

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

OTONOM OTOMOBİL İÇİN ARAÇ MEKATRONİĞİ UYGULAMASI

YÜKSEK LİSANS TEZİ
Zeynel KOÇ

Anabilim Dalı : Mekatronik Mühendisliği

Programı : Mekatronik Mühendisliği

EKİM 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

OTONOM OTOMOBİL İÇİN ARAÇ MEKATRONİĞİ UYGULAMASI

YÜKSEK LİSANS TEZİ
Zeynel KOÇ
518081032

Tezin Enstitüye Verildiği Tarih : 15 Eylül 2011

Tezin Savunulduğu Tarih : 13 Ekim 2011

Tez Danışmanı : Yrd. Doç. Dr. Özgür ÜSTÜN (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Nejat TUNCAY(OKAN Ü.)
Prof. Dr. Metin GÖKAŞAN(İTÜ)

EKİM 2011

ÖNSÖZ

Bu tez çalışmasında, otonom araç için araç mekatroniği uygulaması yapılmıştır. Aracın üzerine yerleştirilen çeşitli donanımlar ile araç otonom hale gelmiştir. Tez çalışmam sırasında beni yönlendiren ve bilgi birikimi ile yol gösteren değerli hocalarım Yrd. Doç. Dr. Özgür ÜSTÜN'e, Prof. Dr. Nejat TUNCAY'a, Prof. Dr. Tevfik AKGÜN'e ve Prof. Dr. Ümit ÖZGÜNER'e, eğitim hayatım boyunca beni destekleyen aileme ve tezimin her aşamasında bana sonsuz destek veren ve katkıda bulunan sevgili arkadaşım Şafak GÜNER'e teşekkür ederim.

Eylül 2011

Zeynel KOÇ

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY	xvii
1. GİRİŞ	1
1.1 Otonom Kara Araçları Konusunda Yapılan Çalışmalar.....	1
1.2 Otonom Otomobil İçin Sistem Tasarımı	2
1.3 Kullanılan Aracın Özellikleri	5
1.4 Kullanılan Donanım Parçalarının Genel Özellikleri	6
1.4.1 PLC	6
1.4.2 Servo motor sürücüsü.....	7
1.4.3 Servo motor (8LVA22.R0030D000) ve redüktör	7
1.4.4 LIDAR(Light Detection and Ranging)	8
1.4.5 CAN kartı (Kvaser USBcan II HS/LS).....	9
1.4.6 Diğer elemanlar	10
1.5 Kullanılan Yazılımlar	10
1.5.1 Automation Studio	10
1.5.2 Kvaser CANKing.....	11
1.5.3 Visual Studio.....	11
1.5.4 Matlab	11
2. DONANIMLARIN MEKANİK MONTAJI VE ELEKTRİK	
BAĞLANTILARI	13
2.1 Mekanik Montaj	13
2.1.1 Lidar	13
2.1.2 Servo motor.....	14
2.1.3 Lineer motor.....	15
2.1.4 Sürücüler ve PLC	18
2.1.5 IMU	18
2.1.6 Besleme	19
2.1.7 Kablolama ve sigortalar	19
2.2 Elektrik Bağlantıları	21
3. ARAÇ ÜZERİNDEKİ CAN HATTINA ERİŞİM	23
3.1 CAN(Controller Area Network).....	23
3.2 Araç Üzerindeki Can Yapısı	30
3.3 Can Hattına Bağlanabilme	31
3.4 Can Hattı Sayesinde Erişilebilen Veriler	34
4. DİREKSİYON KONTROLÜ.....	39

4.1 Direksiyon Sistemini Oluşturan Temel Elemanlar.....	40
4.1.1 Direksiyon simidi	40
4.1.2 Direksiyon mili.....	40
4.1.3 Direksiyon dişli kutusu	41
4.1.4 Pinyon dişli kutusu	41
4.1.5 Dönel bilyeli dişli kutusu	43
4.2 Ön Düzen Geometrisi.....	44
4.2.1 Kamber açısı.....	44
4.2.2 King-Pim açısı.....	45
4.2.3 Toplam açı.....	45
4.2.4 Kaster açısı	46
4.2.5 Toe açısı	47
4.2.6 Dönüş açısı Ackerman geometrisi.....	48
4.3 Yönlendirme Sistemi Çeşitleri	49
4.3.1 Mekanik yönlendirme sistemleri.....	49
4.3.2 Hidrolik güç takviyeli yönlendirme sistemleri.....	50
4.3.3 Elektro hidrolik güç takviyeli yönlendirme sistemleri.....	53
4.3.4 Elektrik güç takviyeli yönlendirme sistemleri	54
4.3.5 Steer-by-wire yönlendirme sistemleri	56
4.4 Otonom Araç İçin Direksiyon Modeli.....	58
4.4.1 Matematik modelin çıkartılması.....	59
4.4.1.1 Servo motor matematik modeli	59
4.4.1.2 Hidrolik güç takviyeli mekanik yönlendirme sisteminin servo motor ile birlikte matematik modeli	60
4.4.2 Simulink modelin kurulması	62
4.4.3 Pozisyon kontrolcüsü tasarımı	67
5. SONUÇLAR	75
KAYNAKLAR.....	77
EKLER.....	79

KISALTMALAR

ACK	: Acknowledged Bit
ADC	: Analog to Digital Converter
CAN	: Controller Area Network
CRC	: Cyclic Redundancy Check
DARPA	: Defense Advanced Research Project Agency
DGPS	: Differential Global Positioning System
DLC	: Data Length Code
EOBD	: Europe On-Board Diagnostic
EOF	: End of Frame
ESP	: Electronic Stability Program
GND	: Ground
GPS	: Global Positioning System
HP	: Horse Power
ID	: Identifier
IDE	: Identifier Extension Bit
IMU	: Inertial Measurement Unit
LIDAR	: Light Detection and Ranging
PLC	: Programmable Logic Controller
r0	: Reserve Bit
RTR	: Remote Transfer Request
SOF	: Start of Frame
USB	: Universal Serial Bus
VIAC	: Vislab Intercontinental Autonomous Challenge

ÇİZELGE LİSTESİ

	<u>Sayfa</u>
Çizelge 3.1 : Veri baytları kodlaması(d:baskın, r:çekinik) [3].....	30
Çizelge 4.1 : Servo motora ait değerler.....	63
Çizelge 4.2 : Redüktör değerleri.....	64
Çizelge 4.3 : Direksiyon sistemi değerleri.	66
Çizelge 4.4 : Hidrolik sistem değerleri.....	66

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Sistemin genel yapısı.....	4
Şekil 1.2 : Projede kullanılan araç	5
Şekil 1.3 : PLC X20 ana modülü	6
Şekil 1.4 : PLC X20 yardımcı modülleri	6
Şekil 1.5 : Servo motor sürücüsü	7
Şekil 1.6 : Servo motor	7
Şekil 1.7 : Redüktör	8
Şekil 1.8 : SICK LMS221	9
Şekil 1.9 : Tarama açısı	9
Şekil 1.10 : CAN hattı erişim kartı	10
Şekil 1.11 : Automation Studio programı örnek görüntüsü.....	10
Şekil 2.1 : Lidar'ın montajı.....	13
Şekil 2.2 : Servo motorun montajı.....	14
Şekil 2.3 : Servo motor ve direksiyon mili(demonte)	15
Şekil 2.4 : Servo motor ve direksiyon mili	15
Şekil 2.5 : Lineer motor	16
Şekil 2.6 : Lineer motorun fren pedalına monte edilişi	16
Şekil 2.7 : Mafsal 1	17
Şekil 2.8 : Mafsal 2	17
Şekil 2.9 : PLC ve sürücüler	18
Şekil 2.10 : IMU	19
Şekil 2.11 : Aküler	19
Şekil 2.12 : Besleme kablolarının geçişi.....	20
Şekil 2.13 : Sigortalar	20
Şekil 2.14 : Elektrik bağlantı şeması	21
Şekil 3.1 : Genel CAN yapısı [3].....	23
Şekil 3.2 : CAN-High ve CAN-Low için gerilim değerleri [5].....	25
Şekil 3.3 : Bir bitin zaman ekseninde incelenmesi [5]	26
Şekil 3.4 : Üç modülün aynı anda hatta veri yazması [5].....	27
Şekil 3.5 : Örnek veri mesajı(Standart format)[6]	29
Şekil 3.6 : Kontrol alanı iç yapısı[6].....	30
Şekil 3.7 : Örnek veri mesajı(Genişletilmiş format)[6]	30
Şekil 3.8 : Araç üzerindeki modüller ve CAN hatları	31
Şekil 3.9 : CAN hattına bağlanabilme	32
Şekil 3.10 : EOBD soket pin diyagramı	33
Şekil 3.11 : CAN kart pin diyagramı	33
Şekil 3.12 : Kvaser CanKing programı ekran görüntüsü.....	35
Şekil 3.13 : Direksiyon açısı-araç hızı-pedal konumu.....	35
Şekil 3.14 : Direksiyon açısı bilgisi.....	37
Şekil 3.15 : Direksiyon açısı bilgisi grafiği	37
Şekil 4.1 : Direksiyon mili[11]	40
Şekil 4.2 : Pinyon Dişli Kutusu [13].....	41

Şekil 4.3 : Pinyon-Kremayer sistem [14].....	42
Şekil 4.4 : Kremayerin parçaları [15]	42
Şekil 4.5 : Dönel bilyeli dişli kutusu [14].....	43
Şekil 4.6 : Dönel bilyeli mekanik direksiyon sistemi [14]	43
Şekil 4.7 : Kamber açısı [11]	44
Şekil 4.8 : Pozitif ve negatif kamber[11].....	44
Şekil 4.9 : King-pim açısı[megep].....	45
Şekil 4.10 : Toplam açı[11]	46
Şekil 4.11 : Kaster açısı [11].....	47
Şekil 4.12 : Toe açısı [17].....	47
Şekil 4.13 : Yanal kayma [11]	48
Şekil 4.14 : Dönüş açısı [17].....	49
Şekil 4.15 : Ackerman geometrisi [18].....	49
Şekil 4.16 : Mekanik yönlendirme sistemi	50
Şekil 4.17 : Hidrolik güç takviyeli yönlendirme sistemi [15].....	51
Şekil 4.18 : Silindir-kremayer-valf-pinyon.....	51
Şekil 4.19 : Hidrolik valfin iç yapısı[14]	52
Şekil 4.20 : Tork sensörü burulma miktarı [19].....	53
Şekil 4.21 : Hidro elektrik güç takviyeli yönlendirme sistemi [20].....	54
Şekil 4.22 : Çiftli pinyon tipi – dişli tipi [13]	55
Şekil 4.23 : Kolon tipi- pinyon tipi [13]	55
Şekil 4.24 : Steer-by-wire	57
Şekil 4.25 : Proje aracı üzerindeki direksiyon yapısı [19].....	58
Şekil 4.26 : Servo motor şeması	59
Şekil 4.27 : Sistem modeli	60
Şekil 4.28 : Servo motor açık çevrim simulink modeli	63
Şekil 4.29 : Açık çevrim servo motor step cevabı	63
Şekil 4.30 : Açık çevrim simulink modeli	64
Şekil 4.31 : Subsistem (Steering system).....	65
Şekil 4.32 : Açık çevrim servo motor step cevabı(mekanik yük bağlı).....	66
Şekil 4.33 : Kapalı çevrim simulink modeli	67
Şekil 4.34 : Sinyal alt-üst sınırları	68
Şekil 4.35 : Alt limit belirleme	68
Şekil 4.36 : Üst limit belirleme	68
Şekil 4.37 : Parametre girişi.....	69
Şekil 4.38 : İterasyon adımları	69
Şekil 4.39 : Sistemin step cevabı(Q_m).....	70
Şekil 4.40 : Sistemin step cevabı(Q_t).....	70
Şekil 4.41 : Çıkış sinyallerinin farkı(Q_t-Q_m)	71
Şekil 4.42 : Sisteme ait Root Locus	71
Şekil 4.43 : Açık çevrim bode diyagramı	72
Şekil 4.44 : Kapalı çevrim bode diyagramı(çıkış Q_m)	72
Şekil 4.45 : Kapalı çevrim bode diagramı (çıkış Q_t)	73
Şekil 5.1 : Deney düzeneği.....	75
Şekil 5.2 : Gerçek system çıkışı	76
Şekil 5.3 : Simülasyon çıkışı.....	76

OTONOM OTOMOBİL İÇİN ARAÇ MEKATRONİĞİ UYGULAMASI

ÖZET

Her geçen gün gelişen teknoloji sayesinde insanoğlu kendi iş yükünü giderek makinelere aktarmaktadır. Öncelikle birçok alanda kullanılan makine otomatik hale getirilmiş ve insanlar tarafından daha kolay kontrol edilebilir duruma gelmiştir. Teknolojinin ilerleyişinin durmayışı sonraları yarı otonom sistemleri ortaya çıkarmış, yakın gelecekte ise tam otonom sistemleri, yani kendi kendine karar veren makine ve cihazları görmemizi sağlayacaktır.

Tüm bu gelişmelerden taşımacılık sektörü de payını alacaktır. Zaten şu an hava taşımacılığında var olan otomatik pilot sistemi bu teknolojinin bir ürünü sayılabilir. Son zamanlarda piyasalara sürülen otomatik park edebilen ya da tehlike anında sürücü yerine frene daha erken basan (şehir içi düşük hızlarda) otomobiller, içinde yarı otonom diyebileceğimiz sistemler barındırmaktadır. Gelecekte bu gibi sistemlerin birçok araçta tam otonom olarak görülmesi kaçınılmazdır. Otonom otomobiller bu teknolojilerin en çabuk ve en yaygın şekilde işleyeceği taşıtlardır.

Otonom otomobiller, insan müdahalesi olmaksızın belirlenen bir noktadan başka bir noktaya erişebilen araçlardır. Normallerinden farklı olarak bu araçlar geliştirilmiş bir takım donanım ve yazılımlara sahiptirler. Bu donanımlar ve onlara özel tasarlanmış yazılımlar sayesinde aracın hareket etmesini sağlayacak fonksiyonlara (direksiyon, gaz pedalı, fren pedalı, vites vb. gibi) etki edilir. Aracın hangi hızda ve hangi yöne hareket edeceği ise ileri düzeyde algoritmalara sahip merkezi bir birim tarafından belirlenir. Bu merkezi birim sensörlerden gelen verilere ve sürücü tarafından önceden belirlenmiş hedef noktaya göre gerekli hızı ve yönü belirler.

Araç sürüş güvenliğini yükseltmenin metotlarından biri araç üzerindeki sensör çeşidini ve sayısını arttırmaktır. Bu ek maliyetin yanı sıra, gelen veri miktarını artırır ve karmaşılaşır (özellikle algılayıcı olarak kamera vb. gibi cihazlar varsa). Merkezi birim karar vermenin yanı sıra sensörlerden gelen verileri de işlemek zorundadır. Fakat bu kadar yükün altına girmek onu ve beraberinde sistemi yavaşlatır, zora sokar. Bu gibi durumlarda genellikle iki ayrı merkezi birim kullanılır. Bunlardan biri gelen verileri işlemeye ve bir noktada toplamaya yarar. Diğeri ise işlenmiş verilere göre uygun durumları hesaplar, karar verir ve bu karara göre ilgili aktüatörlere komutlar gönderilir.

Otonom otomobiller bu kompleks yapılarına ve denenmemiş bir teknoloji olmasına rağmen, beraberinde birçok avantaj getirmektedir. İnsana özgü özelliklere sahip olmaması (uyku, yorgunluk, sinir vb.), insandan çok daha hızlı tepki verebilmesi ve koşullara göre tüm olasılıkları hesaplayıp en doğru kararı seçmesi otonom otomobillerin en önemli özellikleridir. Bu özellikler sayesinde insan kaynaklı trafik kazaları yok olacaktır. Ayrıca takip mesafesi yüksek hızlarda bile birkaç metreye inecek, bu sayede yakıt tasarrufu ve yol kapasitesi arttırılacaktır. Araç kullanamayacak durumda olan insanlar (yaşlı, alkollü vb. gibi) için bile güvenli bir sürüş imkânı sağlayacaktır.

Bu çalışmanın konusu otonom otomobil için araç mekatroniği uygulamasıdır. Çalışmada öncelikli olarak otonom araçların yapısı ve gelişimleri incelenmiştir. Daha sonra üniversitenin otonom otomobil projesinde yer alan araç için sistem tasarlanmıştır. Diğer bölümde tasarıma bağlı olarak projede kullanılan cihazların mekatronik montajları ve elektrik bağlantıları yapılmıştır.

Ardından araç üzerindeki haberleşme ağına (CAN) ait protokol incelenmiştir. Bu ağ üzerindeki verilere erişim için gerekli elektriksel bağlantılar yapılmıştır. Devamında haberleşme protokolüne uygun yazılım geliştirilmiş ve yapılan deneylerle birçok veri elde edilmiştir.

Bir sonraki bölümde otomobiller üzerindeki direksiyon sistemleri incelenmiş, projedeki otomobile ait direksiyon sisteminin matematik model çıkartılmıştır. Bu modele bağlı olarak simülasyonlar yapılmıştır. Sonuç olarak araç üzerindeki gerçek sistemle yapılan deneylerin sonuçları, simülasyon sonuçları ile karşılaştırılmıştır.

VEHICLE MECHATRONICS APPLICATION FOR AUTONOMOUS AUTOMOBILES

SUMMARY

Thanks to technology which developing each passing day, mankind transfers his power to machines. First of all machines that used in various field are made automatic and there are became much more manageable by people. Continuity of developing technology bring out semi-autonomous systems, Also it will provide us to see autonomous namely machines and devices decide by itself.

Transporting sector also receive its share from all these developments. Already automatic pilot system that take part in air transport can be considered as a product oh technology. Cars that park automatic or put on brake more quickly than human in case of danger(lower speed in the city), have semi-autonomous systems. In the future observe of these systems as autonomous is unavoidable. Autonomous cars are the vehicles which these technologies handle the fast and widespread in.

Autonomous cars are the vehicles that drives one point to another point without of human interference. These vehicles has developed hardware and software as distinct from normal vehicles. Thanks to these hardware and software design some functions(steering wheel, gas pedal, break pedal, gear... etc.) are controlled. Speed and direction of vehicle is determined by a central unit which has advanced algorithm. This central unit determines speed and direction according to incoming data form sensors and destination point that assign before by driver.

One of the method of upgrading the safety of vehicles driving is to increase the number and type of the sensors on the car. This brings an additional cost, increases the amount of received data and makes the system more complicated (especially if the system has such as camera or etc. on it as sensor). Central processing unit as well as making decision must process incoming data. But this load slows down the central unit and whole system. In such cases, two separate central unit is usually used. One of them benefit from the data collection process and to collect them at the same point. The other one calculates appropriate situations according to the proceed data, decides send commands to actuators with the help of this decision.

Autonomous vehicles, although these complex structures and untried technology, brings with it many advantages. Lack of human-specific features (sleep, fatigue, nervous, etc..), much faster response than human and calculation all the possibilities to choose the right decision according to the conditions are the most important features of autonomous automobiles. With the help of this features human-induced traffic accidents will be eliminated In addition, follow-up distance will go down to a few meters even at high speeds, thereby saving fuel and road capacity will be increased. Even people who are not able to use the car (old, alcoholic, etc.) will provide the opportunity for safer driving.

Subject of this study is vehicle mechatronics application for autonomous automobiles. The structure and development of autonomous vehicles in the study were examined as a priority. Later, a system is designed for the car which takes part in the university's autonomous car project. In the other chapter the devices used in the project, depending on the design, mechatronic montages and electrical connections were made.

Then, communication network protocols on the car (CAN) are examined. Electrical connections are made to access the data on this network. Then, appropriate communication protocol software developed and many data were obtained from the experiments.

In the next chapter steering systems which are on the automobiles are examined, a mathematical model of the project automobile's steering system is obtained. Depend on this model simulations are done. As the last, the real system experiment results are compared with the simulation results.

1. GİRİŞ

1.1 Otonom Kara Araçları Konusunda Yapılan Çalışmalar

Otonom kara araçları, bir noktadan başka bir noktaya insan müdahalesi olmaksızın kendi kendine karar verebilecek şekilde gidebilen araçlardır.

Otonom kara araçlarının geçmişi, 1939 yılında gerçekleşen FUTURAMA dünya fuarı ile başlamış sayılabilir. Ama asıl otonom araç tarihi 1977 yılında Japonya Tsukuba Makine Mühendisleri Labrotuvarları'nda özel ekipmanlarla beyaz şeritleri takip eden 30 km. hızındaki deneysel araçla başlamıştır.

1980 ile 1995 yılları arasında EUREKA Prometheus projesinde yer alan Mercedes-Benz takımı, görüntü tabanlı 'van' tipi[1] aracı trafiğe kapalı alanda 100 km. hıza çıkararak, 800 milyon Euro'luk bir projenin başlangıcını yapmıştır. Öte yandan aynı zaman dilimi içerisinde DARPA (Defense Advanced Research Projects Agency) AR&GE departmanında araçlarda LIDAR teknolojisini kullanarak sensör bazlı navigasyon sistemleri ile saatte 3 km hız ulaşmış, 600 m. yol kat etmiş ve otonom kara araçları listesine girmeyi başarmıştır.

1994 yılında Daimler-Benz ve Ernst Dickmanns, VaMP ve Vita-2 adlı robot araçları ile Paris'te, üç şeritli otobanda, yoğun trafikte, aralıklı insan müdahaleleriyle saatte 130 km'ye kadar hızlanabilen araçlarını geliştirmişlerdir. 1995 yılında, Dickmanns S-Class'i yeniden dizayn etmiş ve Münih'ten Kopenhag'a gidip gelerek, tam 1600 km'lik yolu, saatte 175 km'lik hızla, sadece 9 km'sinde müdahalede bulunarak %95 verim ile kat etmiştir.

1995 yılında Carnegie Mellon Üniversitesi'nin Navlab isimli projesi ' No Hands Across America ' (Amerikan'nın bir kıyısından diğerine, el değmeden) %98.2 verim ile 5000 km yol almıştır. Bu aracın direksiyonu neural network ile kontrol edilmişti. Ancak fren ve gaz donanımları sürücü kontrolündeydi. 2002 yılında DARPA tarafından insansız araç yarışması düzenleneceği ve birinci olan takıma 2 milyon dolar ödül verileceği duyuruldu. Buna göre ABD silahlı kuvvetleri için,

operasyonlarda kullanılan kara savaş araçlarının üçte birinin 2015 yılı itibari ile insansız olması hedefleniyordu. 2004 ve 2005 yıllarında otonom araçlar şehir dışı olarak tabir edilen bölgelerde yarıştılar. 2007 yılında ise arazide yapılan DARPA Grand Challenge yarışının yerini DARPA Urban Challenge almıştır. Şehir (caddeler, trafik işaretleri, vs.) şeklinde düzenlenmiş bir parkurda 3 Kasım 2007’de yapılan sürücüsüz, robotik arabalar trafik kurallarına uyararak, diğer arabalara, yayalara ve çarpılmaması gereken şeylere çarpmadan 60 millik parkuru altı saatin altında bitirmeye çalışmışlardır.

2010 yılında VisLab adlı şirketin düzenlediği yarışmada, VIAC (Vislab Intercontinental Autonomous Challenge), sürücüsüz dört araç 13000 km. yol katetmiş ve İtalya’dan Çin’e, Shanghai’daki Expo’ya uğrayarak yarışmayı bitirmişlerdir.

Ekim 2010’da ise Google’ın yaptığı insansız kara aracı San Francisco – Los Angeles arasında toplamda 230000 km. yol katetmiştir.

1.2 Otonom Otomobil İçin Sistem Tasarımı

Bu projedeki hedef insansız bir kara aracının, kendi yönünü ve güzergâhını kendisi tayin edebilecek şekilde, bulunduğu noktadan belirlenen konuma güvenli bir şekilde otonom olarak gitmesidir.

Bu hedefi gerçekleştirebilmek için öncelikle aracın bulunduğu konumu bilmesi gerekmektedir. Bunun için uydudan konum bilgisi alabilen bir cihaz olan GPS kullanılacaktır. Hareket halindeyken aracın şerit dışına çıkmaması için GPS’in konum hassasiyetinin santim mertebesinde olması gerekmektedir. DGPS adı verilen Differential GPS’ler daha hassas ölçümler yapabilmek için sıkça tercih edilen konum belirleyicilerdir.

Aracın hedef noktaya ulaşabilmesi için öncelikle en uygun rotanın belirlenmesi gerekmektedir. Bunun yapılabilmesi için GPS’den alınacak harita verisi üzerinden çeşitli algoritmalarla en kısa yol hesabı yapılmalıdır. Bu hesaplamalar aracın merkezi kontrol ünitesinde yapılacaktır.

GPS sisteminin çalışmadığı yada anlık olarak bağlantının kesintiye uğradığı koşullarda (tünel içinde ya da hava şartlarının kötü olması durumunda) aracın konum bilgisini kaybetmemesi gerekmektedir. Bunun için IMU(Inertial Measurement Unit) adı verilen bir cihaz kullanılacaktır. IMU anlık olarak 3 eksenindeki (x, y, z) ivmenin bulunmasını sağlayacak ve bu sayede de ölçülen ivmelerin zamana göre iki kere integrali alındığında x, y ve z eksenindeki konum değişikliği kolayca hesaplanabilecektir. Araç tekerlekleri kaysa bile aracın hangi ekseninde ne kadar ilerlediği kolaylıkla bulunabilecektir. Bu değerler konum bilgisi elde etmenin yanısıra araç dinamiği ile bilgileri de elde etmeye yardımcı olacaktır.

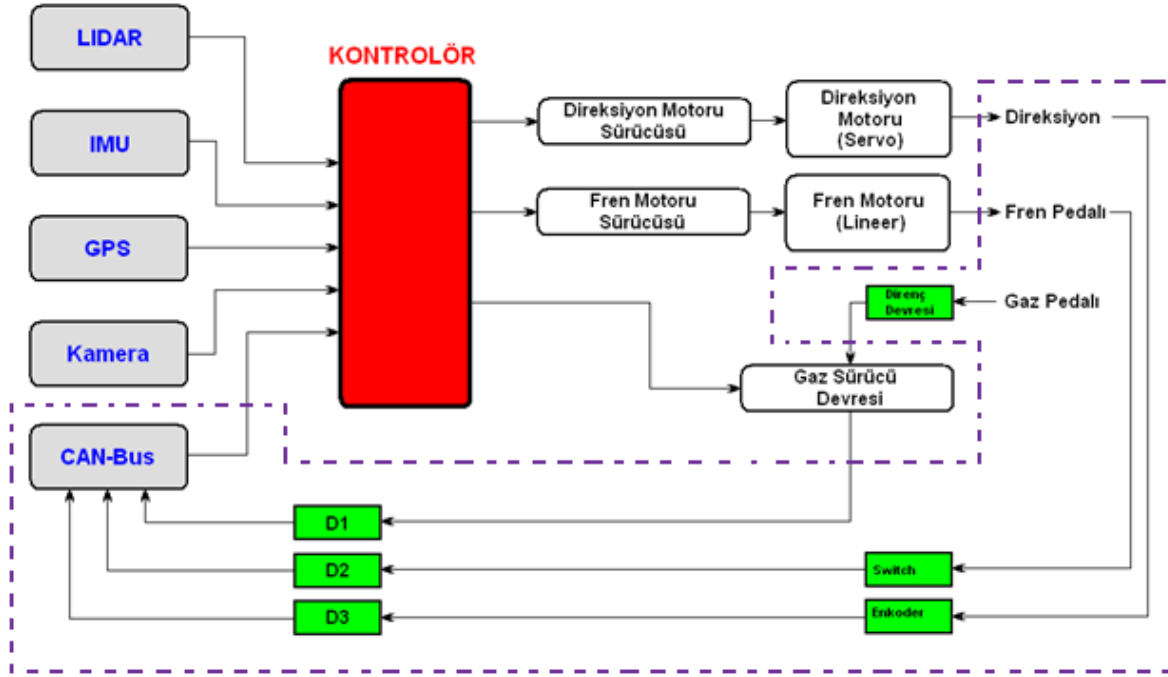
Aracın hedef noktaya hareketi esnasında güvenli bir sürüş gerçekleştirebilmesi, gerektiğinde durması ya da hareket edebilmesi için çevresini algılayabilmesi gerekir. Bunun için kısa mesafede etkili ölçüm yapabilen geniş spektrumlu LIDAR(Light Detection and Ranging) adı verilen bir cihaz kullanılacaktır. Bu cihaz uzaklığı ölçülecek nesne ya da yüzeye gönderilen lazer darbesinin gönderiliş zamanıyla, nesneye çarpıp tekrar kaynağa geri dönme süresi arasındaki farka göre mesafeyi belirler.

Her ne kadar LIDAR kısa mesafede etkili ölçümler yapabilse de araç yüksek hızlara çıktığında ölçüm mesafesi yetersiz kalacaktır. Daha uzun mesafelerde algılama yapabilmek için RADAR kullanılması gerekmektedir. RADAR LIDAR'a göre daha dar bir skalda tarama yapsa da LIDAR'a göre daha uzak mesafelerde çalışabilmektedir.

