

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**OTONOM BİR OTOMOBİL İÇİN HIZ KONTROLÖRÜ TASARIMI
VE
UYGULAMASI**

YÜKSEK LİSANS TEZİ

Şafak GÜNER

Makina Mühendisliği Anabilim Dalı

Sistem Dinamiği Ve Kontrol Programı

Tez Danışmanı: Prof. Dr. Ata MUĞAN

MAYIS 2012

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**OTONOM BİR OTOMOBİL İÇİN HIZ KONTROLÖRÜ TASARIMI
VE
UYGULAMASI**

YÜKSEK LİSANS TEZİ

**Şafak GÜNER
503081623**

Makina Mühendisliği Anabilim Dalı

Sistem Dinamiği Ve Kontrol Programı

Tez Danışmanı: Prof. Dr. Ata MUĞAN

HAZİRAN 2012

İTÜ, Fen Bilimleri Enstitüsü'nün 503081623 numaralı Yüksek Lisans Öğrencisi **Şafak GÜNER**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**OTONOM BİR OTOMOBİL İÇİN HIZ KONTROLÖRÜ TASARIMI VE UYGULAMASI**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Ata MUĞAN**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Nejat TUNCAY**
Okan Üniversitesi

Yrd. Doç. Dr. Pınar BOYRAZ
İstanbul Teknik Üniversitesi

Teslim Tarihi : **4 Mayıs 2012**
Savunma Tarihi : **8 Haziran 2012**

ÖNSÖZ

Bu çalışmanın konusu “Otonom Bir Otomobil İçin Hız Kontrolörü Tasarımı ve Uygulaması” dır. Tez çalışmam boyunca yönlendirici katkısı ve yardımları ile bana yol gösteren tez danışmanım Prof. Dr. Ata MUĞAN’a, pozitif enerjisi ve yapıcı eleştirileriyle bilgi birikimini gençlerle paylaşmaktan çekinmeyen değerli hocalarım Prof. Dr. Nejat TUNCAY’a ve Prof Dr. Tefvik AKGÜN’e, Prof. Dr. Ümit ÖZGÜNER’e, eğitim hayatım boyunca bana destek veren başta annem olmak üzere tüm aileme ve tezimi hazırlamamda yaptığı yardımlardan ve gösterdiği sabırdan ötürü arkadaşım Zeynel KOÇ’A teşekkürü borç bilirim.

Mayıs 2012

Şafak GÜNER

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER	vii
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xix
SUMMARY	xxi
1. GİRİŞ	1
1.1 Otonom Araçlar	4
1.2 Otonom Araçlarla İlgili Yapılan Çalışmalar	7
1.3 Hız Sabitleyicisi (Cruise Control)	8
1.4 Tasarlanan Otonom Aracın Yapısı ve Hız Kontrolör.....	9
2. ARAÇTAKİ HABERLEŞME SİSTEMİ ve BU SİSTEME ERİŞİM	13
2.1 Paralel-Seri Haberleşme	13
2.2 CAN (Controller Area Network).....	15
2.2.1 Ağın fiziksel yapısı	15
2.2.2 CAN protokolü.....	17
2.2.3 Mesaj türleri	18
2.2.3.1 Amaçlarına göre CAN mesaj çeşitleri.....	18
Veri mesajı	18
İstek mesajı.....	18
Hata mesajı.....	19
Aşırı Yük Mesajı	19
2.2.3.2 Yayınlanma anlarına göre CAN mesajları	19
Olay mesajı.....	19
Periyodik mesaj	19
Periyodik olay mesajı.....	19
İstek tabanlı olay mesajı.....	19
Çoklu mesaj.....	19
2.2.3.3 ID uzunluğuna göre CAN mesajları.....	19
Standart format.....	19
Genişletilmiş format.....	20
2.2.4 Mesaj yapısı	20
2.2.4.1 Veri mesajı	20
Standart formatta bir veri mesajı yapısı	20
Genişletilmiş formatta bir veri mesajının yapısı	24
2.2.4.2 İstek mesajı.....	25
2.2.4.3 Hata esajı.....	25
2.2.5 CAN protokolünün avantajları	26
2.3 Araç Üzerindeki CAN Yapısı	27
2.3.1 B-CAN	27

2.3.2 C-CAN	29
2.3.3 CAN hattına erişim.....	32
3. OTONOM OTOMOBİLİN GAZ ve FREN SİSTEMİNİN İNCELENMESİ	35
3.1 İvmelenme	35
3.1.1 Gaz pedalı.....	37
3.1.2 Motor sistemi.....	39
3.1.2.1 ECU (Engine control unit)	39
3.1.2.2 Enjeksiyon sistemi.....	40
3.1.2.3 Dizel motor.....	41
3.1.3 Otomatik şanzıman.....	42
3.1.3.1 Tork dönüştürücü	42
3.1.3.2 Planet dişli grubu.....	44
3.1.3.3 TCU (Transmission control unit)	45
3.1.3.4 Diferansiyel	45
3.1.4 Tekerlekler ve şasi.....	46
3.2 İvmelenme Sistemine Sinyal Uygulama	46
3.3 Frenleme	47
3.3.1 Fren pedalı, merkez silindir ve hidrolik sıvı tankı	49
3.3.2 Hidrolik modülatör, pistonlar, diskler ve kampanalar.....	51
3.3.3 BSU(Brake System Unit) ve ABS(Anti-Lock Break System).....	52
3.4 Fren Sistemine Sinyal Uygulama	55
3.4.1 Lineer motor sürücüsü.....	56
3.4.2 Lineer motor.....	58
3.5 Hız Kontrolörü İçin Tasarlanan Sistem	59
3.5.1 Bilgisayar birimi.....	60
3.5.2 Elektronik kontrol birimi.....	63
3.5.2.1 Besleme	67
3.5.2.2 Merkezi birim (1)	68
Butonlar, göstergeler ve sesli uyarı elmanı	70
Seçici Devre	72
Analog girişler.....	74
PIC16F877'nin programlanması	76
3.5.2.3 Merkezi birim (2)	77
3.5.2.4 Analog devre	78
3.5.2.5 Ara yüzler.....	79
3.5.3 Yapılan deneyler	80
4. BOYUNA ARAÇ MODELİNİN ÇIKARTILMASI ve MATLAB SİMÜLASYONLARI	83
4.1 Boyuna Araç Modeli	83
4.2 Sistemin Matlab Simülasyonu Modeli	94
4.2.1 Sürücü.....	94
4.2.2 Dizel motor.....	94
4.2.3 Dişli sistem.....	98
4.2.4 Fren sistemi	100
4.2.5 Tekerlekler	104
4.2.6 Araç	106
4.2.7 Yol Eğimi	106
4.2.8 Rüzgâr direnci	107
4.3 Simülasyon Sonuçları.....	108
4.3.1 Aracın tam gaz ile hızlanmasının simüle edilmesi.....	108

4.3.2 Eğiminin farklı olduğu yollarda aracın simüle edilmesi	114
4.3.3 Aracın ayrı iki gaz oranı girişiyle hızlanması	116
4.3.4 Aracın hızlanma ve frenleme simülasyonu	119
4.3.5 Farklı gaz ve fren giriş değerleri simülasyonu	124
5. KONTROLÖR TASARIMI VE DENEY SONUÇLARI	127
5.1 Sisteme Ait Transfer Fonksiyonları	127
5.1.1 İvmelenme sistemine ait transfer fonksiyonunun elde edilmesi	127
5.1.2 Fren sistemine ait transfer fonksiyonunun elde edilmesi	131
5.2 Kontrolör Yapısı ve Katsayıların Bulunması	132
5.2.1 İvmelenme sistemi için kontrolör katsayıları	132
5.2.2 Fren sistemi için kontrolör katsayıları	136
5.2.3 Genel yapının MATLAB ortamında simüle edilmesi	139
5.2.4 Bilineer dönüşümler	143
5.3 Deneysel Sonuçlar ve Kontrolörlere İnce Ayar Yapılması	144
5.3.1 İvmelenme deneyleri ve iyileştirilmiş katsayılar	145
5.3.2 Frenleme deneyleri ve iyileştirilmiş katsayılar	147
5.3.3 Hız kontrolörü deneyleri	148
5.4 Deneysel Sonuçların Simülasyon Modeliyle Karşılaştırılması	150
6. SONUÇ	153
KAYNAKLAR	157
EKLER	159
ÖZGEÇMİŞ	189

KISALTMALAR

ABS	: Anti-Lock Brake system
ACK	: Acknowledge Field
ADC	: Analog Digital Converter
AGV	: Autonomous Ground Vehicle
BCU	: Body Control Unit
BSU	: Brake System Control Unit
CAN	: Controller Are Network
CRC	: Cyclic Redundancy Code
CTU	: Convetrgence Telematic Unit
DAC	: Digital Analog Converter
DARPA	: Defense Advanced Research Projects Agency
DLC	: Data Length Code
ECC	: Electronic Climate Control
ECU	: Engine Control Unit
EEROM	: Electronically Erasable Programmable Read-Only Memory
EOF	: End of Frame
EPROM	: Erasable Programmable Read Only Memory
GND	: Ground
ID	: Identifier
IDE	: Identifier Extension
IPC	: Instrument Panel Cluster
Kb	: Kilobit
Kbps	: Kilobit per second
ECC	: Electronic Climate Control
km/h	: kilometre/saat
LCD	: Liquid Crystal Display
LSB	: Least Significant Bit
m/s	: metre/saniye
Mb	: Megabit
Mbps	: Megabit per second
MCLR	: Memory Clear
MHz	: Mega Hertz
MSB	: Most significant Bit
OBD	: On-Board Diagnostic
OP-Amp	: Operational Amplificator
PAU	: Park Assistant Unit
PIC	: Programmable Interface Controller
RAM	: Random Access Memory
REC	: Receiver Error Counter
ROM	: Read-Only Memory
RRU	: Radio Receiver Control Unit
RTR	: Remote Transfer Request
RX	: Receive Data

SAS	: Steering Angle Control Unit
SDU	: Sensing and Diagnostic Unit
SOF	: Start of Frame
SRR	: Substitute Remote Request
TCU	: Transmission Control Unit
TEC	: Transmit Error Counter
TX	: Transmite Data
USB	: Universal Serial Bus
VIAC	: Vislab Intercontinental Autonomous
YRS	: Yaw Rate Control Unit

ÇİZELGE LİSTESİ

Sayfa

Çizelge 1.1 : Ülkelere göre üretilen araç sayısı [1].	1
Çizelge 1.2 : Trafik kazası sebepleri [2].	3
Çizelge 1.3 : 2008 yılı CO ₂ miktarı [3].	3
Çizelge 2.1 : CAN ve diğer haberleşme protokollerinin sınıflandırılması [13].	14
Çizelge 2.2 : Mesaj sınıf biti [13].	21
Çizelge 2.3 : Mesaj öncelik bitleri [13].	21
Çizelge 2.4 : Node numarası.	22
Çizelge 2.5 : Veri uzunluk kodu tablosu [17].	23
Çizelge 2.6 : Modül çalışma durumu.	25
Çizelge 2.7 : B-CAN modülleri.	27
Çizelge 2.8 : C-CAN modülleri.	29
Çizelge 3.1 : Linmot B1100 sürücü özellikleri.	58
Çizelge 3.2 : PIC16F877 genel özellikleri.	69
Çizelge 3.3 : Mikro kontrolör pin bağlantıları.	70
Çizelge 3.4 : Gösterge lambaları durumları.(X:1, X:0)	72
Çizelge 3.5 : Kontak durumları (K: Kontak iletimde, A: Kontak kesimde).	73
Çizelge 3.6 : Merkezi birim 2'nin özellikleri.	77
Çizelge 4.1 : Vites çevrim oranları.	87

ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : 2004 yılı trafik kazaları sayısı [2].....	2
Şekil 1.2 : 2004 yılı trafik kazalarında hayatını kaybeden insan sayısı [2].	2
Şekil 1.3 : Otonom araç genel sistemi	6
Şekil 1.4 : Tasarlanan sistemin genel şeması	9
Şekil 2.1 : Seri ve paralel haberleşme [11].	14
Şekil 2.2 : CAN bağlantı yapısı [11].	15
Şekil 2.3 : CAN hattı uzunluğuna göre iletişim hızı [13].	16
Şekil 2.4 : Can-High ve Can-Low gerilim değerleri [16].	16
Şekil 2.5 : Mesaj önceliği.	18
Şekil 2.6 : Standart formatlı veri mesajı [17].	20
Şekil 2.7 : 11 bitlik ID yapısı [13].	21
Şekil 2.8 : DLC koduna ait bit yerleşimi [17].	22
Şekil 2.9 : Genişletilmiş veri mesajı yapısı [19].	24
Şekil 2.10 : İstek mesajı yapısı [17].	25
Şekil 2.11 : Hata mesajı yapısı [17].	26
Şekil 2.12 : B-CAN hattının bağlantı şeması.....	28
Şekil 2.13 : C-CAN hattının bağlantı şeması.....	29
Şekil 2.14 : Mesaj içindeki örnek veri.	31
Şekil 2.15 : CAN için oluşturulan bağlantı yapısı.	32
Şekil 2.16 : Akış diyagramı.	33
Şekil 2.17 : Araç hızı.	34
Şekil 2.18 : Vites.	34
Şekil 2.19 : Motor devri.....	34
Şekil 3.1 : İvmelenme sistemi genel yapısı.	36
Şekil 3.2 : Gaz pedalı ve iç yapısı.....	37
Şekil 3.3 : Gaz pedalının elektronik yapısı.....	38
Şekil 3.4 : Gaz pedalı çıkış sinyalleri değişimi.....	38
Şekil 3.5 : Enjeksiyon sistemi.....	41
Şekil 3.6 : Dizel motor.....	41
Şekil 3.7 : Türbin-stator-pervane[23].	42
Şekil 3.8 : Tork dönüştürücü[23].	43
Şekil 3.9 : Planet dişli grubu [23].	45
Şekil 3.10 : Frenleme sistemi genel yapısı.	48
Şekil 3.11 : Fren pedalı [26]	49
Şekil 3.12 : Ana silindir ve yağ tankı [27].	50
Şekil 3.13 : Yağ haznesi.	51
Şekil 3.14 : Tekerlek üzerindeki kuvvetler ve uygulama noktaları.	53
Şekil 3.15 : Araç ağırlık merkezinin dingillere olan uzaklığı.....	53
Şekil 3.16 : Kuvvet bağlantı katsayısı-kuvvet ilişkisi [28].	54
Şekil 3.17 : Lineer motorun mekanik bağlantısı.....	56
Şekil 3.18 : Fren bağlantı aparatı.	56

Şekil 3.19 : Lineer motor	57
Şekil 3.20 : Linmot-Talk yazılımı.....	57
Şekil 3.21 : Lineer motor sürücüsü.	58
Şekil 3.22 : Lineer motor.	59
Şekil 3.23 : Hız kontrolörü genel blok diyagramı	59
Şekil 3.24 : Bilgisayar ortamındaki kontrol algoritmasının akış diyagramı.	61
Şekil 3.25 : Test algoritması.	62
Şekil 3.26 : Elektronik kontrol biriminin blok diyagramı.	64
Şekil 3.27 : Elektronik kart devre şeması.	65
Şekil 3.28 : Elektronik kart.	66
Şekil 3.29 : Besleme.	67
Şekil 3.30 : PIC16F877 ve pin diyagramı.....	68
Şekil 3.31 : Kullanıcı ara yüz elemanlarının PIC'e bağlantısı.	71
Şekil 3.32 : 4066 pin diyagramı [31].	72
Şekil 3.33 : Seçicilerin kontak bağlantıları.....	73
Şekil 3.34 : Besleme girişi (sınır değer kontrolü).	74
Şekil 3.35 : ECU'ye giden sinyaller (sınır değer kontrolü).	75
Şekil 3.36 : ECU'ye giden sinyaller (oransal kontrol).....	75
Şekil 3.37 : Fren motoru sürücüsüne giden sinyal (sınır değer kontrolü).	75
Şekil 3.38 : PIC16F877'ye yüklenen kodun akış diyagramı.	76
Şekil 3.39 : Merkezi birim 2 [33].	77
Şekil 3.40 : STM32VLDISCOVERY kart yazılımının akış diyagramı.....	78
Şekil 3.41 : Analog devre şeması.....	79
Şekil 3.42 : MSP430.	80
Şekil 3.43 : Gaz pedalı basış oranı deney 1.	80
Şekil 3.44 : Gaz pedalı basış oranı deney 2.	81
Şekil 4.1 : Boyuna araç modelinin genel blok diyagramı[4].	83
Şekil 4.2 : Motor haritası.	84
Şekil 4.3 : Güç aktarma organları.	86
Şekil 4.4 : Fren diski [34].	88
Şekil 4.5 : Yokuş direnci [34].	92
Şekil 4.6 : Sürücü sinyalleri.	94
Şekil 4.7 : Sistemin ana blok diyagramı.	96
Şekil 4.8 : Dizel motor.	97
Şekil 4.9 : Dişli sistem.	98
Şekil 4.10 : Vites kontrol.	98
Şekil 4.11 : Vites dişli kutusu.	99
Şekil 4.12 : Diferansiyel dişlisi.....	100
Şekil 4.13 : Fren sistemi.	103
Şekil 4.14 : İvmelenme direnci.	104
Şekil 4.15 : Tekerlekler.....	105
Şekil 4.16 : Araç.	106
Şekil 4.17 : Yol eğimi(a).....	107
Şekil 4.18 : Yol eğimi(b).	107
Şekil 4.19 : Rüzgâr direnci.	107
Şekil 4.20 : Araç hızı.	108
Şekil 4.21 : Araç ivmesi.....	109
Şekil 4.22 : Türbin devri.....	109
Şekil 4.23 : Açısal hız oranı.değişimi.	110
Şekil 4.24 : Vites değişimleri.....	110

Şekil 4.25 : Vites kutusuna gelen tork.	111
Şekil 4.26 : Diferansiyel çıkış torku.	111
Şekil 4.27 : Araç hızına bağlı yuvarlanma direnci.	112
Şekil 4.28 : Rüzgâr direnci.	113
Şekil 4.29 : İvmelenme direnci.	113
Şekil 4.30 : 3 farklı eğim için hız grafiği.	114
Şekil 4.31 : Maksimum hız noktaları.	115
Şekil 4.32 : 3 farklı eğim için aracın aldığı yol.	115
Şekil 4.33 : 3 farklı eğim için vites değişim zamanları.	116
Şekil 4.34 : Araç hızı.	117
Şekil 4.35 : Vites değişimleri.	117
Şekil 4.36 : Motor devri.	118
Şekil 4.37 : Araç ivmesi.	118
Şekil 4.38 : Uygulanan gaz ve fren sinyalleri.	119
Şekil 4.39 : Araç hızı.	119
Şekil 4.40 : Türbin devri-vites durumu.	120
Şekil 4.41 : Araç ivmesi.	121
Şekil 4.42 : Alınan yol.	121
Şekil 4.43 : Vites kutusu girişi.	122
Şekil 4.44 : İvmelenme direnci.	122
Şekil 4.45 : Toplam fren kuvveti.	123
Şekil 4.46 : Aç kapa sinyali.	123
Şekil 4.47 : Uygulanan giriş.	124
Şekil 4.48 : Araç hız değişimi.	124
Şekil 4.49 : Araç ivme değişimi.	125
Şekil 4.50 : Alınan yol.	125
Şekil 5.1 : System Identification Toolbox	128
Şekil 5.2 : Verilerin SI Toolbox'a aktarılması.	128
Şekil 5.3 : Araç hızı-gaz pedalı basış oranı grafiği.	129
Şekil 5.4 : Tahmini transfer fonksiyonu için parametre belirleme.	130
Şekil 5.5 : Tahmini çıkış-ölçülen çıkış karşılaştırılması (Benzerlik %91.27).	130
Şekil 5.6 : Araç hızı-fren pedalı basış oranı grafiği.	131
Şekil 5.7 : Tahmini çıkış-ölçülen çıkış karşılaştırılması (Benzerlik %83.44).	132
Şekil 5.8 : İvmelenme sistemi için açık çevrim Root-Locus analizi.	132
Şekil 5.9 : Kontrolörlü Root-Locus analizi.	133
Şekil 5.10 : İvmelenme sistemi MATLAB simülasyonu (1).	133
Şekil 5.11 : İvmelenme sistemi çıkışı.	134
Şekil 5.12 : PID Tuner.	134
Şekil 5.13 : İyileştirilmiş PID katsayıları.	135
Şekil 5.14 : İvmelenme sistemi MATLAB simülasyonu (2).	135
Şekil 5.15 : İyileştirilmiş ivmelenme sistemi çıkışı.	136
Şekil 5.16 : İyileştirilmiş kontrolörlü Root-Locus analizi.	136
Şekil 5.17 : Fren sistemi için açık çevrim Root-Locus analizi.	137
Şekil 5.18 : Fren sistemi için kontrolörlü Root-Locus analizi.	137
Şekil 5.19 : Fren sistemi MATLAB simülasyonu.	138
Şekil 5.20 : Fren sistemi için iyileştirilmiş kontrolörlü Root-Locus analizi(1).	138
Şekil 5.21 : Fren sistemi için iyileştirilmiş kontrolörlü Root-Locus analizi(2).	138
Şekil 5.22 : İyileştirilmiş fren sistemi çıkışı.	139
Şekil 5.23 : Hız kontrolörün Simulink modeli.	140
Şekil 5.24 : Karar verme merkezi akış diyagramı.	141

Şekil 5.25 : Fren ve ivmelenme sisteminin referans hız takibi.	142
Şekil 5.26 : Araç hızının referans değeri takibi.	142
Şekil 5.27 : Kontrolör katsayıları $P=3.161$ $I=0.81955$ $D=1.3611$ $N=3.0518$	145
Şekil 5.28 : $P=2.4828$ $I=1$ $D=1.9038$ $N=2.0479$	146
Şekil 5.29 : $P=3$ $I=0.6$ $D=2.04$ $N=1.5$	146
Şekil 5.30 : $P=16.6$, $I=8.4$, $D=1.73$ ve $N=1.64$	147
Şekil 5.31 : $P=18$ $I=7.3$ $D=1$ $N=2$	147
Şekil 5.32 : Hız kontrolör uzun mesafe deneyi	148
Şekil 5.33 : Hız kontrolör uzun mesafe deneyi (16 s'lik kesit)	149
Şekil 5.34 : Farklı referans değerleriyle hız kontrolörü deneyi.	149
Şekil 5.35 : Hız kontrolörünün araç modeliyle birleştirilmesi	151
Şekil 5.36 : Farklı referans değerleriyle hız kontrolörü simülasyonu	151

OTONOM BİR OTOMOBİL İÇİN HIZ KONTROLÖRÜ TASARIMI VE UYGULAMASI

ÖZET

Bilginin insanlık açısından zengin, kolay erişilebilir ve kullanılabilir olduğu bu çağda, teknolojik birikimin artmasıyla ürün çeşitliliğinin bollaştığı, beraberinde üretimin ve tüketimin zirve yaptığı yadsınamaz bir gerçektir. Bunun sonucunda birçok sektör için ürün bazında çeşitlilik neredeyse limit değerlere ulaştığından, üretici firmaların çoğunluğu pazar paylarını küçültmemek veya satışlarını arttırmak için yeni buluşlar çıkarmak yerine mevcut olanı geliştirme yolunu tercih etmiştir. Daha kaliteli ve nitelikli ürünlerle tanışan tüketici bunu bir alışkanlık haline getirmiş, sanayi de var olan bu talebe daha fazla eğilmiştir. Böylelikle üretici-tüketici arasındaki ticari alış-veriş konfor, kalite vb. özellikler çerçevesinde kısır bir döngüye girmiştir.

Sürekli değişikliğin ve ilerlemenin yaşandığı platformlardan biri de rekabetin can alıcı olduğu otomotiv sektörüdür. Önceleri hız ve güç öncelikli tercihlerin günden güne konfor ve güvenliğe doğru kaymasıyla, üreticiler de Ar-Ge kaynaklarını zamanla bu yöndeki çalışmalara aktarmaya başlamıştır. Otomatik vites, ABS, hidrolik direksiyon, ESP, hava yastığı gibi yarı otonom olarak adlandırılacak yapılar bu çalışmaların ilk aşamalarında çıkmış sonuçlar olarak sayılabilir. Yakın zamanda ise araçlara acil durumda kendiliğinden fren yapma, hız sabitleme, kendi kendine park etme, sürücüyü uyuduğunda uyarma gibi özellikler eklenerek yarı otonom sistemlerdeki gelişim devam etmiştir. Bu tarz sistemler evrimlerini tamamladıklarında bir sonraki safha tam otonom sistemlere, yani otonom araçlara geçiş olacaktır.

Otonom araçlar, bulunduğu noktadan hedef olarak girilen koordinatlara herhangi bir insan müdahalesi olmaksızın yönüne ve hızına kendisi karar vererek giden sistemlerdir. Bu tez çalışmasında, O³ projesi(Okan Otonom Otomobil) kapsamında prototipi geliştirilen otonom araca yönelik bir hız kontrolörü tasarlanmış ve tasarlanan bu yapı hem simulasyon ortamında hem de yol testlerindeki uygulamalarla sınanmıştır. Hız kontrolörün otonom araçtaki temel rolü elektronik bir sinyal olarak verilen referans hızda yolun eğimine ve diğer bozucu etkenlere bakmaksızın aracın gitmesini sağlamaktır.

Hız kontrolörün temelini oluşturan bu tez altı başlık altında hazırlanmıştır. Öncelikle otonom araçlara niçin gereksinim duyulduğu konusuna değinilmiş ve bu sistemlerin getireceği avantajlar ele alınmıştır. Daha sonra bugüne kadar yapılan benzer çalışmalardaki yapılar ilgili kavramlar ve aygıtlarla birlikte incelenmiştir. O³ projesi için planlanan otonom aracın genel yapısına ve bu sistem içinde yer alacak olan hız kontrolörünün önemine yine bu bölümde değinilmiştir.

İkinci bölümde ise otomobillerdeki alt sistemleri denetleyen elektronik modüllerin birbiri ile yüksek hızlarda minimum düzeyde hatayla haberleşmelerini sağlayan ve CAN olarak adlandırılan ağ topolojisi irdelenmiştir. Elde edilen bilgiler ışığında araç

üzerinde bulunan haberleşme hattına bağlanılmış, yapılan denemeler sonucunda sürüş esnasındaki anlık hız, ivme, gaz pedalı basış oranı, motor torku gibi bilgiler okunmuş ve anlaşılmıştır.

Bir sonraki bölümde otonom aracın boyuna kontrolü için ihtiyaç duyulan iki ana fonksiyon, ivmelenme sistemi ve frenleme sistemi, tüm detaylarıyla incelenmiştir. İnceleme sonucunda otomobilin hızlanmasını ve yavaşlamasını sağlayan bu sistemlere dışarıdan elektronik bir işaretle müdahale edebilmek için gerekli mekatronik düzenek ve elektronik devre tasarlanmış, oluşturulan yazılım ile test edilmiştir.

Otonom aracın motor, güç aktarma organları ve fren sistemini de içeren boyuna dinamik modeli dördüncü bölümde çıkarılmış, MatLAB programı da kullanılarak farklı referans hız girişleri için bozucu etkenlerle beraber simule edilmiştir.

Beşinci bölümde ise hiyerarşik bir yapıya sahip hız kontrolörün genel hatlarıyla tasarımı gerçekleştirilmiş ve alt kontrolörler belirlenmiştir. İvmelenme ve frenleme sistemini denetleyecek olan alt kontrolcülere ait katsayıların hesaplanması ve hem benzetim ortamında hem de yol testleri ile iyileştirme işlemleri de yine bu aşamadır. Yapılan hesaplamalar sonucunda hız kontrolörünü somutlaştırmak için tasarım, VisualStudio programı kullanılarak C dilinde bilgisayar ortamına taşınmış, üçüncü bölümdeki oluşturulan donanımla birlikte gerçek ortamda uygulamaya konmuştur.

Son olarak çalışılan konu tekrar gözden geçirilmiş, karşılaşılan zorluklar, hız kontrolörünün eksikleri, iyileştirilebilecek yanları ve gelecekte yapılabilecekler tartışma konusu edilerek, ele alınmıştır.

DESIGN AND IMPLEMENTATION OF SPEED CONTROLLER FOR AN AUTONOMOUS CAR

SUMMARY

In our era, when the information is rich, easily accessed, and utilized; it is an undeniable fact that the product diversity has been increased and the production and consumption have reached at the top with the technological accumulation. As a result, fabricant companies have preferred progressing the existing, instead of making innovations in order to augment the sales or not to decrease the market shares, as the product variety has almost raised a limiting value. The consumer, who met more qualified and more aggravated products, has been used to the situation, and the industry has inclined more to this demand. Consequently, commercial relations between the fabricant and consumer have fallen into a vicious cycle in accordance with comfort, quality, etc.

One of the platforms with continual changes and developments is the automobile sector, in which the competition is crucial. As the preferences have shifted from speed and power to comfort and security day by day, fabricants have started to transfer their research and development resources to the studies in that effect. Structures, which can be called “semi-autonomous” like automatic transmission, ABS, power steering, ESP, airbag can be counted as the first outputs of this kind of works. Recently, the progress in semi-autonomous systems have continued with adding vehicles the features like self-braking in emergency cases, fixing the speed, self-parking, warning the driver in case he sleeps. Once this sort of systems have completely evolved, the next step will be to skip to fully autonomous systems – to autonomous systems, in other words.

Autonomus vehicles are the systems which go to the target coordinates given in the start point without human intervention and by self deciding to speed. Autonomous vehicles perceives their environments with sensors on them. The data from the sensors are evaluated in a central unit and signals are sent to the controllers to activate the actuators.

Autonomous vehicles can be classified into three main classes, autonomous ground vehicles (AGV), autonomous air vehicles (AAV) and autonomous underwater vehicles (AUV). In this thesis focused on the autonomous ground vehicles. There are very important advantages of these vehicles. The most important advantage is ability to minimize accidents. If statistics are examined, it is clear that a large portion of the causes of accidents are human originated. Fatigue, insomnia, attention deficit, drunkenness are a few reasons of the human originated accidents which autonomous vehicles never have. It reduces the errors which causes from driver. In addition, autonomous vehicles can response in milliseconds for sudden situations unlike human and it helps to prevent the accidents. So if an emergency event happens autonomous system will take early precaution than human.

Another important advantage of these vehicles is fuel save and low CO₂ emission. Autonomous vehicles have a lot of sensors on them so with the help of these sensors they can see the road situations and optimize vehicle path. Balanced use of gas and brake also helps fuel efficiency. It is very important for nature.

The number of vehicles in traffic is increasing with each passing day. Road capacity cannot support more vehicles and it causes heavy traffic problem especially in big cities. It is a huge problem in modern age. Autonomous vehicles can use the roads effectively then human because they can go closer to the other vehicles. In a sudden situation they can reflect quickly. Because of these reasons, traffic congestion can be reduced.

Another big advantage of autonomous vehicles is it helps the people who cannot drive car. Especially disabled or elderly people have a lot of problems to go to somewhere. To travel from point A to point B without any human assistance and human help is a big comfort. Autonomous vehicles can also be used to avoid situations that are harmful to people (battlefields or disaster-zones, explosions and bomb disabling).

Considering these advantages an autonomous vehicle is conceived. In this thesis study, a speed controller has been designed for the automobile to which a prototype is developed as part of the O³ project (Okan Autonomous Automobile) and this structure has been tested both in simulation and with applications in road tests. The main role of the speed controller is to ensure the vehicle to go in the reference speed given as an electronic signal regardless of the slope and other vitiating factors. The speed controller consist of two main sub systems which are called brake system and acceleration system. If the road slope is positive then the acceleration system is activated. If the road slope is negative braking system is activated. Acceleration system is controlled with the help of an electronic card which is designed just for this purpose. Unlike acceleration system, braking system is controlled with a mechanical structure. A linear motor is put on the brake pedal and according to the control signal it moves forward or back.

This thesis which constructs the basis of the speed controller is prepared under six titles. First of all, the subject “why autonomous vehicle is needed” is mentioned and the advantages of this system is argued. Then, the structures in studies to date are analyzed with related concepts and devices. Again, the general frame of the autonomous vehicle planned for O³ project and the importance of the speed controller that will be included in this system is discussed here.

In the second section, the network topology called CAN, which provides high speed communication with minimum errors between electronic modules that control subsystems in automobiles is explicated. In the light of information obtained, a connection is made with the communication line on the vehicle, and a software is developed to get data like immediate speed during drive, acceleration, tread ratio of the gas pedal, motor torque are read and explained.

In the next section, the two main functions for longitudinal control, acceleration system and braking system are analyzed in detail. After the analysis, mechatronic mechanism and the electronic circuit needed for intervening to these systems externally with an electronic signal to control the acceleration and slowdown are prepared and tested with the software developed.

The longitudinal dynamic model of the autonomous vehicle which involves motor, power transferring organs and brake system is settled in the fourth section, and along with the MatLAB program, it is simulated with vitiating factors for different speed inputs.

In the fifth section, the design of the speed controller with an hierarchical structure is made broadly and the sub controllers are stated. Calculation of the coefficients of sub controllers which control acceleration and braking, and the redevelopment processing with both in the simulation medium and road tests are again in that part. Design for formalizing the speed controller is shifted to computer by Visual Studio program using C language and executed with the hardware composed in the third section.

Finally, the study is revised, and the difficulties come up, the defects of the speed controller, the points that could be improved and what it can be done in the future are discussed.

It is suspected that autonomous vehicles will improve safety on the road. In near future autonomous ground vehicles will be an important technology of our lives but now it is obvious that it needs to be developed.

1. GİRİŞ

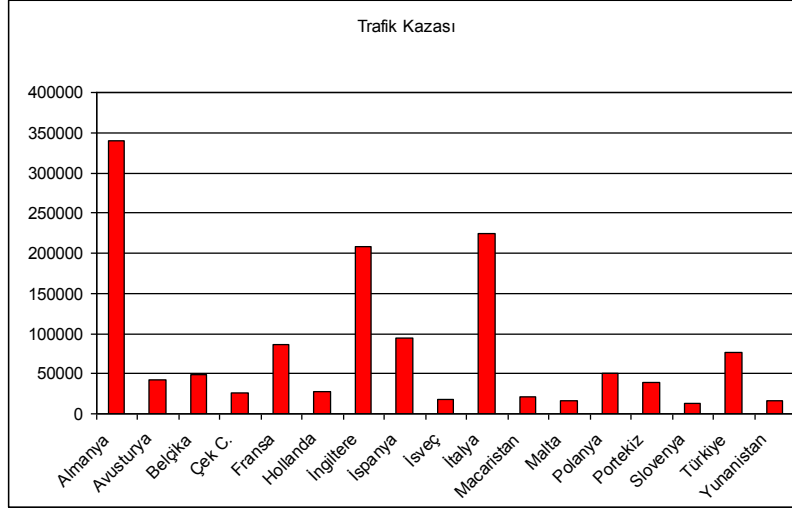
Gelişen teknolojiyle birlikte otomotiv sektörü büyük yatırımların yapıldığı, her geçen gün yeni inovasyonların ortaya çıktığı ve rekabetin gün geçtikçe arttığı bir sektör haline gelmiştir. Önceleri sadece fiyat rekabetinin yaşandığı bu sektörde artık fiyat tek parametre olmaktan çıkmıştır. Pazardan daha büyük bir pay almak isteyen üretici firmalar müşterilerini tatmin etmek için ar-ge'ye her yıl yüzlerce milyon dolarlık yatırım yapmakta ve sürekli yenilik peşinde koşmaktadırlar. Otomotiv sektörü yapısı itibariyle sanayinin diğer kollarıyla işbirliği içinde çalışan bir sektördür. Demir-çelik, plastik, petrokimya, lastik, tekstil, cam gibi sektörlerin önde gelen alıcısı konumundadır. Bu sebeple de üretim ve teknoloji sektörünün tetikleyicisi olarak görülmektedir.

Çizelge 1.1 : Ükelere göre üretilen araç sayısı[1].

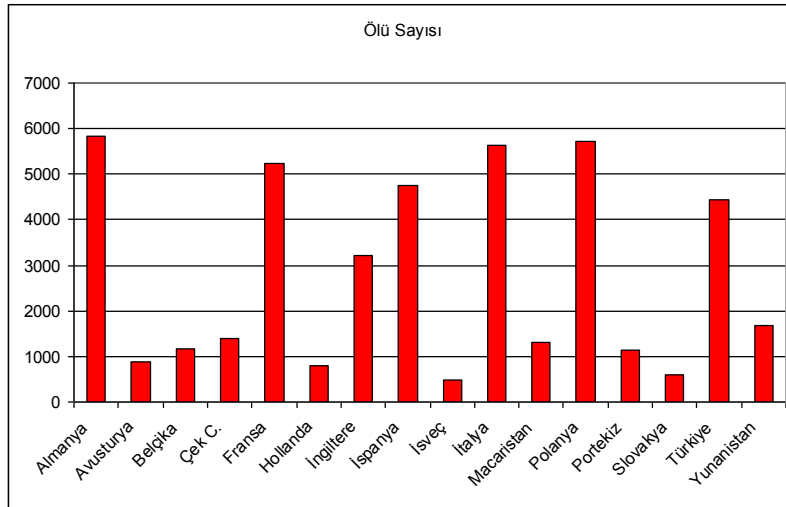
2010		2009		2008	
Çin	18.264.667	Çin	13.790.994	Japonya	11.575.644
Japonya	9.625.940	Japonya	7.934.057	Çin	9.299.180
USA	7.761.443	USA	5.731.397	USA	8.693.541
Almanya	5.905.985	Almanya	5.209.857	Almanya	6.045.730
G. Kore	4.271.941	G. Kore	3.512.926	G. Kore	3.826.682
Brezilya	3.648.358	Brezilya	3.182.923	Brezilya	3.215.976
Hindistan	3.536.783	Hindistan	2.641.550	Fransa	2.568.978
İspanya	2.387.900	İspanya	2.170.078	İspanya	2.541.644
Meksika	2.345.124	Fransa	2.047.693	Hindistan	2.332.328
Fransa	2.227.742	Meksika	1.561.052	Meksika	2.167.944
Kanada	2.071.026	Kanada	1.490.482	Kanada	2.082.241
Tayland	1.644.513	İran	1.394.075	Rusya	1.790.301
İran	1.599.454	İngiltere	1.090.139	İngiltere	1.649.515
Rusya	1.403.244	Tayland	999.378	Tayland	1.393.742
İngiltere	1.393.463	Çek C.	983.243	Türkiye	1.147.110
Türkiye	1.094.557	Polonya	878.998	İran	1.051.430
Çek C.	1.076.385	Türkiye	869.605	İtalya	1.023.774
Polonya	869.381	İtalya	843.239	Çek C.	946.567

İnsan hayatına verilen önemin artması, değişen konfor ve verimlilik anlayışı ile otomotiv sektöründe otonom ve yarı otonom sistemlere olan ilgi artmıştır. Özellikle trafik kazalarında yaşanan artışlar güvenlik ihtiyacını ön plana çıkarmıştır. Her yıl

tüm dünyada birçok trafik kazası meydana gelmekte ve bu kazalar sonucu binlerce insan hayatını kaybetmekte ya da yaralanmaktadır. Bu sebeple daha güvenli araçlar üretmek, araçları otonom hale getirmek büyük önem kazanmaktadır.



Şekil 1.1 : 2004 yılı trafik kazaları sayısı [2].



Şekil 1.2 : 2004 yılı trafik kazalarında hayatını kaybeden insan sayısı [2].

Meydana gelen trafik kazalarının sebepleri yol kusuru, yaya kusuru, araç kusuru gibi etkenler olsa da sürücü kaynaklı hataların kazalarda çok ciddi bir payı olduğu Çizelge 1.2 ‘de açık bir şekilde görülmektedir. Bu hataları minimuma indirebilmek için araçları daha “akıllı” yapabilmek gerekmektedir. Sürücünün alkollü ya da uykusuz olduğu anlarda, hava şartlarının kötü olduğu durumlarda, uzun yollarda otonom ya da yarı otonom sistemler çok büyük avantaj sağlamaktadır. Ayrıca ani olaylar karşısında sürücü tepki süresinin uzun olması, insan veri işleme kapasitesinin

sınırlı olması, konsantrasyon süresinin kısalığı gibi etkenler otonom sistemleri kaza önlemede daha da önemli kılmıştır.

Çizelge 1.2 : Trafik kazası sebepleri [2].

Yıl	Sürücü Kusuru %	Yaya Kusuru %	Yol kusuru %	Araç Kusuru %
1999	95,99	0,23	0,56	0,53
2000	96,06	0,25	0,77	0,52
2001	96,56	0,31	0,43	0,38
2002	96,82	0,23	0,25	0,31
2003	97,03	0,16	0,22	0,27
2004	97,30	0,11	0,19	0,22
2005	97,39	0,11	0,22	0,25
2006	98,07	0,09	0,13	0,10
2007	98,03	0,09	0,11	0,14
2008	90,53	0,43	0,42	0,26
2009	89,60	0,41	0,61	0,29
2010	88,70	0,39	0,69	0,36

Gelişen sanayileşme ile atmosfere salınan sera gazı oranları da hızla artmaktadır. Küresel ısınma ve iklim değişikliği gibi sorunlarının en önemli sebebi olan sera gazlarının büyük bir bölümü taşıtlardan kaynaklanmaktadır. Ülkeler tarafından üretici firmalara getirilen yasal bir takım zorunluluklara rağmen bu durumun önüne geçilememektedir. Çünkü trafiğe çıkan araç sayısı her geçen gün katlanarak artmaktadır. Otonom sistemlerde önemli derecede yakıt tasarrufu yapılmakta olup bu da hem harcanan enerjiyi hem de salınan sera gazını azaltmaktadır.

Çizelge 1.3 : 2008 yılı CO₂ miktarı [3].

Ülke	CO ₂ salınımı (ton)	Ülkelere göre dağılımı %
Çin	7.031.916	23.33%
ABD	5.461.014	18.11%
Hindistan	1.742.698	5.78%
Rusya	1.708.653	5.67%
Japonya	1.208.163	4.01%
Almanya	786,660	2.61%
Kanada	544,091	1.80%
İran	538,404	1.79%
İngiltere	522,856	1.73%
G. Kore	509,170	1.69%
Meksika	475,834	1.58%
İtalya	445,119	1.48%
G. Afrika	435,878	1.45%
S. Arabistan	433,557	1.44%
Endonezya	406,029	1.35%
Avustralya	399,219	1.32%
Brezilya	393,220	1.30%

Yukarıda bahsedilen sebepler göz önünde bulundurularak bu tez kapsamında otonom bir araç için hız kontrolörü tasarımı yapılmıştır. Ancak tasarıma geçmeden önce otomotiv sektöründe uygulamaları görülmeye başlanan yarı otonom sistemlere ve gelecekte birçok uygulamada görülmesi beklenen tam otonom sistemlere değinilmiş, daha sonra bu konu ile ilgili yapılan çalışmalar incelenmiştir. Ardından projede kullanılacak araç için tasarlanan yapı ele alınmış, bu yapıda yer alan her bloğun görevinden bahsedilmiştir. Bu araç için tasarlanan hız kontrolörünün önemi ve yapısı birinci bölümün son konusu olarak ele alınmıştır.

1.1 Otonom Araçlar

İnsan müdahalesi olmaksızın bulunduğu noktadan istenilen noktaya gidebilen yada daha önceden tanımlanmış bir görevi kendi karar mekanizmasıyla gerçekleştiren araçlardır [4]. Bu araçlar çevrelerini üzerlerinde bulunan farklı tipteki sensörler ile algılayıp, durum ve koşulları ortaya çıkartırlar. Bu durum ve koşullar karar verme mekanizması tarafından değerlendirilir ve doğru aktuatörleri çalıştırmak için kontrolörlere gerekli sinyaller yollar.

Otonom araçlar genel olarak kara, hava ve sualtı olmak üzere 3 ana katagoride sınıflandırılabilir. Bu tiplerin çalışma ortamları farklı olduğu kadar çalışma şekilleri de oldukça farklılık gösterir. Genel mantık benzer olsada kullanılan yazılım algoritmaları, sensör ve aktuator çeşitleri farklıdır.

Gelecekte en fazla görülmesi muhtemel tip olan otonom kara araçların(AGV-Autonomous Ground Vehicle) giderek bir gereksinim haline gelmektedir. Bu ihtiyacın en temel sebepleri:

- Trafik kazaların büyük bir kısmının sürücü hatası kaynaklı olması
- Araç sayısının gün geçtikçe artması ve yol kapasitesinin artışının bu hıza yetişememesi
- Fosil yakıtların giderek tükenmesi
- İnsanların gün geçtikçe daha fazla sürüş konforu ve güvenliği talep etmesi

olarak gösterilebilir.

Teknolojinin ilerlemesiyle algılayıcılar da daha hızlı ve daha hassas hale gelmektedir. Aynı şekilde aktuatörler de güçlü ve seri hale gelmekte, kontrol edilebilirlikleri mükemmelleşmektedir. Şuan var olan ve her saniye gelişen

teknolojik alt yapının aracılığıyla, otonom kara araçlar, sistemleri iyi tasarlandıklarında ve doğru programlandıklarında insanların erişemeyeceği bir sürüş güvenliğine ve refleksine sahip olacaklardır. Makinelerin uyku, stres, yorgunluk, alkol, sinir, dalgınlık vb. insana özgü özelliklere sahip olmaması birçok kazanın önüne geçilmesini sağlayacaktır. Yine insanların yaptığı “kurallara uymama” hatasını asla yapmayacak olması kaza sayısının azalmasında etkin olacak bir başka özelliktir.

Bu araçların bir başka getiri sağlayan özelliği ise üzerlerinde bulunan sistemlerin ve karar verme mekanizmalarının çok hızlı olması olmasıdır. Bu yetenek sayesinde araç ani gelişen durumlar karşısında en kısa sürede birçok olasılığı hesaplar, en doğru seçeneği seçer ve uygulamaya koyarak hızlı bir şekilde gerçekleştirir. Belirtilenleri yapmak otonom kara aracı için 100 ms’yi bulmazken, bu süre bir insan için sadece algılama süresidir. Üstelik insanın bu süre zarfında vereceği tepkinin ne kadar doğru olacağı bir başka tartışma konusudur.

Otonom araçların tepki sürelerinin çok kısa olması kendisine birçok avantaj kazandırır. Bunlardan ilki beklenmeyen durumlarda hızlı ve doğru tepki vererek kaza riskini azaltmaktır.

Diğer bir getiri yakıt tasarrufu olacaktır. Tepki süresinin çok hızlı olması öndeki araçla aradaki mesafenin daha kısa olabilmesi ihtimalini getirir. Böylece yüksek hızlarda çok etkin olan rüzgâr kuvveti ve ona harcanan enerji azaltılmış olur. Ayrıca araçlar arasındaki mesafenin daha az olması demek aynı uzunluktaki yol için daha fazla araç kapasitesi anlamına da gelmektedir.

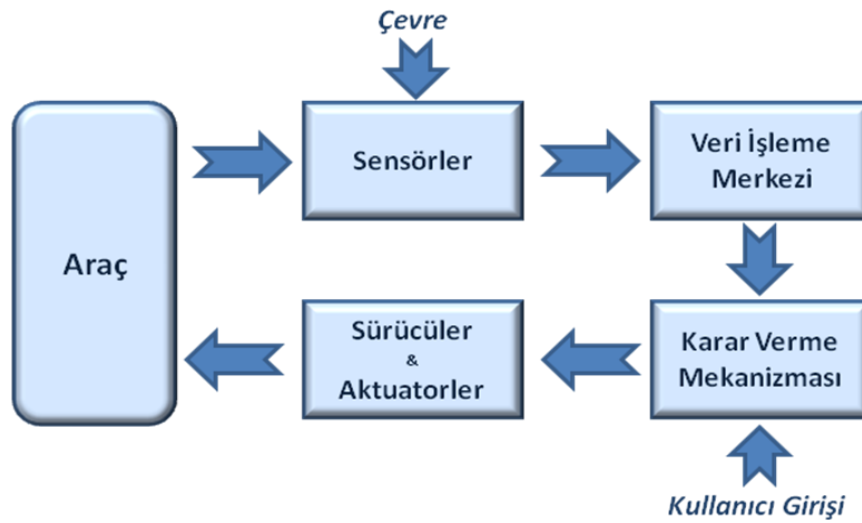
Belli bir mesafe için harcanan yakıt aynı aracı kullansalar bile kişiden kişiye farklılık gösterebilir. Bu fark kişilerin gaz-fren dengesinin aynı olmamasından kaynaklanır. Bir kişi hızlanarak kazandığı enerjiyi yumuşak geçişlerle frene, daha doğrusu ısı enerjisine daha az, yola daha fazla harcarken; diğer kişi tam tersi davranabilir. Böylece aynı mesafeye farklı yakıt harcanmış olur. Otonom kara aracında yakıt tasarrufunu arttıran bir başka özellikte sürüş esnasındaki düzgün gaz-fren dengesi olacaktır. Motoru daha önceden belirlenen en uygun devirde kullanmak da buna dâhil edilebilir.

Otonom kara araçlarının tüm bunların yanında yaşlı, yorgun, alkollü vb. gibi araç kullanmaya müsait olmayan veya araç kullanmayı tercih etmeyen insanlar için de bir seçenek olması başka pozitif yanıdır.

Otonom araçların tüm bu bahsedilen yetenekleri kazanabilmesi için zamana ihtiyaç vardır. Üzerlerine birçok sistem dahil edilmelidir. Fakat günümüzde bu sistemlerden bazıları ufak uygulamalar halinde binek araçlara yavaş yavaş geçmekte, hatta bu özellikler kullanıcıya sunulmaktadır. Kendi kendine park eden araçlar, şehir içinde yaya güvenine yönelik kullanıcıdan bağımsız ani fren sistemleri, kendi hızını öndeki aracın hızına ayarlayan sistemler bunlara örnek sayılabilir. Bu sistemler tam olarak karar verme yetisine sahip olmadığı için yarı otonom olarak sınıflandırılabilirler. Fakat bunun bir geçiş noktası olduğunu, bir sonraki aşamanın otonom sistemler, otonom kara aracı olduğunu unutmamak gerekir.

Genelde otonom kara araçlarındaki yapı birbirine benzerdir ve Şekil 1.3’de de görüldüğü gibi dört temel bloğa ayrılabilir:

- Sensörler
- Veri işleme merkezi
- Karar verme merkezi
- Sürücüler ve Aktüatörler



Şekil 1.3 : Otonom araç genel sistemi

İlk birim çevreden ve araçtan bilgi toplayan algılayıcılardan meydana gelir. Bu algılayıcılar çok farklı nitelikteki elektriksel sinyallere çevirir, bir sonraki birime aktarır.

Algılayıcılardan gelen veriler çok fazla ve çok karmaşık bir formata sahiptir. Bunların direkt olarak karar mekanizması tarafından yorumlanması ve daha sonra buna göre bir karar alınması zaman kaybına yol açabilir. Bu sorunu aşmak için bu

tarz sistemlerde iş yükü ikinci bir birime paylaştırılır. Bu birim veri işleme merkezidir. Görevi verileri işlemek ve sensörlerin doğruluğunu birbirine göre test etmektir. Veri işleme merkezi karmaşık ve yüklü veri topluluğunu, sade ve düzenli hale getirdikten sonra bunları karar verme merkezine yollar.

Karar verme mekanizması kullanıcının ilk girdiği isteği ve veri işleme merkezinden gelen bilgileri değerlendirdikten sonra aracın ne yapması gerektiğini, hangi hızda hangi yöne gideceğini belirler. Bu işlemin sonucunda aktüatör sürücülerini, dolayısıyla da aktüatörleri hareketlendirir.

Sürücüler, içlerinde bulunan kontrol algoritmalarıyla kendilerine gelen referans değerlere göre aracın fonksiyonlarına(gaz, fren, vites, kontak, direksiyon) müdahale ederler. Bu sayede çevrim tamamlanmış olur.

1.2 Otonom Araçlarla İlgili Yapılan Çalışmalar

Birçok araştırmacı tarafından otonom araçlar üzerine yapılan çalışmaların ilk örneğinin 1939 yılında General Motors tarafından düzenlenen Futurama Dünya Fuarı ile başladığı kabul edilmektedir. Bu araç şehir içindeki yollarda, insanları güvenli bir şekilde otomatik olarak taşıyan bir araç olarak tanıtılmıştır [4]. Ancak dikkate değer ilk çalışmanın 1977 yılında Japonya'nın Tsukuba Makine Mühendisliği Laboratuvarı'nda başladığı söylenebilir. Burada geliştirilen araç beyaz bir çizgiyi takip eden ve 30 km/h hıza çıkabilen bir araçtı. Bu hız günümüzle kıyaslandığında çok düşük gibi görünse de o yıllarda öne çıkan bir gelişmeydi [5]. 1980'li yıllarda Ernst Dickmanns ve ekibi tarafından Bundeswehr Üniversitesi'nde geliştirilen araç hızı 95 km/h hıza ulaşan üzeri kamera ve sensörlerle dolu bir araç ürettiler. Güvenlik sebeplerinden ötürü ilk denemeler trafiğe kapalı bir alanda yapıldı [4]. Ardından gelen European Prometheus Projesi ile 800 milyon dolarlık bir proje başlamıştır. 1995 yılında Carnegie Mellon Üniversitesi, Navlab isimli projesi ile, No Hands Across America, %98.2 verim ile 5000 km yol almıştır. DARPA (Defense Advanced Research Projects Agency) 2004 ve 2005 yılında otonom araç yarışları düzenlemiş ve kazanan takıma milyon dolarlık ödül vaat etmiştir. 2010 yılına gelindiğinde VisLab adlı şirket, 13000 km'lik bir yarışma düzenledi, VIAC (Vislab Intercontinental Autonomous Challenge). Yarışma sonunda 4 otonom sürücüsüz elektrikli araç İtalya'dan Çin'e Shanghai'daki Expo'ya uğrayarak yarışmayı bitirdiler

[6]. Google ise 2010 yılında otonom bir araç geliştirmiş ve bu araç 230000 km yol kat etmiştir.

1.3 Hız Sabitleyicisi (Cruise Control)

Önceleri sadece belirli bir güzergâhta gidip gelmeleri istenen otonom araçların gün geçtikçe geliştirilmesiyle birlikte bu araçlara eklenmek istenen özellikler de artmıştır. Hız sabitleyicisi de (Cruise Control) bunlardan biridir. Hız sabitleyicisi otomotiv sektöründe bir süredir kullanılan ve sürücüyü özellikle uzun yolculuklar sırasında rahat ettiren bir uygulamadır. İlk bakışta sadece konfor artırıcı bir uygulama gibi görünse de aslında sadece konfor için değil aynı zamanda güvenlik açısından da önemli bir uygulamadır. Sürücünün uzun yollarda sürekli gaz ya da fren pedalına basmasını engelleyen bu sistem hızı istenilen seviyede sabit tutmaya yaramakta ve sürüşten kaynaklanan yorgunluğu minimize etmektedir. Bu da yolculuk esnasında sürücünün üzerine düşen görevi azaltmakta sürücüyü rahatlatmaktadır. Ancak herhangi bir kaza anında bu sistemin müdahale yapması mümkün değildir. Sadece kazaya neden olabilecek sürücü kaynaklı sorunları bir miktar düşürebilir.

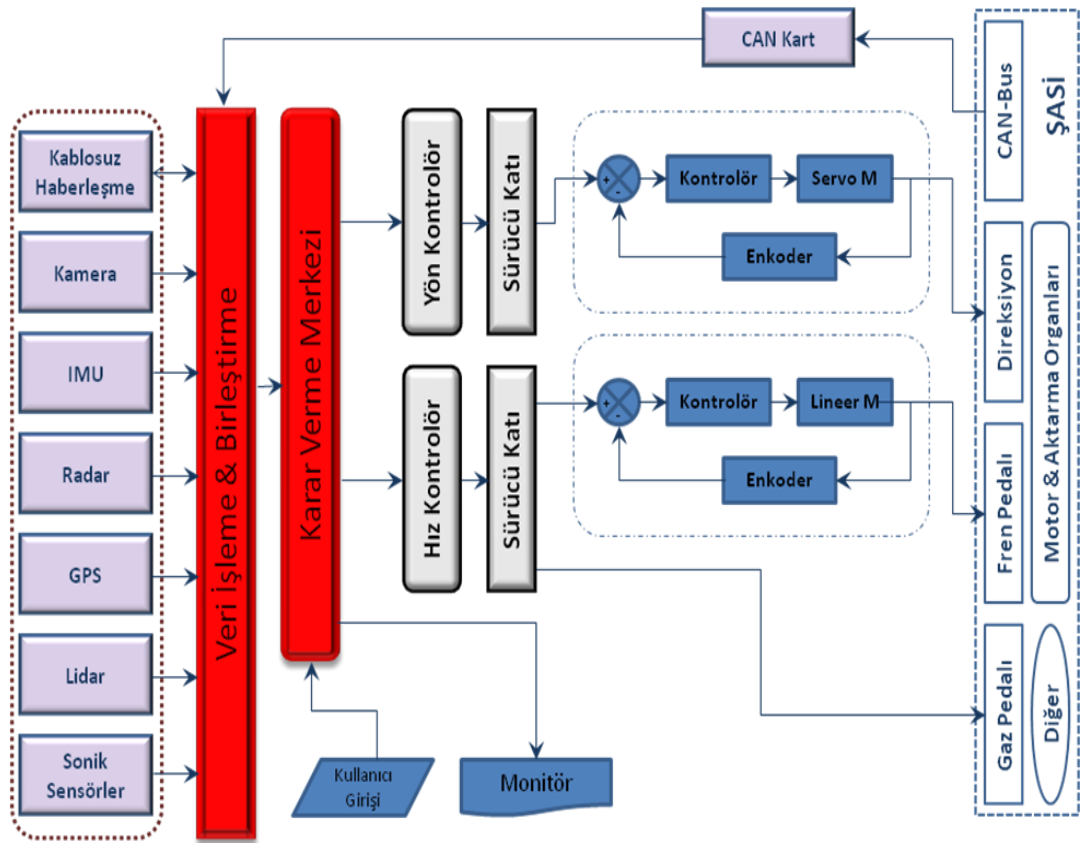
Bu sistemde sürücünün stabil hareketleri hız sabitleyicisi tarafından gerçekleştirilir. Sistemin devreden çıkartılması için frene ya da gaza dokunulması ya da direksiyon üzerinde bulunan düğmelere dokunulması yeterlidir. Hız sabitleyici sistemlerdeki en büyük dezavantaj yokuş yukarı ya da yokuş aşağı gidildiğinde hızda bir miktar değişim olmasıdır. Bu problem zamanla çözülmüş ve yol durumu ne olursa olsun araç hızı istenilen seviyede tutulmuştur. Ardından Adaptif Hız Sabitleyicisi (Adaptif Cruise Control) geliştirilmiştir. Bu sistemdeki mantık da yine hız sabitleyicisinde olduğu gibidir. Ancak buradaki fark, öndeki araçla aradaki mesafe araçta bulunan sensörler yardımıyla otomatik olarak algılanır. Adaptif hız sabitleyici kullanılarak seyahat edilirken sürücünün önüne daha yavaş hareket eden bir araç çıktığında sistem otomatik olarak bunu algılar ve aradaki mesafeyi sabit tutmak için hızı otomatik olarak azaltır. Bu işlemi ilk olarak frene basarak değil motor freni vasıtasıyla sağlar. Öndeki araç hızlandığında ya da menzilden çıktığında sistem kendini otomatik olarak eski hıza ayarlar. Hız sabitleyici sistem daha az stres ve daha az yorgunluk sağladığı gibi yakıt tasarrufu konusunda da büyük avantaj sağlamaktadır.

Hız kontrolörü konusunda bugüne kadar birçok farklı çalışma yapılmıştır. Bu çalışmalardan bazıları aracın akıllı bir yolda sabit hızla ilerlemesi problemini ele

alırken [8], bazıları ise öndeki aracı takip etme problemini ele almıştır [9]. Bu tez çalışmasında ise kullanıcının belirlediği referans hıza göre araç için boylamsal (longitudinal) kontrol yapılmıştır.

1.4 Tasarlanan Otonom Aracın Yapısı ve Hız Kontrolör

O³ projesi kapsamında mevcut olan aracın otonom yeteneğini kazanabilmesi için Şekil 1.4’de de görüldüğü gibi birçok aygıt ve sistemin araçla bir araya getirilmesi düşünülmektedir. Projenin ilk evresinde kısa vadeli hedef, elektronik sinyallerle otomobili yürür hale getirmek olarak belirlenmiştir. Bunun için aracın üç ana fonksiyonunu (ivmelenme sistemi, frenleme sistemi ve direksiyon sistemi) kontrol etmek gerekir.



Şekil 1.4 : Tasarlanan sistemin genel şeması

Aracı yönlendirmek için direksiyona etki eden servo motorlu bir mekanik yapı düşünülmüştür. Kullanılacak olan bu aktuatör bir dişli ve redüktör vasıtasıyla direksiyon miline tahrik verecektir. Motorun sürme işlemi ise kendi içersinde kontrol algoritması bulunan ve referans değeri dışarıdan dijital olarak alan bir sürücü ile gerçekleştirilecektir.

Aracı yavaşlatmak için ise lineer bir servo motor ile fren pedalına müdahale edilecektir. İleri ve geri hareketle pedala etki edecek olan elemana, direksiyon da olduğu gibi kendi içersinde konum kontrolörüne sahip sürücü ile yol verilecektir. Sürücünün referans değeri olarak dışarıdan analog işaret alması düşünülmektedir.

İvmelenme için ise diğerlerinde olduğu gibi yeni bir mekanik aksam yerine, tasarlanacak elektronik bir kart aracılığıyla gaz pedalından araca giden analog sinyallerin birer kopyasının üretilip, bir sonraki eleman olan ECU'ye iletilmesi planlanmıştır. Böylelikle araca pedala basılıyormuş hissi verilecektir. Elektronik kartın referans girişi ise sayısal nitelikte olacaktır.

Protip olarak kullanılan otomobil otomatik vitese sahip olduğu için en azından bu aşamada vitese müdahale etme ihtiyacı duyulmamış, kontak anahtar için ise bir düzenek hazırlanmamıştır.

Eyleyiciler ile ilgili bu çalışmalar sürerken, sensörlerden gelen verilerin toplanma ve kullanılabilir hale getirilme işleminde paralel olarak sürdürülmesi planlanmıştır. Aracı, üç ana fonksiyonuna erişerek otonom olarak kontrol edecek olan sistemin, bulunan ortam ve araç hakkında birçok bilgiye ihtiyacı vardır. Bu verileri elde etmek için donanımsal olarak eklenmesi düşünülen lidar, radar, GPS gibi sensörlerin yanı sıra otomobilin kendi algılayıcılarından da faydalanılması düşünülmüştür. Bu işlem için araca ait bilgileri yer aldığı haberleşme ağına erişilecektir.

Öncelikle yakın mesafede geniş açılı ve ayrıntılı bir alan taraması için aracın ön tamponuna monte edilen lidar cihazı kullanılacaktır. Yüksek hızlarda uzun mesafeli bir görüş için ise tarama alanı daha dar olan radar gereklidir. Aracın bulunduğu konumu ve gideceği hedef noktayı bilebilmesi için ise hassasiyeti yüksek bir GPS düşünülmüştür. Sistemin trafik ışıkları, uyarı levhaları, yol çizgileri gibi öğeleri okuyabilmesi için ise renkleri ayırt edebilecek bir veya iki kamera kullanılması hedeflenmiştir. Şekil 1.4'de yer alan IMU'nun görevi ise aracın üzerindeki ivme ölçerin duyarlılığının yetersiz kaldığı durumlarda GPS verilerinin kalibrasyonunu yapmaktır. Tünel ve benzeri ortamlarda GPS verisi alınamadığında, sisteme geçici olarak konum bilgisi verme işlevi yine IMU'nun görevi olacaktır.

Tüm bu algılayıcılardan gelen verilerin anlamlandırılıp, bir noktada toplanması ve birbirine göre doğruluğunun kontrol edilmesi gerekmektedir. Oldukça fazla iş yükü getiren bu veri işleme görevinin karar verme merkezine verilmesi sistemi fazlasıyla

yavaşlatıcak, belkide riskli bir duruma taşıyacaktı. Bu sebeple bu iş için ikinci bir merkezi birimin kullanılması, verilerin işlenmiş, daha sade ve anlaşılır bir biçimde bir sonraki sistem olan karar verme merkezine aktarılması tasarlanmıştır.

Karar verme merkezini bu noktadaki işlevi ise veri işleme merkezinden gelen bilgileri değerlendirip, mevcut koşula göre uygun kararı belirlemesi ve kontrol algoritmalarını da kullanarak eyleyiciler için referans değer üretmek olacaktır.

