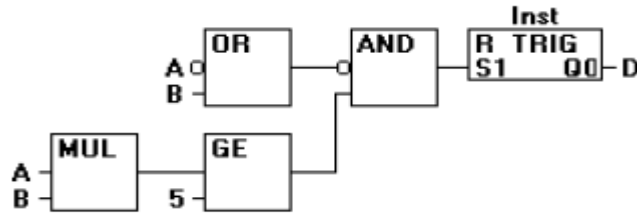
"Sıralı Fonksiyon Tablosu" anlamına gelen Sequential Function Chart (SFC), grafiksel bir dildir. Bir program içerisinde farklı olayların kronolojik sıralamasını tanımlama imkanı verir. Bunun için, olaylara basamaklarla elemanlar atanır ve işlem karesi aktarım elemanlarıyla kontrol edilir [74]. Şekil 4.7'de verilen bir SFC örneği;



Şekil 4.7. SFC'de bir ağ örneği [74].

▪ Function Block Diagram (FBD)

"Fonksiyon Blok Diyagramı" anlamına gelen Function Block Diagram (FBD), grafiksel bir programlama dilidir. Her ağı lojiksel yapı, aritmetik ifade ya da fonksiyon bloğu içerir [74]. Şekil 4.8'de bir FBD örneği gösterilmiştir:



Şekil 4.8. FBD örneği [74].

▪ The Continuous Function Chart Editor (CFC)

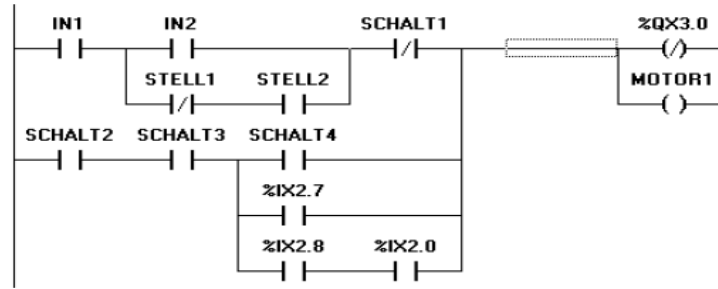
"Sürekli Fonksiyon Grafik Editörü" anlamına gelen The Continuous Function Chart Editor (CFC), FBD'ler gibi ağlarla çalışmazlar. Serbest olarak yerleşirler. Bu da geri beslemeye izin verir. Şekil 4.9'da gösterilen bir CFC örneği:



Şekil 4.9. CFC örneği [74].

▪ Ladder Diagram (LD)

"Basamak Diagramı" anlamına gelen Ladder Diagram (LD), bir elektriksel devrenin yapısıyla alakalı grafiksel bir programlama dilidir. LD lojik anahtarlar için uygunken, diğer taraftan FBD'deki ağlar da oluşturulabilir. Bu yüzden LD, diğer POU'ları çağırırken kontrol etme aşamasında oldukça kullanışlıdır. LD ağ serilerini içermektedir. Bir ağ, sağ ve sol düşey akım hattı ile sağ ve sol tarafta sınırlandırılır. Orta kısım ise bağlantılar, sargılar ve bağlantı hatlarından oluşur. Her ağ, Boolean değerleri TRUE ya da FALSE ile karşılaşan soldan sağa ON veya OFF şartını sağlayan bağlantı serilerinin sol tarafında oluşur. Eğer bu değişken TRUE ise durum bağlantı hattı boyunca soldan sağa geçer. Aksi durumda ise, sağ bağlantı OFF değerini alır [74]. Bağlantılar ve sargılardan oluşan LD'de bir ağ örneği Şekil 4.10'da gösterilmiştir:



Şekil 4.10. Bağlantılar ve sargılardan oluşan LD'de bir ağ örneği [74].

4.4.8. Program Değişkenlerini Gözlemleme

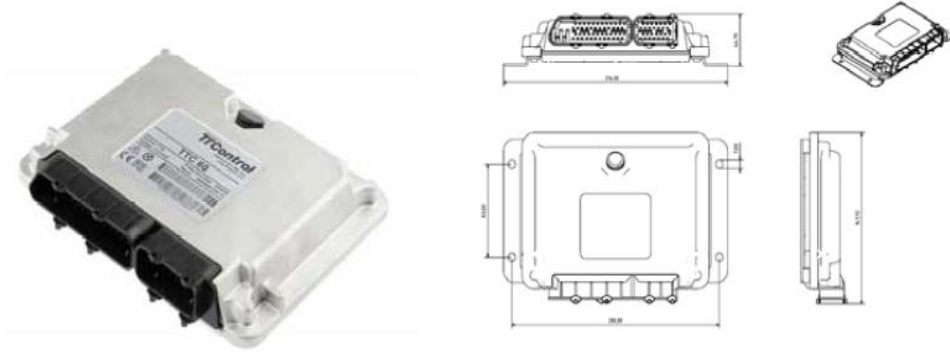
Çevrimiçi olduğunda tüm görüntülenebilir değişkenler kontrolörden okunur ve gerçek zamanlı olarak görüntülenir. Bildirimlerde ve program editöründe bu görüntü yer alacaktır ve görüntüleme, izleme ve alındı bilgisindeki değişkenlerin akım değerleri buradan okunabilir. VAR_IN_OUT değişkenlerini gözlemlerken referans değer çıkıştır. Hem gösterilen değer hem de referans değer açıklama ekranında çıkış olarak görülür. Program bölümünde sadece işaret edilen değer çıkıştır [74]. Bu değerler Şekil 4.11'de görülmektedir:

Tablo 4.1. Integer veri tipleri

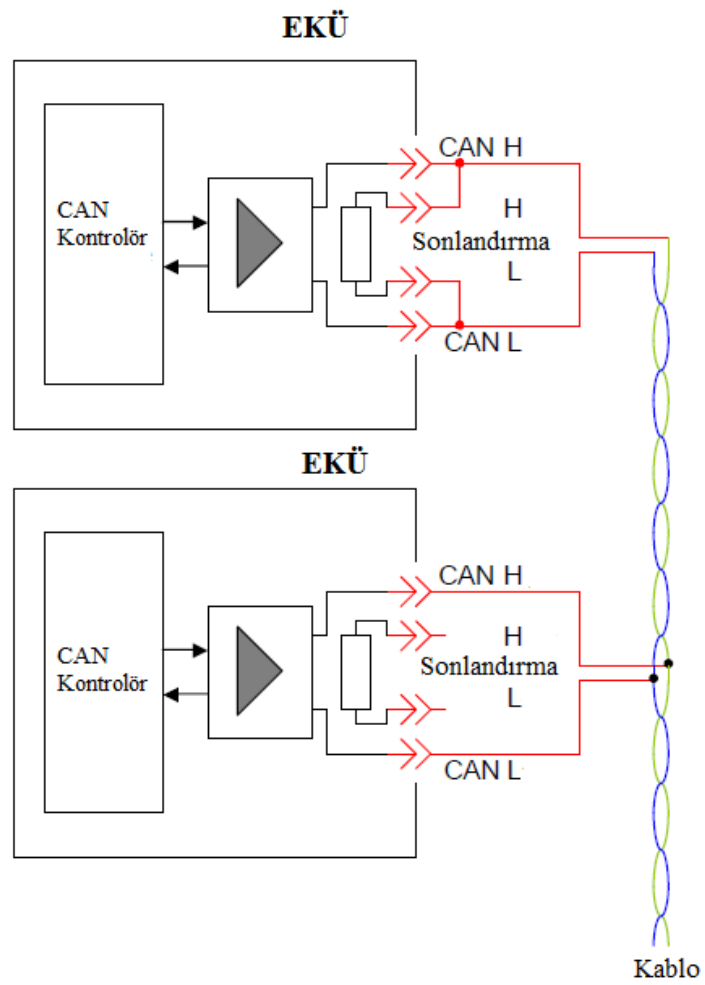
Veri Tipi	Alt limiti	Üst Limiti	Hafıza Alanı
BYTE	0	255	8 Bit
WORD	0	65535	16 Bit
DWORD	0	4294967295	32 Bit
SINT	-128	127	8 Bit
USINT	0	255	8 Bit
INT	-32768	32767	16 Bit
UINT	0	65535	16 Bit
DINT	-2147483648	2147483647	32 Bit
UDINT	0	4294967295	32 Bit