Bunlara ek olarak trafik ışıklarındaki renklerin algılanabilmesi, öndeki araçların fren lambalarının yanıp yanmadığının kontrol edilebilmesi, tabelaların okunabilmesi vb. durumlar için kameranın da kullanılması gerekmektedir. Kamera aynı zamanda görüntü işleme yöntemiyle şerit takibi yaptırılarak yol takibinde GPS'le birlikte yardımcı eleman olarak kullanılabilir.

Aracın otonom hareket edebilmesi için harici sensörlerden alınan verilerin yanısıra araç hızı, direksiyon açısı, direksiyon dönüş hızı gibi bilgilere de ihtiyacı vardır. Gelişmiş otomobillerde bu bilgiler aracın dahili sensörlerinden elde edilir ve diğer mekanizmaların yararlanabilmesi için CAN-Bus (Controller Area Network) adı

verilen veri hattında yayınlanır. İstenilen verinin elde edebilmesi için CAN hattına erişilmesi ve veriyi içeren mesajın ID numarasına göre süzülmesi gerekmektedir.



Şekil 1.1 : Sistemin genel yapısı.

Araç üzerine yerleştirilmiş bütün sensörlerden gelen verilerin merkezi bir noktada toplanması ve işlenmesi gerekmektedir. İşlenen bu veriler sonucunda aracın hangi doğrultuda ve hangi hızda hareket edeceği merkezi birim tarafından belirlenir. Bu birim sistemde PLC olarak tasarlanmıştır. Merkezi birim tarafından karar verildikten sonra aracın istenilen konuma gidebilmesi için en azından temel fonksiyonların (gaz, fren ve direksiyon) elektronik olarak kontrol edilebilmesi gerekmektedir. Gaz kontrolü için pedal ile aracın elektronik sistemi arasına girilerek, pedaldan çıkan sinyale benzer bir sinyal üretilip gaz kontrolü yapılmalıdır. Fren kontrolü içinse lineer bir motor kullanılacaktır. Lineer motor ileri hareket ettiğinde fren pedalına basılacak, geri hareket ettiğinde ise pedala geri çekecektir. Böylece fren motoruna bir konum kontrolörü tasarlanarak istenilen oranda basılma sağlanacaktır. Fren motorunun hareketi merkezi işlem birimi tarafından direkt olarak değil, aradaki sürücü bir devre ile kontrol edilecektir. Direksiyonun hareket edebilmesi için direksiyon miline bir dişli sistemiyle beraber servo motor monte edilecektir.

1.3 Kullanılan Aracın Özellikleri

“Otonom Araç Projesi” kapsamında TOFAŞ tarafından üzerinde çeşitli çalışmalar yapılabilmesi için “FIAT LINEA” marka bir binek otomobil verilmiştir. Araç 2007 model, 1.3lt 75 hp JTD dizel motorlu olup otomatik viteslidir ve aynı zamanda araçta ESP de mevcuttur.



Şekil 1.2 : Projede kullanılan araç.

ESP(Electronic Stability Program), “oversteering” olarak adlandırılan aracın direksiyonunun ani ve hızlı bir şekilde çevrilmesi durumlarında ya da tekerleklerdeki devir sayısı farklı olduğunda aracın kayarak kontrolden çıkmasını ve savrulmasını önler. **ESP sistemi** tekerlek devir sensörleri, direksiyon açısı sensörü, devir ve ivmelenme dengeleyici sensör, yanal hız algılayıcısı, motor kontrol ünitesi, fren basıncı algılayıcısı gibi sensörlerden oluşur . Bu sayede bu bilgilere ulaşmak için harici sensörler kullanma gereksinimi ortadan kalkmıştır. Araç ile ilgili diğer teknik bilgiler Ek-A’da verilmiştir.

1.4 Kullanılan Donanım Parçalarının Genel Özellikleri

1.4.1 PLC

Kullanılan PLC bir ana modül (X20CP1483-1) ve 4 yardımcı modülden oluşmaktadır. Ana modül işlemci ve ona bağlı çevresel üniteleri içerir. İşlemci modülün hızı 100 MHz'dir. 32 MB dinamik ve 128 KB static ram barındırmaktadır. Bu hafıza elemanlarının yanısıra PLC ilk programlamasının yapılabilmesi için harici bir compact flash girişi vardır. Ayrıca haberleşme için iki adet USB ,bir RS-232, bir ethernet ve bir powerlink arayüzü vardır.



Şekil 1.3 : PLC X20 ana modülü.



Şekil 1.4 : PLC X20 yardımcı modülleri.

Kullanılan yardımcı modül şekil 1.4'de gösterildiği sırayla haberleşme modülü, analog modül, dijital çıkış modülü ve dijital giriş modülünden oluşmaktadır. Haberleşme modülü bir CAN arayüzüne sahiptir. Ana modülde CAN arayüzü olmadığından diğer sistemlerle haberleşebilmek için bu modül eklenmiştir.

1.4.2 Servo motor sürücüsü

Direksiyon miline baęlı olan servo motorla PLC arasında sürücü görevini üstlenen modüldür. 3 fazlı iki ayrı motoru sürebilme özelliğine sahiptir.PLC ile powerlink arayüzüyle haberleşmektedir. Besleme voltajı DC 24 Volt olmasına rağmen motor besleme voltajı DC 80 Volt'a kadar verilebilir.



Şekil 1.5 : Servo motor sürücüsü.

1.4.3 Servo motor (8LVA22.R0030D000) ve redüktör

Servo motor direksiyon sistemine hareket veren elemandır. Bu motor 3 fazdır. Ortalama hızı 3000 devir/dakika ve torku 0.65 Nm'dir. Sürücüsüne konum bildiren bir enkodere sahiptir.



Şekil 1.6 : Servo motor.

Motor torkunun mile aktarılması için bire beş oranında bir redüktör kullanır. Redüktör giriş ve çıkış mili aynı eksen üzerindedir ve çıkış milinde 172 Newton'a kadar tork verebilir.



Şekil 1.7 : Redüktör.

1.4.4 LIDAR(Light Detection and Ranging)

LIDAR, lazer ışın demetleri kullanarak mesafe ölçümü yapan bir cihazdır. Projede kullanılan LIDAR'ın (SICK LMS221) özellikleri aşağıdaki gibidir:

- 180 ° tarama açısı
- 0.5° veya 1° tarama çözünürlüğü
- RS-232 ve RS-422 haberleşme arayüzü
- 9.6 Kb, 19.2 Kb, 38.4 Kb, 500 Kb bağlantı hızı seçenekleri
- Santimetre veya milimetre modu (80 metre ya da 8 metre ölçüm mesafesi)



Şekil 1.8 : SICK LMS221.

LIDAR taradığı alanın her derecesine (çözünürlüğe göre 0.5° ya da 1°) bir lazer ışını gönderir ve bu geri yansımalarını bekler. Lazer ışını eğer ölçüm mesafesinde (8 metre ya da 80 metre) bir cisme çarparsa kaybağa geri yansır. LIDAR giden ve gelen ışın arasındaki süreyi ölçer ve bu farka göre cismin uzaklığını saptar.



Şekil 1.9 : Tarama açısı.

Sistemde LIDAR ölçtüğü ham verileri RS-422 arayüzüyle 500 Kb hızında işlemcili bir karta gönderir. Bu kartta işlenen veriler anlamlı hale getirilir ve RS-232 arayüzü ile PLC'ye gönderilir.

1.4.5 CAN kartı (Kvaser USBcan II HS/LS)

CAN kartı araç üzerindeki CAN-C hattındaki mesajları okumak için kullanılmıştır. Yüksek hızlı ve düşük hızlı olmak üzere iki kanal girişine sahiptir. Kartın bir özelliği de okuduğu mesajlarda bir bozulma olup olmadığını kontrol edebilmesidir. Bu gibi durumlarda kart hata biti üretir ve CAN hattı üzerinde yayımlar.



Şekil 1.10 : CAN hattı erişim kartı.

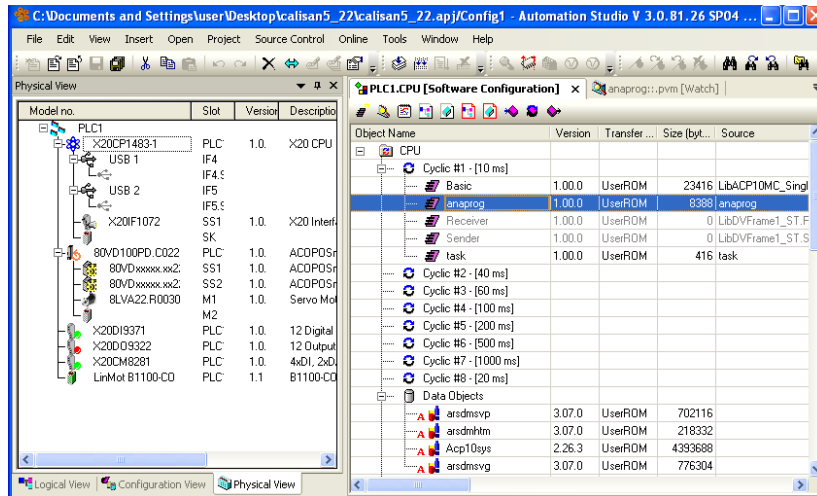
1.4.6 Diğer elemanlar

Otonom aracın temel fonksiyonlarını kontrol edebilmek için yukarıda anlatılan donanım parçalarının yanısıra IMU(Inertial Measurement Unit), lineer motor ve lineer motor sürücüsü kullanılmıştır ancak bu tez çalışması kapsamında ele alınmamışlardır.

1.5 Kullanılan Yazılımlar

1.5.1 Automation Studio

Sistemin kontrolörü olarak kullanılan PLC Automation Studio programı ile Basic programlama dili ile programlanmıştır. PLC'nin ana modülü ve ek modüllerinin başlangıç ayarları, kurulumu yine bu programda yapılmıştır.



Şekil 1.11 : Automation Studio programı örnek görüntüsü.

Programın ve kullanılan PLC'nin en önemli özelliklerinden biri her bir görev için farklı sayfalarda ayrı bir proses oluşturabilmesi ve her bir işlem döngüsüne ait period sürelerinin ayrı ayrı ayarlanabilmesidir. Bu özellik sayesinde PLC içerisinde yazılan farklı görevler aynı anda gerçek zamanlı bir şekilde işleyebilir.

1.5.2 Kvaser CANKing

Aracın CAN hattına erişmek için kullanılan donanım parçasının orijinal yazılımıdır. CAN hattı üzerindeki verilerin tümü program sayesinde bilgisayar ortamına aktarılıp, kaydedilebilir. Anlık olarak veriler süzülürse araç hızı, direksiyon açısı, yaw-rate sensör datası, klima, kapı kilitleri, ABS, direksiyon dönüş hızı gibi bilgiler CAN hattına girilerek okunabilir.

1.5.3 Visual Studio

CAN hattına erişebilmek ve PLC'ye bu hat üzerinden gelen bilgileri aktarabilmek için bilgisayar üzerinde bazı programlar çalıştırılmıştır. Bunlar C programlama dili kullanılarak bu editör program aracılığı ile yazılmış ve derlenmiştir.

1.5.4 Matlab

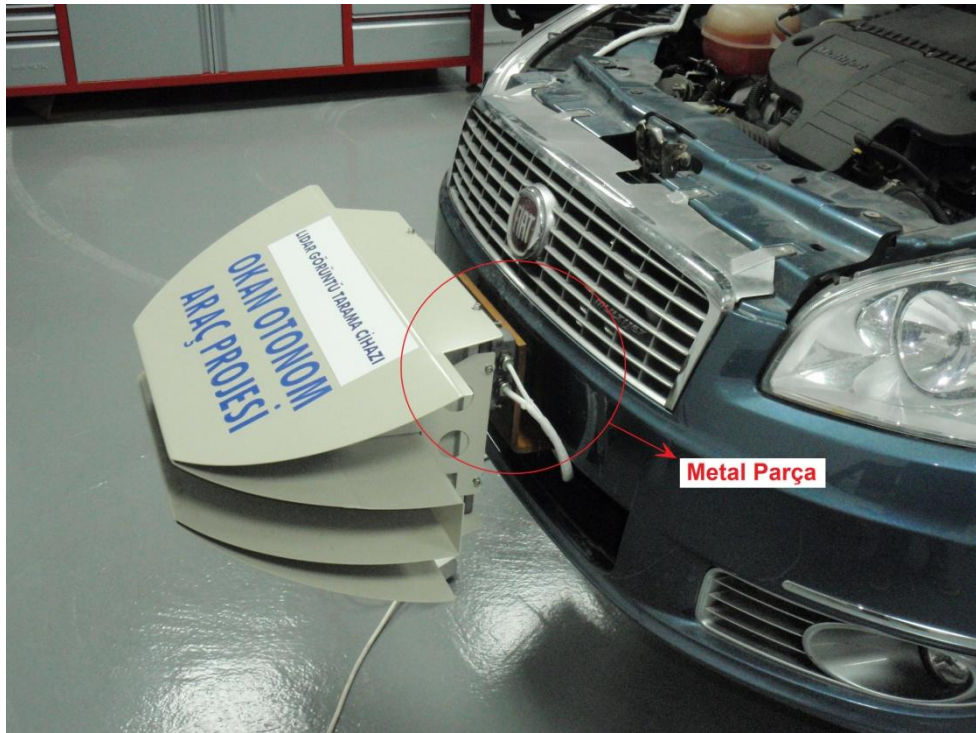
Matematik modeli oluşturulan direksiyon sistemine ait simülasyonlar MATLAB programında kurulmuş ve incelenmiştir. Bunun yanında aynı sistem için tasarlanan kontrolörün katsayıları yine bu programa yaptırılmıştır.

2. DONANIMLARIN MEKANİK MONTAJI VE ELEKTRİK BAĞLANTILARI

2.1 Mekanik Montaj

2.1.1 Lidar

Aracın önündeki nesnelere sezilmesini, ne kadar uzağında bir cisim olduğunu algılayabilmesi için Lidar adı verilen bir cihaz kullanılmıştır. Araç üzerinde yapılan ilk çalışmalardan biri Lidar'ın mekanik montajı olmuştur. Önde bulunan plastik tampon Lidar'ın ağırlığını kaldıramayacağı için, bu donanım parçası aracın asıl tamponuna, içeride bulunan metal tampona, "U" şeklinde metal bir parça ile tutturulmuştur. Lidar'a ait haberleşme ve besleme hatlarından oluşan elektrik kabloları ise aracın kaputundan geçirilerek ve uygun noktalara sabitlenerek araç içine taşınmıştır.



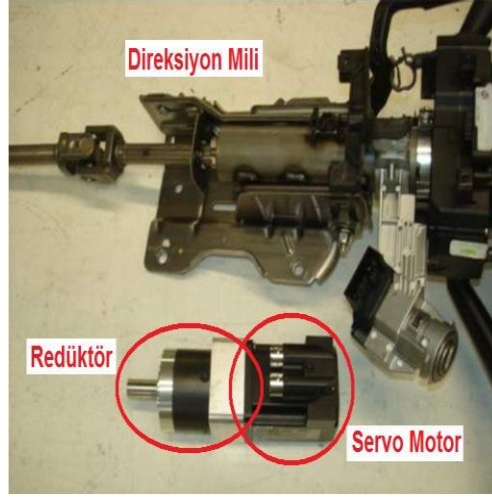
Şekil 2.1 : Lidar'ın montajı.

2.1.2 Servo motor

Direksiyonun otonom olarak hareket edebilmesi için direksiyon simidinin altına, direksiyon miline paralel olarak bir servomotor yerleştirilmiştir. Araba çalışır durumdayken, hidrolik sistemin de desteği ile direksiyonun dönmesi için gereken kuvvet bir forcemetre ile ölçülmüş ve yaklaşık olarak 2-3 Nm olarak tespit edilmiştir. Motor seçimi yapılırken ölçülen bu değerin tekerlerin yere temas ettiği yüzeyin sürtünme katsayısına bağlı olduğu da göz önünde bulundurulmuştur. Direksiyonun dönüş hızının bir servo motor kadar hızlı olması gerekmediğinden ve dönüş için gereken kuvveti sağlamak amacıyla bir redüktör kullanılmıştır. Bu redüktör ile dönüş miktarı 5'e bölünürken, 1 adımdaki çözünürlük ve kuvvet ise 5 kat arttırılmıştır. Servo motor ve ona bağlanan redüktör direksiyonun alt kısmına, mile paralel olarak yerleştirilmiş ve sabitlemiştir. Mil ile arasındaki kuvvet alış verişi, 1/1 oranında ve diş sayıları aynı olan 2 dişli ile sağlanmıştır.



Şekil 2.2 : Servo motorun montajı.



Şekil 2.3 : Servo motor ve direksiyon mili(demonte).

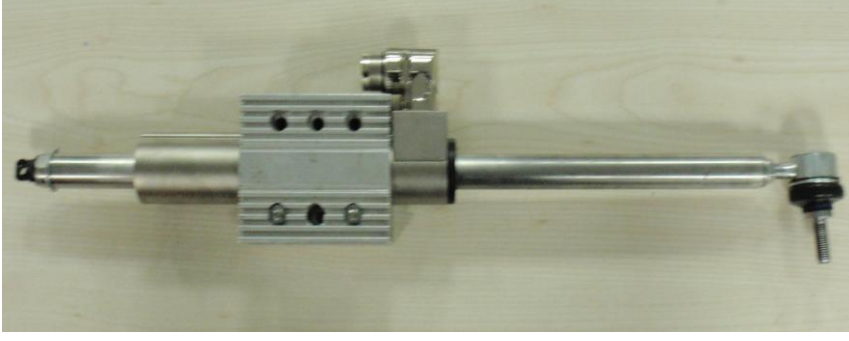


Şekil 2.4 : Servo motor ve direksiyon mili.

Burada mil dişlisi ile redüktör dişlisinin oluşturduğu kısım dışarıdan gelecek yabancı maddelerin aralarına sıkışma olasılığına karşı bir kapakla korumaya alınmıştır. Ayrıca tüm bu aksamlar istenildiğinde demonte edilebilecek şekilde tasarlanmıştır.

2.1.3 Lineer motor

Aracın fren sisteminin otonom olabilmesi için lineer motor kullanılmıştır. Bu motor fren sistemine elektronik olarak müdahale etme şansı olmadığı için fren pedalına direkt etki edecektir.



Şekil 2.5 : Lineer motor.

Pedal takımının hemen önüne uzunlamasına yerleştirilen lineer motor aracın tabanına bir kelepçe sistemi ile tutturulmuştur. İstenildiğinde aracın manuel kontrol ettirilebilmesi için fren pedalının dışarıdan müdahaleye açık olması gerektiğinden, lineer motor fren pedalına değil, ona paralel hareket eden rijit uzantısına etki etmektedir. Böylece fren pedalı dışarıdan kullanımlara da açık olacak, lineer motor tarafından sürekli işgal edilmeyecektir.



Şekil 2.6 : Lineer motorun fren pedalına monte edilişi.

Bu noktada fren pedalının tam olarak lineer hareket etmemesinden kaynaklı bazı kasılma sorunları ortaya çıkmıştır. Fren pedalı tam olarak olmasa da açısız bir harekete benzer şekilde hareket etmektedir. Fakat lineer motorun böyle bir yeteneği yoktur. Bu sorunu aşabilmek için sistemin iki noktasına mafsal yerleştirilmiştir.

Birinci mafsalsal motorun merkezine bağlanmış ve motora küçük bir açı da olsa kendi ekseninde dönme yeteneği sağlamıştır. Diğer mafsalsal ise lineer motora ait rotorun uç noktası ile pedal uzantısının kesiştiği noktaya monte edilmiştir. Bu şekilde fren pedalı ile rotor arasında rijit bir bağlantı olmamış ve rotorun stator içindeki kasılma ihtimali de engellenmiştir.



Şekil 2.7 : Mafsalsal 1.

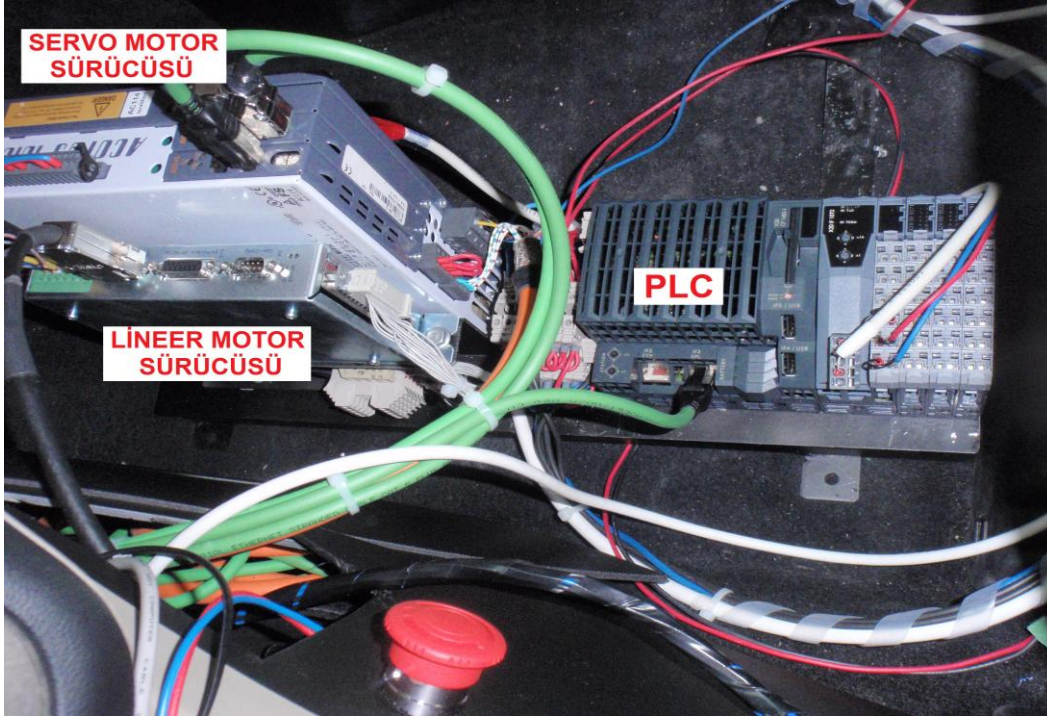


Şekil 2.8 : Mafsalsal 2.

Ayrıca lineer motorun araç tabanına montajından dolayı üzerine basma, sıvı damlama gibi riskli olabilecek dış etkenlere karşı pleksiglasdan yapılan kapakla korumaya alınmıştır.

2.1.4 Sürücüler ve PLC

Kullanılacak olan motorları sürececek ve kontrol edecek donanımları içeren bu kısım küçük bir pano şeklinde yolcu koltuğunun ön-alt kısmına yerleştirilmiş ve hareket etmemesi için araç tabanına sabitlenmiştir.



Şekil 2.9 : PLC ve sürücüler.

2.1.5 IMU

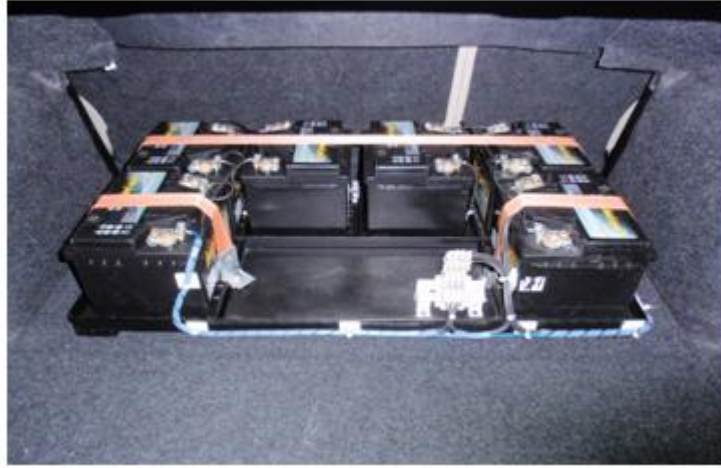
Aracın 3 eksenindeki ivmesinin bulunmasını ve bunun yardımıyla da aracın konumunun tespit edilebilmesi için IMU adı verilen bir cihaz kullanılmıştır. Aracın kendine ait IMU'su olmasına rağmen, daha etkin ölçümler alabilmek için hassasiyeti daha yüksek olan harici bir IMU daha kullanılmıştır. Bu donanım parçası aracın ağırlık merkezine yakın bir nokta olan, vitesin ön kısmına, aracın kendi IMU'sunun hemen üstüne yerleştirilmiş ve araç üzerine sabitlenmiştir. Bu sayede 3 eksende oluşabilecek pozitif ya da negatif ivmelerin daha etkin bir şekilde ölçülmesi ve değerlendirilmesi sağlanmıştır. Ayrıca cihazın kontrol merkezine yakın konulması elektriksel sinyallerin bozulmaması açısından büyük önem taşır.



Şekil 2.10 : IMU.

2.1.6 Besleme

Araç üzerine yerleştirdiğimiz tüm donanım parçalarının enerji ihtiyacı arabanın bagajına yerleştirilmiş ve birbirine seri olarak bağlanmış 6 adet aküden sağlanmıştır. Akülerin hareket halindeyken sabit kalabilmesi ve dağılmaması için bir kafes yaptırılmış ve kemerlerle kafese tutturulmuşlardır.



Şekil 2.11 : Aküler.

2.1.7 Kablolama ve sigortalar

Kontrol ünitesinin ve motor sürücülerinin yerleştirildiği yolcu koltuğunun ön kısmı merkez olarak kabul edilmiş, bağlantı kabloları bu noktaya 3 ayrı yerden getirilmiştir. Bunlardan ilki tüm güç sistemin yerleştirildiği bagaj kısmıdır. Buradan

ıkan gc hatları aracın plastik koruma kapaklarının altına gizlenerek ve sabitlenerek merkez noktaya tařınmıřtır.

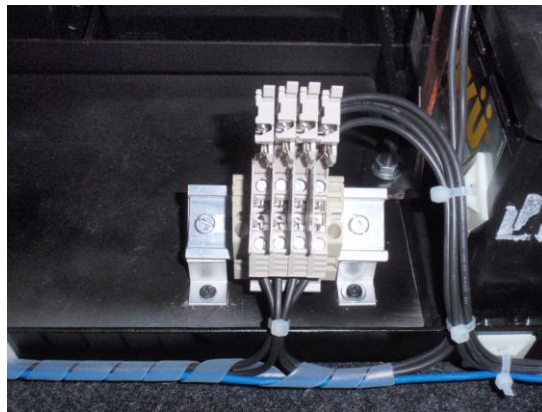


řekil 2.12 : Besleme kablolarının geiři.

Diđer bir kablo grubu ise Lidar ile veri alıř veriřini ve Lidarın beslemesini oluřturan gruptur. Bu hatlar koruma kılıfı ile kaplandıktan sonra aracın n kaputunun iinden geerek, sađ kapının olduđu blmde ieri alınmıřtır.

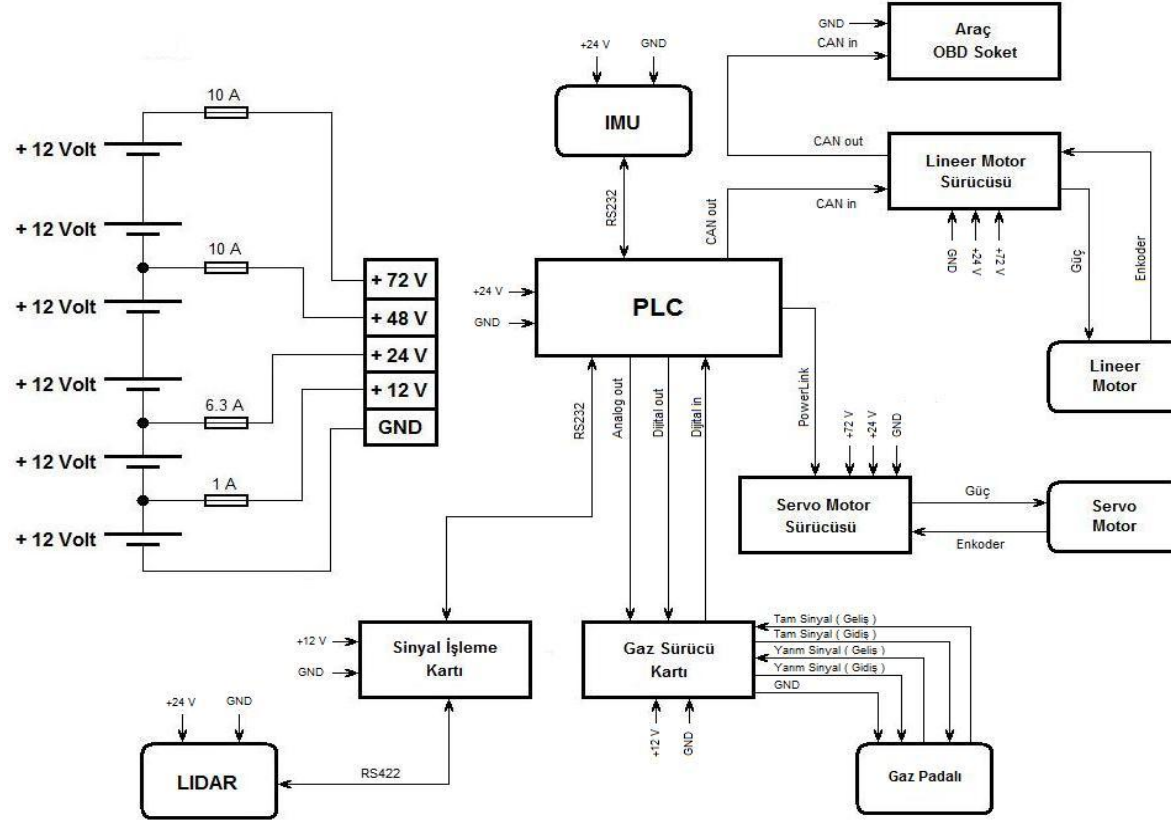
Son kablo grubu ise src koltuđunun n kısmına yerleřtirilen lineer motordan, direksiyonun altına sabitlenen servo motordan ve vitesin n kısmına yerleřtirilmiř olan IMU'dan gelen kablolardır. Buradaki hatlar vites kolunun bađlı olduđu plastik kasanın altından geirilerek merkez noktaya getirilmiřtir.