Sistemde karar verme merkezinin belirlediği istenilen araç hızını tutturmak için, yine aynı birim içersinde yer alan hız kontrolörünün görevi genel çalışma düzeninin işleyebilmesi için büyük önem arz etmektedir. Bu çerçevede hız kontrolörünü oluşturmak için ilk olarak CAN üzerinden araç hızı okunacak, devamında hız için gerekli iki ana fonksiyona(ıvmelenme ve frenleme sistemi) elektronik olarak erişim sağlanacaktır. Bu işlem mekanik ve elektronik tasarımların yanı sıra uygun yazılım oluşturulması ile gerçekleştirilecektir. Devamında alt kontrolörler tasarlanacak, simülasyon ortamında iyileştirilip, gerçek uygulama testleri ile ince ayarlar yapılarak optimum çalışma noktaları yakalanacaktır. Son olarak ise hız kontrolörün üst düzey denetleyicisi oluşturulup, uygulamada ve simülasyonda çeşitli senaryolar ile sınanacaktır.

2. ARAÇTAKİ HABERLEŞME SİSTEMİ ve BU SİSTEME ERİŞİM

Araçlarda gün geçtikçe artan konfor ve güvenlik ihtiyacı üretici firmaları ar-ge çalışmalarına daha fazla yatırım yapmaya ve araçlara yeni özellikler kazandırmaya zorlamaktadır. Yeni her özellik ise araca ek bir elektronik modül yükü getirmekte ya da var olan bir modülün geliştirilmesini mecburi kılmaktadır. Görev atanmış modül sayısının artması ya da var olan modüllerin daha da yüklü bir hale getirilmesi ile beraber araçlardaki kontrol sistemlerinin haberleşmesi zamanla analog sinyalden dijital sinyale dönüşmüştür[10]. Dijital sinyale geçişin yaşandığı bu devrim esnasında, kullanılan kablo yoğunluğunun azaltılması da bir gereklilik halini almıştır. Bu sebeple araçlarda CAN (Controller Area Network) adı verilen bir haberleşme protokolü geliştirilmiş ve tüm birimler arası iletişim bu hat üzerinden yapılmıştır.

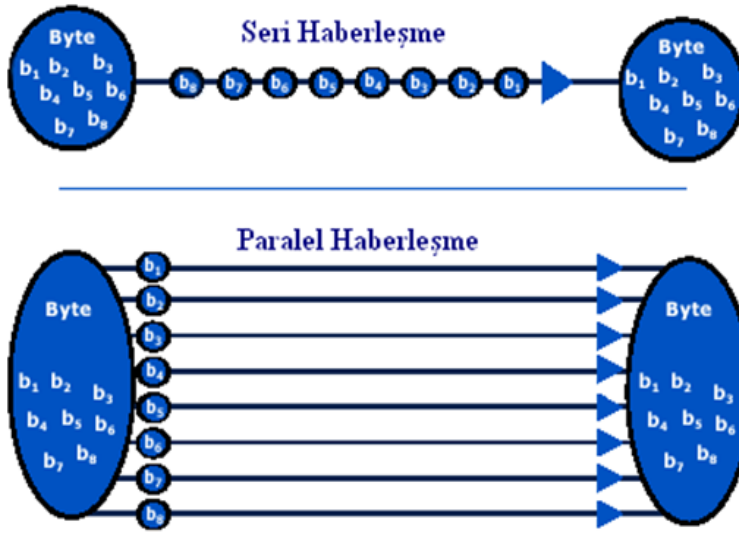
Bu bölümde öncelikle CAN protokolünü daha iyi anlayabilmek için seri ve paralel haberleşme türlerine değinilecektir. Ardından CAN protokolü ve fiziksel yapısı incelenerek çalışma prensibi anlatılacaktır. Son olarak araç üzerindeki haberleşme ağına erişilerek bu ağ üzerinden istenilen veriler çekilip kaydedilecektir.

2.1 Paralel-Seri Haberleşme

Haberleşme türleri iletim hattı sayısına göre seri ve paralel olmak üzere genel olarak ikiye ayrılır. Paralel haberleşme verilerin birçok farklı kanaldan aynı anda taşınması prensibine göre çalışır. Bu durum kısa mesafelerde hız açısından büyük bir avantaj olsa da iletimi sağlayan kablo sayısının artması ve bilgi taşınırken meydana gelebilecek kayıplar açısından dezavantajdır. Bu sebeple aynı anda birden çok sinyal göndermeye yarayan paralel haberleşme zamanla yerini veriyi tek bir hat üzerinden ardışık olarak gönderen seri haberleşmeye bırakmıştır

Seri haberleşmede bilgi aktarımı için Şekil 2.1’de de görüldüğü gibi üniteler birbirine tek bir hat üzerinden bağlanır ve aktarılabilecek olan veriyi bit bit gönderirler. Fakat her iki haberleşme türünde de 1 ve 0 sinyallerinin anlaşılabilmesi için çoğunlukla 0 değerini taşıyan referans sinyalin de fiziksel ortamda alıcı tarafa ulaştırılması

gerekmektedir. Bu sebeple hem seri hem de paralel haberleşmede iletim hattı sayısı Şekil 2.1’de görüldenden bir fazla olacaktır.



Şekil 2.1 : Seri ve paralel haberleşme [11].

Teknolojideki gelişmeler ve gereksinimler seri haberleşmeyi ileri noktalara taşımış RS232, I2C, CAN, LIN, Flexray gibi çeşitli protokoller ve topolojiler ortaya çıkarmıştır [12]. Bu protokoller zamanla standartlaşmış ve hızlarına göre Çizelge 2.1’de olduğu gibi sınıflandırılmıştır.

Çizelge 2.1 : CAN ve diğer haberleşme protokollerinin sınıflandırılması [13].

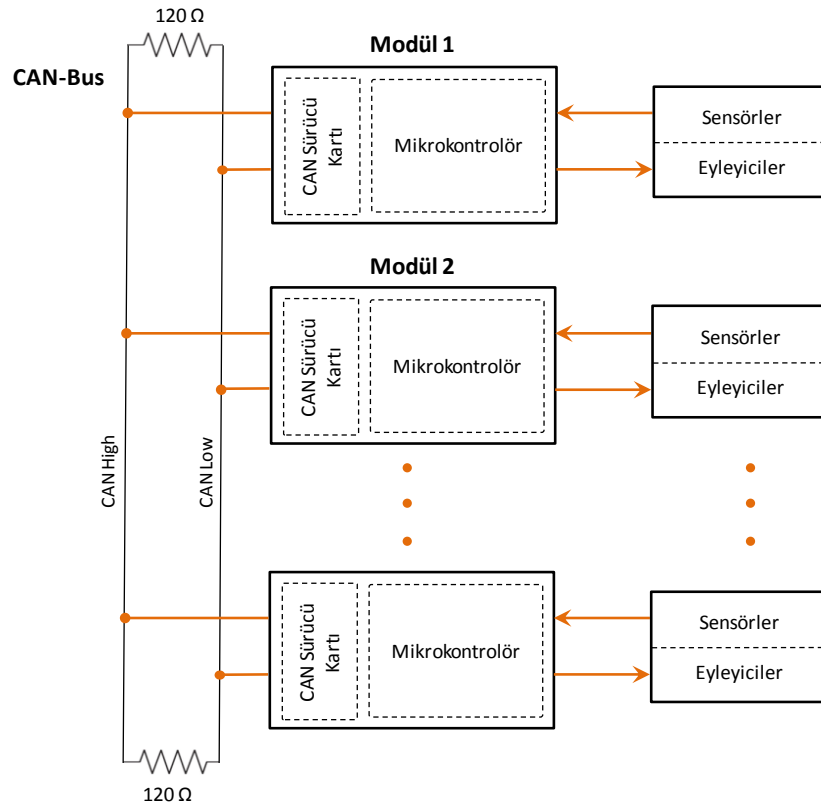
Sınıf	Haberleşme Hızı	Kullanım Amacı	Protokoller	
			CAN	Diğer Protokoller
A sınıfı	10 kbps’a kadar	Lambalar Pencereler Kapı kilitleri Anahtarsız giriş Elektronik metre	Düşük hız ↕ Yüksek hız	— Her üreticinin kendi protokolü
B sınıfı	10 kbps – 125 kbps	Sürüş bilgisi Otomatik klima Hata tanımlama Motor kontrolü		— LIN — J1850 — VAN
C sınıfı	125 kbps – 1 Mbps	Vites kontrolü Fren kontrolü Süspansiyon kontrolü		— Safe-by-Wire
D sınıfı	5 Mbps ve üstü	Video Navigasyon		— D2B optik — MOST — IEEE 1394 — Flexray

2.2 CAN (Controller Area Network)

CAN (Controller Area Network) 1980'li yıllarda Bosch Corporation tarafından geliştirilmiş seri bir haberleşme protokolüdür. Bu protokol her ne kadar endüstrinin çeşitli alanlarında kullanılsa da en yaygın kullanım alanı otomotiv sektörüdür. Araçlardaki modüller arası veri aktarımını hızlı ve güvenli bir şekilde gerçekleştirmek için tasarlanmıştır. CAN sayesinde, kullanılan çoğu mekanik ve hidrolik sistemin yerini elektronik ağ yapıları ve kontrol üniteleri almıştır. Bu sayede kullanılan kablo miktarı azalmış, maliyet düşmüş, bağlantı yapısı sadeleşmiş ve sistemin güvenilirliği artmıştır [14].

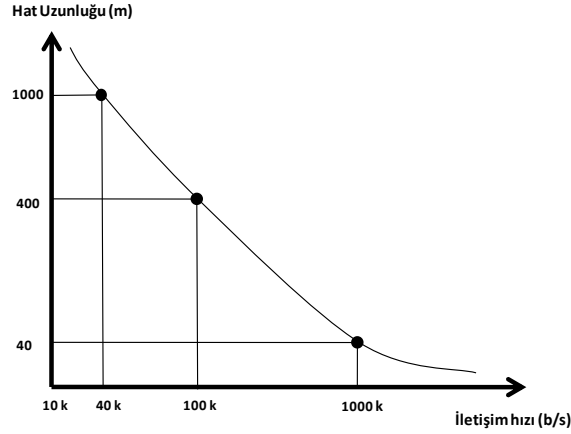
2.2.1 Ağın fiziksel yapısı

CAN hattı farklı seviyelerde sinyal içeren, CAN-High ve CAN-Low olarak adlandırılan iki elektrik hattından meydana gelmektedir. Bu iletim hatları Şekil 2.2'de görüldüğü gibi iki taraflı olarak $120\ \Omega$ 'luk dirençlerle sonlandırılır ve haberleşme kurulacak her modül bu dirençlere paralel olarak bağlanır. Yeni bir birim ekleme işlemi eski sistem bozulmadan yapılabileceği için bu protokolün kullanımı araçlara çok büyük avantajlar getirmektedir.



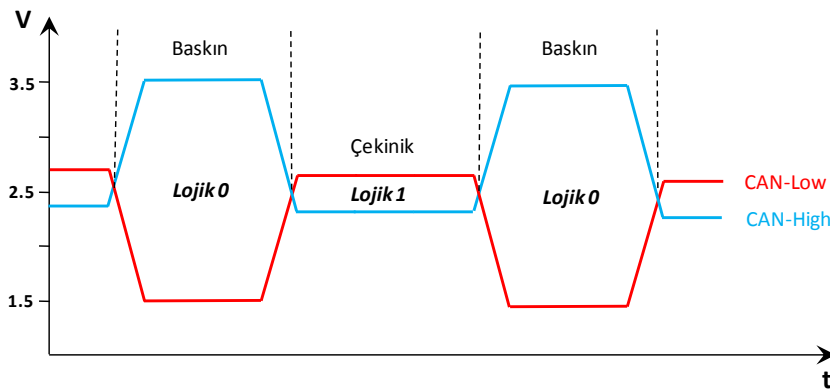
Şekil 2.2 : CAN bağlantı yapısı [11].

CAN üzerindeki iletişim hızı arttıkça, fiziksel iletim hattının maksimum uzunluğu azalır. Bu iki parametre arasında ters bir orantı vardır. Haberleşme esnasında veri kaybına uğramamak için hat uzunluğu optimum düzeyde olmalıdır.



Şekil 2.3 : CAN hattı uzunluğuna göre iletişim hızı [13].

Hat üzerindeki verinin lojik 1 ya da lojik 0 olduğu ise CAN-High ve CAN-Low'daki sinyallerin birbirlerine göre durumlarına bakılarak anlaşılır. CAN-High'daki sinyal, CAN-Low'daki sinyalden genlik olarak daha büyükse hat üzerinde lojik 0, tersi durumda ise lojik 1 değeri vardır. Her iki durumda da sinyaller gerilim seviyesi bakımından GND'nin üzerinde yer alır. Özetle hat üzerinde 1 ya da 0 olması GND referans alınarak değil CAN'deki iletim hatlarının birbirlerine göre durumları ile belirlenir [15]. Bu sayede veri dış etkenlerden kaynaklı parazit sinyallerden daha az etkilenir.



Şekil 2.4 : Can-High ve Can-Low gerilim değerleri [16].

Şekil 2.4'de de görüldüğü gibi lojik 0 baskın bit, lojik 1 ise çekinik bittir. Diğer bir deyişle hatta veri yazmaya başlayan iki birim aynı anda 1 ve 0 bitlerini gönderirlerse diğer tüm birimler hattı lojik 0 seviyesinde okurlar.

CAN'e eklenen her modül daha önceden belirli görevleri yerine getirmek üzere çevresel ünitelerle beraber bir araya getirilmiş ve programlanmış mikrobilgisayarlardır. Bir modülün kendi işlemcisi olduğu gibi ROM, RAM, EPROM, ADC, DAC gibi elemanlara da sahiptir.

2.2.2 CAN protokolü

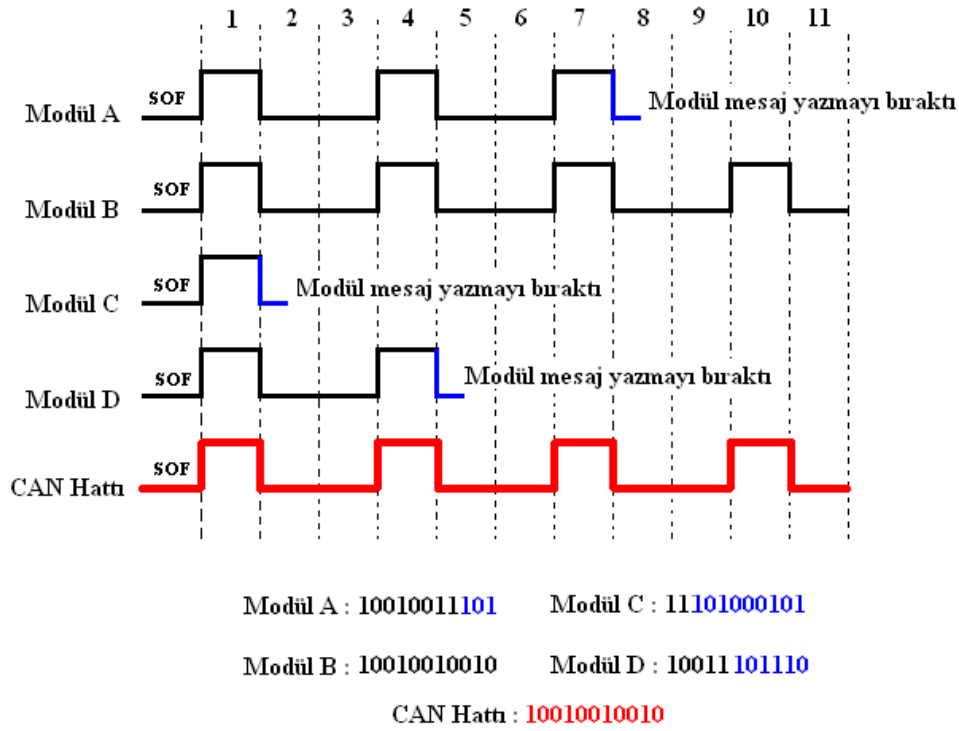
CAN mesaj tabanlı işleyen bir haberleşme protokolüdür. Yapısında haberleşmeyi yöneten herhangi bir merkezi birim bulundurmaz. Bunun yerine bağlı olan tüm birimler kullandıkları verilerin hata kontrolünü ve hattın meşgul olup olmadığını kendileri kontrol ederler. Bir nevi denetim ögeleri arasında dağıtılmıştır. Veri göndermesi gereken modül eğer hattı boş olarak okursa mesaj paketini iletim ortamına yazmaya başlar. Diğer tüm birimler yayınlanan mesajı duyar; ancak sadece ilgili modül ya da modüller yayınlanan mesajı okur ve değerlendirir. Bu noktada lokal bir filtreleme işlemi gerçekleşir.

Yayınlanan mesajlar paketin hemen başında yer alan ve sadece bu mesaja has olan tanımlayıcıya (**ID-Identifler**) sahiptir. Diğer modüllerin yayınlanan mesajı alıp almayacağı bu ID'ye bağlıdır. Çünkü CAN protokolünde veri mesajları alıcı adreslerini içermezler. Bunun yerine alıcı birimler kullanacağı verileri içeren mesajların ID'lerini bilirler ve buna göre bir ayırım yaparlar.

İki modül hat üzerine aynı anda mesaj paketlerini basamaz. Eşzamanlı olarak iki veya daha fazla birim böyle bir durumu gerçekleştirdiklerinde hangi mesajın geçerli olacağı ID numarası tarafından belirlenir. Gönderici konumundaki tüm modüller mesaja ait lojik 1 değerindeki bitleri hat üzerine yazarken, aynı anda bu bitin başka bir modül tarafından bozulup bozulmadığını da denetler. Eğer bir birim lojik 1 yazarken başka bir birim lojik 0 yazıyorsa iletim ortamı baskın olanın(lojik 0) değerini alacaktır. Böylece hat üzerine lojik 1 gönderen modül için basılan bit seviyesi ile okunan bit seviyesi aynı olmayacaktır. Bu durumda gönderdiği veriyi alamayan modül yazma modundan okuma moduna geçecektir. Her mesaja ait tek bir ID numarası olduğundan, paket başında yer alan tanımlama alanına ait bitlerin gönderilmesi sona erdiğinde iletim ortamında en fazla bir modül yazma konumunda kalacaktır. Diğer bir deyişle en baskın(küçük) ID'ye sahip modül hattı kullanır.

Şekil 2.5 incelendiğinde CAN hattına eşzamanlı olarak mesaj basmaya çalışan 4 farklı modül görülecektir. Mesajlar SOF (**SOF-Start of Frame**) bitinin lojik 0, yani

baskın bit olmasıyla başlar. Ardından mesajın 11 bitlik ID kısmı gelir. 2. bit zamanında diğer modüller hatta baskın bit gönderirken, modül C hatta çekinik yani lojik 1 bitini göndermiştir. Bu durumda modül C için bitlerin geri kalınmasına bakılmaksızın veri gönderme işlemi sonlandırılır ve dinleme moduna geçilir. Aynı şekilde 5. ve 8. bit zamanlarında da modül D ve modül A okuma moduna geçmiş, diğer modüllere göre baskın bitler gönderen modül B CAN hattına mesaj yazmaya devam etmiştir.



Şekil 2.5 : Mesaj önceliği.

2.2.3 Mesaj türleri

CAN hattında yayınlanan mesajlar amaçlarına, yayınlanma anlarına ve mesaj ID uzunluklarına göre sınıflandırılırlar.

2.2.3.1 Amaçlarına göre CAN mesaj çeşitleri

Veri mesajı

İçinde diğer modüllerin kullanması için gerekli verileri taşıyan mesaj çeşididir.

İstek mesajı

Her hangi bir modül işlem yapmak için başka bir modüldeki veriye ihtiyaç duyduğunda hatta bir mesaj yazar. Veri içermeyen bu mesaja istek mesajı denir.

Hata mesajı

Yayınlanan bir mesajı alan modül bitlerde bir bozulma olup olmadığını anlamak için aldığı paket içerisindeki CRC (**Cyclic Redundancy Code**) kodu ile veriyi karşılaştırır. Bu iki değer birbiriyle tutarlıysa alınan paket geçerlidir. Eğer değilse diğer modüllerin yanlış mesajı kullanmaması ve yayıncının bilgiyi tekrar göndermesi için hatayı tespit eden modül tarafından CAN hattına yazılan mesajdır. Hata mesajını modüllerden sadece biri bile üretirse veri tekrar yollanır.

Aşırı Yük Mesajı

Alıcının, kullanacağı veriyi içeren mesajın gönderici tarafından yayınlanmasını geciktirmek için hatta yazdığı mesajdır. Alıcı modül daha önceki işini bitirememişse, sıradaki paketi kaçırmamak için bu mesajı kullanır [17].

2.2.3.2 Yayınlanma anlarına göre CAN mesajları

Olay mesajı

Yalnızca belli bir olay gerçekleştiğinde ilgili modül tarafından hat üzerinde yayınlanan mesajdır. Araç kapısı kapandığında yada açıldığında üretilen mesaj olay mesajlarına örnek olarak verilebilir.

Periyodik mesaj

Belli zaman aralıkları ile hat üzerinde yayınlanan mesajlardır.

Periyodik olay mesajı

Belli bir olay gerçekleştiğinde değil, periyodik olarak da hatta basılan mesaj türüdür.

İstek tabanlı olay mesajı

İstek mesajı yayınlandığında bu mesaja karşılık olarak gönderilen özel bir mesajdır.

Çoklu mesaj

Gönderilen bir mesaj paketinde en fazla 8 byte'lık veri bulunabilir. Ancak veri boyutunun 8 byte'ı aştığı durumlarda bilgi küçük paketlere ayrılır ve bu şekilde gönderilir. Parçalanmış verileri taşıyan her paket çoklu mesaj olarak adlandırılır. Gönderici toplam paket sayısını ve gönderilen mesajın kaçınıcı paket olduğunu çoklu olarak ürettiği mesajlara ekler.

2.2.3.3 ID uzunluğuna göre CAN mesajları

Standart format

ID uzunluğu 11 bitten meydana gelen mesajlardır.

Geniřletilmiř format

Mesaj ID uzunluęu 29 bittir. Geniřletilmiř formata mesaj eřitlilięinin fazla olduęu sistemlerde ihtiya duyulur [18].

2.2.4 Mesaj yapısı

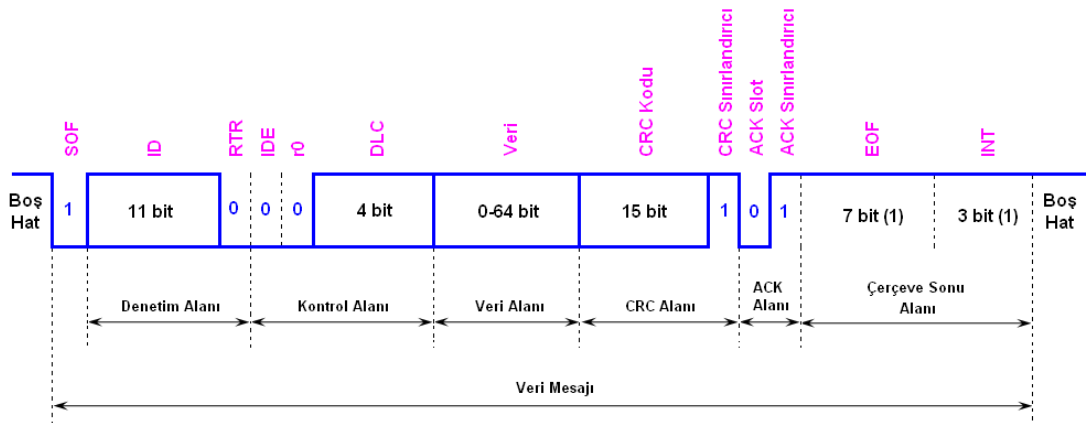
CAN hattında yayınlanan her mesaj bir formatta paketlenir. Bu paketler farklı alanlara ayrılmıř ve CRC, ID, DLC gibi kodlamalar verilerle birlikte bu alanlara yerleřtirilmiřtir.

2.2.4.1 Veri mesajı

Standart formatta bir veri mesajı yapısı

Yayınlanan bir mesajın ilk bitine ereve bařlangı biti (**SOF-Start Of Frame**) denir. Modül hat zerindeki lojik 1 deęerini daha baskın olan ve lojik 0 deęerini taşıyan SOF bitiyle alt seviyeye ekerek mesaja bařlar. Bu bitin grevi dięer modllere hattın meřgul olduęunu ve okumaları iin mesajın bařladıęını bildirmektir.

SOF bitinin ardından, mesaj formatına gre uzunluęu deęiřebilen denetim alanı (**Arbitration Field**) gelmektedir. Bu alan standart formattaki veri mesajları iin 12 bit byklęndedir ve ilk 11 bit mesaj tanımlayıcı numarasıdır (**ID-Identifier**). 12. bit ise uzak iletim istek bitidir (**RTR-Remote Transfer Request**).

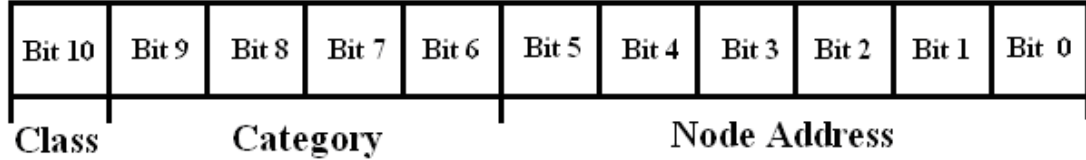


řekil 2.6 : Standart formatlı veri mesajı [17].

Bir modl CAN hattı zerinde yayınlanan tm mesajları grr, fakat tmn kullanmaz. Her modl kullanacaęı verinin hangi mesajda olduęunu bilir. Okuması gereken mesajların her birinin ID numarası o modlde kayıtlıdır. ID numarası her mesaj iin tekil ve yalnızca o mesaja ait olduęundan modl istedięi mesajları ID

numaralarına göre filtreler ve değerlendirir. ID numarasına göre filtreden geçemeyen mesajların devamı değerlendirilmez. İşte bu sebeple CAN protokolü için “mesaj tabanlı” denilmektedir.

ID numarası, mesajı yayınlayan modülün numarası (**Node Number**), mesajın önceliği (**Category or Priority**) ve mesajın türü (**Class**) gibi bilgileri barındırır. 11 bitten oluşan bir ID’nin yapısı Şekil 2.7’de gösterilmiştir.



Şekil 2.7 : 11 bitlik ID yapısı [13].

Mesajın ID’si hat üzerine en önemli bittinden (**MSB-Most Significant Bit**) en az önemli bite (**LSB- Least Significant Bit**) doğru sırası ile yazılır. Şekil 2.7’de verilen yapıya göre ilk olarak mesajın içindeki verilerin türünü belirten bitler yer alır. Bu bitler mesajın bir uygulama mesajı mı yoksa bir durum mesajı mı olduğunu belirtir. Ayrıca mesajın kritik olup olmadığı hakkında da bilgi verir. Lojik 0 durumu baskın olduğundan, kritik ve önceliği olan mesajlara en küçük değerler atanır.

Çizelge 2.2 : Mesaj sınıf biti [13].

Sınıf (bit 10)	
A0 (Uygulama mesajı)	0b
A1 (Durum bildirme mesajı)	1b

ID numarasının kategori kısmında mesajın önceliği daha da belirgin hale gelir. Çizelge 2.3’de gösterildiği gibi kategoriler yüksek önem taşıyan mesajlardan daha az önem taşıyan mesajlara doğru sıralandıkça ID numarası da büyümektedir.

Çizelge 2.3 : Mesaj öncelik bitleri [13].

Kategori (bit 9-bit 6)	
B0 (Yüksek öncelik)	0000b
...	...
B4	0100b
...	...
B9	1001b
...	...
B15 (Düşük öncelik)	1111b

CAN hattına veri yazmak için bağlanan tüm modüller Node numarası da denilen bir adrese sahiptir. ID numarasının son kısmı mesajın üretildiği modülün adresini içermektedir. 11 bitlik ID alanı içersinde Node adreslerine 6 bit ayrılması, o iletim hattı üzerine maksimum 64 modül bağlanabileceği anlamına gelir. Burada üzerinde durulması gereken bir başka nokta da 0'ın baskın bit olmasından dolayı en önemli mesajları üreten modülün ID numarasının en küçük değeri olmasıdır[13].

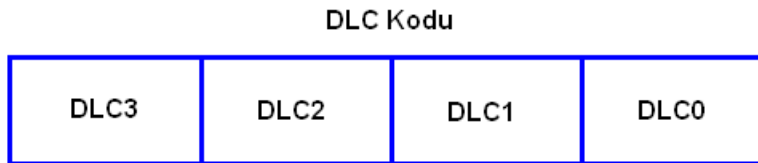
Çizelge 2.4 : Node numarası.

Node ID (bit 5-bit 0)	
0	000000b
...	...
...	...
...	...
63	111111b

İki veya daha fazla birim CAN üzerine aynı anda mesaj yazmaya başladığında ID numarası küçük olan mesaj hatta yazılmaya devam edecek, diğer mesajlar ise ait oldukları yayıncı modül tarafından durdurulacaktır. ID numarası en küçük olan mesaj en öncelikli mesaj olduğu için son olarak CAN hattında en önemli mesaj kalacaktır.

Denetim alanının bir diğer biti olan RTR ise veri mesajında mesajı gönderen birimin diğer modüllerden bir talebi olmadığı için 0 değerini alır. İçinde veri içermeyen istek mesajı olması durumunda ise bu hücreye 1 değeri yerleştirilir.

Denetim alanının ardından 6 bitlik kontrol alanı gelmektedir. Standart formattaki bir veri mesajında bu alan tanımlayıcı uzunluk biti (**IDE-Identifier Extension**), ayrılmış bit (**r0-Reserve**) ve veri uzunluk kodundan (**DLC-Data Length Code**) oluşmaktadır.



Şekil 2.8 : DLC koduna ait bit yerleşimi [17].

IDE biti mesajın ID numarasının 11 bitlik ya da 29 bitlik alan kapladığını belirtir. Diğer bir deyişle mesajın standart formda ya da genişletilmiş formda olduğunu gösteren bittir. Bu bitin 0 olması mesajın standart formatta, 1 olması ise genişletilmiş formatta olduğunu ifade eder.

Her mesaj içersinde 0–8 byte arasında veri taşır. Kontrol alanında yer alan DLC kodu mesajın içindeki veri boyutunu verir. DLC 4 bitten oluşur ve Çizelge 2.5’deki gibi kodlanır. 4 bitlik bir alan 16 farklı durumu temsil edebilir. Ancak mesajlar 0–8 byte veriye sahip olduğunda sadece 9 farklı durum ortaya çıkar. Geri kalan 7 durum boştur ve kullanılmayabilir (d:dominant, r:resesive).

Çizelge 2.5 : Veri uzunluk kodu tablosu [17].

Veri Boyutu(Byte)	Veri Uzunluk Kodu			
	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

Mesaj içindeki bir diğer alan da tamamen bilgi içeren veri alanıdır (**Data Frame**). Bu bilgi en fazla 8 byte, bir başka değişle 64 bittir. Hat üzerinde yayınlanmaya MSB bitinden başlanır ve LSB bitine kadar devam eder.

Veri alanından sonra çevrimli fazlalık kodu gelmektedir (**CRC-Cyclic Redundancy Code**). Bu alan 16 bitten oluşmuştur ve iki parçadır. İlk 15 bit CRC serisinden (**CRC Sequence**), son bit ise CRC sınırlayıcıdan (**CRC Delimiter**) oluşur. CRC serisi, mesajın başlangıcından CRC alanına kadar yazılan bitlerin belli bir matematiksel işleme tabii tutulmasıyla elde edilir. CRC sınırlayıcı biti ise mesajı yazan modül tarafından hatta çekinik olarak yazılır. Oluşan CRC serisi ve gelen mesajın CRC alanına kadar olan bölümü ilgili modüller tarafından karşılaştırılır. Eğer bu karşılaştırma işlemi doğru ise mesajın devamı okunmaya devam eder ama eğer değilse çekinik olarak hatta basılan CRC sınırlayıcı biti hatayı yakalayan modül tarafından değiştirilir ve baskın bit olarak hatta basılır. Bu durum ilgili diğer bütün modüllere mesajın doğru olmadığını ve bir bozulmaya uğradığını bildirir. Bu durumda aynı mesajın gönderici tarafından tekrar hatta basılması istenir.

CAN hattı üzerinde yayınlanan her mesaj en az bir modül tarafından okunmak zorundadır. Aksi takdirde yayınlanmasına gerek yoktur. CRC alanından sonra gelen alındı bilgisi (**ACK-Acknowledge Field**) mesajın başka bir modül ya da modüller

tarafından alınıp alınmadığını belirtir. Bu alan alındı bilgisi bölümü (**ACK Slot-Acknowledge Slot**) ve alındı bilgisi sınırlayıcı (**ACK Delimiter-Acknowledge Delimiter**) bitlerinden oluşur. ACK slot biti mesajı yayınlayan modül tarafından baskın olarak hatta yazılır ve ACK alanının başladığını belirtir. ACK sınırlayıcı ise mesajı yayınlayan modül tarafından çekinik olarak hatta yazılır ve mesajı okuyan modül tarafından baskın hale getirilir. Bu durum mesajın karşı taraftan alındığını gösterir. Eğer hiçbir modül mesajı okumazsa ACK sınırlayıcı çekinik halde kalır. Bu da bir sorun olduğunu gösterir ve mesaj tekrar yayınlanır.

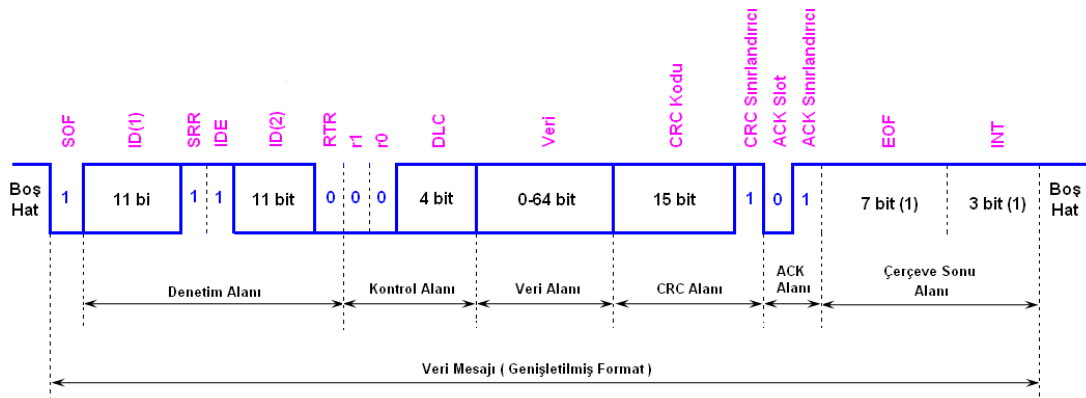
Yayınlanan bir mesajın bittiği paket sonuna yerleştirilmiş 7 bitten oluşan ve mesaj sonu bitleri (**EOF-End of Frame**) olarak adlandırılan alana bakılarak anlaşılır.

Genişletilmiş formatta bir veri mesajının yapısı

Genişletilmiş formattaki veya standart formattaki bir veri mesajının yapısı ve içinde kullanılan bitlerin görevleri denetim ve kontrol alanındaki değişiklikler dışında hemen hemen aynıdır. Genişletilmiş formatta denetim alanı 32 bittir. Şekil 2.9’da da görüldüğü gibi standart formattan farklı olarak ID numarası alanı 29 bite çıkmış, 11 bitlik ve 18 bitlik olmak üzere iki parça halinde yayınlanmıştır.

Bir başka değişiklik ise önceden kontrol alanı içerisinde yayınlanan ve mesajın hangi formatta olduğunu belirten IDE biti denetim alanı içersine taşınmış ve lojik 1 seviyesine çekilmiştir. IDE bitinin kontrol alanındaki eski yeri ise ayrılmış bit haline dönüştürülmüş ve bu formatta kullanılmamıştır.

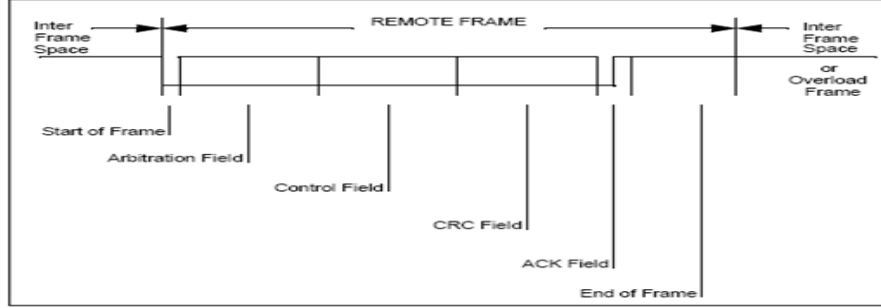
Denetim alanına standart formdan farklı olarak yedek uzak iletim istek biti (**SRR-Substitute Remote Request**) eklenmiştir. Bu bit veri mesajlarında lojik 1 seviyesindedir ve RTR bitinin yerini doldurmak için kullanılmıştır.



Şekil 2.9 : Genişletilmiş veri mesajı yapısı [19].

2.2.4.2 İstek mesajı

Veri talebinde bulunmak için gönderilen mesajdır. İstek mesajının veri mesajından farkı içerisinde veri alanı içermemesi ve RTR bitinin lojik 0 yerine lojik 1 olmasıdır. Hem standart formda hem de genişletilmiş formda alanlar ve bitlerin görevleri veri mesajınıninki ile aynıdır.



Şekil 2.10 : İstek mesajı yapısı [17].

2.2.4.3 Hata esajı

Herhangi bir modülün mesajın bozuk olduğunu tespit ettiği anda iletim ortamına gönderdiği mesajdır [20]. Bu sayede diğer modülleri de mesajın hatalı olduğu yönünde uyarır.

Her hata mesajı oluştuğunda modüller içindeki gönderme hatası sayıcısı (**TEC-Transmit Error Counter**) ve alım hatası sayıcısı (**REC-Receiver Error Counter**) değerleri değişmektedir. Bu sayıcıların içindeki değerlere göre hata mesajının türü ve modülün aktiflik durumu değişir. Çizelge 2.6’de gösterildiği gibi herhangi bir modülün içersinde tuttuğu TEC değeri 255’i aştığında modül kendini kapatır. Bunu çok sayıda hatalı mesaj yayınlayıp hattı meşgul etmemek için yapar.

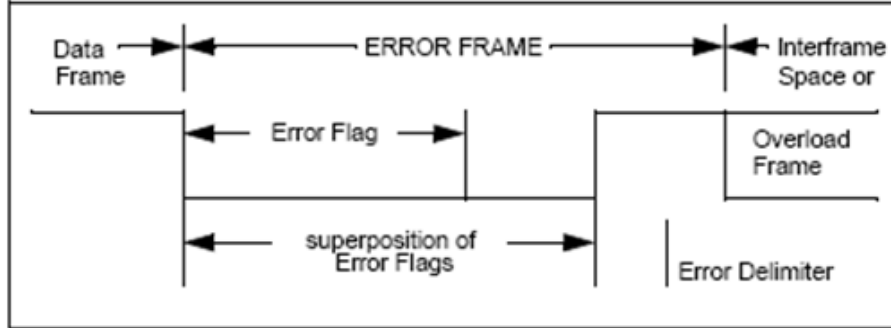
Çizelge 2.6 : Modül çalışma durumu.

Sayaç İçerikleri	Çalışma Modu
TEC=0 & REC=0	Normal çalışma modu
TEC<128 & REC<128	Aktif hata üretme modu
TEC>=128 & REC>=128	Pasif hata üretme modu
TEC>255	Modülün hatta kapatılması

Sayıcı içindeki değerler hata oluştuğunda artmasına rağmen bir müddet hata oluşmazsa veya belli bir sayıda doğru mesaj gelirse zamanla azalır.

Hata mesajı 2 bölümden oluşur:

- **Hata bayrağı:** 6 bittir. Bitlerin durumu aktif hata ya da pasif hata mesajına göre değişir.
- **Hata sınırlandırıcısı:** 8 bittir ve bu bitlerin tamamı çekiniktir.



Şekil 2.11 : Hata mesajı yapısı [17].

Aktif hata mesajı (**Active Error Frame**) ve pasif hata mesajı (**Passive Error Frame**) olmak üzere iki tip hata mesajı vardır. Bir modül hangi tipte hata mesajı yayınlayacağına TEC ve REC değerlerine bakarak karar verir.

- **Aktif Hata Mesajı (Active Error Frame):** Hata bayraklarının tümünün lojik 0 olarak yayınlandığı mesajdır.
- **Pasif Hata Mesajı (Passive Error Frame):** Hata bayraklarının tümünün lojik 1 olarak yayınlandığı mesajdır.

2.2.5 CAN protokolünün avantajları

- Merkezi bir birime ihtiyaç duyulmaz.
- Veriler yayınlanan noktadan birçok modüle eşzamanlı olarak iletilir (Multi-cast özelliği).
- Hatta aynı anda iki istasyon veri göndermeye kalktığında çakışmaya izin verilmez. Böylece verilerin bozulması önlenmiş olur.
- Hat topolojisi kullanıldığı için kablo miktarı oldukça azalır.
- Yeni modüller eklenmek istenildiğinde hatta paralel bağlanıldığından eski sistemin bozulması gerekmez.
- Fiziksel iletim ortamı basit ve esnektir [21].
- İletişim hızı 1 Mbps'a kadar çıkabilir.
- Güvenirliği yüksektir, arıza teşhisi kolaydır [21].

2.3 Araç Üzerindeki CAN Yapısı

Araç üzerinde 50 Kbps ve 500 Kbps hızlarında olmak üzere iki ayrı CAN hattı bulunur. Çizelge 2.1’de de görüldüğü gibi bu hatlar hızlarına göre isimlendirilmiş, yavaş olan B-CAN, hızlı olan diğer hat ise C-CAN olarak tanımlanmıştır. İki hat üzerinde de araçta belli görevleri üstlenmiş modüller bulunmaktadır. Bu modüllerin beslemesi 12 Volt’tur. Ayrıca hem B-CAN üzerinde hem de C-CAN üzerinde yayınlanan mesajların ID uzunlukları 11 bittir (standart format).

2.3.1 B-CAN

Genellikle elektronik konfor ünitelerinin bağlandığı, gerçek zamanlı olmayan ya da kritik olmayan verilerin aktığı, düşük hızlı CAN (**Low-Speed CAN**) olarak da adlandırılan iletişim ortamıdır. Bu yüzden güç tasarrufu moduna da sahiptir. Bu hatta bağlı herhangi bir modül bir süre kullanılmadığında osilatörlerini kapatır ve uyku moduna geçer. CAN’dan tekrar mesaj geldiğinde ise uyku modundan çıkar ve CAN hattına tekrardan bilgi basmaya devam eder. Böylece enerji tasarrufu sağlanmış olur[10].

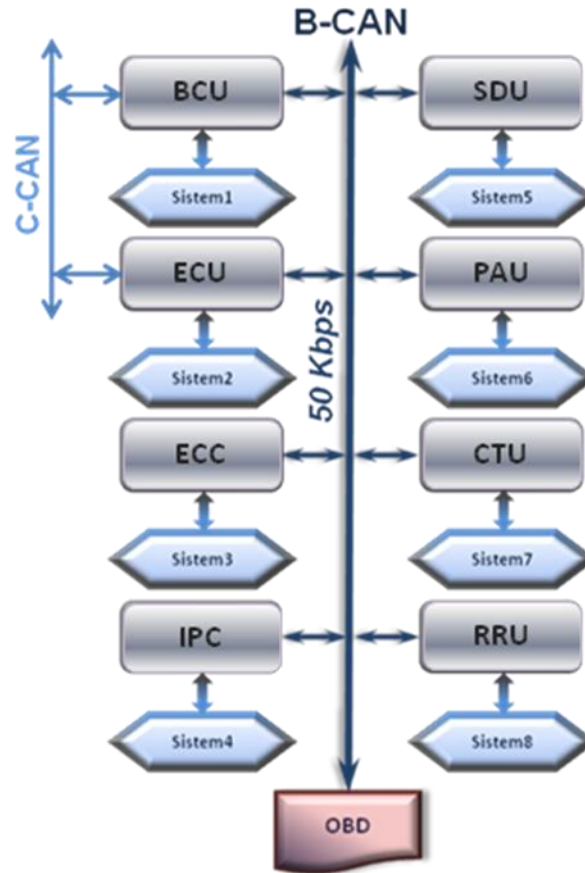
Çizelge 2.7 : B-CAN modülleri.

Modül Adı	Modül Tanımı
BCU	Body Control Unit
ECU	Engine Control Unit
ECC	Electronic Climate Control
IPC	Instrument Panel Cluster
SDU	Sensing and Diagnostic Unit
PAU	Park Assistant Unit
CTU	Convergence Telematic Unit
RRU	Radio Receiver Control Unit

Şekil 2.12’deki blok diyagramda da görüldüğü B-CAN’e bağlı 8 adet modül bulunmaktadır. Her modül programlandığı gibi bağlı olduğu sistemi yönetmek için sensörlerden aldığı verileri ve B-CAN’dan okuduğu bilgileri kullanır. Ayrıca diğer modüllerin gerektiğinde kullanabilmeleri için kontrol ettiği sistemle ilgili verileri hatta basar. Örneğin elektronik klima kontrol ünitesinin görevi (**ECC-Electronic Climate Control**) araç içersindeki ısıtıcı ve soğutucuları kontrol etmek ve ile ilgili verileri CAN hattına yazmaktır. Aynı zamanda araç çalışırken motorun zorlanmaması için klimanın kısa bir süre kapatılması ECC tarafından gerçekleşir.

İşlem sırası şu sırayı takip eder:

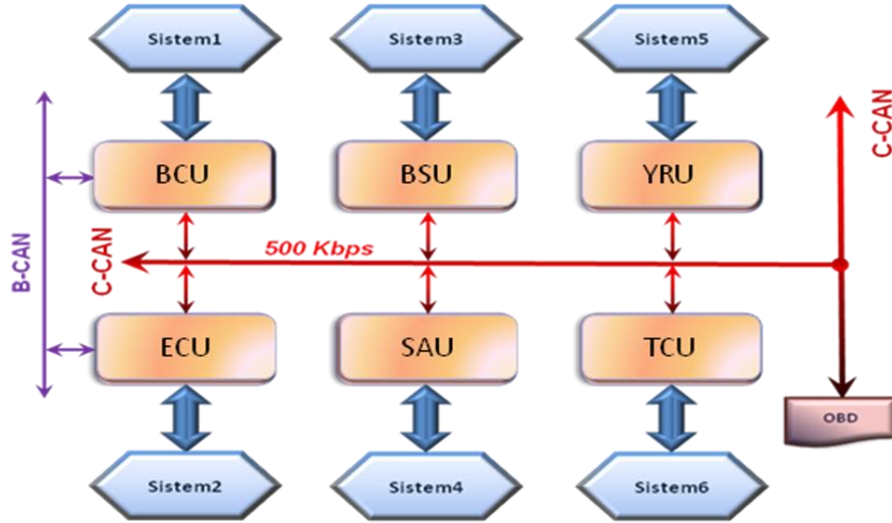
- Kontak çevrildiğinde ilgili modül aracın çalıştırılmak istendiğini hat üzerine yazar ve diğer modüller hazır olana kadar bekler.
- ECC modülü B-CAN üzerinden aracın çalıştırılmak istendiğini okur ve soğutucuyu, ısıtıcıyı ve/veya gerekli diğer birimleri kapatır.
- ECC kapatma işlemi gerçekleştiğinde B-CAN hattı üzerinden klima sistemini istenilen konuma getirdiğini ve beklemede olduğunu diğer modüllere bildirir.
- Daha sonra araç çalıştı bilgisi gelene kadar beklemeye devam eder.
- Motorun çalışmasından sorumlu modül klimanın ya da diğer güç tüketen ünitelerin kapandığını B-CAN'den ve C-CAN'den öğrendiğinde motora çalış komutunu yollar ve motorun çalışıp çalışmadığını kontrol eder.
- Motorun çalıştığını algılayan modül B-CAN hattına bu bilgiyi yazar.
- Motorun çalıştığı işaretini okuyan ECC modülü klima, soğutucu vb. birimlerini tekrar aktif hale getirir.



Şekil 2.12 : B-CAN hattının bağlantı şeması.

2.3.2 C-CAN

Her ne kadar B-CAN üzerinde araca ait birçok bilgi bulunsa da, hız kontrolörü için gerekli verilerin tümü C-CAN hattında yer almaktadır. Çünkü C-CAN hattında, B-CAN’de olduğu gibi konfor ya da periyodik zamanı önemseyen veriler değil, gerçek zamanlı uygulamalara ait ya da zaman açısından kritik önem taşıyan veriler yer alır.



Şekil 2.13 : C-CAN hattının bağlantı şeması.

C-CAN yüksek hızlı CAN hattıdır ve hızı 1 Mbps’a kadar çıkabilir. Ancak Şekil 2.13’de de görüldüğü gibi üzerine bağlı modüller 500 Kbps hızında haberleşirler. Çünkü yüksek hızlarda iletimi sağlayan burgulu-çiftli kablo üzerinde oluşan elektromanyetik gürültünün etkilerini azaltmak ve güvenli veri transferini sağlamak sisteme büyük bir maliyet getirecektir.

Çizelge 2.8’de görüldüğü gibi bu hatta 6 adet modül bağlanmıştır ve bu modüller 12 V gerilim ile beslenmişlerdir. Motor kontrol modülü (ECU-Engine Control Unit) ve gövde kontrol modülü(BCU-Body Control Unit) gibi birimler bağlı oldukları sistemler için çok gerekli olduğundan her iki haberleşme hattına da bağlıdır.

Çizelge 2.8 : C-CAN modülleri.

Modül Adı	Modül Tanımı
BCU	Body Control Unit
ECU	Engine Control Unit
BSU	Brake System Control Unit
YRU	Yaw Rate Control Unit
SAU	Steering Angle Control Unit
TCU	Transmission Control Unit

Aracın C-CAN hattında güvenlik ve yürüme aksamaları için gerekli birçok önemli veri akmakta ve bunların büyük bir bölümü periyodik olarak hat üzerinde yayınlanmaktadır. Gereksinimlere göre periyot zamanları farklılık gösterebilir. Örneğin araç hızını içeren mesaj her 100 ms’de bir BCU modülü tarafından C-CAN’e yazılırken, motor torku 10 ms’de bir yazılır.

C-CAN üzerinde 6 modül bulunmasına rağmen periyodik tipte 18 farklı mesaj üretilmektedir. Üretilen bu paketlerin içersinde mesaj sayısından daha fazla veri vardır. Çünkü hattaki her mesaj en az bir veri taşır. Bu verilerden bazıları şunlardır:

- Aracın boyuna ivmesi
- Aracın yanal ivmesi
- Aracın açısal ivmesi(yaw rate)
- Motor devir hızı
- Motorun ürettiği tork
- Motor suyu sıcaklığı
- Araç hızı
- Tekerlerin ayrı ayrı hızları
- Vites dişlisi konumu
- Gaz pedalı konumu
- Fren pedalını durumu(basılı-basılı değil)
- Direksiyon açısı
- Direksiyon dönüş hızı
- Dış ortam sıcaklığı
- Gaz kelebeği açıklığı(throttle)

Sürekli mesajların aktığı CAN hattından bir veri okunmak isteniyorsa ilk olarak istenilen verinin hangi mesajda yer aldığının bilinmesi gerekir. Daha sonra o mesaj içersindeki konumu ve veriye ait diğer bilgilerin(ofset değeri, çözünürlük vb. gibi) incelenmesi şarttır. Örnek olarak Şekil 2.14’de verilen 0x62 (00001100010) ID numaralı mesajı ve içersinde barındırdığı veri alanından bir kesitin özelliklerini inceleyecek olursak;

ID: \$062	Type: P	Period: 20 ms	Size: 8 byte	Sender: ECU	
Signal Name	Event trans.	Bit position msb-lsb	Size	Property	Value
EngineWaterTemp	No	39-32	8	Accuracy Range High Range Low Unit Signal Age Resolution Offset Startup / Default Not Valid	± 2 °C at 120 °C 215 °C -40°C °C 130 ms 1°C -40 °C 00 hex -40 °C

Şekil 2.14 : Mesaj içindeki örnek veri.

Mesajın özellikleri:

- Standart formatta bir data mesajıdır.
- Periyodik mesaj tipindedir ve tekrar zamanı 20 ms'dir.
- ECU(Engine Control Unit) tarafından üretilir.
- Üzerinde 8 byte bilgi taşır.

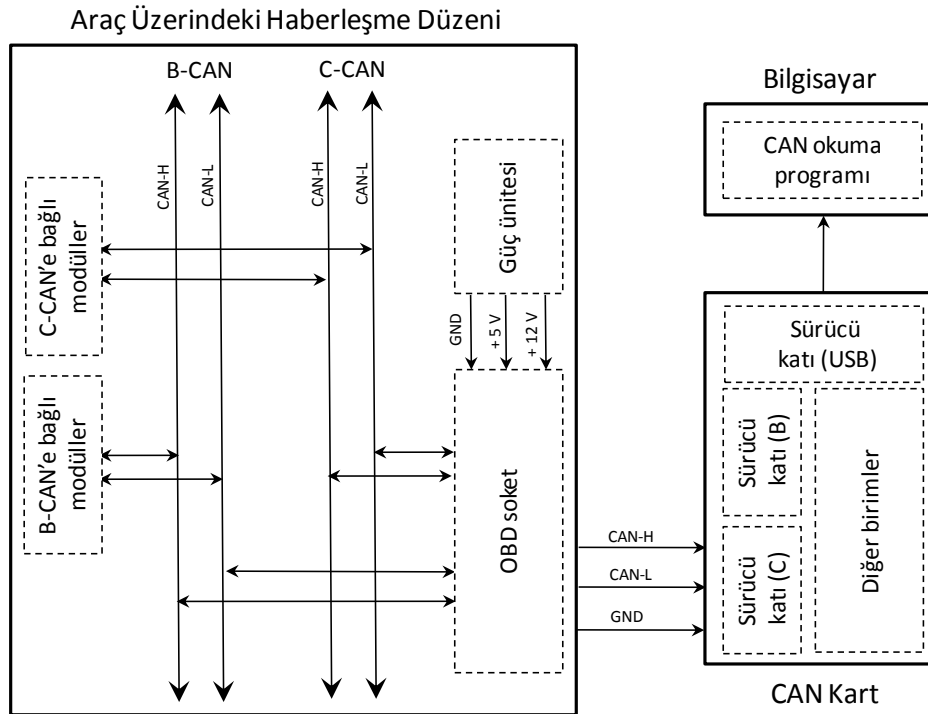
Verinin özellikleri:

- Motor suyu sıcaklığı ile ilgili bilgileri barındırmaktadır.
- Bu veri bir olay gerçekleştiğinde üretilmez, periyodik olarak sıcaklık ölçülür ve mesaj içersine yerleştirilerek C-CAN hattına yazılır.
- Motor suyu sıcaklığının 64 bitlik veri alanı içersideki yeri 39. ve 32. bitler arasındadır.
- 39. bit bilginin MSB biti, 32. bit ise LSB bitidir.
- Motor suyu sıcaklığı toplam 8 bitlik bir veridir.
- Elde edilen sayısal değerin birimi °C'dir.
- LSB bitindeki her değişim, diğer bir deyişle çözünürlük 1 °C'dir.
- Sinyal gecikmesi maksimum 130 ms'dir.
- Bu verinin mesaja yazılabilecek en alt değeri -40 °C, en üs değeri ise 215 °C'dir.
- 120 °C'deki ölçüm hatası ± 2 °C'dir.
- Veri hekzadesimal olarak 00h(desimal 0) ile FFh(desimal 255) arasında değişir ve -40 °C olan başlangıç değerine(ofset) karşılık gelen sayısal değer 00h'dir.
- Bu veriye ait bir geçerlilik biti (valid) vardır. Motor suyu sıcaklığı -40 °C altında ölçüldüğünde geçerlilik biti bu verinin geçerli olmadığını C-CAN üzerinde yayınlacaktır.

Bu çerçevede tez çalışmasında yapılacak olan hız kontrolörü için gereken tüm verilerin özellikleri araç üreticisi firma tarafından sağlanan kataloglar yardımıyla yukarıdaki gibi belirlenmiştir.

2.3.3 CAN hattına erişim

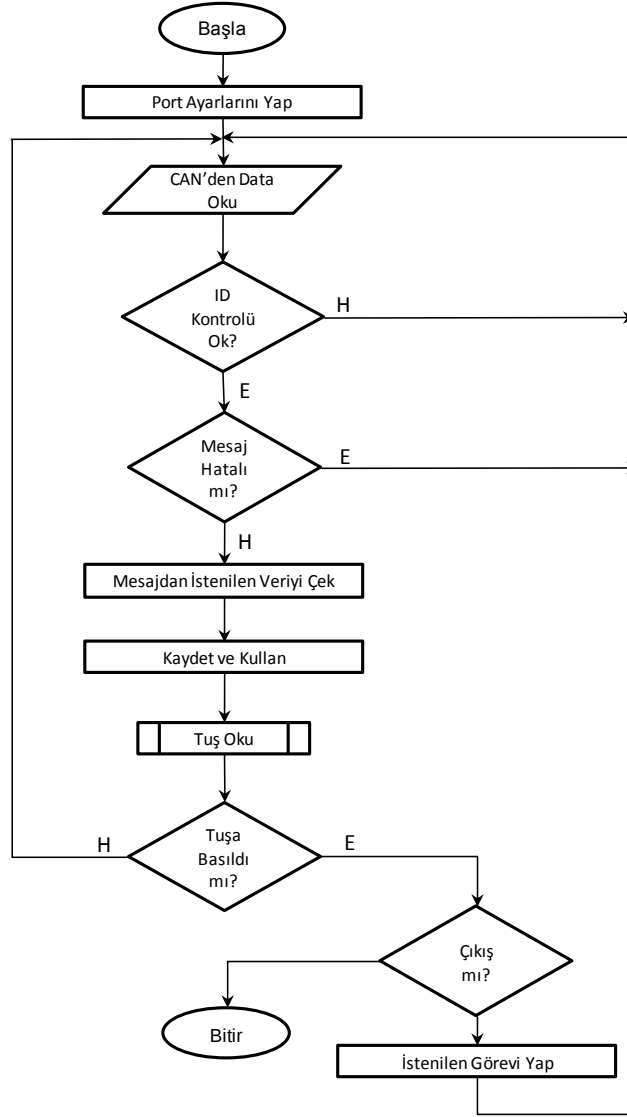
Araca ait CAN hattından data okuyabilmek için öncelikle C-CAN'e ait CAN-High ve CAN-Low uçları ile birlikte aracın GND'sine elektriksel bağlantı yapılmıştır. Tüm bu sinyallere iç ortamda bulunan ön sigorta paneline girilerek erişilmiştir. Daha sonra bu iletim hatları Kvaser marka USBcan2 HS/LS model [22] CAN analizöre, diğer bir deyişle CAN karta girilmiştir.



Şekil 2.15 : CAN için oluşturulan bağlantı yapısı.

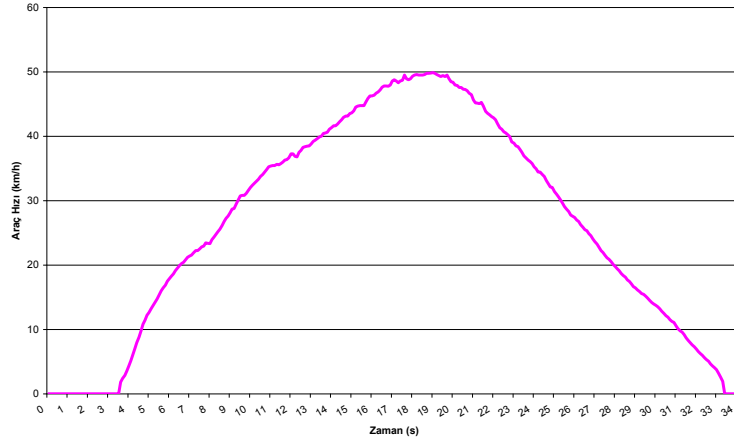
CAN kartın bu noktadaki görevi haberleşme ağına bağlı bir modül gibi davranarak, akan verileri kullanılabilir duruma getirip USB çıkışından bir sonraki elemana göndermektir. Burada veriyi kullanılabilir duruma getirmekten kasıt CRC kodunu kullanarak hatalı mesajları tespit etmek ve diğer modülleri uyarmaktır. Tasarlanan yapıda son olarak CAN kart USB arayüzü ile bilgisayara bağlanmış ve C-CAN'den veri okumak için gerekli donanımsal yapı tamamlanmıştır.

CAN hattına erişimin sağlanıp sağlanmadığını test etmek için donanımın yanında “CAN okuma programı” adında bir yazılım geliştirilmiştir. Bu yazılım gerçek zamanlı olarak çalışmakta olup, istenilen mesajları ID numarasına göre filtreleyip kaydedebilmektedir. Yazılıma ait akış diyagramı Şekil 2.16’da gösterildiği gibidir.

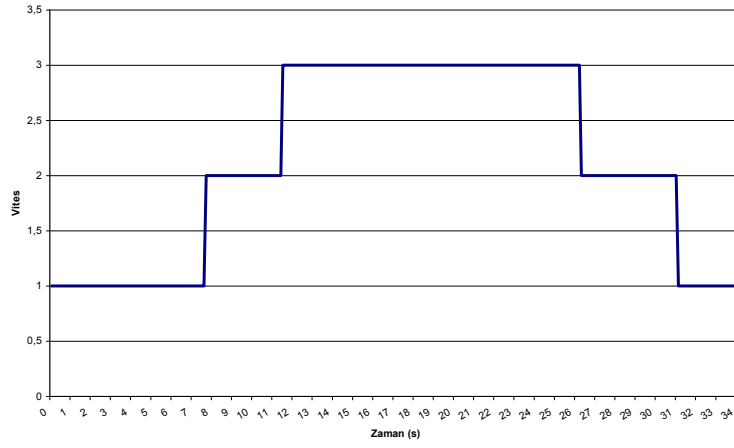


Şekil 2.16 : Akış diyagramı.

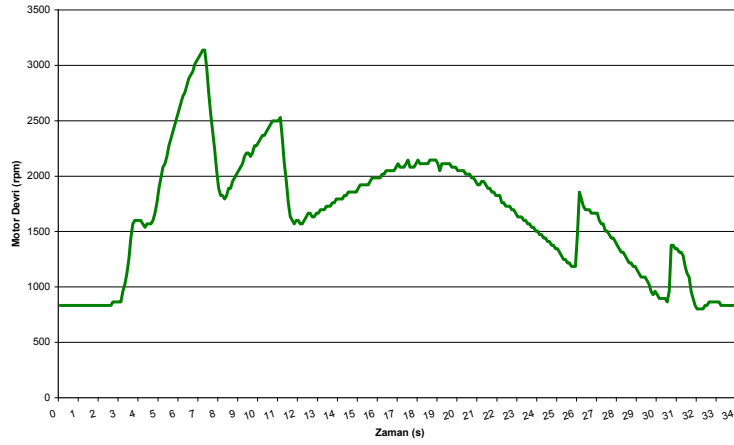
CAN’e erişim için oluşturulan sistemi kullanarak deneysel ortamda araç hareketliyken CAN’den bir takım veriler okunmuş, elde edilen verilere ait grafikler Şekil 2.17, Şekil 2.18 ve Şekil 2.19’de gösterilmiştir.



Şekil 2.17 : Araç hızı.



Şekil 2.18 : Vites.



Şekil 2.19 : Motor devri.