4.5. HY-TTC60

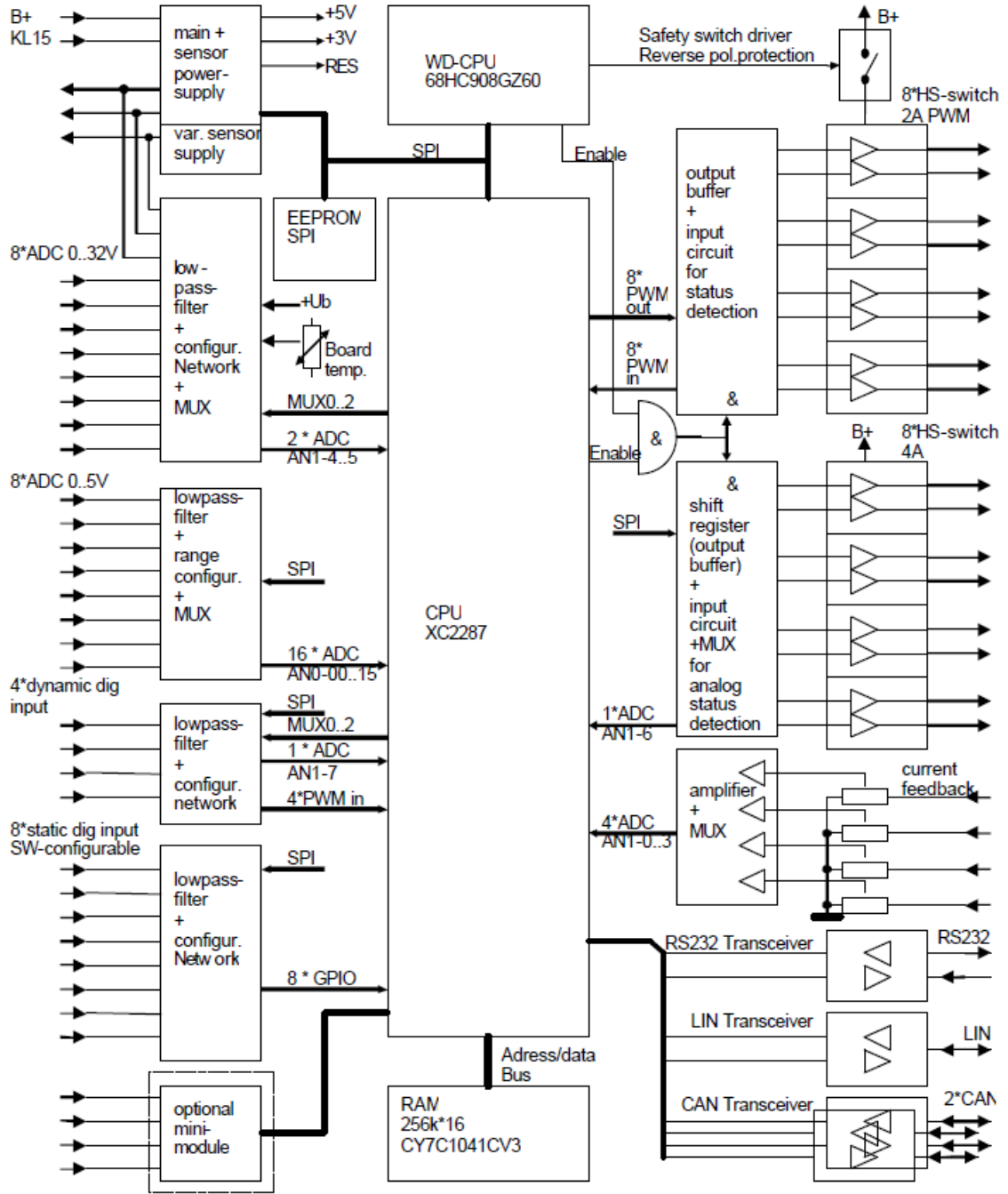
TTControl, TTTech ve HYDAC Uluslararası iki şirketin ortaklığıyla oluşmuş, çeşitli alanlarda kullanılmak üzere operatör ara yüzleri ve elektronik kontrol sistemleri geliştiren bir firmadır [75]. Bu çalışmada kullanılan HY-TTC 60 ise, TTControl firmasının ürettiği 16 bitlik kontrolör ailesinin gelişmiş bir modeli olup, güç ve kontrol kartlarından oluşmuş bir işlemcidir. Temel versiyonlarına kıyasla ek giriş işlevleri sunar. Hem tek çözüm standı olarak hem de modern makinelerdeki ağ sisteminin bir parçası olarak kullanılabilen güçlü bir cihazdır. 2 CAN, 1 RS-232 ve 1 LIN ara yüzü ile seri iletişime sahiptir. HY-TTC 60 ürün ailesiyle uyumludur. CoDeSys 2.3 veya C/C++ ile programlanır. 594 kb RAM, 48 giriş ve çıkış, 16 güç çıkışları, 4 akım ölçüm girişleri, 8 analog girişler (akım/gerilim), 8 analog girişler (gerilim/yapılandırılabilir) bulunur. Tüm giriş ve çıkışlar yapılandırılabilir, gerilim yükselmeleri ve kısa devrelere karşı korunabilirler. Dahili bir monitörle ayarlanabilir sensör gerilim kaynağı mevcuttur. Motor çalıştığında gerilimdeki eğimden kaynaklanan dalgalanma yoktur. 80 kutuplu su geçirmez bağlantı ile sağlam alüminyum gövde ve su geçirmez zar ile basınç dengeleme gibi özelliklere sahiptir. Şekil 4.12'de HY-TTC 60 modülü, Şekil 4.13'te CAN ara yüzü ve Şekil 4.14'te ise blok diyagramı görülmektedir:



Şekil 4.12. HY-TTC 60 modülü [76].



Şekil 4.13. CAN ara yüzü [77].



Şekil 4.14. HY-TTC 60'ın blok diyagramı [78].

HY-TTC 60 işlemci modülün besleme uçları ve CAN hattına bağlantıları Tablo 4.2'de görüldüğü gibi yapılır. CAN 0 ve CAN 1 olmak üzere iki tane CAN hattı mevcuttur. Bu projede CAN 0 kullanılmıştır. HY-TTC 60'ı CAN 0'a bağlarken CAN L ve CAN H pinleri 273 ve 279, hat başı ve hat sonu dirençleri olan 280 ve 272 ile kısa devre edilir. Böylece dirençler devreye alınarak PCAN-USB kablusunun CAN H ve CAN L uçlarına bağlantısı yapılır.

Tablo 4.2. HY-TTC 60 bağlantı pinleri.

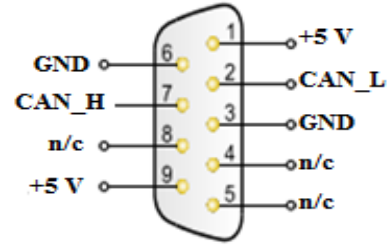
	PİN	GÖREVİ	AÇIKLAMA
CAN 0	273	CAN L (Siyah)	Hat başı olduğu için 280 ile kısa devre edilir.
CAN 0	279	CAN H (Kırmızı)	Hat sonu olduğu için 272 ile kısa devre edilir.
CAN 0	272	Sonlandırma direnci	
CAN 0	280	Sonlandırma direnci	
CAN 1	265	CAN H (Siyah)	Hat sonu olduğu için 258 ile kısa devre edilir.
CAN 1	259	CAN L (Kırmızı)	Hat başı olduğu için 266 ile kısa devre edilir.
CAN 1	258	Sonlandırma direnci	
CAN 1	266	Sonlandırma direnci	
Besleme	101	24 V Besleme	
Besleme	127	24 V Besleme	
Besleme	128	24 V Besleme	
Besleme	102	GND	
Besleme	271	24 V Besleme	

4.6. PCAN-USB

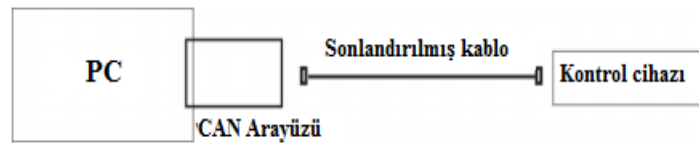
Şekil 4.15'de görülen PCAN-USB kablosu, usb girişi için CAN ara yüzü sağlar. Şekil 4.17'de görüldüğü gibi bir ucu bilgisayarın usb girişine bağlanırken, çıkışı ise CAN-H ve CAN-L uçlarına sahip CAN-BUS hattını oluşturur. Şekil 4.16'da pin tanımlamaları gösterilmiştir. CoDeSys'ten hesaplanan ön tekerlek hızları PCAN-USB kablosuyla motor sürücülerine gönderilir ve motorların istenilen hızda dönmesi sağlanır.



Şekil 4.15. PCAN-USB



Şekil 4.16. PCAN-USB pinleri [79].

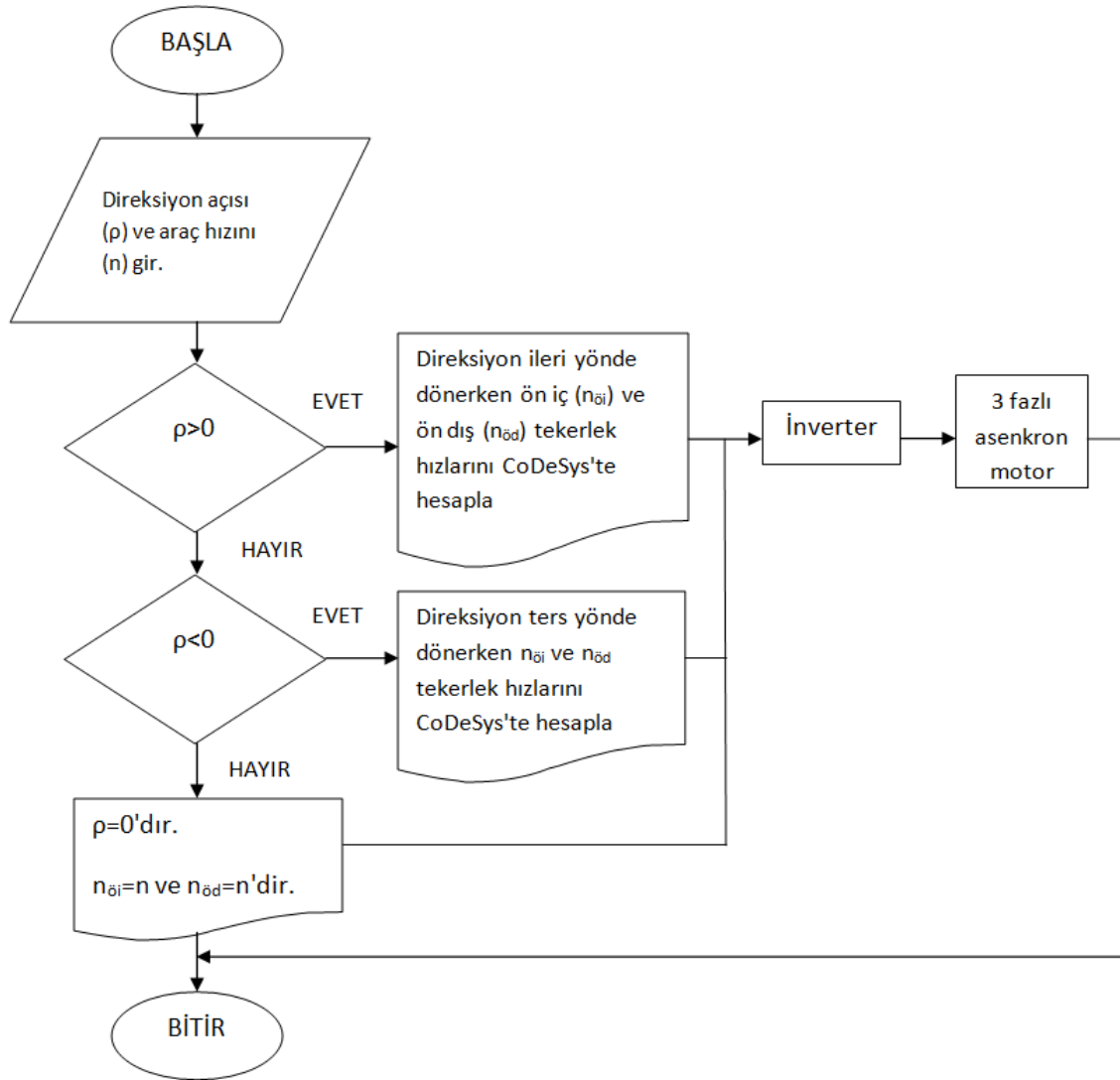


Şekil 4.17. PCAN-USB bağlantı şeması [79].

PCAN-USB ile 5 kbit/s'den 1 Mbit/s'ye kadar yüksek hızlarda veri iletimi yapılabilmektedir. CAN mesajları, PCAN-View programı üzerinden alınır ve gönderilir.