Sistemimizi besleyecek olan hatları koruyacak olan sigortalar ise, enerji hatlarının ara iinde kısa devre ihtimali de dřnlerek, enerji ıkıř noktasına ok yakın bir yere, aklerin yanına, bagaja yerleřtirilmiřtir.



řekil 2.13 : Sigortalar.

2.2 Elektrik Bağlantıları



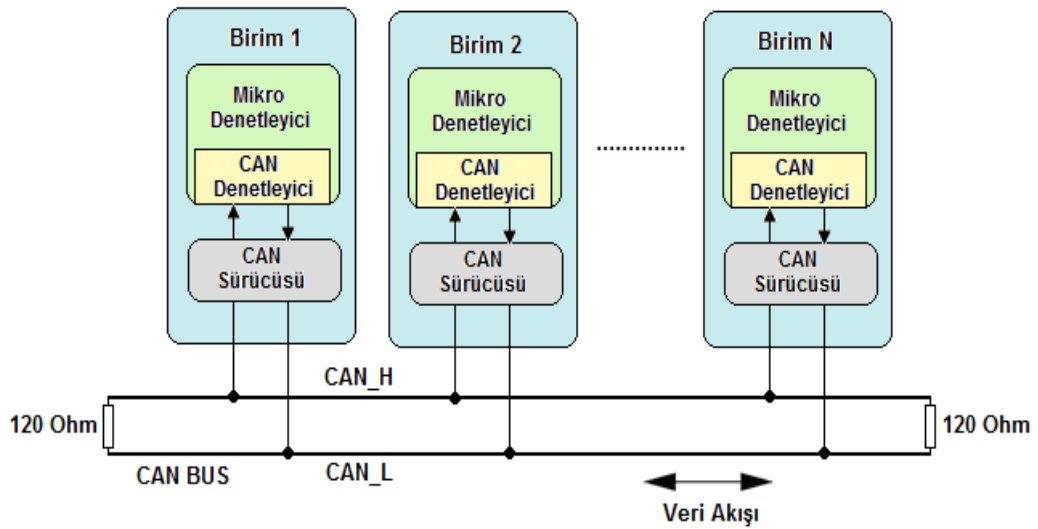
Şekil 2.14 : Elektrik bağlantı şeması.

3. ARAÇ ÜZERİNDEKİ CAN HATTINA ERİŞİM

3.1 CAN(Controller Area Network)

CAN protokolü araç içinde bulunan farklı üniteleri minimum sayıda kablo ağıyla birbirine bağlamak için 1983 yılında Bosch Corporation tarafından başlangıçta sadece otomotiv sektörüne yönelik tasarlanmış bir haberleşme protokoldür. Daha sonraları bu yapının sağladığı birçok avantaj ve kolaylık nedeniyle diğer uygulama alanlarında da kullanılmaya başlanmış ve günümüzde de yaygın olarak kullanılmaktadır.[2]

CAN ağında haberleşme kurulacak birimler arasındaki bağlantı yalnızca 2 hat(CAN-Low ve CAN-High) kullanılarak yapılır. Aşağıdaki şekilde de görüldüğü gibi sisteme eklenecek her modül hatta paralel olarak bağlanır. Teoride bağlanacak birim sayısında bir sınır yoktur, fakat pratikte mesafenin uzamasıyla meydana gelebilecek gecikme zamanları ya da hattın besleyebileceği sınırlı elektrik akımı dolayısıyla bu sayı sonsuz değildir.



Şekil 3.1 : Genel CAN yapısı [3].

Can'i diğ er haberleş me protokollerinden ayıran en önemli özellik istasyona dayalı ya da adres temelli değı l, mesaj temelli çalış masıdır.[4] Bu çalış ma prensibinde mesajı üreten modül hatta veriyi gönderdiğ inde hangi birim ya da birimlerin bu mesajı aldığı ve kullandığı önemli değı ldır. Karşı taraftan tek bir modülün bu mesajı okuması ve ardından okundu bilgisini geri göndermesi mesaj üreten modül için yeterlidir.

Her modül ürettiğı mesajı hat boş olduğ unda hatta yazar. Diğ er tüm birimler de bu mesajı görür, fakat sadece kullanılacak verileri içeren mesajları birim içine alır ve iş ler. İş ine yaramayacak olanlarla ise ilgilenmez. Bunu her mesajın başında yer alan ve sadece o mesaja özgü ID numaraları sayesinde gerçekleştirir. Her modülde modülün istediğı ya da yayınladığı bilgilere ait mesajların ID numaraları kayıtlıdır. Hat üzerine mesaj yazan birimler o mesajın başına ID numarasını mutlaka eklerler. Mesajı okuyacak olan birimler ise hat üzerinde gördüğü mesajları içinde bulunan filtreler yardımıyla yine ID numarasına göre süzer ve kullanır. Yani tüm veriler tüm ünitelere ulaşır, fakat her mesaj her ünite tarafından kabul edilmez.

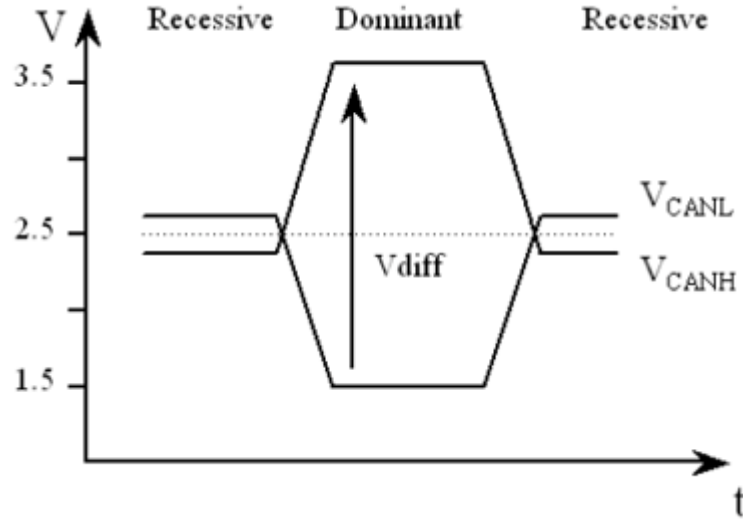
Böyle bir yapının oluşabilmesi için CAN hattı 3 farklı katmana ayrılabilir.[2]

Bunlar:

- Nesne Katmanı
- İletim Katmanı
- Fiziksel Katman

Fiziksel katman üniteler arasındaki elektrik hatlarından meydana gelir ve görevi bilgilerin diğ er ünitelere ulaşmasını sağlamaktır. İletim katmanı ise veri transferin nasıl yapılacağını belirleyen kısımdır. Mesaj önceliğinin belirlenmesi, hata kontrolü, hata sinyalleşmesi, mesaj kontrolü bu katmanın başlıca görevlerindedir. Nesne katmanı bu yapıyı belirleyicisi konumundadır. Hangi mesajın transfer edileceğı, hangi mesajın alınacağı gibi görevler bu katmana aittir.

CAN hattındaki lojik seviyelerin (Lojik 1, Lojik 0) gerilim deęerleri bilinen çoęu dijital sinyalin biraz dıřına çıkmaktadır. Ařaęıdaki řekilde de grldęi gibi CAN-High ve CAN-Low uęlarındaki sinyallerin ikisi de GND'nin stnde yer alır. Hat zerindeki bilginin Lojik 0 ya da Lojik 1 olduęu GND referans alınarak deęil, CAN-High ve CAN-Low uęlarındaki gerilim deęerlerinin birbirlerine gre durumuna bakılarak belirlenir. Bylece sinyaller dıřarıdan gelecek manyetik parazitlerin etkilerine karřı da korunmuř olur.



řekil 3.2 : CAN-High ve CAN-Low iin gerilim deęerleri [5].

Burada dikkat edilmesi gereken bir bařka nokta hattaki her nite iin Logic 0 seviyesinin Lojik 1 seviyesine gre baskın olmasıdır. İki farklı birim hatta aynı anda farklı lojik seviyeler uyguladıęında Lojik 0 deęeri hat zerinde etkin olacaktır ve dięer tm modller hattı Lojik 0 olarak okuyacaklardır.

Bir bitin yapısını zaman ekseninde inceleyecek olursak, ncelikle kuantı kavramına deęinmek gerekir. CAN donanımlarında baud rate n blcs vardır, fakat sadece kuantı denilen kk bir zaman dilimi retmek iin kullanılır. Bir bitlik sre cihazların uzaklıęına baęlı olarak 8-25 kuantı sresine karřılık gelir. Kuantı miktarı arttıka haberleřme hızı dřer, ama aynı zamanda haberleřme kurabileceęimiz mesafe miktarı artar.

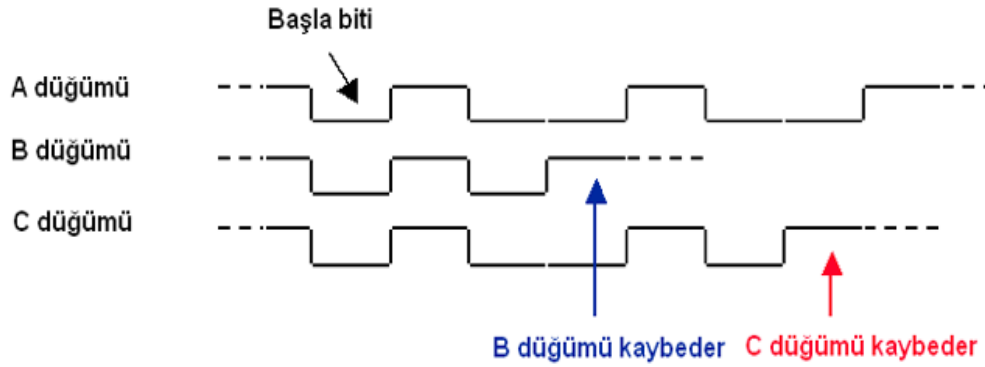
Hattın uzunluđuna gre bit bařına deđiřebilen kuanta detine rađmen, genel olarak bir bitlik yapıyı zaman ekseninde 3 ayrı kısma ayırabiliriz. Birinci kısım senkronizasyon kısmıdır ve sabit olarak 1 kuanta uzunluđundadır. İkinci kısım TSeg1'dir. Gnderilen bitin alıcıda okunduđu an TSeg1'in sonunda yer alan rnekleme noktasıdır. Tseg2'deki kuantalar esnasında ise alıcılardan mesajı gnderen birime geri bildirimler olabilir. Mesajı yazan birim her biti yazdıktan sonra, aynı bitin TSeg2 blmnde tekrar bir okuma yapar ve yazdıđu bitte bir deđiřim olup olmadığını gzlemler.



řekil 3.3 : Bir bitin zaman ekseninde incelenmesi [5].

Bu haberleřme yapısının mesaja dayalı olmamasının bir bařka getirisi de hattı kontrol eden merkezi birime gerek olmamasıdır. İki veya daha fazla birim aralarındaki mesaj iletimini dođrudan, ynetici birim olmadan yapar[2][3]. Haberleřme hızları nceden belirlenen modller, hat zerinde yayımlanan bitlerin deđiřimlerine ve durumlarına gre senkronizasyonlarını kurarlar ve iletiřimlerini gerekleřtirirler.

Lojik 0'ın Lojik 1'e göre baskın olmasının bu haberleşme stili için önemli avantajları vardır. Bu avantajlardan ilki mesaj önceliğidir. Her birim hatta mesaj yollamadan önce yolun boş olup olmadığını kontrol eder. Eğer hat boş ise kendi mesajını hatta yazar. Yolun başka bir mesajla dolu olması durumunda ise boşalmasını bekler. Çoğu haberleşme protokollerinde iki düğüm hat boşken aynı anda hatta kendi mesajını yazmaya kalkarsa bu durum çakışma(collision) olarak algılanır ve sinyali çakışan birimler datalarını o an yollamayı durdururlar. Rastgele (t_1 , t_2) süreler beklerler ve sonrasında tekrar data göndermeyi denerler. Fakat bu tip yapılarda bu andan itibaren hat boş bile olsa datalar bekletilebilir. CAN protokolünde ise aynı anda birden fazla ünite hatta mesaj yazmaya başladığında baskın seviye lojik 0 olduğundan düşük ID'li mesajı yazan düğüm hattı ele geçirir ve diğer üniteler aradan çekilerek kendi mesajlarını tekrar göndermek için yolun boşalmasını beklerler.



Şekil 3.4 : Üç modülün aynı anda hatta veri yazması [5].

Yukarıdaki şekilde de görüldüğü gibi 3 birim aynı anda hatta veri yazmaya başladığında aynı değerleri yazdıkları sürece 3'ü de veri yazmaya devam ederler. Kısa bir süre sonra mesajdaki bitler farklılaşmaya başlayacaktır. Böylece aynı anda çekinik ve baskın bit hatta yazıldığında çekinik bite sahip birim yazmayı durduracaktır. Her mesajda start bitinden sonra mesajın ID numarası geldiği ve her mesajın da ID numarasının farklı olduğu göz önünde bulundurulursa, mesaja ait ID hatta yazıldığında geride sadece bir birim kalacaktır, bu da en küçük ID'ye sahip olan(arbitration) öncelikli mesajdır.

CAN'de yer alan mesajlar amaçlarına göre 4'e ayrılabilir.[3]

Bunlar:

- Data Frame (Veri mesajı)
- Remote Frame (İstek mesajı)
- Error Frame(Hata mesajı)
- Overload Frame(Aşırı yük mesajı)

Data Frame

Veri göndermek isteyen birim tarafından hatta yazılan mesaj tipidir.

Remote Frame

Her hangi bir veriye ihtiyaç duyan birimin veri talebi için hatta yazdığı mesaj tipidir. İçerisinde veri içermez.

Error Frame

Hatta bağlı herhangi bir birim hat üzerinde akan her hangi bir mesajda hata olduğunu fark ettiğinde hatta yazdığı mesaj tipidir. Bu mesajı alan tüm birimler hatalı mesajı yok sayarlar.

Overload Frame

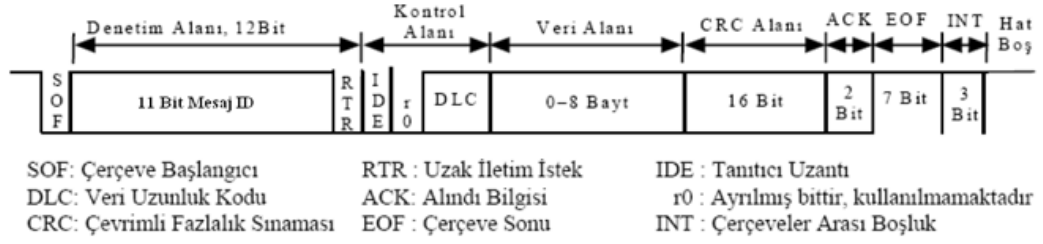
Çok meşgul olan bir birim tarafından kendisinin kullanacağı data frame'i geciktirmek için hatta yazılan mesaj tipidir. Fakat günümüzde CAN birimlerinin hızı çok yüksek olduğu için hat üzerinde bu tip mesajları görmek çok zordur.

Mesaj ID uzunluğuna bakımından CAN mesajlarını 2'ye ayırabiliriz. Bunlar **Standart Form** ve **Genişletilmiş Form**'dur.

Standart form CAN 2.0A mesajı olarak da bilinir ve mesajın ID numarası 11 bitten meydana gelir. Mesaj içinde ID numarası bir bütündür.

Genişletilmiş form CAN 2.0B mesajı olarak da bilinir ve mesajın ID numarası 29 bitten meydana gelir. Ancak ID numarası standart formda olduğu gibi bir bütün değildir, 2 parçadan oluşur. Bunlardan ilki yine standart formdaki gibi start bitinden sonra gelir ve 11 bittir. İkinci parça SRR ve IDE bitlerinden sonra gelir ve 18 bittir.

Aşağıda örnek bir standart form veri mesajının yapısını verilmiştir.



Şekil 3.5 : Örnek veri mesajı(Standart format)[6].

SOF: Diğer birimlere mesajın başladığını bildirir.

ID: Sadece o mesaja ait bir numaradır. Diğer tüm birimlere mesajın hangi datayı taşıdığını bildirir. Bu sayede diğer birimler mesajı kabul ederler.

RTR: Mesajın data mesajı mı(Lojik 0), yoksa istek mesajını olduğunu bildirir.

IDE: Mesajın Standart Formda mı yoksa genişletilmiş formda mı olduğunu bildirir.

r0: Ayrılmış bittir, bu yapıda kullanılmamaktadır.

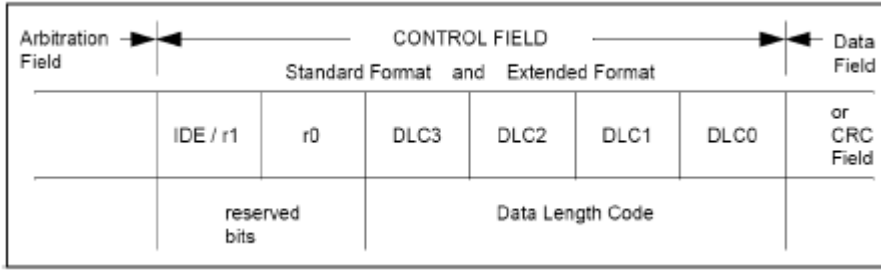
DLC: Her mesaj 0-8 Bayt arasında veri taşıyabilir. DLC kendinden sonra gelen veri alanının uzunluğunu verir. Aşağıda verilen tablodaki gibi kodlanmaktadır.

CRC: Start bitinden itibaren kendinden önce gelen dataya göre kodlanmış 15 bitlik bir bilgidir. 16. bit mesajı gönderen tarafından çekinik bit olarak yazılır. Aynı mesajı okuyan birimler 15 bitlik dataya göre mesajın bu kısmına kadar olan bitlerinin yolda bir bozulmaya uğrayıp uğramadığını kontrol ederler. İçlerinden biri bile mesajın bozulduğunu yakalarsa 16. biti baskın bite(Lojik 0) çekerler. Diğer tüm alıcı birimler bu durumu görür ve bozulan mesajı iptal ederler. Mesajı gönderen birim ise hat boş olduğunda aynı veriye sahip mesaj düzeltilip, hatta tekrar yazar.

EOF: Mesajın sonu olduğunu gösteren 7 adet çekinik bittten meydana gelir.

INT: Gelecek olan yeni mesajdan ayrılmak için kullanılan çekinik 3 bittten meydana gelir.

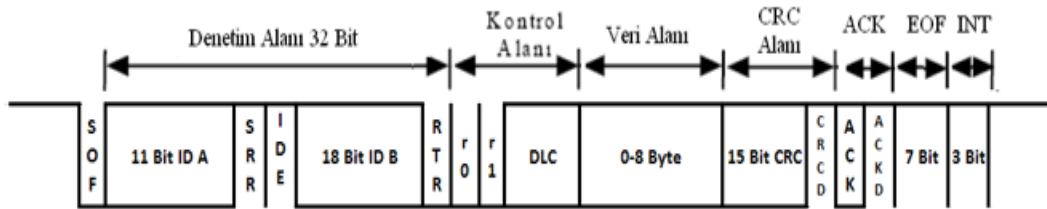
ACK: 2 bittten meydana gelir. Okundu onay bitidir. Gönderen birim ilk biti çekinik olarak işaretler. Her hangi bir birim bu mesajı okursa bu biti baskın hale dönüştürür. Böylece gönderen tarafından mesajın okunduğu anlaşılır. Mesaj hiçbir birim tarafından okunmazsa aynı mesaj tekrar gönderilir. 2. ACK biti ise sınırlandırıcı bittir.



Şekil 3.6 : Kontrol alanı iç yapısı[6].

Çizelge 3.1 : Veri baytları kodlaması(d:baskın, r:çekinik) [3].

Veri Baytları Sayısı	Veri Uzunluğu Kodu (DLC)			
	DLC3	DLC2	DLC1	DLC0
0	d	d	d	d
1	d	d	d	r
2	d	d	r	d
3	d	d	r	r
4	d	r	d	d
5	d	r	d	r
6	d	r	r	d
7	d	r	r	r
8	r	d	d	d



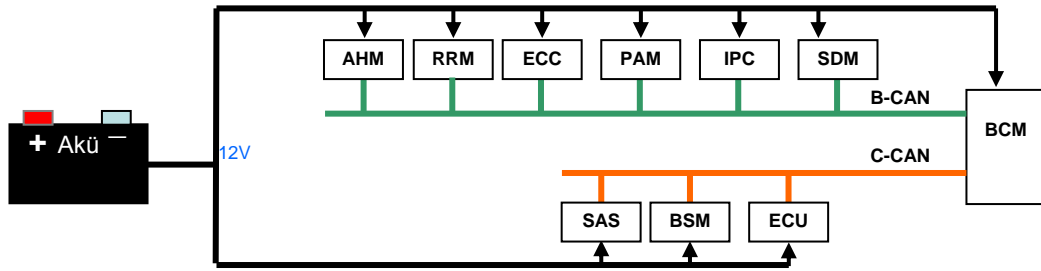
Şekil 3.7 : Örnek veri mesajı(Genişletilmiş format)[6].

3.2 Araç Üzerindeki Can Yapısı

Araç üzerinde iki ayrı CAN hattı bulunur. Bunlardan ilki düşük hızlarda iletilmesi gereken bilgileri taşıyan B-CAN hattıdır. B-CAN hattında haberleşme 50 Kb/s hızında yapılır. Aracın dil seçenekleri, kapı kilitlerinin durumları, yakıt durumu, araç içi sıcaklığı gibi bilgiler B-CAN hattında taşınan bilgilerden bazılarıdır.

Araç hızı, direksiyonun konumu, fren pedalının durumu, gaz kelebeğinin açıklık oranı gibi bilgiler C-CAN’de taşınırlar. Bu haberleşme hattının hızı 500 Kb/s’dir. C-CAN hattında genellikle ölçüm ya da iletim periyodu daha kısa olan bilgiler taşınır.

Aşağıdaki şemada da görüldüğü gibi birçok modül ve sistem B-CAN ve C-CAN hatları sayesinde birbirlerine bağlanır. BCM modülü gibi bazı modüller ise iki hatta da aynı anda bağlanabilirler.



Şekil 3.8 : Araç üzerindeki modüller ve CAN hatları.

C-CAN hattında yayınlanan mesajlar standart formata göre oluşturulurlar. Diğer bir deyişle her mesaj 11 bitlik ID içerir ve içinde 64 bite kadar veri tutabilir.

Araç üzerindeki C-CAN hattına erişim sağlamaya çalışılmasının nedeni, aracın kontrol edilebilmesi için gerekli olan bilgilerin bu hat üzerinde var olmasıdır. Bu bilgilerin araçtan alınabilmesi durumunda aynı bilgileri elde etmek için harici sensörler kullanılmasına gerek kalmayacaktır.

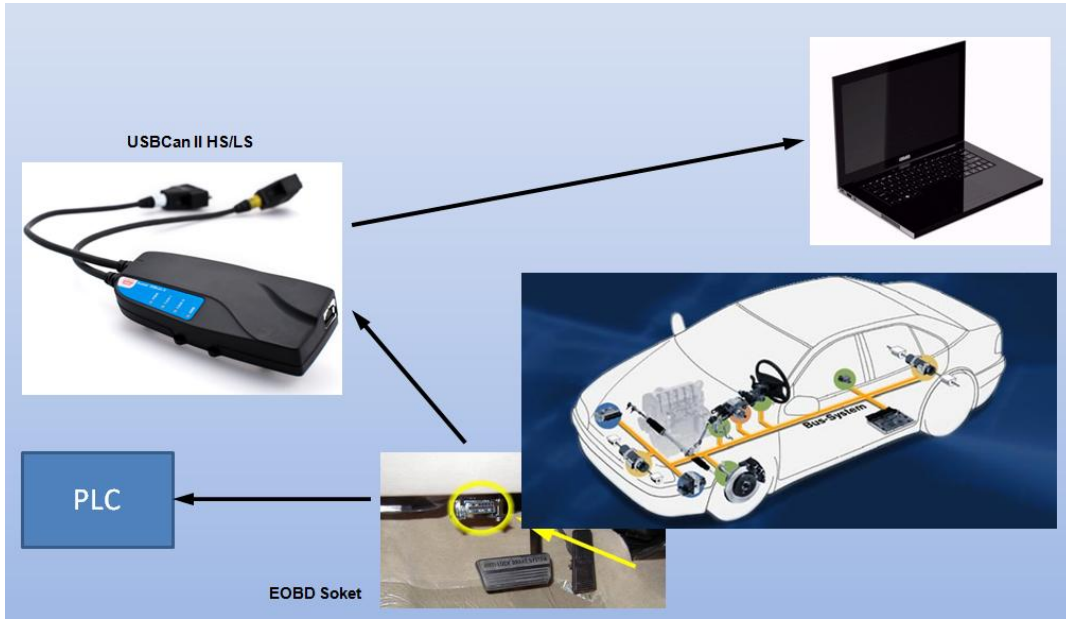
3.3 Can Hattına Bağlanabilme

Bu çalışma kapsamında ihtiyaç duyulan bilgiler C-CAN hattında bulunduğu için bu hatta bağlanılmıştır. Bağlantı araç üzerinde bulunun EOBD soket üzerinden gerçekleştirilmiştir. EOBD soket araçlarda genellikle ön kısımda ve direksiyon ile pedal takımı arasında bir noktaya yerleştirilir.

Şekilde de görüldüğü gibi aracın haberleşme hattına girebilmek için EOBD sokete iki farklı şekilde bağlanılabilir. Hat üzerindeki yayınlanan bilgileri bir bilgisayarda görmek ve bilgiler üzerinde işlem yapmak istiyorsak CAN kart kullanılmalıdır. Bu

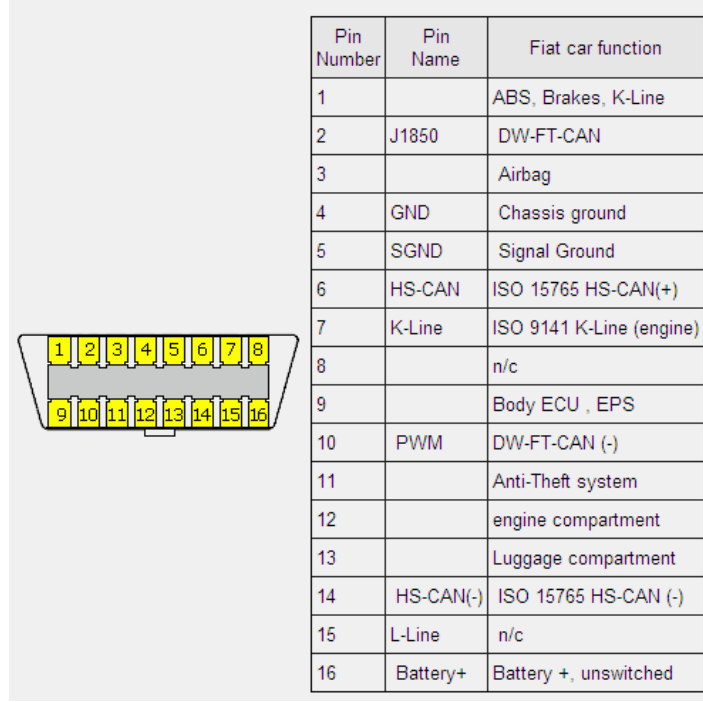
kart bilgisayara USB ara yüzüyle, sokete ise HS-CAN(+), HS-CAN(-) ve GND uçlarının bağlanmasıyla haberleşmeyi sağlar. CAN karta ve EOBD sokete ait pin diyagramı Şekil 3.9 ve Şekil 3.10’da verilmiştir.

CAN hattından istenilen ID’li veriyi çekebilmek için elektriksel bağlantının yanı sıra CAN karta ait yazılım programını çalıştırmak gereklidir. Bu program kullanıcıların isteklerine tam olarak cevap veremeyebilir. Çözüm ise daha spesifik çalışan gerekli kodu oluşturmak ve çalıştırmaktır

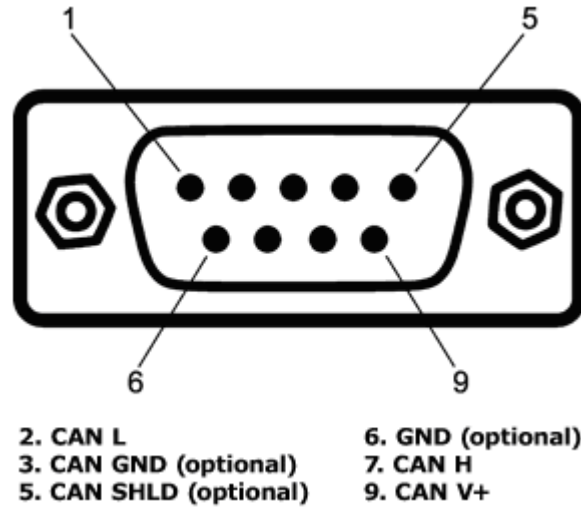


Şekil 3.9 : CAN hattına bağlanabilme.