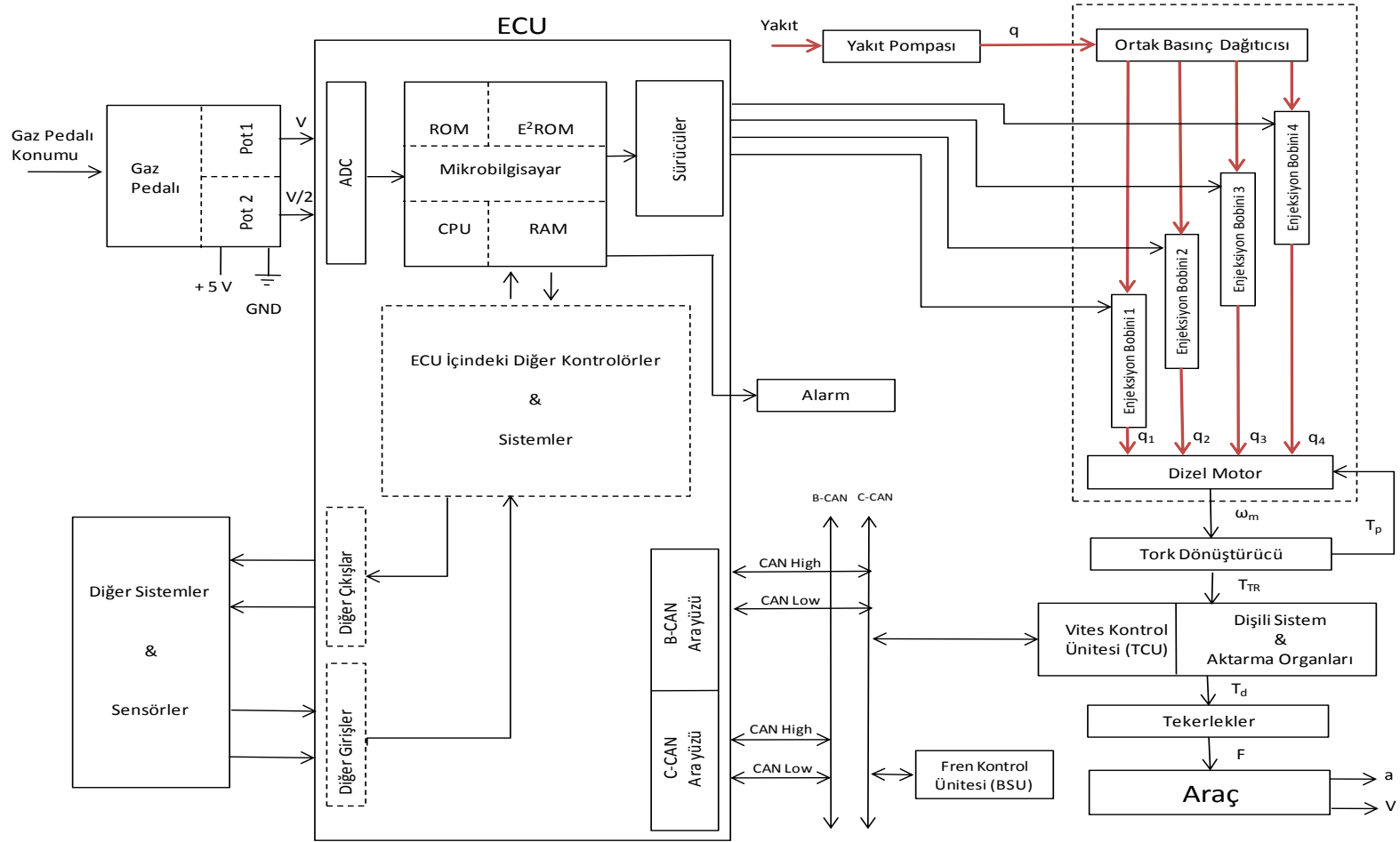
3. OTONOM OTOMOBİLİN GAZ ve FREN SİSTEMİNİN İNCELENMESİ

Otomobillerde ivmelenme sistemi aracın hızlanmasını, fren sistemi ise aracın güvenli bir şekilde yavaşlamasını ve durmasını veya duran bir aracın sabit kalmasını sağlar. Bu bölümde projede kullanılan otomobilin hızlanmasının nasıl gerçekleştiği irdelenmiş ve ivmelenme sisteminin elemanları tek tek ele alınmıştır. Sistemin özellikleri belirlendikten sonra hızlanmanın dışarıdan bir elektrik sinyaliyle nasıl kontrol edileceği tespit edilmiş, ardından fren sisteminin yapısı ele alınmıştır. Fren sistemine müdahale edebilmek için lineer motor içeren bir mekanik aksam tasarlanmıştır. Her iki sistemi de kontrol edebilmek için bilgisayarla haberleşebilen elektronik bir kart dizayn edilmiş ve programlanmıştır. Kart iki sisteme de uygulayacağı sinyal bilgisini bilgisayar içinde yer alan yazılımdan almaktadır. Bu şekilde çeşitli deneyler yapılmış ve bölüm sonunda bu deneylerden bazı örnekler verilmiştir.

3.1 İvmelenme

Kullanıcıdan alınan referans değere göre aracın boyuna hareket etmesini sağlayan düzeneğe ivmelenme sistemi denir. Hızlanma sistemi olarak da adlandırılır. Bu yapı proje için kullanılan araçta gaz pedalı, ECU (Engine Control Unit), dizel motor, dizel motora bağlı enjeksiyon sistemi, tork dönüştürücü, dişli ve aktarma organları gibi öğelerden oluşmaktadır.

Şekil 3.1 incelenecek olursa, meyilsiz bir yolda ivmelenme gaz pedalına sürücünün müdahalesi ile başlar. Pedal, içinde barındırdığı elektronik devre sayesinde konumuyla doğru orantılı olarak iki adet analog sinyal üretir ve ECU'ye gönderir. Gelen bu işaretler önce kuantalanarak dijital hale getirilir ve ECU'de var olan yazılım ortamında değerlendirilir. Bu işlem sonucunda enjeksiyon bobinlerinin tetikleme anları ve tetikleme süreleri hesaplanmış olur. Yüksek sıvı basıncı altındaki enjeksiyonların selenoidleri ECU'nun gönderdiği elektrikselsinyallerle belli sürelerde açılır ve kapanır. Böylece bir an için elektrikleştirilen valflerin içindeki sıvı yakıt motorun silindirlrine doğru püskürtülmüş olur. Dizel motor bu noktada



Şekil 3.1 : İvmelenme sistemi genel yapısı.

enjektörlerin silindirlere aktardığı yakıtı kimyasal enerjiden mekanik enerjiye dönüştürür ve püskürtülen yakıt miktarına göre çıkışında bir tork meydana getirir. Oluşan torka ve motor çıkışındaki yüke bağlı olarak bir açısal hız da ortaya çıkar. Daha sonra tork dönüştürücüye gelen bu güç yumuşatılarak vites kutusuna gönderilir. Vites durumuna göre tork ve açısal hız dönüşümleri bu aşamada gerçekleşir, enerji aktarma organları vasıtasıyla diferansiyel dişlisinden de geçerek tekerlekler kadar iletilir. Yerle temas halinde olan tekerlekler torku dikey bir kuvvete çevirir ve aracın hızlanmasını sağlar.

3.1.1 Gaz pedalı

Gaz pedalı mekanik ve elektronik olmak üzere iki kısımdan oluşmaktadır. Mekanik kısım kullanıcının kuvvetini uyguladığı bölümdür. Girilen kuvvet sonucu pedal konum değiştirir ve böylece hızlanma için gerekli referans hareket alınmış olur. Ayrıca bu kısmın içinde, sürücü pedala baskı uygulamayı bıraktığında ya da yaptığı baskıyı azalttığında pedalın başlangıç konumuna dönmesini sağlayan yaylı bir mekanizma vardır. Bu mekanizmanın diğer bir görevi de kullanıcının uyguladığı kuvvette ters bir kuvvet uygulamak ve pedalın boşta olmadığı hissiyatını vermektir. Bu şekilde pedal sertliği de belirlenmiş olur.

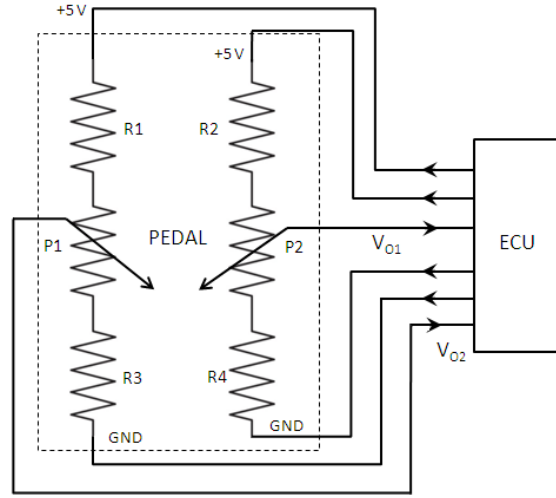


Şekil 3.2 : Gaz pedalı ve iç yapısı.

Gaz pedalının elektronik kısmı ise pedal pozisyonunu belirlemede kullanılan elektronik algılayıcılardan meydana gelir. Algılayıcı olarak potansiyometre kullanılmıştır ve güvenlik amacıyla bunlardan iki adet mevcuttur.

Gaz pedalının elektronik devre şeması Şekil 3.3’de verilmiştir. Görüldüğü gibi bir sonraki sistem olan ECU ile pedal arasında 6 adet iletim hattı bulunmaktadır. İki algılayıcının da besleme gerilimleri olası bir güvenlik problemi nedeniyle algılayıcılara ayrı ayrı getirilir. Bu yüzden iletim hatlarından dördü, iki adet GND ve

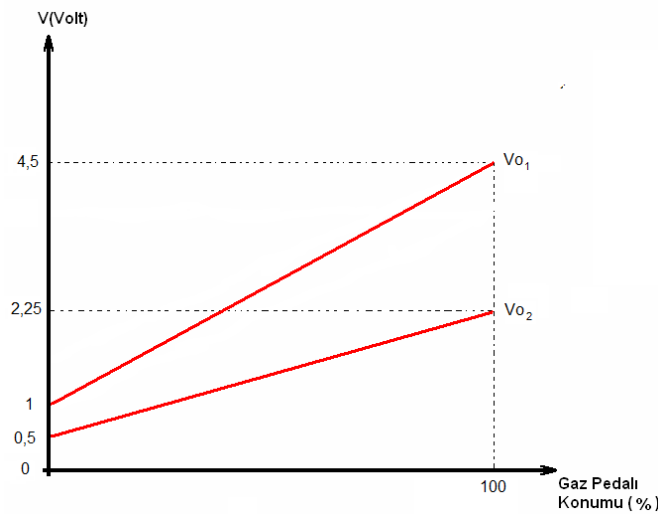
iki adet (+)5 Volt olmak üzere besleme hattıdır. Diğer iki hat ise V_{O1} ve V_{O2} olarak adlandırılan, pedaldan ECU'ye dönen analog sinyalleridir.



Şekil 3.3 : Gaz pedalının elektronik yapısı.

Pedal konum değiştirdikçe V_{O1} ve V_{O2} sinyalleri Şekil 3.4'deki gibi değerler almaktadır. Pedal pozisyonu ile sinyaller arasındaki ilişki doğrusaldır. Şekil 3.4'de de görüldüğü gibi iki işaretin de ofset değerleri farklıdır. Pedala hiç basılmadığı andaki başlangıç gerilim değerlerinin (+) 1 V ve (+) 0.5 V'dur. Pedala tam basıldığında ise iki işaretin en yüksek gerilim değerleri (+) 4.5 V ve (+) 2.25 V'a ulaşır.

ECU'ye giden bu iki çıkış üzerindeki gerilimlerin biri birine oranının pedal hangi konuma getirilirse getirilsin yarı yarıya olduğu görülmektedir. Buna ek olarak algılayıcıların ürettiği sinyallerin minimum ve maksimum gerilim değerlerinin pedal hangi konumu alırsa alsın besleme gerilimleri ile asla eşit olmadığı da görülmektedir.



Şekil 3.4 : Gaz pedalı çıkış sinyalleri değişimi.

Gerilim değerlerinin birbirine eşit olmaması,

- Potansiyometrelerden birinin bozulması ama hala gerilim üretmeye devam etmesi,
- İki referans değeri taşıyan iletim hatlarının birbiri ile kısa devre olması,
- İki referans değeri taşıyan iletim hatlarından yalnız birinin GND veya (+) 5 V ile kısa devre olması,
- İki referans değeri taşıyan iletim hatlarının aynı anda(ikisinin birden) GND veya (+) 5 V ile kısa devre olması,
- Besleme hatlarından birinin veya birkaçının kopması,

gibi güvenlik açısından ortaya çıkabilecek birçok istemeyen durumun merkezi birim tarafından fark edilmesini sağlar.

3.1.2 Motor sistemi

Motor ünitesi elektronik sinyallerle kontrol edilen, kimyasal enerjinin mekanik enerjiye dönüştürüldüğü sistemdir. Genel hatlarıyla yakıt miktarını kontrol eden enjeksiyon sistemi, enerji dönüşümünün yapıldığı dizel motor ve sistemin tümünü yönetip, kontrol eden ECU modülünden oluşur.

3.1.2.1 ECU (Engine control unit)

ECU (Motor Kontrol Ünitesi) sensörler aracılığıyla birçok noktadan gelen verileri işleyen ve dizel motorun değişen koşullarda en verimli şekilde yönetimini sağlayan merkezi birimdir. Mikroişlemci mimarisiyle tasarlanmış olup kendine ait işlemcisi, RAM'i, ROM'u, E²ROM'u, giriş-çıkış ve diğer çevresel birimleri vardır. Özetle kullanıcı ara yüzü olmayan bir mikrobilgisayardır.

ECU motor kontrolünü gaz pedalı, dizel motor, tork dönüştürücü gibi elemanlar üzerinde bulunan sensörlerden aldığı sinyallere bakarak gerçekleştirir. Dizel motorun devrini arttırmak için bakacağı referans değer gaz pedalından okuduğu 2 analog işaretin değerleridir. Bu analog sinyaller, ECU içersindeki işlemci tarafından kullanılabilmesi için ADC (Analog-To-Digital Converter) birimi tarafından sayısal hale getirilir. Motor kontrol ünitesi pedal konum sinyallerine göre enjeksiyon bobinlerinin tetikleme zamanlarını ve tetikleme sürelerini belirler. Bu belirlediği sinyalleri sürücü katından geçirerek enjektörlere gönderir ve onların bobinlerini elektrikleştirerek açılıp kapanmasını kontrol eder. Araç motoru 4 silindirli

olduğundan enjeksiyon sisteminde de 4 adet de enjektör bulunmaktadır ve hepsi ECU'nün kontrolü altındadır.

Gaz pedalından okunan değere karşılık ECU tarafından hesaplanan gaz pedalı oranı C-CAN haberleşme hattına da gönderilir. Çünkü gaz pedalı basılma oranı bilgisi fren sistemi kontrol modülü (BSU) ve vites kontrol modülü (TCU) gibi birimler tarafından da kullanılmaktadır.

ECU'nün tek görevi gaz pedalının konumuna göre enjektörler arasındaki bağlantıyı sağlamak değildir. Gaz pedalında meydana gelen istemeyen durumların tespiti, bu gibi durumlarla karşı karşıya kalındığında gerekli önlemlerin alınması da ECU'nün görevlerindendir. İstenmeyen durumlar karşısında alarm jeneratörünü çalıştırmak, enjektörleri kapatmak (gaz kesmek), CAN hattı üzerinden diğer birimleri uyarmak alınacak önlemlerden bazılarıdır.

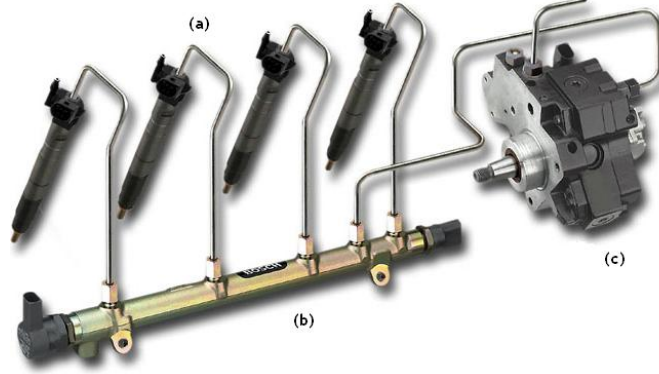
ECU'nün bunların dışında başka önemli görevleri de vardır. Özellikle kontrol ettiği ya da gözlemlediği alt sistemlerden sürekli olarak sensörler aracılığı ile veri alır. Dizel motorun devir hızı, yakıt durumu, atmosfer basıncı, motor suyu sıcaklığı vb. gibi birçok parametre sensörler tarafından ölçülür ve ECU'ye aktarılır. ECU gelen tüm bu parametrelere göre program akışını belirler ve gerekli çıkış sinyallerini üreterek kendine bağlı alt sistemleri kontrol eder. Ayrıca bu verileri diğer modüllerin kullanabilmesi için sınıflandırarak B-CAN ya da C-CAN hattına yazar.

3.1.2.2 Enjeksiyon sistemi

Enjektörler, içinden geçen ve dizel motora giden yakıt akışını kontrol etmekte kullanılan aktüatörlerdir. Her enjektörün girişinde yüksek basınçlı sıvı yakıt bulunmaktadır. Bu sıvı, yakıt pompası tarafından sağlanır ve sistemdeki tüm enjektörlerin bağlı olduğu ortak basınç dağıtıcısına gelir. Aynı zamanda sıvı basıncının sabit bir seviyede tutulması buradaki regülâtörler tarafından gerçekleştirilir. Ortak basınç dağıtıcısına bağlı her hangi bir enjektörün içindeki selenoid valf elektriksel olarak tetiklendiğinde o enjektöre bağlı sıvı hattı kısa bir süre açılır ve yakıt geçişine izin verilir. ((a) selenoid valf, (b) ortak basınç hattı, (c) yakıt pompası.)

Enjeksiyonun tetikleme işlemi ECU'den gelen sinyalle sağlanır. Tetikleme süresi milisaniyeler mertebesinde ve bu fazla yakıt geçişini engeller. Sıvının basıncının yüksek olmasından ve enjektör çıkışının yapısından dolayı yakıt püskürtülerek bağlı

bulunan silindire aktarılır. Tetikleme süresi her ne kadar kısa da olsa periyotlar çok siktir. Aynı zamanda motorun dönüş hızı bu periyot sıklığıyla ayarlanır.



Şekil 3.5 : Enjeksiyon sistemi.

Bir enjeksiyon sisteminde motorda bulunan silindir adedi kadar enjektör bulunur. Projede kullanılan araç motorunda 4 adet silindir bulunduğu için, 4 ayrı enjektöre sahiptir.

3.1.2.3 Dizel motor

Kimyasal enerjinin mekanik enerjiye dönüştürüldüğü elemandır. Projede kullanılan araç üzerinde dizel motor bulunmaktadır ve özellikleri Ek-B’de verilmiştir.



Şekil 3.6 : Dizel motor.

Enjektörlerden motorun silindirlerine gelen yakıt burada krank milinin pistonla verdiği hareket nedeniyle sıkışır ve ısınır. Yeterli basınca ve ısıya ulaştığında parlar. Bu şekilde silindir içersindeki basıncın aşırı yükselmesi ile silindire ait piston itilir. Böylece mekanik enerji elde edilmiş olur. Tüm pistonların aynı işlemi sırası ile yaptığı düşünülürse krank milinin yapısından da kaynaklı dairesel harekete sahip bir

güç ortaya çıkar. İstenen hız değerine ulaşmak için ayakla müdahale edilerek pedaldan başlayan istek, burada tork olarak ve motor hızı olarak ortaya çıkar. Daha sonra elde edilen bu tork otomatik şanzımana aktarılır.

3.1.3 Otomatik şanzıman

Motordan gelen gücün kullanıcının isteği doğrultusunda belli oranlarda değiştirilerek bir sonraki sisteme aktarılmasını sağlayan alüminyum muhafazaya sahip olan parçadır. Proje aracında olduğu gibi önden çekişli araçlarda otomatik şanzıman kendi içersinde tork çevirici, yağ pompası, planet dişli grubu, hidrolik kontrol sistemi gibi birimler barındırır ve TCU(Transmission Control Unit) tarafından kumanda edilir. Motordan gelen güç şanzımana tork dönüştürücünün bulunduğu kısımdan girer, planet dişli grubun çıkarak diferansiyele aktarılır.

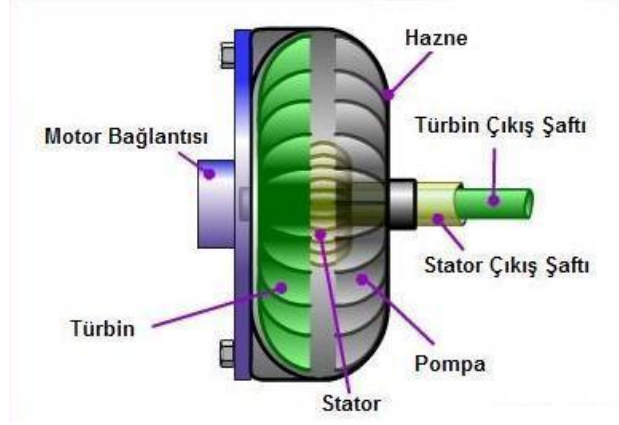
3.1.3.1 Tork dönüştürücü

Tork dönüştürücü, otomatik şanzımana sahip araçlarda klasik debriyaj kavrama sistemi olmadığı için motor ile şanzıman arasındaki bağlantıyı sağlayan ve vites boşta değilken araç durduğunda bu bağlantıyı kesen elemandır. Motorun teklememesi ve daha konforlu bir sürüş için motor torkunun devir ile doğru orantılı olarak planet dişli grubuna aktarılması, yüksek devirlerde ise motor ile dişli grubu arasında sabit bir bağlantı kurulması görevleri de tork dönüştürücüye aittir. Genellikle otomatik şanzımanın içinde yer alır ve 3 ana parçadan oluşur. Bunlar:

- Pervane (impeller)
- Türbin
- Stator



Şekil 3.7 : Türbin-stator-pervane[23].



Şekil 3.8 : Tork dönüştürücü[23].

Dönüştürücünün içi şanzımanda yer alan hidrolik bir pompa tarafından basılan basınçlı yağ ile doludur. Pompa tahrik gücünü iki kulakçık vasıtasıyla tork dönüştürücünden alır. Böylece motor hızı arttıkça hazne içindeki sıvının basıncı da artar. Basıncın devire bağlı olarak değişmesi tork dönüştürücüye kendi özelliklerini katmaktadır.

Hem pervane, hem türbin, hem de stator içindeki basınçlı sıvıya yön vermek ya da sıvının akışından faydalanmak için spiral şekline kanatçıklara sahiptir. Pervane motor eksenine, diğer bir deyişle krank miline bağlıdır. Türbin ise şanzımana bağlıdır. Stator ikisi arasında bir milde hareket eder.

İlk olarak motordan gelen tork pervaneyi döndürür. Devir arttıkça içindeki yağa etki eden pervanenin kanatları merkezkaç kuvveti ile yağı türbin kanatçıklarına doğru fırlatır. Sıvının türbine ait kanatçıklara çarpmasıyla türbin döner ve bağlı olduğu şanzımana gücü aktarır.

Stator ise pervane ile türbin arasında bir destek miline takılıdır. Bir avare dişli sayesinde krank mili ile aynı yönde boşa döner, aksi yönde ise kilitlenir. Pervane ve türbin arasına stator konulmadığında dönen sıvı pervaneye ait kanatçıklara dönüşün ters yönünde çarpıp, motordan gelen torkun bir kısmını yok edecektir. Statorun görevi ise üzerindeki kanatçıkları kullanarak türbin kanatlarına çarpıp tekrar geri dönen sıvıya yön vererek pervanenin dönüş yönü ile aynı hale getirmektir. Böylece dönen sıvının bozucu etki yapmamasının yanı sıra, sıvının kinetik enerjisi motor torkunun yanı sıra pervaneye pozitif etki yapar. Tüm bu aşamalarda stator dönmez. Çünkü dönen sıvının statorun kanatlarına çarptığında onu krank mili ile ters yönde döndürmeye çalışır. Bu durumda avare dişli buna izin vermez ve onu hareketsiz bir

biçimde kilitler. Böylece hazne içinde sıvıya yön verme işlemi gerçekleşmeye devam eder.

Tork dönüştürücü hidrolik bir yapıya sahip olduğu için pervane ve türbin dönüş hızları çoğunlukla farklılık gösterir. Ancak yüksek devirlere çıkıldığında pompanın da sağladığı yüksek basıncın etkisiyle pervane ve türbinin devirleri eşitlenir. Bu durumda türbine çarpan yağ krank mili ile aynı yönde dağılarak eski yörüngesini izlemez. Bunun yerine statorun kanatlarının arka taraflarına çarpar. Böylece stator krank mili ile aynı yöne dönmeye başlar ve devreden çıkar.

Statorun dönmeye başladığı ana kavrama noktası denir. Tam bu noktada güç kaybını ve yakıt tüketimini azaltmak için pervane ile türbin arasında sabit bir bağlantı oluşturan kilitli bir kavrama geliştirilmiştir. Stator dönmeye başladığında selonoid bir valf vasıtasıyla kilitli kavrama devreye sokulur. Bu andan itibaren motordan gelen tork şanzıman içerisinde bulunan planet dişli grubuna tam olarak aktarılır.

Motor rölantide çalıştığında ise pompanın da yavaş dönmesinden de kaynaklı içerdeki sıvı basıncı çok düşüktür. Bu etki pervanenin de yavaş dönmesiyle türbine giden gücü oldukça azaltır. Fakat yine de bu esnada planet dişli grubuna ufak da olsa bir güç iletilir. Bu güç duran bir aracın hareket etmesini ya da hareket eden bir aracın fren yapılamaması halinde durmamasına neden olabilir. Bunu engellemek için vitesi boşa almak veya aracı fren durumunda tutmak gerekir. Bu özellik her ne kadar bir dezavantaj gibi görünse de, klasik debriyaj kavrama sisteminde olduğu gibi motor devri artarken iletilen gücün de yavaş yavaş artmasını sağlar [24].

3.1.3.2 Planet dişli grubu

Tork dönüştürücüden gelen döndürme momentini kullanıcının isteğine göre ya da aracın o anki hız, ivme vb. parametrelerine göre artırmak veya azaltmak için kullanılır. İçerisinde oranları farklı birçok dişli bulunur. Dişlilerin birbiri ile bağlantıları hidrolik kontrol sistemini oluşturan selonoid valfler aracılığı ile değiştirilerek, giriş ve çıkış mili arasındaki tork ve açısal hız oranı ayarlanır.



Şekil 3.9 : Planet dişli grubu [23].

3.1.3.3 TCU (Transmission control unit)

Planet dişli grubunun konum değiştirmesini sağlayan hidrolik sistemi TCU modülü yönetir. Tıpkı ECU modülü gibi içersinde mikroişlemci mimarisine sahip ve çevre birimleri olan elektronik bir karttır.

TCU manuel sürüşte sürücünün isteği doğrultusunda, otomatik sürüşte ise ECU modülünün yayınladığı gaz pedalının konumuna ve motorun devrine bakarak selonoidlere vitesin üst ya da alt kademeye geçmesi için komut gönderir. Bu işlemi şanzıman üzerine yerleştirilmiş sensörlerden aldığı değerlere ve CAN-C üzerinden okuduğu verilere bakarak gerçekleştirir. Hangi durumlarda üst veya alt vites geçilmesi gerektiği aracın tasarımı esnasında hesaplanmış, denenmiş ve TCU içinde yer alan EPROM'a Look-Up Table olarak yüklenmiştir.

Vites artıma ve azaltma durumuna ek olarak, vites dişli konumunun boşa veya geri vites geçişi de bu birimin denetimi altındadır. Sürücü dişli üzerindeki bu işlemi gerçekleştirebilmesi için fren pedalına basmak zorundadır. TCU fren pedalının altında bulunan bir anahtar sayesinde pedala müdahale olup olmadığını sürekli olarak kontrol eder. Belirdiği gibi bir işlem gerçekleştirilmek istediğinde pedala basılmazsa vites geçişine izin vermez ve kullanıcıyı uyararak hata moduna geçer.

Modül asıl görevinin yanı sıra şanzımanda meydana gelebilecek yağ sıcaklığının aşırı artması, vites geçişinin gerçekleşmemesi gibi olası sorunlar için sürekli tetiktedir. Ayrıca vites konumu, tork arttırma veya azaltma isteği, türbin ve pervanenin sabit bağlantı durumu gibi bilgileri diğer ünitelerin kullanması için CAN-C hattına yazmaktan da sorumludur.

3.1.3.4 Diferansiyel

Projede kullanılan araçta olduğu gibi önden çekişli araçlarda diferansiyel şanzımanla birleşik halde bulunur. Vites dişli grubundan aldığı momenti, 90° yön değiştirip

dikey duruma getiren ve tekerleklerin bağılı bulunduğu ön akslara ileten elemandır. [25]

Diferansiyeli diğer bir görevi aracın dönüşlerinde ortaya çıkar. Araç dönerken içteki tekerlek, dıştaki tekerleğe göre daha az yol almak zorundadır. Aksi takdirde tekerleklerde kayma veya aşınma, iki tekerlek arasındaki dişlilerde ise aşırı zorlanmaya bağlı sıyırma veya bozulma gibi istenmeyen durumlar ortaya çıkabilir. Her ne kadar aracın ön geometri açıları bu durumu ortadan kaldıracak şekilde tasarlansa da, bu görevi asıl üstlenen eleman diferansiyel dişlisidir.

Dişli yapısından dolayı vitesten bağımsız olarak sabit bir oranda torku arttırıp, açısal hızı da düşürür. Projede kullanılan aracın diferansiyel tork çevrim oranı $k_D = 3.733$ 'dür.

3.1.4 Tekerlekler ve şasi

Tekerlekler akslardan gelen torku yarıçaplarına bağılı olarak yerle temas eden yüzeyde dikey bir kuvvete dönüştürürler. Gaz pedalının basış oranı, motor devri, vites durumu, fren pedalının konumu gibi birçok faktör bu kuvvetin büyüklüğünü belirler. Bu kuvvet sayesinde araç ileri ya da geri hareket eder ve ivmelenir. Diğer bir taraftan şasi ve üzerinde bağılı bulunan diğer tüm parçalar kütle ve ataletlerinden dolayı ivmelenme sırasında ivmelenme kuvvetinin tersi yönünde bir etki yaratırlar.

3.2 İvmelenme Sistemine Sinyal Uygulama

Projeye ait aracın ivmelenme sistemi incelendiğinde kapalı bir yapıya sahip olan ECU'nün içine girmenin ve mevcut algoritmayı değiştirmenin hem zor hem de araç için tehlike oluşturabilecek bir çözüm olduğu görülmüştür. Yine ECU'yü by-pass edip yerine bir modül oluşturup programlamak da oldukça zor, yapılsa bile araca bağılı alt sistemlerde deformasyon oluşturabilecek bir durum ortaya çıkarabilir. Özellikle motorun fiziksel yapısı göz önünde bulundurularak oluşturulan motor haritasını bilmeden enjeksiyon sistemini yönetmek büyük bir problem olabilir.

TCU hariç dizel motor çıkışından tekerleklerle kadar olan kısım ya hidrolik ya da mekanik bir yapıya sahiptir. Bu noktada TCU'ya elektronik bir sinyalle müdahale edip vites kontrolü gerçekleştirmek, gelen tork kontrol edilemediği için ivmelenme için yetersiz kalacaktır.

Mevcut durumda gaz pedalından ECU modülüne gelen sinyalleri devre dışı bırakıp tıpkı kullanıcının pedala müdahale ettiğinde oluşan sinyaller gibi benzer iki işaret üretmek aracın motor devrinin ve hızının ayarlanması için en uygun ve en ergonomik çözüm olacaktır. Basit bir dizayna sahip olması, eski yapıyı bozmaması bu yöntemin en büyük avantajları arasında sayılabilir.

3.3 Frenleme

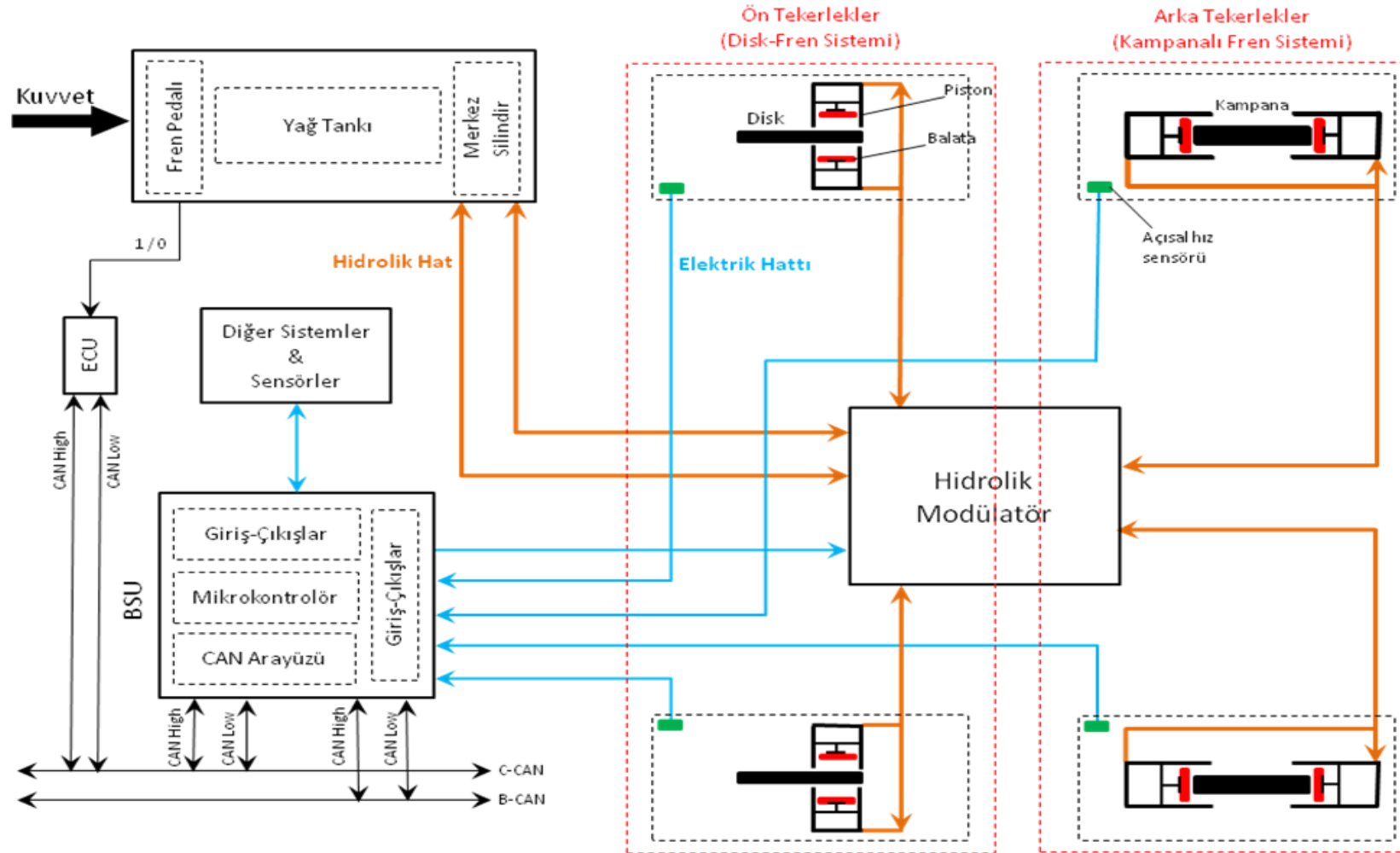
Fren sistemi hareket halindeki bir aracın emniyetli bir şekilde yavaşlamasını veya durmasını sağlayan, hareketsiz bir aracın ise sürücünün isteği dışında ivmelenmesini engelleyen sistemdir.

Araç üzerinde park freni ve servis freni olmak üzere iki ayrı fren sistemi bulunmaktadır. Park freni diğer bir deyişle el freni sürücünün aracı olduğu yere sabitlemek için kullandığı yapıdır.

Fren kolu kullanılarak fren telinin oluşturduğu mekanik bir bağlantıyla tekerleklerin kilitlenmesi sağlanır. Seyir esnasında yapılan frenlemelerde park freni kullanılmadığı için bu bölümde yalnızca servis incelenecek ve devamında kontrolü için gereken sistem tasarlanacaktır.

Servis freni fren pedalından başlayıp tekerlekler kadar olan hidrolik, mekanik ve elektronik bir bütünden oluşur, ayakla kontrol edilir. Sistem fren pedalı, merkez silindir, hidrolik sıvı tankı, hidrolik modülatör, pistonlar, diskler, kampanalar, algılayıcılar ve tüm sistemi kontrol eden BSU(Brake System Control Unit) gibi elemanlardan oluşmaktadır.

Frenleme Şekil 3.10'da görüldüğü gibi fren pedalına tahrik uygulanmasıyla başlar. Gelen kuvvet bir sonraki eleman olan merkez silindir içersindeki pistonları hareketlendirir. Hareketlenen pistonlar silindir haznesinde bulunan hidrolik sıvıyı sıkıştırır ve iki ayrı çıkıştan hidrolik modülatöre gönderir. Normal çalışma esnasında hidrolik modülatör basınçlı sıvıyı ön ve arka tekerlekler üzerinde bulunan pistonlara iletir. Bu elemanlar basınçlı sıvı sayesinde hareket eder ve üzerine bağlı olan balataları tekerlekle aynı hızda dönen disk ve kampanalara doğru iter. Balataların disk ve kampanalara teması ile dönme yönünün tersinde bir sürtünme kuvveti oluşturur. Bu noktada kinetik enerji-termal enerji dönüşümü gerçekleşir ve böylece aracın yavaşlaması veya olduğu yerde sabit kalması sağlanmış olur.



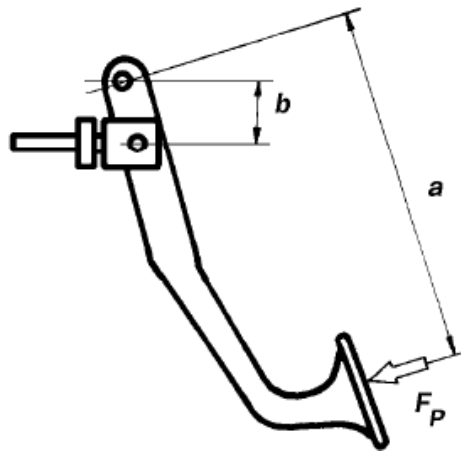
Şekil 3.10 : Frenleme sistemi genel yapısı.

Çok ani frenleme durumlarında tekerlekler kilitlenebilir. Bu esnada tutunma oldukça düşer ve durma mesafesi artar. Benzer durumları engellemek için ABS(Anti-Lock Braking System) olarak adlandırılan sistem geliştirilmiştir. ABS hem yazılımsal olarak BSU'nun içersine hem de donanımsal olarak var olan fren sistemine entegre edilerek oluşturulur. ABS, sürtünmeli fren elemanlarına giden basınçlı sıvının modülatör aracılığıyla aç-kapa yapılması prensibine dayanarak çalışır.

3.3.1 Fren pedalı, merkez silindir ve hidrolik sıvı tankı

Fren pedalı sürücünün frenleme için referans değeri kuvvet olarak aktardığı elemandır. İki kısımdan oluşur. İlki girilen kuvveti bir sonraki eleman olan merkez silindire ileten mekanik kısımdır. Fakat pedal bu işlemi gerçekleştirirken gelen kuvveti Şekil 3.11'de görülen yapısı nedeniyle daha kolay bir frenleme için $ip = a/b$ oranında büyüterek silindire gönderir.

İkinci kısım ise fren pedalının arkasına yerleştirilmiş olan ve pedala basılıp basılmadığını algılayan switch elemanıdır. Bu eleman “frene basıldı” sinyalini üretir ve motor kontrol ünitesine (ECU) gönderir. Bu işaretin gelmesi ile ECU yönettiği birimleri gerekli konuma getirmesinin yanı sıra CAN aracılığı ile araç üzerindeki diğer modülleri de frene basıldığını konusunda uyarır. Böylece diğer tüm sistemler frene basıldı bilgisini alır ve buna göre konumlanır. Tez kapsamında hız kontrolör tasarlanan otomatik şanzımana sahip aracın vites değişimi(ileri-geri-boş) frene basılarak gerçekleştiğinden, TCU modülü de ECU'nün verdiği bu bilgiyi kullanmaktadır.



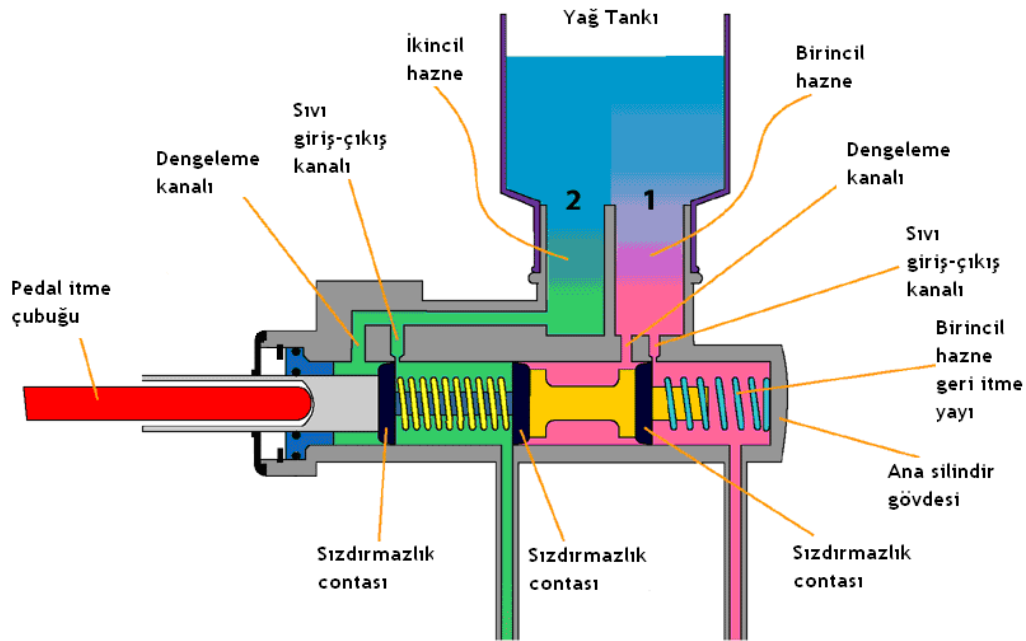
Şekil 3.11 : Fren pedalı [26]

Merkez silindir içersinde pistonları, geri dönüş yaylarını, sıvı haznesini ve bu sıvının geçiş yapabileceği kanalları barındırır. Pedaldan galen kuvvet pistonları tahrik ederek silindir içersinde hareket etmelerini sağlar. Böylece silindir haznesinde bulunan hidrolik sıvı iki ayrı çıkıştan sıkıştırılmış bir şekilde hidrolik modülatöre gönderilir. İki çıkış olmasının sebebi sıvı taşıyan hatlarda veya elemanlarda olası bir probleme karşı alınmış güvenlik önlemidir.

Fren pedalından gelen kuvvet sonlandığında yay sistemi hem pistonları hem de pedalı başlangıç konumuna getirir. Bu yayların asıl görevi ise silindirin haznesinde vakum oluşturarak modülatöre gönderilen sıvıyı geri çekmektir. Bu işlemin yapılmaması halinde tekerleklerin fren durumunda kalma olasılığı vardır.

Fren hidroliğini içeren hidrolik sıvı tankının merkez silindirin hemen üstünde, kanallarla ya da taşıyıcı hatlarla silindire bağlı olduğu Şekil 3.12’de görülmektedir. Zamanla hidrolik sıvının buharlaşma veya ufak-tefek kaçaklar yüzünden eksilmesi halinde sisteminin yağ ihtiyacı buradan karşılanmaktadır.

Tank kapağındaki hava deliklerin sayesinde haznedeki sıvı basıncı sürekli olarak atmosfer basıncına eşit olacaktır. Bu yapı fren pedalına basılmadığı anlarda sistem içersindeki tüm sıvı basıncının da atmosfer basıncına eşit olmasını, böylece sistemin başlangıç konumuna daha rahat gelmesini sağlar.



Şekil 3.12 : Ana silindir ve yağ tankı [27].

Ayrıca yağ tankındaki sıvı seviyesi BCU (Body Control Unit) tarafından periyodik olarak bir sensör aracılığı ile sürekli ölçülür. Kap içerisindeki sıvı azaldığında sürücü hazneye yağ takviyesi konusunda uyarılır. Bu kontrolün yapılması aracın sürüş güvenliği açısından zorunludur. Çünkü güç iletimini sağlayan hidrolik sıvının eksilmesi halinde frenleme aksayabilir veya fren sistemi devre dışı kalabilir.



Şekil 3.13 : Yağ haznesi.

3.3.2 Hidrolik modölatör, pistonlar, diskler ve kampanalar

Hidrolik modölatör gelen basınçlı sıvıyı ön ve arka tekerlekler üzerinde bulunan fren elemanlarına istenilen şekilde dağıtan ve kontrol eden elemandır. Modölatörün ABS'nin devrede olup olmamasına bağlı olarak iki önemli görevi vardır.

Frenleme esnasında aracın ön ve arka akslardaki yük dağılımı araçta bulunan süspansiyon gibi sistemlerden dolayı normal sürüş sırasındaki dağılım ile farklılık gösterir. Yük miktarı ön aksta normal sürüşe göre artar, arka aksta ise azalır. Fren kuvveti akslara binen dikey kuvvetlerle orantı olarak dağıtılmadığı takdirde araç kontrolünün kaybedilmesine sebep olabilir. Çünkü arka tekerleklerin yere basma kuvvetinin daha az olmasına rağmen bu tekerleklerde daha fazla fren yapılması arka tekerleklerin kaymasına yol açacaktır. Modölatörün görevi ön ve arka sistemlere giden hidrolik basıncı limitleyerek düzgün bir dağılım sağlamaktır.

İkinci görevi ise ABS'nin aktif olmasıyla ortaya çıkar. ABS aktif olduğunda modölatör içerisindeki valflerin yüksek hızlarda sürekli aç-kapa yapılması sayesinde basınçlı sıvı pistonlara kesik kesik iletilmektedir. Bu valflerin, dolayısıyla modölatörün kontrolü BSU tarafından yapılır.

Pistonlar diskli fren sistemi ve kampanalı fren sistemi üzerinde yer alan, basınçlı sıvı sayesinde lineer bir hareket yapan elemanlardır. Ön tekerleklerle takılmış diskli fren sistemine ait mekanizmanın piston tarafından hareketlendirilmesi ile balatalar dönen

tekerleğe sabitlenmiş ve tekerlekle ile beraber dönen diske doğru itilir. Balataların disk yüzeyine teması açısal hızın tersi yönünde bir sürtünme kuvveti oluşturur. Bu da tekerleğin yerle temas ettiği noktada aracın hızını yavaşlatan ters yönde bir kuvvet oluşturur, böylece frenleme gerçekleşmiş olur. Sıvı basıncı azaldığında veya atmosfer basıncına eşit olduğunda piston hem yaylar sayesinde hem de içerisinde sıvının vakum etkisinden dolayı başlangıç konumuna gelir. Böylece balatalar disk yüzeyinden ayrılır ve frenleme sona ermiş olur.

Arka tekerleklerde bulunan kampanalı fren sisteminde ise mekanizma için ihtiyaç duyulan hareket yine kampana üzerindeki pistonlardan alınır. Piston, balatalar ve yaylardan oluşan mekanizma dönen bir çembere benzeyen kampananın içinde yer alır. Pistona basınçlı sıvı geldiğinde balatalar içten dışa doğru bir hareket ile açılarak kampananın iç yüzeyine dokunur ve diskli fren sistemindeki gibi sürtünme kuvveti meydana gelir. Bu kuvvetle birlikte frenleme yapılmış olur. Diskli sistemde olduğu gibi frenlemeyi tetikleyici etkenler ortadan kalktığında yaylar ve vakum aracılığıyla sistem ilk haline geri döner ve frenleme biter.

3.3.3 BSU(Brake System Unit) ve ABS(Anti-Lock Break System)

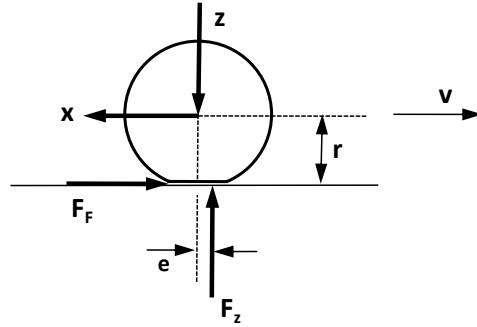
BSU tüm fren sistemini yöneten, kontrol eden, arıza vb. durumları tespit eden mikroişlemci mimarisi ile tasarlanmış ve programlanmış elektronik birimdir. Belirtilen işlemleri yerine getirirken sistem üzerinde farklı noktalara monte edilmiş algılayıcılardan gelen sinyalleri ve elektronik olarak kumanda edilebilen valf vb. elemanları kullanır. Aynı zamanda araç üzerinde yer alan diğer modüllerin CAN üzerine yazdığı verilerden de faydalanır.

ABS araçların daha hızlı ve güvenli bir şekilde durmasını veya yavaşlamasını sağlayan bir sistemdir. ABS'nin temel prensibi frenleme esnasında tekerleklerin kilitlenmesini engelleyerek maksimum tutunmayı sağlama üzerinedir.

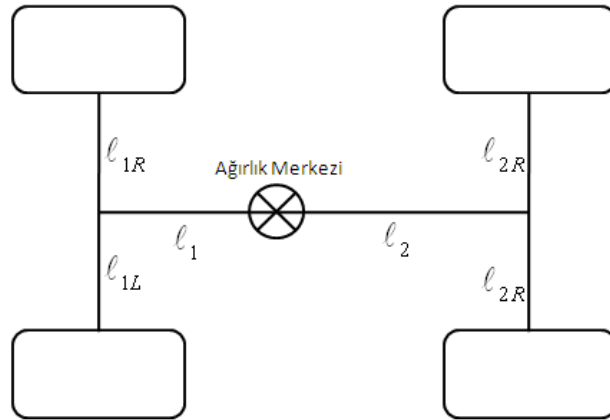
Frenleme sırasında araç momentumundan dolayı ileri gitmek isterken, tekerlekler balatalar ile dönen elemanlar arasındaki sürtünme kuvvetinden ötürü yavaşlamak veya sabit kalmak isteyecektir. Bu yüzden her tekerleğin yerle temas ettiği noktada araç hızına ters yönde aracı durdurmaya çalışan yatay ekseninde bir kuvvet oluşur (F_F). Her teker için bu kuvvetin alabileceği maksimum bir değer vardır. Tüm tekerleklerdeki F_F değerlerinin toplamı ne kadar büyük olursa otomobil o kadar kısa mesafede duracaktır. Kuvvetin alabileceği limit aşağıdaki gibi belirlenir.

$$F_{F \max} = \mu_{\max} F_z \quad (3.1)$$

F_z aracın kütlesinden kaynaklı tekerleğin üstüne binen dik kuvvettir. Bu kuvvet şekilde de görüldüğü gibi aracın ağırlık merkezinin tekerleğe olan enine ve boyuna uzaklığı ile ilgilidir. Bir araçta ağırlık merkezi birçok sebepten dolayı yer değiştirebilir. Binen bir yolcunun ağırlığı, oturduğu koltuk, taşınan yükün konduğu nokta bunlardan bazılarıdır. Bir başka neden ise aracın kendi içersinde elemanlarındaki yer alan elemanların yapısından veya çalışma prensibinden kaynaklanır. Tekerleklerdeki esneme veya süspansiyon sistemindeki yaylanma bu duruma örnektir. Duran bir aracın ağırlık merkezi bahsedilen bu nedenlerden dolayı yavaşlarken ön dingile doğru biraz yaklaşır. Ayrıca aracın aerodinamik yapısı göz önüne alındığında hareket esnasında çarpan rüzgârın dikey kuvvet değeri de F_z 'i etkileyen faktörlerdendir. Bu proje kapsamında bahsi geçen etkenler göz ardı edilmiş, her tekerleğe düşen F_z kuvveti araç sabit ve düz konumdayken, yolcusuz ve yüksüz durumdayken yaklaşık olarak hesaplanmıştır.



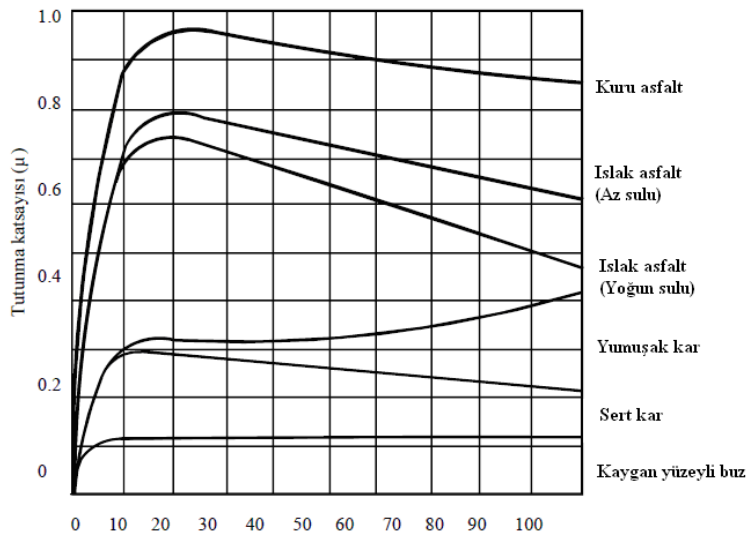
Şekil 3.14 : Tekerlek üzerindeki kuvvetler ve uygulama noktaları.



Şekil 3.15 : Araç ağırlık merkezinin dingillere olan uzaklığı.

F_F kuvvetini etkileyen diğer bir faktör yer ile lastiğin arasındaki sürtünme kuvveti katsayısıdır. μ katsayısını belirleyen birçok etken vardır. Lastiğin aşınmışlığı, lastiğin yapısı, lastik hava basıncı, yolun yapım maddesi, yolun ıslak-kuru olması ve hatta yolun sıcaklığı gibi parametreler örnek olarak sayılabilir. Fakat belirtilen parametreler aynı bile olsa Şekil 3.16’da olduğu gibi μ katsayısı kaymaya bağlı olarak değişkenlik gösterir. $\mu_{\max}$ ise yer ile tekerleğin lastik yüzeyi arasındaki sürtünme katsayısının en yüksek değerini temsil eder ki bu tutunma katsayısı olarak adlandırılır. Hemen hemen her durumda μ katsayısı için maksimum tutunma 0.15-0.2 arasında bir kayma miktarı ile gerçekleşir. Bu maksimum tutunma noktası tekerlekle yer arasındaki en yüksek yatay kuvvet ($F_{F\max}$) kuvveti doğurur. Diğer bir deyişle $F_{F\max}$ frenleme esnasında tek tekerleğin aracı durdurmak veya yavaşlatmak için aksi yönde verebileceği en büyük kuvvete karşılık gelir.

ABS içermeyen bir sistemde çok sert ve ani bir frenlemelerde balatalar ile dönen yüzeyler arasındaki sürtünme kuvveti pedala sert basılmasından dolayı çok yüksek değerlere ulaşabilir. Bu tekerleklerin kilitlenmesine ve kaymasına neden olur. Kayma 1’e doğru yaklaştıkça μ değeri azalır. Devamında aracı yavaşlatmayı sağlayan yatay teker kuvvetlerinden (F_F) hiç biri en yüksek değerini ($F_{F\max}$) yakalayamaz. Bu otomobilin daha geç durmasının yanı sıra tekerleklerin herhangi birinin kilitlenmesine de sebep olabilir. Tekerleklerin kilitlenme durumu kaymanın 1 olduğu anlamına gelir ve o andaki μ değeri kayma katsayısı olarak adlandırılır. Kayma değerinin 1 veya 1’e yakın olması aracın spin atması ya da sürücünün kontrolünü kaybetmesi gibi tehlikeli durumları doğurabilir.



Şekil 3.16 : Kuvvet bağlantı katsayısı-kuvvet ilişkisi [28].

ABS'li sistemlerde, frenlemenin büyük dezavantajı olan tekerlek kilitlenmesi gibi bir durumla karşılaşılmaz. ABS, μ ne olursa olsun tekerlerdeki kaymaları 0.15 - 0.2 civarında tutar ve tüm tekerleklerde F_{Fmax} değerinin elde edilmesini sağlar. Böylece aracı en kısa mesafede durdurmuş olur.

ABS sisteminin yazılım kısmı BCU'nun içine gömülür. Buna ek olarak sistemin çalışabilmesi için ihtiyaç duyacağı donanım parçaları da var olan fren sistemine eklenmiştir. Sistem kaymayı direkt olarak ölçemez, ama onun yerine Şekilde 3.10'da görüldüğü gibi her tekerlek üzerine yerleştirilmiş açısal hızölçerleri kullanır. BSU her tekerin açısal dönüş hızını bu algılayıcılar sayesinde periyodik olarak ayrı ayrı ölçer. Ölçüm sonuçları sonucunda elde edilen tekerlek hızlarını birbirine göre kıyaslayarak ve CAN hattından okuduğu diğer bilgileri de değerlendirerek hangi tekerleğin ne kadar kaydığını tespit eder. Sonrasında hidrolik modülatördeki valfleri çok seri bir şekilde aç-kapa yaparak kaymayı 0.15-0.2 civarında tutmaya çalışır. Tekerlekteki tutunmanın azalacağı anda üzerindeki balataların dönen yüzeyle olan teması pistona giden basınçlı sıvının geri çekilmesi ile kesilir ve tekerlek serbest bırakılarak tekrar tutunma sağlanır. Tekrar tutunma sağlandığında fren hidroliği pistona hızlı bir şekilde yine gönderilir. Bu işlem çok seri bir şekilde yapıldığı için balatalar ile dönen yüzey arasında maksimum tutunmayı sağlayan ortalama bir sürtünme kuvveti oluşur.

3.4 Fren Sistemine Sinyal Uygulama

Araca ait fren sistemi incelendiğinde BSU modülü dışında hiçbir aksamın elektriksel olarak kontrol edilemeyeceği görülmüştür. Fakat BSU kontrol edilerek fren pedalına basıldığında meydana gelen hidrolik basınç gibi bir etkiyi oluşturmak mümkün olmadığından fren sistemini, gaz sisteminde olduğu gibi elektriksel olarak kontrol etmek mümkün değildir. Bu nedenle mevcut sisteme, fren pedalına müdahale eden mekanik bir yapı tasarlanmış ve bu mekanik sistem daha sonra elektriksel olarak kontrol edilmiştir. Mekanik sistem lineer motor, lineer motor sürücüsü ve mafsallardan oluşmaktadır. Motor sürücüsü kontrol merkezinden aldığı analog sinyale göre lineer motoru ileri-geri hareket ettirmekte, bu hareket de fren pedalına basılma ya da basılmama işlemini gerçekleştirmektedir. Birkaç farklı noktada kullanılan mafsallar ise motorun hareketi esnasında bir kasılma meydana gelmemesi için yerleştirilmiştir.



Şekil 3.17 : Lineer motorun mekanik bağlantısı.

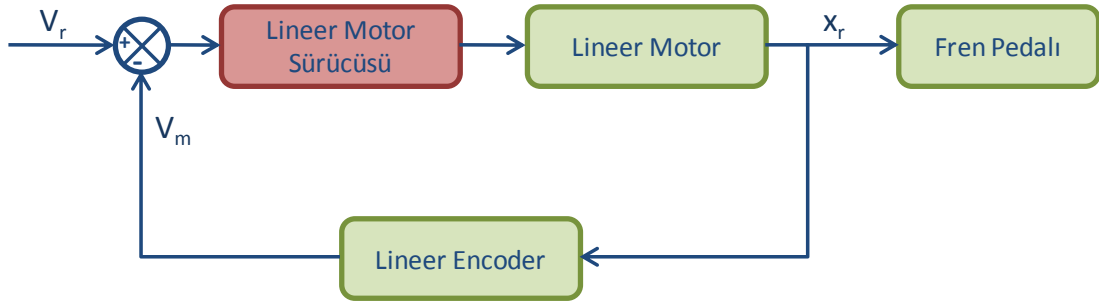
Ayrıca frenin sürücü tarafından kontrol edilmek istenileceği de düşünülerek, bu motor monte edilirken fren pedalına direkt olarak değil bir bağlantı aparatı yardımıyla bağlanmıştır.



Şekil 3.18 : Fren bağlantı aparatı.

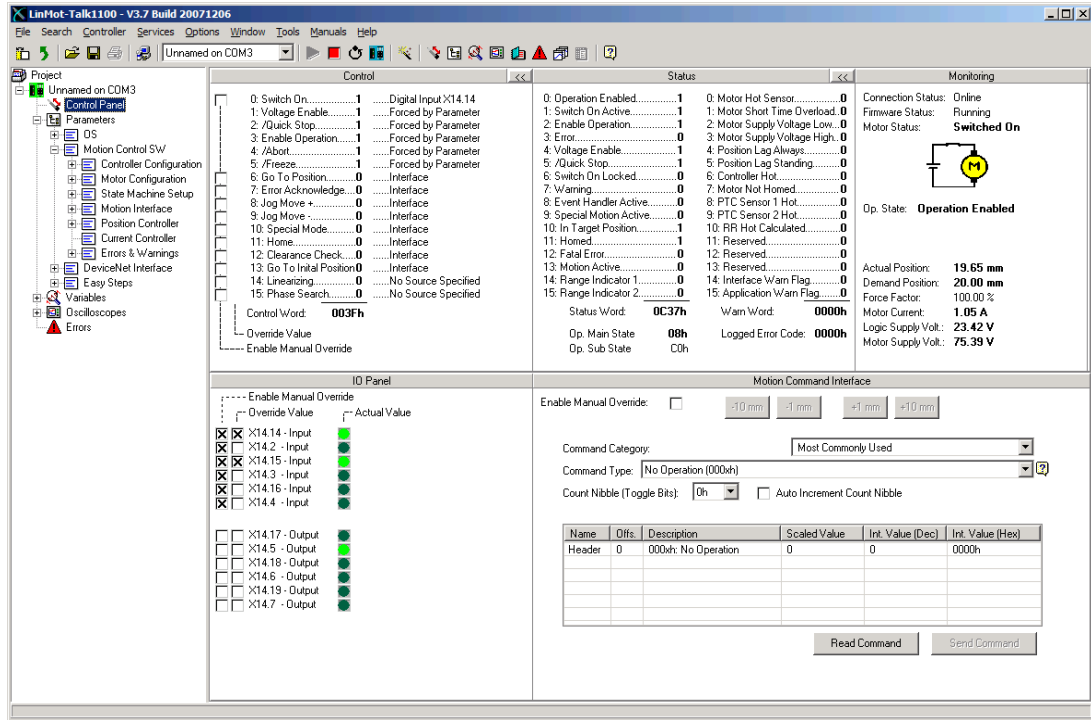
3.4.1 Lineer motor sürücüsü

Oluşturulan elektromekanik yapıda sürücü olarak Linmot marka B1100 model [29] servo kontrolör kullanılmıştır. Şekil 3.19’da görüldüğü gibi bu ünite konum kontrolörü olarak çalışmaktadır. Kontrolör lineer motoru hangi konuma götüreceğini belirten referans değeri analog girişine gelen 0 – 10 V arası değişen elektriksel işaretten (V_r) almaktadır. Daha sonra bu referans değerden lineer motor üzerinde bulunan ve lineer motorun konumunu mutlak olarak ölçmemizi sağlayan enkoderden gelen işareti çıkartarak hata miktarını bulur. Yapılandırma esnasında girilen sabit parametrelerle birlikte hata miktarı değerlendirilir ve kontrol sinyali üretilir.



Şekil 3.19 : Lineer motor

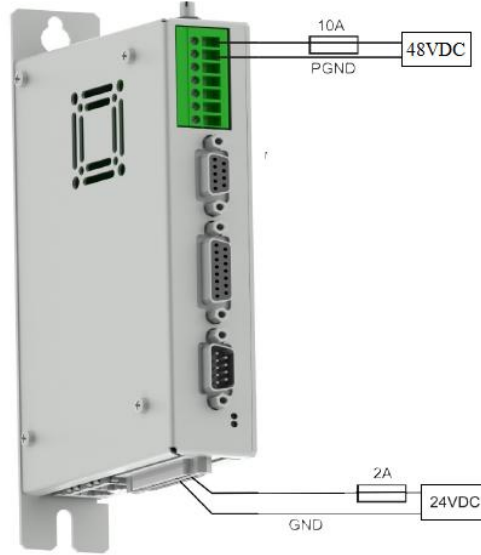
Kontrolör çalıştırılmadan önce Linmot-Talk isimli bir yazılım ile yapılandırılmıştır. Bu yapılandırma sırasında hangi motorun sürüleceği(model), hangi encoderden değer alınacağı gibi bilgiler, kullanılan elemanlara göre girilmiş, böylece diğer birimler kontrolöre tanıtılmıştır. Bunların dışında kontrol işareti hesaplanırken kullanılan referans değerini hangi giriş-çıkış biriminden alınacağı, motorun istenilen konuma gidiş hızı, motorun çalışma akımı gibi parametreler de bu yazılım sayesinde ayarlanır.



Şekil 3.20 : Linmot-Talk yazılımı.

Referans değeri analog girişten alınmaktadır ve 0 – 10 V aralığındadır. Bu gerilim değeri kontrolör içerisinde 10 bitlik bir ADC ile 0 – 1023 arası değer alan bir dijital sinyale dönüştürülür. Ek olarak bu aralığa karşılık gelen pedal hareketi de yapılandırma esnasında 0–58 mm'ye denk gelecek şekilde belirlenmiştir. Bu

mesafeler fren pedalının %0 ve %100'lük konumuna denk gelecek şekilde ayarlanmıştır. Ek olarak motorun istenilen konuma gidiş hızı 1 m/s, çalışma akımı ise 8 A olarak ayarlanmıştır.



Şekil 3.21 : Lineer motor sürücüsü.

Elektrik bağlantısına bakıldığında ise sürücünün iki ayrı besleme girişi olduğu görülmektedir. Bunlardan biri motora giden gerilimdir ve DC 24 V – DC 85 V arasında girilebilir. Otonom araç sisteminde bu değer DC 48 V olarak bağlanmıştır. Diğeri ise sürücünün kendi beslemesidir ve DC 24 V olarak bağlanmıştır.