4.7. Elektronik Diferansiyelin CoDeSys Yazılımı ile Gerçekleştirilmesi

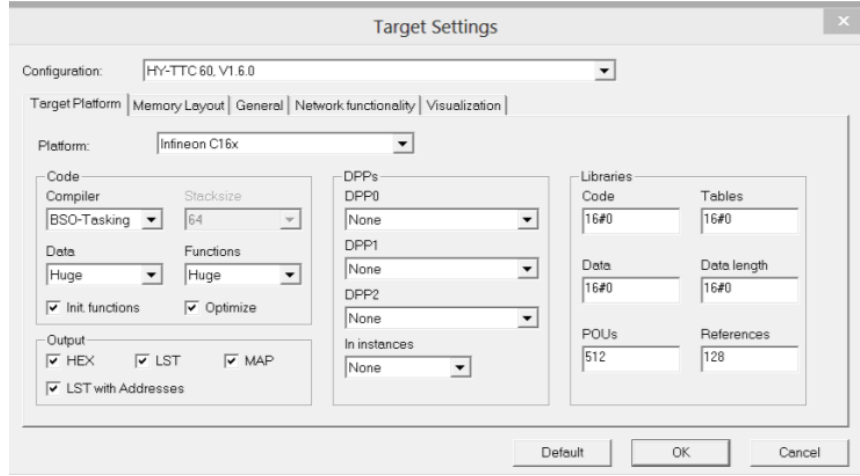
Ackermann-Jeantand Modeli'nden alınan matematiksel denklemler CoDeSys'e atılarak elektronik diferansiyel için gerekli yazılım gerçekleştirilmiştir. Tasarlanan sistemin akış diyagramı Şekil 4.18'de görülmektedir. Buna göre, direksiyon açısı ve aracın hız bilgileri alınarak direksiyonun dönüş yönüne göre ön tekerlek hızları hesaplanmış ve asenkron motor sürücülerine gönderilmiştir. Direksiyon açısı sıfırdan büyükse direksiyon ileri yönde, küçük ise ters yönde dönecektir. Sıfır olduğu durumda da araç viraja girmeyeceğinden tekerlekler aynı hızda dönmeye devam edecektir.



Şekil 4.18. CoDeSys'te gerçekleştirilen elektronik diferansiyel sistemin akış diyagramı

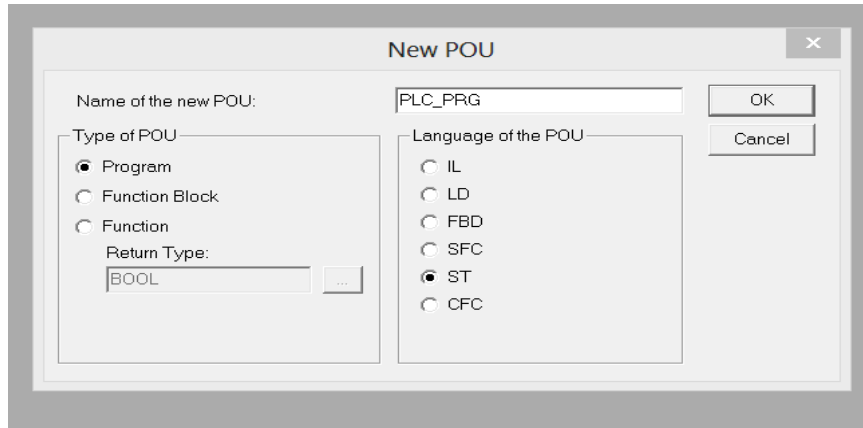
CoDeSys, program organizasyon birimi, veri tipleri, görüntülemeler ve kaynaklar olmak üzere dört ana bölümden oluşur. Programda öncelikle yeni bir proje oluşturulur. Bunun

için File'dan "New" seçilir. Hedef ayarlarından (Target Settings) konfigürasyonu HY-TTC 60,V1.6.0 olarak ayarlanır. Bu durum Şekil 4.19'da gösterilmiştir:



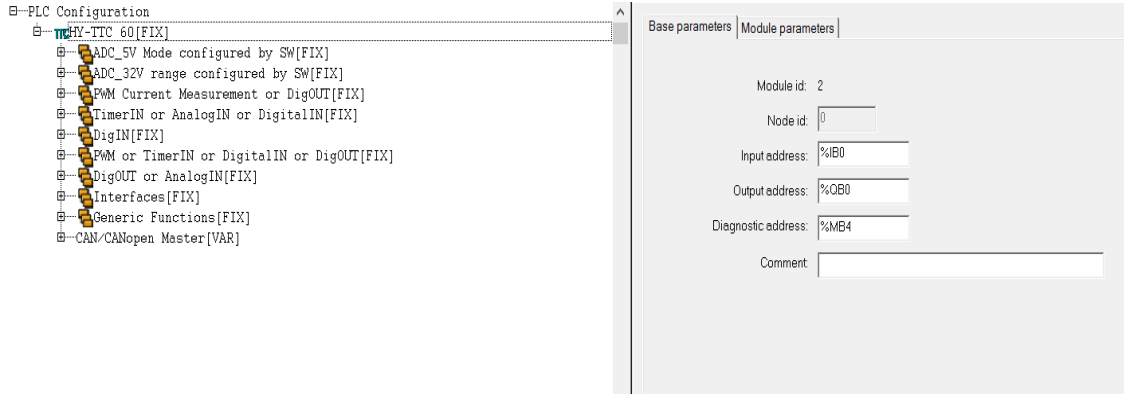
Şekil 4.19. Hedef ayarları (Target Settings)

Bundan sonra çıkan tüm kütüphaneler seçilerek Şekil 4.20'de görülen "New POU"dan yeni bir program penceresi açılır. Programlama şekli ve program dili tanımlanarak PLC program (PLC_PRG) oluşturulur. Bu program içerisinde oluşturulan tüm programlar çağırılarak çalıştırılabilir.



Şekil 4.20. Yeni bir POU oluşturma

"PLC Configuration" kısmından konfigürasyon ayarlamaları yapılır. CAN uyumlu sensörler ya da aygıtlar kullanıldığı zaman elektronik veri dosyaları (electronic data-sheet, eds) eklenerek o aygıta ait veriler otomatik olarak yüklenebilir. Bunun için PLC Configuration'da HY-TTC 60'a sağ tıklayarak "Append Subelement CAN/CANopen Master" seçilir. Böylece HY-TTC 60 işlemcisinin eds dosyası yüklenmiş olur. HY-TTC 60'ın üzerine tıklanarak modül parametre ayarları Şekil 4.21'de gösterildiği gibi yapılabilir:

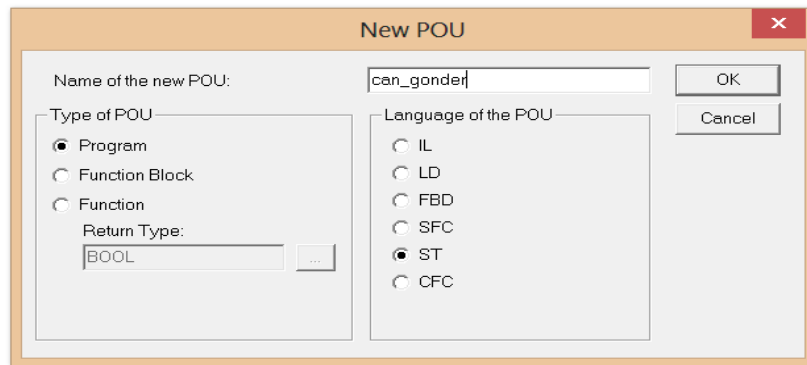


Şekil 4.21. PLC konfigürasyon ayarları

Bu projede direksiyon açısı bilgisini almak için bir enkoder ve aracın hız bilgisi için ise mesafe ölçer sensörü kullanılmıştır. Mesafe ölçer, analog bir sensör olup, çıkışı 4-20 mA'lık akım bilgisi verir. PLC konfigürasyonda görülen (ADC_5V) 8 tane analog girişten istenilen pine bağlanarak bir isim verilir. Bu isim, CoDeSys'te global değişken olarak tanımlanmış olur. PLC programının içerisinde bu isimle çağrılarak mesafe ölçer sensöründen akım bilgisi okunabilir. Direksiyon açısı bilgisi için kullanılan enkoder ise CAN uyumlu bir sensördür ve CAN hattından veri almak için gerçekleştirilen programda tanımlamaları yapılmıştır.

4.7.1. CAN Hattından Veri Alma Programı

Kullanılan cihazların eds dosyaları mevcut değilse CoDeSys'te CAN-BUS hattından veri alıp gönderebilmek için can veri alma ve can veri gönderme programları oluşturulmalıdır. Bunun için "POUs" bölümüne sağ tıklayarak "Add Object" ile "New Pou" seçilir. POU'nun tipi "program", dili de "ST" seçilerek istenilen program Şekil 4.22'de görüldüğü gibi oluşturulmuştur. Eds dosyası olması halinde bu programa gerek olmadan PLC konfigürasyon ayarları yapılarak haberleşmesi sağlanabilir.



Şekil 4.22. CAN alma/CAN gönderme programı oluşturma

Bu programda öncelikle veri almak istenilen ağıta ait ID numarası, veri uzunluğu ve CAN hattının formatı gibi bilgiler girilerek, her byte'tan istenilen bilginin okunması için aşağıda görüldüğü gibi bir tampon (buffer) oluşturulmuştur. Burada ID numarası 190 H olan enkoderden açılı bilgisi alınacağı için can alma programında gerekli tanımlamaları yapılmıştır.

```
alici_frame.id := 16#190;
```

```
alici_frame.id_format:= 0 ;
```

```
alici_frame.length:= 8 ;
```

```
IF xc_can_read ( alici_buffer , ADR (alici_frame), 1, ADR (frames_read)) = IO_E_OK
```

```
THEN
```

```
byte0:= alici_frame.data[0];
```

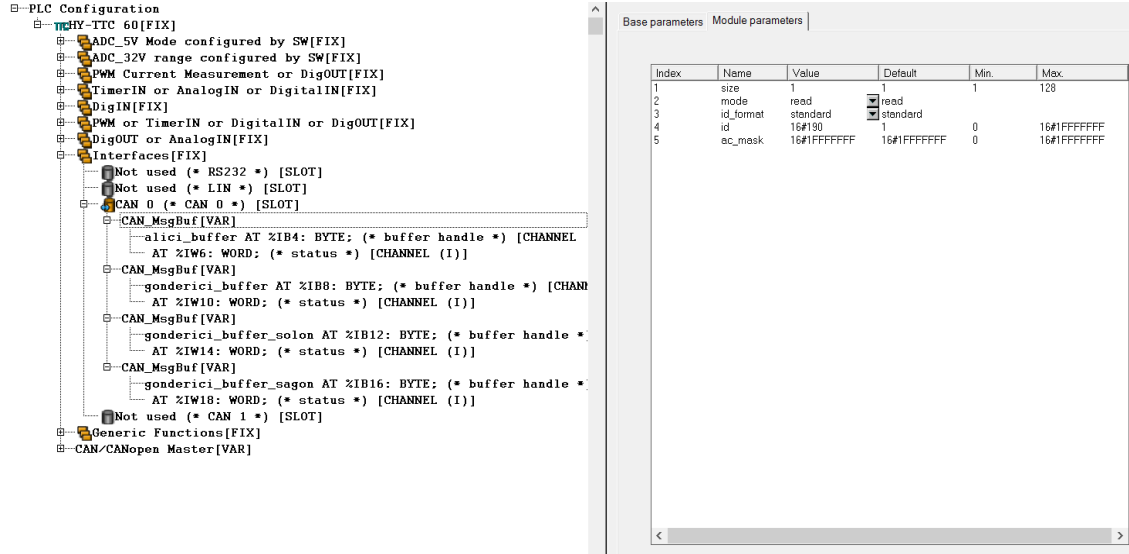
```
byte1:= alici_frame.data[1];
```

```
byte2:= alici_frame.data[2];
```

```
byte3:= alici_frame.data[3];
```

```
END_IF
```

Daha sonra alıcı tamponun PLC konfigürasyonda tanımlanması gerekir. Bunun için Şekil 4.23'te gösterildiği gibi "Interfaces" seçeneğinden sırasıyla CAN 0, Replace Element ve CAN 0'a sağ tıklanarak CAN 0 ara yüzü oluşturulur. CAN 0'a sağ tıklanarak "Append CAN_MsgBuf" seçeneğinden CAN_MsgBuf oluşturularak modül parametreleri girilir.