Araçtan veri okumanın bir başka yolu ise kontrolörü EOBD sokete arada bir kart olmadan direkt bağlamaktır. Bunun için öncelikli şart kontrolörün CAN ara yüzüne sahip olmasıdır. Veri okumak için kontrolöre gömülen yazılım bilgisayarda oluşturulan koda göre daha kompleks olacaktır. Çünkü okunan mesaj bloklarının hatalı olup-olmadığını tespit etme, bu hatayı hatta bildirme gibi CAN kartın yaptığı işler de bu yazılıma eklenmelidir.



Şekil 3.10 : EOBD soket pin diyagramı.



Şekil 3.11 : CAN kart pin diyagramı.

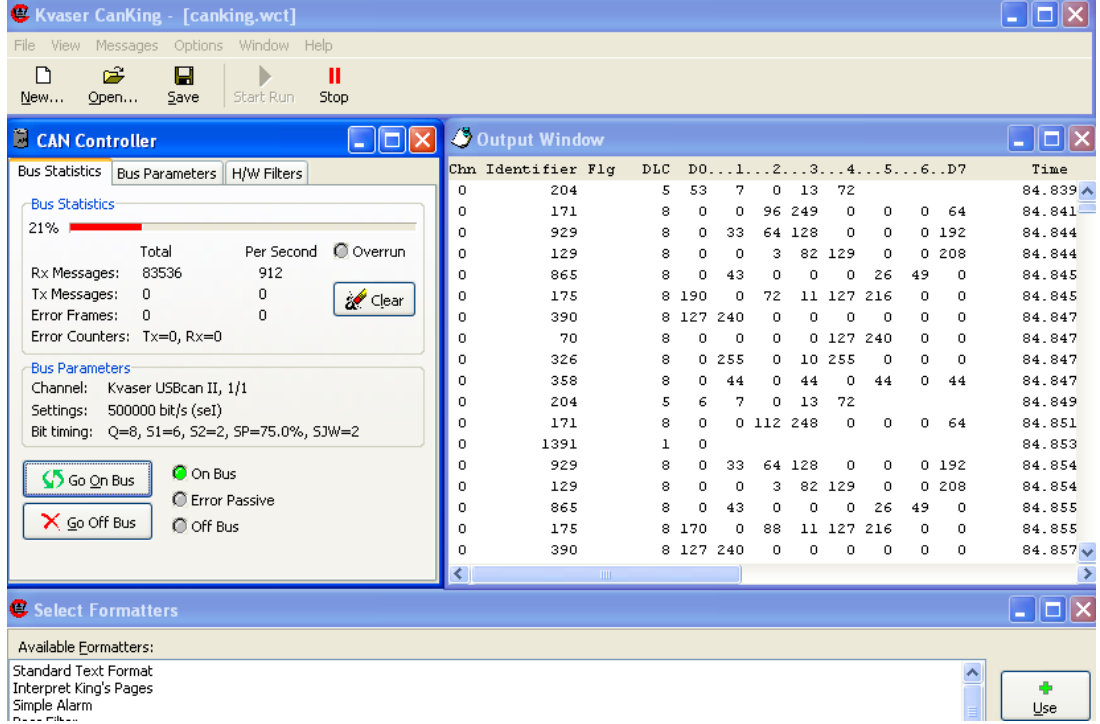
3.4 Can Hattı Sayesinde Erişilebilen Veriler

Her ne kadar direksiyonu çevirmek ya da fren pedalının ileri-geri hareketini sağlamak için sistemin kendisine ait konum sensörleri olsa da CAN'den veri alınmasının bazı önemli nedenleri vardır.

Bunlardan ilki, başka verilere ihtiyaç duyulması, ya da duyulabilme ihtimali olmasıdır. Örneğin, direksiyonu istenilen hızda istenilen konuma getirmek yeterli midir? diye düşünüldüğünde cevabın hayır olduğu anlaşılır. Bunun sebebi direksiyonun dönüş açısı limitlerinin aracın hızına da bağlı olması gerekliliğidir. Eğer bu yapılmazsa aracın takla atma ihtimali çok yüksektir. Direksiyon tek başına bir sistem olsaydı, diğer bilgilere gerek duyulmazdı ama araca bağlı bir sistem olduğu için diğer verilere de ihtiyaç duyulur.

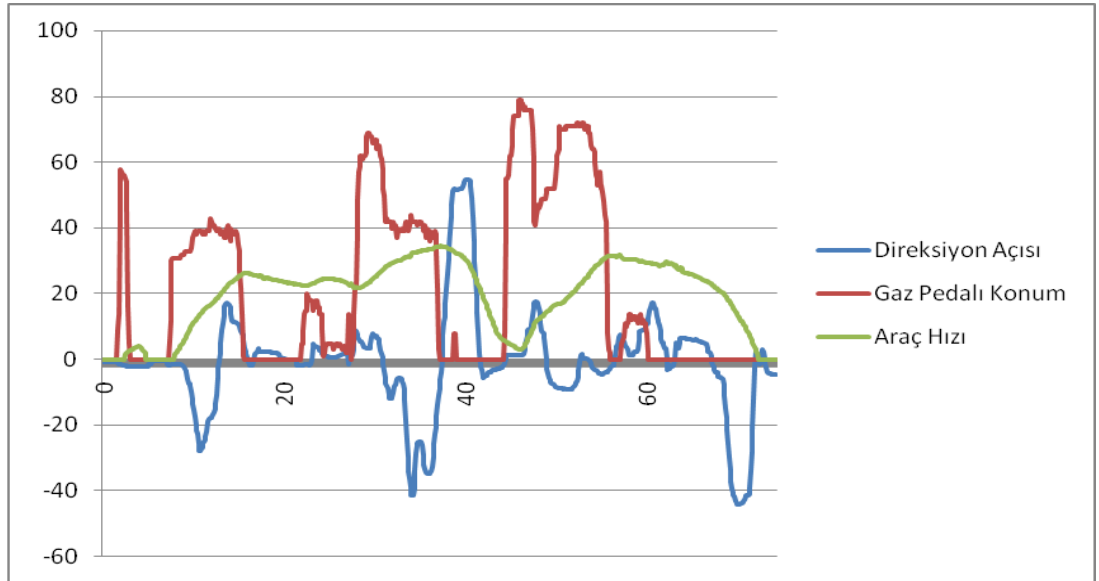
Araç üzerindeki verilerin okunmasını gerektiren bir başka neden de tasarlanacak kontrol bloğunda birden fazla noktadan geri besleme alınmasının, bir noktadan geri besleme alınmasından daha güvenli olmasıdır. Kontrol sistemindeki sensörlerde meydana gelebilecek bir bozulma ya da arıza durumunda diğer sensörlerden alınan verilere göre hemen kalibrasyon yapılabilir yada düzeltilemeyecek bir arıza meydana geldiğinde otomatik sistem uyarı verip, kendini en güvenli şekilde kapatabilir.

Aşağıdaki şekilde CAN-C hattına bağlanabilmek için kullanılan yazılımın hat üzerinden veri okunurken alınmış ekran çıktısı görülmektedir. Ara yüzde de görüldüğü gibi mesajlar ID numarası, veri uzunluğu kodu ve verilerden oluşmaktadır. Mesajın içinde bulunan diğer alanlar(CRC vb.) kullanıcıya yansıtılmamıştır. Bununla birlikte yayınlanan mesajların içindeki verilerin başka sistemler tarafından gerçek zamanlı olarak kullanılması bu programla mümkün değildir.



Şekil 3.12 : Kvaser CanKing programı ekran görüntüsü.

Araç üzerinde gerçek bir sürüş testi yapılırken CAN King programı yardımıyla CAN hattından veriler okunmuştur. Bu okunan veriler sayesinde Şekil 3.13 'de görüldüğü gibi zamana bağlı direksiyon açısı-araç hızı-pedal konuna ait grafik oluşturulmuştur.



Şekil 3.13 : Direksiyon açısı-araç hızı-pedal konumu.

Grafik üzerinde yatay eksen saniye cinsinden zamanı belirtmektedir. Direksiyon açısı ise tekerleklerin dönüş miktarının yaklaşık olarak on katı olduğundan dönüş açısını bu değerin onda biri kabul edebiliriz. Veri çizgilerine bakarak aracın davranışını yorumlayacak olursak gaz pedalına basıldığında aracın hızlandığı net bir şekilde görülmektedir. Dönüş açılarının çok büyük olmaması yolun fazla virajlı olmadığını da ortaya koyar.

Direksiyon sisteminde anlık olarak çalışmadığı için bu programın kullanımı olanaksız olsa da, fren ve gaz pedalı için aracın hız modelinin transfer fonksiyonu çıkarılırken kullanılabilir. Girişler ve bunlara bağlı çıkışlar kaydedilip, MATLAB'de sistem tanımlama yöntemi ile yaklaşık bir transfer fonksiyonu çıkartılıp, bu fonksiyona göre bir hız kontrolörü tasarlanabilir.

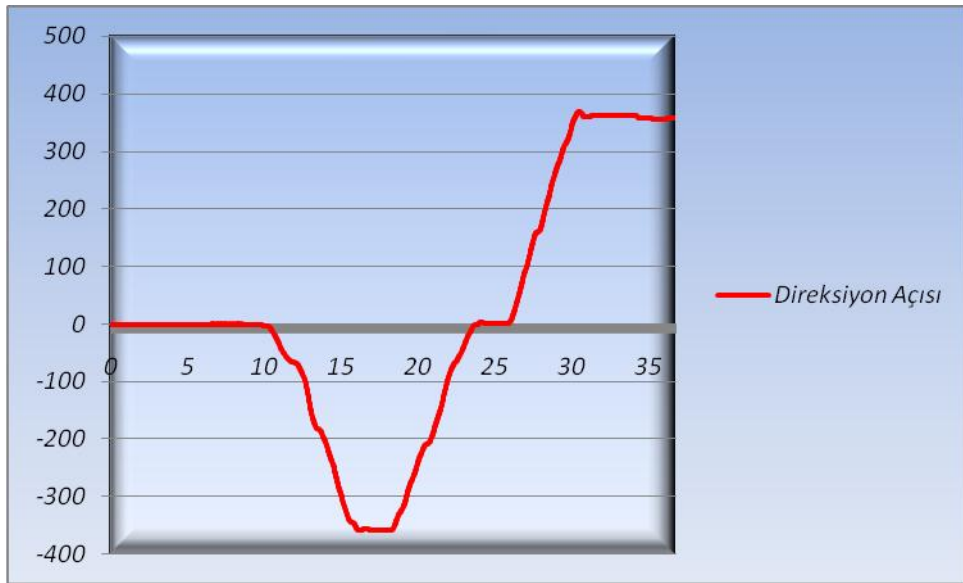
Otonom araç projesinde direksiyon sistemi için gerekli verilere anlık olarak ihtiyaç duyulduğundan CAN kartına ait kütüphaneler de kullanılarak bir program yazılmıştır. Bu program sayesinde istenilen mesaj ID numarasına göre diğerlerinden ayrılabilmekte ve mesaj içinden gerekli veri çeşitli algoritmalarla çekilebilmektedir. Elde edilen değer gerçek zamanlı sistemlerin çalışması için bilgisayarın bir başka portundan gerekli birimlere gönderilmektedir. İstenilen verinin sürekli kaydedilebilmesinin yanı sıra, ihtiyaç duyulan veriye en fazla mesajın yayınlanma periyodu kadar bir gecikmeyle ulaşılabilmesi bu yazılımın en büyük avantajıdır.

Şekil 3.14'de yazılan programın komut ekranında çalıştığı andaki ekran çıktısı verilmektedir. En sol sütunda seçilen mesajın ID numarası(204), hemen yanındaki sütunda mesajdaki veri uzunluğu(5 Byte) ve daha sonra direksiyon sistemine ait veriler yer almaktadır. Direksiyon sürücü müdahalesiyle döndürüldüğünde okunan direksiyon açısı değerine göre aşağıdaki grafik ortaya çıkmıştır.


```
C:\WINDOWS\system32\cmd.exe - candump.exe
204 05 6 7 0 c a7 aci= 3239 : 0 = 0 9.973
204 05 17 7 0 c a7 aci= 3239 : 0 = 0 9.983
204 05 8 7 1 c a8 aci= 3240 : 0 = 0 9.993
204 05 19 7 1 c a8 aci= 3240 : 0 = 0 10.003
204 05 3a 7 0 c a8 aci= 3240 : 0 = 0 10.013
204 05 2b 7 0 c a8 aci= 3240 : 0 = 0 10.023
204 05 5c 7 1 c a9 aci= 3241 : 0 = 0 10.033
204 05 5d 7 0 c a9 aci= 3241 : 0 = 0 10.043
204 05 7e 7 1 c a9 aci= 3241 : 0 = 0 10.053
204 05 6f 7 1 c a9 aci= 3241 : 0 = 0 10.063
204 05 80 7 0 c a9 aci= 3241 : 0 = 0 10.073
204 05 81 7 2 c aa aci= 3242 : 0 = 0 10.083
204 05 92 7 0 c aa aci= 3242 : 0 = 0 10.093
204 05 83 7 1 c ab aci= 3243 : 0 = 0 10.103
204 05 b4 7 2 c ac aci= 3244 : 0 = 0 10.113
204 05 85 7 0 c ac aci= 3244 : 0 = 0 10.123
204 05 96 7 2 c ac aci= 3244 : 0 = 0 10.133
204 05 a7 7 1 c ad aci= 3245 : 0 = 0 10.143
204 05 58 7 2 c ae aci= 3246 : 0 = 0 10.153
204 05 79 7 1 c ae aci= 3246 : 0 = 0 10.163
204 05 6a 7 2 c af aci= 3247 : 0 = 0 10.173
204 05 ab 7 1 c b0 aci= 3248 : 0 = 0 10.183
204 05 ec 7 2 c b0 aci= 3248 : 0 = 0 10.193
204 05 ed 7 2 c b1 aci= 3249 : 0 = 0 10.203
204 05 de 7 1 c b2 aci= 3250 : 0 = 0 10.213
```

Şekil 3.14 : Direksiyon açısı bilgisi.

Şekil 3.15’deki grafikte direksiyon öncelikle 10 saniye kadar 0° açıda bekletilmiş, sonra saat yönünde yaklaşık 350° çevrilmiştir. Kısa bir süre beklendikten sonra direksiyonun “sıfır” konumuna geri getirilebilmesi için saat yönünün tersinde 350° çevrilmiştir. Ardından aynı işlem diğer yön için de yapılmıştır. Bu grafik aracın CAN hattına girilerek elde edilen veriler yardımıyla çıkarılmıştır. CAN hattına erişebilmek için yazılan programın kodları Ek-B’de verilmiştir.



Şekil 3.15 : Direksiyon açısı bilgisi grafiği.

4. DİREKSİYON KONTROLÜ

Sürücünün direksiyon simidini döndürerek ön tekerleklere istenen açıyı vermesini sağlayan birbiriyle bağlantılı elemanlar topluluğuna direksiyon sistemi denir. Direksiyon sistemi araca yön verilmesini sağlamanın yanı sıra aracın dinamiğine ve dengesine de etki etmektedir. Bu nedenle direksiyon sistemi araç kontrolü ve sürüş güvenliği açısından son derece önemlidir.[8] Yeni teknolojilerle birlikte araçlardaki yönlendirme sistemleri de kontrol ve sürüş güvenliği açısından büyük bir gelişme göstermiştir.

Taşıtın sürüş güvenliği açısından hareket esnasında, özellikle dönüşlerde, ön tekerleklerin yol tutuşunun iyi ve aracın yol düzlemi üzerindeki hareketinin kararlı olması istenir. Bu yüzden sürücünün araca yaptırmak istediği dönme hareketi belli bir geometriye bağlı olmalıdır.[9] Bu geometrik yapının sonucunda araç ön tekerlekleri dönüşler esnasında farklı açılarda konumlanırlar. Bu ideal dönme hareketi sayesinde tekerleklerin kayması ve aşınması da engellenmiş olur.[10]

Konfor bakımından özellikle çok düşük hızlarda sürücünün araca yön vermesi tekerleklere gelen yanal kuvvetlerin artmasından dolayı oldukça zorlaşmaktadır. Bu sorunu giderebilmek için mekanik çözümlerin geliştirilmesinin yanı sıra sürücüye destek olabilecek farklı güç takviyeli sistemler de üretilmiştir.

Otonom araç projesi kapsamında içinde aracın yönlendirme sisteminin de bulunduğu birçok temel fonksiyon kontrol edilmiştir. Bu sistem otonom ve manuel olmak üzere iki farklı şekilde çalışmaktadır. Manuel çalışmada direksiyon sistemi sürücünün direksiyon simidine uyguladığı tork ile uyarılmakta, otonom çalışmada ise direksiyon sistemi bir servomotor aracılığıyla tahrik edilmektedir. Bu bölümde öncelikle yönlendirme sistemlerinin genel yapısı ve çeşitleri incelenmiş, aracın direksiyon sisteminin modeli kurulmuş ve gerekli simülasyonlar elde edilmiştir. Ardından bu sonuçlara göre uygun kontrolör seçilmiş ve gerekli tasarım yapılmıştır.

4.1 Direksiyon Sistemini Oluşturan Temel Elemanlar

4.1.1 Direksiyon simidi

Sürücünün uyguladığı döndürme hareketini direksiyon miline aktaran elemandır. Aynı zamanda yoldan gelen sabit sürüş, viraj tutuşu ve yol özellikleri gibi bilgileri sürücüye aktararak hissetmesini sağlayan elemandır.

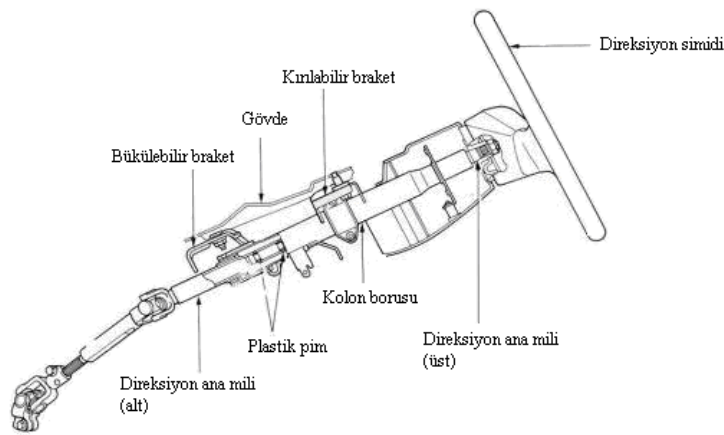
Hafif metal alaşımları kullanılarak yapılmasının yanı sıra, daha konforlu ve güvenli bir şekilde direksiyon simidini kavraması için üzeri çeşitli örtü malzemeleri ile kaplanır.[11] Direksiyon simidinin çapının daha büyük olması uygulanacak olan kuvvetin daha küçük olmasını sağlar. Fakat ergonomik açıdan bu bir dezavantajdır.

4.1.2 Direksiyon mili

Direksiyon mili direksiyon simidinden direksiyon dişli kutusuna güç aktarımı yapan elemandır. Sabit ve ayarlanabilir olmak üzere iki çeşittir. Direksiyon mili bazı araçlarda koruma amaçlı olarak direksiyon kolon borusu ile kaplıdır. Direksiyon mili ve etrafını saran direksiyon kolon borusu direksiyon sütunu veya direksiyon kolonu olarak adlandırılır.

Kolon gövdeye kırılabilir bir braket ile bağlanır. Bu sayede her hangi bir kaza anında braket kırılarak sürücünün zarar görmesine neden olabilecek darbeler emilir.

Direksiyon mili alt kısımda bulunan dişli kutusuna mafsallı istavroz aracılığıyla bağlanmıştır. Bu sayede yoldan gelen titreşimler direksiyon simidinde daha az hissedilir. Bu da sürüş konforunu artırır. [11]



Şekil 4.1 : Direksiyon mili[11].

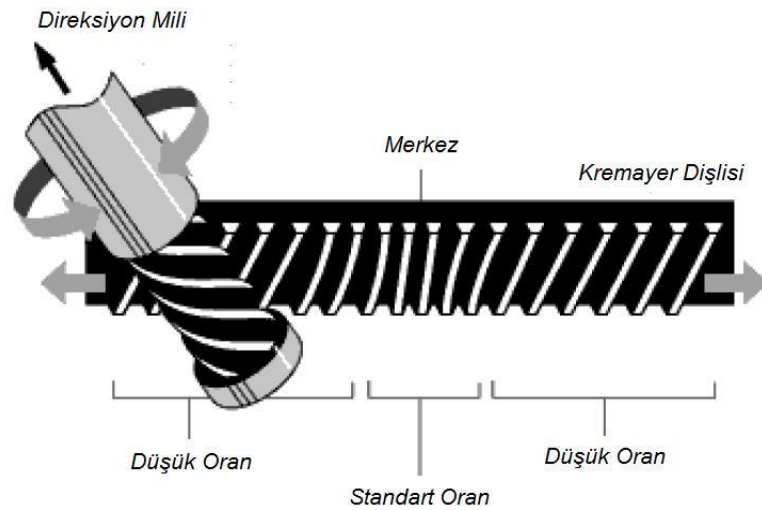
4.1.3 Direksiyon dişli kutusu

Dişli kutusu milden gelen dönel hareketi doğrusal harekete çevirir ve bir sonraki elemana iletir. Başlıca iki tür direksiyon kutusu vardır: pinyon dişli kutusu ve dönel bilyeli dişli kutusu.

Direksiyon dişli kutularında tork iletimi dişli oranına ve verime bağlıdır. [12] Sürücünün döndürme kuvveti dişliler tarafından çoğaltılır ve bir sonraki elemana aktarılır. Böylece sürücü daha az kuvvetle direksiyon simidini çevirebilir.

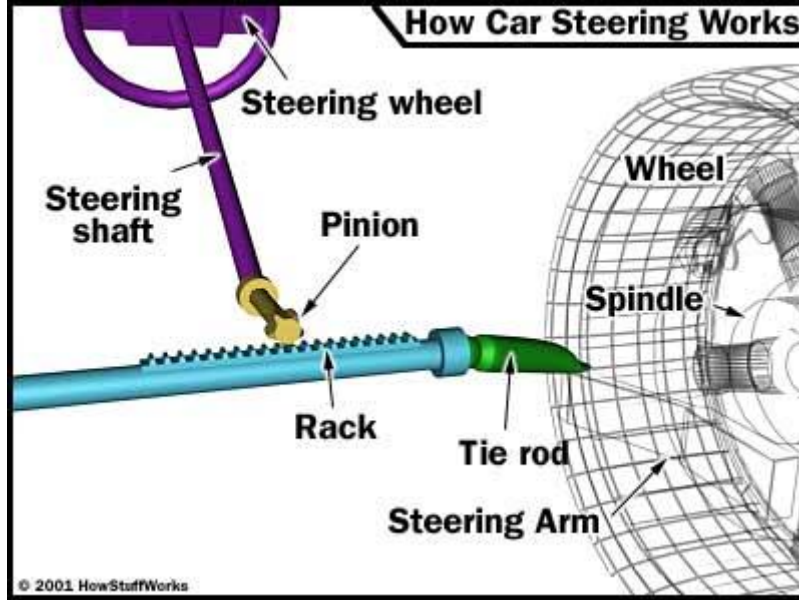
4.1.4 Pinyon dişli kutusu

Bir mafsal ile direksiyon miline bağlı olan pinyon dişlisi sistemde kremayer dişlisi(mili) ile birlikte kullanılır. Pinyondaki dönel hareket kremayerle olan kavramalı bağlantı sayesinde kremayeri çizgisel bir harekete zorlar. Hareketin iletimi sırasında döndürme kuvvetini artırarak sürücüye kolaylık sağlar.[11] Bu dönüşüm esnasındaki dişli giriş-çıkış oranı pinyon dişli dönme miktarının kremayerin ilerleme miktarına oranıyla bulunur. Bazı pinyon-kemayer yapılarında bu oran kremayer dişlisi üzerindeki diş aralıklarının farklı olmasından dolayı değişken olabilir. Böyle bir yapı sayesinde orta konumda kuvvetten daha fazla kazanç sağlanırken, sağa ve sola çevirdikçe yoldan kazanç artmaya başlayacaktır. Bu sayede orta konumda daha hassas bir sürüş elde edilir. [13]

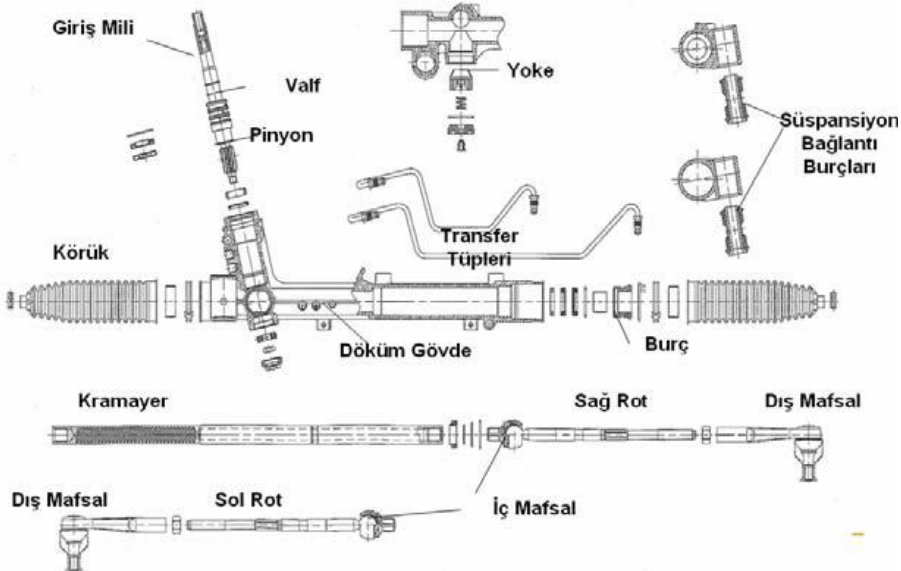


Şekil 4.2 : Pinyon Dişli Kutusu [13].

Kremayer üzerindeki hareket rot bağlantıları aracılığı ile dingil başlarına, oradan da tekerleklere iletilir ve dönme hareketi sağlanmış olur.

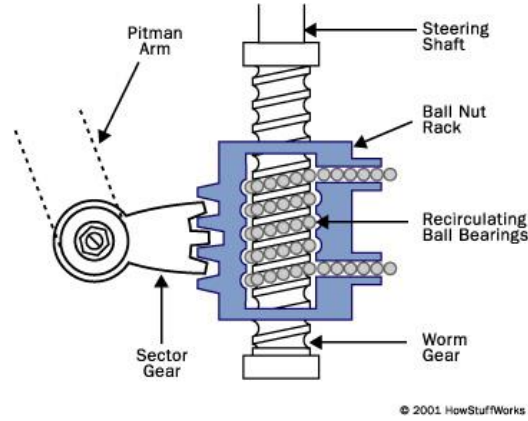


Şekil 4.3 : Pinyon-Kremayer sistem [14].



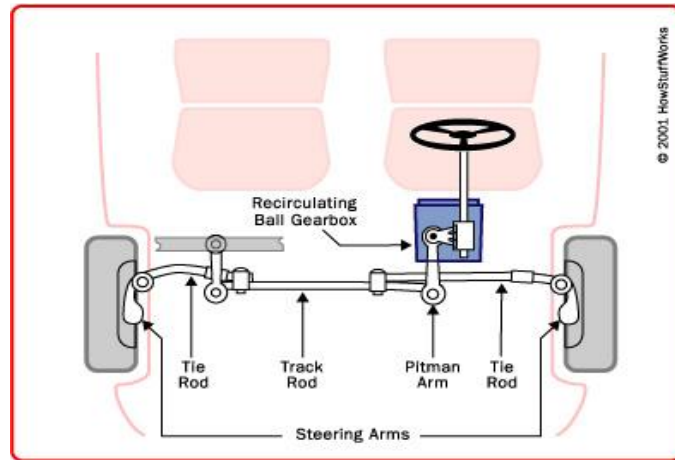
4.1.5 Dönel bilyeli dişli kutusu

Bu sistemde direksiyon mili hareketini sonsuz vidaya aktarır. Sonsuz vidanın üzerinde çelik bilyeler için yataklar bulunur. Sonsuz vida hareket ettiğinde yatak üzerindeki düşük sürtülmeli bilyeler dış taraftaki somunu destekleyerek hareket ederler. Bu sırada somundan çıkan bilyeler bir kanal sayesinde tekrar somuna taşınırlar.



Şekil 4.5 : Dönel bilyeli dişli kutusu [14].