Sürücüye ait diğer özellikler Çizelge 3.1’de verilmiştir.

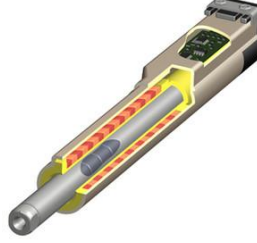
Çizelge 3.1 : Linmot B1100 sürücü özellikleri.

Linmot B1100	
Haberleşme arayüzü	RS232 / RS485 / Can Open
Ünite besleme gerilimi	DC 24 V
Motor besleme gerilimi	DC 24....DC 85 Volt
Dijital giriş	6 Adet, 0-24 V, 1 mA
Dijital çıkış	6 Adet, 0-24 V, 100 mA
Analog giriş	0-10 V, 10 bit ADC, 400µs örnekleme zamanı
Diferansiyel analog giriş	(-)10 V – (+)10 V, 10 bit ADC, 400 µs örnekleme zamanı

3.4.2 Lineer motor

Fren pedalını hareket ettirecek eyleyici olarak doğrusal hareket eden Linmot marka PS01-23Sx80 model stator ve PL01-12x170 model sliderdan oluşan lineer motor

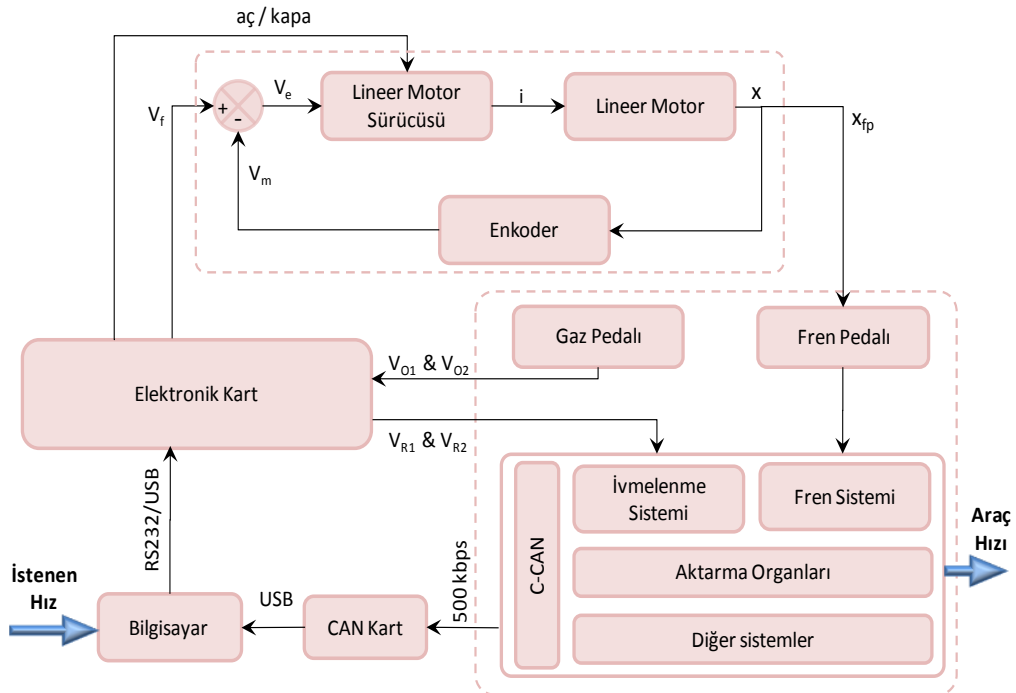
kullanılmıştır. Bu model ürünler seçilirken motorun verebileceği kuvvet, pedala olan uzaklık ve pedalın hareket aralığı dikkate alınmıştır. Motorun ortalama kuvveti 10 N, sliderın hareket mesafesi ise 140 mm'dir.



Şekil 3.22 : Lineer motor.

3.5 Hız Kontrolörü İçin Tasarlanan Sistem

Otonom aracın hız kontrolörü için ivmelenme ve frenleme sistemlerini kontrol etmek gerekmektedir. Bu iki sisteme nasıl müdahale edileceği, sistemlerin özellikleri incelenirken belirlenmiştir. İvmelenme sistemi için pedalın iki çıkışının kopyası üretilerek ve ECU'ye gönderilecektir. Fren için ise lineer bir motor aksamı düşünülmüş, bu motorun konumu 0–10 V arası analog sinyal ile kontrol edilmiştir. Bu sinyalleri üretecek olan hız kontrolörünün ve sistemin genel blok diyagramı Şekil 3.23'deki gibi tasarlanmıştır.



Şekil 3.23 : Hız kontrolörü genel blok diyagramı

Şekil 3.23’de görüldüğü gibi kontrol algoritması bilgisayarda olacak ve referans hız yine aynı program aracılığı ile kullanıcıdan alınacaktır. Kontrol algoritmasının gerekli olan sinyallerin dijital karşılıklarını üretebilmesi için aracın ve motorun dinamiği ile ilgili bilgilere ihtiyacı vardır. Bu bilgiler önceki bölümdeki deneylerde de yapıldığı gibi oluşturulmuş bir başka yazılımla CAN-C hattından okunacaktır. Bu CAN hattına erişim CAN kart donanımı ile yapılacaktır.

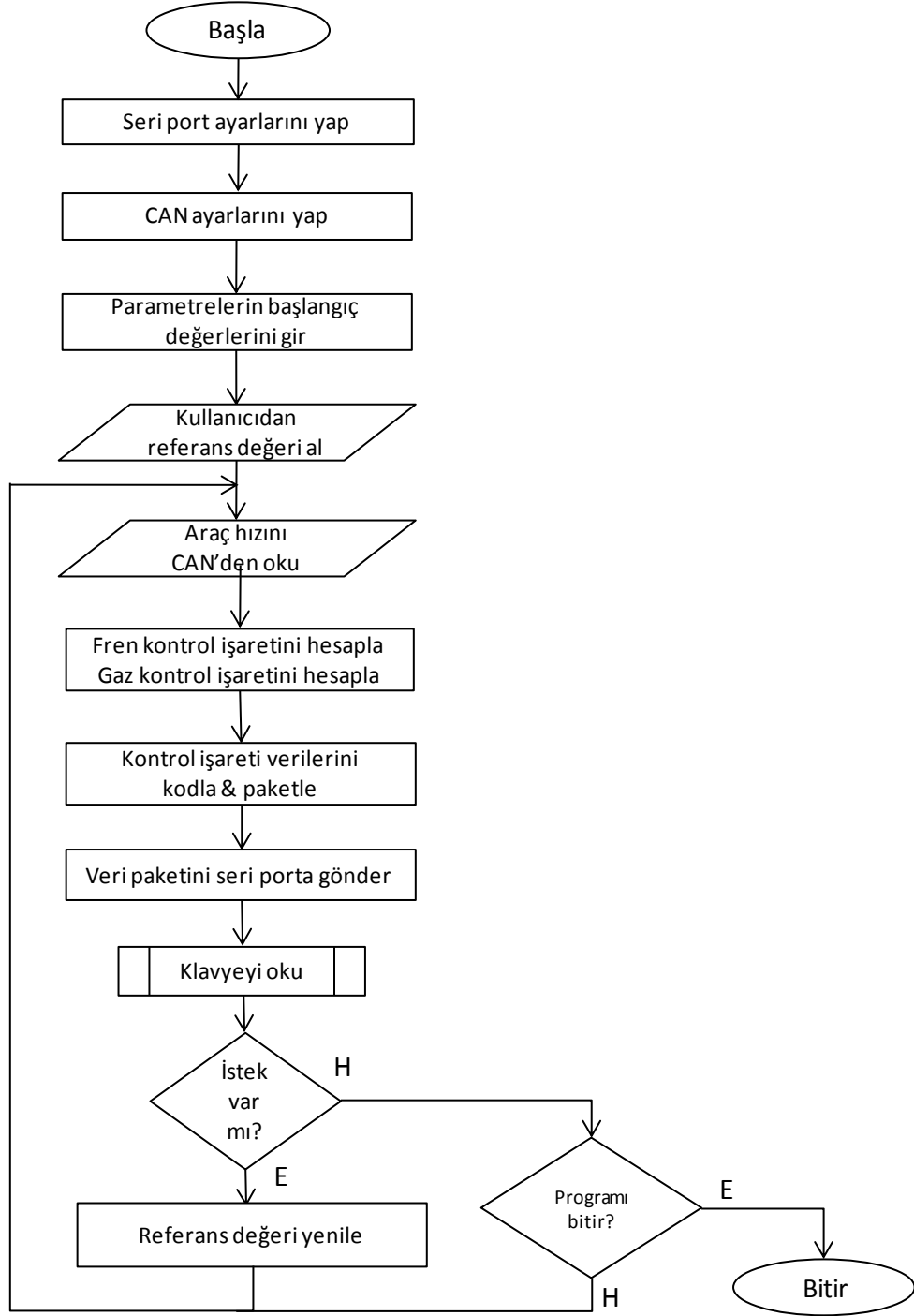
Kontrol algoritmasında işlenen verilere göre iki sistem için iki kontrol işareti elde edilecektir. Fakat bu sinyaller dijital yapıda olacağı için direkt olarak sisteme girmesi mümkün değildir. Analog hale getirilip, seviyelerinin ayarlanması gerekir. Bunun dışında gaz pedalı için oluşturulan kontrol sinyali yarı yarıya olacak şekilde çoklanmalıdır. Tüm bu işlemlerin yapılabilmesi için mikro kontrolörlü bir elektronik kontrol birimi tasarlanacaktır.

Elektronik kontrol birimi fren için işlediği sinyali analog olarak motorun kontrolörüne göndermektedir. Lineer motor kontrolcüsü bu sinyalin değerine göre kendi kapalı çevriminde lineer motoru konumlandıracaktır. Şekil 3.23’de görüldüğü gibi analog sinyalin dışında iki birim arasında bir başka iletim ortamı daha vardır. Bu hat aracın kontrolü kullanıcıya verildiğinde lineer motoru devre dışı bırakmak veya otonom konuma geçildiğinde tekrar devreye almak için gereken bilgileri taşımaktadır.

Kart tarafından işlenen diğer sinyal ise direkt olarak ECU birimine gönderilir. Yine aracın hem otonom çalışabilmesi hem de manüel çalışma özelliğini kaybetmemesi için gaz pedalının tamamen devre dışı bırakılmaması gerekir. Sürücü, otomobili kendisi kullanmak isterse elektronik kontrol birimi kendi işlediği sinyaller yerine gaz pedalından gelen sinyalleri ECU’ye gönderir.

3.5.1 Bilgisayar birimi

Otonom araç için tasarlanan hız kontrolörüne ait sistemde gaz ve fren sistemini kontrol edecek sinyalleri hesaplayan ve üreten algoritma bilgisayar üzerinde bulunan ve gerçek zamanlı olarak çalışan program içine kodlanmıştır. İlk aşamada kontrolör katsayıları boş bırakılan bu yazılım, katsayılar girildiğinde çalışır hale gelecektir. Uygun katsayıların bulunması için öncelikle sistemin simülasyon ortamında oluşturulması ve bazı hesapların yapılması gerekmektedir.

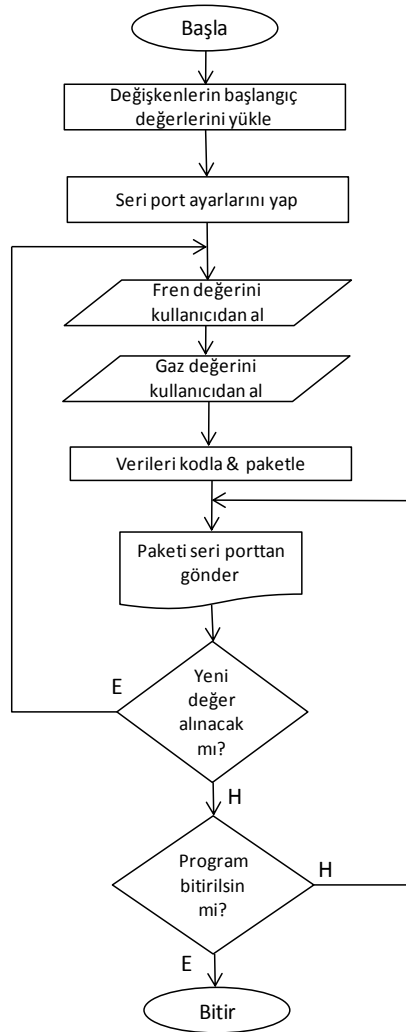


Şekil 3.24 : Bilgisayar ortamındaki kontrol algoritmasının akış diyagramı.

Program Visual Studio ortamında C kodu kullanılarak geliştirilmiştir. Üretilen bu yazılımın genel hatlarını ortaya koyan akış diyagramı Şekil 3.24'deki gibidir. Akış diyagramı incelendiğinde oluşturulan bu yazılımın tek görevinin kontrol işaretlerini hesaplamak olmadığı görülecektir. Araç için gereken referans hızın kullanıcıdan alınma işlemi yine bu program ile gerçekleştirilmiştir. Aynı zamanda referans hızı arttırma-azaltma, otonom moddan çıkma, verileri kaydetme gibi kullanıcının istediği

doğrultusundaki talepler de bu yazılım sayesinde yerine getirilmektedir. Kontrol işaretinin hesaplanabilmesi için referans değerin yanında aracın o andaki hızı da gerekmektedir. Önceki bölümde anlık araç hızının CAN hattında var olduğu anlatılmış ve bu bilgiyi okumak için gerekli kod parçacığı yazılıp denenmişti. Denenmiş olan bu kod parçası, kontrol algoritmasını da içeren ana program içine gömülmüş ve gerekli olan araç hızının okunmasını sağlamıştır.

Elde edilen kontrol işaretinin sayısal karşılığı seri port aracılığıyla bir sonraki birim olan elektronik karta yine bu program aracılığıyla gönderilmiştir. Fakat bu veriler direkt olarak gönderilmezler. Öncelikle paketin içersine bu bilgilerin yanı sıra belli bir matematiksel işlemle hesaplanmış doğrulama kodu yerleştirilir. Doğrulama kodu elektronik kart tarafından çözülür ve verinin bozulup bozulmadığı anlaşılır. Eğer gelen veride bir deformasyon tespit edilirse bu paketteki veriler kullanılmaz. Tüm bu işlemleri yapan program kodu Ek-D’de verilmiştir.



Şekil 3.25 : Test algoritması.

Elektronik kartın, devamındaki gaz ve fren sisteminin umulduğu gibi çalışıp çalışmadığını anlamak için öncelikli olarak kontrol katsayıları gerektirmeyen daha basit bir yazılım geliştirilmiştir. Şekil 3. 25’de akış diyagramı verilen bu program kontrol işaretlerini klavye aracılığıyla kullanıcıdan almaktadır. Bu programda da veriler doğrulama kodu ile birlikte bütün bir paket haline getirilir ve seri port aracılığıyla elektronik karta gönderilir. Böylece var olan sistemin büyük bir kısmı kontrol algoritması olmadan denenmiştir.

3.5.2 Elektronik kontrol birimi

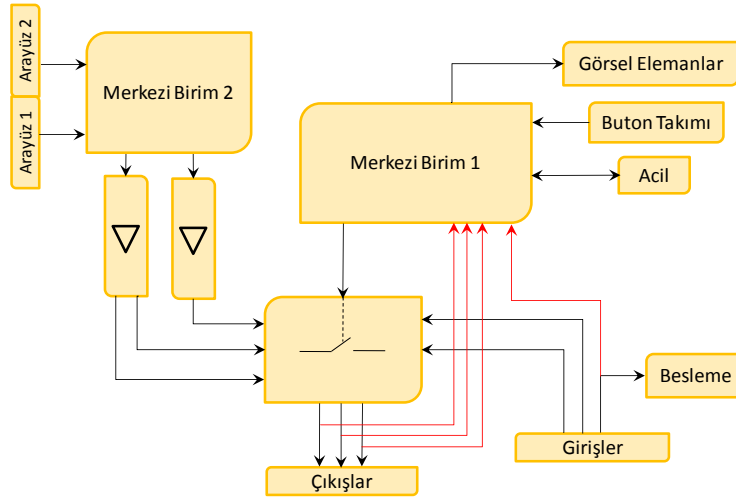
Araç üzerindeki ivmelenme ve fren sistemine müdahale edebilmek için bilgisayar tarafından üretilen değerleri uygun sinyaller haline getirip, gerekli noktalara göndermek elektronik kontrol biriminin temel görevidir. Bu kart ayrıca,

- Beslemeye gelen gerilimi sistemin ihtiyacı doğrultusunda regüle etmelidir.
- Bilgisayarla haberleşebilecek bir protokole sahip olmalıdır.
- Bilgisayardan gelen sinyaller hem fren hem de gaz sistemi için uygun hale getirilmelidir.
- Kullanıcıya otonom veya manüel çalışma seçeneği sunabilmelidir.
- Manüel moda geçildiğinde fren motorunu dijital bir sinyalle devre dışı bırakılabilmelidir.
- Manüel çalışma modunda gaz pedalı devreye sokulup, otonom modda devre dışı bırakılabilmelidir.
- Güvenlik açısından besleme gerilimi mutlaka kontrol edilebilmelidir.
- Güvenlik açısından gaz pedalına gönderilen sinyal kontrol edilmelidir.
- Kullanıcıya bilgisayarın yanı sıra başka bir görsel ara yüz sağlayabilmelidir.
- Acil durumlar için bir butona ve bu butona basıldığında da çalışacak acil durum koduna sahip olmalıdır.

Bilgisayardan gelen verilerin işlenip en kısa zamanda çıkışa gönderilmesi kontrol algoritmasının düzgün ve verimli çalışması açısından çok önemlidir. Bu yüzden Şekil 3.26’da görüldüğü gibi veri işleme hızını ve güvenliğini arttırmak için elektronik kontrol kartında iki ayrı merkezi birim oluşturulması düşünülmüştür ve görevler bu iki birim arasında paylaştırılmıştır.

Görsel verileri kullanıcıya aktarmak, kullanıcının veri girişi yapmasını sağlamak, kart üzerindeki belli noktalardan analog veya sayısal ölçümler alarak acil durumları

gözetlemek, seçicileri konumlandırıp sistemi otonom veya manüel konuma getirmek birinci merkezi birim tarafından yapılacaktır.

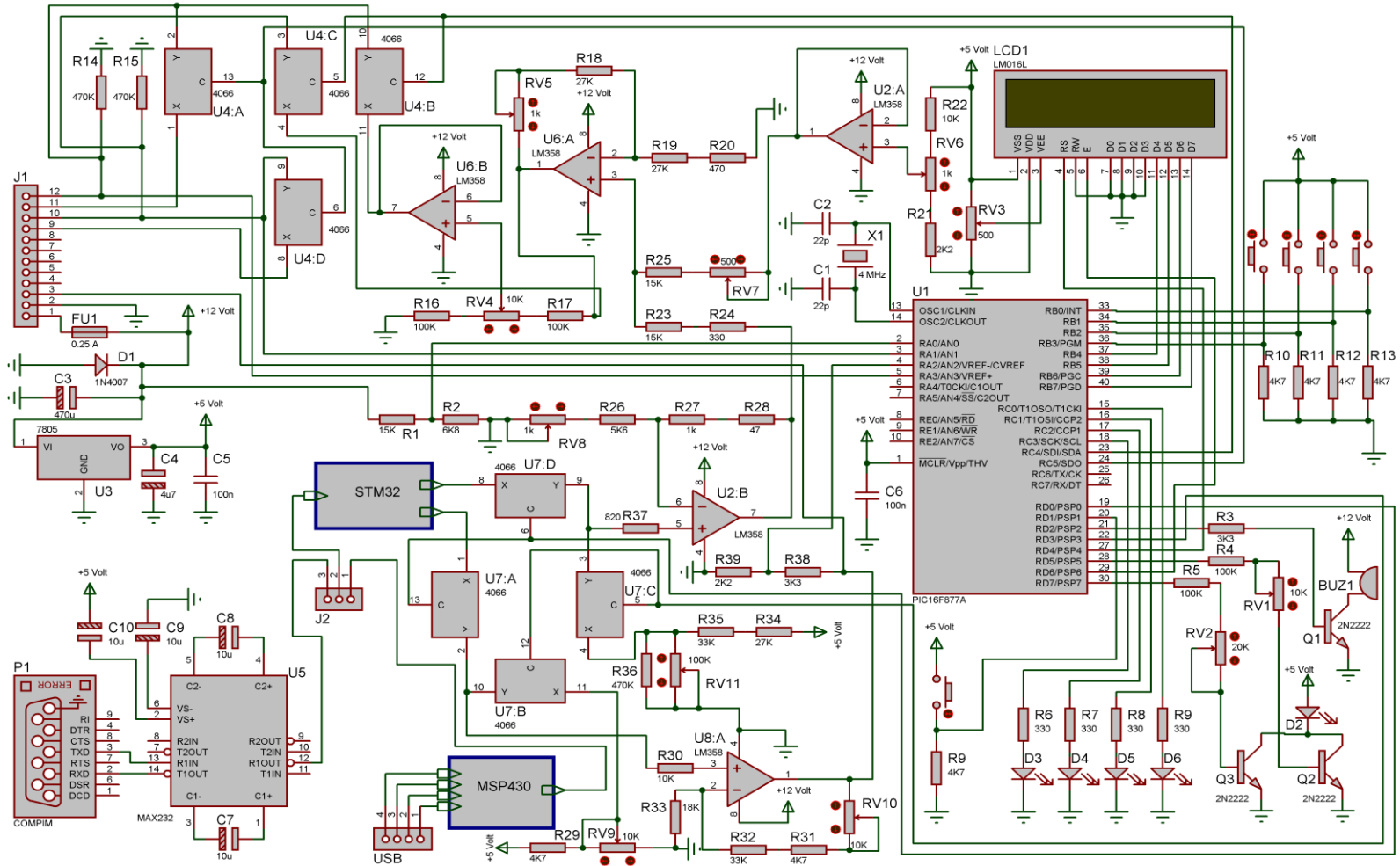


Şekil 3.26 : Elektronik kontrol biriminin blok diyagramı.

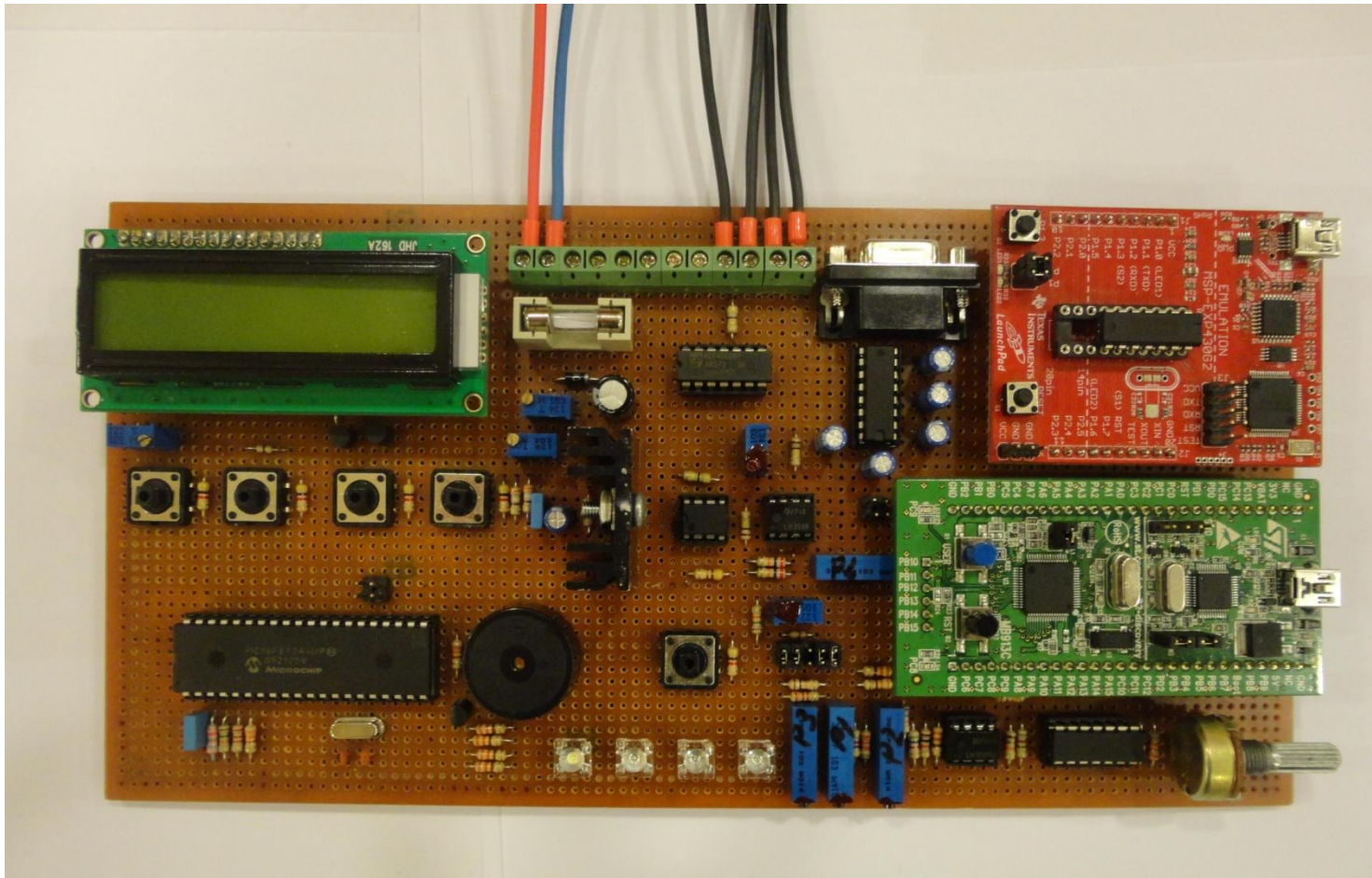
Bilgisayar ile iletişimi sağlamak, sayısal verileri bilgisayardan alıp-işlemek ve daha sonra bu verilere karşılık analog değerler üretip çıkışa vermek ikinci merkezi birimin görevleridir. Veri işlemede zaman kaybetmemek için bu birime başka görev atanmaz. Zaten iki ayrı merkez kullanılmasının asıl amacı veri işleme görevini diğerlerinden ayırmak ve minimum zaman kaybıyla işlemleri gerçekleştirmektir.

İkinci merkezi birimin ürettiği analog değerler ile bu sinyalleri kontrol sinyali olarak kullanacak olan gaz ve fren sisteminin alt ve üst sınırları farklıdır. Fren sistemine müdahale edecek olan lineer motor bloğu 0 V–10 V arası gerilimle çalışmaktadır. Gaz için ise ECU elektronik kontrol biriminden iki ayrı sinyal beklemektedir. Bunlardan ilki 1 V–4.5 V arasında diğeri ise 0.5 V–2.25 V arasındadır. Bu yüzden ikinci merkezin ürettiği analog sinyallerin alt ve üst seviyeleri bu sinyalleri referans değer olarak kullanacak sonraki sistemler için uygun hale getirilmelidir. Bu işlem bir yükselteç elemanı ile ve ofset değeri eklenerek gerçekleştirilecektir.

Kontrol işareti haline getirilmiş analog değerler birinci merkezi birimin denetiminde seçicilerden geçerek çıkışa gönderilecektir. Birinci merkezi birim kullanıcının isteğine göre çıkışa üretilen analog referans değerleri veya pedaldan gelen analog değerleri yazar. Bu işlem analog seçiciler ile yapılacaktır. Fakat her iki durumda da istenmeyen veya tehlikeli durumları belirlemek için merkezi birim (1) çıkışları sürekli ölçer. Bu işlemleri gerçekleştiren donanımın devre şeması Şekil 3.27’de verilmiştir.



Şekil 3.27 : Elektronik kart devre şeması.



Şekil 3.28 : Elektronik kart.

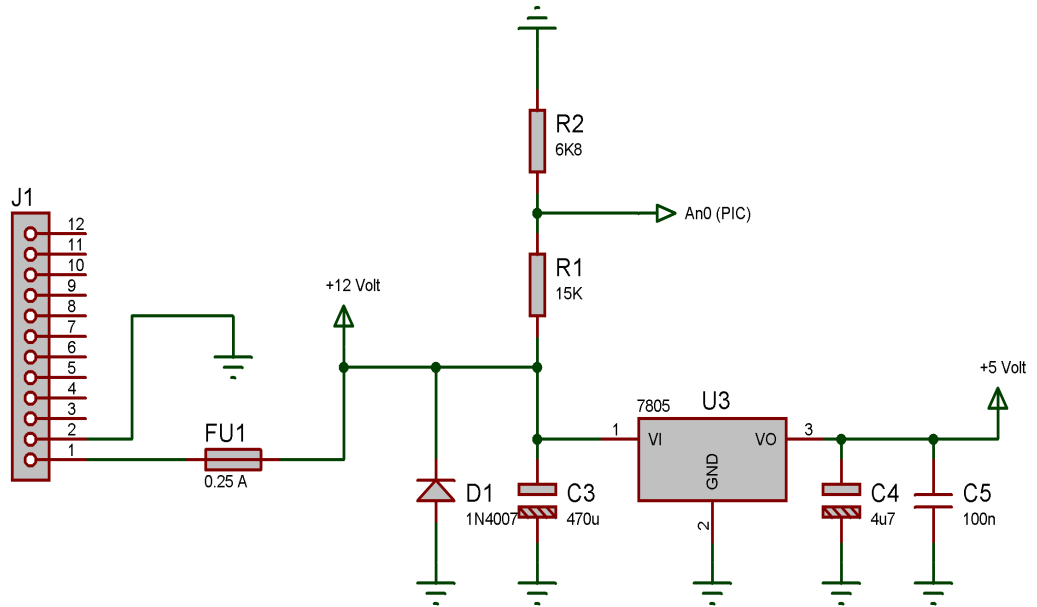
3.5.2.1 Besleme

Besleme devresinin girişı 1 ve 2 numaralı klemenslerdir. Gerilim değeri olarak 11 –14 V arası giriş yapılabilir. Giriş gerilimi birinci merkezi birim tarafından sürekli olarak ölçülür. Belirlenen sınırların dışına çıkmadığında sistem uyarı moduna geçer. Limitler daha fazla aşılsa sistem alarm durumuna geçer.

Sistemi yüksek akımdan ya da ters akımdan korumak için 250 mA’lık bir sigorta yerleştirilmiştir. Bu sigortanın değeri normal çalışma esnasındaki akıma göre hesaplanmıştır. D1 diyotu ise giriş gerilimi ters bağlandığında yüksek akım geçirir ve sigortayla beraber çalışarak beslemeyi keser. Bu devre elektronik kart üzerindeki diğer elemanları ters gerilimden korumak için yapılmıştır.

Kart üzerindeki elemanlar için besleme gerilimi olarak iki değeri gerekmektedir. Bunlar +12 V ve +5 V’dir. +12 V gerilim ile beslenen birimler bu konuda biraz esnektir. Bu yüzden sabit bir değeri elde etmek için ayrı bir regülatör kullanılmamıştır. Gerilimdeki olası küçük dalgalanmaları önlemek için ise kapasitör kullanılmıştır.

Mikrokontrolörlü veya mikroişlemcili sistemler için ise +5 V zorunludur ve iyi regüle edilmiş olması istenir. Hem devrede +5 V olmaması sebebiyle hem de bu gerilimin sabit olması gerektiğinden 7805 regülatörü kullanılmıştır. Çıkışında elde edilen +5 V gerilim diğer elemanlara bu terminalden dağılır.



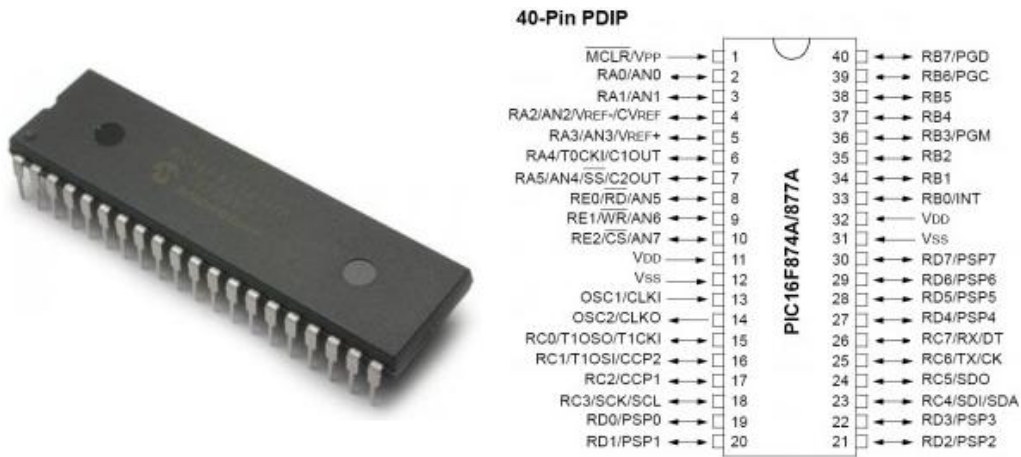
Şekil 3.29 : Besleme.

3.5.2.2 Merkezi birim (1)

Kart üzerindeki giriş çıkışları kontrol eden, güvenlik sebebiyle belirlenen noktalardan analog okuma işlemi yapacak olan elemandır.

Uygun mikrokontrolörü seçmek için görev tanımını açıkça belirlemek gerekir. Hangi özelliklere ihtiyaç duyulduğu sıralayacak olursa;

- Kart üzerinde 4 ayrı noktadan analog okuma yapılması gerekmektedir. Bunlar; kartın besleme gerilimi, gaz pedalına giden 2 analog sinyal ve fren pedalına giden analog sinyaldir. Bu okuma tek bir analog girişi olan mikro kontrolör ve analog bir seçici kullanarak yapılabilir. Böyle bir durumda seçiciyi yönetmek için mikrokontrolörün birkaç dijital girişinin bu iş için ayrılması gerekir. İkinci bir tercih de bu tasarımda olduğu gibi en az 4 analog girişi olan bir mikro kontrolör seçmektir.
- Mikro kontrolör ilk tasarım için en az 20 dijital giriş-çıkışa sahip olmalıdır. Bu portlar LCD sürmek için(6), butonlar için(5), uyarı lambaları için(4), seçiciler için(4) ve buzzer için(1) adet olarak kullanılacaktır.
- Kritik zamanlı verileri işleme yükü bu mikro kontrolöre ait olmadığı için çok yüksek hızlarda çalışmasına gerek yoktur.
- Kolay bulunabilir ve programlanabilir olması tercih edilmelidir.



Şekil 3.30 : PIC16F877 ve pin diyagramı.

Tüm bu özellikler göz önüne alındığında Microchip firmasının [30] ürettiği bir mikrokontrolör olan PIC16F877'nin bu iş için uygun olduğu belirlenmiştir. Elemanın genel özellikleri çizelge 3.2'de verilmiştir.

Çizelge 3.2 : PIC16F877 genel özellikleri.

Özellik	PIC16F877
Çalışma hızı	Max. 20Mhz
Program belleği	8 Kword Flash ROM
EEPROM belleği	256 Byte
Kullanıcı RAM	368 Byte
Giriş/Çıkış port sayısı	33
Zamanlayıcı	Timer1, Timer2, Timer3
A/D çevirici	8 kanal, 10 Bit
Çalışma voltaj aralığı	2,5 V – 5,5 V
Sink/Source akımı	25 mA / 20mA
Programlama sayısı	1000000

PIC için çalışma hızı 4 MHz olarak düşünülmüştür. Bu yüzden osilatör girişlerine 4 MHz'lik kristal ve kristale uygun kondansatörler bağlanmıştır. Kontrolörün bir buton ile yanlışlıkla resetlenmesi istenmediğinden reset butonu konulmamış, MCLR girişine +5 V verilmiştir.

Butonlar, göstergeler ve sesli uyarı elemanı kartın kullanıcı ile teması içindir ve tüm bu elemanlar PIC tarafından dijital giriş-çıkışlar kullanılarak yönetilir. Mikro kontrolörün dijital çıkışları ile yönettiği diğer blok ise analog seçicilerdir. PIC bu elemanları kullanarak çıkışlara giden bağlantı yollarını değiştirir. Böylece sistemi istediği moda sokabilir. Seçiciler mikro kontrolör tarafından 4 farklı şekilde konumlandırılır. Bunlar;

- Otonom mod konumu
- Manüel mod konumu
- Acil durum konumu
- Potansiyometre ile kontrol konumu

Mikrokontrolörün dijital portlarının yanı sıra analog girişlerinden bir kısmı da kullanılmıştır. Bu pinleri kullanım amacı kartın önemli noktalarındaki gerilimlerini ölçmektir. PIC, kart üzerindeki 4 farklı noktanın gerilimini ölçer. Tüm giriş-çıkışlara ait pin bağlantı diyagramı Çizelge 3.3'de gösterilmiştir.

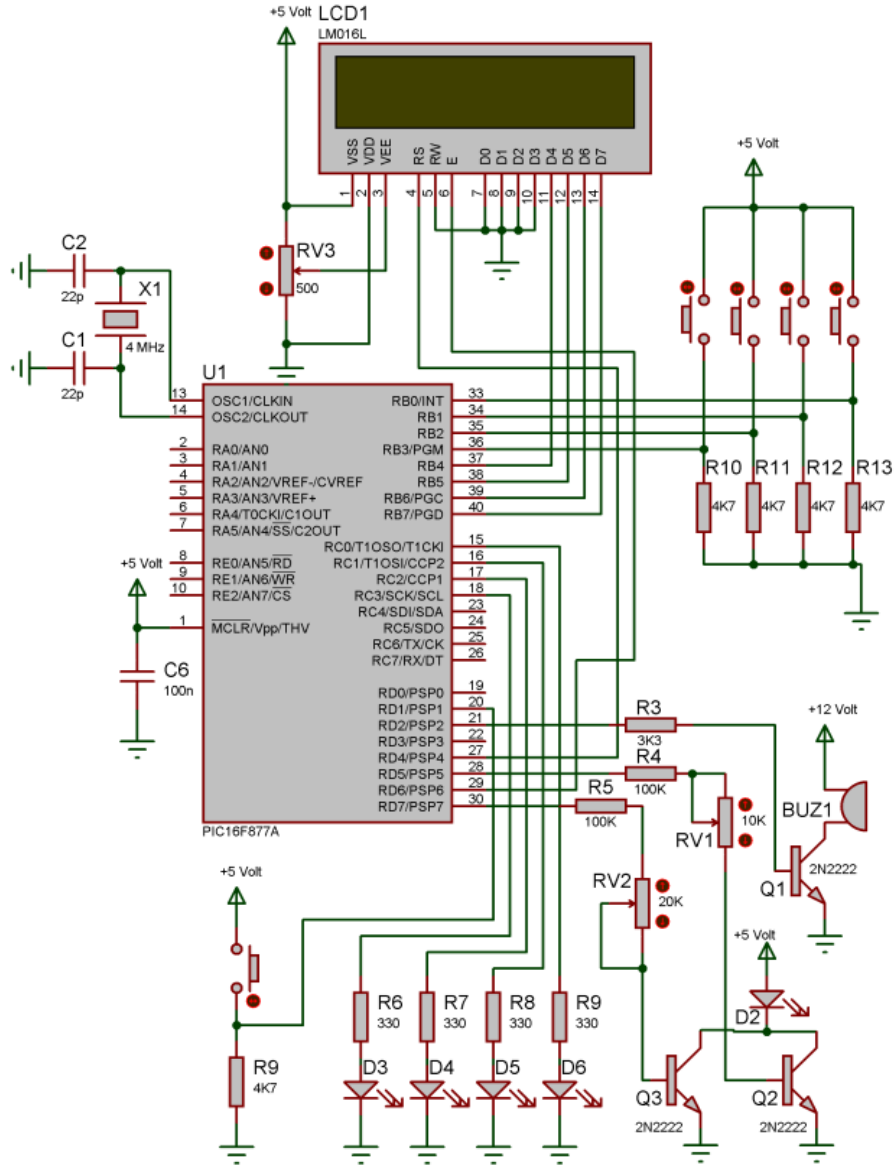
Çizelge 3.3 : Mikro kontrolör pin bağlantıları.

Port	Giriş / Çıkış	Bağlı olduğu eleman	Bağlı olduğu pin
A0	Giriş	Besleme Voltajı	-
A1	Giriş	Gaz Pedal Sinyali(Tam)	-
A2	Giriş	Fren Konumu	-
A3	Giriş	Gaz Pedal Sinyali(Yarım)	-
A4	Giriş	Fren Acil	-
A5	Giriş	Boş	-
B0	Giriş	Buton 4	-
B1	Giriş	Botun 3	-
B2	Giriş	Buton 2	-
B3	Giriş	Buton 1	-
B4	Çıkış	LCD	D4
B5	Çıkış	LCD	D5
B6	Çıkış	LCD	D6
B7	Çıkış	LCD	D7
C0	Çıkış	Kırmızı LED	-
C1	Çıkış	Beyaz LED	-
C2	Çıkış	Mavi LED	-
C3	Çıkış	Sarı LED	-
C4	Çıkış	Seçici 2	B, C
C5	Çıkış	Seçici 2	A, D
C6	Giriş	TX	-
C7	Giriş	RX	-
D0	Çıkış	Seçici 1	A,D
D1	Çıkış	Acil Durum Butonu	-
D2	Çıkış	Buzer	-
D3	Çıkış	Seçici 1	B,C
D4	Çıkış	LCD	RS
D5	Çıkış	LCD Işık	-
D6	Çıkış	LCD	Enter
D7	Çıkış	LCD Işık (Acil)	-
E0	Giriş	Boş	-
E1	Giriş	Boş	-
E2	Giriş	Boş	-

Butonlar, göstergeler ve sesli uyarı elmanı

Kullanıcı ile etkileşim için 5 adet buton kullanılmıştır. Bu butonlar normalde açık tiptedirler. Basıldıklarında lojik 1 sinyali oluşturmaları için mikro kontrolöre gitmeyen uçları +5 V'a bağlamıştır. Basılmadıklarında mikro kontrolör için lojik 0 üretmeleri için 4.7 K Ω 'luk pull-down dirençleri kullanılmıştır.

Butonlardan biri acil durum butonudur. Sistemin çalışması hangi konumda olursa olsun kullanıcı bu butona bastığı anda sistem mikro kontrolör tarafından acil durumuna getirilir. Diğer 4 butonun kendilerine ait kesin görevleri yoktur, çalışma anlarına göre farklı fonksiyonları vardır.



Şekil 3.31 : Kullanıcı ara yüz elemanlarının PIC'e bağlantısı.

Kart üzerinde ana gösterge elemanı olarak 2 satır 16 karakterlik LCD kullanılmıştır. LCD 4 data ve 2 kontrol (Enter & RS) pini ile sürülmektedir. Şekil 3.31'de de görüldüğü gibi LCD lambasını yakmak için iki ayrı transistör devresi vardır. İki adet farklı devre olması LCD lamba parlaklığının normal ve acil durumda farklı yanması içindir. Parlaklık ayarları transistörlerin baz girişlerindeki potansiyometrelerle yapılır.

Diğer gösterge elemanları 4 farklı renkteki ledlerdir. Her birinin belirttiği durum çizelge 3.4'de gösterilmiştir. Gerekliğinde kullanıcıyı sesli olarak uyarmak için ise bir buzzer kullanılmıştır. Bu eleman için +12 V besleme gerektiğinden PIC bu elemanı bir transistör yardımıyla kontrol etmektedir.

Çizelge 3.4 : Gösterge lambaları durumları.(X:1, X:0)

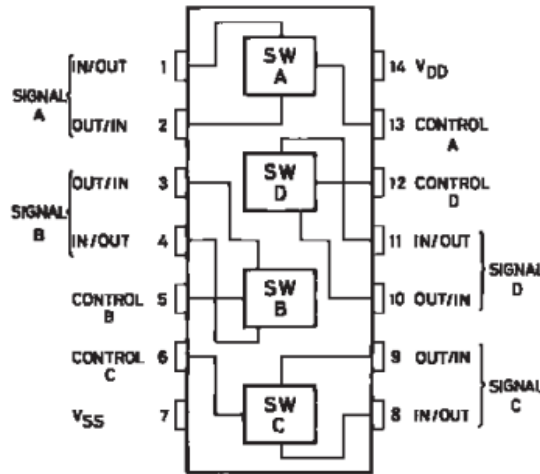
LED	K.Bilgisayar	K.Pedal	K.Potans	Uyarı	Acil
Beyaz	1	0	1	X	X
Mavi	0	1	1	X	X
Sarı	X	X	X	1	X
Kırmızı	0	0	0	0	1

Seçici Devre

PIC'in kontrolü altında çıkışa istenen sinyalleri yönlendirmeyi sağlayan devredir. Bilgi seçici olarak analog multiplexer 4066 kullanılmıştır. Şekil 3.32'de görüldüğü gibi her seçicinin içinde dört adet kontak bulunur ve her kontak açılıp kapanması o kontakta ait kontrol pini ile sağlanır.

Elektronik kart üzerindeki secici devre sekiz kontak içerir ve her kontak PIC'in dijital çıkışları ile kontrol edilir. Seçici devre 4 farklı şekilde konumlandırılır.

Manuel durum için gaz pedalından gelen sinyaller ECU'ye gönderilir. İkinci merkezi birimde gaz için üretilen referans sinyalin kontakı açık tutularak çıkışa verilmez. Yine bu durumda fren sinyali devreden çıkarılır. Lineer motor kontrolcüsüne kapat (lineer motoru devre dışı bırak) sinyali gönderilir.

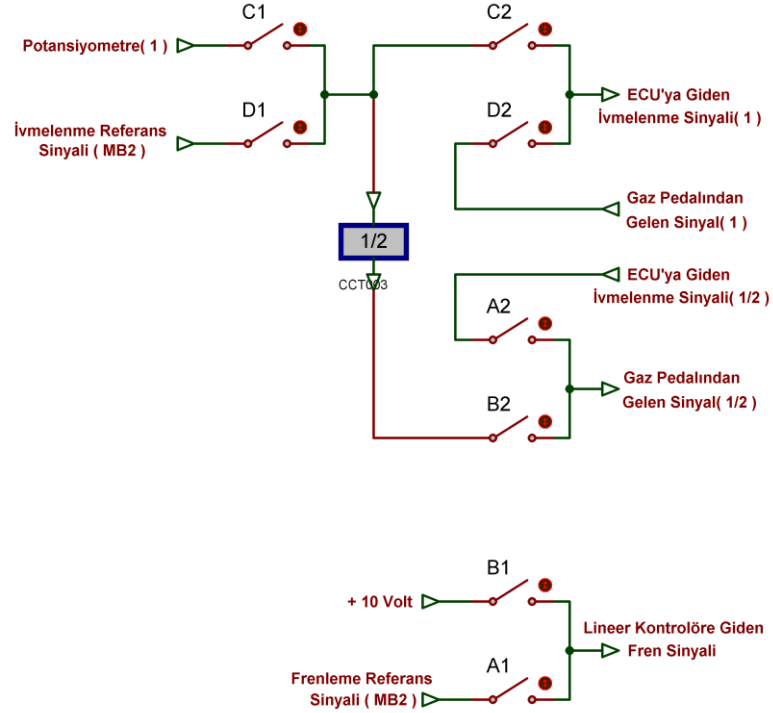


Şekil 3.32 : 4066 pin diyagramı [31].

Otonom konumda ise kontaklar çıkışa bilgisayarın ürettiği ve merkezi birim (2) ile analog devre üzerinden gelen kontrol işaretlerini gönderecek şekilde bağlanır. Bu hem fren hem ivmelenme için yapılır. Gaz pedalından gelen sinyaller veya potansiyometreden gelen gerilim açık duruma getirilir, çıkışa verilmez.

Acil durum konumunda ivmelenme için ECU'ye gaz pedalından gelen sinyaller gönderilir. Fakat fren için aynısı düşünülmemektedir. Fren kontrolcüsüne ne kontrol

sinyali ne de kapat komutu gönderilir. Seçici devre lineer motor açık iken +10 V referans sinyali çıkışa bağlar. Bu fren pedalına tam basmak anlamına gelir. Özetle gaz pedalı kullanıcıya bırakılır, ama fren pedalı tam basılı konumuna getirilir.



Şekil 3.33 : Seçicilerin kontak bağlantıları

Son durum ise potansiyometre ile kontrol etmek içindir. Bu durumda seçici devre potansiyometrenin orta ucu gerilimini ivmelenme referansı olarak çıkışa verir, yani ECU' ye gönderir. İkinci merkezi birimin ivmelenme için ürettiği referans değer veya gaz pedalından gelen değerler açığa alınırlar. Fren için ise lineer motor sürücüsüne kapat komutu gider. Bu yüzden fren için giden referans değerın önemi yoktur. Seçicilerin kontak bağlantısı Şekil 3.33'de verilmiştir ve Çizelge 3.5'deki gibi konumlanır.

Çizelge 3.5 : Kontak durumları (K: Kontak iletimde, A: Kontak kesimde).

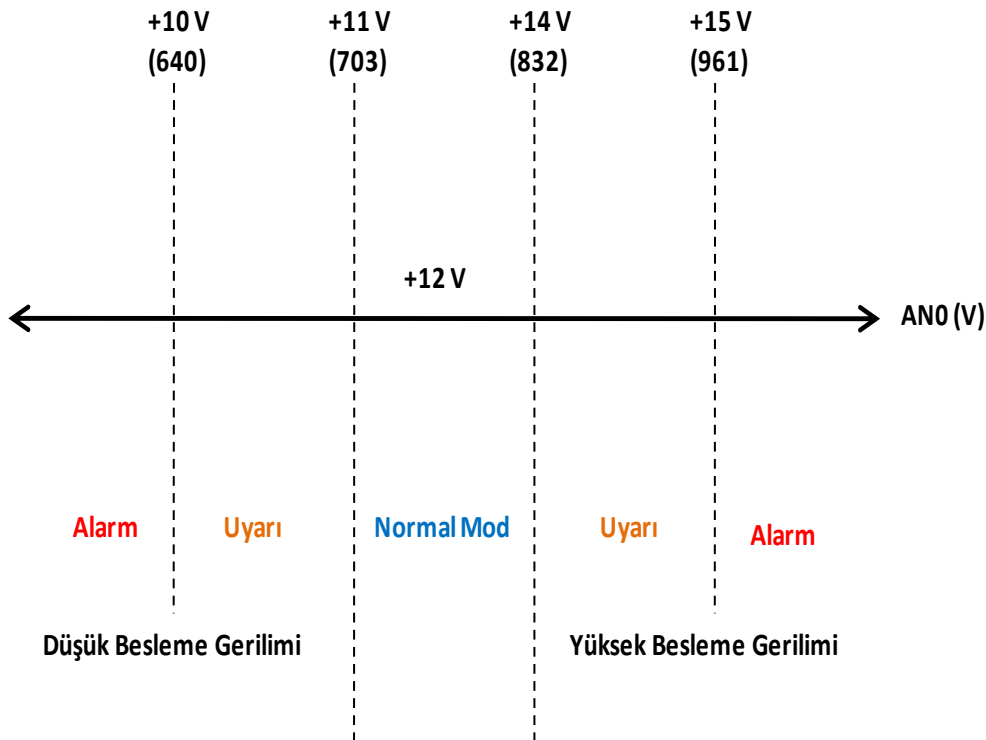
Durumlar	A1	B1	C1	D1	A2	B2	C2	D2
Otonom K.	K	A	A	K	A	K	K	A
Manüel K.	K	A	A	K	K	A	A	K
Potans K.	A	K	K	A	A	K	K	A
Acil	A	K	K	A	K	A	A	K

Analog girişler

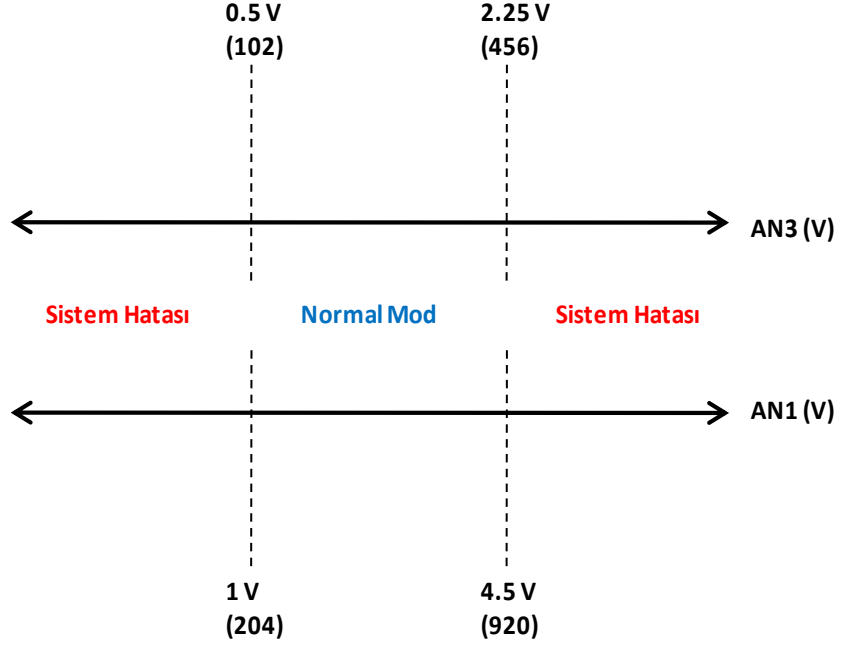
Kullanılan mikro kontrolörün üzerinde 8 adet analog giriş kanalı bulunmaktadır. Bu kanallar minimum 0 V maksimum 5 V olmak üzere ölçüm alabilirler. Ölçüm aralıkları V_{+REF} ve V_{-REF} değerleri ile oynayarak değiştirilebilir. Bu noktada belirtilen özelliğe ihtiyaç duyulmamış, tüm ölçümlerin aralığı varsayılan değer olan 0 – 5 V olarak alınmıştır. PIC içindeki analog çeviriciler 10 bitlik olduğu için her kanaldan okunan değer $0-1023(2^{10}-1)$ arasında olacaktır. Yani 0 V değerine karşılık 0 sayısı, +5 V değerine karşı 1023 sayısı elde eldir. Ara değerlerdeki değişim lineerdir.

PIC kart üzerinde dört ayrı noktadan analog değer okur. Bu okuma işini periyodik olarak gerçekleştirmektedir. Analog değer okumasının iki temel amacı vardır. En temel amaç güvenliktir. Değer alınan terminallerden herhangi birindeki değer olağan dışına çıkarsa uyarı veya alarm sinyali üretir. Aşağıdaki şekillerde tüm analog girişlerin sınır değerleri (hem analog gerilim değeri olarak hem de PIC içinde kullanılan dijital değeri) çalışma modları ile birlikte verilmiştir.

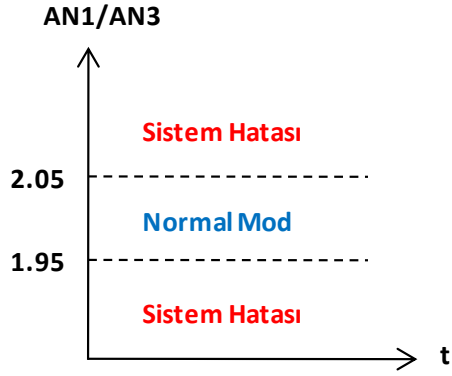
AN1 ve AN3 ölçtükleri gerilim noktalarına direkt bağlıdırlar. Fakat AN0 ve AN2 gerilim bölücü dirençlerle bağlıdırlar. Bunun sebebi bu noktalardaki voltaj değerlerinin +5 V'un üstüne çıkabilmesidir.



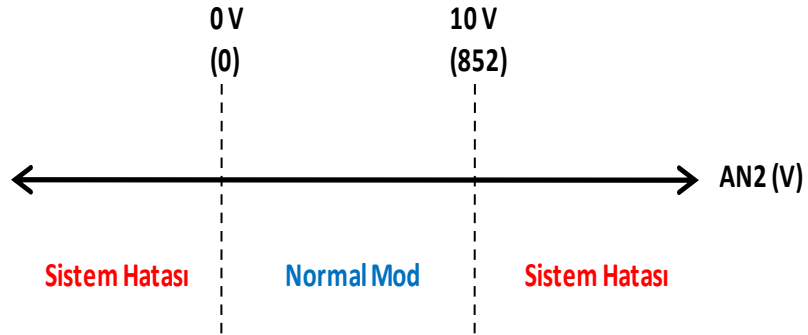
Şekil 3.34 : Besleme girişi (sınır değer kontrolü).



Şekil 3.35 : ECU'ye giden sinyaller (sınır değer kontrolü).



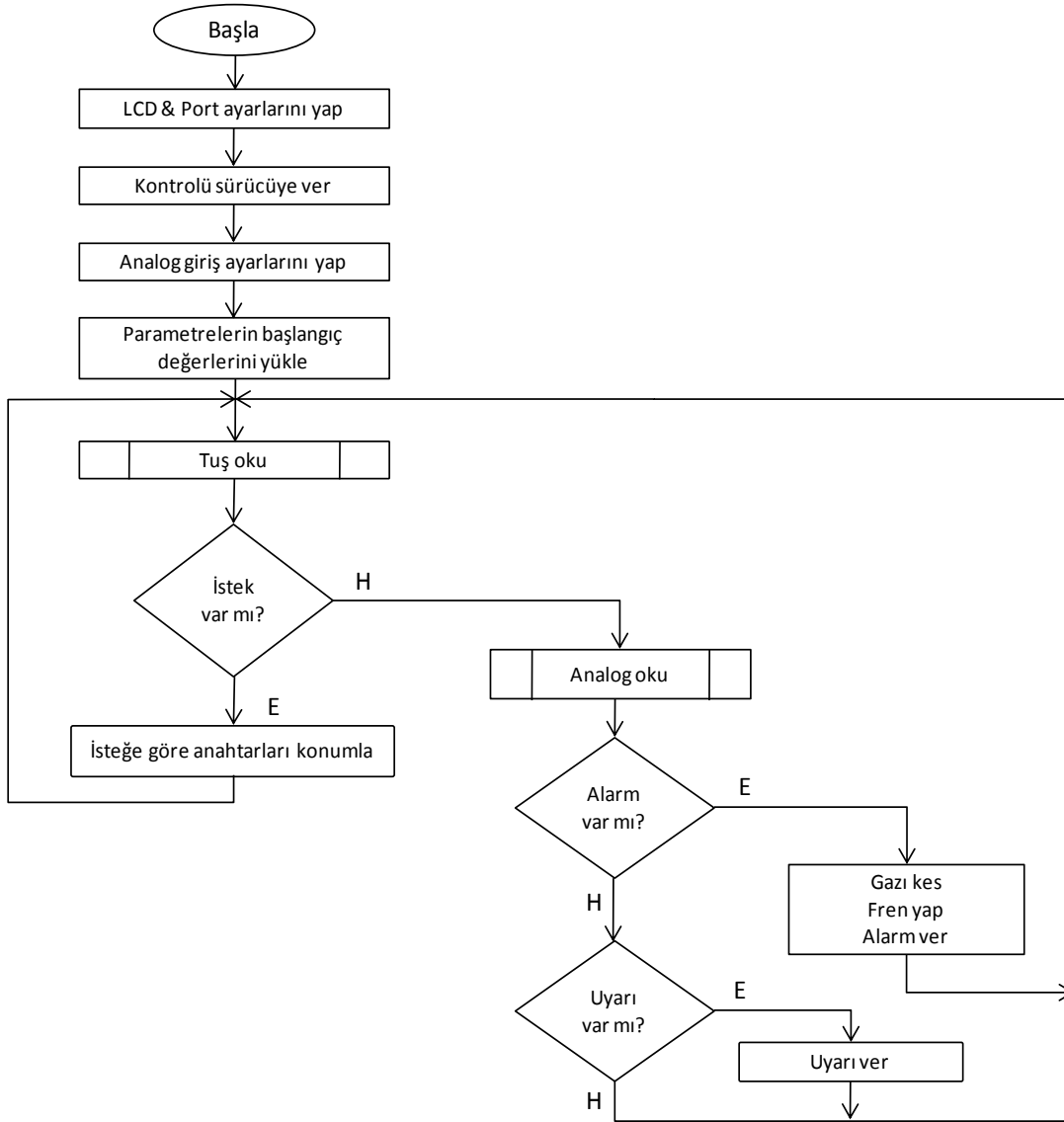
Şekil 3.36 : ECU'ye giden sinyaller (oransal kontrol).



Şekil 3.37 : Fren motoru sürücüsüne giden sinyal (sınır değer kontrolü).

PIC16F877'nin programlanması

Bu mikrokontrolörün içindeki yazılım Microcode Studio ortamında PicBasic dili kullanılarak geliştirilmiştir ve kodu Ek-A'da verilmiştir. Bu kodun genel yapısını veren akış diyagramı Şekil 3.38'deki gibidir.



Şekil 3.38 : PIC16F877'ye yüklenen kodun akış diyagramı.

Akış diyagramı incelendiğinde mikrokontrolörün sürekli bir çevrim içersinde buton takımını gözlemlediği ve analog girişleri okuduğu görülür. Analog girişlerde olağanüstü durumlar ortaya çıktığında uyarı yada alarm durumuna geçmektedir. Her iki durumda alınan önlemler birbirinden farklıdır. Bunun yanında kullanıcı talepleri doğrultusunda gaz ve fren sisteminin kontrolünü anahtarlama yaparak bilgisayara veya sürücüye vermektedir.

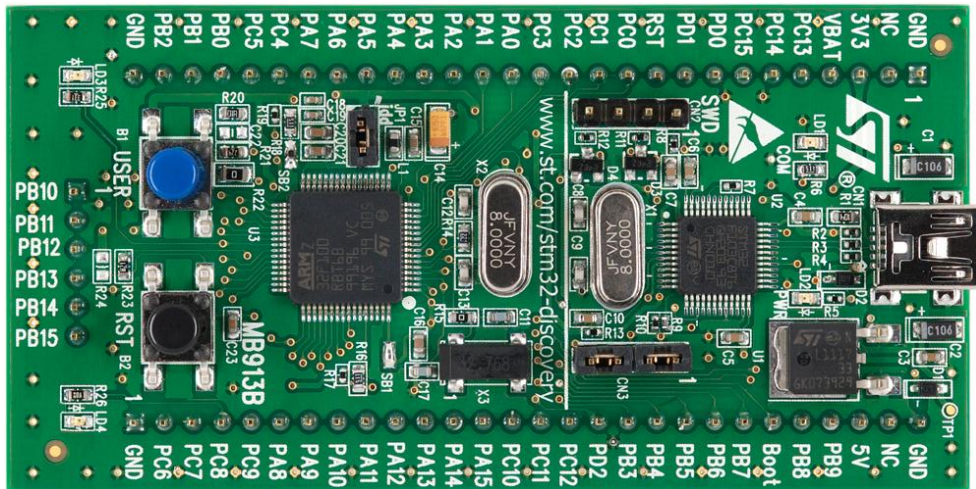
3.5.2.3 Merkezi birim (2)

PIC16F877'nin dışında ikinci bir merkezi birimin kullanılmasının temel sebebi bilgisayarda hesaplanan kontrol işaretinin en kısa sürede, zaman kaybı olmadan sisteme gönderilme gerekliliğidir. Çünkü PIC16F877 mikrokontrolörü üzerinde güvenlik ve kullanıcı ile arayüz oluşturma gibi birçok iş yükü bulunmaktadır. Bu iş yükü kontrol işaretinin bir sonraki sisteme iletilmesini geciktirebilir. Böyle bir gecikme hem fren hem de gaz için olan kontrolü zorlaştıracaktır.

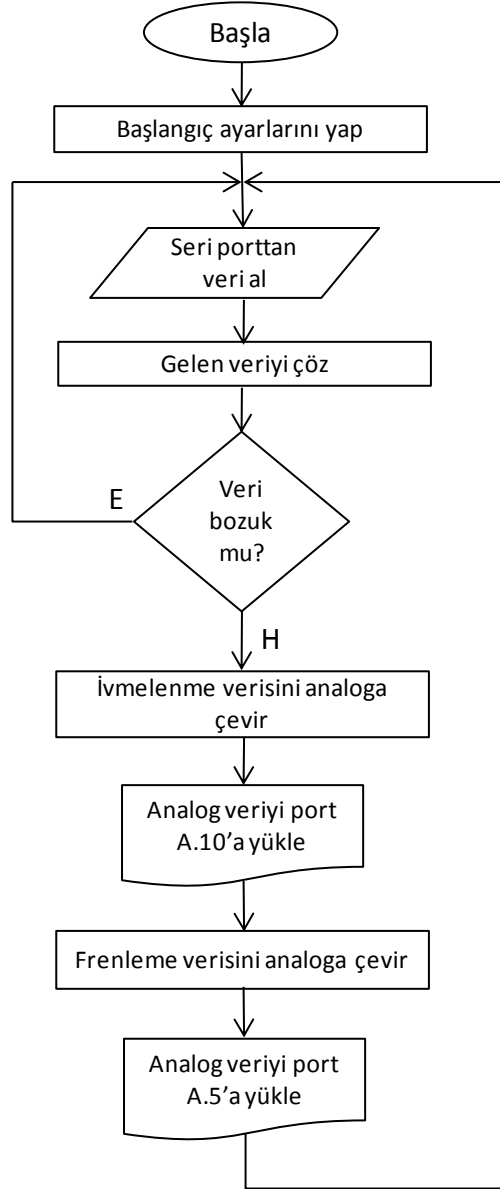
Diğer bir sebep ise PIC16F877'nin analog çıkışa sahip olmamasıdır. Dijital çıkışın sonuna bir DAC ekleyerek analog bir yapı oluşturulabilir ama bu durumda dijital çıkışlar yetersiz kalabilir ve tasarlanan devre karmaşık hale gelebilir. Tüm bu sebeplerden dolayı PIC16F877'nin yanında ST [32] marka STM32VLDISCOVERY model analog çıkışlara da sahip ikinci bir merkezi birim kullanılmıştır. Bu kartın görevi seri porttan gelen veri paketinin doğruluğunu sınamak ve paketteki kontrol işaretlerini dijitalden analoga çevirip ayrı ayrı port çıkışlarından bir sonraki birim olan analog devreye göndermektir.