Şekil 4.23. PLC konfigürasyonunda alıcı tamponun tanımlanması

4.7.2. CAN Hattından Veri Gönderme Programı

CAN hattından veri alma programından sonra veri gönderme programı oluşturularak istenilen veriler gönderilebilir. Aynı şekilde "Add Object"ten ST dilinde bir "New POU" seçilir. Bu programda, can alma programından okunan açısal hız bilgileri gönderici çerçevesinin istenilen byte'ından gönderilebilir. Gönderici çerçevesinin ID bilgisi, CAN formatı ve veri uzunluğu bilgileri tanımlanmıştır. Aşağıda gösterildiği gibi, gönderici çerçevenin ilk dört byte'ından enkoderden alınan hız bilgisi, mesafe ölçer sensöründen alınan akım bilgisi 4. byte'tan, 5. byte'tan "real" data tipinden byte'a çevrilmiş ön iç tekerlek hız bilgisi, 6. ve 7. byte'lardan ise "int" veri tipinden byte'a çevrilmiş ön tekerleklerin açısal hız bilgileri gönderilmiştir.

```
gonderici_frame.id := 16#150;

gonderici_frame.id_format:= 0 ;

gonderici_frame.length:= 8 ;

gonderici_frame.data[0]:= byte0;

gonderici_frame.data[1]:= byte1;

gonderici_frame.data[2]:= byte2;

gonderici_frame.data[3]:= byte3;
```

```
gonderici_frame.data[4]:= WORD_TO_BYTE(akim_girisi);
```

```
gonderici_frame.data[5]:= REAL_TO_BYTE(n_oi);
```

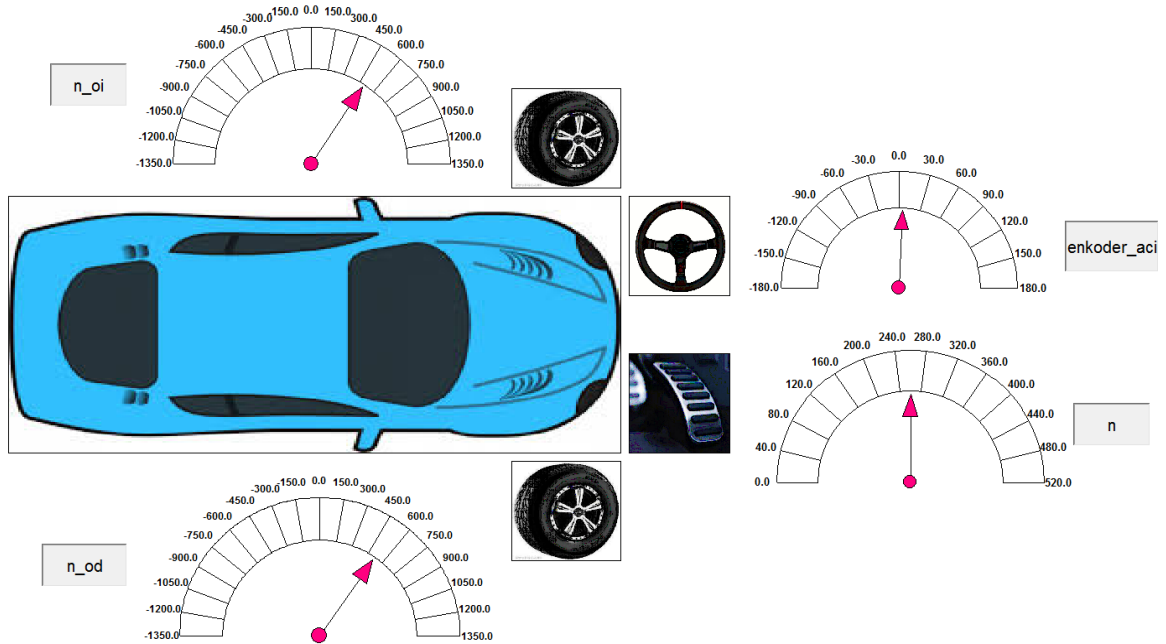
```
gonderici_frame.data[6]:= INT_TO_BYTE(motor_hizi_oi);
```

```
gonderici_frame.data[7]:= INT_TO_BYTE(motor_hizi_od);
```

Daha sonra, PLC konfigürasyonda, CAN 0 hattından bir gönderici tampon oluşturularak modül parametreleri girilmiştir. Oluşturulan bu programlar PLC_PRG'da çağırılmalıdır. Aksi halde çalışmazlar. CAN hattına gönderilen veriler global değişkenler olarak tanımlanır. CoDeSys'in içerisinde değişkenleri her yerde kullanmak istiyorsak burada tanımlamaları yapılmalıdır.

4.8. Alınan ve Gönderilen Mesajları Görüntüleme

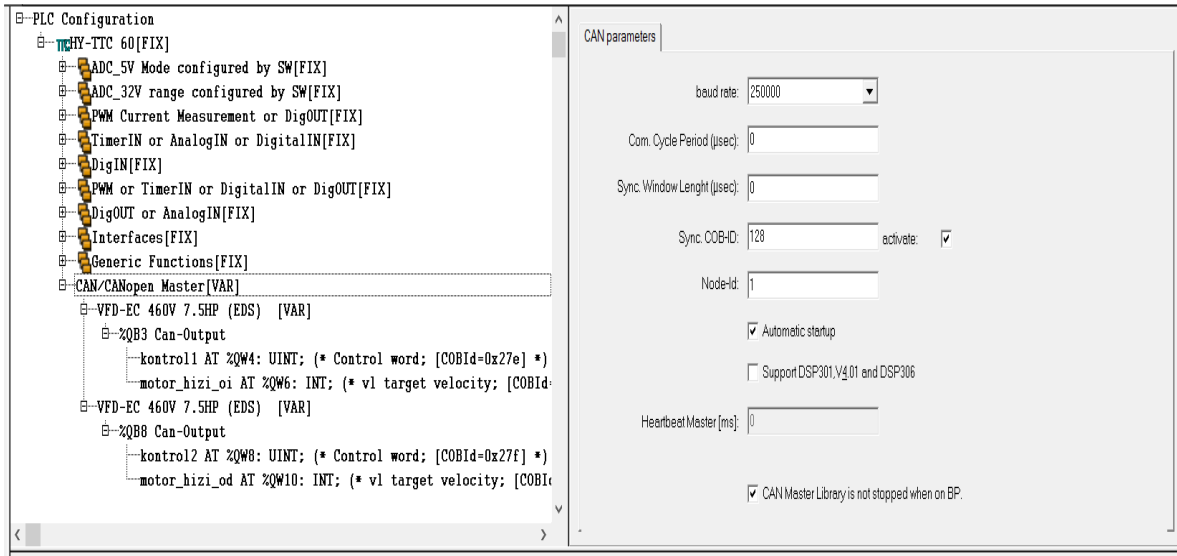
Şekil 4.24'te görüldüğü gibi görsel ara yüz olan "Visualizations" bölümünde, gerçekleştirilen yazılımdan alınan bilgiler resimlerin içerisine atılarak bu değerlerin değişimi göstergelerden okunabilmektedir. Burada, enkoder açı bilgisi direksiyondan, mesafe ölçerden alınan akım bilgisi gaz pedalından, ön tekerlek içi motorların açısal hız bilgileri ise tekerleklerden görüntülenmiştir.



Şekil 4.24. Motor kontrolü için CoDeSys'te gerçekleştirilen simülasyon.

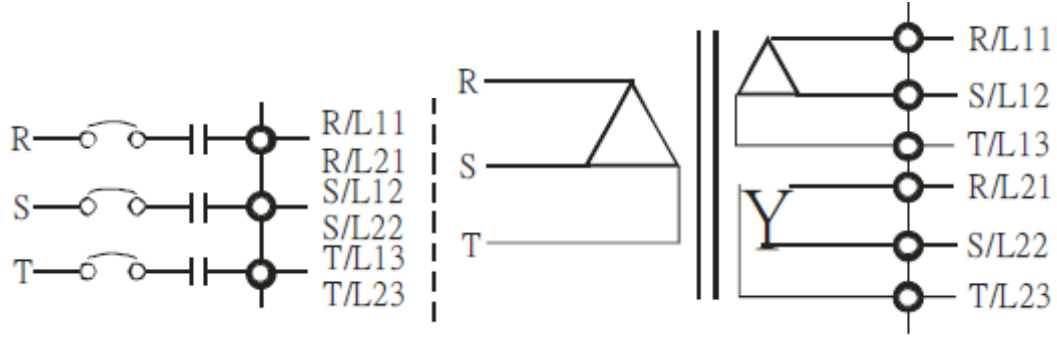
4.9. Deney Düzenegi

Ön tekerleklerin hesaplanan hız değerlerini, tekerlek içerisine yerleşecek motorlara göndermek için iki adet asenkron motor ve sürücüsü kullanılmıştır. Sürücüler CAN uyumlu olduğundan eds dosyaları, menüden "extras" ve "add configuration file" seçeneğiyle yazılıma alınmıştır. Daha sonra CAN/CANopen Master'a sağ tıklanarak Şekil 4.25'te gösterildiği şekilde "Append Subelement'ten" kullanılan sürücünün eds dosyası yüklenmiştir.