Bilyelerin desteklediği somun sonsuz vidanın dönüş yönüne göre aşağı ve yukarı hareket eder. [16] Bu hareket somun tarafından önce sektör dişlisine, oradan da pitman koluna aktarılır. Pitman koluna gelen kuvvet sırasıyla rotlara ve daha sonra tekerlere geçer. Böylelikle dönme hareketi gerçekleşmiş olur.

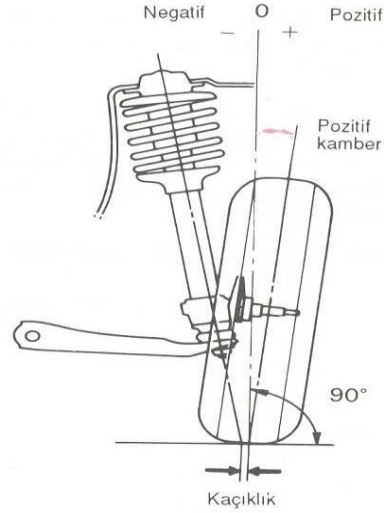


Şekil 4.6 : Dönel bilyeli mekanik direksiyon sistemi [14].

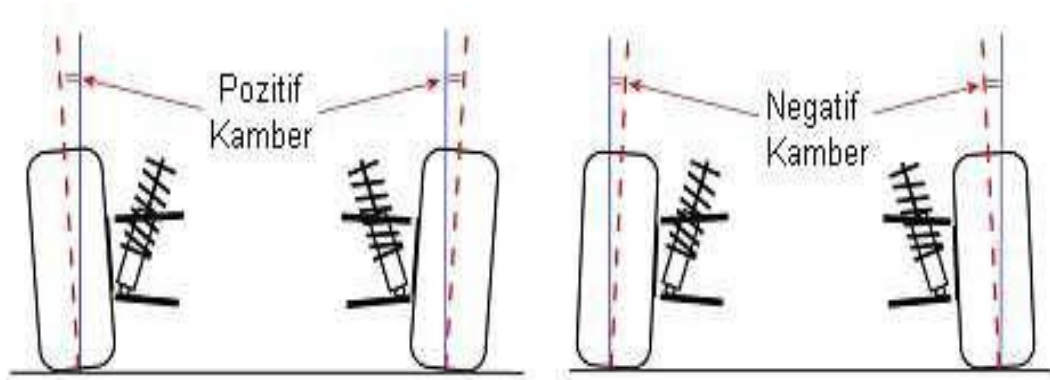
4.2 Ön Düzen Geometrisi

4.2.1 Kamber açısı

Bilyelerin desteklediği somun sonsuz vidanın dönüş yönüne göre aşağı ve yukarı hareket eder. [16] Bu hareket somun tarafından önce sektör dişlisine, oradan da pitman koluna aktarılır. Pitman koluna gelen kuvvet sırasıyla rotlara ve daha sonra tekerlere geçer. Böylelikle dönme hareketi gerçekleşmiş olur.



Şekil 4.7 : Kamber açısı [11].

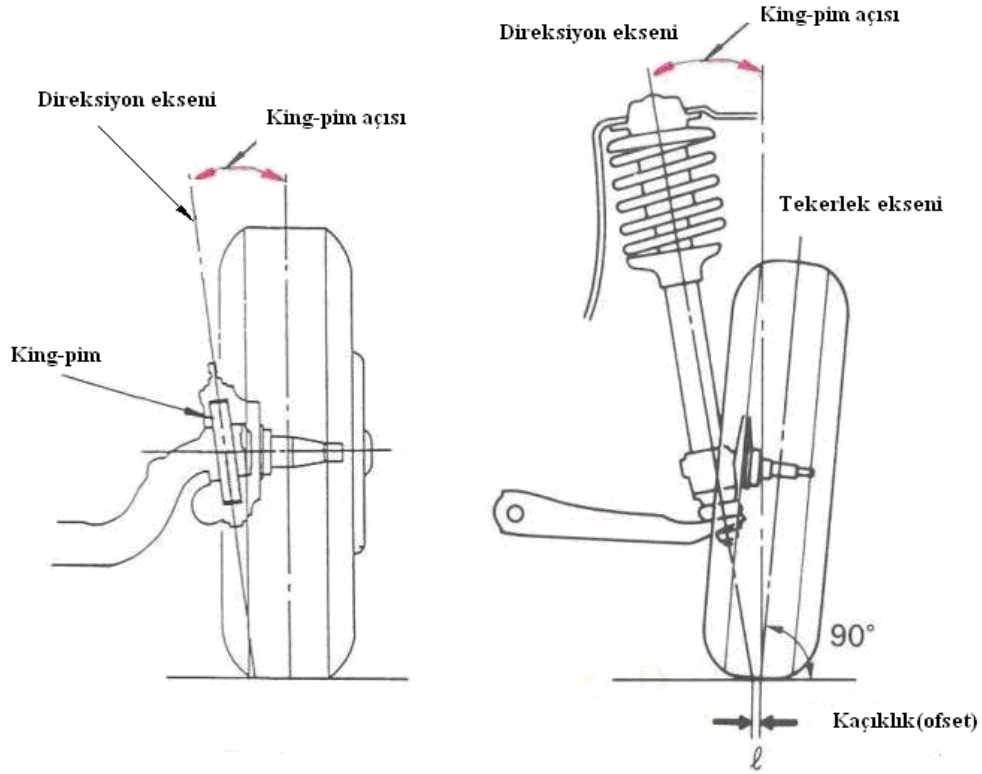


Şekil 4.8 : Pozitif ve negatif kamber[11].

Kamber açısından dolayı yere eğimli olan tekerlekler üzerlerine araç yükü geldiğinde yere dik konuma gelirler. Bu durumda lastik yol yüzeyine daha iyi bir temas sağlar. Bu da lastiklerdeki aşınmanın dengeli olmasını sağlar.

4.2.2 King-Pim açısı

Tekerleğe önden bakıldığında dingil piminin(King-pim) düşey eksen ile yaptığı açığa denir. Yüksek kamber açısına olan ihtiyacını azaltmasının yanı sıra tekerlekleri 0° 'lik konuma getirmeye yönelik bir moment de üretir. Bu moment sayesinde araç düz giderken çevreden gelebilecek bozucu etkenlere karşı(tekere gelen bir taş) daha kararlı hale gelir. Yine bu moment sayesinde dönüş sonrasında araç tekerlekleri tekrar düz konuma gelmesi konusunda sürücüye yardımcı bir kuvvet de oluşur.



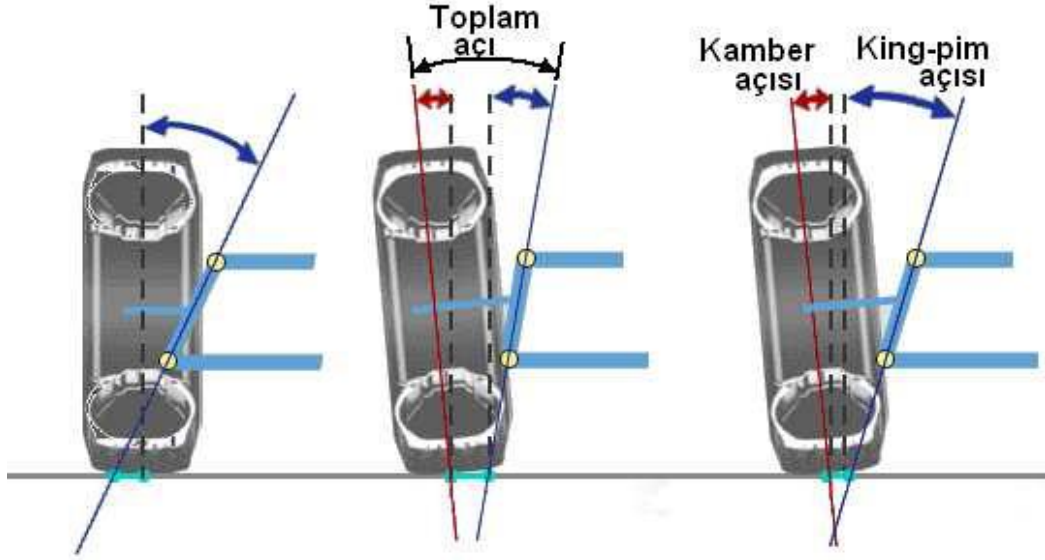
Şekil 4.9 : King-pim açısı[11].

4.2.3 Toplam açı

Kamber açısı ile king-pim açısının toplamına toplam açı denir. Bu açının büyüklüğünden ziyade kamber ve king-pim açılarının kesişim noktası daha önemlidir. Kesişim noktasının yerin 5-6 cm altında ya da üstünde olması istenir.

Bu kesişim noktasının yerin üstünde olması durumunda tekerlekler içe doğru kapanmaya, yerin altında ise dışa doğru açılmaya zorlanır. Kuvvetler eşit ve karşılıklı olacağı için yönlendirme sistemi dengede olacaktır.

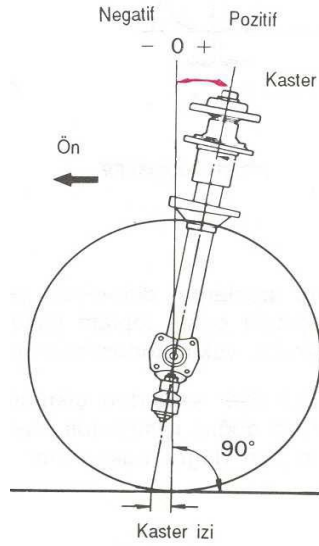
Kesişim noktasının yer yüzeyine sıfır olması durumunda dışarıdan her hangi bir kuvvet oluşmaz, sistemdeki dişli, pim vb. elemanlardaki küçük boşluklardan ötürü tekerlekler sağa-sola yalpa yapar. [11]



Şekil 4.10 : Toplam açı[11].

4.2.4 Kaster açısı

Aracın tekerleklere yandan bakıldığında king-pim eksen çizgisi ile düşey eksen arasında oluşan açığa denir. Araç ileri doğru giderken king-pim eksenini düşey eksenin arkasında kalıyorsa pozitif, önünde ise negatif kaster açısı olarak arlandırılır. Kaster açısı aracın stabilitesini sağlar. Yani dönüşlerden sonra taşıtın tekrar doğrusal konuma gelmesine yardımcı olur. Bu sayede king-pim açısının etkisini destekler. Yanlış bir kaster açısı direksiyonun dönmesi için gereken kuvveti artırır. Kaster açısı ön dingil ağırlı düşük araçlarda yüksek, ön dingil ağırlığı yüksek olan araçlarda düşük seçilir.[17]

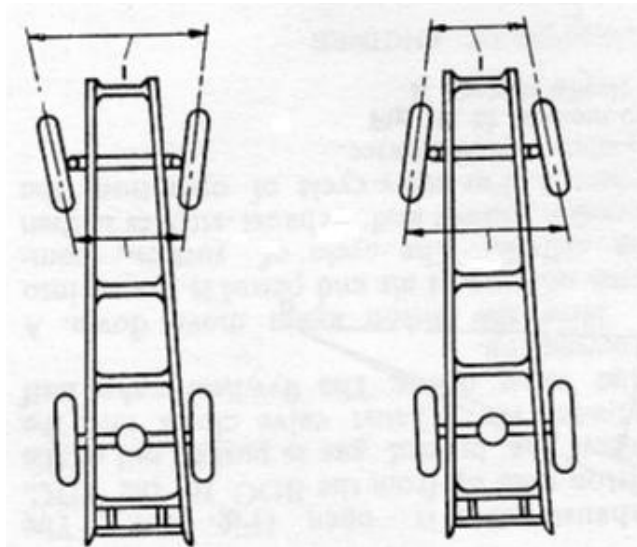


Şekil 4.11 : Kaster açısı [11].

4.2.5 Toe açısı

Araca yukarıdan bakıldığında ön tekerler araca ve birbirlerine paralel değildir. Direksiyon sıfır konumunda iken araçla yaptıkları açıya toe açısı denir. Tekerlerin ön tarafının arka tarafına göre daha kapalı olma durumuna toe-in, açık olma durumunu da toe-out denir.

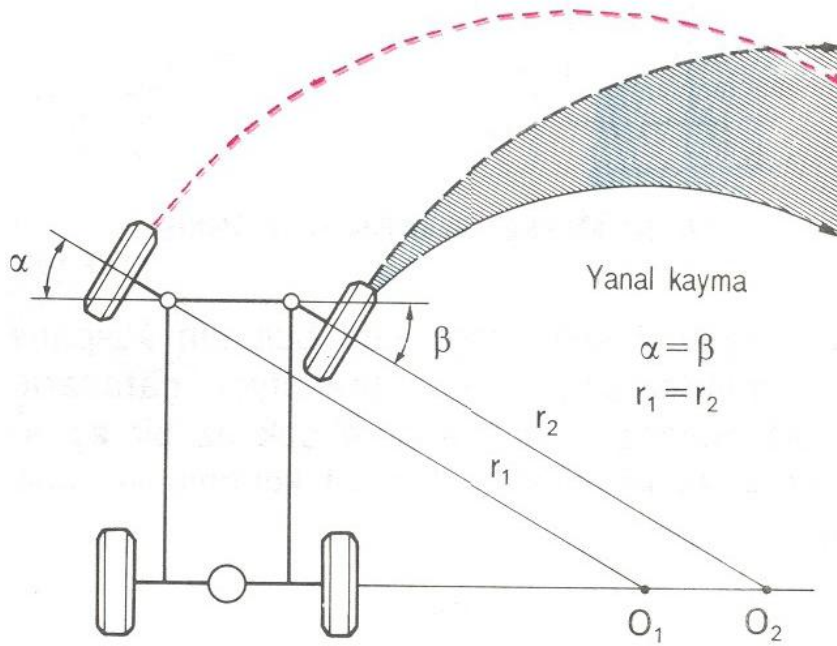
Arkadan itişli araçlarda ön tekerlekler dışa doğru açılmaya, önden çekişli araçlarda içe doğru kapanmaya zorlanır. Burada araç dinamiğinden dolayı oluşan kuvvetin tekerlekle yaptığı etkiyi sıfırlamak için üretim esnasında tekerlekler küçük bir toe-out veya toe-in açılarıyla beraber üretilirler.



Şekil 4.12 : Toe açısı [17].

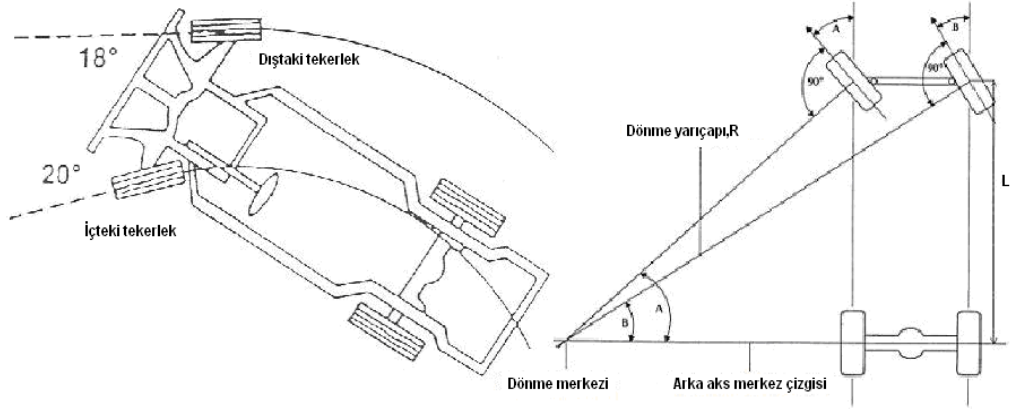
4.2.6 Dönüş açısı Ackerman geometrisi

Aracın dönüşleri esnasında dört tekerin merkezlerinden çıkan dikmelerin kesiştiği nokta aracın ani dönme merkezidir. Ön tekerlerin aynı açılarla paralel hareket etmesi bu merkezin oluşmasına engel olur. Aracın etrafında döndüğü nokta O_1 ve O_2 arasında, hatta bu iki nokta arasındaki doğrunun biraz dışında oluşacaktır ve stabil olmayıp sürekli değişecektir. Araç üzerindeki her teker ani dönme merkezini kendi eksenine oturtmaya çalışacağından bütün tekerlerde kaymalar ve aşınmalar meydana gelecektir. Özellikle ön tekerlerde meydana gelen kayma ve aşınma daha fazla olacaktır. Dıştaki ön teker dışa doğru, içteki ön teker içe doğru baskı görecektir.

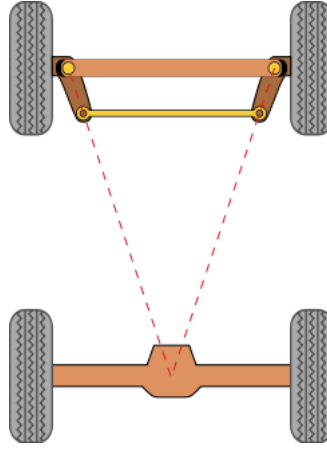


Şekil 4.13 : Yanal kayma [11].

Ackerman'a göre hatasız bir dönüş hareketi için iç taraftaki tekerlek dönüş merkezine daha yakın olduğundan dış taraftaki tekerleğe göre daha büyük bir açıyla dönmelidir. Bu bir geometrik yapı gerektirmektedir. Bu yapıyı tam anlamıyla oluşturmak için karmaşık sistemlere ihtiyaç vardır. Fakat yaklaşık olarak bu geometriyi sağlayan trapezoid yapılar güncel sistemlerde kullanılmaktadır. Deveboynu bağlantı elemanlarının rota dik olarak değil, belli bir açıyla bağlanmasıyla bu geometrik yapı büyük oranda sağlanmış olur.



Şekil 4.14 : Dönüş açısı [17].

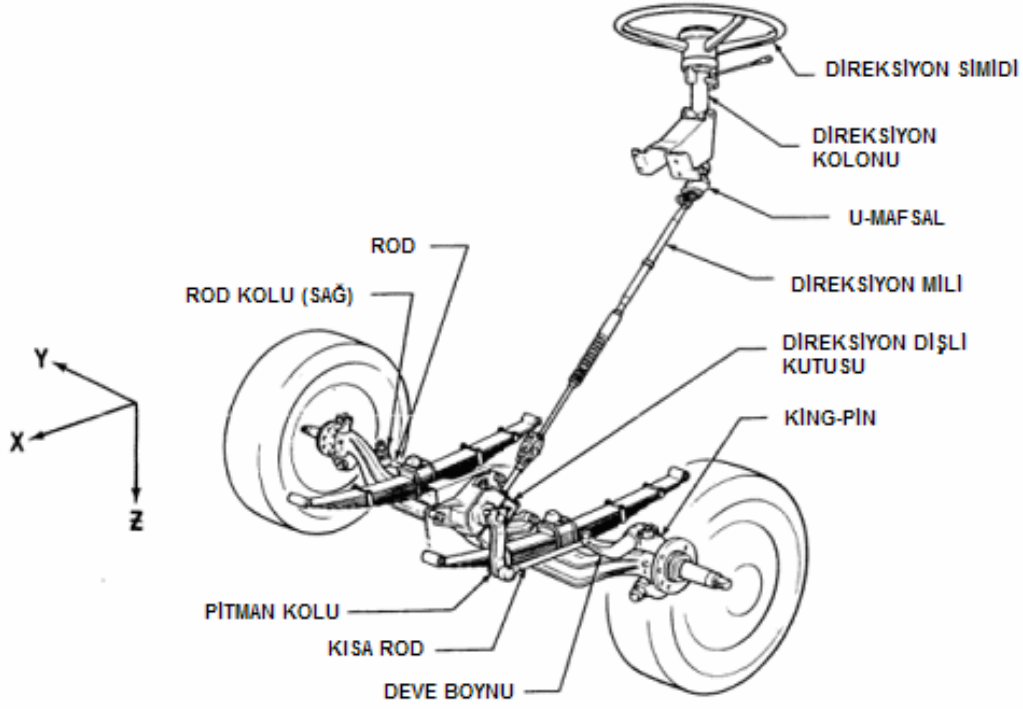


Şekil 4.15 : Ackerman geometrisi [18].

4.3 Yönlendirme Sistemi Çeşitleri

4.3.1 Mekanik yönlendirme sistemleri

Bu sistemlerde sürücü tarafından oluşturulan yönlendirme hareketi sadece mekanik aksamlar vasıtasıyla tekerleklere iletilir ve dönme hareketi yalnızca insan gücüne bağlıdır. Yönlendirme için herhangi bir yardımcı kuvvet olmadığından sürücünün uyguladığı güç sistem içindeki her kademede büyütülerek bir sonraki mekanizmaya ve son olarak da tekerleklere aktarılır. Bu aktarım esnasında bir çok farklı yapı kullanılmasına rağmen, pinyon-kremayer tipi mekanik direksiyon sistemi ve döneleli mekanik direksiyon sistemi öne çıkmaktadır. Bütün yönlendirme sistemleri mekanik sisteme entegre edilerek ortaya çıkartılmıştır.

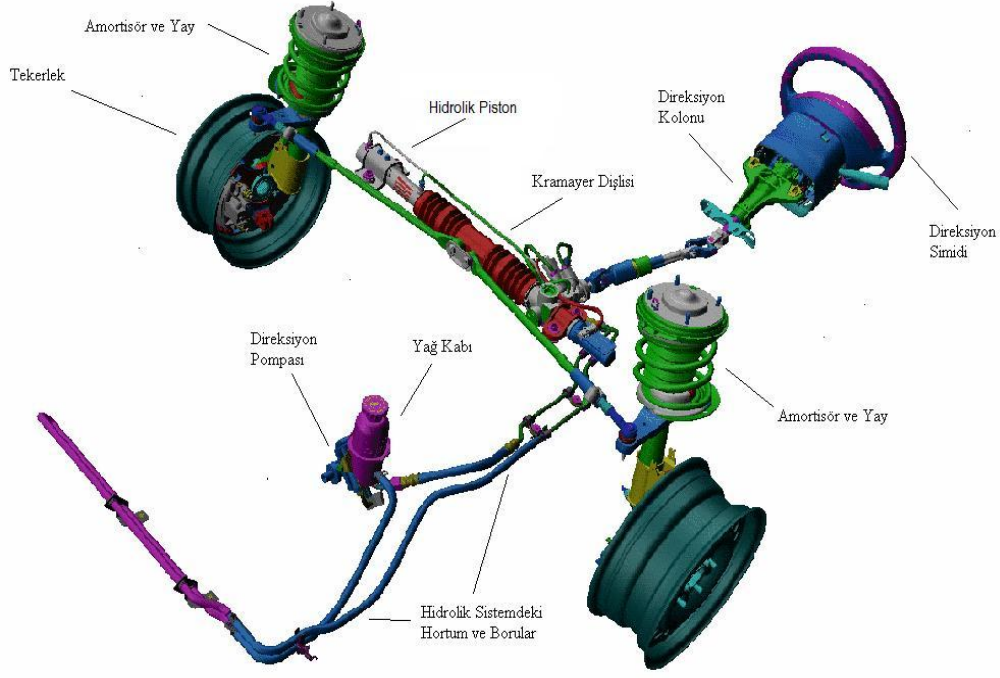


Şekil 4.16 : Mekanik yönlendirme sistemi.

Sürücünün mekanik direksiyonu çevirebilmek için harcadığı kuvvet diğer gelişmiş sistemlerden daha fazladır. Bu da konfor ve sürüş kolaylığı açısından bir dezavantajdır. Bunun yanında direksiyon geri toplama özelliği yardımcı herhangi bir kuvvet olmadığı için çok zayıftır.

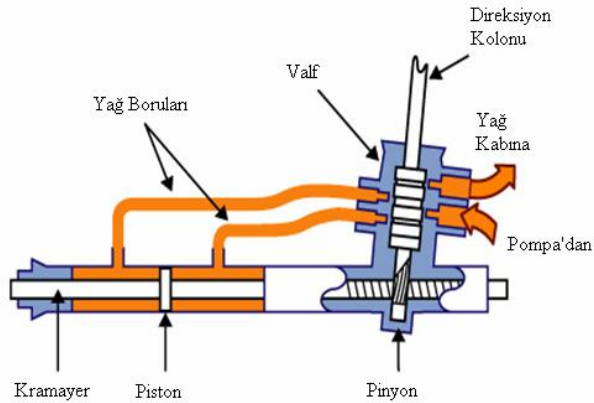
4.3.2 Hidrolik güç takviyeli yönlendirme sistemleri

Araçlardaki aerodinamik yapının gelişmesiyle ve lastik teknolojisinin ilerlemesiyle yol tutuşu artmış, tekerlekler ve yol arasındaki sürtünmesi kuvveti artmıştır. Böylece mekanik yönlendirme sistemlerinde yüksek olan döndürme kuvveti daha da artmıştır ve bu durum konfor ve sürüş kolaylığı açısından büyük bir dezavantaj olmuştur. Buna çözüm olarak hidrolik güç takviyeli yönlendirme sistemi geliştirilmiştir. Bu sistemin temel prensibi sıvı basıncını kullanarak sürücünün direksiyonu çevirmesine yardımcı olmaktır.



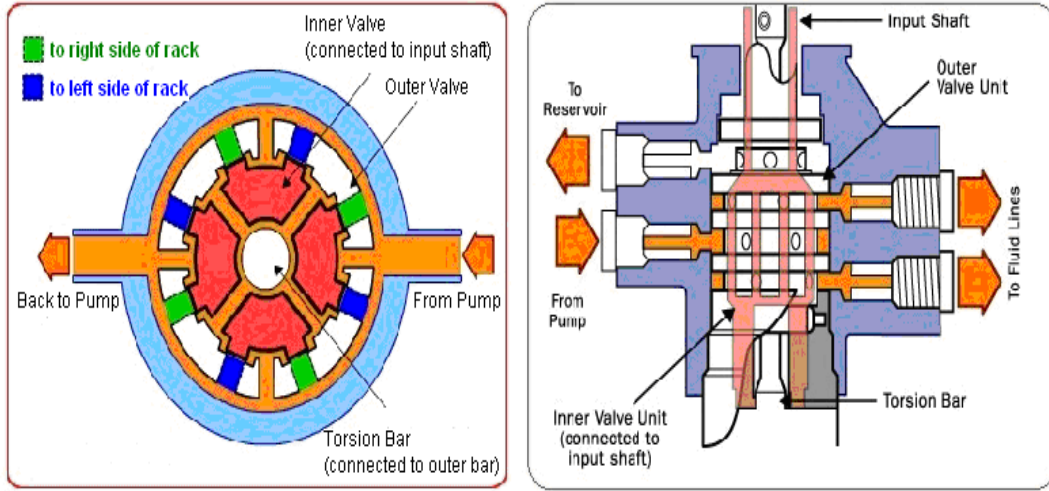
Şekil 4.17 : Hidrolik güç takviyeli yönlendirme sistemi [15].

Hidrolik sistem mekanik yapıya ek olarak hidrolik piston, hidrolik pompa, yön kontrol valfi, yağ tankı, hidrolik sıvıdan ve diğer ara elemanlardan oluşur. Öncelikle pompa aracın içten yanmalı motorundan aldığı tahrikle hidrolik basıncı meydana getirir. Sürücü direksiyona tork uyguladığında direksiyon milinin altındaki yön kontrol valfi konum değiştirir. Bu değişimle birlikte valf içindeki bazı sıvı kanalları açılır ve torkun yönüne göre basınçlı sıvı pistonun ön ya da arka haznesine gönderilir. Pistonda oluşan basınç çizgisel kuvvet olarak kremayer dişlisine destek olur ve sürücünün direksiyonu çevirmesi kolaylaşır.



Şekil 4.18 : Silindir-kremayer-valf-pinyon.

Hidrolik valf sürücünün uyguladığı torka göre pistonu sağa ya da sola iter ancak sürücünün herhangi bir tork uygulamadığı anlarda hidrolik pompadan valfe gelen basınçlı sıvı silindire gönderilmek yerine valfin egzoz kanalından tanka geri döner.



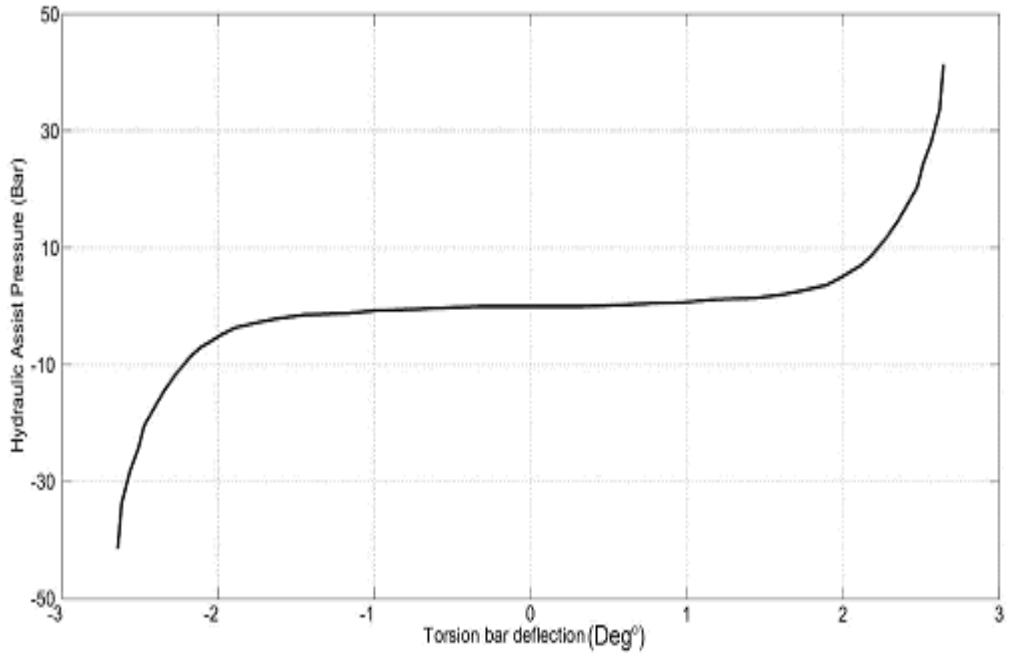
Şekil 4.19 : Hidrolik valfin iç yapısı[14].