Çizelge 3.6 : Merkezi birim 2'nin özellikleri.

Özellik	STM32VLDISCOVERY
Giriş/Çıkış port sayısı	50
Program belleği	128 KB Flash memory
RAM	8 KB
Çalışma voltaj aralığı	5 V – 3,3 V
Buton	1 adet
Üzerindeki ledler	Mavi, yeşil



Şekil 3.39 : Merkezi birim 2 [33].



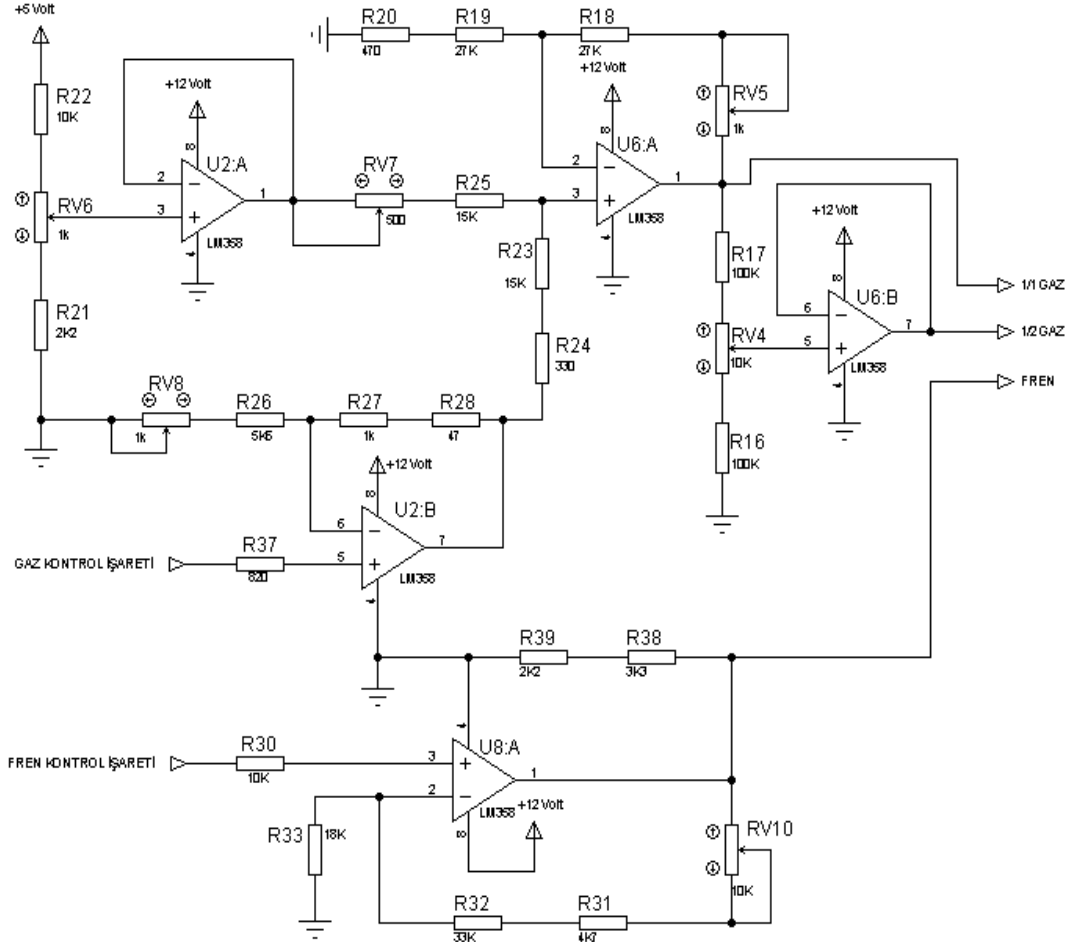
Şekil 3.40 : STM32VLDISCOVERY kart yazılımının akış diyagramı.

3.5.2.4 Analog devre

Bilgisayarda bulunan kontrol algoritmasında gaz ve fren için üretilen kontrol işaretleri 12 bitliktir. Diğer bir deyişle 0 - 4095 arasındadır. Oluşturulan bu sayısal değerler STM32VLDISCOVERY yardımıyla 0-3 V aralığında analog değerlere dönüştürülürler. Fakat araç üzerindeki sistemler incelendiğinde gaz için birbirleri arasında $\frac{1}{2}$ orana sahip iki ayrı analog sinyale ihtiyaç duyulduğu görülür (0.5-2.25 V ve 1-4.5 V aralığında). İhtiyaç duyulan bu işaretlerin hem ofset değerleri, hem aralıkları hem de adetleri merkezi birim (2)'nin ürettiğinden farklıdır.

Fren için ise tasarlanan elektromekanik sisteme ait elemanların özelliklerinden dolayı pedal hareketi için 0-10 V arasında bir sinyale ihtiyaç duyulmaktadır. Bu

gerilim aralığının STM32VLDISCOVERY'in ürettiği aralıktan farklı olduğu için mikrokontrolör ve bir sonraki sistem arasında analog bir devre dizayn edilmiştir. Şekil 3.41 'de de görüldüğü gibi bu analog devrede OP-AMP'lar kullanılarak gerilim aralıkları istenilen değerlere çekilmekte ve gerekli ofset gerilimleri eklemektedir.



Şekil 3.41 : Analog devre şeması.

3.5.2.5 Ara yüzler

Bilgisayarla STM32VLDISCOVERY'nin haberleşmesi için RS-232 ve USB olmak üzere iki ayrı haberleşme türü kullanılmıştır. İlk aşamada STM32VLDISCOVERY ile haberleşebilmek için Max232 seri port bufer entegresi kullanılarak RS-232 arayüzü oluşturulmuş ve bilgisayarla bağlantı sağlanmıştır. STM32VLDISCOVERY 'nin portlarından ikisinin RX / TX uçları olarak özelleştirilmiş olması bu bağlantıda kolaylık sağlamıştır. Ancak projenin ileriki aşamalarında elektronik kart ile bağlantı kuran tüm bilgisayarlarda seri port girişi olmadığı için USB arayüzüne ihtiyaç duyulmuştur. Bu arayüzü oluşturmak için MSP430 elektronik kartı programlanmış

ve kullanılmıştır. Bu kart USB arayüzüyle bilgisayara, RX/TX uçlarıyla ise STM32VLDISCOVERY'e bağlanarak bir köprü görevi görmüştür.

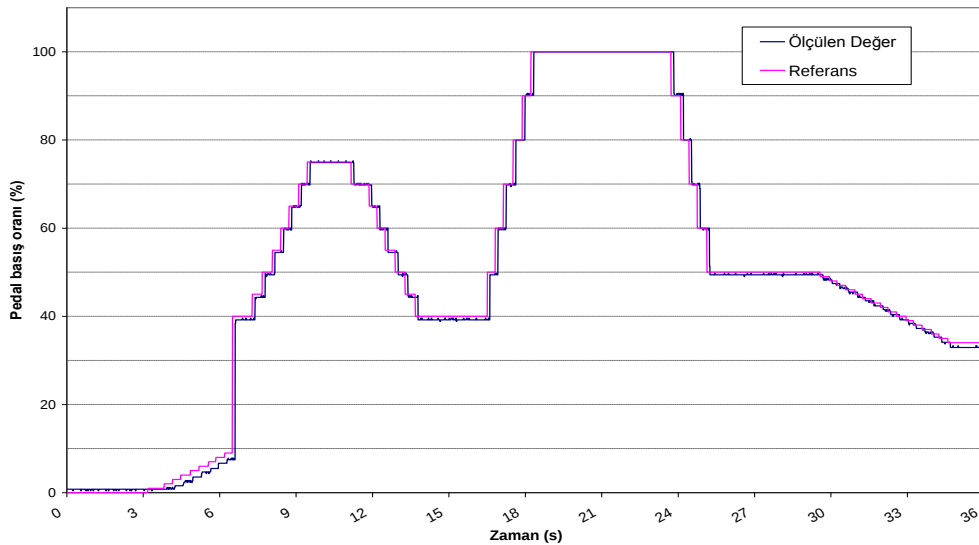


Şekil 3.42 : MSP430.

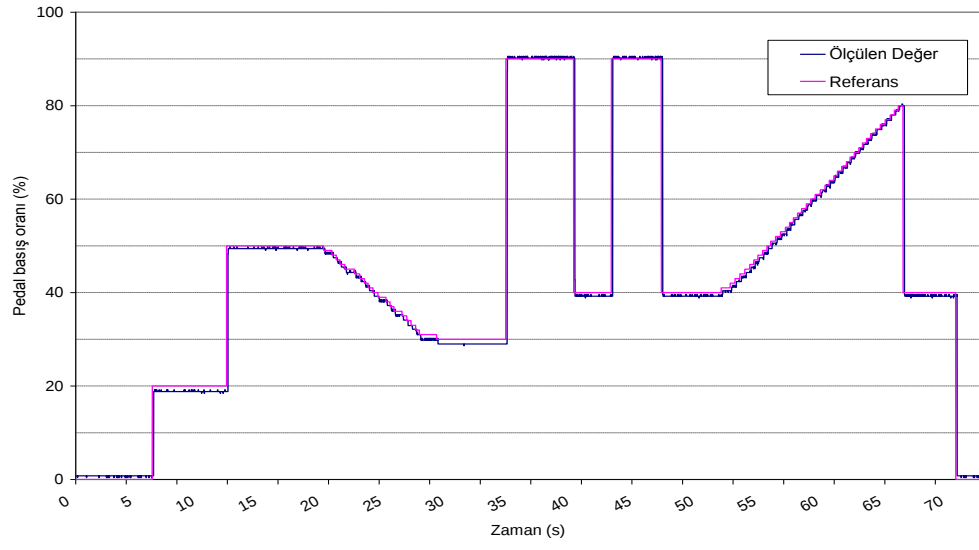
3.5.3 Yapılan deneyler

Hız kontrolörü için kontrol katsayıları henüz hesaplanmamış olsa da daha basit bir algoritmayla sistem test edilmiştir. Bu test esnasında ivmelenme sistemi için kontrol işareti klavyeden girilmiş, gaz pedalının konumu ise CAN-Bus hattından gerçek zamanlı olarak okunmuş ve kaydedilmiştir. Aşağıdaki grafiklerde referans kontrol sinyali ile ölçülen kontrol sinyalinin küçük bir faz kaymasıyla birebir örtüştüğü görülmektedir. Bu faz kaymasının oluşmasında bilgisayarda çalışan kodun işleme süresi, elektronik karta gelen verilerin doğrulanması için harcanan süre, ECU'de verilerin işlenmesi, CAN'de yayınlanma süresi gibi birçok etken vardır.

Sonuç olarak hem bilgisayar tarafında hem de elektronik kart tarafında tasarlanan sistemin doğru bir şekilde çalıştığını görülmektedir.



Şekil 3.43 : Gaz pedalı basış oranı deney 1.



Şekil 3.44 : Gaz pedalı basış oranı deney 2.

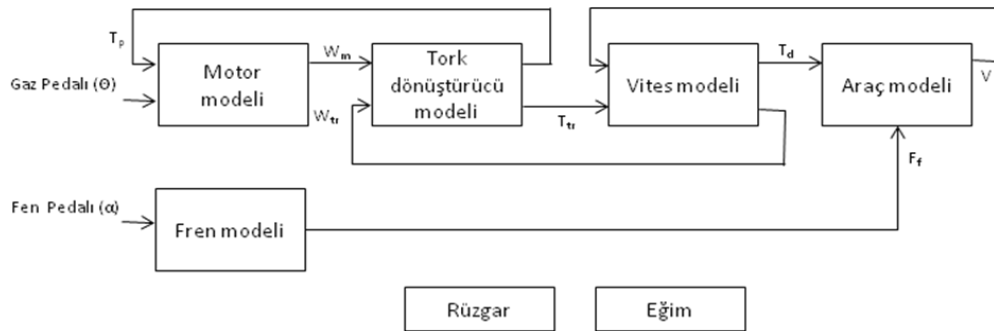
4. BOYUNA ARAÇ MODELİNİN ÇIKARTILMASI ve MATLAB SİMÜLASYONLARI

Hız kontrolörüne ait katsayıların bulunması için öncelikli olarak matematik modelinin çıkartılması ve bu modele göre simülasyonların yapılması gerekmektedir. Elde edilecek modelde boyuna dinamik denklemlerin olması yeterlidir. Çünkü bu tez kapsamında aracın sadece boyuna kontrolü yapılacak olup, yanal etkiler ihmal edilecektir.

Bu bölümde ilk olarak aracın hızlanmasına etki eden kuvveti üreten motor ve üretilen bu kuvveti tekerleklerle kadar taşıyan vites kutusu, diferansiyel, tork çevirici vb. aktarma organlarının matematik modeli çıkartılmıştır. Ardından aracın yavaşlamasına ya da durmasına sebep olan fren sistemi modele eklenmiştir. Ayrıca model son halini alırken yuvarlanma direnci, rüzgâr direnci, yol eğimi vb. etkenler de göz önünde bulundurulmuştur. Daha sonra elde edilen boyuna matematik modelin doğruluğunu görebilmek için Matlab programında Simulink modeli oluşturulmuş ve farklı senaryolar için bu model test edilmiştir.

4.1 Boyuna Araç Modeli

Boyuna model çıkartılırken ilk olarak aracın pozitif ivmelenmesi için gereken güç (T_m) üreten motor elemanına bakmak gerekir. Motorun çıkışında ne kadar tork olması gerektiği hız kontrolörünün girişlerinden biri olan gaz pedalı basış oranına ve motorun o andaki devrine (ω_m) bakılarak ECU tarafından karar verilir.



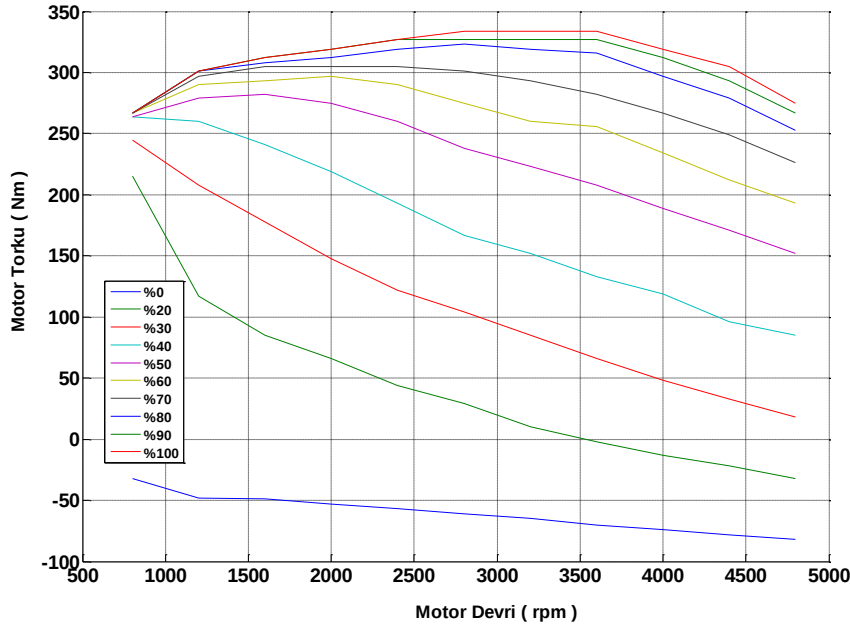
Şekil 4.1 : Boyuna araç modelinin genel blok diyagramı[4].

Motorun çıkış torkuyla gaz pedalı ve motor devri arasında fonksiyonel bir bağlantı vardır.

$$T_M = f(\omega_m, \Theta) \quad (4.1)$$

Θ : Gaz pedalı konumu

Bu fonksiyonun giriş değerleri ile çıkış değerleri arasındaki ilişki motorun karakteristiğini belirler ve motor haritası olarak adlandırılır. Motor haritası üretilen motora özgü olup üretici firma tarafından belirlenir. Otonom otomobilin motoruna ait harita grafiksel olarak Şekil 4.1’de verilmiştir.



Şekil 4.2 : Motor haritası.

Motor haritası incelendiğinde dizel motorun dönüş hızının gaz pedalı hangi konumda olursa olsun minimum 800 devir/dakika olduğu görülmektedir. Bu motorun rölanti devridir. Motor torkunun maksimum olduğu noktanın ise pedal basış oranına göre değiştiği görülmektedir. Örneğin pedala %100 basıldığında maksimum tork 3000–3500 devir/dakika arasında elde edilir. Motor haritasından da anlaşılabileceği gibi motor sadece pozitif tork üretmekle kalmayıp, bazı motor devirlerinde negatif tork da üretmektedir. Aracın momenti motoru daha hızlı devirlerde çalışmaya zorladığında motor araca göre yavaş kalacağından aktarma organları aracılığıyla tekerleklerle negatif tork iletir. Böylece araç fren pedalına basılmadan motor sayesinde yavaşlar. Bu da motor freni olarak da adlandırılır.

Denklem 4.2'deki eşitlik dizel motorun matematik modelini vermektedir. Bu eşitlik Şekil 4.1 ile beraber incelendiğinde Θ 'nın gaz pedalı konumu olarak modele etki ettiği, T_p 'nin ise motora dışarıdan binen ve tork çeviriciden gelen yük olduğu görülmektedir.

$$\dot{\omega}_m = \frac{1}{J_M} [T_m(\omega_m, \Theta) - T_a(\omega_m) - T_p(\omega_m, \omega_{tr})] \quad (4.2)$$

ω_{tr} : Türbin dönüş hızı

T_a : Motordaki elemanların sürtünmeye harcadığı tork

J_M : Motordaki hareketli parçaların eşdeğer ataleti

Eşitlik 4.3'de tork çeviricinin pervane girişindeki tork, eşitlik 4.4'de ise tork çeviricinin türbin çıkışındaki vites kutusu dişlisine giden tork miktarının matematiksel ifadesi verilmiştir.

$$T_p = 2m_1\omega_{m0}\omega_m + m_2\omega_{m0}\omega_{tr} + 2m_3\omega_{tr0}\omega_{tr} \quad (4.3)$$

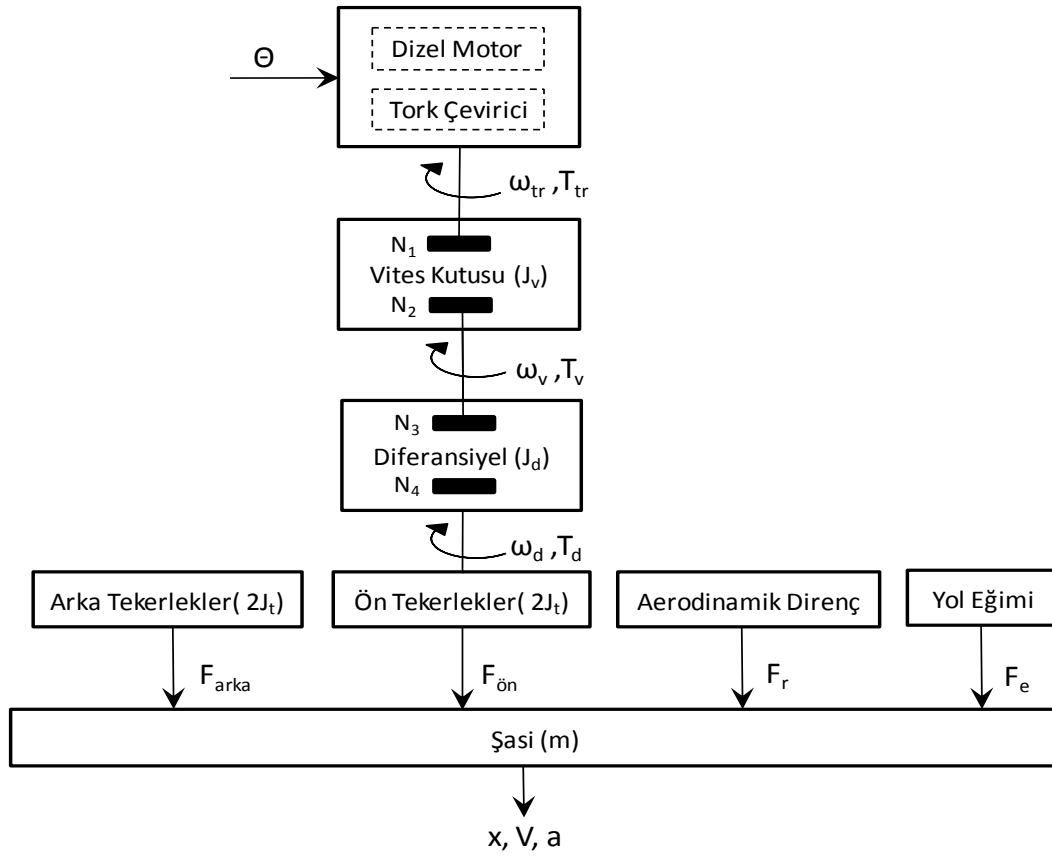
$$T_{tr} = 2n_1\omega_{m0}\omega_m + n_2\omega_{m0}\omega_{tr} + 2n_3\omega_{tr0}\omega_{tr} \quad (4.4)$$

T_{tr} : Türbin torku

T_m : Dizel motorun ürettiği tork

$m_{1,2,3}$: Tork dönüştürücü dönüşüm katsayıları (giriş yönünde)

$n_{1,2,3}$: Tork dönüştürücü dönüşüm katsayıları (çıkış yönünde)



Şekil 4.3 : Güç aktarma organları.

J_v : Vites kutusu içinde yer alan dönen parçaların eşdeğer ataleti

ω_d : Diferansiyel dönüş hızı

T_d : Diferansiyel çıkışında oluşan kuvvet

J_d : Diferansiyel kutusu içinde yer alan dönen parçaların eşdeğer ataleti

$N_{1,2,3,4}$: Giriş ve çıkış dişli sayıları

$F_{ön}$: Ön tekerleklerden yere aktarılan kuvvet

F_{arka} : Arka tekerleklerden yere aktarılan kuvvet

F_e : Yol eğiminin oluşturduğu kuvvet

F_r : Rüzgar kuvveti

m : Araç kütlesi

x : Alınan yol

a : Araç ivmesi

Şekil 4.3’de de görüleceği gibi tork çeviriciden çıkan güç vites kutusunda k_v oranında arttırılarak bir sonraki sisteme aktarılır. Açısal hız ise $1/ k_v$ oranında azalarak vites kutusundan çıkar. Vites durumuna göre değişen k_v katsayıları Çizelge 4.1’de verilmiştir.

$$k_v = \frac{N_2}{N_1} \quad (4.5)$$

$$\dot{\theta}_v = \frac{\dot{\theta}_{tr}}{k_v} = \omega_v \quad (4.6)$$

$$T_v = \eta_v k_v T_{tr} \quad (4.7)$$

ω_v : Vites kutusunun çıkışındaki açısal dönüş hızı

T_v : Vites kutusu çıkış torku

η_v : Vites dişlisi verimi

Çizelge 4.1 : Vites çevrim oranları.

Vites durumu	Vites çevrim oranı (k_v)
1	4.273
2	2.238
3	1.444
4	1.029
5	0.767

Vites kutusundan çıkan tork diferansiyele gönderilir. Burada k_d oranında değişerek farklı bir eksen de ön tekerlere gönderilir. Diferansiyel çevrim oranı (k_d) her vites için aynıdır, değişmez. Bu oran otonom araçta 3.733’dür.

$$k_d = \frac{N_4}{N_3} = 3.733 \quad (4.8)$$

$$\dot{\theta}_d = \frac{\dot{\theta}_v}{k_d} = \omega_d \quad (4.9)$$

$$T_d = \eta_d k_d T_v \quad (4.10)$$

η_d : Diferansiyel verimi

Hem vites kutusundan geçiş esnasında hem de diferansiyel üzerinden geçerken, aktarılan güçte verimin %100 olmamasından dolayı bir kayıp meydana gelir. Bu kayıplar denklem 4.10 ve 4.7’de görülmektedir.

$$\dot{\theta}_d = \frac{\dot{\theta}_{tr}}{k_v k_d} \quad (4.11)$$

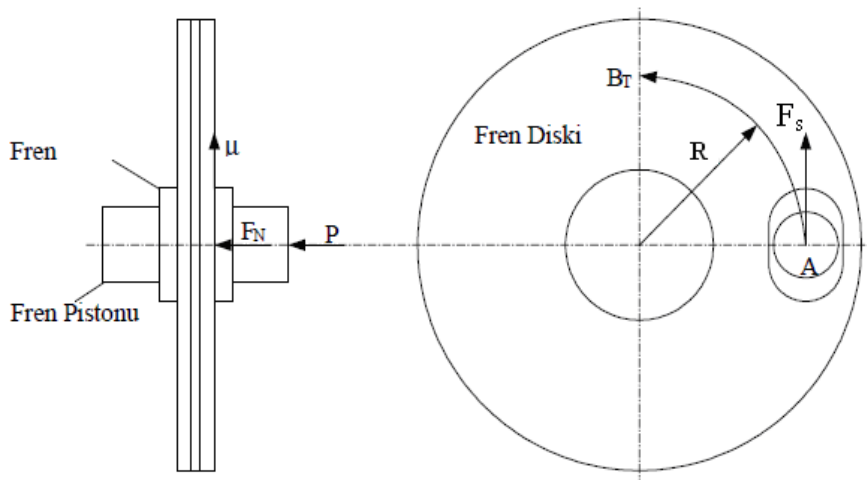
$$T_d = \eta_v \eta_d k_v k_d T_{tr} \quad (4.12)$$

$$T_{1,2} \cong \frac{T_d}{2} \quad (4.13)$$

Diferansiyelden gelen torku tekerlerin yerle temas ettiği noktada oluşan kuvvete indirgeyecek olursak,

$$F_{1,2} = \frac{T_{1,2}}{r_{1,2}} \quad (4.14)$$

Ön tekerlere etki eden bu kuvvetin yanında bir de fren kuvveti vardır. Şekil 4.4’den yararlanılarak bu kuvvet hesaplanacak olursa;



Şekil 4.4 : Fren disk [34].

$$F_{f1,f2,f3,f4} = \frac{\mu P_{1,2,3,4} A_f n R}{r_{1,2,3,4}} \quad (4.15)$$

olarak bulunur.

$$F_{\ddot{O}N} = (F_1 - F_{f1})(1 - s_1) - F_{y1} + (F_2 - F_{f2})(1 - s_2) - F_{y2} \quad (4.16)$$

s : Kayma miktarı

$F_{f1,f2}$: Ön tekerleklerde oluşan fren kuvveti

$F_{y1,y2}$: Ön tekerleklerde harcanan yuvarlanma kuvveti

$F_{\ddot{O}N}$ iki ön tekerin aracı hareketlendirmek için şasiye verdikleri net kuvveti temsil etmektedir. Eşitlik 4.16 incelendiğinde motordan gelen kuvvet ile fren sisteminden gelen kuvvet arasındaki farkın alındığı görülür. Ön tekerlerin yerle temas noktalarında oluşan kuvvet bulunurken kayma miktarı göz önünde bulundurulmuştur. Her iki tekerin araca aktardığı net pozitif veya net negatif kuvveti hesaplarken iki tekerin de yuvarlanma direnci, yerle temas edilen noktada oluşan kuvvetten çıkartılmıştır.

$$F_{ARKA} = (F_{f3})(1 - s_3) + F_{y3} + (F_{f4})(1 - s_4) + F_{y4} \quad (4.17)$$

$F_{f3,f4}$: Arka tekerleklerde oluşan fren kuvveti

$F_{y3,y4}$: Arka tekerleklerde harcanan yuvarlanma kuvveti

Araç önden çekişli olduğu için arka tekerleklerde motordan gelen kuvvetler denklem 4.17'ye dâhil edilmemiştir.

Frenleme esnasında aracın hızı (V), tekerin çevresel hızına göre (V_c) daha yavaş olacaktır. İvmelenme esnasında ise tekerin çevresel hızı, araç hızından daha fazla olacaktır. Kayma ise araç hızı ve tekerleğin çevresel hızı arasındaki bu farktan kaynaklanır.

$$s = \begin{cases} \frac{V - V_c}{V} & V \geq V_c \\ \frac{V - V_c}{V_c} & V < V_c \end{cases} \quad (4.18)$$

Tekerlek bir tur attığında yuvarlanma çevresi kadar yol alır. Bu yuvarlanma çevresi hesaplanırken tekerlek hareket ettiği için dinamik yarıçap (R) kullanılır. Denklem 4.20'deki gibi yuvarlanma çevresi (U) tekerleğin devir sayısı ile çarpıldığında tekerleğin çevresel hızı bulunabilir.

$$U = 2\pi R \quad (4.19)$$

$$V_c = Uf_c = 2\pi Rf_c = R\omega_c \quad (4.20)$$

f_c : Tekerleğin dönüş frekansı

ω_c : Tekerleğin açısal dönüş hızı

Tekerlekler üzerlerine binen yükten dolayı daire şeklini kaybederler ve Şekil 3.14 görüldüğü gibi yerle temas ettikleri noktadan deforme olurlar. Araç hareket etmek istediğinde tekerlek yüzeyindeki bozulan bu alan yer değiştirmek istemeyecek, böylece araç hareketine ters yönde olan ve yuvarlanma direnci olarak adlandırılan kuvvet ortaya çıkacaktır. Yuvarlanma direnci tekerleğin üzerine binen dik kuvvet (F_z) ve bu kuvvetin uygulama noktasının tekerleğin düşey eksenindeki merkezine olan uzaklığı (e) ile bağlantılıdır.

$$\left. \begin{aligned} F_{y1} &= \frac{e_1}{r_1} F_{z1} = f_1 F_{z1} \\ F_{y2} &= \frac{e_2}{r_2} F_{z2} = f_2 F_{z2} \\ F_{y3} &= \frac{e_3}{r_3} F_{z3} = f_3 F_{z3} \\ F_{y4} &= \frac{e_4}{r_4} F_{z4} = f_4 F_{z4} \end{aligned} \right\} \Rightarrow F_y = \frac{e}{r} F_z = f F_z \quad (4.21)$$

Kuvvet uygulama noktasının merkezden uzaklığının (e) yarıçapa (r) oranı yuvarlanma sabiti (f) olarak da adlandırılır.

$$f = 0.005 + \frac{1}{p} \left(0.01 + 0.0095 \left(\frac{V}{100} \right)^2 \right) \quad (4.22)$$

p : Lastik basıncı

F_z tekerleğin üzerine binen dik kuvvettir ve her teker için bu kuvvetin büyüklüğü ağırlık merkezine uzaklıkla bağıntılıdır.

Şekil 3.15’de de görüldüğü gibi ön tekerlerin ağırlık merkezine olan uzaklıklarının birbirine eşit ($l_{1R} = l_{1L}$) olduğu kabul edilebilir. Aynı varsayım arka tekerlekler için de geçerlidir. ($l_{2R} = l_{2L}$)

$$F_{z1} = F_{z2} = \frac{l_2}{l_1 + l_2} \cdot \frac{mg}{2} \quad (4.23)$$

$$F_{z3} = F_{z4} = \frac{l_1}{l_1 + l_2} \cdot \frac{mg}{2} \quad (4.24)$$

Ön ve arka tekerleklerden gelen kuvvetlerin dışında araca etki eden başka kuvvetler de vardır. Bunlar rüzgâr direnci (aerodinamik direnç), eğim direnci (yokuş direnci) ve ivmelenme direnci olarak sıralanabilir. Araca çarpan hava kütesinin etki ettiği kuvvete rüzgâr direnci (F_r) denir. Bu kuvvet yüksek hızlara çıkıldıkça eksponansiyel olarak artar.

$$F_r = 0.5c_w \rho A V^2 \quad (4.25)$$

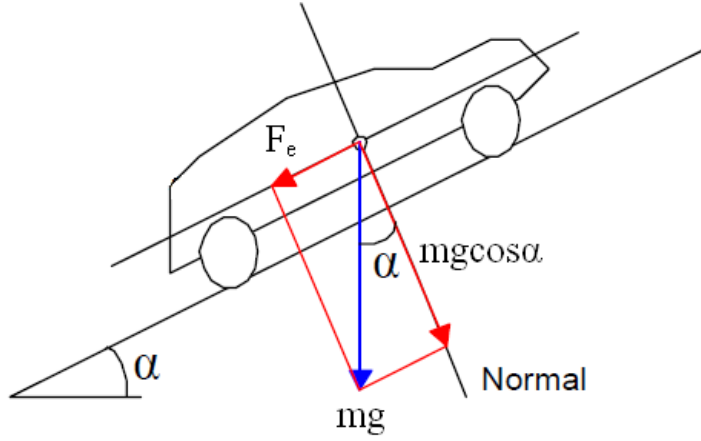
c_w : Direnç katsayısı

ρ : Havanın yoğunluğu

A : Aracın dik kesit alanı

İnişli bir yolda aracı hızlandırmak veya çıkışlı bir yolda aracı yavaşlatmak için yer çekiminin uyguladığı kuvvete eğim direnci kuvveti (F_e) denir. Yol eğimiyle ve araç ağırlığıyla ilişkilidir.

$$F_e = mg \sin \alpha \quad (4.26)$$



Şekil 4.5 : Yokuş direnci [34].

Otomobil pozitif veya negatif yönde ivmelendirilmek istendiğinde araçta bulunan güç aktarma elemanlarının dönen parçalarının ataletleri ve aracın kütlesi ivmelenmeye karşı bir kuvvet oluşturur. Bu kuvvete ivmelenme direnci denir. İvmelenmeye harcanan kuvveti hesaplayabilmek için ilk olarak sistemin toplam kinetik enerjisini hesaplamak gerekir. Sistemdeki farklı hızla dönen parçaların kinetik enerjilerini toplayacak olursak,

$$E_{KT} = E_{\omega_r} + E_{\omega_v} + E_{\omega_d} \quad (4.27)$$

Farklı hızlarda dönen kütlelerin kinetik enerjileri hesaplanacak olursa,

$$E_{\omega_r} = \frac{1}{2}(J_M + J_{m1})\omega_{tr}^2 = \frac{1}{2}(J_M + J_{m1})(k_v k_d \omega_d)^2 \quad (4.28)$$

$$E_{\omega_v} = \frac{1}{2}(J_v + J_{m2})\omega_v^2 = \frac{1}{2}(J_v + J_{m2})(k_d\omega_d)^2 \quad (4.29)$$

$$E_{\omega_d} = \frac{1}{2}(4J_t + J_{m3} + J_{m4})\omega_d^2 \quad (4.30)$$

$$E_{KT} = \frac{1}{2}(J_M + J_{m1})(k_vk_d\omega_d)^2 + \frac{1}{2}(J_v + J_{m2})(k_d\omega_d)^2 + \frac{1}{2}(4J_t + J_{m3} + J_{m4})\omega_d^2 \quad (4.31)$$

$J_{m1,m2,m3,m4}$: Aktarma organlarındaki ara millerin ataleti

J_t : Tekerlek ataleti

Dönen tüm parçaların toplam ataleti tek bir değişken olarak yazılacak olursa,

$$E_{KT} = \frac{1}{2}[(J_M + J_{m1})(k_vk_d)^2 + (J_v + J_{m2})(k_d)^2 + (4J_t + J_{m3} + J_{m4})]\omega_d^2 = \frac{1}{2}J_{E\mathcal{S}}\omega_d^2 \quad (4.32)$$

$$J_{E\mathcal{S}} = (J_M + J_{m1})(k_vk_d)^2 + (J_v + J_{m2})(k_d)^2 + (4J_t + J_{m3} + J_{m4}) \quad (4.33)$$

olarak bulunur. Araç ivmesiyle tekerleğe ait açısal ivme arasındaki bağıntı denklem 4.34'deki gibi hesaplanabilir. Bu durumda sistemin eşdeğer ataletinin ivmelenme direncine olan katkısı denklem 4.35'de verilmiştir.

$$\ddot{x} = (1-s)\ddot{\theta}_d R \quad (4.34)$$

$$F_{d1} = \frac{J_{E\mathcal{S}}\dot{\omega}_d}{r} = \frac{J_{E\mathcal{S}}\ddot{\theta}_d}{r} = \frac{J_{E\mathcal{S}}\ddot{x}}{rR(1-s)} \quad (4.35)$$

Aracın kendi kütlelerinin ivmelenme direncine olan etkisi de denklem 4.36'de hesaplanmıştır. Toplam ivmelenme direnci (F_D) ise denklem 4.37'deki gibi bulunur.

$$F_{d2} = m\ddot{x} \quad (4.36)$$

$$F_D = F_{d1} + F_{d2} \quad (4.37)$$

$J_{E\dot{S}}$: Eşdeğer atalet

Araca etki eden tüm negatif kuvvetler eşitliğin bir tarafına, tüm pozitif kuvvetler de eşitliğin diğer tarafına yazılacak olursa,

$$F_{\ddot{O}N} = F_r + F_{ARKA} + F_D + F_e \quad (4.38)$$

kuvvet dengesi sağlanmış olur.

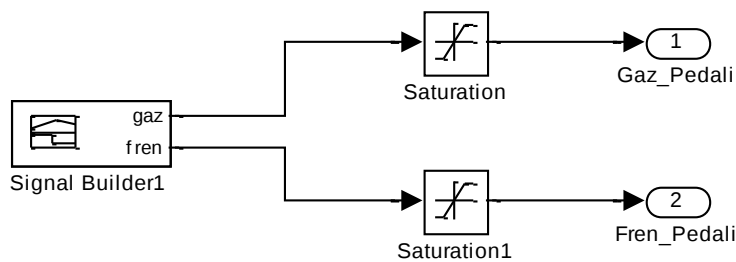
4.2 Sistemin Matlab Simülasyonu Modeli

Oluşturulan araç modeline ait Matlab simülasyonunun genel blok diyagramı Şekil 4.7’de gösterilmiştir. Model sürücü, dizel motor, dişli sistemi, tekerlekler, fren sistemi, araç, yol eğimi ve rüzgâr alt sistemlerinden oluşmaktadır.

4.2.1 Sürücü

Sürücünün uygulaması muhtemel gaz ve fren sinyallerini üretmek için “signal builder” yardımıyla bir alt sistem tasarlanmış ve farklı senaryolar üretilerek aracın hızı, ivmelenmesi, motor torku gibi parametrelerin değişimleri gözlemlenmiştir.

Burada gaz ve fren sinyallerine ait değerler 0–1 arasında değişmektedir. Bu değerler pedallara ne kadar basıldığını ifade etmektedir. Kullanılan limitörler ise sistemin devamına 0’dan küçük ya da 1’den büyük olası bir değer gönderilmesini engellemek amacıyla koyulmuştur.



Şekil 4.6 : Sürücü sinyalleri.

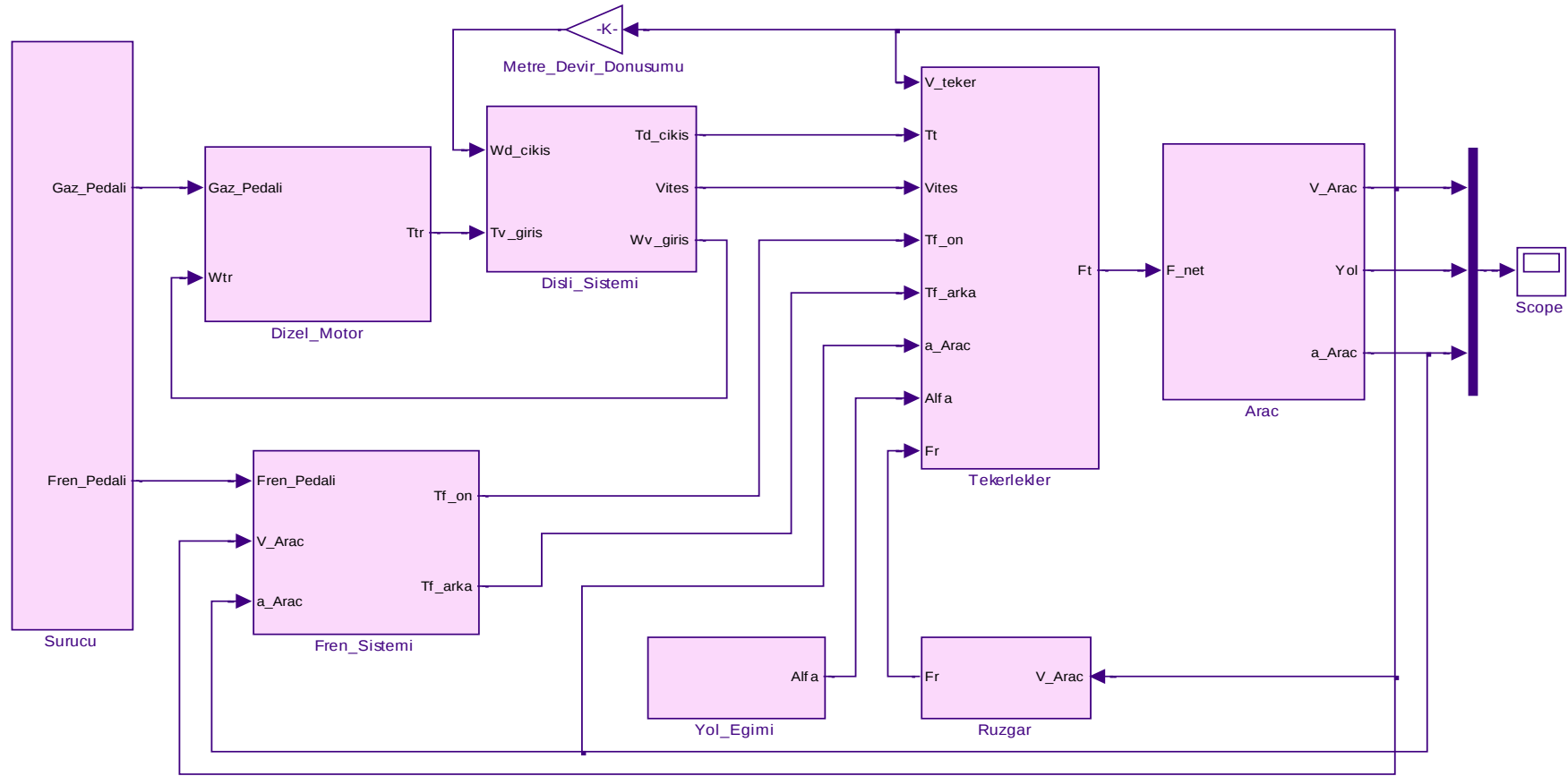
4.2.2 Dizel motor

Araç üzerinde bulunan dizel motor, gaz pedali basış oranı ve motor devrine göre çıkışta bir tork üretir. Bu üç değer arasındaki ilişkiyi ifade eden motor haritası araç üzerinde farklı hız ve devirlerde yapılan deneyler sonucunda tahmini olarak

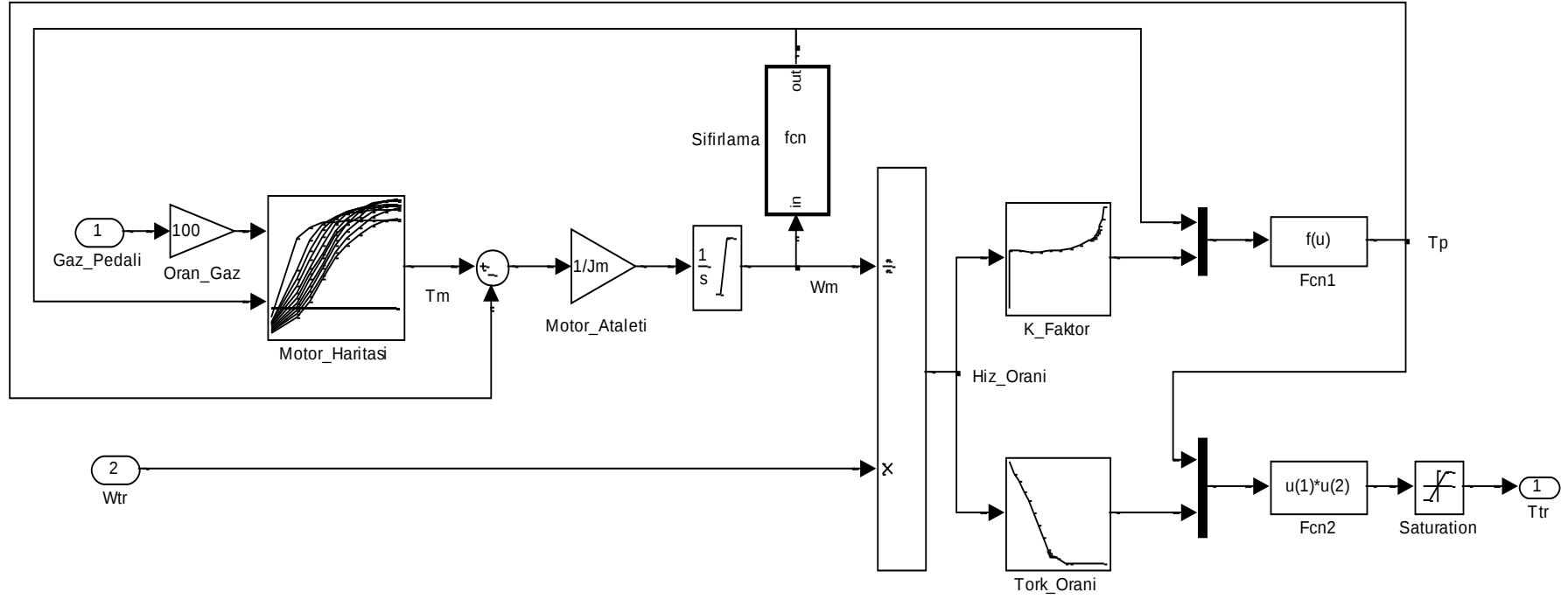
oluşturulmuş ve Şekil 4.8’de de görüldüğü gibi look-up table olarak simülasyona yerleştirilmiştir. Motor modelinin yanı sıra tork dönüştürücünün modeli de bu alt sisteme dâhil edilmiştir. Hem motora ait hem de motorla aynı hızda dönen tork dönüştürücüye ait hareketli parçaların ataleti J_M olarak ifade edilmiş ve motor devrini hesaplamada kullanılmıştır.

Motor üzerine binen yük tork dönüştürücünün pervane kanadı üzerindeki yüke (T_p) eşittir. Pervanenin tork gücünü hesaplayabilmek için motor dönüş hızının (ω_m) ve türbin hızının (ω_{tr}) yanı sıra tork dönüştürücünün yapısal özelliğini taşıyan K katsayıları da kullanılmıştır. Bu katsayılar tahmini olarak belirlenmiş ve look-up table biçiminde modele eklenmiştir.

Pervanedeki gücün bir kısmı türbine geçer ve tork dönüştürücünün çıkışındaki gücü (T_{tr}) oluşturur. Modelde karşılıklı kanatlar arasındaki güç aktarımının hangi oranda olacağı tork oranı isimli look-up table’a eklenmiştir.



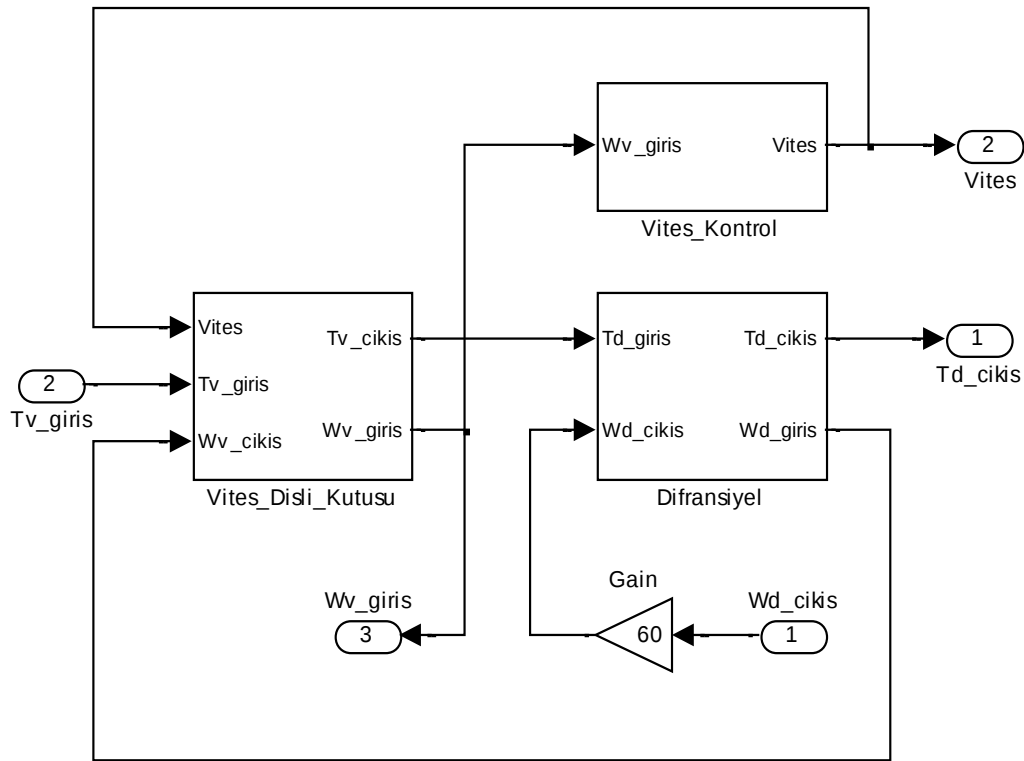
Şekil 4.7 : Sistemin ana blok diyagramı.



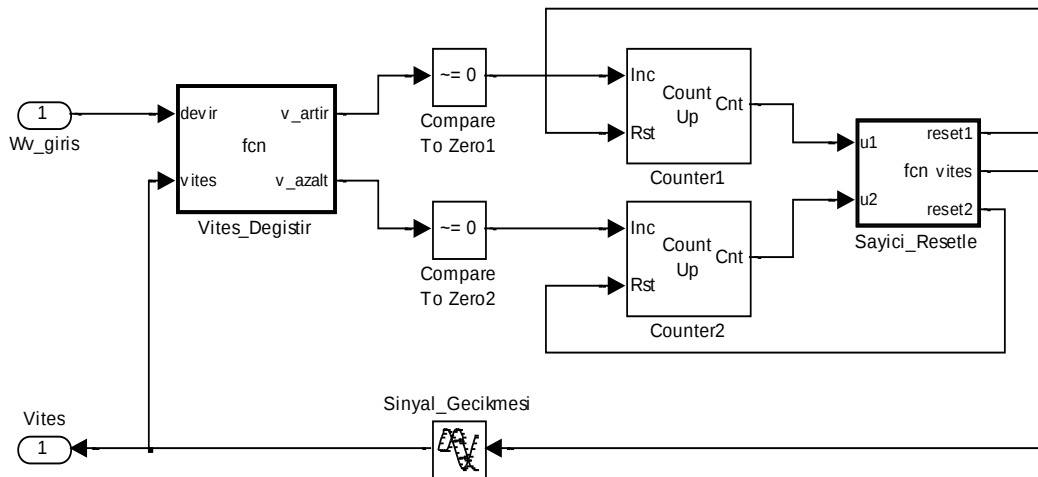
Şekil 4.8 : Dizel motor.

4.2.3 Dişli sistem

Dişli sistem vites dişli kutusunun modeli, geçişleri düzenleyen vites kontrol ünitesinin modeli ve diferansiyel kutusun modelinden oluşmaktadır. Bu alt sistemin öncelikli görevi motordan gelen torku o anki vites durumuna göre belli oranlarda değiştirmek ve çıkışa aktarmaktır. Diğer bir görevi de türbine ve tekerlere ait açısal dönüş hızları arasında bir bağlantı kurmaktır. Aynı zamanda motor dönüş hızına göre vites durumunun belirlenmesi de burada gerçekleşir.

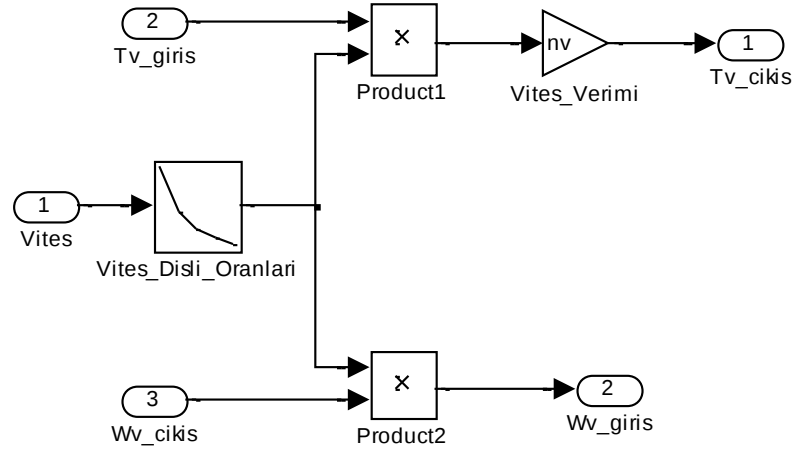


Şekil 4.9 : Dişli sistem.



Şekil 4.10 : Vites kontrol.

Vites kontrol ünitesi türbin devrine göre vites kademesini arttırmakta ya da azaltmaktadır. Araca uygun olması için üst limit 5. vites, alt limit 1. vites, üst vites geçiş 3600 rpm, alt vites geçiş 1000 rpm olarak belirlenmiştir. Bu sınırlandırmalar ve geçişler Şekil 4.10’da yer alan “Vites_Degistir” fonksiyonunun içine yazılımsal olarak gömülmüş ve kodu Ek-C ‘de verilmiştir. Vites değişim zamanı geldiğinde fonksiyonun çıkışında arttır ya da azalt sinyali üretilir. Sinyalleri sayan sayıcılar arasındaki fark o anki vites durumunu verir. “Sayici_Resetle” bloğu ise sayıcıların gerektiğinde resetlenmesi ve sayıcılar arasındaki farkın 1–5 arasında kalmasını sağlamaktadır. Vites geçişlerindeki gecikmeyi simüle etmek için sinyal gecikme elemanı konulmuş ve değeri tahmini olarak 0.1 saniye girilmiştir.



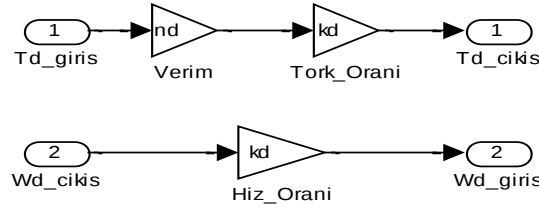
Şekil 4.11 : Vites dişli kutusu.

Şekil 4.11’de de görüldüğü gibi vites dişli kutusunun iki görevi vardır. Bunlardan biri vites değerine göre motordan gelen torku diferansiyele belirlenen oranda aktarmaktır. Gerçek sistemde sürtünmelerden dolayı torkta bazı kayıplar meydana gelir. Tork ile verim katsayısı(n_v) çarpılarak bu kayıpların etkisi modelde simüle edilmiştir.

Vites dişli kutusu modelinin diğer bir görevi ise türbin dönüş hızını bulabilmektir. Bunun için teker yönünden gelen hareket vites dişli oranında çarpılır. Vites dişli oranının bölen yerine yine çarpan olarak açısal hıza etki etmesinin sebebi açısal hıza ait giriş-çıkışların torka göre ters olmasıdır.

Diferansiyel dişlisi araç hangi viteste olursa olsun vites kutusunun aksine sabit bir oranda (k_d) torku ve açısal hızı değiştirir. Fakat vites dişli kutusunda olduğu gibi

tork bakımından diferansiyel dişlisinde de kayıplar mevcuttur. Bu kayıpların etkisini modele eklemek için verim katsayısı (n_d) kullanılmıştır.



Şekil 4.12 : Diferansiyel dişlisi.

4.2.4 Fren sistemi

Fren sistemine ait alt sistemin görevi fren pedalından gelen işaret ile doğru orantılı olarak negatif bir ivme meydana getirerek aracın yavaşlamasını veya durmasını sağlamaktır. Şekil 4.13’de de görüldüğü gibi anlık araç ivmesi, anlık araç hızı ve fren pedalının konumu bu modelin girişlerini, aracı durdurmak için ön ve arka tekerlekler için ayrı ayrı üretilen fren kuvvetleri ise çıkışlarını oluşturmaktadır.

Proje aracında da olduğu gibi ABS’li fren sistemleri frenleme esnasında oluşan tekerlek kaymasını teker hızını ve araç hızını kullanarak bulurlar. Tekerleklerle ait hızlar üzerlerinde bulunan sensörler yardımı ile ölçülür yani kayma direkt olarak ölçülemez.

Simülink model kurulurken tekerleklerin açısal hızlarını gerçek hayatta olduğu gibi bire bir ölçmek mümkün olmadığından modelde tekerlekler ile yer yüzeyi arasındaki kaymaları bulmak için farklı bir yol izlenilmiştir. Aerodinamik etkenlerin aracın dikey kuvvetlerine olan etkisi ihmal edildiğinde denklem 4.39’da olduğu gibi tüm tekerleklerle binen dikey kuvvetlerin toplamının (F_{zT}) aracın ağırlığına eşit olduğu görülmektedir. Denklem 3.1’den yararlanılarak toplam dikey kuvvetin, bir başka deyişle araç ağırlığının, dört tekere ait eşdeğer bir sürtünme katsayısı ile ($\mu_{(eş\ max)}$) çarpıldığında tüm tekerleklerle aracı durdurmak için verilebilecek yatay kuvvetin maksimum değeri ($F_{T\ max}$) bulunacaktır.

$$mg = F_{z1} + F_{z2} + F_{z3} + F_{z4} = F_{zT} \quad (4.39)$$

$$F_{T\ max} = \mu_{(max\ eş)} F_{zT} = \mu_{(eş\ max)} mg \quad (4.40)$$

Bu noktada frenlemenin kuru ve asfalt bir yolda yapıldığı var sayılırsa Şekil 3.16'nın da yardımıyla sürtünme katsayısının en fazla 1 değerine eriştiği görülür.

$$F_{T \max} = \mu_{(eş \max)} mg = ma \quad (4.41)$$

Sürtünme katsayısının en yüksek değeri yerine yazıldığında, aracı durdurmak için verilecek maksimum fren kuvvetinin yavaşlama ivmesinin yer çekimi ivmesine eşit olduğu an ortaya çıktığı denklem 4.42'de bulunmuştur.

$$g = a \quad (4.42)$$

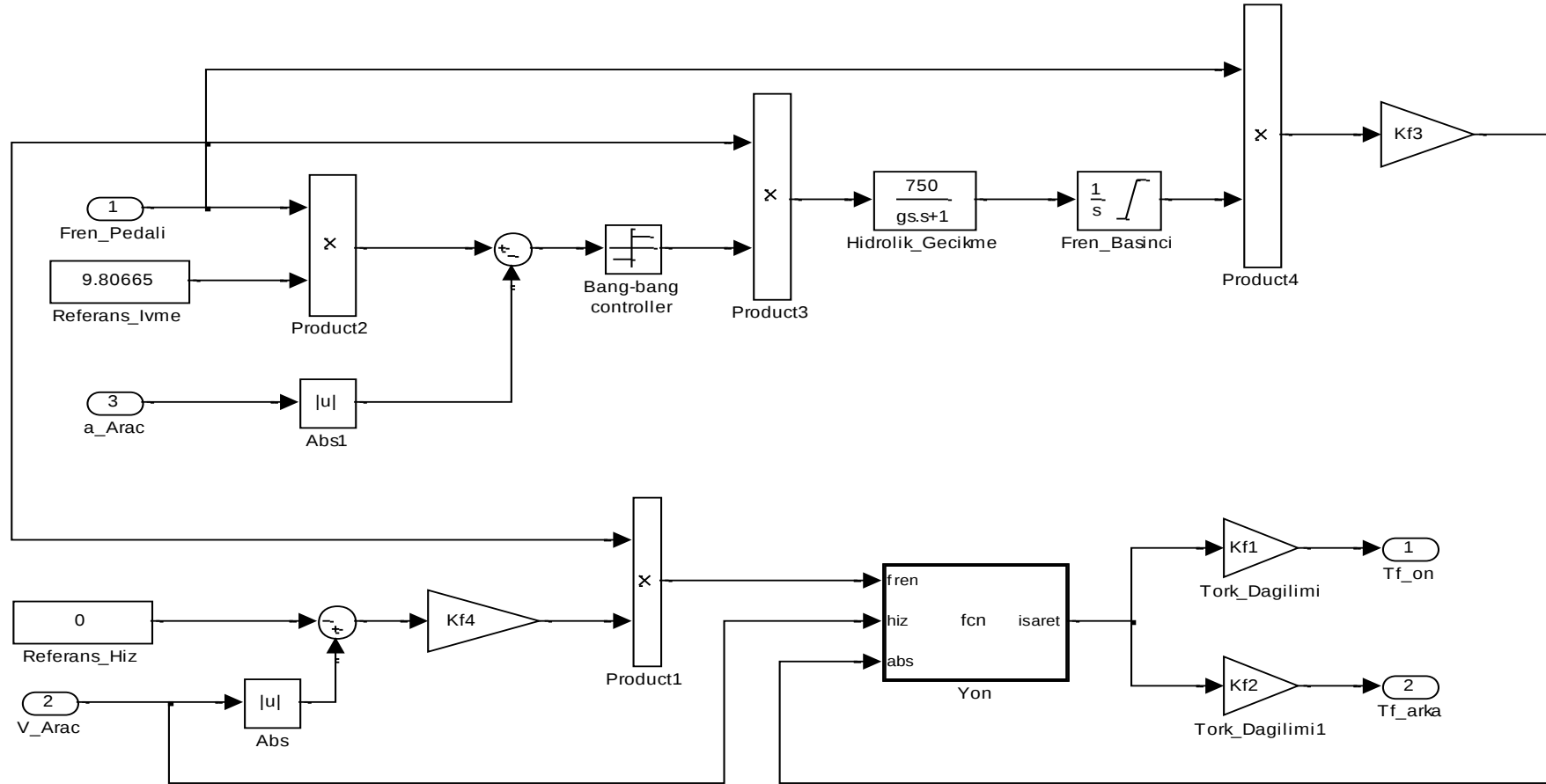
Simülasyonda bu noktada yapılması gereken iki temel işlem vardır. Bunlardan ilki fren pedalı oranına göre hidrolik fren basıncı meydana getirmek, diğeri ise ön ve arka tekerlekler dikey kuvvetler göz önünde bulundurularak oluşan fren torkunu dengeli bir biçimde dağıtmaktır.

Oluşturulan modelde fren pedalına basılmasıyla yavaşlama için referans bir ivme belirlenir. Üretilen referans değer aracın gerçek ivmesi ile karşılaştırıldıktan sonra çıkan hatanın işaretine ve büyüklüğüne göre hidrolik hat üzerinde basınç oluşturulur. “Hidrolik_Gecikme” isimli transfer fonksiyonu bu basıncı oluştururken hem hidrolik sistemin gecikmesini hem de basıncın artış hızı gibi karakteristik özellikleri de modele yansıtır. Tekerleklerin hareketi esnasında transfer fonksiyonunun çıkışındaki sınırlandırılmış basınç ile yer yüzeyinde oluşan yatay kuvvet arasında denklem 4.15'de olduğu gibi lineer bir ilişki olduğu için sonraki aşamada hidrolik basınç Kf3 sabiti ile torka dönüştürülür. “Yon” isimli fonksiyondan geçen fren gücü Kf1 ve Kf2 sabitleri çarpılarak ön ve arka tekerlere belli oranlarda gönderilir. Tekerlekler giden bu fren kuvvetleri tekerleklerin kaymasına sebep olursa ABS devreye girer ve Şekil 4.13'de görüldüğü gibi hidrolik basıncı aç-kapa yaparak tekerlerin yere tutunmasını sağlar. Modelde ABS, tekerleklerin kaydığını yavaşlama ivmesinin referans değerinin üstünde kalmasıyla anlar.

Çok küçük hızlarda veya araç dururken ABS hidrolik basıncı sürekli açıp kapamayacağı için yine aynı modelde ek bir frenleme yapısı mevcuttur. Sürücü fren bastığında referans hızı 0 km/h yapmak için bir tork üretir ve ABS sistemi gibi “Yon” isimli fonksiyona gönderir. Aynı şekilde bu fonksiyondan geçen tork ön ve arka tekerlere dağıtılır ve araç hızına negatif etki edecek şekilde işaretlendirir.