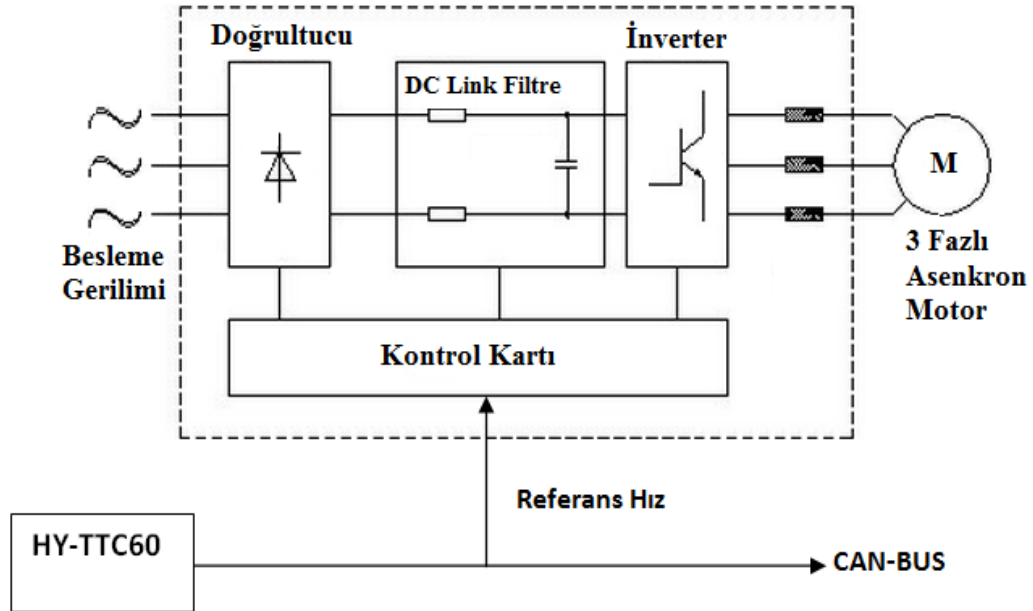
Şekil 4.25. Sürücünün "eds" dosyasını yükleme

Şekil 4.26'da bu eds dosyasının üzerine gelerek CAN parametre ayarları, SDO ve PDO ayarları yapılmıştır. Alınan ve gönderilen mesajların adres atamaları yapılarak CAN-BUS'tan gönderilmek istenen hız bilgisi seçilmiştir. Alınan işlem veri nesnesinden (Receive PDO) motor sürücüsüne gönderilmek istenen hız bilgisi seçilir. Böylece (200+Sürücü ID numarası) H adresinden sürücü hız bilgisini alır. SDO bölümünde ise istenilen hız bilgisi girilir. Bu değere başlangıçta 0 verilerek CAN alma programından hesaplanan tekerlek hız bilgileri bu adrese gönderilmiştir.

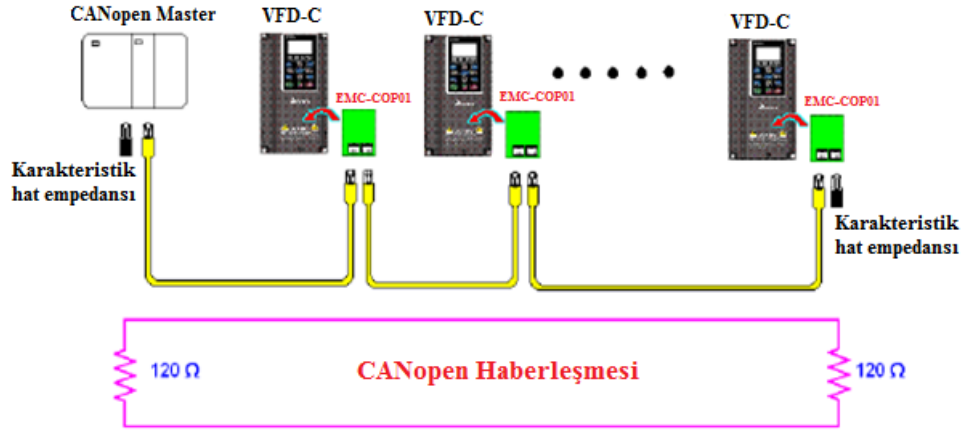


Şekil 4.28. Motorun sürücüye bağlanma şeması [80].

Şekil 4.29'da bir motor sürücüsünün CAN-BUS hattına bağlantı blok diyagramı görülmektedir. Tasarlanan sistemde, CAN hattına iki adet asenkron motor sürücüsü bağlanmıştır. Böylece referans hız bilgisi bu hattan motor sürücülerine gönderilerek motorların istenilen hızda dönmesi sağlanmıştır. Sürücülerin CAN-BUS üzerinden haberleşebilmesi için sürücü üzerinde Şekil 4.30'da görüldüğü gibi "slot 1" kısmına EMC-COP01 harici haberleşme kartı takılmıştır. Haberleşme ağı RJ-45 kablosuyla sağlanır. "Master" olarak HY-TTC 60 kullanılırken, VFD C-2000 serisi sürücüler ise "slave"lerdir. Hat başında ve sonunda 120 Ω 'luk dirençler kullanılmalıdır.



Şekil 4.29. Bir motor sürücüsünün CAN-BUS hattına bağlantı blok diyagramı



Şekil 4.30. Sürücülerin CAN hattına bağlantı şeması [80].

Direksiyon açısı bilgisini almak için CAN hattına Şekil 4.31'de görülen bir CAN uyumlu enkoder ve aracın hız bilgisi için ise Şekil 4.32'de görüldüğü gibi akım çıkışlı bir ultrasonik sensör, besleme bağlantıları yapıldıktan sonra HY-TTC 60'ın analog giriş pinlerinden birine bağlanır:

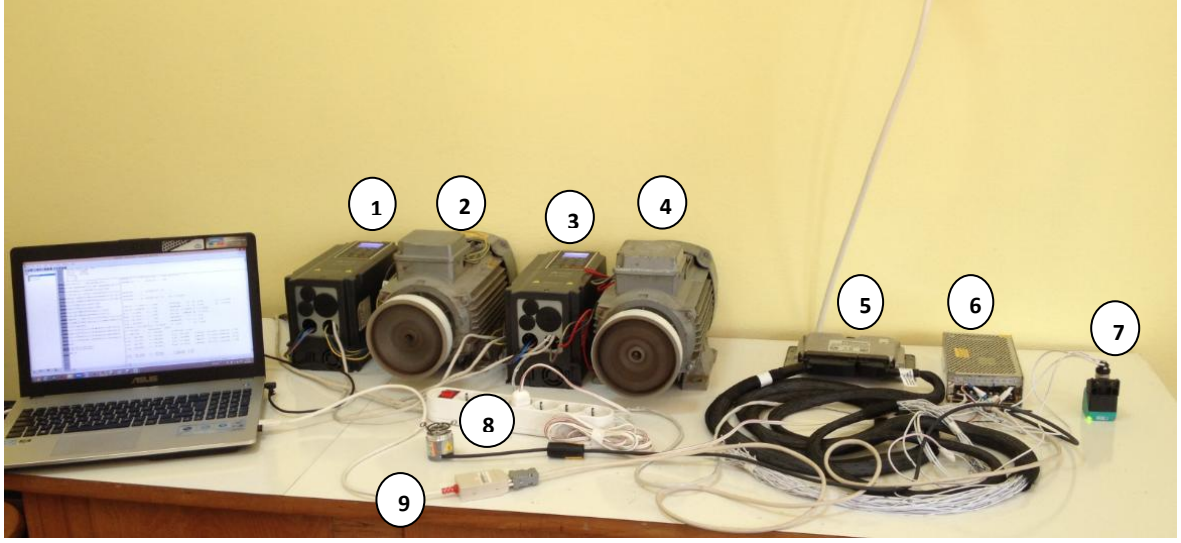


Şekil 4.31. Enkoder



Şekil 4.32. Ultrasonik sensör

Şekil 4.33'teki deney düzeneği kurularak deneysel sonuçlar elde edilmiştir. Bu düzende 2 ve 4 numarayla gösterilen 3 fazlı asenkron motorlar, 1 ve 3 numaradaki VFD C-2000 serisi sürücülere bağlanmıştır. 9 numarada gösterilen PCAN-USB aygıtının bir ucu bilgisayara bağlı, diğer ucu da CAN H ve CAN L'dan oluşan CAN-BUS hattıdır. Sürücüler RJ 45 kablosu ile CAN hattına bağlanır. Sırasıyla 5, 7 ve 8 numarada verilen HY-TTC 60, mesafe ölçer sensörü ve enkoder 6 numarayla gösterilen 24 V'luk güç kaynağından beslenir. HY-TTC 60 ve CAN uyumlu enkoder PCAN-USB kablusunun uçlarına bağlanırken, analog sensör olan mesafe ölçer HY-TTC 60'ın analog pinlerinden birine bağlanarak bu cihaz üzerinden CAN hattına bağlanmıştır.



Şekil 4.33. Deney düzeneği.

1 ve 3- VFD C-2000 Sürücü 2 ve 4- Asenkron motor 5- HY-TTC 60 6- Güç kaynağı
7- Mesafe ölçer sensörü 8- Enkoder 9- PCAN-USB

Deney düzeneği kurulduktan sonra sürücülerin parametre ayarları yapılmalıdır. Kontrol durumu için CANopen'in Delta ve DS402 olmak üzere iki standardı bulunur. DS402 standardı için parametre 09.40 1, Delta standardı için ise 0 yapılmalıdır. Bu çalışmada DS402 standardı kullanılmıştır. Asenkron motorları CAN ile haberleştirmek için aşağıda verilen parametreler ayarlanmalıdır:

1- Motorun çalışma şeklini sürücü üzerindeki "Run/Stop" tuşundan sağlamak için, parametre 00.21, 0'a (keypad) ayarlanır.

2- CANopen haberleşme için parametre 00.20, 6 olarak seçilir.

3- Parametre 09.36'dan CANopen istasyon ayarı yapılır. Sürücülere 1 ile 127 arasında birbirinden farklı olacak şekilde bir ID numarası verilir. CAN hattına bağlı bütün cihazların ID numaraları farklı olmalıdır.

4- CANopen haberleşme hız ayarı parametre 09.37'den yapılır. 1 Mbit/s için 0, 500 kbit/s için 1, 250 kbit/s için 2, 125 kbit/s için 3, 100 kbit/s için 4, 50 kbit/s için ise 5'e ayarlanmalıdır. Bu çalışmada sistem, 250 kbit/s haberleşme hızında kurulmuştur.

5- Parametre 09.46'dan CAN hattına master olarak bağlanan cihazın ID numarası girilir. Bu sistemde master olarak kullanılan modül HY-TTC 60 olduğundan bunun ID numarası olan 1 girilmiştir.

Tüm ayarlamalar yapıldıktan sonra sistemin enerjisi verilir. CoDeSys'te gerçekleştirilen yazılım "online"dan "login"e (Alt+F8) tıklanarak yüklenir ve "run" (F5) komutuyla çalıştırılır. "Run" komutu yerine sistemin enerjisi kapatılıp açılırsa da aynı işlem yapılmış olur.

Şekil 4.34'te görüldüğü gibi, enkoder ve mesafe ölçerden alınan değerlerin değişimine göre ön tekerlek içi motorların hız bilgileri PLC programda görülmektedir. Örneğin; enkoderin açısı 5°, gaz pedalı 505.538 devir/dak (200 km/h) için ön iç tekerleğin hızı 495.4643 devir/dak, ön dış tekerleğin hızı ise 518.9576 devir/dak olarak hesaplanmıştır. Bu hız bilgisine göre motorların frekans değerleri;

$$n=(120*f)/p \quad (4.12)$$

$n_{\text{ön_iç}}=495.4643$ devir/dak ve $p=4$ için $f=16.515$ Hz

$n_{\text{ön_dış}}=518.9576$ devir/dak ve $p=4$ için $f=17.298$ Hz bulunur.