Hidrolik pompa içten yanmalı motora mekanik olarak bağlı olduğu için aracın devri arttığında valfe giden hidrolik basıncın ve buna bağlı olarak kuvvetin de artması beklenebilir. Böyle durumlar sorunlara yol açabileceğinden hidrolik pompanın çıkışında basınç regülatörü (akış kontrol valfi) kullanılır.

Sürücünün direksiyonu döndürmek için uygulayacağı kuvvetin minimum düzeylere inmesi ve her açıda, her hızda uygulanan kuvvetin yaklaşık aynı seviyelerde olması hidrolik güç takviyeli sistemin en önemli getirisidir. Bunun yanı sıra yoldan gelen istenmeyen titreşimlerin sürücüye daha az yansıtılması ve aracın yön kontrolünün zorlaştığı anlarda (lastik patlaması vb.) sürücüye destek olması bu sistemin bir başka avantajıdır.

Buna karşın parça sayısının artmasıyla sistemin maliyeti ve karmaşıklığı artmaktadır. Aynı zamanda bu elemanların kapladıkları alan ve yerleşimleri üreticiye ek bir yük getirmektedir. Hidrolik pompa içten yanmalı motor sayesinde tahrik edildiği için araç çalıştığı sürece motor için bir yük durumundadır. Bu yakıt ekonomisi bakımından tüketici için cazip olmayan bir durumdur.

Şekil 4.20’de direksiyon tork sensörünün burulma miktarına karşılık gelen “hidrolik sistem yardımcı kuvvet miktarı” verilmiştir. Burulma miktarı sürücünün direksiyon simidine uyguladığı torka bağlıdır. Burulma olmadığında (0°) hidrolik sistemden gelen yardımcı kuvvet sıfırdır. $(+)-1.5^{\circ} - (-)1.5^{\circ}$ ‘lik burulmalarda burulma açısıyla yardımcı kuvvet arasındaki ilişki lineer olarak kabul edilebilir. Bu değerler dışında kuvvet miktarı eksponansiyel olarak artmaktadır. Böyle bir sistem sayesinde direksiyonu döndürmek için sürücüye 2-3 Nm’den fazla kuvvet gerekmez.



Şekil 4.20 : Tork sensörü burulma miktarı [19].

4.3.3 Elektro hidrolik güç takviyeli yönlendirme sistemleri

Hidrolik güç takviyeli yönlendirme sistemleri mekanik sistemlere göre birçok avantaj sağlamasına rağmen yakıt tasarrufu bakımından büyük bir dezavantaja sahiptir. Hem hidrolik sistemin artılarını barındıran hem de yakıt açısından %4 ‘e kadar bir iyileşme sağlayabilmek için elektro hidrolik güç takviyeli sistemler(EHPS) geliştirmiştir. Bu sistemde hidrolik sisteme ek olarak bir elektrik motoru, bir güç ünitesi ve direksiyon miline yerleştirilmiş tork sensörü kullanılmıştır. Pompa hareketini içten yanmalı motor yerine bir elektrik motorundan almaktadır. Elektrik motoru ise enerjisini güç ünitesi aracılığıyla araç aküsünden sağlamaktadır. [8]

Sürücü direksiyonu çevirmeye başladığında tork sensörü tarafından mil üzerindeki güç algılanır ve bu bilgi güç ünitesine gönderilir. Güç ünitesi ise gelen tork değerine göre elektrik motoruna gerekli dönüş hızını sağlayacak enerjiyi aktarır. Elektrik motorunun dönüşüyle hareketlenen pompa hidrolik gücü hidrolik güç takviyeli sistemdeki gibi önce valfe daha sonra da silindire aktarır.

Sürücü düz bir yolda ilerlediğinde direksiyona herhangi bir tork uygulamadığı için güç ünitesi elektrik motoruna rölantide çalışabilecek kadar enerji gönderir (minimum hızda). Elektrik motorunun sürekli yüksek hızlarda çalışmaması da yakıtta tasarruf sağlar. Buna ek olarak sistemin bir başka avantajı da pompa içten yanmalı motorun yanına ya da yakınına konulmak zorunda değildir. Bu da bir serbestlik sağlar.



Şekil 4.21 : Hidro elektrik güç takviyeli yönlendirme sistemi [20].

4.3.4 Elektrik güç takviyeli yönlendirme sistemleri

Elektrik güç takviyeli yönlendirme sistemlerinde(EPS) direksiyona yön verebilmeyi kolaylaştırabilmek için hidrolik silindir, hidrolik sistem yerine elektrik motoru ve elektrik kontrol ünitesi kullanılmaktadır. Elektrik motoru, kontrol ünitesi, tork sensörü, konum sensörü bu sistemin temel parçalarıdır. Sürücü direksiyona tork uyguladığında bu güç tork sensörü tarafından algılanır ve kontrol ünitesine gönderilir. Burada değerlendirildikten sonra elektrik motorunun gelemesi gerek konum hesaplanır ve gerekli sinyal elektrik motoruna gönderilir. Elektrik motorunun istenen konuma gelip gelemediği üzerine ya da direksiyon miline bağlı bir konum sensörüyle geri besleme olarak okunur.

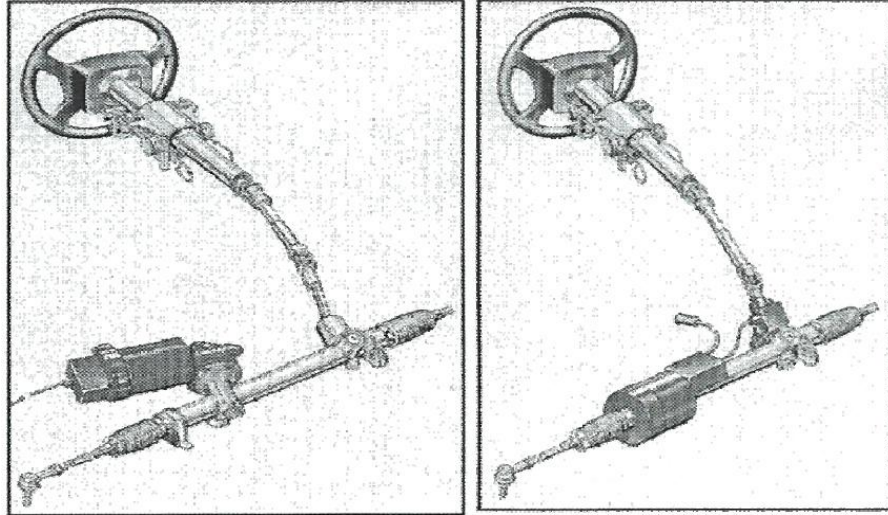
Bu sistem elektrik motoru direksiyonun çevrilmesi için gereken yardımcı gücü aktardığı noktaya göre 4'e ayrılır:

Kolon Tipi : Motor kolon üzerindedir.

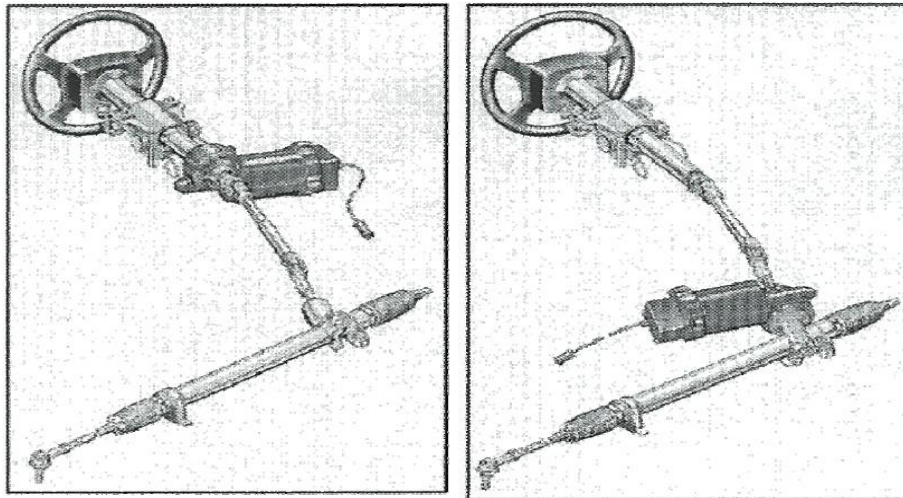
Pinyon Tipi : Pinyon dişlisiyle kremayer dişlisinin bağlandığı noktaya etki etmektedir.

Çiftli Pinyon Tipi : Kremayer dişlisi üzerinde ayrı noktalarda iki pinyon dişlisi bulunur. Birinci pinyon dişliden milden gelen tork etki ederken ikinci pinyon dişliden elektrik motorundan gelen tork etki etmektedir.

Dişli Tipi : Elektrik motoru kremayer dişlisine bağlıdır.



Şekil 4.22 : Çiftli pinyon tipi – dişli tipi [13].



Şekil 4.23 : Kolon tipi- pinyon tipi [13].

Pompa ve piston gibi hareketli, srtnmesi yksek, hortum ve valf gibi direnci yksek elemanların kullanılmaması verimi olduka arttırmıřtır. Bu sayede de yakıt tasarrufu bakımından mekanik sistemlere ok yaklařılmıřtır. Buna ek olarak iyi bir EK sistemi oluřturulduėunda manevra kabiliyeti de artar.

Hidrolik altyapılı sistemlerde elemanlar kaput ierisine yerleřtirilir. Alanın dar olması sebebiyle bu durum ciddi bir sorun olarak kabul edilebilir. Fakat elektrik g takviyeli sistemlerde elektrik motorunun direksiyon kolonuna da baėlanabilir olması, bunun yanında diėer elemanların (EK, sensrler) kaput iinde bulunma zorunluluėunun olmaması bu sorunu ortadan kalkar.

Bu sistemin bir bařka zelliėi de src tork uygulamayı bıraktıėında direksiyon sistem tarafından otomatik olarak sıfır konumuna getirilir.

4.3.5 Steer-by-wire ynlendirme sistemleri

Ynlendirme sistemleri ierisinde en son teknoloji sayılabilecek olan bu sistemi diėer geleneksel sistemlerden ayıran en nemli zellik direksiyon simidi ile tekerlekler arasında mekanik bir baėlantının bulunmamasıdır. Mekanik aksamaların yerini kablolar, sensrler, aktatrler ve kontrolc almıřtır. Son yıllarda bu ynlendirme sistemiyle ilgili birok alıřma yapılmasına raėmen henz gnlk hayata tam olarak yansıdaėı sylenemez.

Bu sistem st kısımda direksiyon altına monte edilmiř bir elektrik motoru ve buna baėlı bir pozisyon sensr, alt kısımda ise kremayer diřlisine baėlı bir veya tercihen iki(gvenlik sebebiyle) elektrik motoruyla bunlara ait pozisyon sensrlerini ierir. Sensrlerden aldıėı bilgiyi iinde barındırdıėı kontrol algoritmalarında deėerlendiren ve motorları gerekli řekilde sren bir elektronik birim mevcuttur.

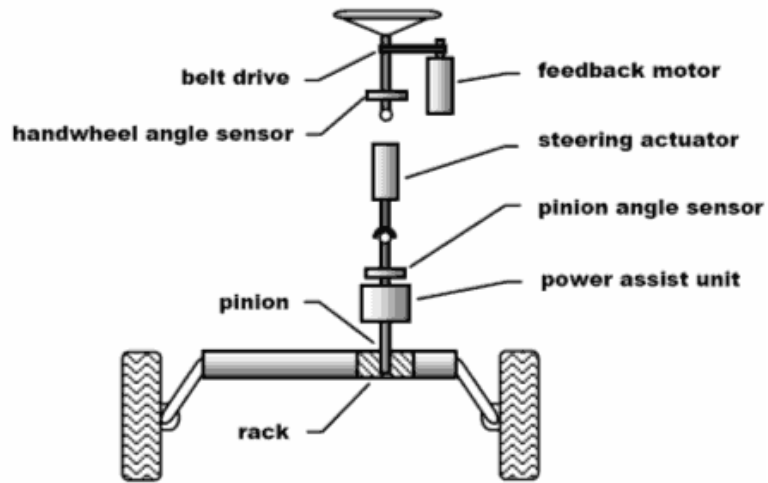
Src direksiyonu evirmeye bařladıėında direksiyona baėlı olan sensrdeki mutlak aı bilgisi EK tarafından okunur. Bu referans deėere gre kremayer zerindeki elektrik motorları referans deėere oturacak řekilde kontrol algoritması aracılıėıyla pozisyonlandırılır. İstenilen konuma gelip gelmediėi motorlar zerindeki konum sensrleri vasıtasıyla geri besleme olarak algılanır. Bu sensrlerin diėer bir grevi srř esnasında yoldan gelen titreřimleri algılamak ve bu verileri kontrol nitesine gndermektir.

Kontrol ünitesi yoldan gelen bu verilerin istenmeyen kısmını filtre eder ve sürücüye yol hissini aktarmak için direksiyona bağlı olan motoru sürer. Bu işlem esnasında motorun doğru konumlanıp konumlanmadığı motor üzerindeki sensör aracılığıyla EKÜ tarafından sürekli okunur. Direksiyon simidine bağlı olan motorun bir başka görevi de direksiyona yük bindirerek boşa kalmasını engellemektir.

Steer-by-wire yönlendirme sistemi diğer yöntemlere sistemlerine göre birçok avantaja sahiptir. İlk olarak aracı yönlendirmek için harcanacak olan kuvvet sürücü tarafından belirlenebilir. Direksiyon simidiyle tekerlekler arasında herhangi bir mekanik aksam olmaması bu esnekliği getirir. Direksiyona bağlı olan motora sürücünün belirleyeceği miktarda yük kontrol ünitesi tarafından verilerek sürüş konforu artırılmış olur. Yine sürücünün isteğine göre direksiyona verilen yol hissi bilgileri(özellikle titreşimler) artırılabilir, azaltılabilir ve hatta tamamen engellenebilir. Bu sistemde direksiyon milinin olmaması kaza anında direksiyondan sürücüye gelebilecek muhtemel zararları engeller. Milin olmaması araç dizaynını yapan mühendisler için de ergonomik açıdan bir avantaj sağlar.

Hidrolik veya elektro hidrolik sistemdeki gibi yağ kullanılmaması ve yakıt tasarrufu sağlaması çevresel açıdan iyi bir getirdir. (dönme oranının değişebilmesi)

Tüm bu avantajlarına rağmen yeni ve yönlendirme sistemleri için denenmemiş farklı bir teknoloji olması kullanıcılarda bu sistem için bir güven sorunu oluşturabilir.

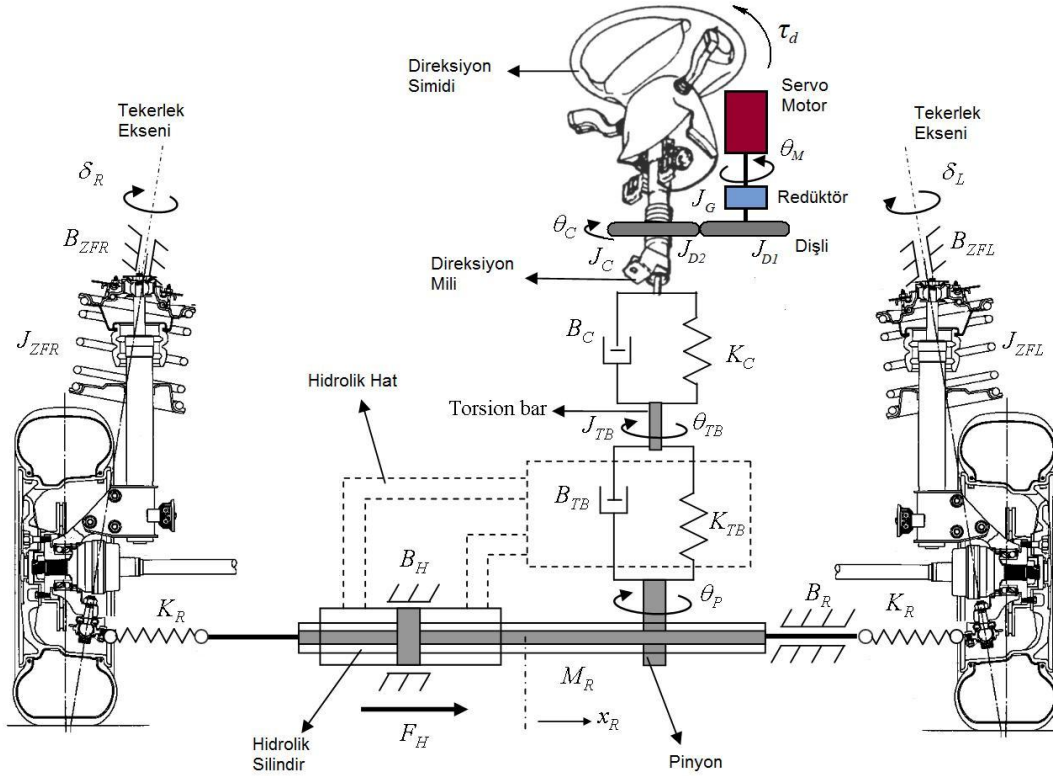


Şekil 4.24 : Steer-by-wire.

4.4 Otonom Araç İçin Direksiyon Modeli

Aracımızdaki yönlendirme sistemi şu ana kadar anlattığımız tüm standart yapıların dışında kalmaktadır. Şekilde de görüldüğü gibi araç üzerinde hidrolik güç takviyeli bir direksiyon sistemi mevcuttur. Bu çalışmada asıl yapıya ek olarak bir servo motor, ona bağlı bir redüktör ve iki dişli eklenmiştir.

Servo motor otonom konumda kontrolörden gelen sinyale göre sistemi hareketlendirecek aktüatör olarak kullanılmaktadır. Araç üzerinde ölçülen ve hesaplanan değerlere göre servo motorun ürettiği tork yetersiz kalabileceğinden motorda üretilen tork bir redüktör aracılığı ile beş kat yükseltilmiştir. Burada üretilen tork bire bir oranlı iki dişli ile direksiyon miline aktarılmıştır.



Şekil 4.25 : Proje aracı üzerindeki direksiyon yapısı [19].

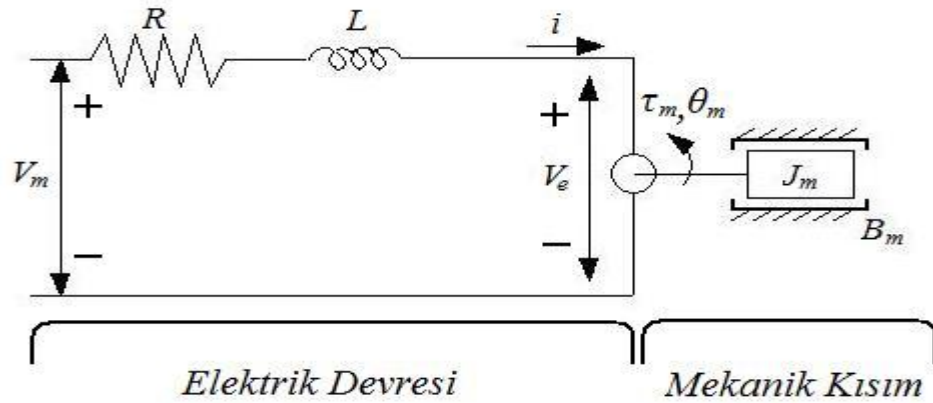
Araç manuel konumda çalışırken, yani aracı sürücü kullandığı anlarda, sistem girişleri sürücünün direksiyon simidine uyguladığı tork ve dönüş esnasında tekerleklerin sürtünmesinden kaynaklanan bozucu yanıl kuvvet olacaktır.

Servo motor ve onla beraber sisteme eklene parçalar giriş tarafından bakıldığında yük olarak gözükecektir. Bu noktada hidrolik destek sistemi sürücünün aracı yönlendirmesine yardımcı olacaktır.

Araç otonom konuma geçtiğinde ise sisteme güç girişi servo motor tarafından olacaktır. Sistemin bozucu girişleri yine tekerleklerden yanal kuvvet olarak gelse de, buna ek olarak sürücünün direksiyon simidine uygulayacağı tork da sistemde bozucu giriş olarak davranacaktır.

4.4.1 Matematik modelin çıkartılması

4.4.1.1 Servo motor matematik modeli



Şekil 4.26 : Servo motor şeması.

Elektrik devresine ait denklemler yazılacak olursa:

$$V_m(t) = Ri(t) + L \frac{di}{dt} + V_e \quad (4.1)$$

$$V_e = K_e \omega(t) \quad (4.2)$$

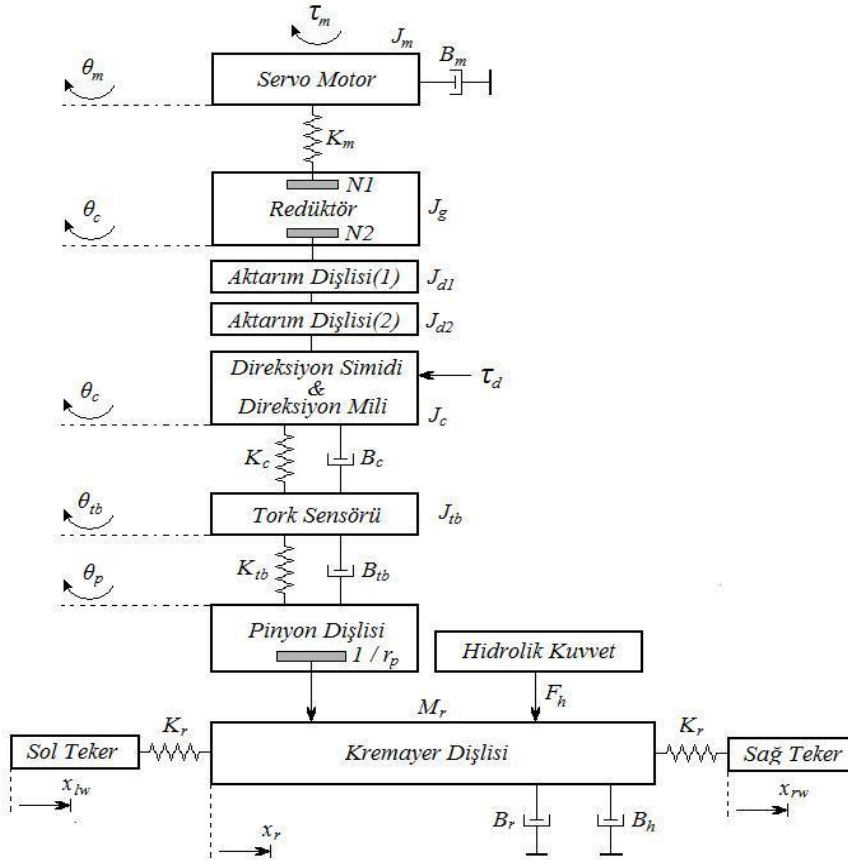
$$\omega = \frac{d\theta_m}{dt} = \dot{\theta}_m \quad (4.3)$$

Akıma bağlı mekanik kısma ait denklemler ise:

$$i = \frac{\tau_m}{K_t} \quad (4.4)$$

$$\tau_m = j_m \ddot{\theta}_m + B_m \dot{\theta}_m + 0.2\tau_c \quad (4.5)$$

4.4.1.2 Hidrolik güç takviyeli mekanik yönlendirme sisteminin servo motor ile birlikte matematik modeli



Şekil 4.27 : Sistem modeli.

Motor tarafından gelen tork motor milinin burulması varsayılabilir. Bu durumda mile gelen tork:

$$\tau_c = K_m(\theta_m - 5\theta_c) \quad (4.6)$$

Direksiyon mili, dişli kutusunun çıkış mili ve aktarım dişlileri aynı hızda döndüğü için ataletleri tek bir atalet gibi düşünülebilir.

$$j_t = j_c + j_g + j_{d1} + j_{d2} \quad (4.7)$$

Dişli dönüş miktarını beşe bir oranında azaltır. Bu durumda aktarılan dişli kutusu girişinin beş katının verimle çarpılmış halidir.

$$N_1 = 5N_2 \quad (4.8)$$

$$5\eta\tau_c = j_t \ddot{\theta}_c + B_c \left(\dot{\theta}_c - \dot{\theta}_{tb} \right) + K_c (\theta_c - \theta_{tb}) + \tau_d \quad (4.9)$$

Pinyon dişlisi rijit kabul edilirse tork sensöründeki burulma miktarı sensörün girişindeki dönüş miktarı ile kremayerin ilerleme miktarının pinyon dişlisindeki dönüş miktarına denk olan değer farkı kadardır. $\left(\theta_{tb} - \frac{x_r}{r_p} \right)$

$$B_c \left(\dot{\theta}_c - \dot{\theta}_{tb} \right) + K_c (\theta_c - \theta_{tb}) = j_{tb} \ddot{\theta}_{tb} + B_{tb} \left(\dot{\theta}_{tb} - \frac{\dot{x}_r}{r_p} \right) + K_{tb} \left(\theta_{tb} - \frac{x_r}{r_p} \right) \quad (4.10)$$

Hidrolik kuvvet kremayer dişlisini tahrik ettiğinden pinyondan gelen kuvvetin yanı sıra pozitif kuvvet olarak denkleme eklenir.

$$\frac{B_{tb}}{r_p} \left(\dot{\theta}_{tb} - \frac{\dot{x}_r}{r_p} \right) + \frac{K_{tb}}{r_p} \left(\theta_{tb} - \frac{x_r}{r_p} \right) + F_h =$$

$$M_r \ddot{x}_r + B_r \dot{x}_r + B_h \dot{x}_r + K_r (x_r - l_{sa} \sin \delta_l) + K_r (x_r - l_{sa} \sin \delta_l) + F_l$$
(4.11)

Denklemin daha basit yazılabilmesi için tekerleklerin kremayer direkt bağlandığı varsayıldı. Bu durumda K_r yok sayıldı ve tekerleklerin mille birlikte sabit hareket ettiği varsayıldı. Bu durumda sadece yerden gelen sürtünme ele alındı.

$$F_l \cong 2B_{tr} \dot{x}_r$$
(4.12)

Hidrolik kuvvetin değeri :

$$F_h = P_a f_h \left(\theta_{tb} - \frac{x_r}{r_p} \right)$$
(4.13)

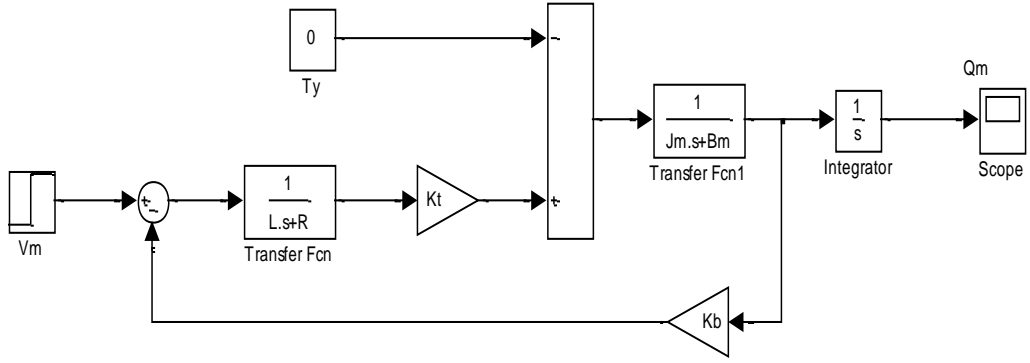
Denklemler yerine yazıldığında:

$$\frac{B_{tb}}{r_p} \left(\dot{\theta}_{tb} - \frac{\dot{x}_r}{r_p} \right) + \frac{K_{tb}}{r_p} \left(\theta_{tb} - \frac{x_r}{r_p} \right) + P_a f_h \left(\theta_{tb} - \frac{x_r}{r_p} \right) =$$

$$M_r \ddot{x}_r + B_r \dot{x}_r + B_h \dot{x}_r + 2K_r x_r + 2B_t \dot{x}_r$$
(4.14)

4.4.2 Simulink modelin kurulması

Servo motorun açık çevrim simulink modeli Şekil 4.28'deki gibidir:

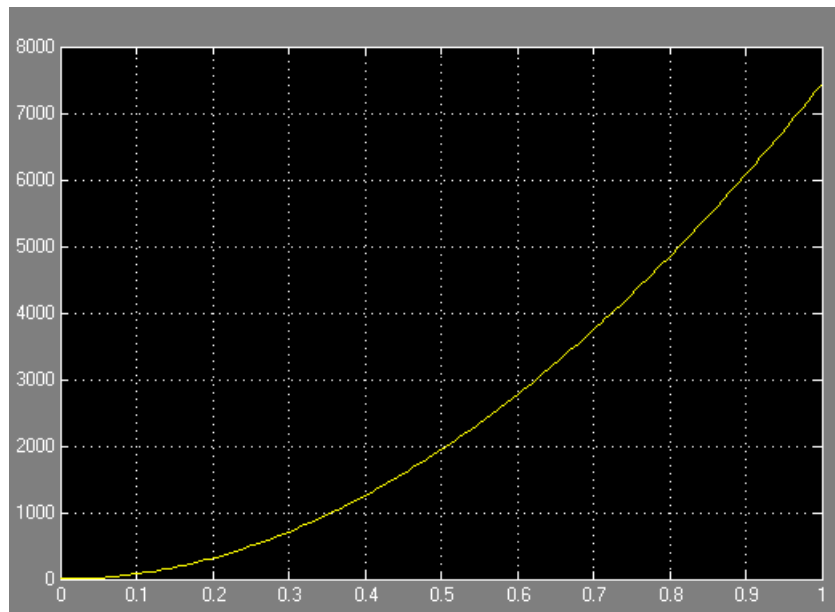


Şekil 4.28 : Servo motor açık çevrim simulink modeli.