“Yon” fonksiyonun görevi aracı etkileyecek olan torku hangi girişten alacağını seçmektir. Bu kararı araç hızına bakarak belirler. Fonksiyonun bir diğer görevi ise fren kuvvetini her zaman araç hızına ters olacak şekilde işaretlendirmektir. Bu fonksiyonun kodu EK C’de mevcuttur.

Model fren pedalına basılmadığı anlarda ise fren pedal basış oranı çarpan olarak kullanıldığı için çıkışında ne ön tekerler için ne de arka tekerler için herhangi bir tork üretmez.



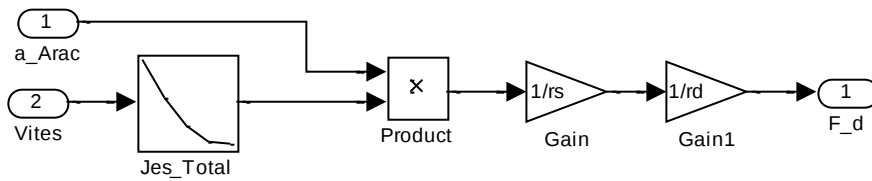
Şekil 4.13 : Fren sistemi.

4.2.5 Tekerlekler

Bu alt sistemde dizel motorun ürettiği ve dişlilerden geçerek gelen tork ile fren sisteminden gelen tork birleştirilir ve aracı hareketlendirecek net kuvvet elde edilir. Net kuvvet hesaplanırken rüzgâr, eğim gibi dış etkenlerden kaynaklı kuvvetler de modelde yer almıştır. Dış etkenlerin haricinde aracın kendi dinamiklerinden kaynaklı dirençler de bu noktada iki ayrı fonksiyon ve bir alt sistem olarak modele eklenmiştir.

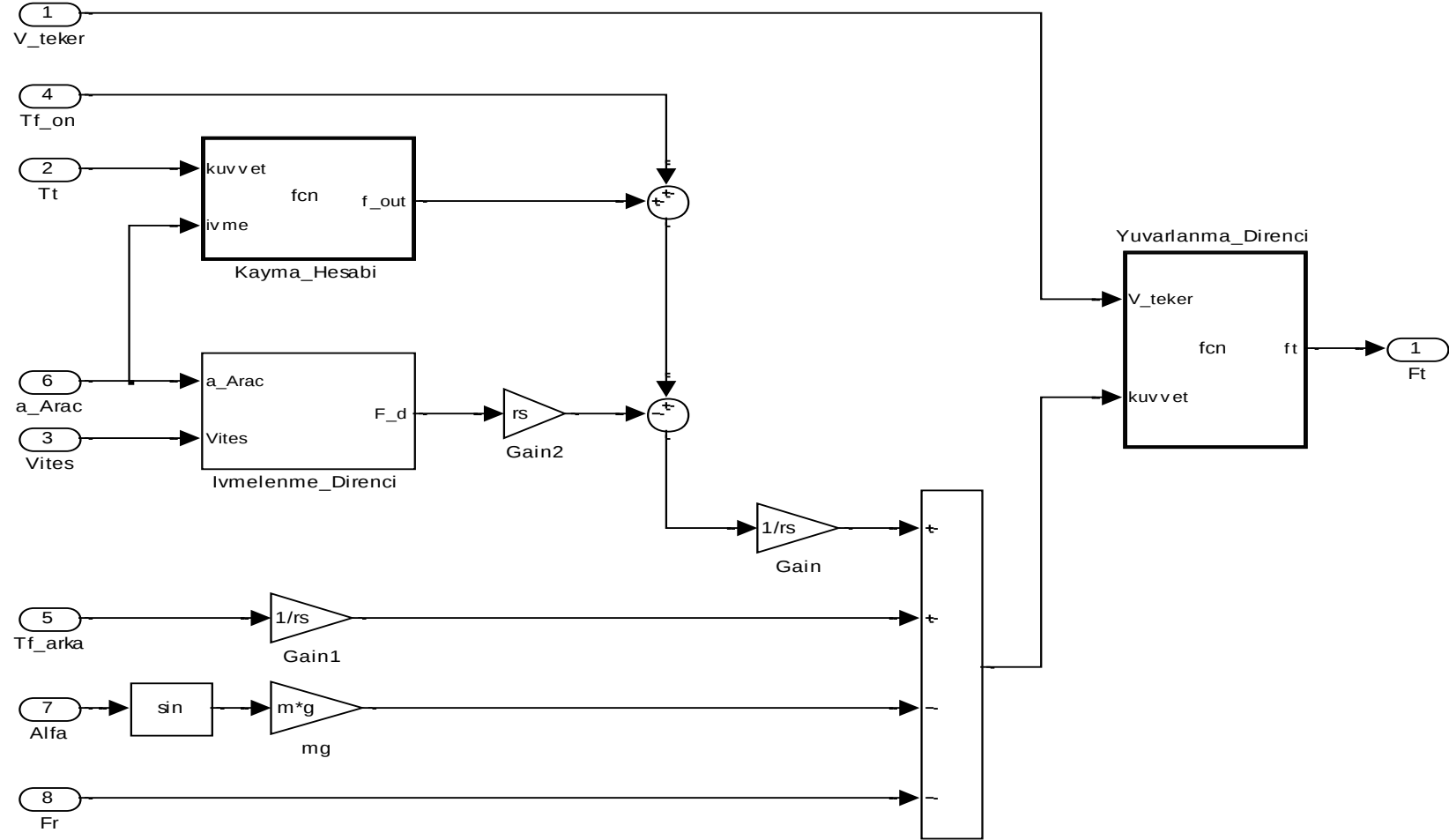
“Kayma_Hesabi” isimli fonksiyonun modeldeki görevi motordan gelen torku belli bir noktada limitleyerek ön tekerleklerin patinaj çekmesini engellemektir. Sınırlama ön tekerlekler ile yer ile arasındaki sürtünme kuvvetinin en yüksek olduğu anda yapılır. Böylece tekerleklerin yere iyi bir şekilde tutunması sağlanarak patinajın önüne geçilir. İvmelenme esnasında çekiş tekerlerinde çok küçük de olsa sürekli var olan kayma bu fonksiyonda hesaba katılmıştır.

Modelde fren sisteminin ön tekerleklerde oluşturduğu tork motordan gelen torktan çıkartıldıktan sonra elde edilen sonuca araçtaki güç aktarım organlarının ataletlerinin etkisi denklem 4.35’de olduğu gibi dâhil edilmiştir. Denklemde yer alan eşdeğer atalet ($J_{EŞ}$) her vites için ayrı ayrı hesaplanmış ve girişi vites konumu olan ”Jes_Total” isimli look-up table oluşturulmuştur. Ataletlerin modele dâhil edildiği bu işlemlerin tümü Şekil 4.14’de içyapısı verilen “İvmelenme_Direnci” isimli alt sistemde gerçekleştirilmiştir.



Şekil 4.14 : İvmelenme direnci.

Şekil 4.15’e bakıldığında ivmelenme direnci isimli alt sistemden hemen sonra rüzgâr direncinin(F_r), yokuş direncinin ve arka tekerleklerden gelebilecek frenleme kuvvetinin(F_{ARKA}) tam bu noktada var olan pozitif veya negatif kuvvetlere eklendiği görülmektedir. Tüm bu kuvvetlerin bileşkesinin aracı hareket ettirebilmesi için yuvarlanma direncini yenebilmeleri gerekmektedir.



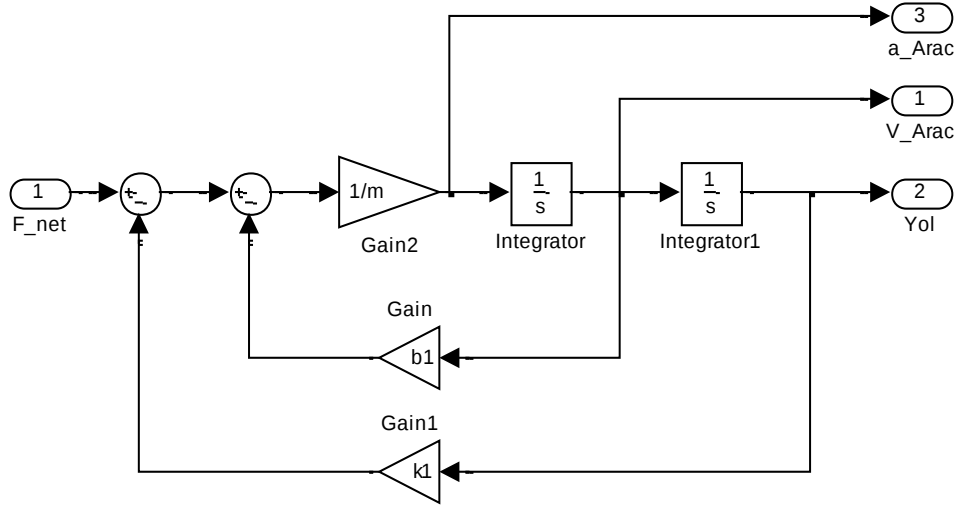
Şekil 4.15 : Tekerlekler.

“Yuvarlanma_Direnci” isimli fonksiyon tekerleklerin deformasyonundan kaynaklanan harekete ters yöndeki kuvveti hesaplamaktır. Bu kuvvet tekerleğin açısal hızına bağlı olarak değiştiğinden, araç hızı bu fonksiyona girilmiş, içerde tekerleğin açısal hızına dönüştürülerek hesaba katılmıştır. Bu fonksiyona ait kod EK C’de bulunabilir.

Son olarak aracı hızlandırmak veya yavaşlatmak isteyen tüm bu kuvvetlerin toplamı net kuvvet olarak hesaplanmış ve bir sonraki sistemin kullanması için çıkışa bağlanmıştır.

4.2.6 Araç

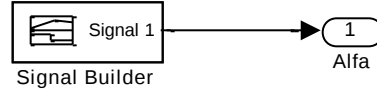
“Araç” alt istemine giriş olarak sadece bir önceki blokta hesaplanan net kuvvet gelmektedir. Bu net kuvvet araç kütlesini etkilemekte, böylece ortaya ivme, hız ve mesafe bilgileri ortaya çıkmaktadır. Araç üzerinde bulunan güç aktarma organlarına ait dönen parçaların ataletlerinin etkisi “İvmelenme_Direnci” isimli fonksiyonda hesaba katılsa da, ivmelenme direncinin bir başka etkeni olan araç kütlesi de bu alt sistemde modele eklenmiştir.



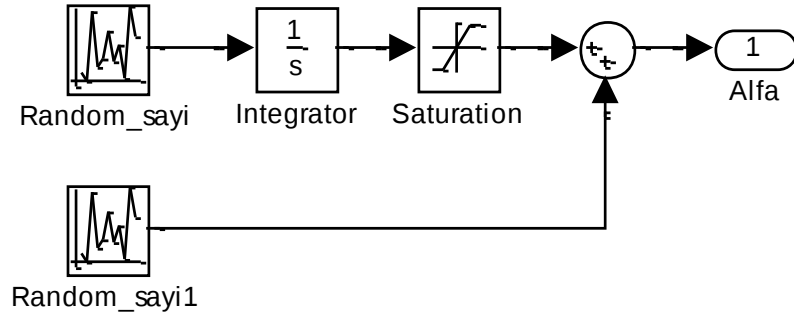
Şekil 4.16 : Araç.

4.2.7 Yol Eğimi

Ana blok diyagramda yer alan “Yol_Eğimi” isimli alt sistemde Şekil 4.17’de ve Şekil 4.18’de olduğu gibi iki farklı eğim modelde simüle edilmiştir.. Bunlardan ilkinde belli bir senaryo izlenerek yokuş miktarı belirlenmiş, diğerinde ise sınırlandırılmış ama rastgele bir durum oluşturulmuştur.



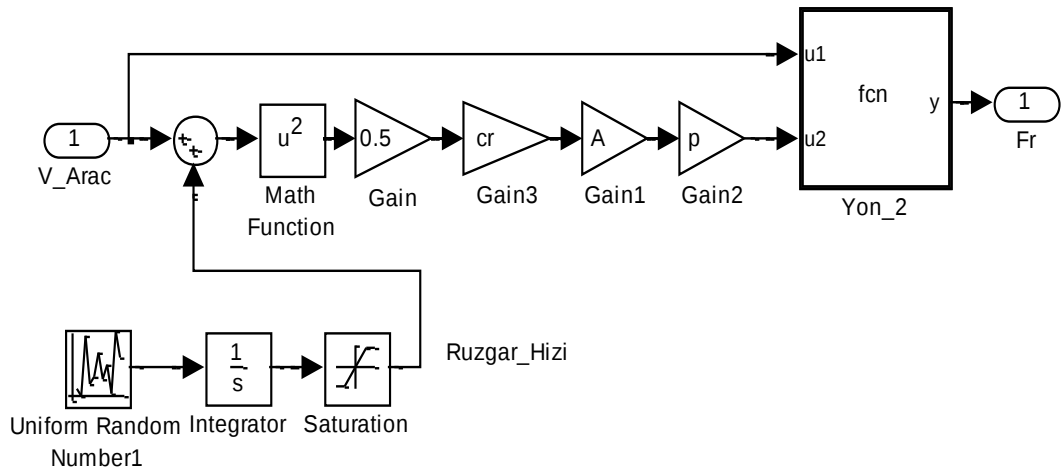
Şekil 4.17 : Yol eğimi(a).



Şekil 4.18 : Yol eğimi(b).

4.2.8 Rüzgâr direnci

Bu alt sistemde aracın hareketlenmesiyle birlikte dik yüzeye çarpan havanın oluşturduğu kuvvet, aracın parametreleri kullanılarak hesaplanmış ve “YON_2” fonksiyonu sayesinde araç hızına ters yönde etki edecek biçimde sisteme dâhil edilmiştir. Şekil 4.19 incelendiğinde aracın hareketinden kaynaklanan rüzgâr kuvvetinin yanı sıra çevre şartlarından dolayı var olan, yönü ve şiddeti bilinmeyen bir hava akımının etkisinin modellendiği görülebilir.



Şekil 4.19 : Rüzgâr direnci.

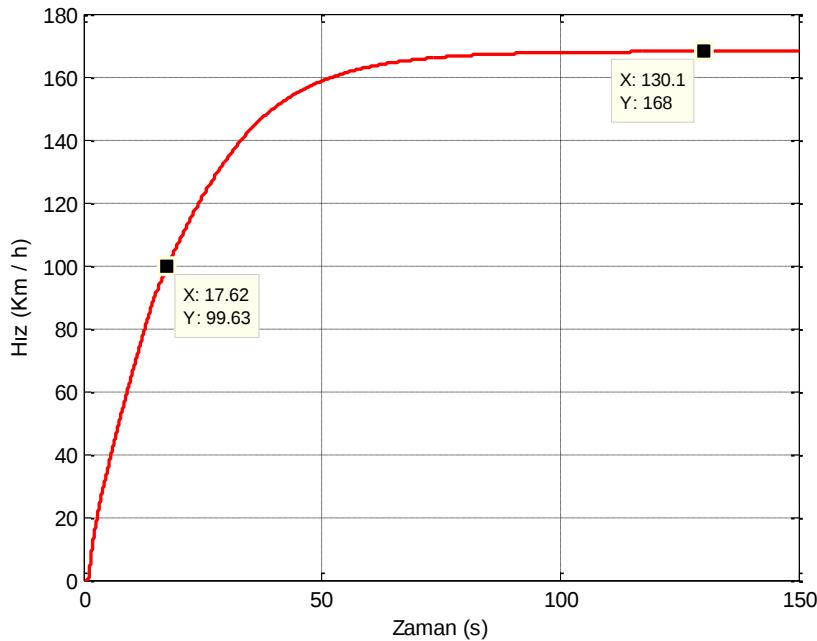
4.3 Simülasyon Sonuçları

Matlab/Simulink'te kurulan modelde sürücünün çeşitli senaryolarla gaza ve frene bastığı düşünülerek sisteme farklı girişler uygulanmış, bu girişlere karşılık gelen araç hızı, araç ivmesi, motor devri, vites değişimleri, açısal hız oranı, türbin devri, diferansiyel torku gibi çıkışlar gözlenmiştir.

4.3.1 Aracın tam gaz ile hızlanmasının simüle edilmesi

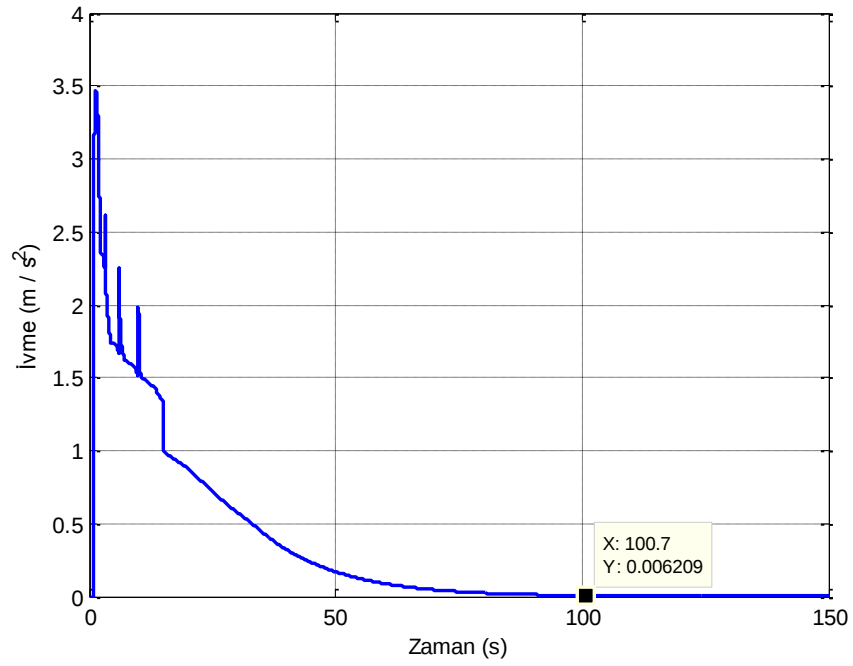
Simülasyon ortamındaki ilk deneyde fren pedalına herhangi bir müdahalede bulunulmazken, 1. saniyede gaz pedalına %100 oranında basılmış ve aracın minimum sürede maksimum hıza çıkması gözlemlenmiştir. Deneyde yolun eğimsiz olduğu ve rüzgâr hızının 0 m/s olduğu kabul edilmiştir.

Projede kullanılan aracın maksimum ulaşabileceği hız 168 km/h'dir. Ayrıca araç 100 km/h hıza yaklaşık 16 saniyede ulaşmaktadır. Üretici firmadan alınan bu veriler ışığında elde edilen simülasyon sonucu Şekil 4.20'deki gibidir. Grafik incelendiğinde aracın 100 km/h hıza 17,62 saniyede çıktığı ve maksimum 168 km/h hıza ulaştığı görülmektedir. Gaz pedalına 1 saniye gecikmeli basıldığı düşünülecek olursa bu değer gerçek değerlerle neredeyse birebir örtüşmektedir.



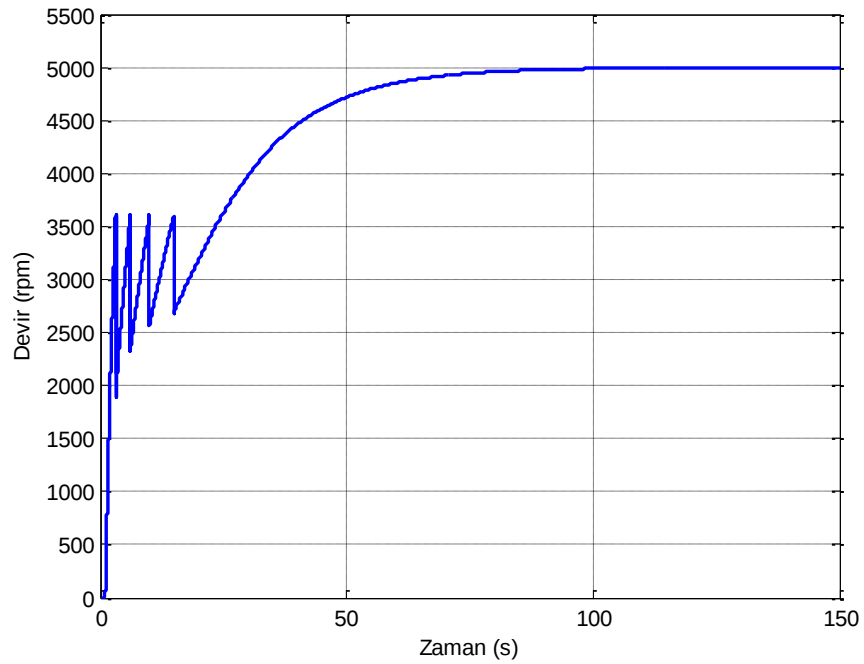
Şekil 4.20 : Araç hızı.

Şekil 4.21'de ise araç maksimum hıza eriştiğinde, diğer bir deyişle hızlanması bittiğinde ivmesinin 0'a yaklaştığı net bir şekilde görülmektedir.



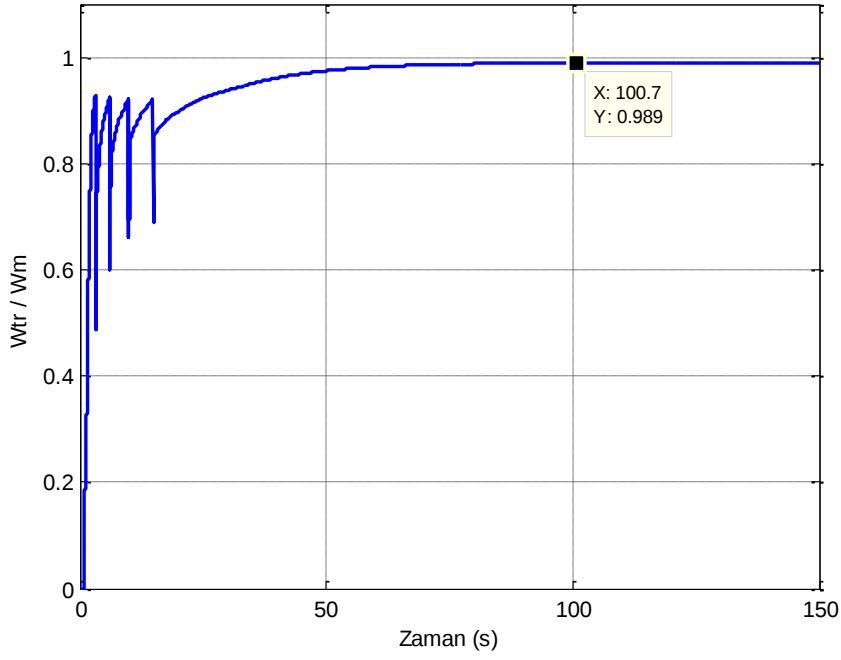
Şekil 4.21 : Araç ivmesi.

Araca ait 5. vitesteki maksimum türbin devrinin deneysel olarak elde edilememesi sebebiyle simülasyon oluşturulurken bu değer 5000 rpm olarak kabul edilmiştir. Şekil 4.22'de görüldüğü gibi araç maksimum hıza ulaştığında türbin devri yaklaşık 5000 rpm'e oturmaktadır. Ayrıca aynı grafikte vites artış anlarında türbin devrinin dişli oranlarıyla orantılı olarak düştüğü de görülmektedir.

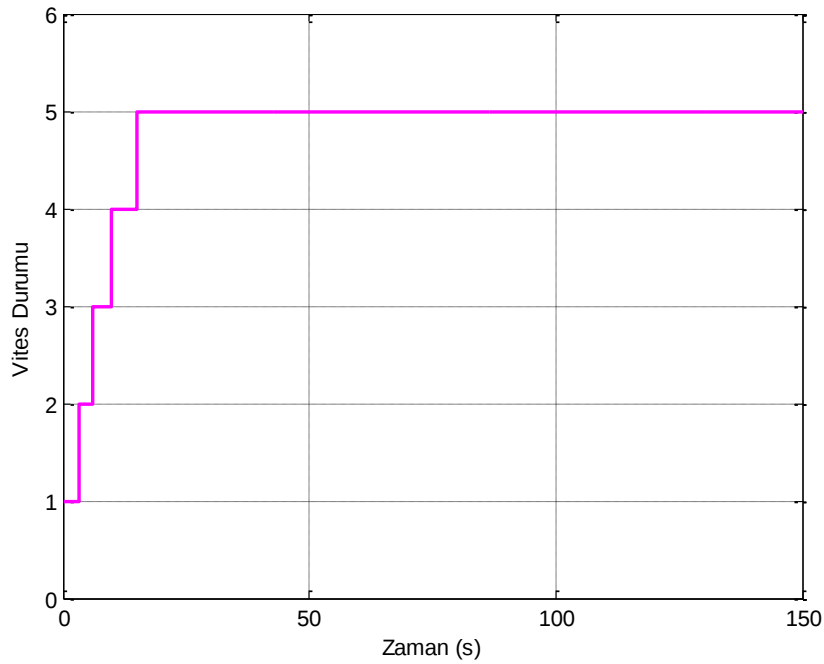


Şekil 4.22 : Türbin devri

Açısal hız oranı türbin devrinin motor devrine bölünmesiyle elde edilir. Araç dizel motorun ürettiği tork sayesinde ileri doğru ivmelenirken tork dönüştürücünün yapısından dolayı motor devri her zaman türbin devrinden daha büyük olacaktır. Bu yüzden açısal hız oranı yüksek devirlere çıkıldıkça 1'e yaklaşıp da, 1'in üzerine çıkamaz.

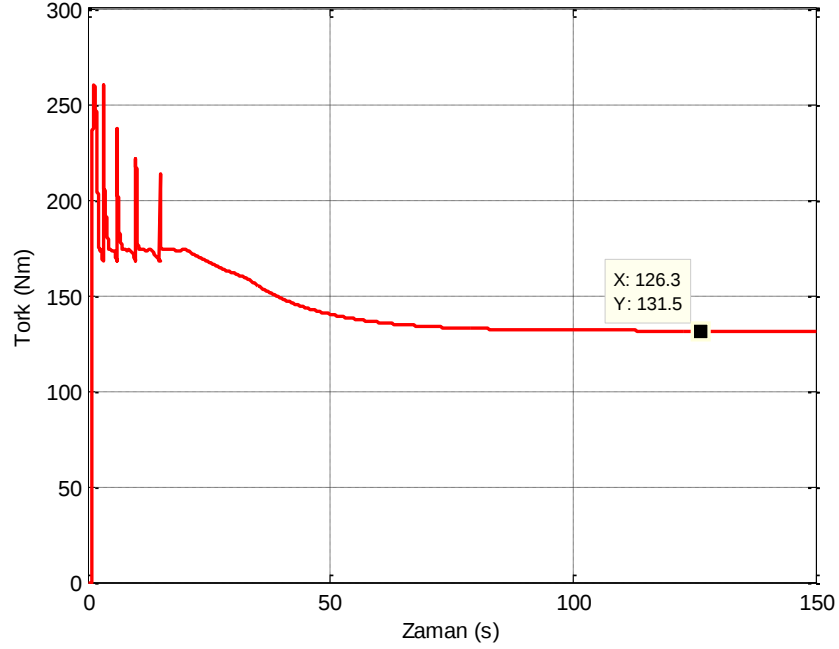


Şekil 4.23 : Açısal hız oranı değişimi.

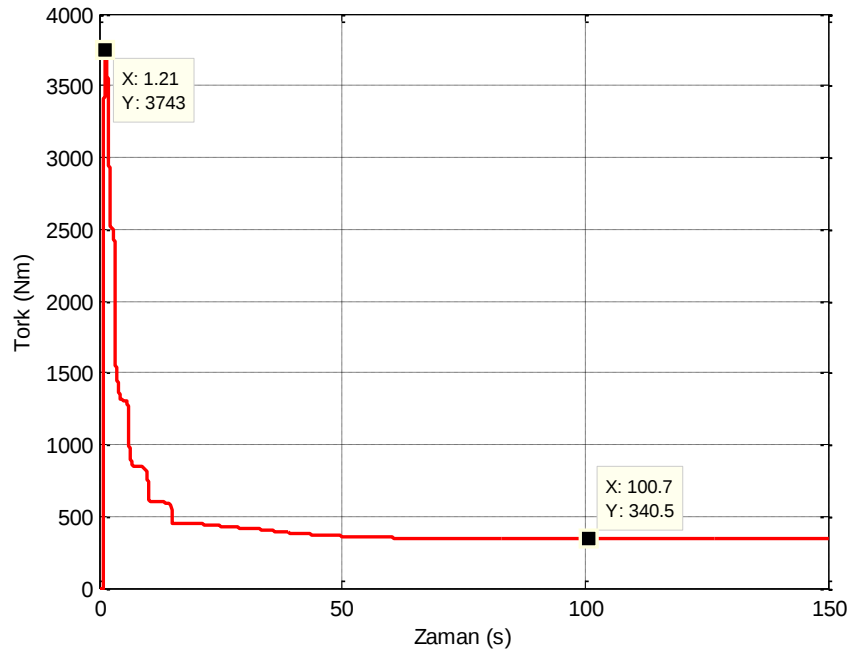


Şekil 4.24 : Vites değişimleri.

Vites kutusuna gelen tork, vites deęişim anlarında ani yükselmeler göstermektedir. Kısa süreli bu impulslerin sebebi vites kutusuna göre motor yönündeki elemanların ataletlerinden kaynaklanan enerji birikimidir.



Şekil 4.25 : Vites kutusuna gelen tork.



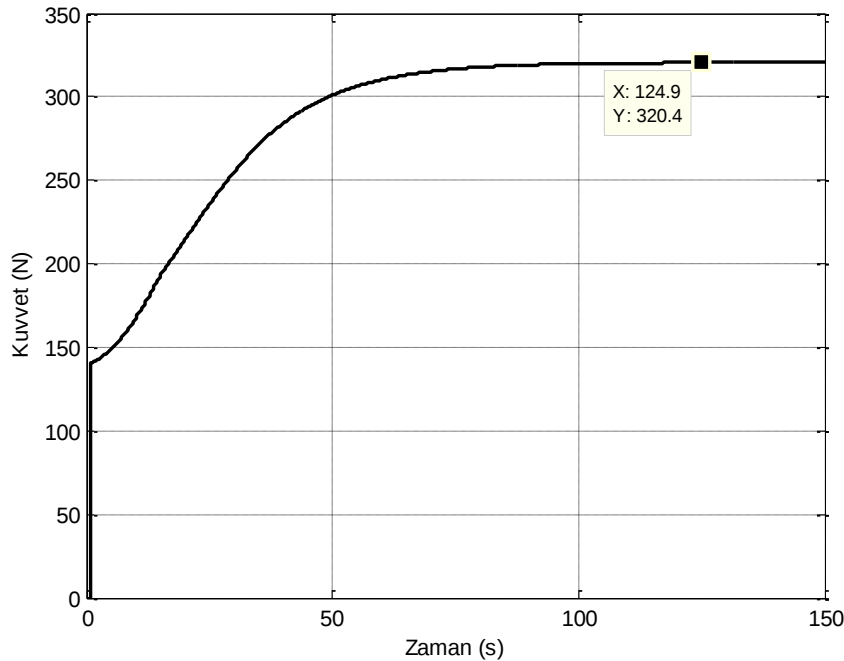
Şekil 4.26 : Diferansiyel çıkış torku.

Diferansiyelden tekerleklerle aktarılan tork, tork dönüştürücünden gelen deęerle aynı deęildir. Bunun sebebi aracın hızlandıkça vites arttırması ve buna baęlı olarak çıkışa

aktarılan tork oranının değişmesidir. Şekil 4.26’de vites arttıkça tork kademeli olarak düşmektedir. Fakat bu kademeli düşüşün dışında vites sabit kalsa bile torkta değişimler meydana gelebilir. Bunun sebebi dizel motorun dönüş hızının değişmesinden dolayı farklı miktarlarda tork çıkışı vermesidir.

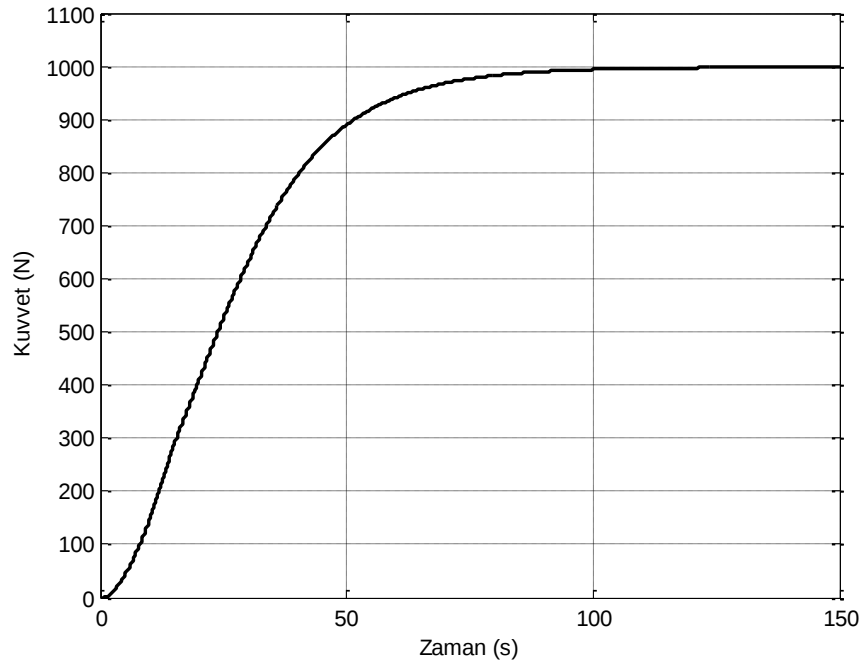
İvmelenirken aracı durdurmaya çalışan kuvvetlerden biri de yuvarlanma direncidir ve tekerleğin hareket edebilmesi için tekerleğe gelen kuvvetin başlangıç yuvarlanma direncini yenebilecek değerde olması gerekir. Araç sabit bir hıza ulaştığında ise yuvarlanma direnci de sabit kalacaktır.

Yapılan simülasyonda araca ait lastik basıncı, lastiğe uygulanan dikey kuvvet gibi parametrelerin gerçek değerleri girilmiş ve yuvarlanma direncinin değişimi Şekil 4.27’deki gibi elde edilmiştir. Grafik incelendiğinde araç hızlandıkça yuvarlanma direncine harcanan kuvvetin de arttığı görülmektedir. Ayrıca yuvarlanma direncinin bir başlangıç değeri olduğu ve bu değer dört teker için toplamda yaklaşık olarak 140 N’dur. Grafikten çıkarılabilecek bir başka sonuç ise araç hızı arttıkça yuvarlanma direncinin de arttığıdır.

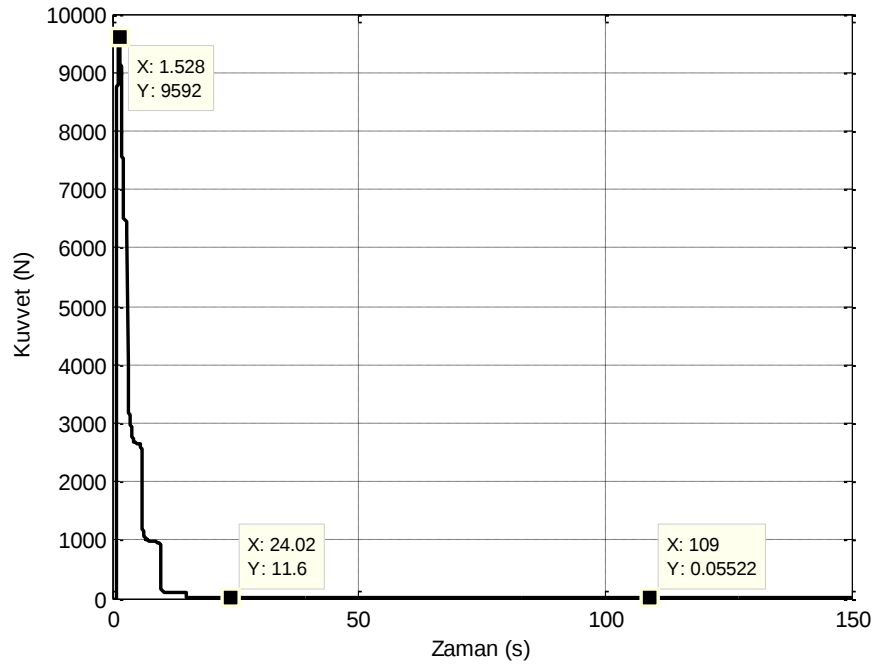


Şekil 4.27 : Araç hızına bağlı yuvarlanma direnci.

Simülasyonda mutlak rüzgar hızı 0 m/s olarak girilse de Şekil 4.28’de araç yüksek hızlara çıktıkça rüzgar direncine ciddi bir kuvvet harcandığı görülmektedir. Bunun sebebi aracın ileri hareketinin oluşturduğu ters yöndeki hava akımıdır.



Şekil 4.28 : Rüzgâr direnci.



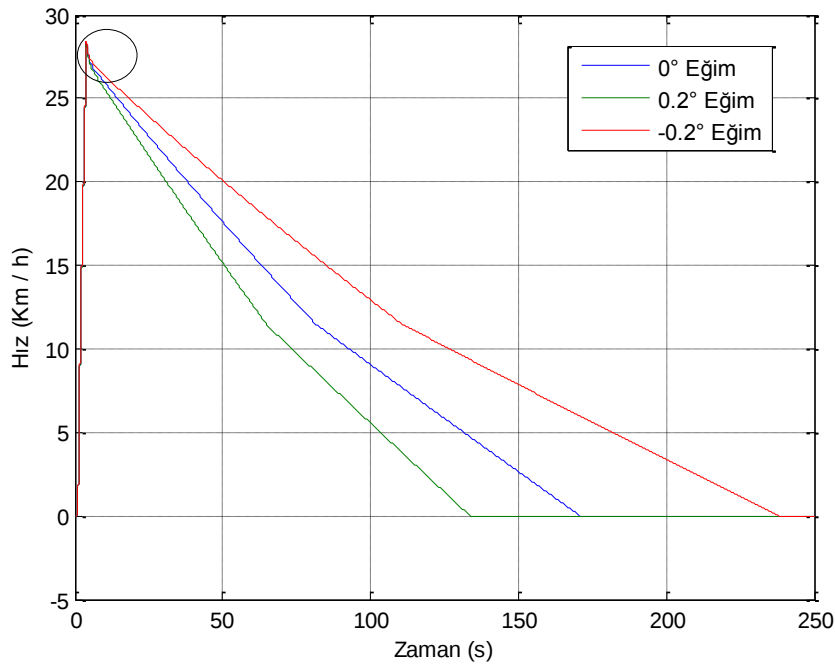
Şekil 4.29 : İvmelenme direnci.

İvmelenme direnci ise sadece araç ileri ya da geri hızlandığında ortaya çıkmaktadır. Şekil 4.29'da araç ileri yönde ivmelenirken ivmelenme direncine harcanan kuvvetin 9500 N seviyesine kadar çıktığı gözlemlenmiştir. Fakat araç hızı sabitlendikçe bu dirence harcanan kuvvetin de 0 N değerine yaklaştığı net olarak görülmektedir.

4.3.2 Eğiminin farklı olduğu yollarda aracın simüle edilmesi

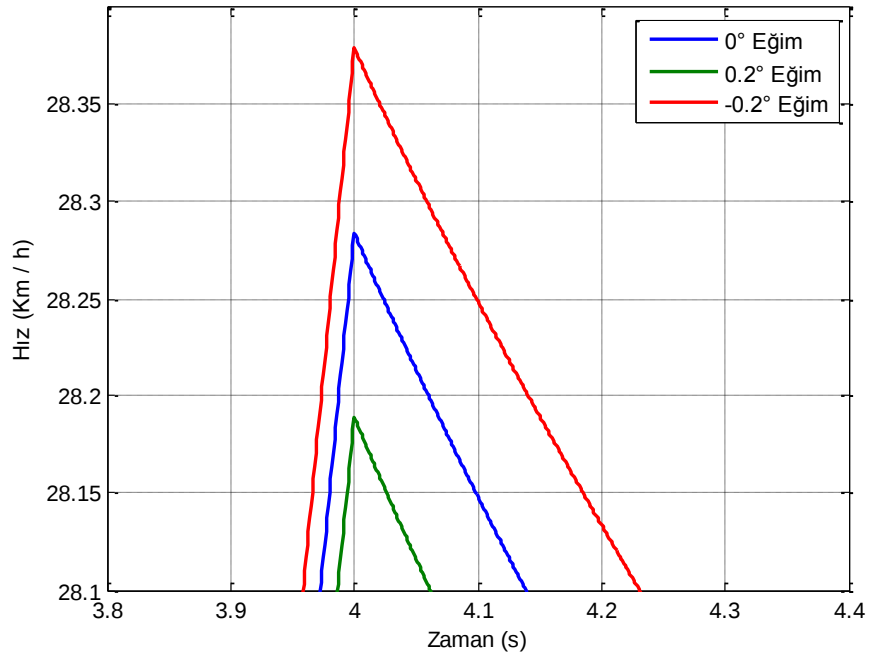
Bu deneyde $t=1$ s - $t=4$ s arasında gaz pedalına %100 basılmış ve aracın kendiliğinden durması beklenmiştir. Aynı deney 3 farklı yol eğiminde gerçekleştirilmiştir. Mutlak rüzgâr hızı 0 m/s olarak sisteme girilmiştir. Önceki deneyde de olduğu gibi aracın hareketinden kaynaklı bir hava yoğunluğu aracın dikey yüzeyine çarpmakta ve harekete ters bir kuvvet oluşturmaktadır

Simülasyonlarda pozitif eğim yokuş, negatif eğim ise iniş olarak kabul edilmiştir. Aracın kendiliğinden durabilmesi için eğimden gelecek kuvvetin, yuvarlanma veya motor freni gibi aracın hareket etmesini engelleyecek kuvvetlerin toplamından daha az olması zorunludur. Bu sebeple simülasyon için 0° 'lik eğimin yanı sıra, $(+)$ $0,2^\circ$ ve $(-)$ $0,2^\circ$ 'lik çok küçük eğimler de giriş olarak tercih edilmiş ve simülasyona girilmiştir. Böylece aracın düz, hafif inişli ve hafif çıkışlı yollardaki hareketleri gözlemlenmiştir.



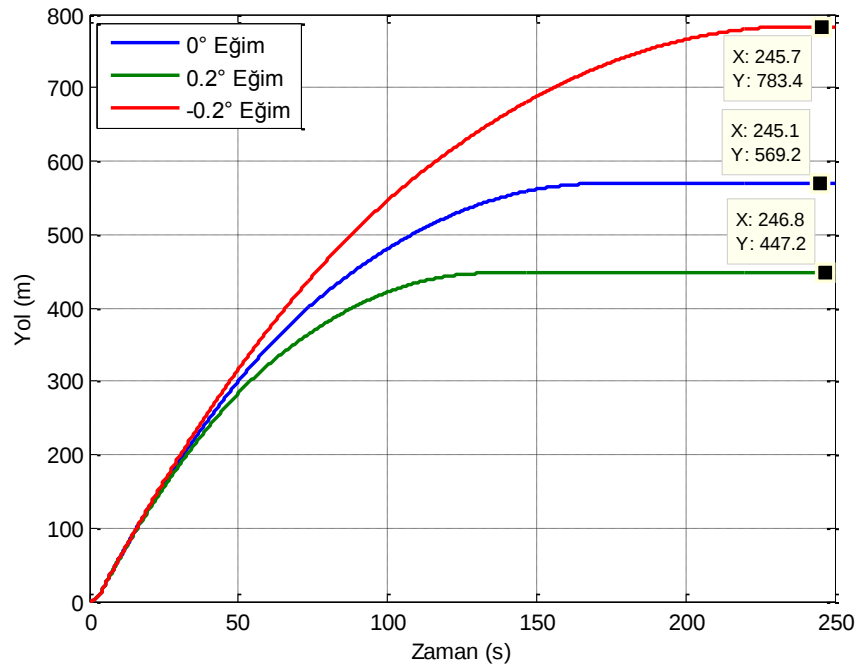
Şekil 4.30 : 3 farklı eğim için hız grafiği.

Yol eğiminin değişimine göre araç hızı da farklılık göstermektedir. Hafif yokuş (eğim= $0,2^\circ$) bir yolda aracın enerjini eğimsiz bir yola göre daha çabuk tükettiği Şekil 4.30'da gözlemlenmektedir. Hafif inişli($-0,2^\circ$ eğim) bir yolda ise tam tersi bir durum gerçekleşmekte ve araç daha geç durmaktadır.



Şekil 4.31 : Maksimum hız noktaları.

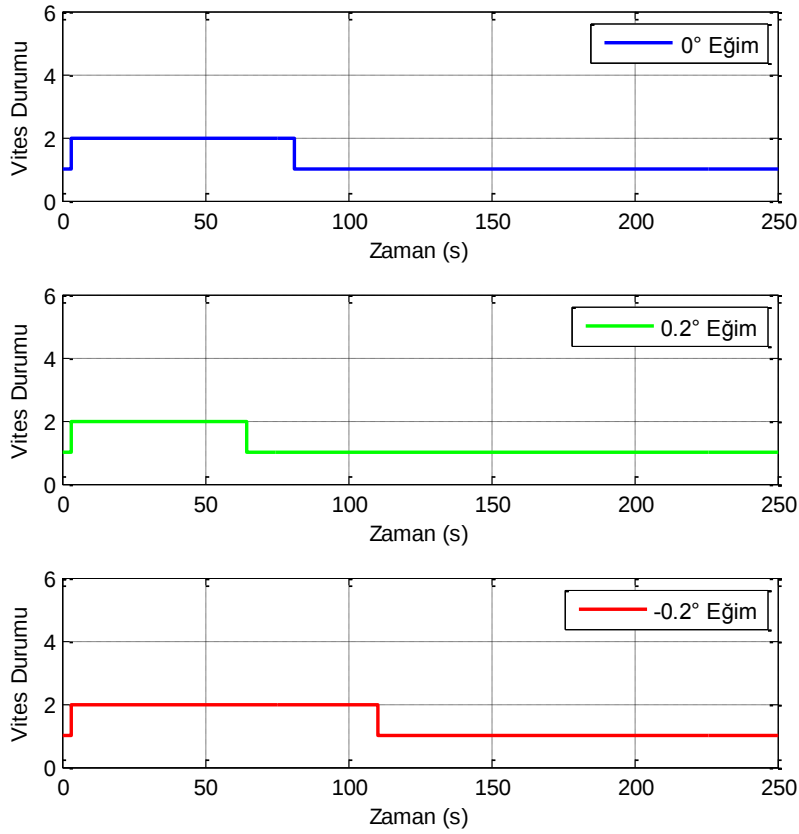
Her ne kadar maksimum hız aynı gibi görünse de Şekil 4.31’de daha yakından incelendiğinde (-) 0,2°’ lik eğime sahip yolda araç hızının maksimum değerinin daha yüksek seviyelere ulaştığı görülecektir. Aynı zamanda inişli bir yolda yükselmenin daha hızlı, yokuş yolda ise daha yavaş olduğu da grafikte gözlemlenmektedir.



Şekil 4.32 : 3 farklı eğim için aracın aldığı yol.

Şekil 4.33'deki grafik ise eğimin pozitif olduğu yolda aracın daha çabuk durduğunu ve daha az yol kat ettiğini, eğimin negatif olduğu yolda ise tam tersi bir durumun ortaya çıktığını göstermektedir.

Farklı eğimlerde aynı gaz pedalı girişine rağmen araca ait hız profillerinin farklılık göstermezsiden dolayı, motor devrinde de farklıklar olacaktır. Bunun sonucunda vitesin değişim anları da değişecektir.

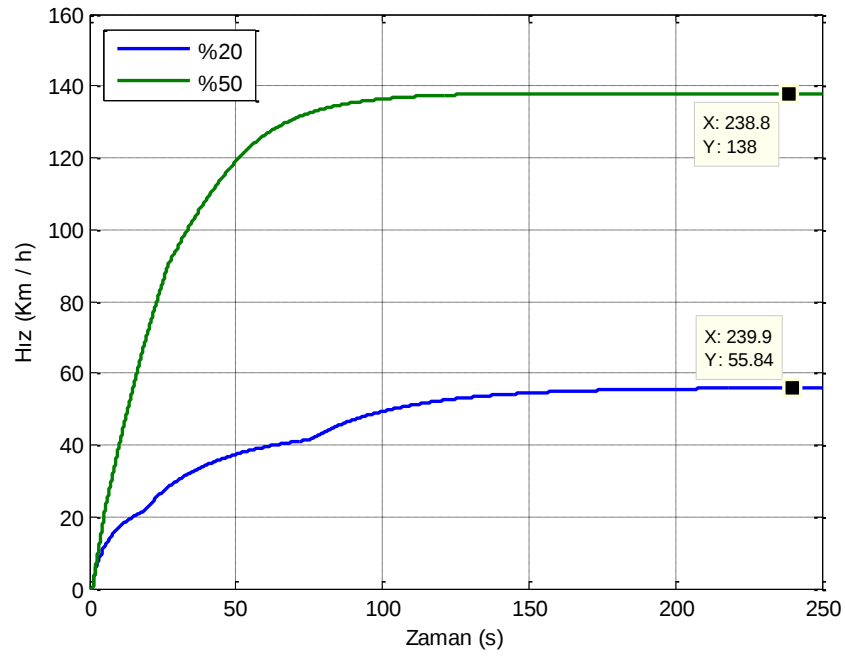


Şekil 4.33 : 3 farklı eğim için vites değişim zamanları.

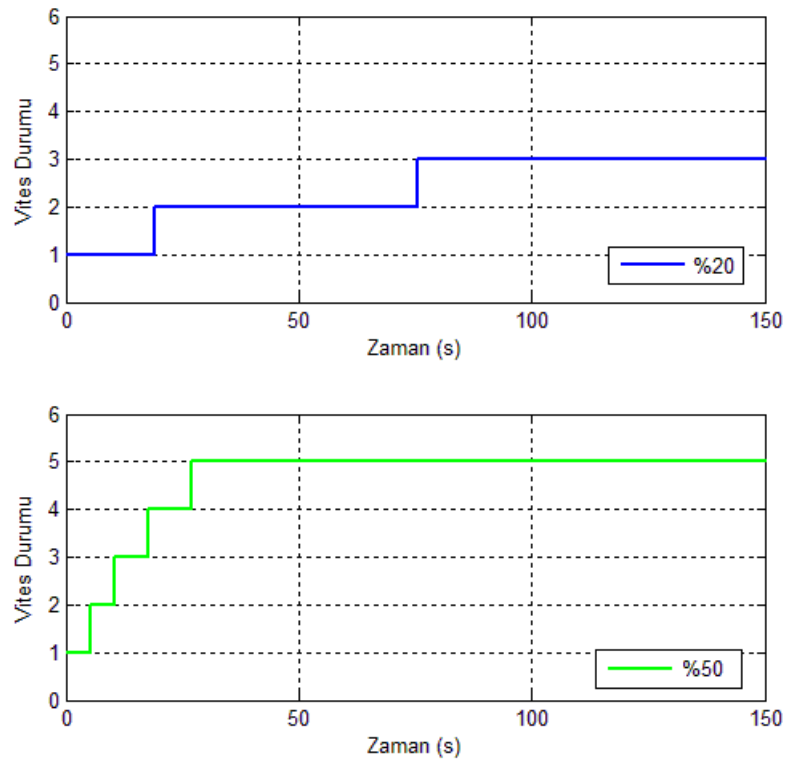
4.3.3 Aracın ayrı iki gaz oranı girişiyle hızlanması

Bu deneyde gaz pedalına $t=1$ s anında %20 ve %50 oranında basılmış ve bir karşılaştırma yapılmıştır. Frene bir müdahalede bulunulmamıştır. Araç eğimin 0° ve mutlak rüzgâr hızının 0 m/s olduğu kabul edilmiştir.

Gaz pedalına %20 oranında basıldığında araç hızı 55 km/h 'a, %50 basıldığında ise 138 km/h'a ulaşmıştır. Şekil 4.34'da gaz pedalına %20 basıldığında vites geçiş noktaları rahatlıkla görülmektedir. Grafikte giriş seviyesi yükseldikçe araç hızının daha üst bir seviyeye yerleştiği net bir şekilde görülmektedir.



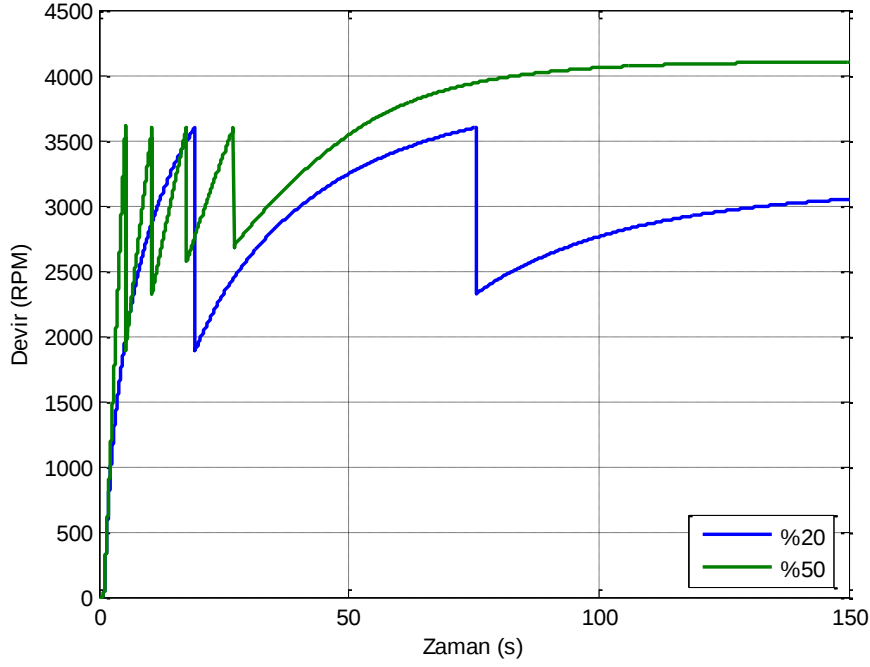
Şekil 4.34 : Araç hızı.



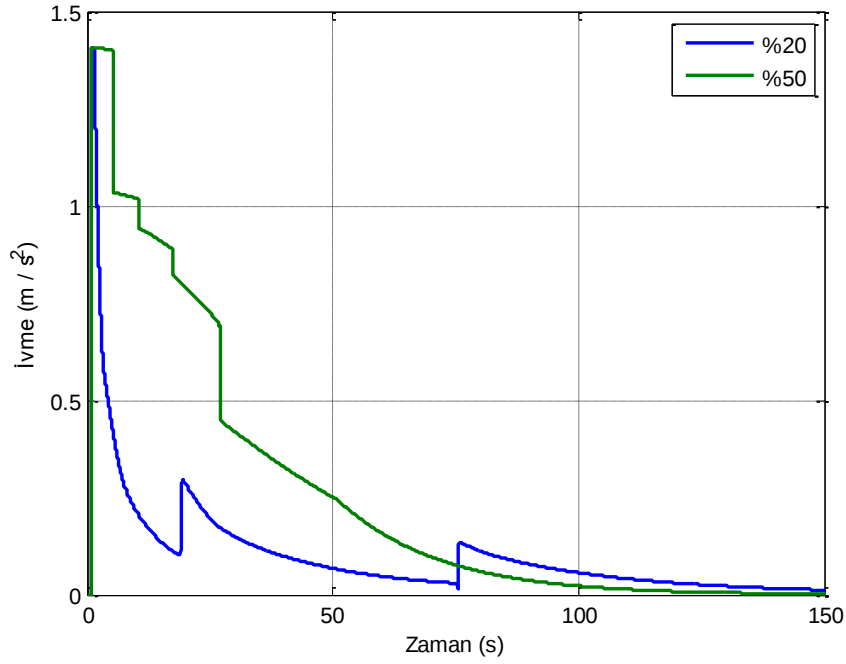
Şekil 4.35 : Vites değişimleri.

Ayrıca vites durumunu gösteren Şekil 4.35 incelendiğinde, araç gaz pedalına %20 oranında basıldığında en fazla 3. vites kadar çıkabilirken, %50 basıldığında ise 5. vites kadar çıkabilmiştir.

Şekil 4.36'daki grafikte, motor devrinin 3600 rpm olduğu veya geçtiği anlarda vitesin arttığı, gaz pedalına daha çok basıldığında motorun daha hızlı bir şekilde yüksek devirlere çıktığı, daha yüksek ivmeler ile hareketlendiğini görülmektedir.



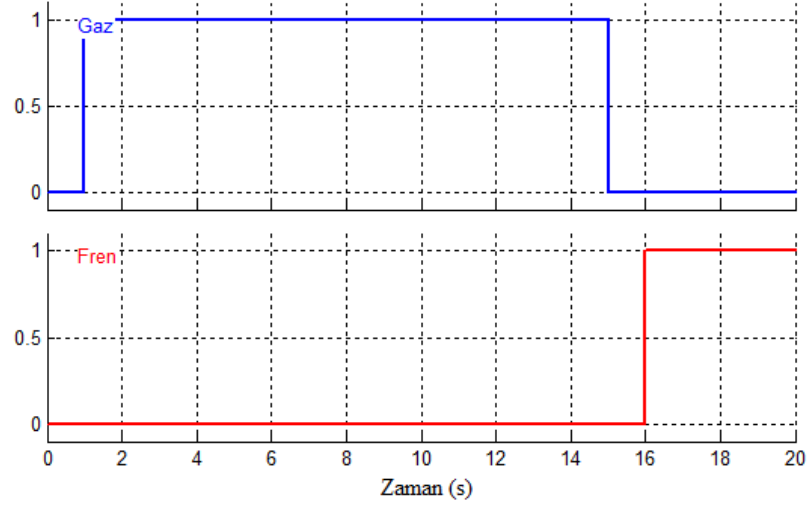
Şekil 4.36 : Motor devri.



Şekil 4.37 : Araç ivmesi.

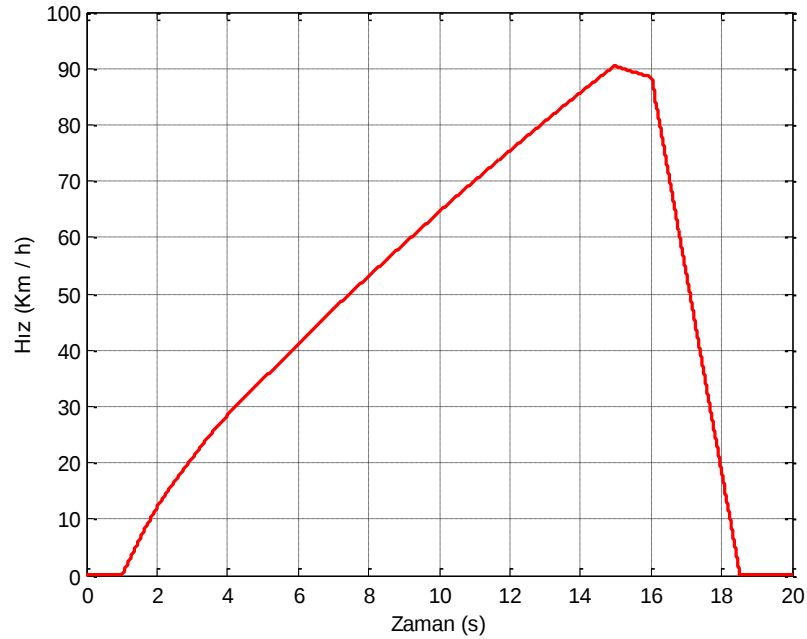
4.3.4 Aracın hızlanma ve frenleme simülasyonu

Bu deneyde gaz ve fren girişlerine uygulanan sinyaller Şekil 4.38’de verilmiştir. Sinyaller 0 – 1 arasında değiştiğinden pedal basış oranı %0 - %100 oranında değişmektedir. Aynı zamanda yol eğimi 0° ve rüzgâr hızı 0 m/s olarak kabul edilmiştir.



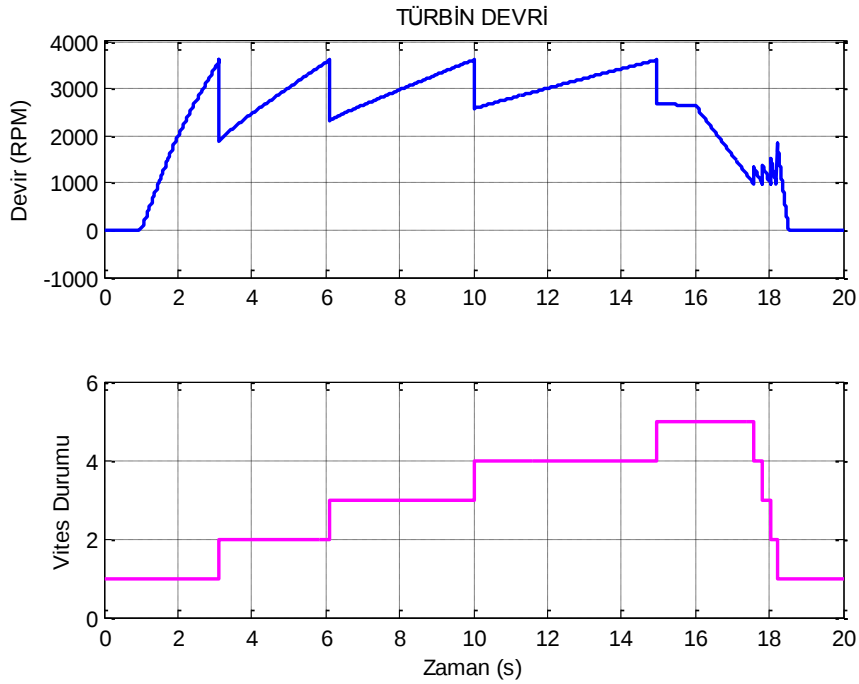
Şekil 4.38 : Uygulanan gaz ve fren sinyalleri.

Şekil 4.39’da araç hızı değişimi görülmektedir. 15.- 16. saniyeler arasında araç kendiliğinden yavaşlamakta, 16. saniyeden sonra ise fren pedalına %100 basıldığı için yaklaşık 2 saniye içinde durmaktadır.



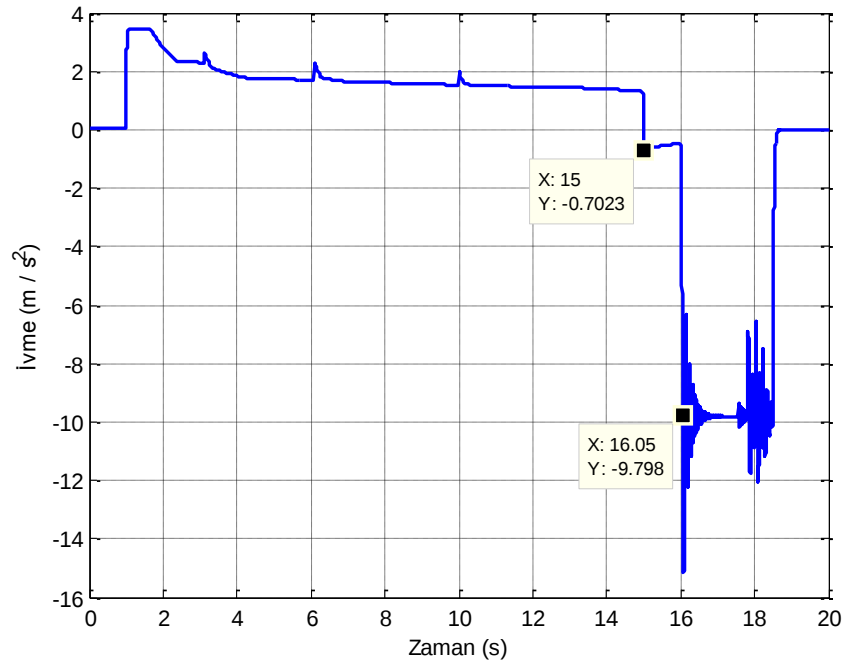
Şekil 4.39 : Araç hızı.

Şekil 4.40’da türbin devrine ve vites geçişlerine ait grafikler verilmiştir. Frene basıldığında vites hızlı bir biçimde küçülmektedir.