Denklem (4.12)'de p kutup sayısı, f frekans olup, n devir sayısını gösterir.

```
absEncode... = 5    encoder_aci = 5
absEncode... = 5
encoderde... = 8... encoder_aci = 5    pi = 3.141593

teta_oi = 5.118844 L = 2.285    encoderde... = 8... K = 1.219    pi = 3.141593
teta_od = 4.886536 L = 2.285    encoderde... = 8... K = 1.219    pi = 3.141593
R_oi = 25.61031    L = 2.285    teta_oi = 5.118844    pi = 3.141593
R_od = 26.82467    L = 2.285    teta_od = 4.886536    pi = 3.141593
R_ai = 25.44267    L = 2.285    encoderde... = 8... d = 1.35
R_ad = 26.79267    L = 2.285    encoderde... = 8... d = 1.35
R_merkez = 26.13101 R_ai = 25.44267    d = 1.35    La = 0.835
V = 199.6875    akim_girisi = 19975
w = 505.538    V = 199.6875    r_tekerlek = 0.395
w_oi = 495.4643    V = 199.6875    R_oi = 25.61031    R_merkez = 26.13101 r_tekerlek = 0.395
w_od = 518.9576    V = 199.6875    R_od = 26.82467    R_merkez = 26.13101 r_tekerlek = 0.395
w_ai = 492.2211    V = 199.6875    R_ai = 25.44267    R_merkez = 26.13101 r_tekerlek = 0.395
w_ad = 518.3386    V = 199.6875    R_ad = 26.79267    R_merkez = 26.13101 r_tekerlek = 0.395

w_oi = 495.4643    V = 199.6875    r_tekerlek = 0.395
w_od = 518.9576    V = 199.6875    r_tekerlek = 0.395
```

Şekil 4.34. CoDeSys programında hesaplanan değerler

Motorlardan alınan bu hız bilgileri PCAN-View programından Şekil 4.35'teki gibi görülmektedir. Burada alınan ve gönderilen mesajlar, Cob-ID numaraları, mesaj uzunlukları, döngü süreleri ve sayma sayıları gibi bilgileri mevcuttur. 190 H (180+ ID numarası) adresinden enkoderden alınan bilgi görülmektedir. Alınan işlem veri nesneleri (200+ID) H adresinden, gönderilen işlem veri nesneleri ise (180+ID) H adresinden okunur. Sürücülerden birinin ID numarasına 126, diğerine ise 127 verilmiştir. Bu nedenle, sonu 7F ve 7E ile biten mesajlar sürücülerden gelen bilgilerdir. 150 H adresinin ilk 4 byte'ndan enkoder verisi, 5. byte'ndan mesafe ölçerden alınan akım bilgisi, 7. ve 8. byte'larından ise ön iç ve dış tekerleklerin hız bilgileri gönderilmiştir.

Message	DLC	Data	Cycle Time	Count	Trigger	Comment
000h	2	01 7E	40	30		
0FEh	8	00 00 00 00 00 00 00 00	377	10		
0FFh	8	00 00 00 00 00 00 00 00	377	10		
110h	6	90 F7 02 03 9C 72	242395802	6		
111h	4	6F 08 FD FC	242395802	6		
150h	8	5D 03 03 03 07 EF EF 07	9	1050638		
151h	8	8A C2 01 00 07 EF EF 07	10	1050636		
152h	8	B8 CB 01 00 07 EF EF 07	10	1050636		
190h	4	5D 03 03 03	10	1060830		
1FEh	0		10	956655		
1FFh	0		242413657	7		
27Eh	4	00 00 EF 01	10	115524		
27Fh	4	00 00 07 02	10	127422		
390h	2	00 00	242395801	6		
581h	8	10 00 00 00 00 00 00 00	3	2889784		
5FEh	8	60 42 60 00 00 00 00 00	10	147		

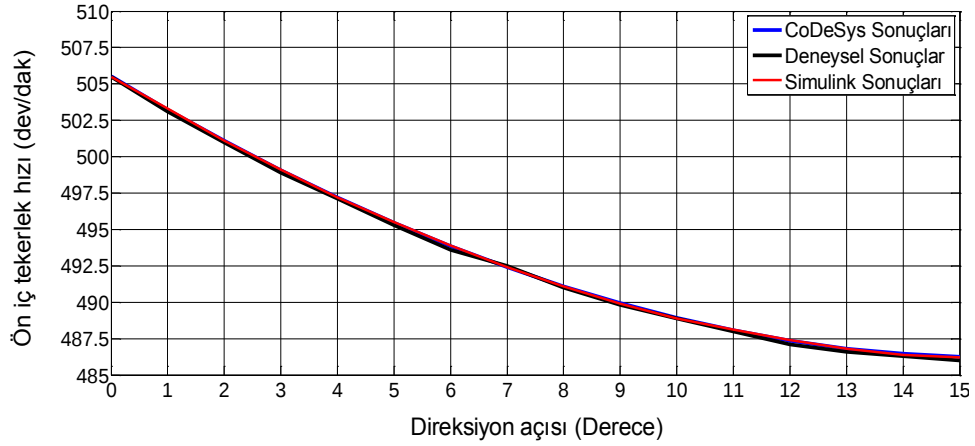
Message	DLC	Data	Cycle Time	Count	Trigger	Comment
000h	2	01 10	400	276785	Time	
000h	2	81 17	Wait	0		reset
000h	8	00 00 00 00 00 00 00 00	100	0		

Şekil 4.35. PCAN-View programından alınan ve gönderilen mesajlar

Direksiyon açısı 1°'den 15°'ye kadar birer derece aralıklarla değiştirilmiş ve bir takometre yardımı ile motorların hız değerleri deneysel olarak ölçülmüştür. Tablo 4.3 ve Tablo 4.4'te görüldüğü gibi CoDeSys'ten alınan sonuçlar, MATLAB/Simulink'ten alınan sonuçlarla ve deneysel çalışmalarla karşılaştırılarak tasarlanan sistem doğrulanmıştır. CoDeSys sonuçları ve deneysel sonuçlara göre hata oranı yüzde olarak hesaplanmıştır. Direksiyon açısının ileri ve tersi yönüne göre ön tekerlek hızlarının CoDeSys, deneysel çalışmalar ve MATLAB/Simulink'ten elde edilen sonuçlarla karşılaştırması, Şekil 4.36, Şekil 4.37, Şekil 4.38 ve Şekil 4.39'daki grafiklerde görülmektedir.

Tablo 4.3. CoDeSys'ten alınan sonuçların deneysel ve MATLAB/Simulink sonuçlarıyla karşılaştırması

Direksiyon açısı (Derece)	Araç hızı (dev/dak)	CoDeSys Sonuçları		Deneysel Sonuçlar		MATLAB Simulink Sonuçları		% Hata (Deneysel Sonuçlara Göre)	
		$n_{\text{öi}}$	$n_{\text{öd}}$	$n_{\text{öi}}$	$n_{\text{öd}}$	$n_{\text{öi}}$	$n_{\text{öd}}$	$n_{\text{öi}}$	$n_{\text{öd}}$
0	505.538	505.538	505.538	505.5	505.5	505.5	505.5	0.007	0.007
1	505.538	503.251	507.958	503.1	508.0	503.3	508	0.03	0.008
2	505.538	501.099	510.510	501.0	510.3	501.1	510.5	0.019	0.041
3	505.538	499.083	513.195	498.9	512.8	499.1	513.2	0.036	0.076
4	505.538	497.204	516.01	497.1	515.9	497.2	516	0.061	0.021
5	505.538	495.464	518.957	495.3	518.9	495.5	519	0.033	0.010
6	505.538	493.865	522.035	493.6	521.7	493.9	522	0.053	0.064
7	505.538	492.409	525.243	492.5	524.8	492.4	525.2	0.018	0.084
8	505.538	491.098	528.583	491.0	528.2	491.1	528.6	0.019	0.072
9	505.538	489.936	532.054	489.8	531.8	489.9	532.1	0.027	0.047
10	505.538	488.924	535.657	488.9	535.6	488.9	535.7	0.004	0.010
11	505.538	488.065	539.394	488.0	538.9	488.1	539.4	0.013	0.091
12	505.538	487.364	543.264	487.1	543.0	487.4	543.3	0.054	0.048
13	505.538	486.822	547.270	486.6	546.9	486.8	547.3	0.045	0.067
14	505.538	486.444	551.412	486.3	551.1	486.4	551.4	0.029	0.056
15	505.538	486.234	555.693	486.0	555.3	486.2	555.7	0.048	0.070



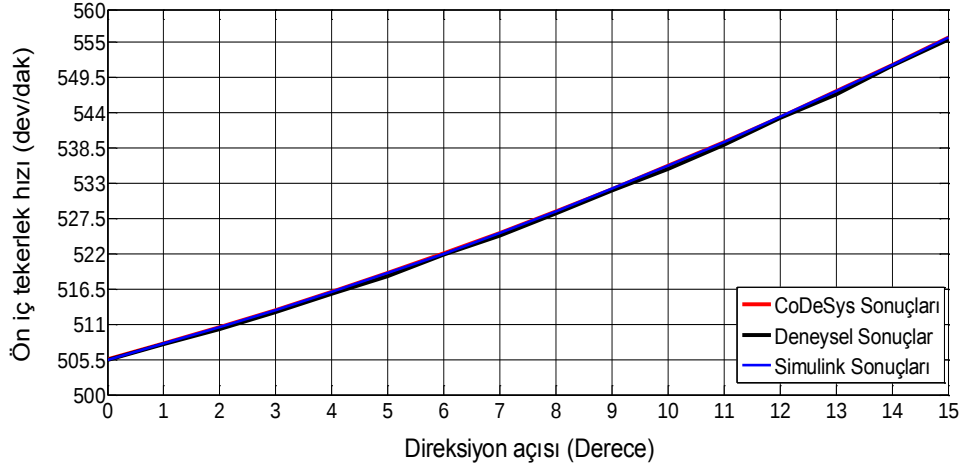
Şekil 4.36. Ön iç tekerlek hızının CoDeSys, deneysel çalışma ve Simulink sonuçlarının karşılaştırması



Şekil 4.37. Ön dış tekerlek hızının CoDeSys, deneysel çalışma ve Simulink sonuçlarının karşılaştırması

Tablo 4.4. CoDeSys'ten alınan sonuçların deneysel ve MATLAB/Simulink sonuçlarıyla karşılaştırması (Direksiyon açısı ters yön için)