Çizelge 4.1 : Servo motora ait değerler

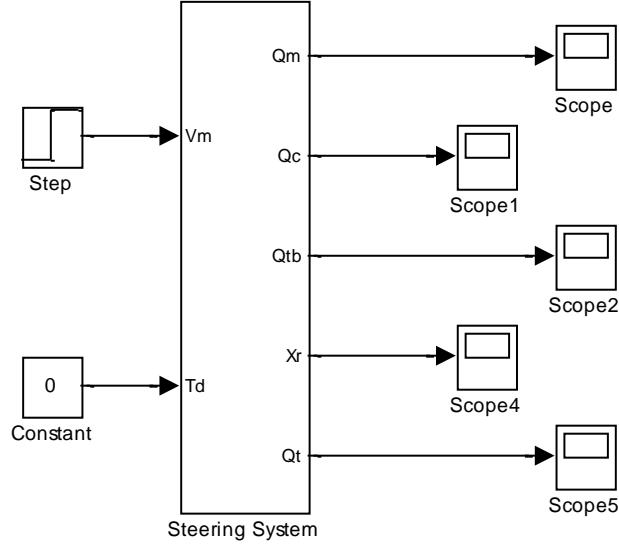
	Değişken	Değer	Birim
Servo Motor 8LVA22.R0030D000	R	1	Ω
	L	2.05	mH
	K_T	0.23	Nm/A
	K_E	0.13	$V/1000\text{min}^{-1}$
	J_M	0.000014	Kgm^2
	B_M	0.0000001	$\text{N-m}/(\text{rad/s})$
	K_M	20000	Nm/rad

Çizelge 4.1'deki değerler açık çevrim modele girildiğinde sistemin step cevabı aşağıdaki gibi olur:



Şekil 4.29 : Açık çevrim servo motor step cevabı.

Servo motora mekanik sistem yük olarak bağlanmış ve aşağıdaki model ortaya çıkmıştır. Aşağıda görüldüğü gibi bu model bir subsystem'e gömülmüş ve giriş çıkışlar şekil 4.30'daki gibi bağlanmıştır.

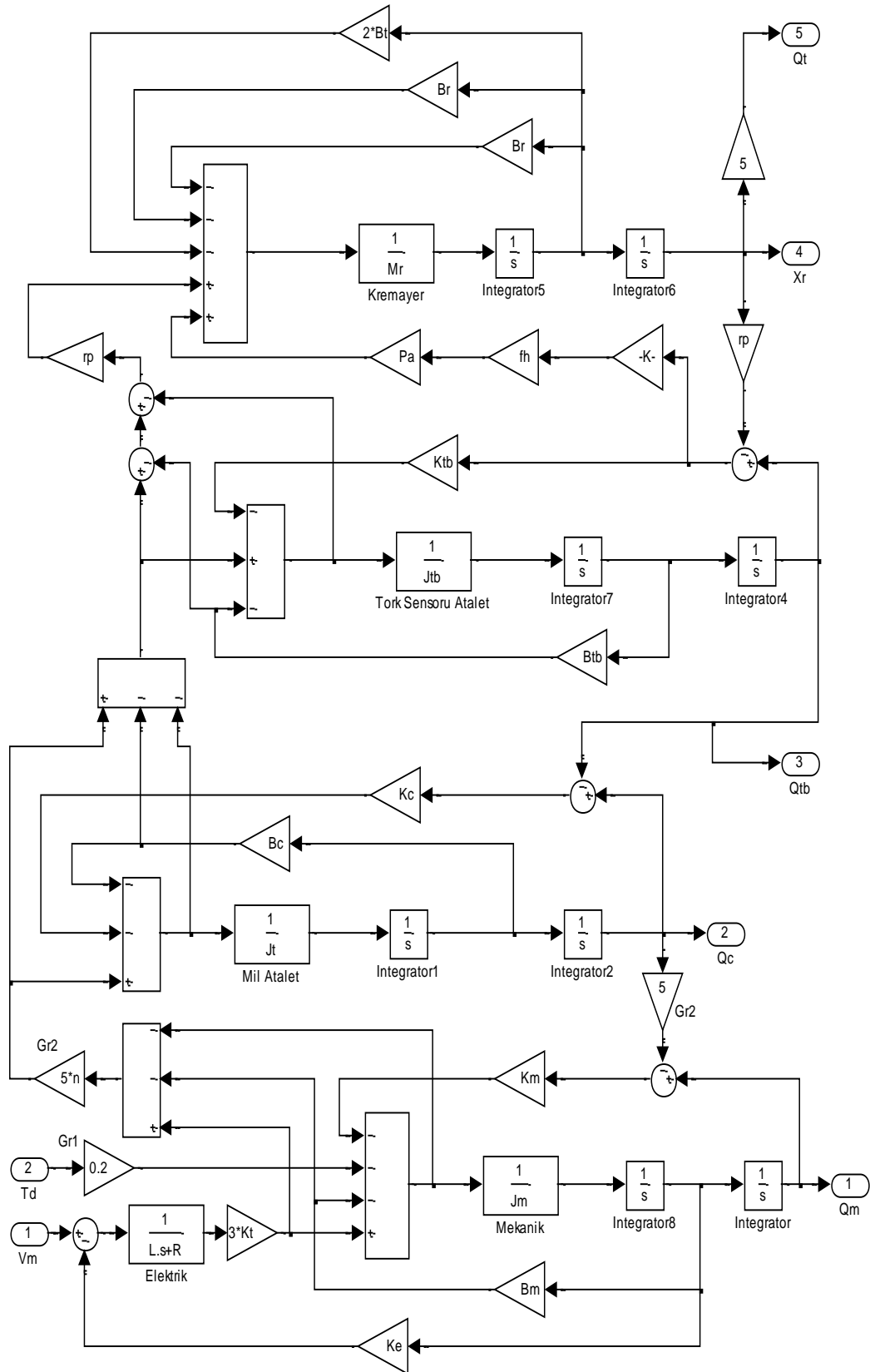


Şekil 4.30 : Açık çevrim simulink modeli.

Bağlantılar Şekil 4.30'daki gibi iken Çizelge 4.2, Çizelge 4.3 ve Çizelge 4.4'deki değerler sistemin parametreleri olarak girilmiş ve açık çevrim modelin step cevabı tekrar uygulanmıştır.

Çizelge 4.2 : Redüktör değerleri.

	Değişken	Değer	Birim
Redüktör 8GP30-060-005S2L4	n	96	%
	Gr1	0.2	-
	Gr2	5	-
	J_g	0.014	Kgm^2
	J_{D1}	0.008	Kgm^2
	J_{D2}	0.008	Kgm^2



Şekil 4.31 : Subsisitem (Steering system).

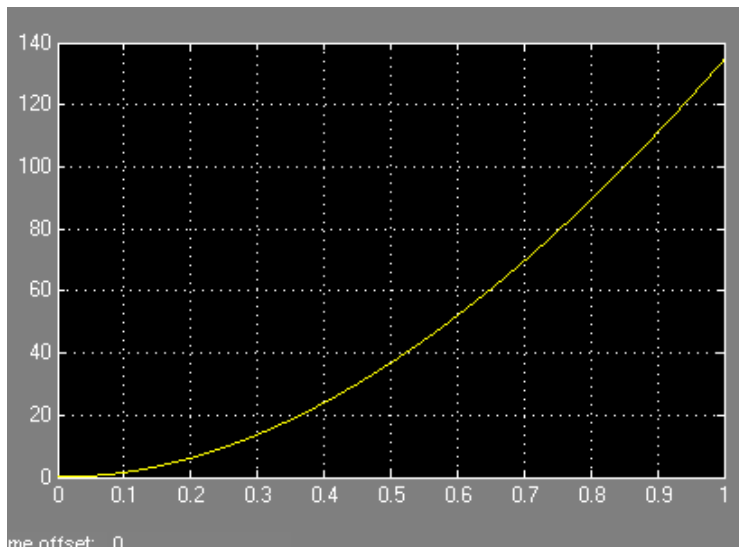
Çizelge 4.3 : Direksiyon sistemi değerleri.

	Değişken	Değer	Birim
Direksiyon Sistemi	J_C	0.04	Kgm^2
	B_C	0.02	N-m/(rad/s)
	K_C	1000	Nm/rad
	J_{TB}	0.0005	Kgm^2
	B_{TB}	0.01	N-m/(rad/s)
	K_{TB}	10	Nm/rad
	r_p	20	-
	M_R	10	Kg
	B_R	0.04	N-m/s
	K_R	100	Nm

Çizelge 4.4 : Hidrolik sistem değerleri.

	Değişken	Değer	Birim
Hidrolik Sistem	P_a	0.0008	m^2
	B_H	0.0001	N-m/(rad/s)

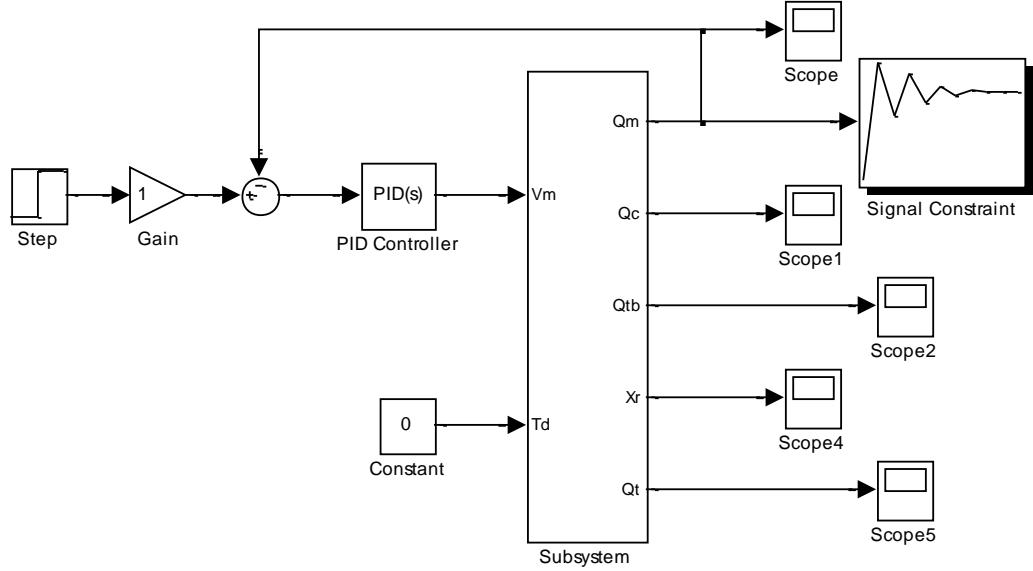
Sonuç olarak iki scope çıkışının da açık çevrim olmasından dolayı sonsuza doğru gitmesi olağandır. Bu iki sonuç karşılaştırıldığında açık çevrimde yüksüz modelin yüklü modele göre daha hızlı hareket ettiği, yüksüz modelin çıkışı 1 s içinde 7500 civarına yükselirken mekanik yüklü modelin çıkışının aynı süre içerisinde 140'a ulaşmadığı görülebilmektedir.



Şekil 4.32 : Açık çevrim servo motor step cevabı(mekanik yük bağlı).

4.4.3 Pozisyon kontrolcüsü tasarımı

Sistemde kontrolcü olarak PID seçilmiştir. Sistemin girişi V_m seçilmiştir. Çıkış ise enkoderin motor üzerinde bulunması sebebiyle Q_m olarak alınmıştır.

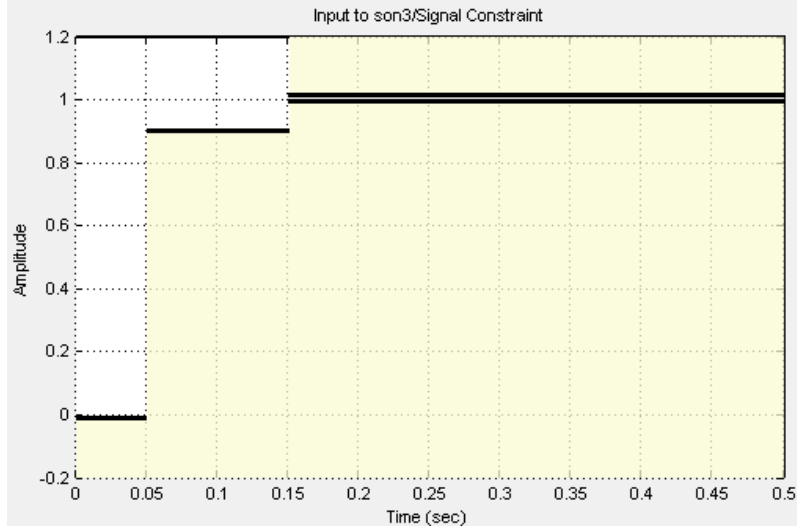


Şekil 4.33 : Kapalı çevrim simulink modeli.

PID katsayılarını(k_i , k_p , k_d) bulmanın birkaç yolu vardır. Bunlardan biri de “Sinyal Sınırlama” yöntemi ile MATLAB’e hesaplatmaktır. Bu yöntemin aşamaları şu şekildedir:

- Yukarıdaki şekildeki gibi sistem girişi ve çıkışı tanıladıktan sonra geri beslemeli bir şekilde seçilmiş kontrolör sisteme bağlanır. Buna ek olarak çıkışa “Sinyal Sınırlama” bloğu bağlanır. Bu metotla P, PI, PD veya PID kontrolör çeşitlerinin katsayıları bulunabilir.
- PID katsayıları tanımlanır ve sistemin hızına göre bu değişkenlere yaklaşık bir değer atanır. Sistem çok hızlı ise bu değerler için küçük, yavaşsa biraz daha büyük sayılar kullanılabilir. Bu işlem en fazla birkaç denemede sonuç verecektir.
- “Sinyal Sınırlama” bloğuna çift tıklayarak içine girilir. Şekilde görülen alt ve üst limit sınırlarının genlikleri, başlangıç zamanları ve bitiş zamanları seçilir.

Aslında bu işlemi yapılırken aynı zamanda istenilen sinyale ait yükselme zamanı, oturma zamanı, maksimum aşım gibi özelliklerde belirlenmiş olur.



Şekil 4.34 : Sinyal alt-üst sınırları.

Design requirement: Lower limit from 0 to 1 (sec)

Design requirement parameters

Segments:

Start		End		Slope	Weight
Time	Amplitude	Time	Amplitude		
0	-0.0100	0.2000	-0.0100	0	1
0.2000	0.7000	0.4000	0.7000	0	1
0.4000	1	1	1	0	1

Insert Delete

Şekil 4.35 : Alt limit belirleme.

Design requirement: Upper limit from 0 to 1 (sec)

Design requirement parameters

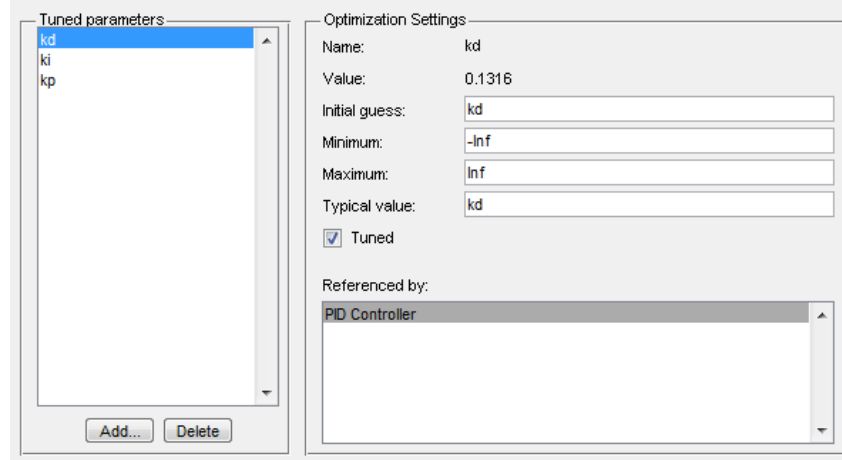
Segments:

Start		End		Slope	Weight
Time	Amplitude	Time	Amplitude		
0	1.0500	0.4000	1.0500	0	1
0.4000	1.0010	1	1.0010	0	1

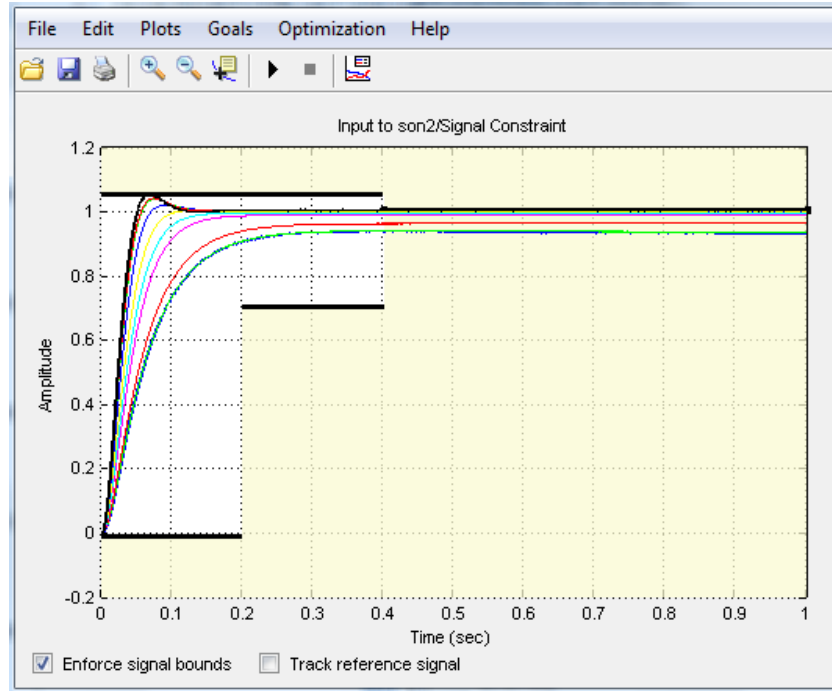
Insert Delete

Şekil 4.36 : Üst limit belirleme.

- Son olarak deęerlerinin bulunmasını istediđimiz parametreleri ařađıdaki řekilde seęmek ve programı alıřtırmak yeterlidir.



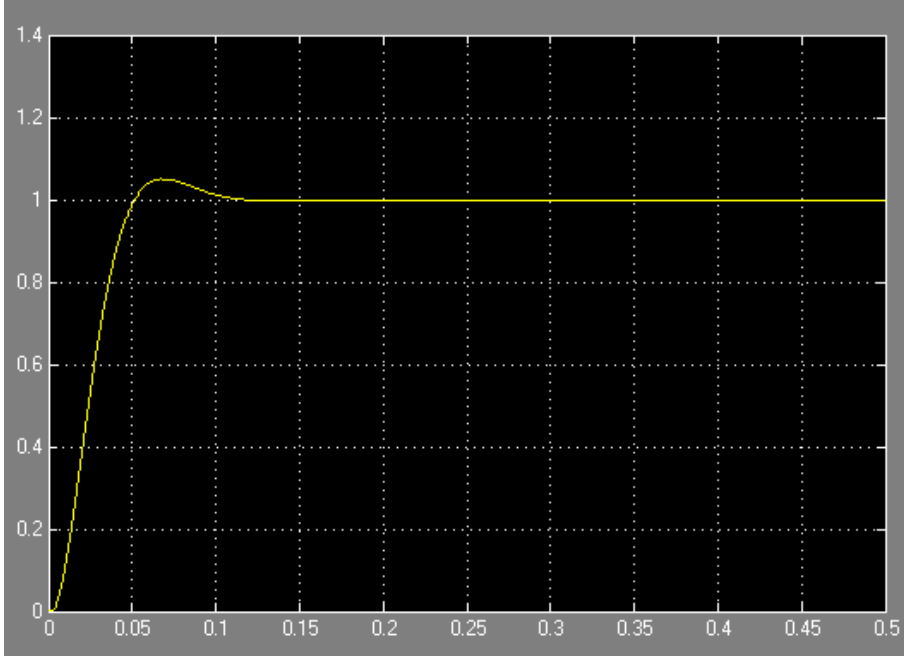
řekil 4.37 : Parametre giriři.



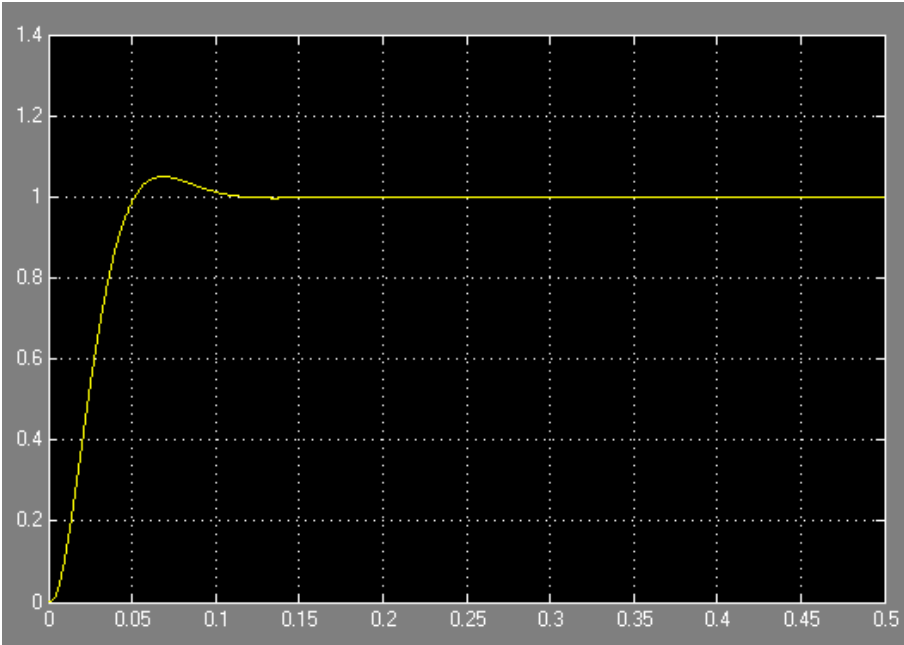
řekil 4.38 : İterasyon adımları.

Program iterasyon adımından sonra řekilde de grldđ istenilen sinyale biraz daha yaklařır. Sonun da belirlenen en uygun deęer(siyah) izilir. Bu metodu oluřturulan modele uyguladıđımızda yukarıdaki grafik elde edilmiřtir.

Sonuç olarak en uygun deęerler $k_p= 0.0755$, $k_d=0.1316$, $k_i=0.0017$ olarak bulunmuştur. Bu deęerlere göre kapalı çevrim sistem çalıştırıldığında çıkışlar aşağıdaki gibi olur.

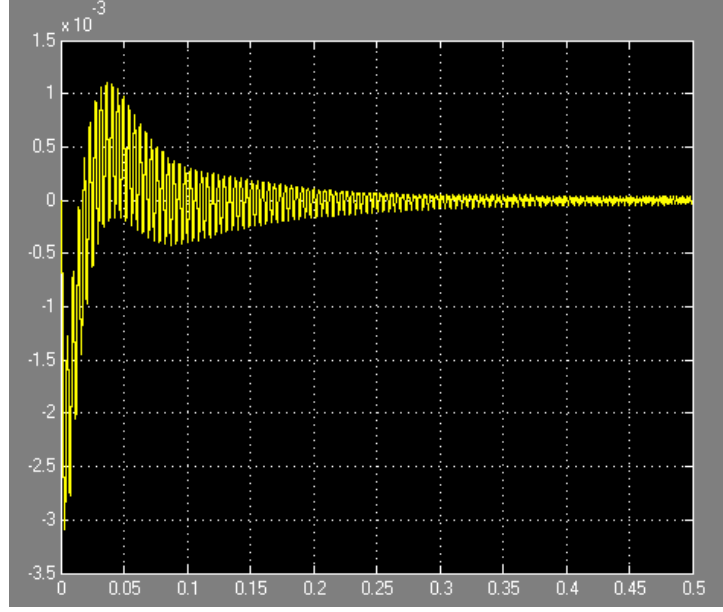


Şekil 4.39 : Sistemin step cevabı(Qm).



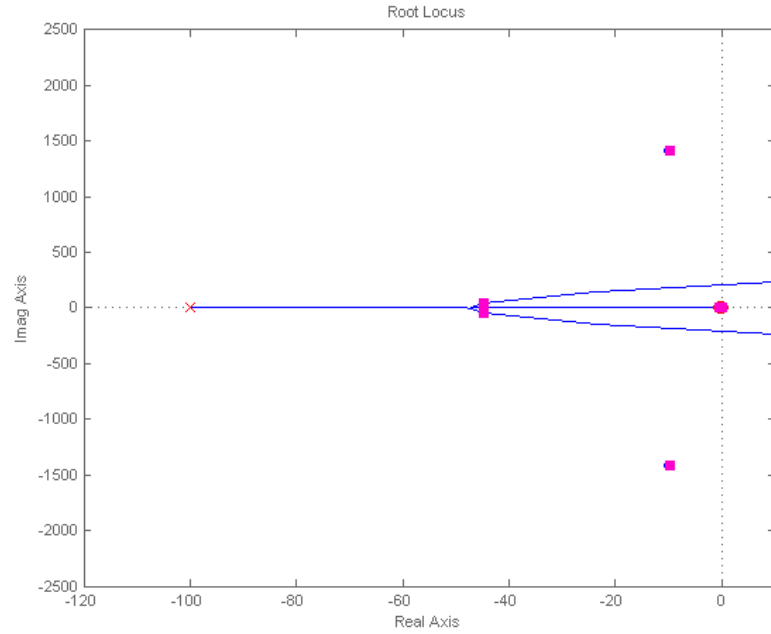
Şekil 4.40 : Sistemin step cevabı(Qt).

Sistemdeki mekanik istemdeki burulmalardan ötürü servo motor çıkışı ve tekerlek açısı arasında ufakta olsa bir fark oluşur. Bu fark aşağıdaki şekilde verilmiştir.



Şekil 4.41 : Çıkış sinyallerinin farkı($Q_t - Q_m$).

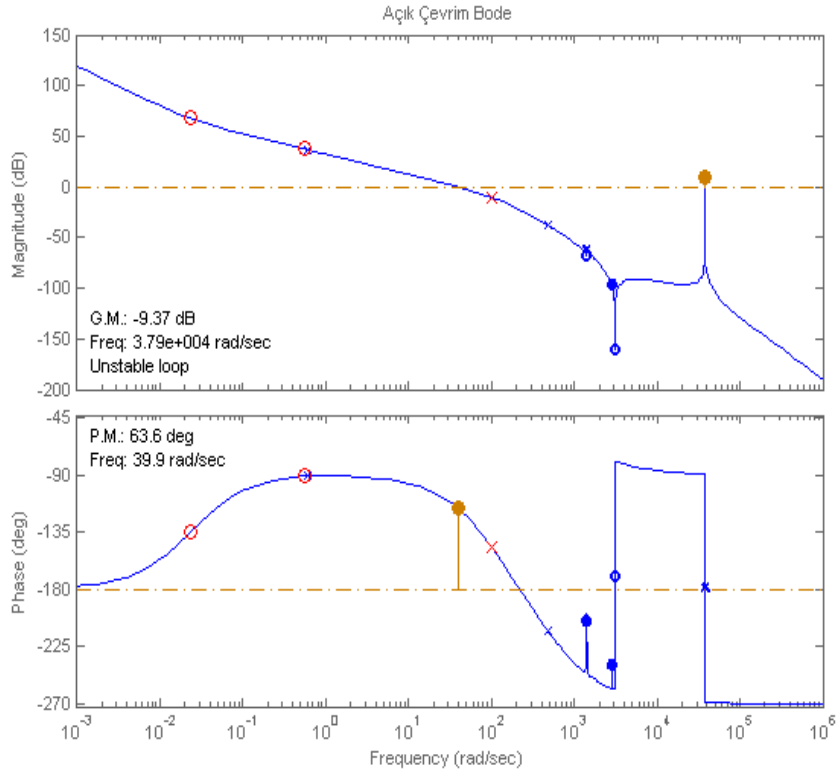
Sistemin root locus grafiğine bakarsak kökleri sanal eksenin negatif tarafında olduğu görülmektedir.