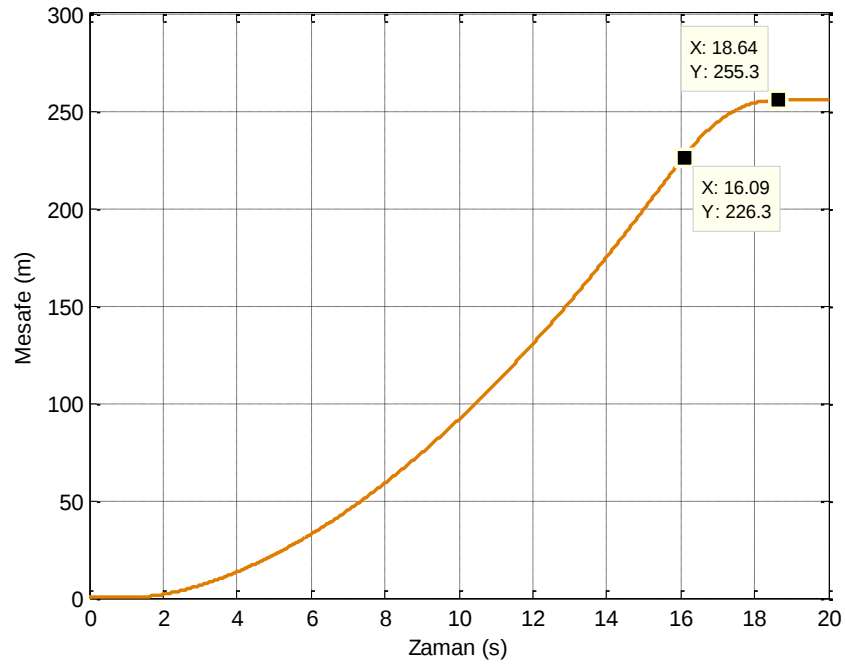


Şekil 4.40 : Türbin devri-vites durumu.

Araç ileri yönde giderken en yüksek ivmelenmeyi ilk kalkış anında göstermektedir. Şekil 4.41’de görüldüğü gibi 15. ve 16. saniyeler arasında frene ya da gaza basılmadığı için araç kendi iç dinamiklerinden ve dış etkenlerden (bağlı rüzgâr) dolayı yaklaşık $-0,7 \text{ m/s}^2$ gibi düşük bir ivmeyle yavaşlamıştır. 16. saniyede yüksek bir oranda frene basıldığı için ABS modeli devreye girmiş ve aracı kısa sürede durdurmuştur. Bunun gerçekleşebilmesi için yavaşlama ivmesinin yer çekimi ivmesinin negatif değerine eşit olması gerekmektedir. Tasarlanan modelde ABS araç ivmesi bu değere yakın tutulmaya çalışılmıştır. Kayma gerçekleştiğinde ise fren sisteminden tekerleklerle giden durdurma kuvvetini azaltarak tekerleklerin tekrar yere tutunmasını Şekil 4.42’de olduğu gibi sağlar.

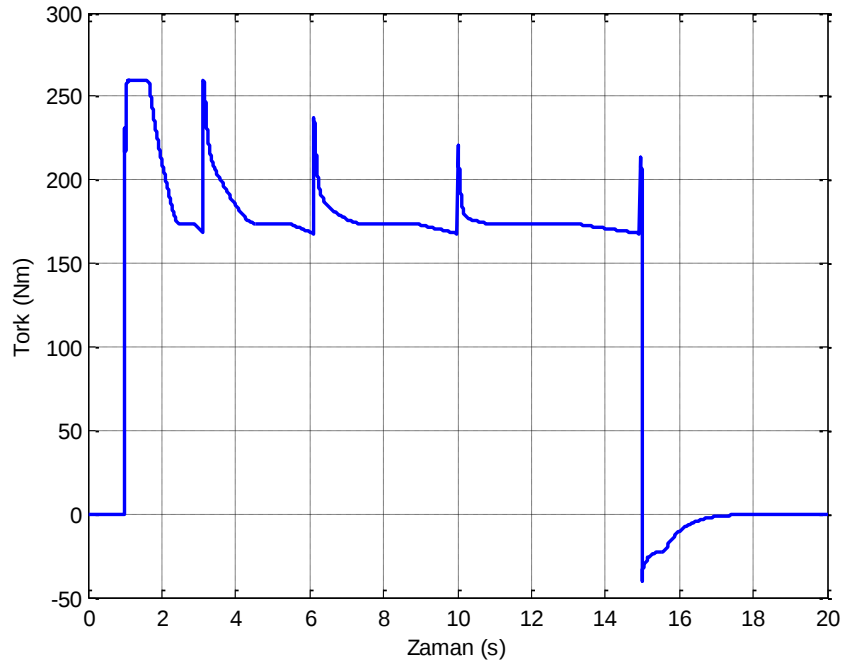


Şekil 4.41 : Araç ivmesi.

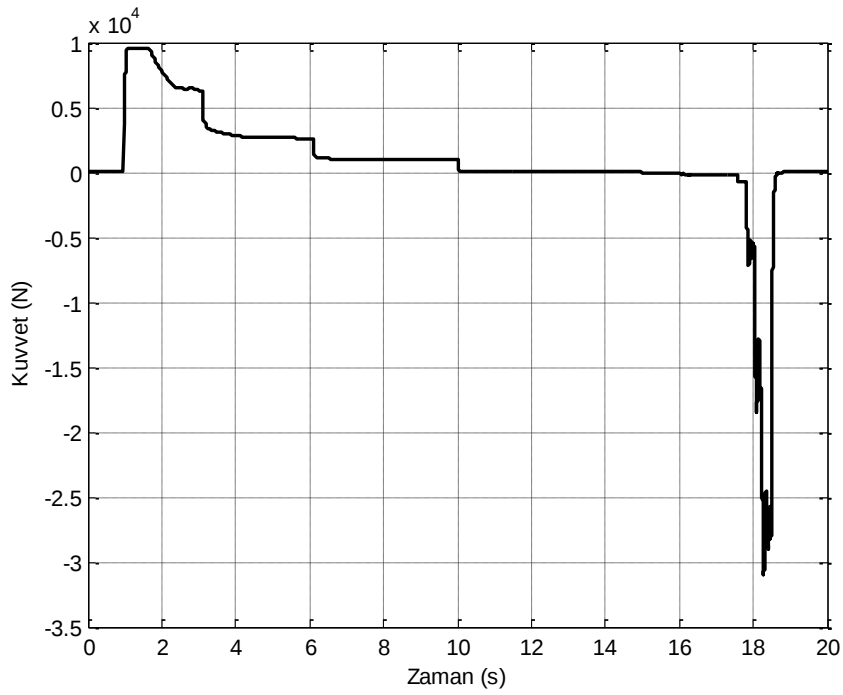


Şekil 4.42 : Alınan yol.

Şekil 4.43’de vites dişli kutusunun girişindeki, tork dönüştürücünün çıkışındaki tork değerini gösteren grafiğe bakıldığında gaz pedalına basılma bırakıldığı andan (15. saniye) araç durana kadar motor freni de denilen negatif tork ortaya çıkmıştır. Bunun sebebi motorun tekerlere göre daha yavaş kalması ve teker için bir yük gibi görülmesidir.

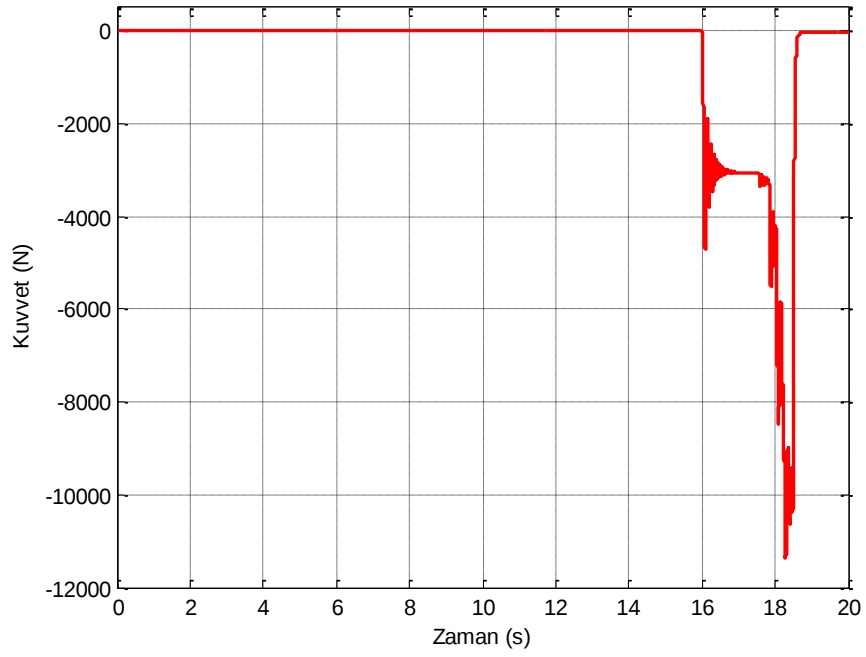


Şekil 4.43 : Vites kutusu girişi.



Şekil 4.44 : İvmelenme direnci.

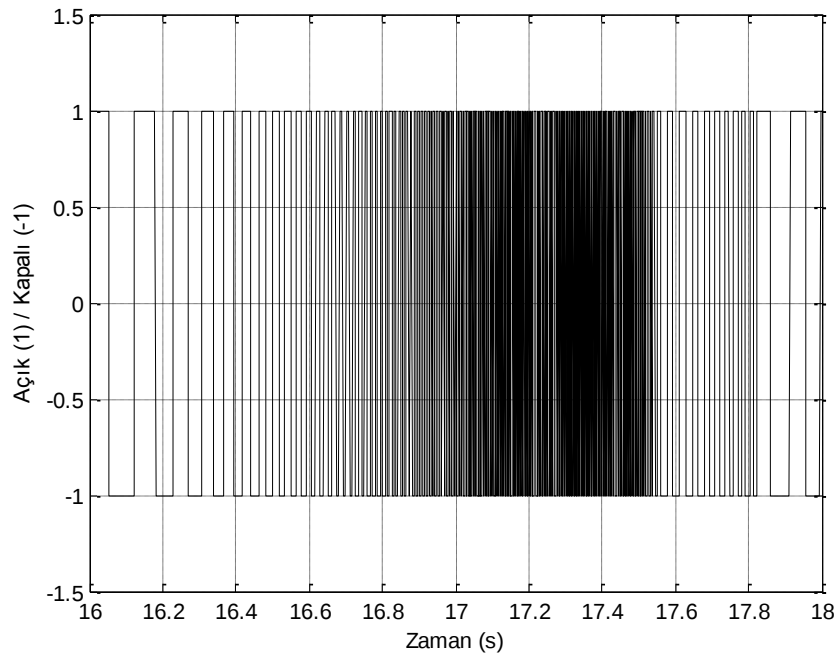
Dönen parçaların ilk hareketini vermek veya var olan hareketini durdurmak için gerekli kuvvet Şekil 4.45’de görülmektedir.



Şekil 4.45 : Toplam fren kuvveti.

Frene basılmadığı anda tekerleklere giden fren kuvveti 0'dır. Basıldığı anda ise ABS tekerleklerin kilitlenmeyeceği bir kuvvet oluşturmaya çalışır.

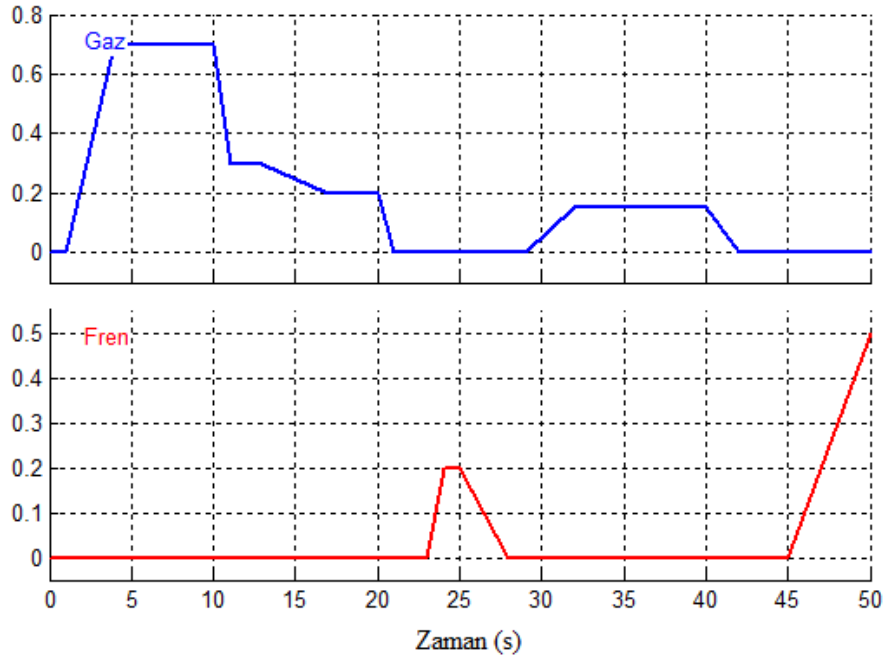
Simülasyonda oluşturulan ABS sisteminin fren balatalarına gönderdiği aç-kapa sinyali Şekil 4.46'da verilmiştir. 16. saniye'de frene basıldığı için grafik bu andan itibaren gösterilmiştir.



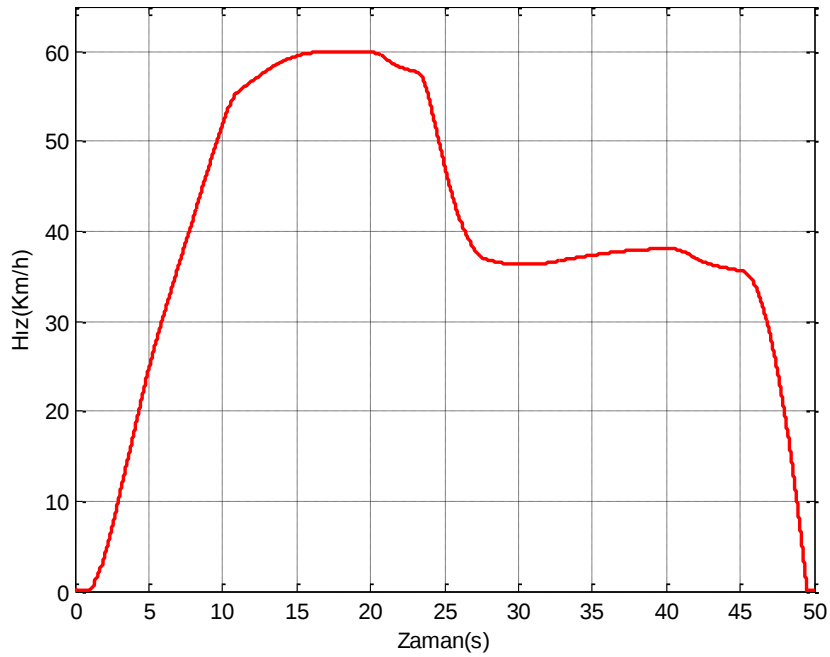
Şekil 4.46 : Aç kapa sinyali.

4.3.5 Farklı gaz ve fren giriş değerleri simülasyonu

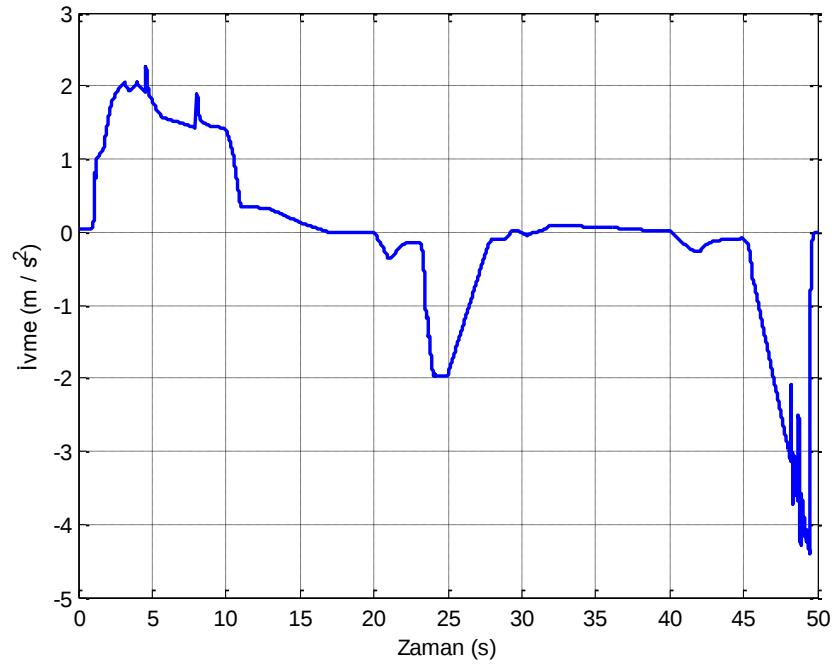
Bu deneyde gaz ve frene verilen girişler Şekil 4.47’de gösterildiği gibidir. Yol eğimi 0° olarak simülasyona girilmiştir. Maksimum 5 m/s, minimum -5 m/s hızında rastgele rüzgâr üretilmiştir. Bu girişlere göre aracın hız, ivme ve yol grafikleri elde edilmiştir.



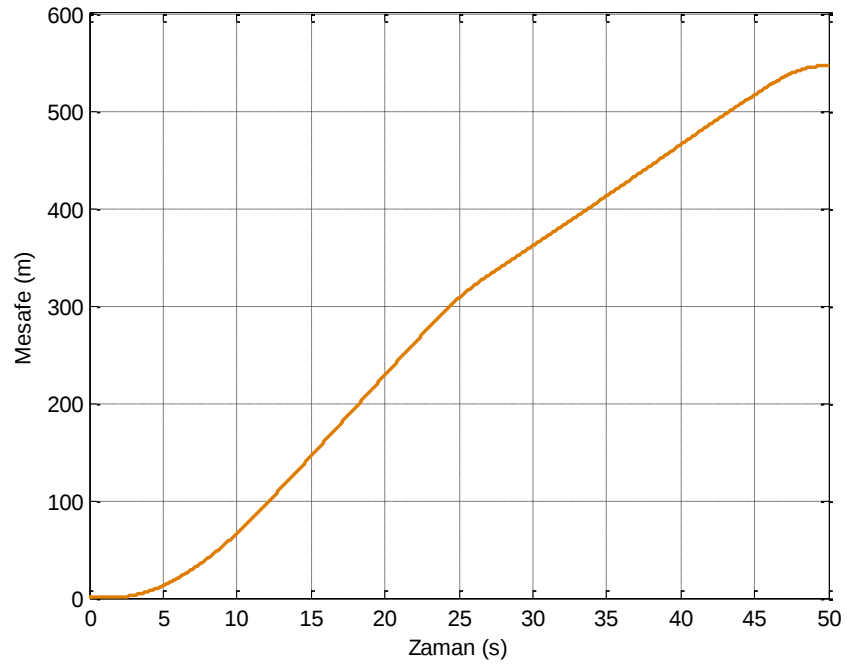
Şekil 4.47 : Uygulanan giriş.



Şekil 4.48 : Araç hız değişimi.



Şekil 4.49 : Araç ivme değişimi.



Şekil 4.50 : Alınan yol.

5. KONTROLÖR TASARIMI VE DENEY SONUÇLARI

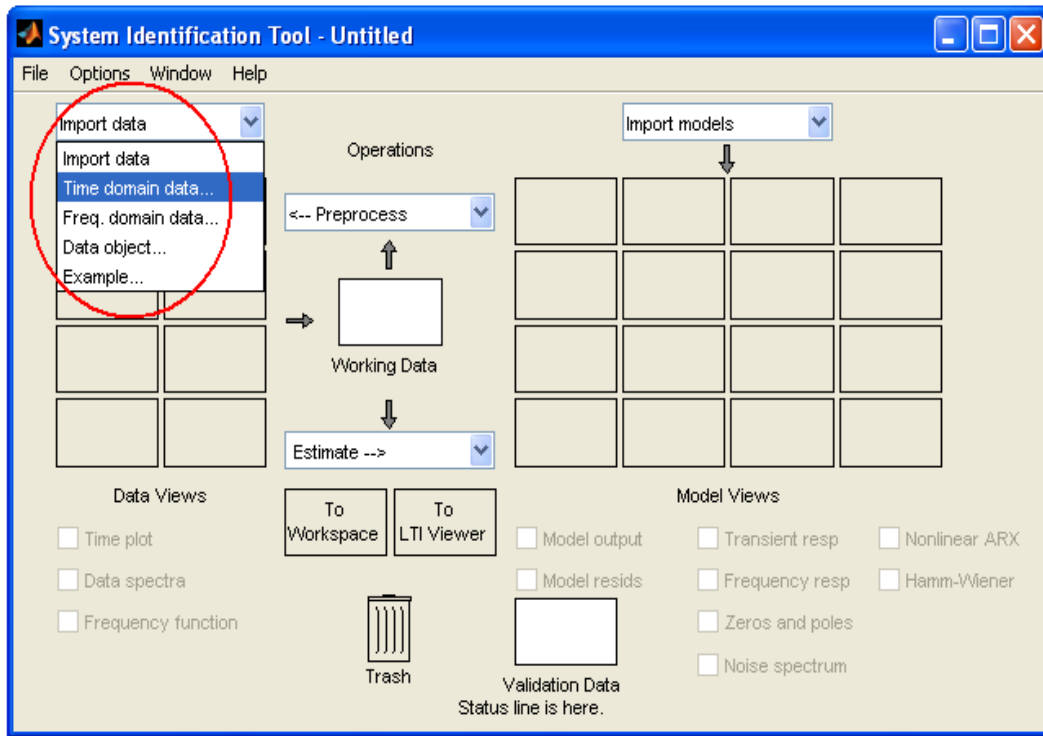
Yapılan çalışmalar ve ortaya konan simülasyonlar sonucunda hız kontrolörünün tasarımı ve kontrolöre ait katsayıların hesaplanması bu bölümde gerçekleştirilmiştir. Bunun için öncelikli olarak ivmelenme ve fren sistemine ait transfer fonksiyonları elde edilmiş ve her ikisi için de kapalı çevrim bir model kurulup uygun kontrolcü katsayıları belirlenmiştir. Simülasyon ortamında ayrı ayrı denenilen bu modeller gerçek sisteme de aktarılmış ve çeşitli deneyler yapılarak kontrolcüler iyileştirilmiştir. Son olarak her iki kontrolcünün birlikte çalışmaları sağlanmış ve hız kontrolcüsü oluşturulmuştur.

5.1 Sisteme Ait Transfer Fonksiyonları

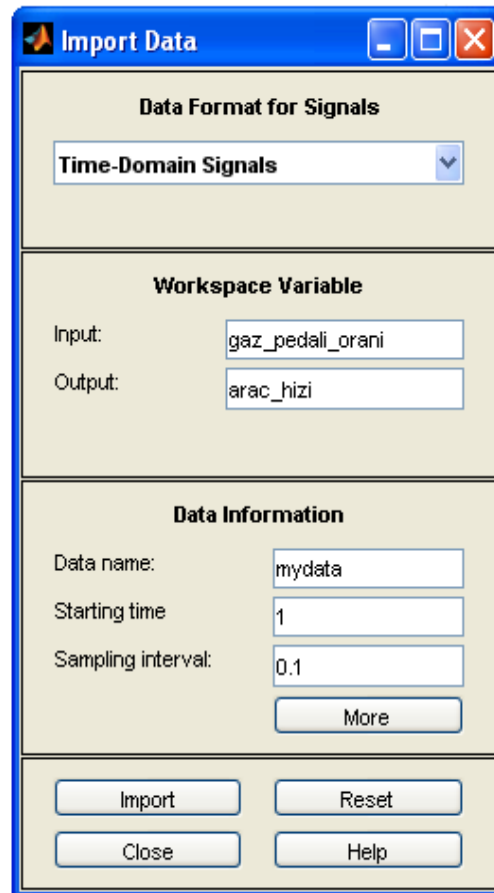
Her ne kadar araca ait matematiksel model çıkartılmış olsa da aracın tüm parametreleri bilinmediğinden (dönen parçaların ataleti, motor haritası vb.) gaz ve fren sisteminin transfer fonksiyonlarının belirlenmesi için farklı bir yol izlenmiştir. Uygulanan bu metotta önceki bölümlerde tasarlanmış yazılım ve donanımlar kullanılarak deneysel ortamda kontrolör olmadan her iki sisteme de ayrı ayrı birim basamak giriş uygulanmış ve çıkışlar kaydedilmiştir. Araçtan alınan bu deneysel veriler sayesinde MATLAB’de bulunan System Identification Toolbox’tan yararlanılarak “tahmini” transfer fonksiyonları belirlenmiştir.

5.1.1 İvmelenme sistemine ait transfer fonksiyonunun elde edilmesi

İvmelenme sistemi için elde edilecek transfer fonksiyonunun girişi gaz pedalı basış oranı, çıkışı ise araç hızı olarak belirlenmiştir. Bu aşamada oluşturulmuş olan bilgisayar yazılımı ve elektronik donanım ile sisteme birim basamak giriş uygulanmış ve çıkış olan araç hızı CAN kart yardımıyla 0.1 saniyelik örnekleme periyotlarıyla kaydedilmiştir. Daha sonra deneysel olarak elde edilen bu veriler MATLAB’e girilmiş ve Şekil 5.2’de de görüldüğü gibi “gaz_pedali_orani” ve “arac_hizi” isimleriyle zaman domaininde System Identification Toolbox’a aktarılmıştır.

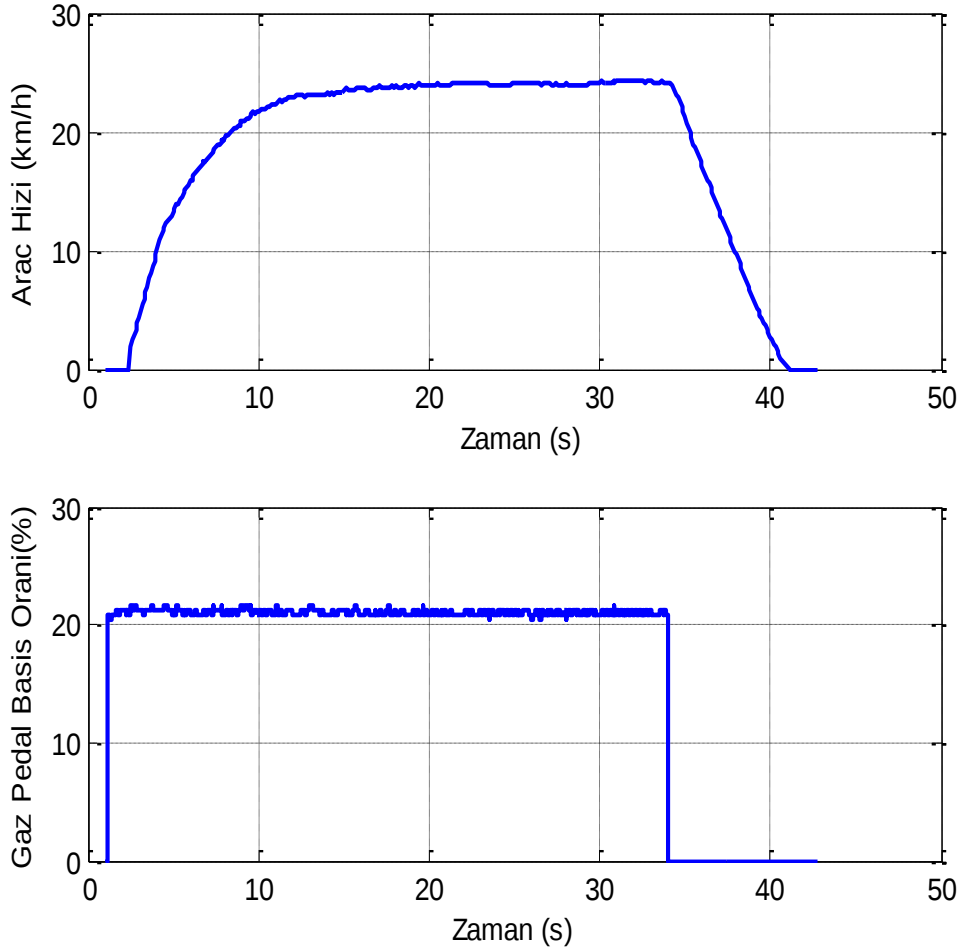


Şekil 5.1 : System Identification Toolbox



Şekil 5.2 : Verilerin SI Toolbox'a aktarılması.

Şekil 5.3’de araç hızı ve gaz pedalı basış oranı ilişkisi verilmiştir. Gaz pedalı basış oranı %22 olduğunda araç hızı yaklaşık olarak 25 km/h ’a çıkmış, %0 olduğunda ise araç hızının belli bir süre sonra (fren müdahalesi olmaksızın) 0 km/h ’a düşmüştür.



Şekil 5.3 : Araç hızı-gaz pedalı basış oranı grafiği.

Veriler System Identification Toolbox’a aktarıldıktan sonra “estimate” bölümünden “process model” seçeneği seçildiğinde Şekil 5.4’de görüleceği gibi sistemin parametreleriyle ilgili çeşitli ayarların belirlenebildiği pencere açılmaktadır. Tahmin edilecek sistemin derecesi, köklerin tipi (reel yada kompleks), kutup, sıfır veya gecikme eklenip eklenmeyeceği bu bölümden ayarlanır.

İvmelenme sistemi içerisinde durum değişkeni olarak hız ve ivme barındırdığından sistem ikinci dereceden seçilmiştir. Aynı zamanda hem ECU’deki hem de geliştirilen diğer yazılımlardaki kod işleme süreleri göz önünde bulundurularak sisteme 0.03 saniyelik bir gecikme eklenmiştir.

Process Models

Model Transfer Function

$$\frac{K \exp(-T_d s)}{(1 + T_{p1} s)(1 + T_{p2} s)}$$

Poles

2 All real

☐ Zero
☒ Delay
☐ Integrator

Parameter Known	Value	Initial Guess	Bounds
K	1.5744	1.5744	[-Inf Inf]
Tp1	0.7851	0.7851	[0.001 Inf]
Tp2	0.3538	0.3538	[0.001 Inf]
Tp3	0	0	[0.001 Inf]
Tz	0	0	[-Inf Inf]
Td	0.03	0.03	[0 3]

Initial Guess

☐ Auto-selected
☐ From existing model:
☒ User-defined: Value-->Initial Guess

Disturbance Model: None Initial state: Auto
Focus: Simulation Covariance: Estimate Options...

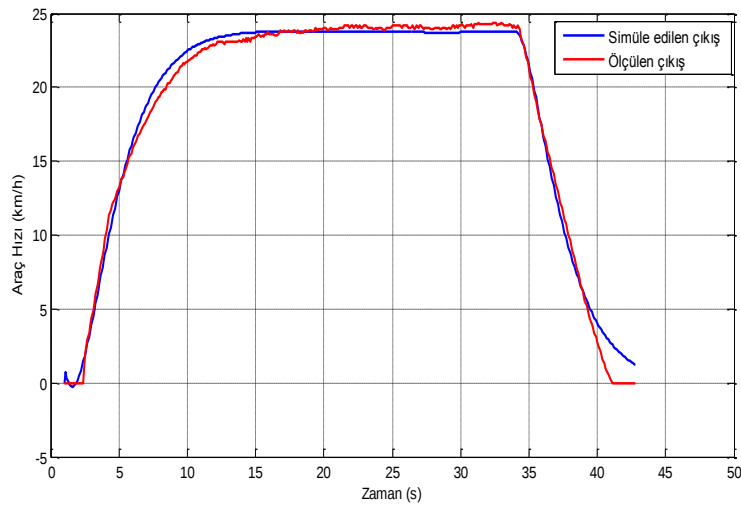
Iteration 20 Fit: 0.513 Improvement 0 % ☐ Display Stop Iterations

Name: P2D Estimate Close Help

Şekil 5.4 : Tahmini transfer fonksiyonu için parametre belirleme.

“Estimate” işlemi gerçekleştirildiğinde tahmini transfer fonksiyonunun çıkışıyla gerçek transfer fonksiyonunun çıkışının Şekil 5.5’deki gibi %91.27 oranında örtüştüğü görülmüştür. Elde edilen transfer fonksiyonu ise eşitlik 5.1’de verilmiştir.

$$T.F = \frac{1.5744}{(1.2736s + 1)(2.826s + 1)} = \frac{1.5744}{3.6s^2 + 4.1s + 1} \quad (5.1)$$

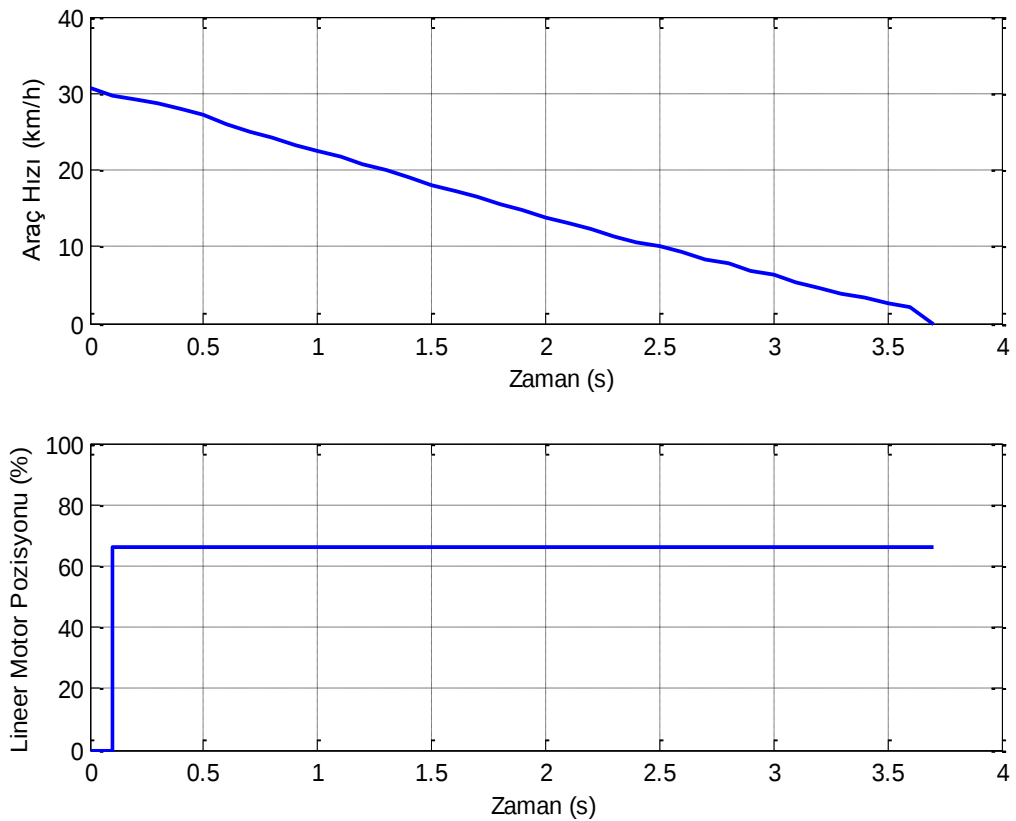


Şekil 5.5 : Tahmini çıkış-ölçülen çıkış karşılaştırılması (Benzerlik %91.27).

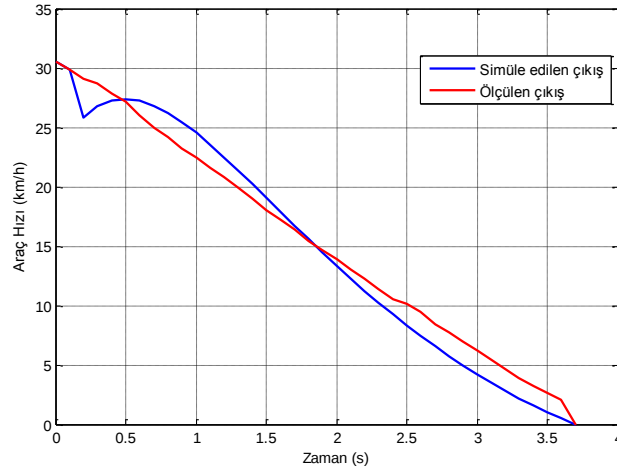
5.1.2 Fren sistemine ait transfer fonksiyonunun elde edilmesi

Fren sistemi için elde edilecek transfer fonksiyonunun girişi fren pedalına müdahale eden lineer motor pozisyonu, çıkışı ise ivmelenme sisteminde olduğu gibi araç hızıdır. Fakat deneysel ortamda veriler toplanırken fren sisteminin işleyişini görebilmek için ivmelenme sisteminden farklı olarak araç önce belli bir hıza getirilmiş, ardından birim basamak giriş uygulanmıştır. Sistemin giriş-çıkış grafiği Şekil 5.6'da, bu giriş-çıkışa göre elde edilen tahmini ve gerçek transfer fonksiyonlarının benzerliği Şekil 5.7'de verilmiştir.

Fren sistemi mekanik olarak kontrol edildiği için 0.15 saniyelik bir gecikme eklenmiştir. İvmelenme sisteminde olduğu gibi sistem ikinci dereceden seçilmiştir. Lineer motora pozitif yönde bir etki varken araç hızının negatif yönde olmasından dolayı sisteme ait transfer fonksiyonu ters işaretli (eksi) çıkmıştır.



Şekil 5.6 : Araç hızı-fren pedalı basış oranı grafiği.



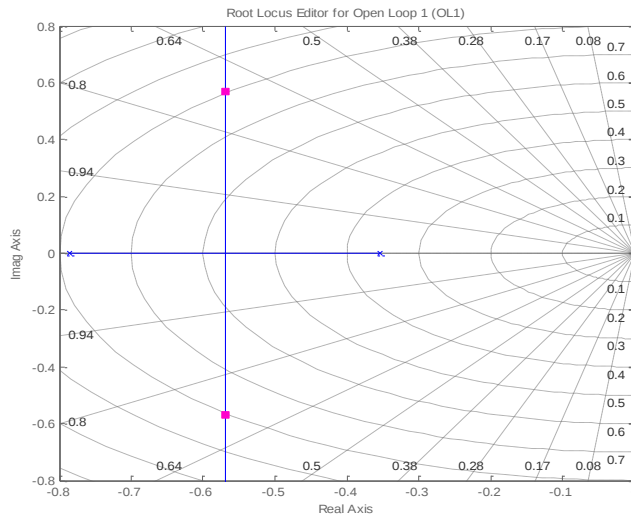
Şekil 5.7 : Tahmini çıkış-ölçülen çıkış karşılaştırılması (Benzerlik %83.44).

$$T.F = \frac{-0.12}{1.14s^2 + 2.26s + 1} \quad (5.2)$$

5.2 Kontrolör Yapısı ve Katsayıların Bulunması

5.2.1 İvmelenme sistemi için kontrolör katsayıları

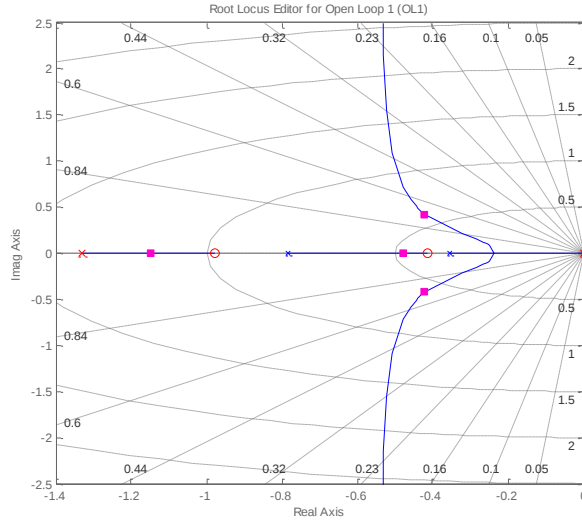
İvmelenme sistemine ait transfer fonksiyonunun açık çevrim Root-Locus analizi Şekil 5.8’de görüldüğü gibidir. Sisteme sadece oransal kontrolör (P) eklendiği varsayılarak sönüm oranı (ζ) 0.707 olarak ayarlandığında çalışma noktaları $s_{1,2} = -0.569 \pm 0.57i$ olmakta ve doğal frekans (ω_n) ise 0.806 rad/sec değerine denk gelmektedir.



Şekil 5.8 : İvmelenme sistemi için açık çevrim Root-Locus analizi.

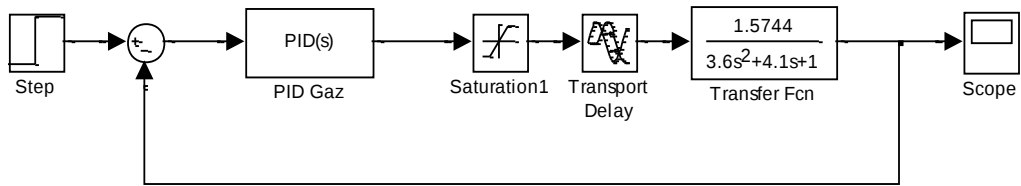
Sisteme, katsayıları iyileştirilmemiş PID kontrolör eklendiğinde ise Root-Locus analizi Şekil 5.9'da gösterildiği gibidir. Bu analizde PID katsayıları $P=11.395$, $I=0.33516$, $D=0.8043$, $N=1.33$ olarak bulunmuştur. N burada türevsel etkinin filtre katsayısıdır.

Sönüm oranı 0.707 olarak ayarlandığında çalışma noktaları $s_{1,2} = -0.421 \pm 0.422i$ olmakta ve doğal frekans ise 0.596 rad/sec değerine denk gelmektedir.



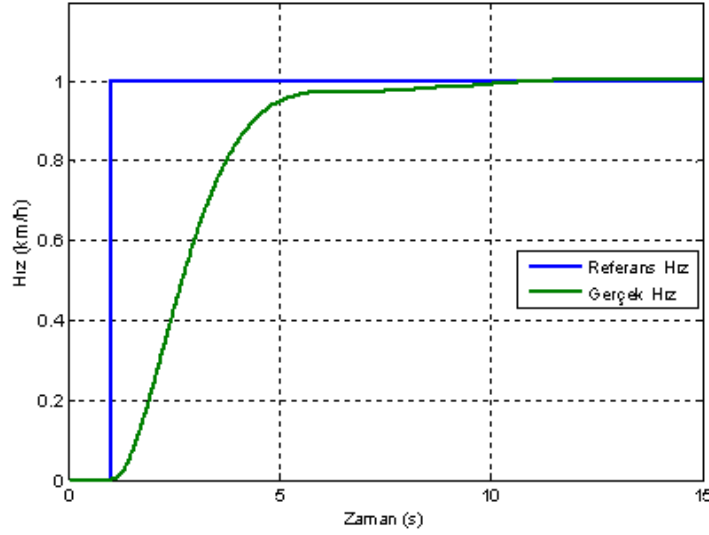
Şekil 5.9 : Kontrolörlü Root-Locus analizi.

Elde edilen transfer fonksiyonu, iyileştirilmemiş PID kontrolörle birlikte MATLAB/Simulink'de kapalı çevrim bir sisteme dönüştürülmüş ve referans girişi birim basamak uygulanarak Şekil 5.11'deki gibi bir cevap alınmıştır.



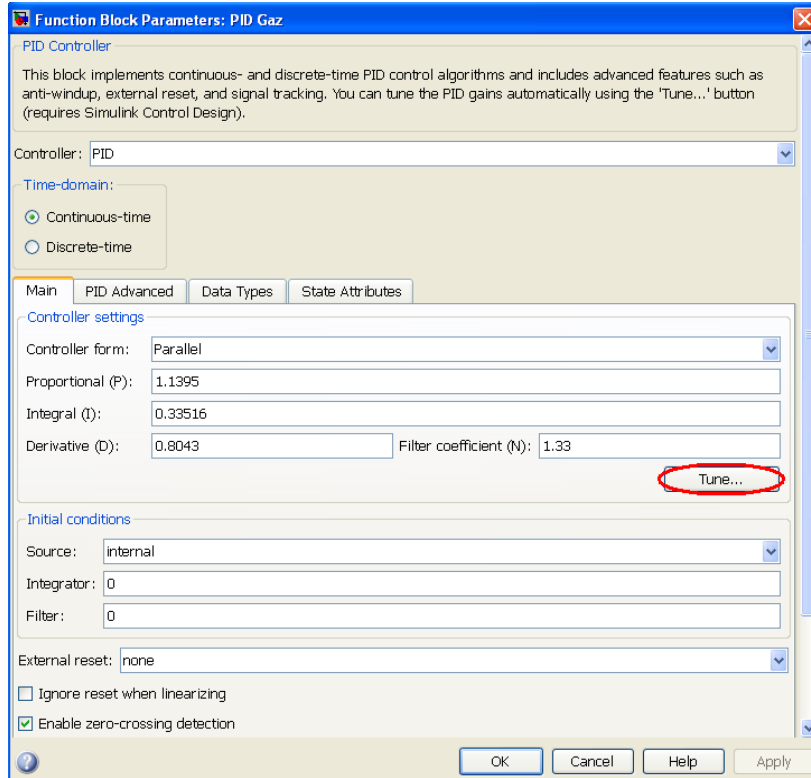
Şekil 5.10 : İvmelenme sistemi MATLAB simülasyonu (1).

Çıkış incelendiğinde hiç aşım olmaması bir avantaj olarak görülse de, sistemin geç cevap (yaklaşık 7-8 saniye) vermesi büyük bir dezavantajdır. İvmelenmenin daha hızlı ve iyi bir çıkış verebilmesi için kontrol katsayılarının iyileştirilmesi gerekmektedir.

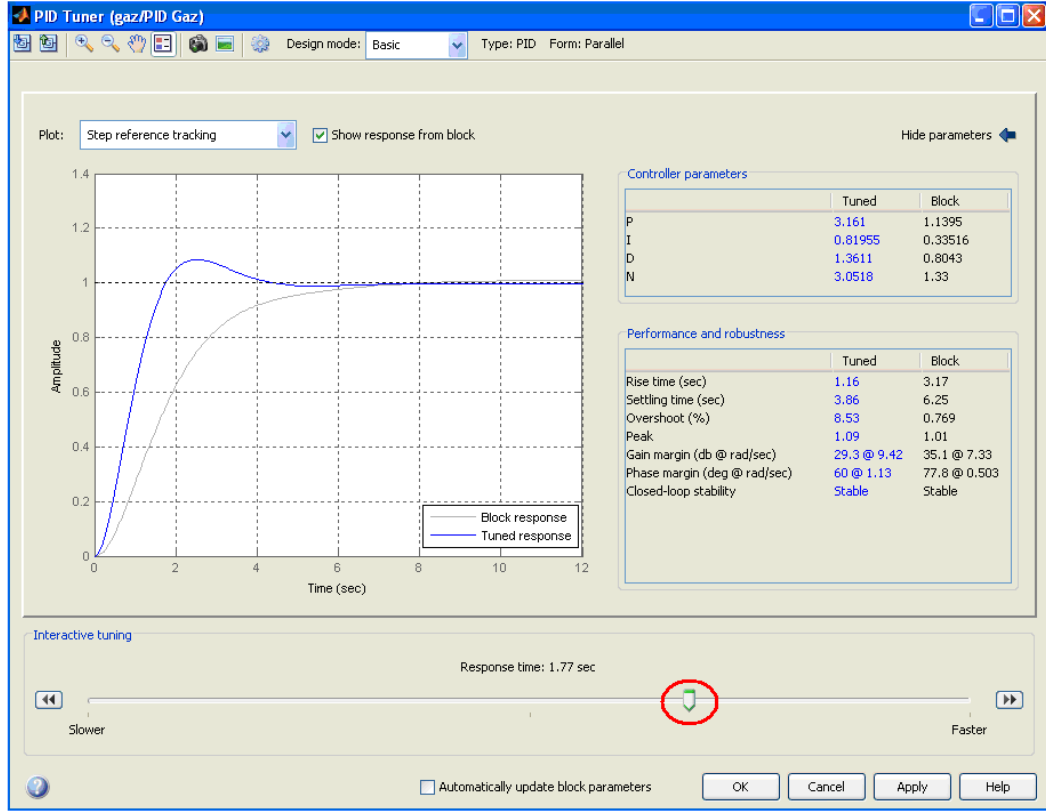


Şekil 5.11 : İvmelenme sistemi çıkışı.

İyileştirme için ilk olarak PID bloğunun içerisinde bulunan “PID Tuner” çalıştırılmıştır. Şekil 5.13’de de görüldüğü gibi sistem cevabı süresi ayar çubuğu üzerinden değiştirildiğinde, bu çıkışı oluşturabilecek kontrolör katsayıları da “parametreler” bölümünde görülmektedir. MATLAB/PID Tuner kullanılarak yapılan bu işlemin sonunda en uygun katsayıların $P=3.161$, $I=0.81955$, $D=1.3611$, $N=3.518$ olduğu belirlenmiştir.

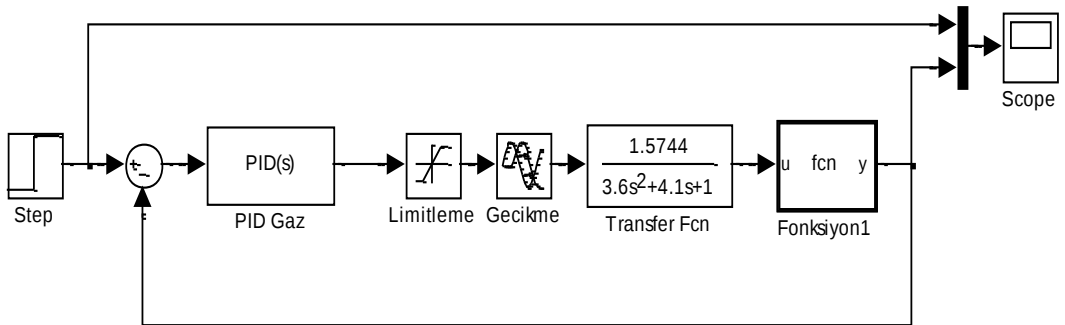


Şekil 5.12 : PID Tuner.



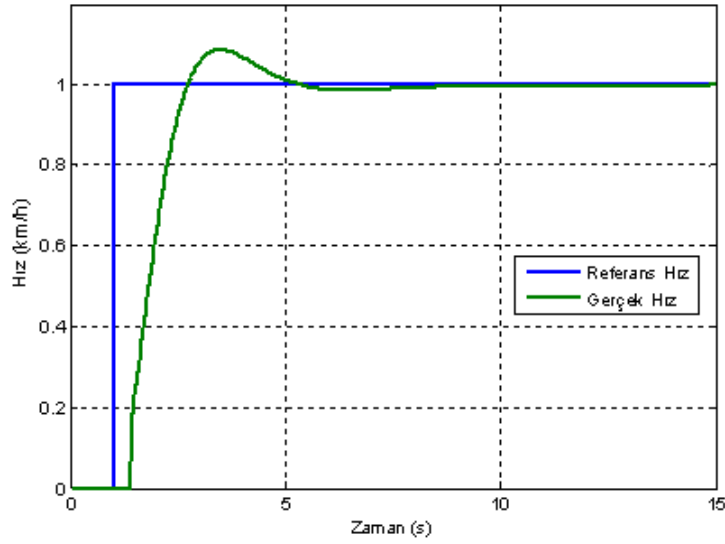
Şekil 5.13 : İyileştirilmiş PID katsayıları.

Deneyisel ortamdan alınan sonuçlar incelendiğinde aracın ilk hareketi esnasında gaz pedalına basış oranına bağlı olarak yaklaşık 1 saniyelik ölü bir zaman olduğu tespit edilmiştir. Bu gecikme sadece ilk hareketlenmede mevcut olduğundan Simulink modele hat gecikmesi olarak değil, Şekil 5.14’de görüldüğü gibi “Fonksiyon1” bloğu olarak eklenmiştir.



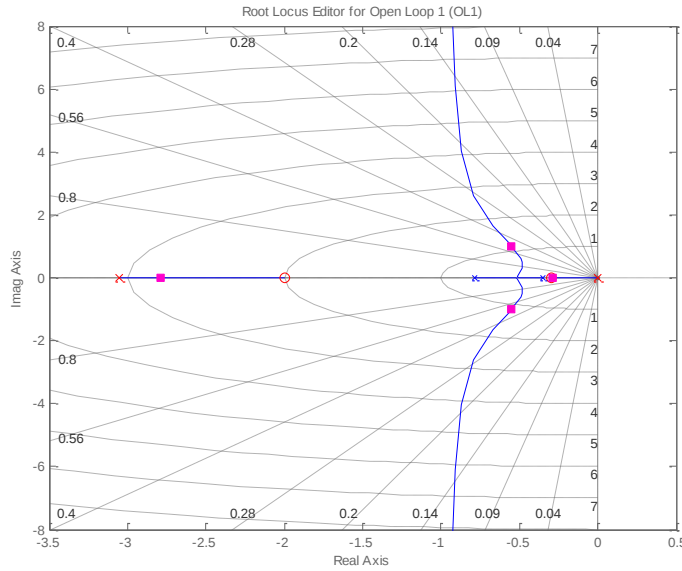
Şekil 5.14 : İvmelenme sistemi MATLAB simülasyonu (2).

İyileştirilen katsayılarla kurulan modele birim basamak giriş uygulandığında elde edilen çıkış Şekil 5.15’de olduğu gibidir.



Şekil 5.15 : İyileştirilmiş ivmelenme sistemi çıkışı.

İvmelenme sisteminin kökleri ile birlikte iyileştirilmiş PID kontrolöre ait sıfır ve kutupların bulunduğu noktalar kapalı çevrim çalışma noktaları ile birlikte Şekil 5.16'da gösterilen Root-Locus grafiğinde verilmiştir. Analizde de görülebileceği gibi sistem için en uygun çalışma anında sönüm oranı 0.481, kapalı çevrim kökler $s_{1,2} = -0.553 \pm 0.101i$, doğal frekans ise 1.15 rad/s değerine oturmaktadır.

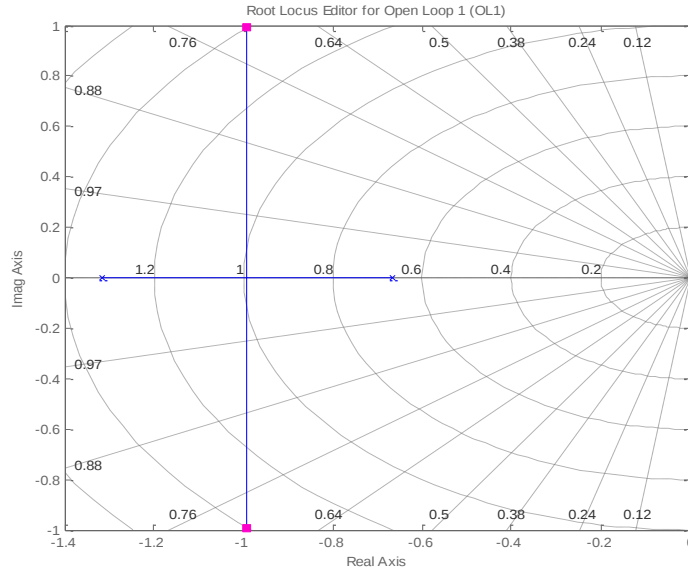


Şekil 5.16 : İyileştirilmiş kontrolörlü Root-Locus analizi.

5.2.2 Fren sistemi için kontrolör katsayıları

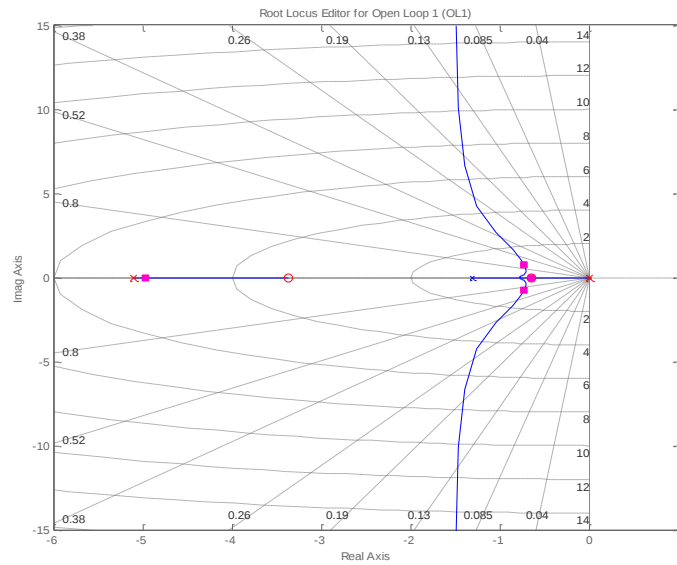
Frenleme sistemine ait açık çevrim Root-Locus analizi Şekil 5.17'de verilmiştir. Bu noktada ivmelenmede olduğu gibi sistemde sadece oransal kontrolör(P) olduğu

varsayılabilir. Transfer fonksiyonun negatif olan pay değeri ise kontrolör katsayılarının da ters işareti alınarak pozitif olarak kabul edilmiştir.



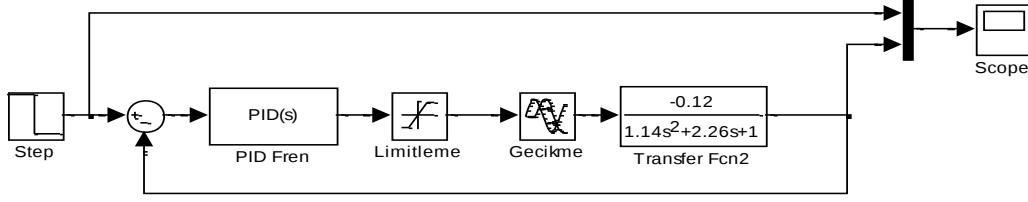
Şekil 5.17 : Fren sistemi için açık çevrim Root-Locus analizi.

PID kontrolden gelecek olan kutuplar ve sıfırlar analize eklendikten sonra sistemin sönüm oranı 0.707 değerine ayarlandığında Şekil 5.18’de de görülebileceği gibi çalışma noktası $s_{1,2} = -0.734 \pm 0.736i$, doğal frekans da 1.04 rad/s değerlerini alır. Bu durumda kontrolör katsayıları $P=11.72$, $I=6.51$, $D=2.93$ ve $N=5$ olmaktadır. Ancak bu kontrolör ile çalışan sistemin birim basamak cevabının yeterli olmadığı görülmüştür.

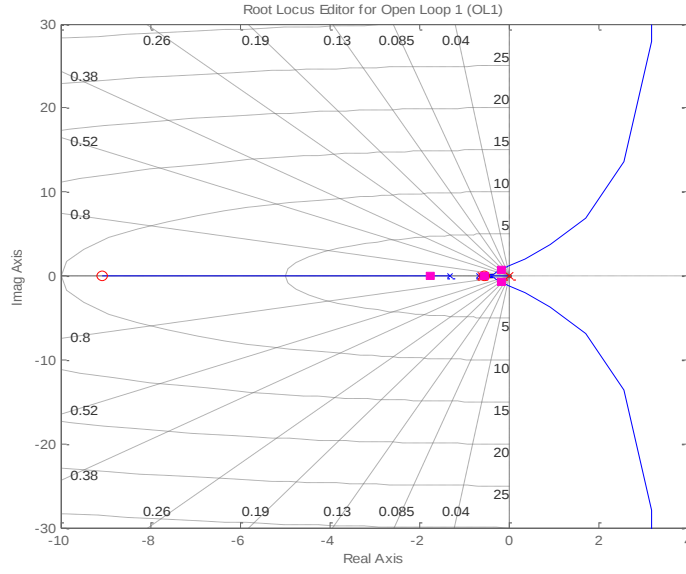


Şekil 5.18 : Fren sistemi için kontrolörlü Root-Locus analizi.

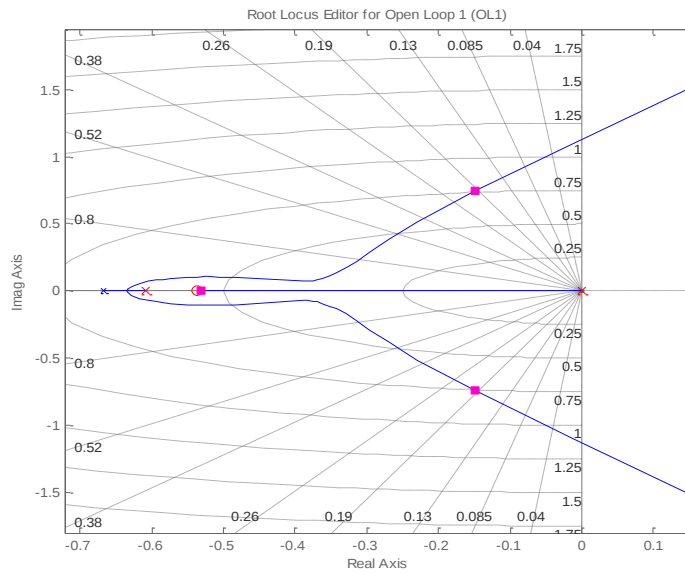
İvmelenmede olduğu gibi katsayılarını iyileştirmek için Şekil 5.19'daki Simulink model kurulmuş ve “PID Tuner” kullanılmıştır.



Şekil 5.19 : Fren sistemi MATLAB simülasyonu.

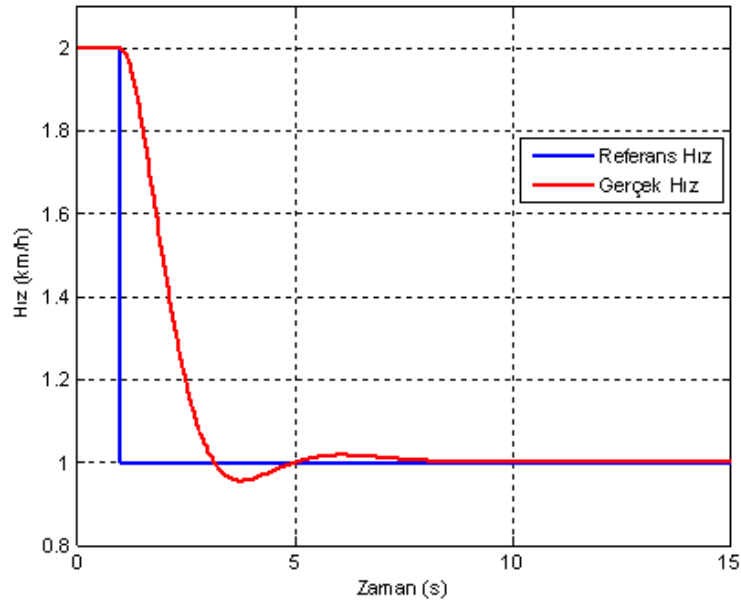


Şekil 5.20 : Fren sistemi için iyileştirilmiş kontrolörlü Root-Locus analizi(1).



Şekil 5.21 : Fren sistemi için iyileştirilmiş kontrolörlü Root-Locus analizi(2).

Uygun kontrolör katsayıları $P=16.6$, $I=8.4$, $D=1.73$ ve $N=1.64$ değerlerini almış ve bu değerlerle çalışan sistemin kapalı çevrim kökleri $s_{1,2} = -0.149 \pm 0.744i$, sönüm oranı 0.197 doğal frekansın 0.758 rad /s olarak bulunmuştur. Şekil 5.20 ve 5.21’de görüleceği gibi diğer Root_Locus’lardan farklı olarak asimtotlar imajinel eksenin sağ tarafına da geçmekte ve sistemin kararsız duruma gelme ihtimali doğmaktadır. Kritik noktadaki kapalı çevrim kazancı(K_{kritik}) 21.8 değerine eşittir. Fakat kapalı çevrim köklerin eksenin sol tarafında olmasında da anlaşılacağı gibi kapalı çevrim kazancı bu değerinin altındadır ($K=8.4$) ve sistem bu kontrolör katsayıları ile asla kararsız hale gelmemektedir. Birim basamak girişe karşılık gelen cevap ise Şekil 5.22’de verilmiştir.



Şekil 5.22 : İyileştirilmiş fren sistemi çıkışı.