Direksiyon açısı ters yön (Derece)	Araç hızı (dev/dak)	CoDeSys Sonuçları		Deneysel Sonuçlar		MATLAB Simulink Sonuçları		% Hata (Deneysel Sonuçlara Göre)	
		$n_{öi}$	$n_{öd}$	$n_{öi}$	$n_{öd}$	$n_{öi}$	$n_{öd}$	$n_{öi}$	$n_{öd}$
0	505.538	505.538	505.538	505.5	505.5	505.5	505.5	0.007	0.007
1	505.538	507.958	503.251	507.9	503.5	508	503.3	0.011	0.049
2	505.538	510.510	501.099	510.2	500.9	510.5	501.1	0.060	0.039
3	505.538	513.195	499.083	512.9	498.8	513.2	499.1	0.057	0.056
4	505.538	516.01	497.204	515.7	497.0	516	497.2	0.060	0.041
5	505.538	518.957	495.464	518.5	495.3	519	495.5	0.088	0.033
6	505.538	522.035	493.865	521.9	493.6	522	493.9	0.025	0.053
7	505.538	525.243	492.409	524.8	492.2	525.2	492.4	0.084	0.042
8	505.538	528.583	491.098	528.3	490.9	528.6	491.1	0.053	0.040
9	505.538	532.054	489.936	531.8	489.8	532.1	489.9	0.047	0.027
10	505.538	535.657	488.924	535.2	488.7	535.7	488.9	0.085	0.045
11	505.538	539.394	488.065	539.0	488.0	539.4	488.1	0.073	0.013
12	505.538	543.264	487.364	543.1	487.2	543.3	487.4	0.030	0.033
13	505.538	547.270	486.822	546.8	486.7	547.3	486.8	0.085	0.025
14	505.538	551.412	486.444	551.3	486.3	551.4	486.4	0.020	0.029
15	505.538	555.693	486.234	555.4	486.0	555.7	486.2	0.052	0.048



Şekil 4.38. Ön iç tekerlek hızının CoDeSys, deneysel çalışma ve Simulink sonuçlarının karşılaştırması



Şekil 4.39. Ön dış tekerlek hızının CoDeSys, deneysel çalışma ve Simulink sonuçlarının karşılaştırması

5. SONUÇLAR

Literatürde yapılan çalışmalar incelendiğinde genellikle arka tekerlekler için tasarlanan elektronik diferansiyel sistemler yerine, bu tez çalışmasında, ön tekerlekler için bir elektronik diferansiyel sistem tasarlanmıştır. Sistemin yazılımı CoDeSys V2.3 programı üzerinden gerçekleştirilmiştir. Ackermann-Jeantand modelinden, direksiyon açısına ve aracın hız bilgisine göre ön iç ve dış tekerleklerin hız değerleri matematiksel denklemlerden elde edilmiştir. Ackermann-Jeantand modelinden alınan denklemler kullanılarak MATLAB/Simulink'te modellemesi yapılmıştır. Değişik direksiyon açıları ve aracın hız değerlerine göre ön tekerleklerin hızları Simulink'te hesaplanmıştır. Bu denklemler CoDeSys yazılımına atılarak ön tekerlek içi motorlar için elektronik diferansiyel sistem tasarlanmıştır. Sistemin haberleşmesi CAN-BUS protokolü ile sağlanmıştır. CAN-BUS'ın hızlı, esnek, düşük maliyetli olması, az kablo kullanımı ve dolayısıyla aracın ağırlığını azaltması gibi önemli avantajlarından ötürü otomobil sektöründe yaygın olarak kullanılması bu çalışmada tercih edilmesini sağlamıştır. CAN-BUS haberleşmesi için alıcı ve gönderici çerçeveler oluşturularak, ünitelerin ID numaraları, adresleri, veri uzunlukları, CAN formatları ve hangi byte'tan hangi bilginin gönderileceği yazılımda tanımlanmıştır. Konfigürasyon tanımlamaları ile alınan ve gönderilen mesajların PDO ve SDO ayarlamaları yapılmıştır.

Sistemde, elektrikli aracın ön iki tekerleğinin içine yerleştirilmek üzere iki adet asenkron motor ve sürücüsü kullanılmıştır. Direksiyon açısı bilgisini almak için bir enkoder ve aracın hız bilgisi için ise bir ultrasonik sensör bağlanmıştır. Master olarak kullanılan HY-TTC 60 işlemcisi, slave birimlerden aldığı bilgileri CAN hattına gönderir. Hattan gereken mesajlar ilgili üniteler tarafından alınır. Motorlara bağlı sürücüler, enkoder ve akım sensörü slave olarak kullanılmıştır. Bir viraja girerken direksiyonun dönme açısı ve aracın o andaki hızına göre ön tekerleklerin hız değerleri hesaplanarak, CAN-BUS üzerinden motor sürücülerine gönderilmiştir. Böylece motorların gereken hızda dönmesiyle elektronik diferansiyel sistem tasarlanmıştır. Daha sonra, bir takometre yardımıyla farklı direksiyon açılarına göre motorların hız değerleri deneysel olarak ölçülmüştür. Direksiyon açısının 1°'den 15°'ye kadar birer derece aralıklarla değişimine göre, CoDeSys'ten hesaplanan hız değerleri, Simulink'ten alınan sonuçlar ve deneysel çalışmalar karşılaştırılarak sistem doğrulanmıştır. İleriki çalışmalarda, tekerlek içi elektrikli

araçların sürüş esnasındaki güvenliği ve konforu için her motorun ayrı ayrı kontrolleri yapılabilir. Bu kontrollerde güvenlikle ilgili tüm sensör ve kameraların bilgileri işlenerek doğrudan araç kontrolü veya sürücüye uyarı verilebilmesi üzerine çalışmalar gerçekleştirilebilir.

KAYNAKLAR

- [1] **Farsi, M., Ratcliff, K., Barbosa, M.,** 1999. An Overview of Controller Area Network, *Computing and Control Engineering Journal*, pp. 113-120.
- [2] **Chaari, L., Masmoudi, N., Kamoun, L.,** 2002. Electronic Control in Electric Vehicle Based on CAN Network, *IEEE International Conference on Systems, Man and Cybernetics*, Vol.7, pp. 4-pp.
- [3] **Ünal, İ.,** 2006. CAN Üzerinden PIC Programlama, Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi.
- [4] **Ahn, B., Park, B., Ki, Y., Jeong, G., Ahn, H.,** 2006. Development of a Controller Area Network Interface Unit and Its Application to A Fuel Cell Hybrid Electric Vehicle, *IEEE International Joint Conference on SICE-ICASE*, Korea, Oct. 18-21, pp. 5545-5549.
- [5] **Ran, P., Wang, B., Wang, W.,** 2008. The Design of Communication Converter Based on CAN Bus, *IEEE International Conference on Industrial Technology (ICIT)*, pp. 1-5.
- [6] **Li, R., Liu, C., Luo, F.,** 2008. A Design for Automotive CAN Bus Monitoring System, *IEEE Vehicle Power and Propulsion Conference (VPPC)*, China, pp. 1-5.
- [7] **Li, F., Wang, L., Liao, C.,** 2008. CAN (Controller Area Network) Bus Communication System Based on Matlab/Simulink, *IEEE 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08)*, pp. 1-4.
- [8] **WAN, X., XING, Y., CAI, L.,** 2009. Application and Implementation of CAN Bus Technology in Industry Real-time Data Communication, *IEEE International Conference on Industrial Mechatronics and Automation*, pp. 278-281.
- [9] **Karaduman, A.,** 2009. CAN ve TTCAN Sistemlerinin Modellenmesi ve Performans Analizi, Yüksek Lisans Tezi, Muğla Üniversitesi.
- [10] **Jianfeng, W., Dafang, W., Jie, X.,** 2009. The design of Electric Motor Car's Body Network Based on CAN-bus Distributed Control, *IEEE Chinese Control and Decision Conference (CCDC)*, pp. 3712-3717.
- [11] **Kara, İ.,** 2009. CAN Haberleşme Protokolünün İncelenmesi ve Bir Sıcaklık Kontrol Sistemine Uygulanması, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi.
- [12] **Xin, F., Chun, H.,** 2010. Design and Research on Air Conditioning Control Network of Electric Vehicle Based on CAN-bus, *IEEE International Conference on Measuring Technology and Mechatronics Automation*, pp. 970-973.

- [13] **Ye, Q.**, 2010. Research and Application of CAN and LIN Bus in Automobile Network System, *IEEE 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, pp. 150-154.
- [14] **Ran, L., Junfeng, W., Haiying, W., Gehen, L.**, 2010. Design Method of CAN BUS Network Communication Structure for Electric Vehicle, *IFOST 2010 Proceedings*, pp. 1-4.
- [15] **Öztürk, M.**, 2010. Hidrojen Hibrit Otobüslerde CAN Bus Sistemleri ve Uygulaması, Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü.
- [16] **Diñer, E.**, 2010. CAN BUS ile Dağıtık Kontrol Uygulaması, Yüksek Lisans Tezi, Niğde Üniversitesi.
- [17] **Yardım, F.**, 2010. Bir Tren Eğitim Simülatörünün CAN Protokolü Kullanarak Gerçeklenmesi, Yüksek Lisans Tezi, Kocaeli Üniversitesi.
- [18] **Tuncay, R. N., Üstün, Ö., Yılmaz, M., Gökçe, C., Karakaya, U.**, 2011. Design and Implementation of an Electric Drive System for In-Wheel Motor Electric Vehicle Applications, *IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp.1-6.
- [19] **Civelek, U.**, 2012. A Software Tool for Vehicle Calibration, Diagnosis and Test Via Controller Area Network, Yüksek Lisans Tezi, ODTÜ.
- [20] **Henderson, R. J., Conrad, J. M., Pavlich, C.**, 2014. Using a CAN Bus for Control of an All-Terrain Vehicle, *IEEE SOUTHEASTCON*, pp. 1-5.
- [21] **Yılmaz, M. Y.**, 2014. Design and Implementation of a CAN Communication Line Diagnosis Systematic for Road Road Vehicles, Yüksek Lisans Tezi, Çukurova Üniversitesi.
- [22] **Lee, J., Ryoo, Y., Lim, Y., Freere, P.**, 2000. A Neural Network Model of Electric Differential System for Electric Vehicle, *IEEE 26th Annual Conference IECON*, Vol. 1 pp. 83-88.
- [23] **Economou, J. T., Luk, P. C. K., Tsourdos, A., White, B. A.**, 2003. Hybrid Modelling of An All-Electric Front-Wheel Ackerman Steered Vehicle, *IEEE 58th Vehicular Technology Conference (VTC)*, Vol. 5, pp. 3294-3298.
- [24] **Gair S., Cruden, A., McDonald, J., Hredzak, B.**, 2004. Electronic Differential with Sliding Mode Controller for a Direct Wheel Drive Electric Vehicle, *IEEE International Conference on Mechatronics (ICM'04)*, pp. 98-103.
- [25] **Haddoun, A., Benbouzid, M. E. H., Diallo, D., Abdessemed, R., Ghouili, J., Srairi, K.**, 2008. Modeling, Analysis, and Neural Network Control of An EV Electrical Differential, *IEEE Transactions on Industrial Electronics*, Vol. 55, No. 6, pp. 2286-2294.