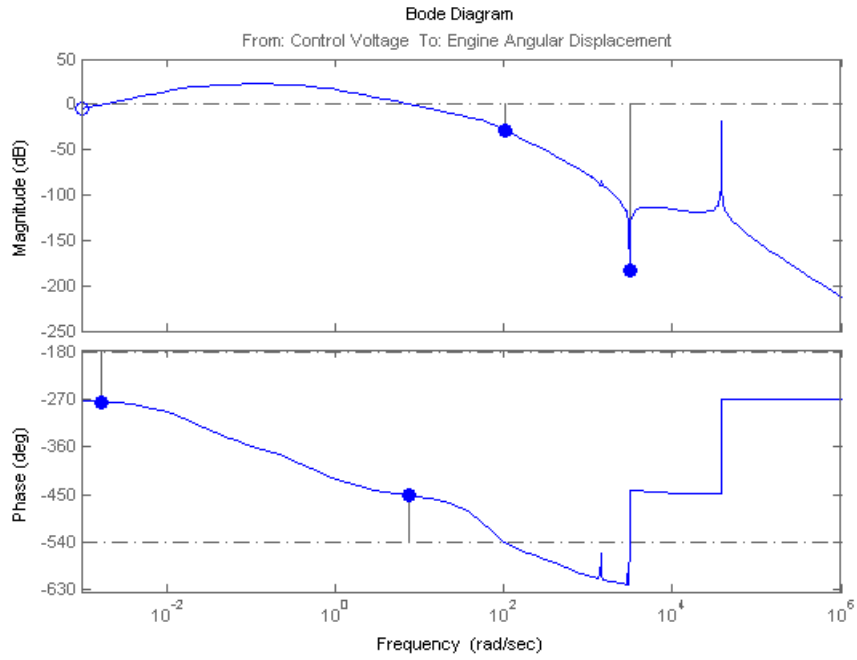
Şekil 4.42 : Sisteme ait Root Locus .

Açık çevrime göre çıkışı motor mili olarak çizdirilen bode diyagramında kazancın pozitif olduğu ve buna bağlı olarak sistemin stabil olmadığı görülmektedir.

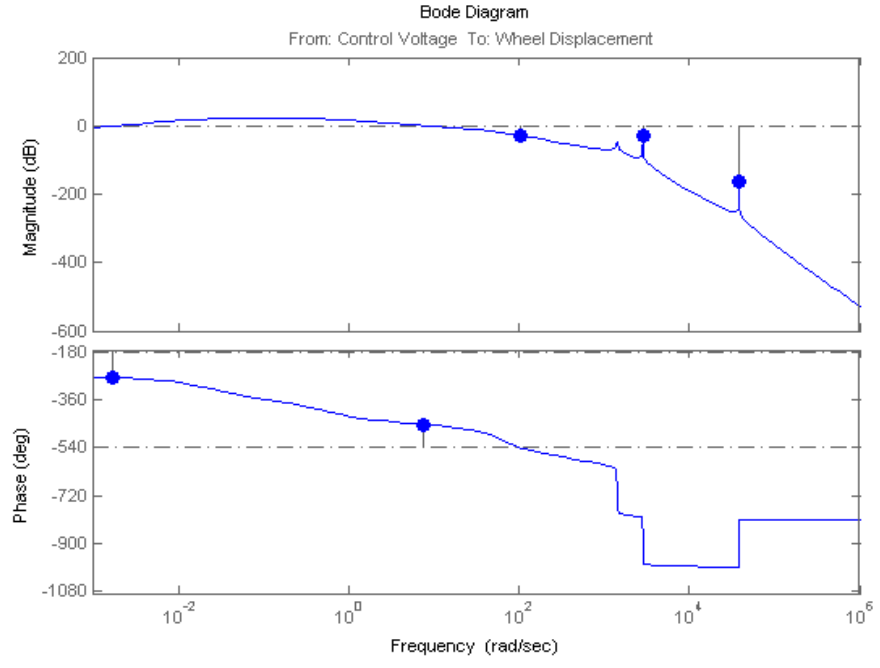
Aynı modelin kapalı çevrim bode diyagramları hem Qm hem de Qt çıkışı için çizdirilmiştir.



Şekil 4.43 : Açık çevrim bode diyagramı.



Şekil 4.44 : Kapalı çevrim bode diyagramı(çıkış Qm).



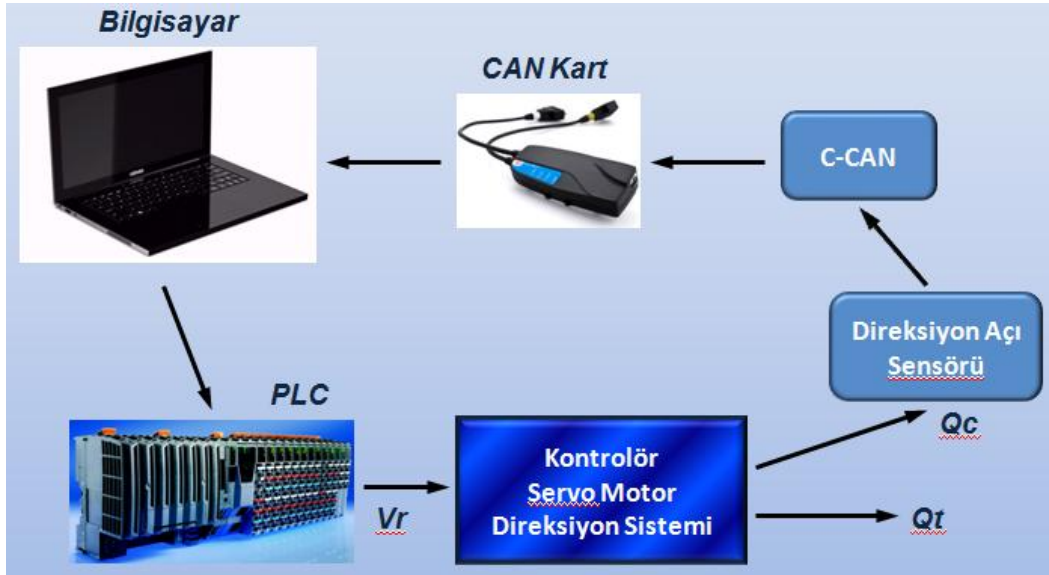
Şekil 4.45 : Kapalı çevrim bode diagramı (çıkış Qt).

5. SONUÇLAR

Otonom araçlar üzerinde yapılan çalışmalar her ne kadar sistemin bütünü bir anda ele almayıp sistemleri parça parça incelese de, zamanla bu çalışmalar bir bütün hale getirilmekte ve sayıları da gün geçtikçe artmaktadır.

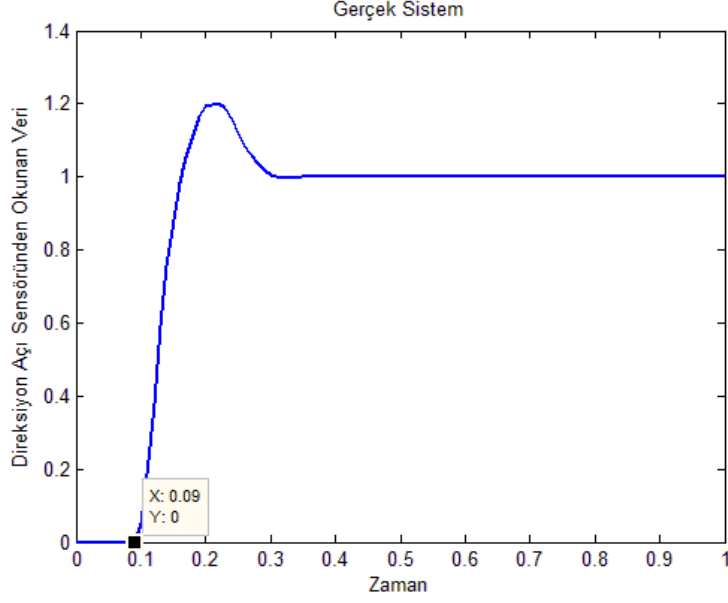
Yapılan bu çalışma da ise otonom araca ait sistemlerin montajı, elektriksel bağlantıları yapıldı. Bunlara ek olarak direksiyon sisteminin modelinin çıkartılıp, sisteme uygun bir kontrolör tasarlandı. Yine otonom araçta yapılması gereken işlemlerden biri olan aracın haberleşme ve kontrol ağı olan CAN hattına girildi. Bu sayede sistem üzerinde gerçek deney sonuçları elde edildi.

Gerçek sistemle yapılan bu deneylerdeki bağlantı Şekil 5.1’de verilmiştir. Bu yapıda dönüş açısı ve verileri tekrar toplanması işlemini bilgisayar üstlenmiştir.

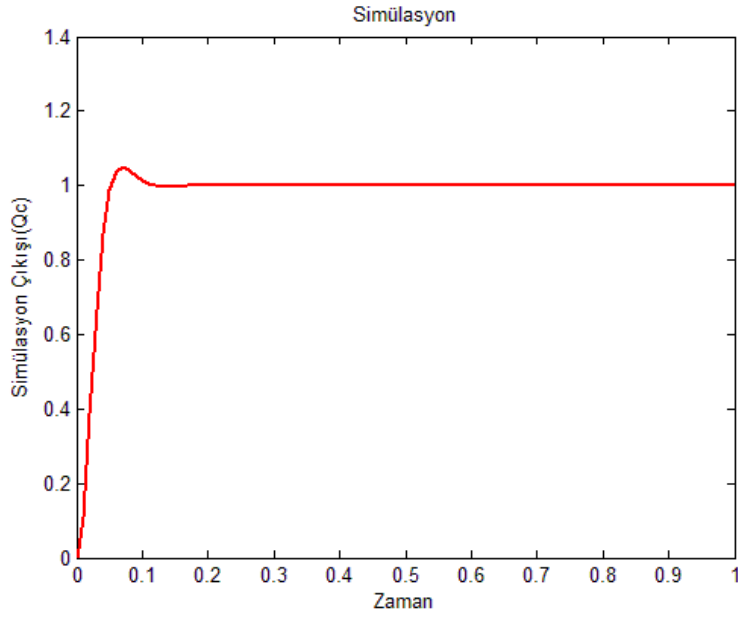


Şekil 5.1 : Deney düzeneği.

Şekil 5.2 ve Şekil 5.3’deki sonuçlar karşılaştırıldığında gerçek sistemden alınan çıkışta simülasyondan farklı olarak ilk tepki süresinin 0.9 saniyelik bir gecikme ile başladığı görülmektedir. Yine gerçek sistemde maksimum aşımın 20% değerinde 6% civarında olduğu ortadadır.



Şekil 5.2 : Gerçek sistem çıkışı.



Şekil 5.3 : Simülasyon çıkışı.

Maksimum aşımındaki bu farklılık gerçek sistemdeki bazı parametrelerin tam değerleri bilinmediğinden simülasyonlarda tahmini olarak girilmesidir. Sonuçlar üzerindeki diğer farklılık olan gecikme ise PLC ve CAN yazılımlarındaki (PLC içerisine gömülen kod EK-C'de verilmiştir) kod işleme süresi vb. direksiyon sisteminin dışındaki diğer sistemlerin gecikme sürelerinden kaynaklanır.

KAYNAKLAR

- [1] **Url-1** <http://www.dot.wisconsin.gov/statepatrol/docs/vantype.pdf> ,05.05.2011.
- [2] **Ünal, İ.**, 2006: CAN(Control Area Network) üzerinden PIC Programlama. Tez(Yüksek Lisans), Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta.
- [3] **Coşkun U.**, 2008. Controller Area Network ve Uygulaması. Tez(Yüksek Lisans), Gebze Yüksek Teknoloji Enstitüsü, Gebze
- [4] **Farsi, M., Ratcliff, K., Barbosa.**, 1999: An Overview of Control Area Network, IEE Computing and Control Engineering Journal, 10, 3, 113-120.
- [5] **Url-2** < <http://www.barissamanci.net/Makale/15/can-bus-nedir-can-protokolu-incelemesi> >, 29.04.2011.
- [6] **Robert Bosch GmbH, Stuttgart**, CAN Specification Version 2.0, 1991
- [7] **Aksoy, R., and Kurnaz, S.**, 2008: İnsansız Kara Araçları ve Muharebe Gereksinimleri. In Havacılık ve Uzay Teknolojileri Dergisi, , 17–19 January 2009.
- [8] **Sarıkaya A.**,2007, Güç Takviyeli Direksiyon Sistemlerinin Modellenmesi ve Kontrolü Tez(Yüksek Lisans), İTÜ, Fen Bilimleri Enstitüsü, İstanbul
- [9] **Gillespie T.D.** , “Fundamentals Of Vehicle Dynamics”, Warrendale, PA : Society of Automotive Engineers, 275-307 (1992).
- [10] **Sethi H.M.** , “Automotive Technology”, Hill Publishing Company Limited, New Delhi, 112-134 (1991).
- [11] **Megep** (Mesleki Eğitim ve Öğretim Sisteminin Güçlendirilmesi Projesi), Öndüzen ve Tekerlekler,2005.
- [12] **Durstine, John W.**“The Truck Steering System from Hand Wheel to road Wheel”
- [13] **Holt D. J.**, 2002. Electric steering: A revolution in steering technology, *Society of Automotive Engineers, Inc., USA*
- [14] **Url-3** www.howstuffworks.com
- [15] **Küçükaya T.**,2008, Direksiyon Sisteminin Matematiksel Modellenmesi ve Simülasyonu , Tez(Yüksek Lisans), Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli
- [16] **Ünal F.**,2007, Elektrikli Direksiyon Sistemi, Tez(Yüksek Lisans), İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul

- [17] **Yelken B.**,2006, Varolan Bir Direksiyon Sisteminin İyileştirilmesi İçin Yeniden Modellenmesi, Tasarı ve Denenmesi, Tez(Yüksek Lisans), Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara
- [18] **Url-4** www.wikipedia.com
- [19] **Çetin A.**,2009, Compliant Control of electric Power Assisted Steering Systems in Vehicles, , Tez(Doktora), Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul
- [20] **Harter W., Pfeiffer W., Dominke P., Ruck G., Blessing P.**, 2000. FutureElectrical Steering Systems: Realizations with Safety Requirements, Society of Automotive Engineers, Inc., USA
- [21] **L. Vlacic, M. Parent, F. Harashima**, Intelligent Vehicles Technologies

EKLER

EK A. : Araç Bilgileri

Araçın Teknik Özellikleri

Yakıt tüketimi (Şehriçi): 6,5 lt/100 km

Yakıt tüketimi (Şehriçi): 4,0 lt/100 km

Maksimum hız: 170 km/s

0-100 km/s hızlanma: 13,8 sn

Boyutlar (u/g/y): 4560/1728/1494 mm

Ağırlık: 1235 kg

Aktarma: Önden çekişli

Şanzıman: 5 ileri manuel

Lastikler: 195/55 R16

Frenler (ön/arka): Hava kanallı disk/kampana

Depo hacmi: 45 lt

Bagaj hacmi: 500 lt

Güç: 75 hp

Motor hacmi: 1.3 lt

EK B. : Can Hattı Erişim Kodları

```
#include <stdlib.h>
#include <stdio.h>
#include <memory.h>
#include <windows.h>
#include <conio.h>
#include <time.h>
#define stepim 30 // 1 ... 127
union datam{
short int aci;
unsigned char veri[2];} acim;

#include "canlib.h"
#define idsi 204
#ifdef __BORLANDC__
#pragma argsused
#endif

int Bitrate = BAUD_500K;
int Source = 0;
char Filename[512];
int Quiet = 0;
char deger=0;
int saydir,sayac=0;
char gonder[10];
int port=0;
HANDLE hComm = NULL;

COMMTIMEOUTS ctmoNew = {0}, ctmoOld;
DCB dcbCommPort={0};

    DWORD dwBytesRead;
#ifdef UNICODE
wchar_t cm1[] = L"COM1";
#else
char cm1[] = "COM1";
#endif

#ifdef UNICODE
wchar_t cm2[] = L"COM2";
#else
char cm2[] = "COM2";
#endif

void ErrorExit(char* id, canStatus stat)
{
    char buf[50];
    if (stat != canOK) {
        buf[0] = '\0';
        canGetErrorText(stat, buf, sizeof(buf));
        fprintf(stderr, "%s: failed, stat=%d (%s)\n", id, (int)stat, buf);
        exit(1);
    }
}

int InitCtrl(int ctrl)
```

```

{
    int stat;
    int hnd;
    hnd = canOpenChannel(ctrl, 0);
    if (hnd < 0) {
        ErrorExit("canOpenChannel", (canStatus)hnd);
    }

    stat = canSetBusParams(hnd, Bitrate, 0, 0, 0, 0, 0);
    if (stat < 0) {
        ErrorExit("canSetBusParams", stat);
    }

    stat = canBusOn(hnd);
    if (stat < 0) {
        ErrorExit("canBusOn", stat);
        exit(1);
    }

    return hnd;
}

void main(int argc, char* argv[])
{
    canStatus stat;
    int i;
    int hnd;
    FILE *f;
    int Ready;
    unsigned long MessageCount;
    unsigned long timeOffset;
    int Overrun;

    Filename[0] = '\0';
    for (i=1; i<argc; i++) {
        int tmp, c;
        if (sscanf(argv[i], "-%d%c", &tmp, &c) == 1) {
            Source = tmp;
        } else if (sscanf(argv[i], "-B%d%c", &tmp, &c) == 1) {
            switch (tmp) {
                case 1000 : Bitrate = BAUD_1M; break;
                case 500  : Bitrate = BAUD_500K; break;
                case 250  : Bitrate = BAUD_250K; break;
                case 125  : Bitrate = BAUD_125K; break;
                default : Usage(argc, argv);
            }
        } else if (strcmp(argv[i], "-q") == 0) {
            Quiet++;
        } else if (strcmp(argv[i], "-h") == 0) {
            Usage(argc, argv);
        } else if (argv[i][0] != '-') {
            strcpy(Filename, argv[i]);
        } else {
            Usage(argc, argv);
        }
    }

    if (Quiet == 0) {
        fprintf(stderr, "Logging to %s (candump -h for help; ESC to quit.)\n",
            Filename[0]? Filename : "standard output");
    }
    if(port==0)

```

```

hComm = CreateFile(cm1,
                  GENERIC_READ | GENERIC_WRITE,
                  0,
                  0,
                  OPEN_EXISTING,
                  FILE_ATTRIBUTE_NORMAL,
                  0);
else
hComm = CreateFile(cm2,
                  GENERIC_READ | GENERIC_WRITE,
                  0,
                  0,
                  OPEN_EXISTING,
                  FILE_ATTRIBUTE_NORMAL,
                  0);

if(hComm == INVALID_HANDLE_VALUE){
    if(port==0)printf("COM1 Hatali!");
    else printf("COM17 Hatali");
return ;
}

GetCommTimeouts(hComm,&ctmoOld);
ctmoNew.ReadIntervalTimeout      = 250;
ctmoNew.ReadTotalTimeoutMultiplier = 1;
ctmoNew.ReadTotalTimeoutConstant  = 250;
ctmoNew.WriteTotalTimeoutMultiplier = 1;
ctmoNew.WriteTotalTimeoutConstant  = 250;

SetCommTimeouts(hComm, &ctmoNew);

dcbCommPort.DCBlength = sizeof(DCB);
GetCommState(hComm, &dcbCommPort);

dcbCommPort.BaudRate=CBR_38400;//CBR_57600;
dcbCommPort.ByteSize=8;
dcbCommPort.StopBits=ONESTOPBIT;
dcbCommPort.Parity=NOPARITY;
if(!SetCommState(hComm, &dcbCommPort))return ;
    canInitializeLibrary();
    hnd = InitCtrl(Source);
    if (strlen(Filename) > 0) {
        f = fopen(Filename, "w");
        if (!f) {
            fprintf(stderr, "Can't create log file '%s'.\n", Filename);
            canClose(hnd);
            exit(1);
        }
    } else {
        f = stdout;
    }
timeOffset = canReadTimer(hnd);

if (Quiet == 0) {
    time_t t;
    time(&t);
    fprintf(f, "; Logging started at %s", ctime(&t));
    fprintf(f, "; (x = Extended Id, R = Remote Frame, o = Overrun, N =
NERR)\n");
    fprintf(f, "; Ident xRoN DLC Data 0.....7
Time\n");

```

```

    }

    Ready = FALSE;
    MessageCount = 0;
    Overrun = FALSE;

    while (!Ready) {

        while (!kbhit()) {
            long id;
            unsigned int dlc, flags;
            unsigned char msg[8];
            DWORD time;
            long t, timeFrac, timeInt;
            unsigned int i;
short int aci;

            do {
                stat = canRead(hnd, &id, msg, &dlc, &flags, &time);
                if (stat == canOK && (Quiet <= 1)) {
                    if ((flags & canMSG_ERROR_FRAME) == 0) {
                        if(id==idsi) fprintf(f, "%8lu%c%c%c%c %02lu ",
                            id,
                            flags & canMSG_EXT      ? 'x' : ' ',
                            flags & canMSG_RTR      ? 'R' : ' ',
                            flags & canMSGERR_OVERRUN ? 'o' : ' ',
                            flags & canMSG_NERR      ? 'N' : ' ', // TJA
1053/1054
                            dlc);
                    } else {
                        fprintf(f, " (%04lx) Error ", id);
                    }
                }
                if(id==idsi){ if ((flags & canMSG_RTR) == 0) {
                    for (i=0; i < ((dlc > 8)? 8: dlc); i++) fprintf(f,
"%2x ", msg[i]);
                    for (; i<8; i++) fprintf(f, " ");
                }
                aci=0;
                acim.veri[0]=msg[4];
                acim.veri[1]=msg[3];
                fprintf(f, "aci= %d : %d ",acim.aci,deger);
                saydir++;
                if(saydir==1){
                    saydir=0;
                    sayac++;
                    if(sayac>=4)sayac=0;
                    gonder[0]=deger;
                    WriteFile(hComm, gonder, 1, &dwBytesRead, NULL);
                }

                } else {
                    fprintf(f, " ");
                }
            }

            t = time - timeOffset;
            if(id==idsi){if (t < 0) {
                timeFrac = (-t) % 1000;
                timeInt = (-t) / 1000;
                fprintf(f, "-%5ld.%03lu\n", timeInt, timeFrac);
            } else {
                timeFrac = t % 1000;
                timeInt = t / 1000;
                fprintf(f, "%6ld.%03lu\n", timeInt, timeFrac);
            }
        }
    }
}

```

```

}
}
        if (flags & canMSGERR_OVERRUN) Overrun = TRUE;
        MessageCount++;
    }
} while (stat == canOK);
Sleep(50);
}
switch (getch()) {
    case 3:
    case 27:
        Ready = TRUE;
        break;
    case 'q':
        if (Quiet > 0) Quiet--;
        fprintf(stderr, "Quiet=%d\n", Quiet);
        break;
    case 'Q':
        if (Quiet < 5) Quiet++;
        fprintf(stderr, "Quiet=%d\n", Quiet);
        break;
case 'p':
        deger=stepim;
        break;
case 'n':
        deger=-1*stepim;
        break;
case 's':
        deger=0;
        break;

    }
}
if (Quiet == 0) {
    time_t t;
    time(&t);
    fprintf(f, "; Logging ended at %s", ctime(&t));
    fprintf(f, "; %lu messages.%s\n",
        MessageCount,
        Overrun? " There were overruns." : "");
}

fclose(f);
(void)canBusOff(hnd);
(void)canClose(hnd);
}

```


EK C. : PLC

```
PROGRAM _CYCLIC
LinmotHomeDone
  if HomeLinBR = 1 then
    inc Say
    if Say = 10 then
      LinmotDrive5 = 2049
      MotorKontrol.Command.Power=1
    endif
    if Say = 100 then
      MotorKontrol.Command.Home=1
      MotorKontrol.Parameter.HomePosition= 0
      MotorKontrol.Parameter.HomeMode = mcHOME_DIRECT
    endif
    if Say= 150 then
      MotorKontrol.Command.Home=0
      LinmotDrive5 = 1
      Say = 0
      HomeLinBR = 0
    endif
  endif

FRM_read_01.enable = 1
FRM_read_01.ident = frm_ident
FRM_read_01 FUB FRM_read()
read_buffer = FRM_read_01.buffer
read_buffer_length = FRM_read_01.buflng
status_read = FRM_read_01.status

IF (status_read = 0) THEN
  memcpy(ADR(read_data), read_buffer, read_buffer_length)
  tursay=0
  lidargeldi=1
  FRM_rbuf_01.enable = 1
  FRM_rbuf_01.ident = frm_ident
  FRM_rbuf_01.buffer = read_buffer
  FRM_rbuf_01.buflng = read_buffer_length
  FRM_rbuf_01 FUB FRM_rbuf()
  status_rbuf = FRM_rbuf_01.status
  IF (status_rbuf <> 0) THEN
    error_frm = 6
  ENDIF
ELSE
  error_frm = 5
ENDIF

if (buton4=1) then
```

```

LinmotDrive5 = 1
LinmotDrivew10 = 400
cobidSay = 1
durum=0
gazsinyal=gazorani1
endif

if ((buton1=1) and (tursay<6)) then
LinmotDrivew10 = 10
cobidSay = 1
durum=1
gazsinyal=gazorani1
kart=1
pedal=0
gazsinyal=gazorani2
endif

if ((buton3=1) and (bayrak=0)) then
HomeLinBR = 1
endif

if (buton2=1) then
MotorKontrol.Command.Power=0
durum=0
LinmotDrive5 = 0
gazsinyal=gazorani1
pedal=1
kart=0
bayrak=0
basla = 0
HomeLinBR = 0
Say = 0
endif
if (tursay>5) then
if(durum=1) then
durum=0 ; lidardan data gelmiyor
gazsinyal=gazorani1
pedal=1
kart=0
bayrak=0
LinmotDrivew10 = 600
MotorKontrol.Command.Power=0
cobidSay = 1
endif
else
tursay=tursay+1
endif

if (lidargeldi=1) then
lidargeldi=0

```

```

r8bitlik =read_data[0]
sayacL= r8bitlik and 3
if (sayacL<>eskisayacL) then
eskisayacL=sayacL
else
sayacLdurumu=sayacLdurumu+1;çok paket kaçtı
endif

degerL = r8bitlik / 4
isaretL= r8bitlik and 128

if isaretL = 128 then
degerL = degerL + 192
endif

if (degerL<>31) then
AbsAdim = - 2500* degerL ; direksiyon açısı.
endif

if AbsAdim > 45000 then ; servo motor için limit ( + yön )
AbsAdim = 45000
endif

dintAbsAdim= -45000 ; servo motor için limit ( - yön )

if AbsAdim < dintAbsAdim then
AbsAdim = -45000
endif
endif ; iflidargeldi

if (durum=1) then
MotorKontrol.Parameter.Position = AbsAdim ; motor pozisyon bilgisi
MotorKontrol.Parameter.Velocity = AbsHiz ; motor hız bilgisi
MotorKontrol.Parameter.Acceleration = AbsivmeA ; Motor pozitif ( kalkma )
ivmelenme bilgisi
MotorKontrol.Parameter.Deceleration = AbsivmeD ; Motor negatif ( durma )
ivmelenme bilgisi
MotorKontrol.Command.MoveAbsolute=1 ;Absolute harekete izin verir.
if degerL = 31 then
gazsinyal=gazorani1
LinmotDrivew10 = 600 ; linmot fren ileri ( basma )
else
LinmotDrivew10 = 10 ; linmot fren geri ( çekme )
gazsinyal=gazorani2
endif
cobidSay = 1
endif

LinmotEndPos = LinmotMoveDone
LinmotOkuma5

```

```
LinmotOkuma6
LinmotOkuma7
LinmotOkuma8
LinmotOkuma9
LinmotDrive1
LinmotDrive2
LinmotDrive3
LinmotDrive4
LinmotDrive6= LinmotDrivew6
LinmotDrive7= LinmotDrivew7
LinmotDrive8= LinmotDrivew8
LinmotDrive9= cobidcount; cob id count  LinmotDrivew9
LinmotDrive10= LinmotDrivew10; target position
LinmotDrive11= LinmotDrivew11 ; speed
LinmotDrive12= LinmotDrivew12 ; acceleretion
LinmotDrive13= cobidcount ; com id count = LinmotDrivew13;
LinmotDrive14= LinmotDrivew14; deccelaretion
LinmotDrive15= LinmotDrivew15; force parametre
LinmotDrive16= LinmotDrivew16
```

```
if cobidSay = 1 then ; .
cobidcountArti = cobidcountArti + 1
cobidcount = cobidcountArti +2305
cobidSay = 0
if cobidcount > 2313 then
cobidcount= 2305
cobidcountArti = 0
endif
endif
```

```
END_PROGRAM
```

ÖZGEÇMİŞ

Zeynel KOÇ, 24 Kasım 1982’de İstanbul’da doğdu. İlköğretimini Yunus Emre İlköğretim Okulu’nda, ortaöğretimini ise Zeytinburnu Anadolu Teknik Lisesi Otomatik Kumanda Bölüm’ünde birincilikle tamamladı. Ardından 2002 yılında Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Haberleşme Öğretmenliği bölümünde tamamladı. Halen İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü’ne bağlı Mekatronik Mühendisliği yüksek lisans programına devam etmektedir.