- [26] **Jain, M., Williamson, S. S.**, 2009. Suitability Analysis of In-Wheel Motor Direct Drives for Electric and Hybrid Electric Vehicle, *IEEE Electrical Power & Energy Conference (EPEC)*, pp. 1-5.
- [27] **Zhao, Y. E., Zhang, J. W., Guan, X. Q.**, 2009. Modeling and Simulation of Electronic Differential System for an Electric Vehicle with Two-Motor-Wheel Drive, *IEEE International Journal of Vehicle Systems Modelling and Testing*, Vol. 4, No. 1, pp. 117-131.
- [28] **Zhou, Y., Li, S., Zhou, X., Fang, Z.**, 2010. The Control Strategy of Electronic Differential for EV with Four In-Wheel Motors, *IEEE Chinese Control and Decision Conference*, pp. 4190-4195.
- [29] **Sampaio, R. C. B., Becker, M., Lemos, V. L., Siqueira, A. A. G., Ribeiro, J., Caurin, G. A.**, 2010. Robust Control in 4×4 Hybrid-converted Touring Vehicles During Urban Speed Steering Maneuvers, *IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1-6.
- [30] **Yu, C., Tseng, C., Hsu, Y.**, 2011. Electronic Stability Control for Direct-Drive Electric Vehicle, *IEEE International Conference on Electric Information and Control Engineering (ICEICE)*, pp. 4987-4991.
- [31] **Zhai, L., Dong, S.**, 2011. Electronic Differential Speed Steering Control for Four In-Wheel Motors Independent Drive Vehicle, *IEEE Proceedings of the 8th World Congress on Intelligent Control and Automation*, Taiwan, pp. 780-783.
- [32] **You, S., Lee, H., Lee, D., Mok, H., Lee, Y., Han, S.**, 2011. Speed Ratio Control for Electronic Differentials, *Electronics Letters*, Vol. 47, No. 16, pp. 933-934.
- [33] **Sampaio, R. C. B., Hernandez, A. C., Fernandes, V. V. M., Becker, M., Siqueira, A. A. G.**, 2012. A New Control Architecture for Robust Controllers in Rear Electric Traction Passenger HEVs, *IEEE Transactions on Vehicular Technology*, Vol. 61, No. 8, pp. 3441-3453.
- [34] **Wu, X., Xu, M., Wang, L.**, 2013. Differential Speed Steering Control for Four-Wheel Independent Driving Electric Vehicle, *IEEE International Symposium on Industrial Electronics (ISIE)*, pp. 1-6.
- [35] **Draou, A.**, 2013. Electronic Differential Speed Control for Two In-Wheels Motors Drive Vehicle, *IEEE 4th International Conference on Power Engineering, Energy and Electrical Drives*, İstanbul, pp. 764-769.
- [36] **Kahveci, H., Okumus, H. İ., Ekici, M.**, 2013. An Electronic Differential System Using Fuzzy Logic Speed Controlled In-Wheel Brushless DC Motors, *IEEE 4th International Conference on Power Engineering, Energy and Electrical Drives*, İstanbul, pp. 881-885.

- [37] **Munusamy, B. R., Weigl, J. D.,** 1999. Design of Rear Wheel Torque Vectoring for Dual Motor Electric Drive System, *IEEE 2014 Nineth International Conference on Ecological Vehicles and Renewable Energies Patent*, No: 5947855 dated 7.9.1999.
- [38] **Wu, X., Yang, L., Xu, M.,** 2014. Speed Following Control for Differential Steering of 4WID Electric Vehicle, *IEEE 40th Annual Conference IECON*, pp. 3054-3059.
- [39] **Uğur, E.,** 2011. Prototip Bir Elektrikli Araç Üzerinde Enerji Yönetim Sisteminin Uygulanması, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi.
- [40] **Güner, C.,** 2013. Dışarıdan Şarj Edilebilen Hibrit Elektrikli Araç ile Menzil Artırıcı Elektrikli Araç Konseptlerinin Karşılaştırmalı Analizi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi.
- [41] **Mökükçü, M. S.,** 2014. Elektrikli Araç Sürüş Sistemi Tasarımı ve İmalatı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi.
- [42] **Altanneh, N.,** 2012. Güneş Pili ve Hidrojen Yakıt Pilinden Beslenen Küçük Bir Elektrikli Araç İçin Batarya Şarj Sistemi Tasarımı ve Gerçekleştirilmesi, Yüksek Lisans Tezi, Gazi Üniversitesi.
- [43] **Mutlu, Y.,** 2011. Elektrikli Araç Motorunun Soğutma Sistem Tasarımı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi.
- [44] www.wikipedia.org.tr Elektrikli Otomobil.
- [45] **Larminie, J., Lowry, J.,** 2004. Electric Vehicle Technology. Oxford Brookes University, UK.
- [46] **Arna, S.,** 2010. Pil ve Bataryaların Ömürleri.
- [47] **Stevic, Z.,** New Generation of Electric Vehicles.
- [48] **Soylu, Ş.,** 2011. Electric Vehicles- The Benefits and Barriers.
- [49] **Husain, I.,** 2005. Electric and Hybrid Vehicles Design Fundamentals.
- [50] **Ehsani, M., Gao, Y., Emadi, A.,** 2005. Modern Electric, Hybrid Electric, and Fuel Cell Vehicles, Fundamentals, Theory, and Design. Boca Raton, FL: CRC.
- [51] **M. H., Westbrook, B. Sc., (Eng.), C., Eng., F. I. E. E., F. I. Mech. E., F. Inst. P,** 1986. Automotive Electronics, IEEE, Vol. 133, No. 4, pp. 241-258.
- [52] **Tuncay, R. N., Üstün, Ö.,** Otomotiv Elektroniğindeki Gelişmeler.
- [53] **Aydoğan, H., Acaroğlu, M.,** 2009. Günümüz İçten Yanmalı Motorlarında Kullanılan Kontrol ve Veri İletim Sistemleri ve Gelişmeler, Taşıt Teknolojileri Elektronik Dergisi (TATED), Vol. 1, No. 2, pp. 51-62.

- [54] **Vlasic, L., Parent, M., Harashima, F.,** 2001. Intelligent Vehicle Technologies: Theory and Applications, Butterworth-Heinemann.
- [55] **M.E.B.,** 2012. Motorlu Araçlar Teknolojisi, Araç Yönetim Sistemleri, Ankara.
- [56] **Bayır, R., Soylu, E.,** 2014. Endüstriyel İletişim Sistemleri Ders Notu.
- [57] **Çayan, N.,** Pnömatikte Alansal Veriyolu Uygulamaları, 3. Ulusal Hidrolik Pnömatik Kongresi ve Sergisi, pp. 453-475.
- [58] **Deepika, T. P., Bhagya, P.,** 2014. Dual Redundancy CAN-Bus Controller Based on FPGA, Vol. 4, pp. 1-4.
- [59] **Caner, H.,** 2006. FPGA Donanımı Üzerinde Araç Plakası Tanıma Sistemi, Yüksek Lisans Tezi, Hacettepe Üniversitesi.
- [60] **Khan, R. M., Thakare, A. P.,** 2010. FPGA Based Design & Implementation of Serial Data Transmission Controller, Vol. 2, pp. 5526-5533.
- [61] **Boterenbrood, H.,** 2000. CANopen High-level Protocol for CAN-bus, NIKHEF, Amsterdam.
- [62] www.kurumahmut.blogcu.com CAN-BUS Haberleşme Sistemi.
- [63] **Ekiz, H., Kutlu, A., Baba, M. D., Powner, E.T.** 1996. Performance Analysis of a CAN/CAN Bridge, *IEEE International Conference on Network Protocols*, pp. 181-188.
- [64] **Tsou, T.,** 2012. An Implementation of Controller Area Network Bus Analyzer Using Micriblaze and Petalinux, Yüksek Lisans Tezi, Western Michigan University.
- [65] **Alkan, M.,** 2010. Inter-Connected Flexray and CAN Networks for In-Vehicle Communication Gateway Implementation and End-to-End Performance Study, Yüksek Lisans Tezi, ODTÜ.
- [66] **Santur, Y.,** 2006. Kontrol Alan Ağı Protokolü Tabanlı Bir Endüstriyel Ağın Kurulması ve Yönetilmesi, Yüksek Lisans Tezi, Fırat Üniversitesi.
- [67] **Kashif, H., Bahig, G., Hammad, S.,** 2009. CAN Bus Analyzer and Emulator, *IEEE 4th International Design and Test Workshop*, pp. 1-4.
- [68] **Cao, W., Lin, C., Zhou, W., Sun, F.,** 2009. A real-time Planning-based Scheduling Policy with CAN for Automotive Communication Systems, *IEEE World Congress on Computer Science and Information Engineering*, pp. 186-191.
- [69] **Kirschbaum, A., Renner, F. M., Wilmes, A., Glesner, M.,** 1996. Rapid-Prototyping of a CAN-BUS Controller: A Case Study, *IEEE Seventh International Workshop on Rapid System Prototyping*, pp. 146-151.

- [70] **Pantiruc, C., Negru, M.,** 2011. FPGA Based CAN Data Visualization, *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 245-252.
- [71] **Xiang-Dong, H., Hui-Mei, Y., Xiao-Xu, Z.,** 2013. Design of Dual Redundancy CAN-bus Controller Based on FPGA, *IEEE 8th Conference on Industrial Electronics and Applications*, pp. 843-847.
- [72] **Kammerer, R., Frömel, B., Obermaisser, R., Milbredt, P.,** 2013. Composability and Compositionality in CAN-Based Automotive Systems based on Bus and Star Topologies, *IEEE 11th International Conference on Industrial Informatics (INDIN)*, pp. 116-122.
- [73] www.emrah-kaba.blogcu.com Codesys Nedir
- [74] TTControl Targets 2012_a_SP1, CoDeSys_V_23_E
- [75] TTControl Core Brochure
- [76] TTControl HYDAC International Data Sheet
- [77] HY-TTC 50_60_90 User Manual V2.2
- [78] HYDAC International Universal Mobile Controller HY-TTC 60 Data Sheet
- [79] PCAN-USB User Manual V2.1.3
- [80] VFD C-2000 User manual

ÖZGEÇMİŞ

Merve YILDIRIM

Fırat Üniversitesi

Mühendislik Fakültesi

Elektrik-Elektronik Mühendisliği Bölümü

Elazığ

merveyildirim@firat.edu.tr

- | | |
|------------------|---|
| 1989 | Elazığ'da doğdu. |
| 2007-2011 | Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği'nden mezun oldu. |
| 2012- | Fırat Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü'nde Araştırma Görevlisi olarak görev yapmaya devam etmektedir. |