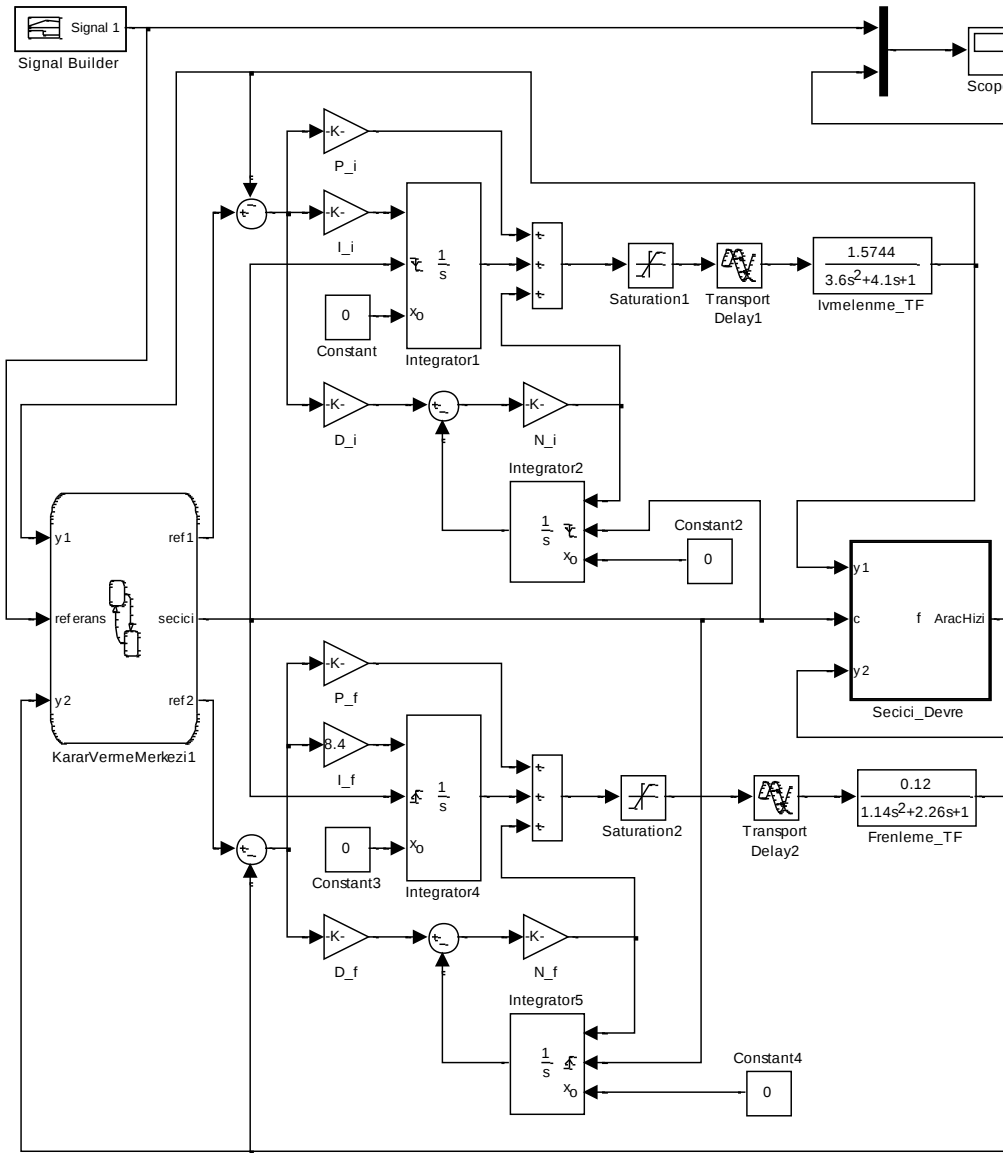
5.2.3 Genel yapının MATLAB ortamında simüle edilmesi

Frenleme ve ivmelenme için tasarlanan sistemler bir araya getirildiğinde Şekil 5.23’de modellendiği gibi hız kontrolörü oluşmaktadır. Buradaki en büyük problem araç hızına hangi kontrolörün etki edeceğidir. Bu sorun çalışma algoritması Şekil 5.24’de verilen “Karar Verme Merkezi” isimli akış diyagramı yardımıyla çözülmüştür. Araç hızı sisteme girilen referans hızın %12 altındaysa ivmelenme sistemi ve ona ait kontrolör, %10 üstündeyse frenleme sistemi ve ona ait kontrolör devreye girmektedir. Belirtilen %22’lik histerisis araç hızındaki küçük değişimlerde kontrolörler arası geçişe, diğer bir deyişle aracın sürekli gaz-fren yapmasına engel olmak içindir. Frenlemeden ivmelenmeye geçişteki hatanın tersi durumdaki hatadan

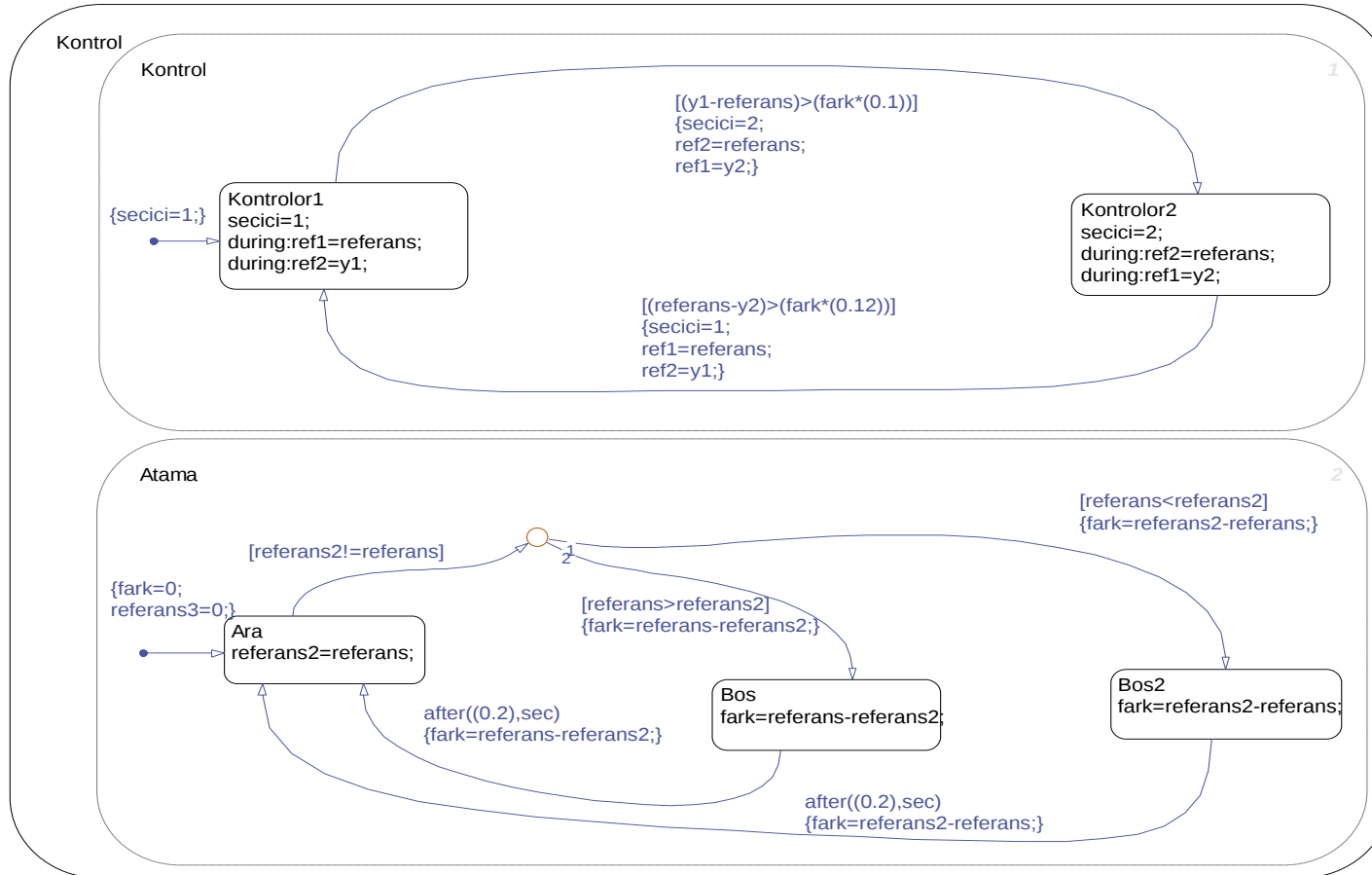
%2 fazla olmasının sebebi frenleme sistemi dışında aracı durdurmaya çalışan başka etkenlerin de olmasıdır.

Bu yapıda her zaman için sistemlerden biri aktif diğeri ise pasif durumdadır. Aktif olan sistemin referansı hız kontrolöre gelen ve aracın ulaşması gereken referans değeridir. Hem fren hem de ivmelenme için hesaplanan iyileştirilmiş katsayılar model içinde bulunan kontrolörlerde kullanılmıştır.

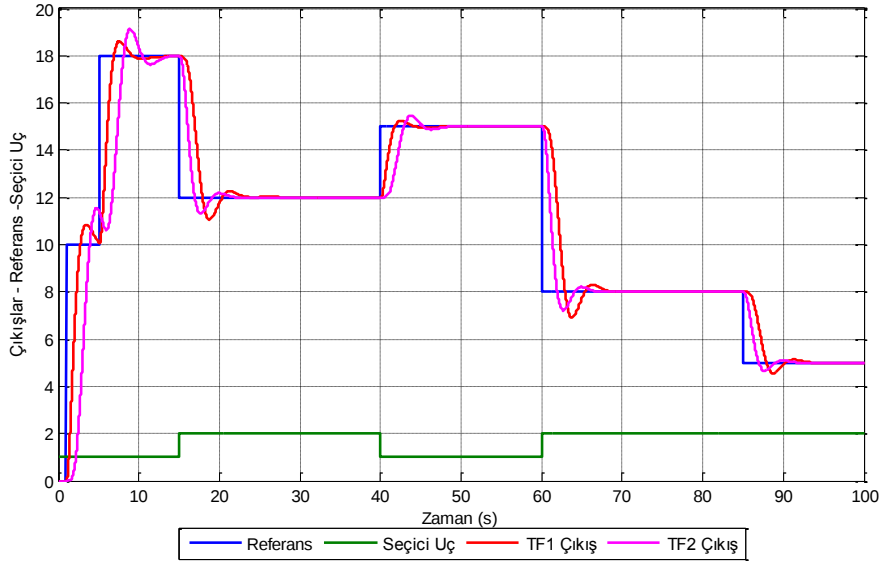
Hız kontrolörün ivmelenme ve frenleme sistemi arasındaki geçişleri esnasında her zaman aynı şekilde davranması için bu geçişlerde biriken integral etki sıfırlanmıştır. Şekil 5.14’de görülen kalkış anında bir gecikme oluşturan “fonksiyon1” bloğu “Secici_Devre” fonksiyonunun içine gömülmüştür.



Şekil 5.23 : Hız kontrolörün Simulink modeli.

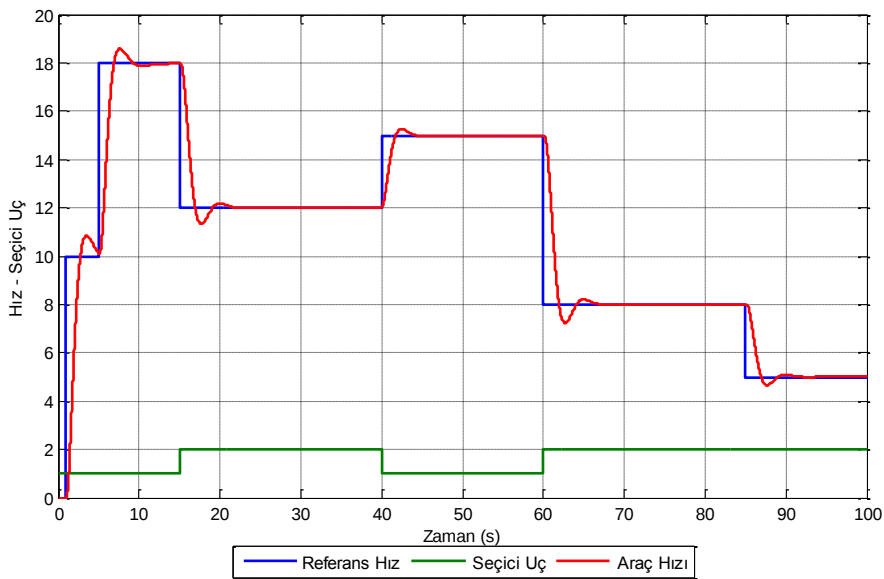


Şekil 5.24 : Karar verme merkezi akış diyagramı.



Şekil 5.25 : Fren ve ivmelenme sisteminin referans hız takibi.

Şekil 5.25’de referans hızın takibinde fren ve gaz sisteminin verdiği cevaplar görülmektedir. Grafik incelendiğinde ivmelenme sisteminin araç hızlanırken, frenleme sisteminin ise araç yavaşlarken daha iyi cevap verdiği görülmektedir. Yine aynı grafikte seçici ucun, diğer bir deyişle hangi kontrolörün aktif olacağını belirleyen işaretin aracın hızlanması gerektiğinde kontrolör 1’i seçtiği, aracın yavaşlaması gerektiğinde ise kontrolör 2’yi seçtiği görülmektedir. Şekil 5.26’da ise aynı sistem için araç hızı çıkışı verilmiştir.



Şekil 5.26 : Araç hızının referans değeri takibi.

5.2.4 Bilineer dönüşümler

Modelde hesaplanan ve simüle edilen s domainindeki(sürekli zaman) iyileştirilmiş kontrol katsayılarının gerçek sistemdeki yazılıma aktarılabilmesi için öncelikli olarak z domainine (ayrık zamana) dönüştürülmesi gerekmektedir. Bu işlem için eşitlik 5.3’de verilen bileneer dönüşüm yöntemi kullanılmıştır. Türev için bir filtreye sahip olan paralel formdaki PID kontrolörün denklemi eşitlik 5.4’de verilmiştir.

$$s = \left[\frac{2}{T} \frac{z-1}{z+1} \right] \quad (5.3)$$

$$PID(s) = K_p + \frac{K_i}{s} + K_d \left(\frac{Ns}{s+N} \right) \quad (5.4)$$

$$PID(s) = \frac{K_p s + K_i}{s} + \frac{K_d Ns}{s+N} = \frac{K_p s^2 + K_p Ns + K_i s + K_i N + K_d Ns^2}{s^2 + Ns} \quad (5.5)$$

s yerine z domainindeki karşılığı yazılırsa,

$$PID(z) = \frac{K_p \left[\frac{2}{T} \frac{z-1}{z+1} \right]^2 + K_p N \left[\frac{2}{T} \frac{z-1}{z+1} \right] + K_i \left[\frac{2}{T} \frac{z-1}{z+1} \right] + K_i N + K_d N \left[\frac{2}{T} \frac{z-1}{z+1} \right]^2}{\left[\frac{2}{T} \frac{z-1}{z+1} \right]^2 + N \left[\frac{2}{T} \frac{z-1}{z+1} \right]} \quad (5.6)$$

$$= \frac{\frac{K_p (2z-2)^2}{(Tz+T)^2} + \frac{2K_p Nz - 2K_p N}{(Tz+T)} + \frac{2K_i z - 2K_i}{(Tz+T)} + \frac{K_d N(2z-2)^2}{(Tz+T)^2} + K_i N}{\left[\frac{2}{T} \frac{z-1}{z+1} \right]^2 + N \left[\frac{2}{T} \frac{z-1}{z+1} \right]} \quad (5.7)$$

elde edilir. Denklemi 5.8’deki gibi daha sade bir hale getirilecek olursa katsayılar,

$$PID(z) = \frac{(Az^2 + Bz + C)}{z^2 + Dz + E} \quad (5.8)$$

$$A = (4K_p + 2K_p NT + 2K_i T + K_i NT^2 + 4K_d N) / (4 + 2N) \quad (5.9)$$

$$B = (2K_i T^2 N - 8K_p - 8K_d N) / (4 + 2N) \quad (5.10)$$

$$C = ((4K_p - 2K_p NT - 2K_i T + K_i NT^2 + 4K_d N) / (4 + 2N)) \quad (5.11)$$

$$D = -8 / (4 + 2N) \quad (5.12)$$

$$E = (4 - 2NT) / (4 + 2N) \quad (5.13)$$

olarak belirlenir. Eşitlik 5.8'in pay ve paydası z^2 'ye bölünürse eşitlik 5.14 elde edilir.

$$\frac{u(z)}{e(z)} = \frac{A + Bz^{-1} + Cz^{-2}}{1 + Dz^{-1} + Ez^{-2}} \quad (5.14)$$

Son olarak ayrık zamanda z 'li terimler giriş ve çıkışlara aktarılıp formül tekrar düzenlendiğinde denklem 5.16'daki gibi bulunmuş ve var olan bilgisayar yazılımına bu şekilde gömülmüştür. Sistemin örnekleme zamanı ise araç hızının CAN hattında yayınlanma periyoduna göre 100 ms olarak belirlenmiştir.

$$u(k) + Du(k-1) + Eu(k-2) = Ae(k) + Be(k-1) + Ce(k-2) \quad (5.15)$$

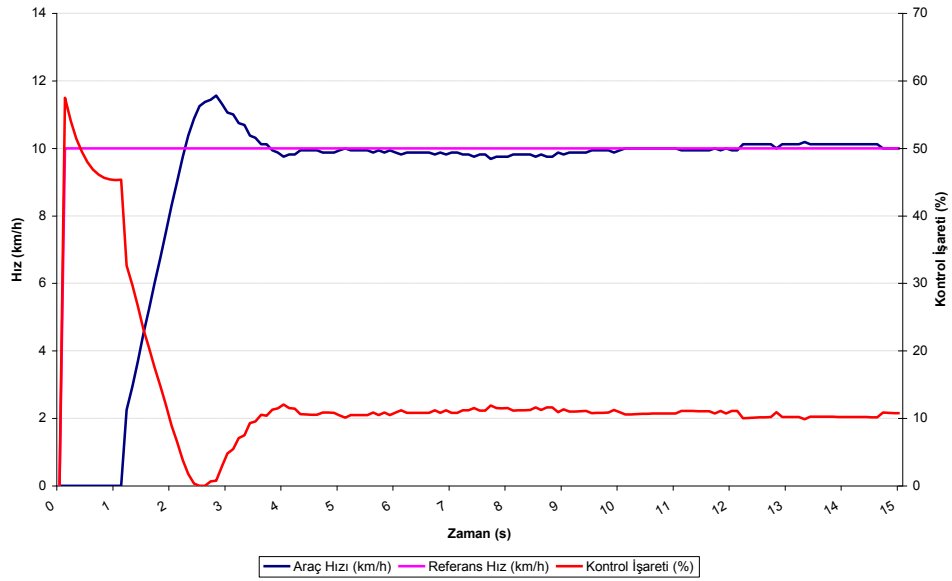
$$u(k) = Ae(k) + Be(k-1) + Ce(k-2) - Du(k-1) - Eu(k-2) \quad (5.16)$$

5.3 Deneysel Sonuçlar ve Kontrolörlere İnce Ayar Yapılması

Modelde kullanılan transfer fonksiyonlarının %85 ve üzerinde bir benzerlik oranıyla belirlenmesine rağmen birebir örtüşmemesi ve diğer parametrelerin (kod işleme süresi, hat gecikmesi vb.) yaklaşık olarak alınmasından dolayı kontrolör katsayılarına deneysel ortamda ince ayar (fine tuning) yapılması gerekmektedir. Bu işlemi gerçekleştirmek için yapılan deneyler sırasında iki sistem ayrı ayrı çalıştırılmış ve P, I, D değerlerinde küçük değişiklikler yapılmıştır.

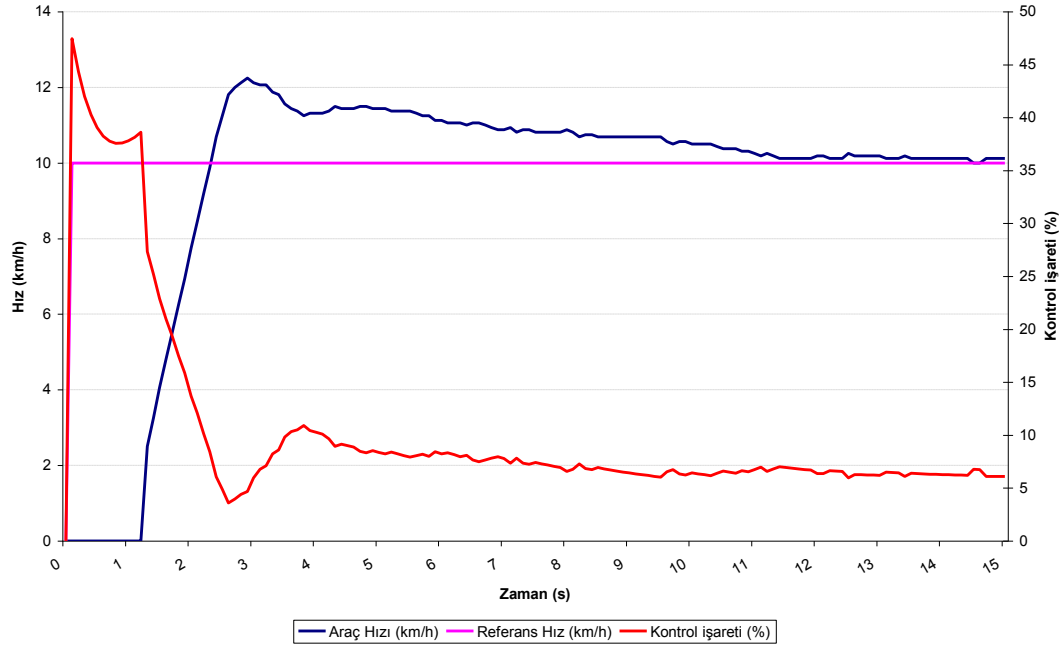
5.3.1 İvmelenme deneyleri ve iyileştirilmiş katsayılar

Modelde tespit edilen iyileştirilmiş kontrolör katsayıları bilgisayar yazılımına gömülüp referans hız 10 km/h olarak belirlendiğinde düze yakın bir deneysel ortamda elde edilen araç hızı ve üretilen kontrol işaretinin büyüklüğünü gösteren grafik Şekil 5.27’de verildiği gibidir. Deney alanı tam olarak düz bir ortam olmadığı için kalıcı durumdaki titreşimler normal olarak karşılanabilir. Fakat araç hızında meydana gelen aşımın %10’un üzerine çıkması tasarlanan hız kontrolörü için uygun değildir.



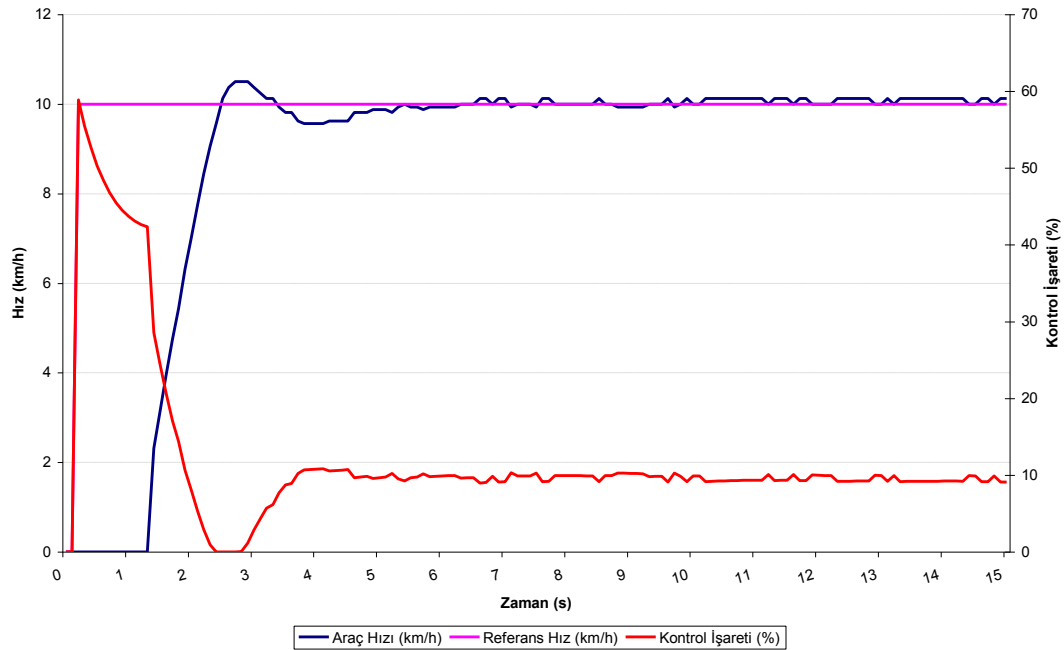
Şekil 5.27 : Kontrolör katsayıları $P=3.161$ $I=0.81955$ $D=1.3611$ $N=3.0518$.

Sistemin cevap hızını etkilemeden aşımı azaltmak için birçok deney yapılmış, bu deneyler esnasında ortaya çıkan sonuçlardan biri Şekil 5.28’de verilmiştir. Aşımı azaltmak için P katsayısı azaltılmış, sistem cevap hızının yavaşlamaması için I katsayısı arttırılmış, türev etkisi ise değiştirilmemiştir. Buna rağmen aşımın daha da arttığı ve sistem çıkışının istenilen seviyeye çok geç oturduğu görülmüştür.



Şekil 5.28 : $P=2.4828$ $I=1$ $D=1.9038$ $N=2.0479$

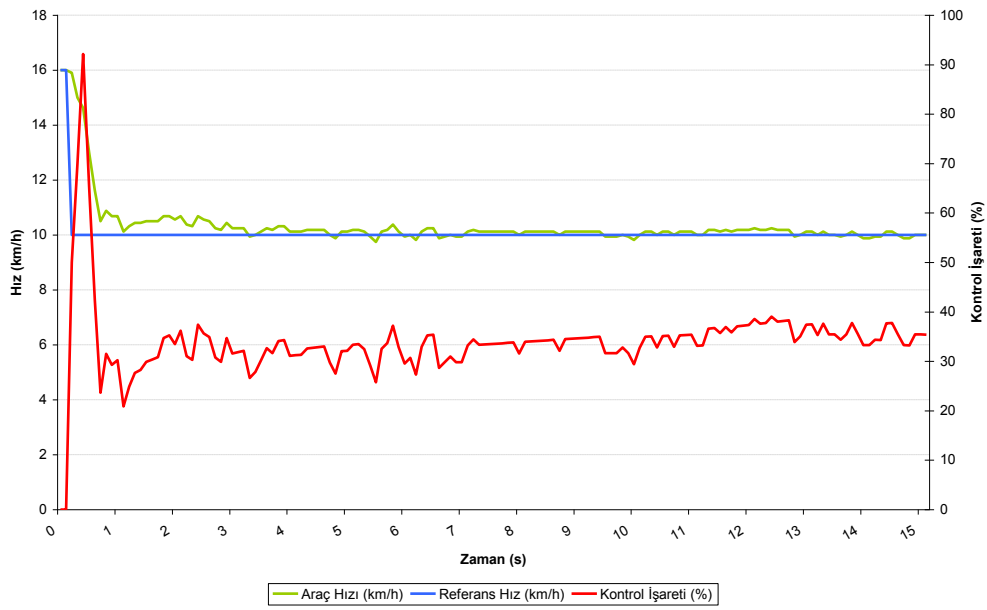
Yapılan deneyler sonucunda en iyi sistem cevabı Şekil 5.29'daki katsayılarla elde edilmiştir. Aşım %10'un altında kalmakta, kalıcı hal hatası bulunmamakta ve sistem hızlı bir şekilde cevap vermektedir.



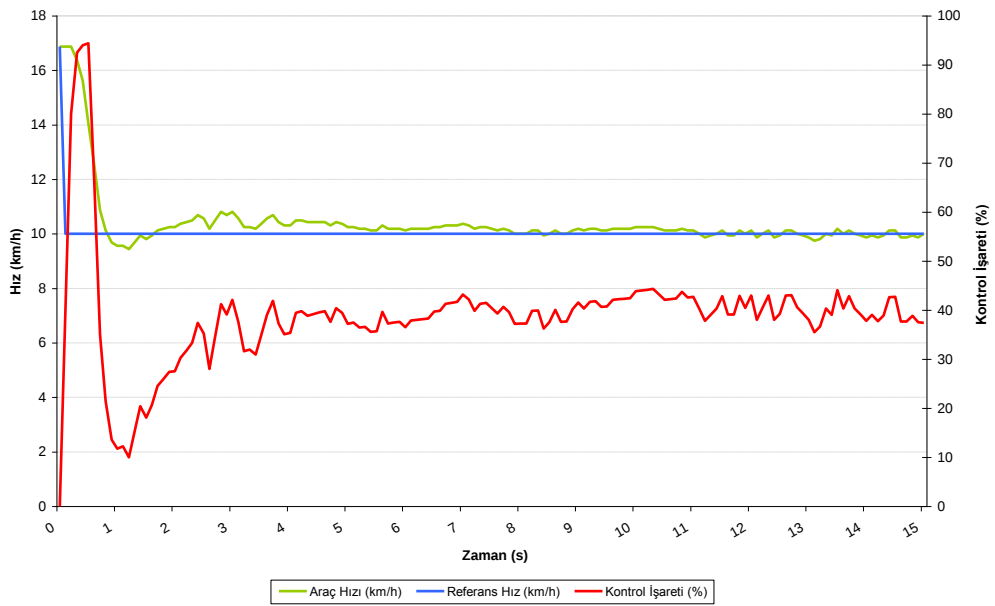
Şekil 5.29 : $P=3$ $I=0.6$ $D=2.04$ $N=1.5$.

5.3.2 Frenleme deneyleri ve iyileştirilmiş katsayılar

Simülasyon ortamında tasarlanan fren kontrolörü gerçek ortamda test edilirken ivmelenme deneylerinden farklı olarak araç öncelikle belli bir hıza getirilmiş ardından kontrolör devreye sokulmuştur. Bununla birlikte aracın kendiliğinden yavaşlamaması için bu deneyler inişli bir yolda gerçekleştirilmiştir. Ancak bu yol sabit eğimli olmadığından yapılan tüm deneylerde öngörüldüğü üzere çıkışta titreşimler gözlemlenmiştir.



Şekil 5.30 : P=16.6, I=8.4, D=1.73 ve N=1.64.

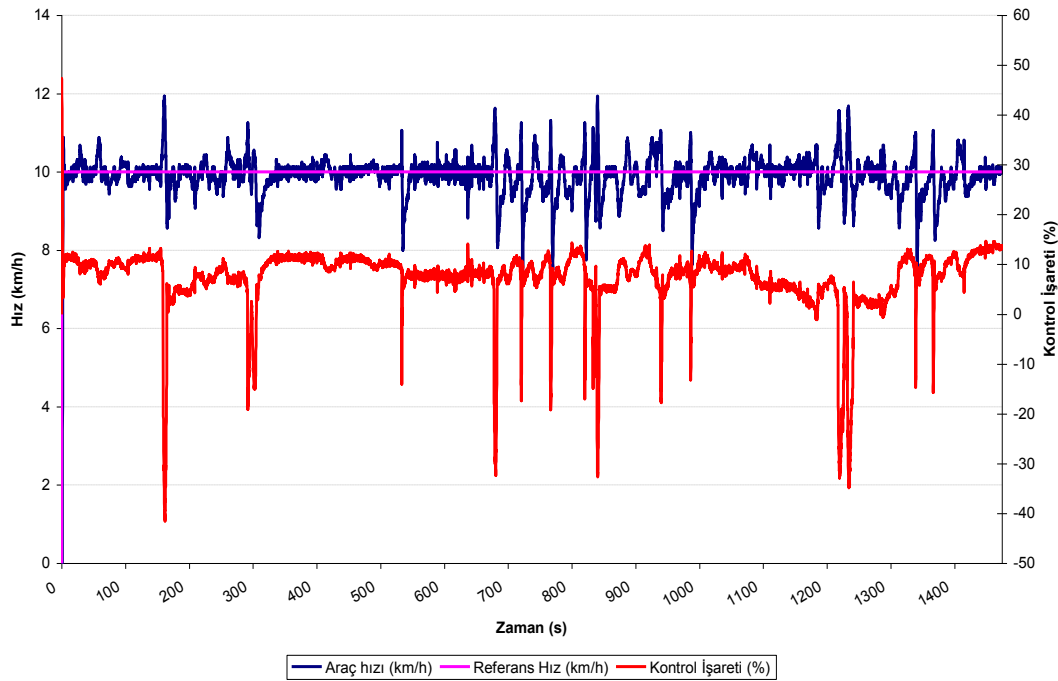


Şekil 5.31 : P=18 I=7.3 D=1 N=2.

Yapılan ilk denemede modelde hesaplanan kontrolör katsayıları kullanılmış ve Şekil 5.30'daki çıkış elde edilmiştir. Aracın hız grafiği incelendiğinde yavaşlama esnasında hiç aşım olmadığı ve sistem cevabının ufak bir aşım karşılık daha hızlı hale getirilebileceği görülmüştür. Bu sebeple kontrolör katsayılarında bazı küçük değişimler yapılmış ve en iyi sistem cevabı Şekil 5.31'deki gibi alınmıştır.

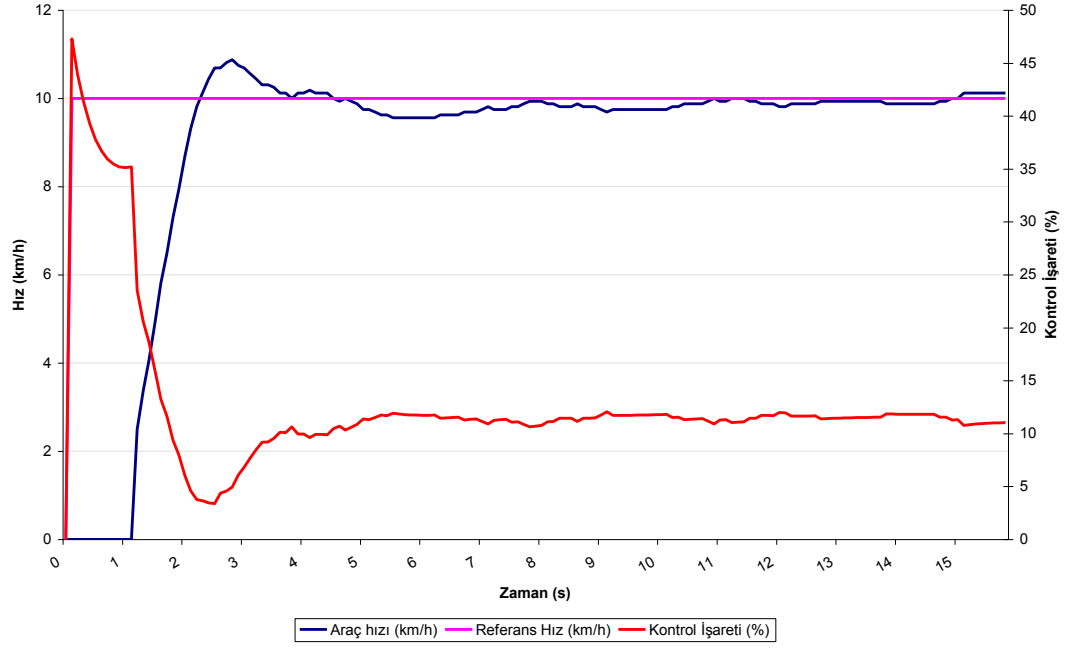
5.3.3 Hız kontrolörü deneyleri

Ayrı ayrı test edilip son hallerini alan kontrolörler bu aşamada birleştirilip hangisinin aktif olacağını belirleyen karar verme algoritmasıyla beraber tek bir yazılım içine gömülmüş ve hız kontrolör oluşturulmuştur. Bu hız kontrolörüyle yapılan yaklaşık 25 dakikalık uzun mesafeli deneyin grafiksel sonucu Şekil 5.32'de gösterildiği gibidir. Grafik incelendiğinde referans hızın 10 km/h olduğu, araç hızının ise bu değerin etrafında salındığı görülmektedir. Tüm bu süreçte araç hızının zaman zaman referans değerden uzaklaştığı, 11 km/h'nin üstüne veya 9 km/h'nin altına inerek pik yaptığı grafikten çıkartılan bir sonuçtur. Bu piklerin nedeni deney yapılan yoldaki farklı değer ve işaretteki eğimlerin kontrolör değişikliğine sebebiyet vermesidir. Modelde de değinildiği gibi karar verme merkezi ivmelenme sistemine etki eden kontrolörün mü, frenleme sistemine etki eden kontrolörün mü çalışacağına araç hızına ve istenilen referans hıza bakarak karar verir.



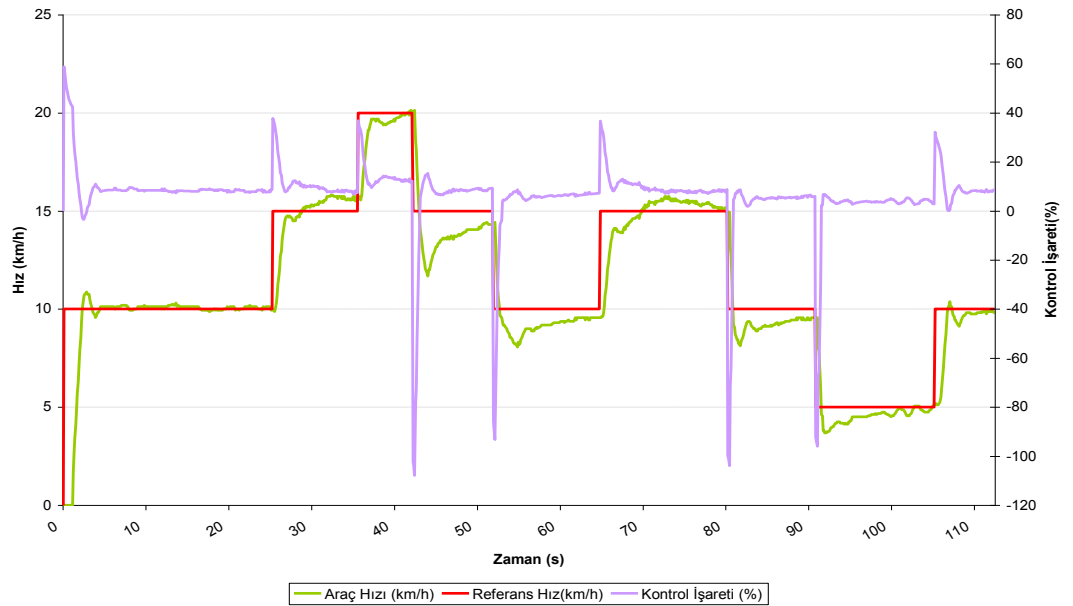
Şekil 5.32 : Hız kontrolör uzun mesafe deneyi.

Şekil 5.32'deki grafikten Şekil 5.33'deki gibi bir kesit alınırsa araç hızının referans değere oturduğu daha net bir şekilde görülmektedir.



Şekil 5.33 : Hız kontrolör uzun mesafe deneyi (16 s'lik kesit)

Yapılan deneylerden birinde de referans hız zamanla değiştirilmiş ve araç hızının bu referans değeri takibi Şekil 5.34'de olduğu gibi gözlemlenmiştir. Grafikte kontrol işaretinin pozitif olması ivmelenmeye ait kontrolörün aktif olduğunu, negatif olması ise frenlemeye ait kontrolörün aktif olduğunu gösterir.



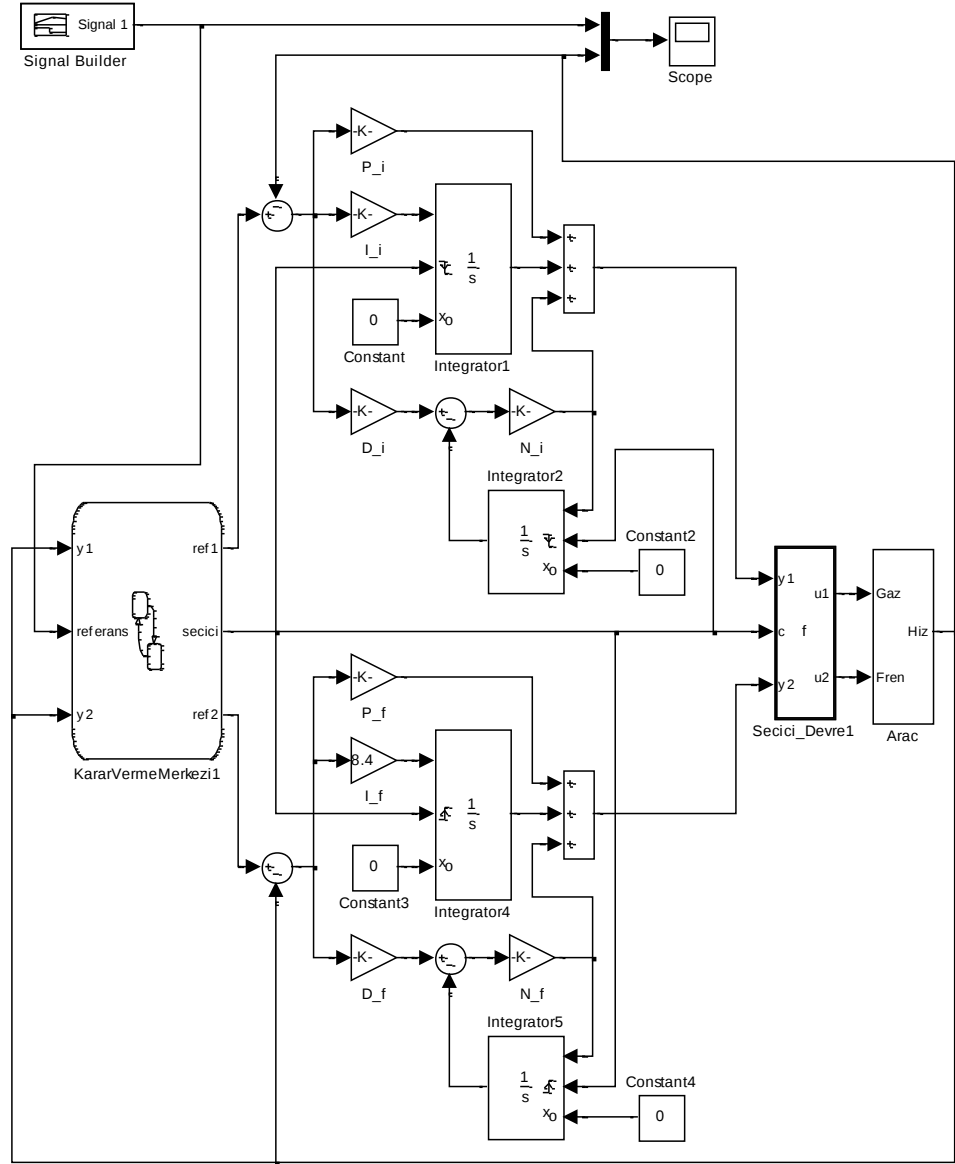
Şekil 5.34 : Farklı referans değerleriyle hız kontrolörü deneyi.

Aracın ilk hareketi dışında diğer referans değişimlerinde sistemin istenildiği gibi davranmadığı görülmüştür. Bunun sebebi aracın ilk kalkış anındaki ivmelenmeye ait transfer fonksiyonuyla, belli bir hızdan başka bir hıza giderkenki transfer fonksiyonunun aynı olmayışıdır. Bu sorun ivmelenme için iki ayrı transfer fonksiyonu elde edilerek ve bunlara ait kontrolörler tasarlanıp yazılıma eklenmesiyle çözülebilir.

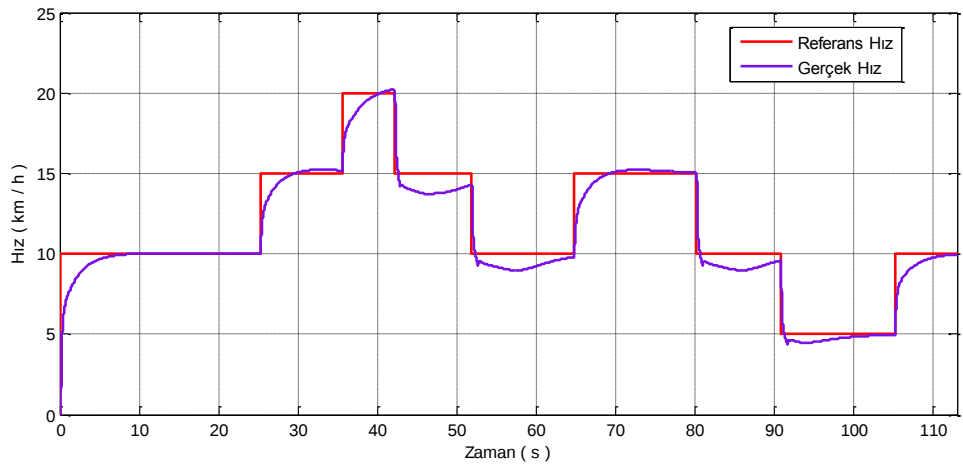
5.4 Deneysel Sonuçların Simülasyon Modeliyle Karşılaştırılması

Bu aşamaya kadar tasarlanan ve iyileştirilen kontrolörlerin bölüm 4’de ortaya çıkartılan araç modeli ile birlikte nasıl çıkış verdiğini görmek için iki yapı Şekil 5.35’de olduğu gibi küçük değişikliklerle birbirine entegre edilmiştir. Daha önceden ivmelenme ve frenleme sistemini temsil eden transfer fonksiyonları kaldırılmış, yerine bu iki sistemle beraber aracın tüm yapısını içeren model “Arac” isimli alt sistemin içine gömülmüştür. Bir başka değişiklik ise araç modelinin gaz ve fren işaretlerini iptal edilen “signal builder” elemanı yerine kontrolör çıkışlarından almasıdır.

Şekil 5.35’deki model gerçek ortamdaki gibi referans değeri değiştirilerek bir senaryo oluşturulduğunda Şekil 5.36’daki gibi çıkış vermektedir. Simülasyon sonucu oluşan bu grafik aynı girişlere sahip Şekil 5.34’deki gerçek deney sonucuyla karşılaştırıldığında benzerlik net bir biçimde görülmekte, modelin genel hatlarıyla çalıştığı ortaya çıkmaktadır. Fakat hem ivmelenme hem de frenleme için aşımelerde ve oturma zamanında farklılıklar mevcuttur. Örneğin araç ivmelenirken gerçek deneylerde bir aşım meydana gelirken, araç modeliyle yapılan simülasyonlarda hiç aşım görülmemiş ama buna karşılık sistem cevabı daha yavaş kalmıştır. Bunun nedeni araç modeli oluşturulurken tüm parametrelerin bilinmemesi ve birçoğunun tahmini olarak girilmiş olmasıdır.



Şekil 5.35 : Hız kontrolörünün araç modeliyle birleştirilmesi.



Şekil 5.36 : Farklı referans değerleriyle hız kontrolörü simülasyonu.

6. SONUÇ

Bu tez çalışması ile O³ projesi(Okan Otonom Otomobil) kapsamında prototipi geliştirilen otonom araca yönelik bir hız denetleyicisi tasarlanmış ve uygulamaya koyulmuştur. Bu çerçevede hazırlanan hız kontrolcünün başlangıçtaki hedefi aracın manuel sürüş özelliğini kaybettirmemesi, ivmelenme ve frenleme performansını mevcut duruma göre kötüleştirmemesi, araca otonom özelliği kazandıracak diğer sistemlerle uyumlu olması, vites değişimlerine müdahil olmaması, düşük hızlarda da etkin biçimde çalışabilmesi olarak belirlenmiştir. Beklenildiği gibi tez çalışması sonucunda yukarıdaki hedefleri kapsayacak yüksek başarımlı sonuçlara ulaşılmıştır. Hız denetleyicisi oluşumunun beş temel safhada şekillendiği söylenebilir. İlk olarak hem otonom araçlar hem de hız denetleyicinin yapısı ve çalışma mantığı hakkında bir literatür taraması yapılmış, sonrasında tüm süreç boyunca otomobil üzerinde yer alan ya da dışarıdan eklenmesi muhtemel aygıt veya parçalar için araştırmalar devam etmiştir. İkinci aşamada denetleyici için ihtiyaç duyulan veriler belirlenmiş, devamında bu verilerin gerçek zamanlı olarak nerden ve nasıl temin edileceği problemi çözülmüştür. Sonraki adımda ise aracın hızlanmasını ve yavaşlamasını sağlayan mevcut tertibatlar tüm ayrıntılarıyla irdelenmiştir. İnceleme sonucunda arabaya ait bu alt sistemlere hız denetleyicinin istediği biçimde etki edecek uygun mekatronik yapı tasarlanmış ve bu amaç için oluşturulan yazılım ile birlikte var olan sisteme entegre edilmiştir. İvmelenme ve frenleme sistemlerine müdahale edecek olan bu düzenek ve yazılım, araç üzerinde çok fazla bir yapısal değişikliğe neden olmayacak ve manuel sürülebilirliğini etkilemeyecek şekilde planlanmıştır. Dördüncü aşama ivmelenme ve frenleme sistemlerini denetleyecek alt denetleyicilerin tasarımı ile ilgilidir. Aracın boyuna modelinin oluşturulması ve simülasyon ortamında çalıştırılması, alt denetleyiciler için gereksinim duyulan alt sistemlerin transfer fonksiyonlarının çıkarılması, denetleyicilere ait katsayıların hesaplanması, katsayıların MatLab yardımıyla ve deneysel ortamdaki denemelerle optimize edilmesi bu evrede gerçekleşmiştir. Son safhada ise üst kontrolör oluşturulmuş, alt kontrolörler ve veri toplama yazılımı birleştirilerek hız denetleyici meydana getirmiştir. Yine bu aşamada bütünlük hale getirilen yazılım; donanım ve

araç, içinde eğim ve rüzgar gibi bozucu etkenleri de barındıran çeşitli yol testlerine tabi tutularak, hız denetleyicinin farklı senaryolar için performansı sınanmıştır.

Yapılan deneyler sonucunda hız kontrolörün çalışması esnasında elde edilen grafiklerde, referans değere karşın araç hızı incelendiğinde hem oturma zamanı, hem maksimum aşım, hem de kalıcı hal hatası bakımından çıkışın, umulan ve istenilen değerde olduğu görülmüştür. Ayrıca yol testlerinde üst kontrolcünün fren-gaz geçişlerini simülasyon ortamındaki gibi gerçekleştirmesi, gerçek sistemin imkanlar dahilinde oluşturulan modelle örtüştüğünü de göstermektedir. İstenildiği gibi işleyen bu sistemin var olan yapı bozulmadan ve sürüş özelliği yitirilmeden araca adapte edilmesi bir avantaj olarak sayılsa da, proje kapsamındaki otonom otomobil için sistemin asıl getirisi, her türlü şartta hızın karar verme merkezi tarafından elektronik işaretlerle kontrol edilebilir hale gelmesidir.

Tez kapsamında ortaya konan model, aracın tüm viteslerini ve hemen hemen her türlü durumu etraflıca kapsayacak şekilde oluşturulmuş olsa da, içinde yer alan bazı parametrelerin gerçek ve kesin değerlerine erişilemediğinden modelin optimum olduğunu söylemek güçtür. Bu yüzden alt denetleyiciler oluşturulurken gerekli olan transfer fonksiyonlarını elde etmek için matematiksel modelin veya simülasyon modelinin yerine deneysel olarak bulunan ve aracın dinamik davranışını içeren veriler kullanılmıştır. Fakat deney alanının yüksek hızlı denemelere müsaade etmemesinden dolayı yalnızca 1. vites için transfer fonksiyonları çıkartılabilmiş ve hız kontrolör de bu çerçevede hazırlanmıştır. Çalışmada karşı karşıya gelinen bir diğer zorluk ise hem otonom araçta hem de özelde hız denetleyicide yer alan farklı yapıların birbiri ile donanımsal ve yazılımsal açıdan entegrasyonu olmuştur. Kullanılan bu ekipmanların kendine has editör programlara ve yazılımlara sahip olması, aralarındaki haberleşmelerde protokol bakımından bazen birbiri ile uyum gösterememesi gibi sebepler bu güçlükler örnek olarak sayılabilir.

Şu an için hız denetleyicinin her ne kadar bazı küçük eksiklikleri bulunsa da, denetleyicinin modüler olması avantajı sayesinde gelecekte bu çalışmanın bir devamı olarak ufak modifikasyonlarla kolaylıkla bu açıklar giderilebilir. Tez kapsamında temeli oluşturulan uygulama birçok yönüyle geliştirmeye açıktır. Başlangıç olarak her vites kademesine ayrı bir transfer fonksiyonu çıkartılarak otonom otomobilin yakın zamanda ihtiyaç duyacağı yüksek hızlara ulaşılabilir. Araç ivmelenme açısından kalkış esnasında veya vitesin değiştiği anda, sistemin oturduğu ana göre

farklı davranış göstermektedir. Belki alt denetleyicilerin mükemmelleşmesi için aynı viteste iki farklı katsayı grubu kullanılması da ileride konu edinilebilir.

Tasarlanan hız denetleyicinin çalışma prensibi referans hıza en kısa sürede ve en küçük aşım ile ulaşma, sistemin limitlerini görme-zorlama üzerine kurulmuştur. Fakat bu özellik yanında arzu edilmeyen bazı durumları da türetebilir. Örneğin bu sistemin, zaman zaman çok ani duruşlar veya çok yüksek ivme ile hareketlenmeler sebebiyle sürüş konforunu olumsuz etkilediği söylenebilir. Yine aynı sebeple yakıt tüketimi ve emisyon değerleri düşünüldüğünde bir başka dezavantaj da olduğu ortaya konulabilir. Fakat sistem girişine direkt referans değer göndermek yerine, gerçek değerden referans değere giden kademeli bir hız profili uygulanırsa belirtilen problemler kolaylıkla aşılabılır. Ayrıca çalışma daha da derinleştirilip otonom otomobilde bulunan LIDAR, RADAR, GPS, IMU vb. gibi sistemlerden gelen bilgilerin tümü (ileride yokuş veya viraj olduğu, cm hassasiyetle öndeki araç ile mesafesi) kullanılabilir duruma getirildiğinde gaz-fren dengesi mükemmel halini alacak, yakıt tasarrufu en üst seviyelere varacaktır.

Sonuç olarak yapılan bu çalışmada konulan hedefler çerçevesinde beklenen neticelere iyi bir şekilde ulaşılmış, sistem uygulamada da işler hale getirilmiştir. Tasarlanan yapının ileriki süreçlerde geliştirilebileceği ve genişletilebileceği yönleri de olduğu düşünülerek, sistem modüler ve bu tarz bir çalışmaya temel olabilecek biçimde oluşturulmuştur.

KAYNAKLAR

- [1] **OICA**, Organisation Internationale des Constructeurs d'Automobiles.
- [2] **TUIK**, Türkiye İstatistik Kurumu, Ulaştırma İstatistikleri.
- [3] **CDIAC**, (2008). The United States Department of Energy's Carbon Dioxide Information Analysis Center for the United Nations.
- [4] **Ozguner, U., Acatman, T. ve Redmil**, (2011). Autonomous Ground Vehicles.
- [5] **Schmidhuber, J.** (2005) Highlights of Robot Car History, Alındığı tarih: 18.12.2011, adres:<http://www.idsia.ch/~juergen/robotcars.html>.
- [6] **Forret, A., Konca, M.** (2007). Autonomous Cars and Society.
- [7] **Tuncay R. N., Uçarol H., Ustun O.**, (2011). Elektrikli Kara Yolu Taşıtlarında Teknoloji Ufuk Turu, Çağrılı Bildiri, EUSIS 2011, Elektrikli Ulaşım Sistemleri Sempozyumu, Bursa-Eskişehir.
- [8] **Hedrick, J.K., McMahon, D., Narendra, V. and Swaroop, D.** (1991). Longitudinal Vehicle Controller Design For IVHS Systems, *Proceedings of the 1991 American Control Conference*, Boston, Massachusetts, June 1991, pp. 3107- 3112.
- [9] **Haskara, İ., Hatipoglu, C. and Özgüner, Ü.** (1997). Combined Decentralized Longitudinal And Lateral Controller Design For Truck Convoys, *Intelligent Transportation System, ITSC 97. IEEE Conference on* 9-12
- [10] **Leen, G., Hefferman, D.** (2002). Expanding Automotive Electronic Systems IEEE Computer(2002) 88-93.
- [11] **Url-1** < <http://www.quatech.com/support/comm-over-parallel.php> >, alındığı tarih: 18.12.2011.
- [12] **Paret, D.** (2007). Multiplied Networks for Embebed Systems CAN, LIN, Flexray, Safe-by-wire, Wiley, 25-351.
- [13] **TOFAŞ Türk Otomobil Fabrikası**, (2011). adres: <http://www.tofas.com.tr/>
- [14] **Younqin, Z., Xudong, W., Meilan, Z.** (2006). The Research and Realization For Passenger Car CAN Bus, IEEE The First International Forum on Strategic Technology 2006.
- [15] **Koç, Z.** (2011). Bir Otomobil İçin Araç Mekatroniği Uygulaması, *Yüksek Lisans Tezi* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [16] **Url-2** < <http://www.barissamanci.net/Makale/15/can-bus-nedir-can-protokolu-incelemesi> >, alındığı tarih: 18.12.2011.
- [17] **Robert Bosch GmbH, Stuttgart**, (1991). CAN Specification Version 2.0
- [18] **Coşkun, U.** (2008). Controller Area Network ve Uygulaması, *Yüksek Lisans Tezi*, GYTE, Mühendislik ve Fen Bilimleri Enstitüsü, Gebze.

- [19] **Yabanova, İ., Taşkın, S., Ekiz, H., Çimen H.** (2010). Denetleyici Alan Ağı Üzerinden Mekatronik Bir Sistemin Kontrolü, *Makine Teknolojileri Elektronik Dergisi* , 2, 63-72.
- [20] **Santur, Y.** (2006). Kontrol Alan Ağı Protokolü Tabanlı Bir Endüstriyel Ağı Kurulması ve Yönetilmesi, *Yüksek Lisans Tezi*, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.
- [21] **Ersoy, S., Taşkın, S.** Otomotivde CAN-Bus Teknolojisi ve Otomotiv Mekatroniğinin Geleceği, Marmara Üniversitesi, Teknik Eğitim Fakültesi, İstanbul.
- [22] **Url-3** < <http://www.kvaser.com/>>, alındığı tarih: 21.01.2012
- [23] **Url-4** < <http://www.howstuffworks.com/>>, alındığı tarih: 01.02.2012
- [24] **Url-5** < <http://www.otomobil.net/>>, alındığı tarih: 24.03.2012
- [25] **MEGEP** (2005). Fren Sistemleri Modülü, Milli Eğitim Bakanlığı.
- [26] **Göktan, A.G.** Taşıtlarda Aktarma Organları Ders Notları.
- [27] **Url-6**< <http://www.tegger.com/hondafaq/mastercylinderreplace/static-schematic.gif> >, alındığı tarih: 18.12.2011.
- [28] **Göktan, A. G., Güney, A., Ereke, M.** (1995). Taşıt Frenleri, İTÜ Makine Fakültesi, Otomotiv Anabilim Dalı.
- [29] **Url-7** < <http://www.linmot.com/>>, alındığı tarih: 03.12.2011
- [30] **Url-8** < <http://www.microchip.com/>>, alındığı tarih: 26.04.2012
- [31] **Url-9** < <http://www.alldatasheet.com/>>, alındığı tarih: 01.02.2012
- [32] **Url-10** < <http://www.st.com/>>, alındığı tarih: 12.03.2012
- [33] **Url-11** < <http://www.st.com/stm32-discovery/> >, alındığı tarih: 18.12.2011.
- [34] **Özdalyan, B.** Motorlu Taşıtlar Mekaniği Ders Notları.

EKLER

EK A : PIC 16F877 Mikrokontrolöründeki Kod

EK B : Dizel Motor Özellikleri

EK C : Matlab/Simulink Fonksiyon Kodları

EK D : Bilgisayar Ortamındaki Kontrol Algoritmasına Ait Yazılım

EK A : PIC 16F877 Mikrokontrolöründeki Kod

Include "modedefs.bas"

```
PORTA=0
PORTB=0
PORTC=0
PORTD=0
PORTE=0
```

```
DEFINE LCD_DREG PORTB
  DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTD
  DEFINE LCD_RSBIT 4
DEFINE LCD_EREG PORTD
  DEFINE LCD_EBIT 6
  DEFINE LCD_BITS 4
  DEFINE LCD_LINES 2
  DEFINE OSC 4
```

```
PAUSE 500
define ADC_BITS 10
DEFINE ADC_SAMPLEUS 50
ADCON1=%10000010
```

```
TRISA=%11111111
TRISB=%00001111
TRISC=%11000000
TRISD=%00000000
TRISE=%11111111
PAUSE 500
```

```
BUTON1 VAR PORTB.3
BUTON2 VAR PORTB.2
BUTON3 VAR PORTB.1
BUTON4 VAR PORTB.0
KONTROLAD1 VAR PORTD.0
BUTONACIL VAR PORTD.1
BUZER VAR PORTD.2
KONTROLBC1 VAR PORTD.3
LCDISIK VAR PORTD.5
LCDISIKACIL VAR PORTD.7
KIRMIZI VAR PORTC.0
BEYAZ VAR PORTC.1
MAVI VAR PORTC.2
SARI VAR PORTC.3
KONTROLBC2 VAR PORTC.4
KONTROLAD2 VAR PORTC.5
TX VAR PORTC.6
RX VAR PORTC.7
```

```

BESLEME VAR WORD
i var byte
k VAR BYTE
GAZORANI VAR WORD
DURUM VAR BYTE
GIRISVOLT VAR WORD
GAZTAM VAR WORD
GAZYARIM VAR WORD
FRENACIL VAR WORD
FRENKONUM VAR WORD
FRENORANI VAR WORD
ACILORANI VAR WORD

```

```

'*****
'*****

```

```

BAS:

```

```

    K=0
    KONTROLAD1=0
    KONTROLBC1=0
    KONTROLBC2=0
    KONTROLAD2=1

```

```

    LCDISIK=1
    PAUSE 200

```

```

    LCDOUT $FE,1
        LCDOUT "HOSGELDINIZ.."
        PAUSE 200

```

```

    lcdout $fe,$c0
    FOR i=1 to 16
        LCDOUT ">"
        IF (i=1) THEN
            BEYAZ=1
        ENDIF
        IF (i=5) THEN
            BEYAZ=0
            MAVI=1
        ENDIF
        IF (i=9) THEN
            MAVI=0
            SARI=1
        ENDIF
        IF (i=13) THEN
            SARI=0
            KIRMIZI=1
        ENDIF
        pause 120
    next
    SARI=1

```

```

MAVI=1
BEYAZ=1
    PAUSE 400
GOSUB SES
GOSUB SES
BAS2:
    SARI=0
    MAVI=0
    BEYAZ=0
    KIRMIZI=0
        GOSUB ANALOGOKU
        IF ((DURUM=3) OR (DURUM=4)) THEN
            LCDOUT $FE,1
            LCDOUT "GERILIM HATASI"
            lcdout $fe,$c0
            lcdout "BESLEME ",DEC(BESLEME/10),".",DEC (BESLEME//10)
            PAUSE 1500
            GOTO BAS2
        ELSE
            LCDOUT $FE,1
            LCDOUT "BASLAMAK ICIN"
            lcdout $fe,$c0
            lcdout "HAZIR -> B4"
BAS3:
    PAUSE 100
    GOSUB ANALOGOKU
        IF ((DURUM=3) OR (DURUM=4)) THEN
            GOTO BAS2
        ENDIF
        IF (BUTON4=0) THEN
            GOTO BAS3
        ENDIF
    ENDIF
GOSUB ANALOGOKU
    if (DURUM<3) then
        goto A0
    else
        Goto ACIL
    Endif

'*****
'*****

A0:
    KONTROLAD1=0
    KONTROLBC1=0
    KONTROLBC2=0
    KONTROLAD2=1
    MAVI=0
    K=0
        LCDOUT $FE,1

```

```

        LCDOUT "KNTRL:PEDAL"
A1:
    k=k+1
    if (K=1) then
        lcdout $fe,$c0
        lcdout "B1 -> DEGISTIR "
    endif
    if (k=21)then
        lcdout $fe,$c0
        lcdout "GAZ ORANI -> %",DEC2 GAZORANI
    endif
    if (k=41)then
        GOSUB ANALOGOKU
        IF (DURUM=0) THEN
            K=0
            SARI=0
        ENDIF
        IF (DURUM=1) THEN
            lcdout $fe,$c0
            lcdout "ALCAK GERILIM "
        ENDIF
        IF (DURUM=2) THEN
            lcdout $fe,$c0
            lcdout "YUKSEK GERILIM "
        ENDIF
    endif
    IF ((DURUM<>0)aND(k/3=1)) THEN
        toggle SARI
    ENDIF
    IF (K>60) THEN
        GOSUB SES
        K=0
    ENDIF
    IF (DURUM>2) THEN
        GOSUB SES
        GOTO ACIL
    ENDIF

BEYAZ=1
FOR i=1 TO 120
    if (BUTON1=1) THEN
        GOTO A2
    ENDIF
    if (BUTONACIL=1) THEN
        DURUM=10
        GOSUB SES
        GOTO ACIL
    ENDIF
NEXT

```

```

BEYAZ=0
FOR i=1 TO 255
  if (BUTON1=1) THEN
    GOTO A2
  ENDIF
  if (BUTONACIL=1) THEN
    DURUM=10
    GOTO ACIL
    GOSUB SES
  ENDIF
  pause 1
NEXT
GOTO A1

```

```

A2:
GOSUB SES
LCDOUT $FE,1
  LCDOUT "B2:BILGISAYAR"
  lcdout $fe,$c0
  lcdout "B3:POT B4:IPTAL"

```

```

A3:
FOR i=1 TO 255
  IF (BUTON2=1) THEN
    GOSUB SES
    GOTO B0
  ENDIF
  IF (BUTON3=1) THEN
    GOSUB SES
    GOTO C0
  ENDIF
  IF (BUTON4=1) THEN
    GOSUB SES
    GOTO A0
  ENDIF
  PAUSE 1
  if (BUTONACIL=1) THEN
    DURUM=10
    GOSUB SES
    GOTO ACIL
  ENDIF
NEXT
toggle BEYAZ
GOTO A3

```

```

'*****
'*****

```

```

B0:
KONTROLAD2=0
KONTROLBC2=1
KONTROLBC1=0

```

```

KONTROLAD1=1
BEYAZ=0
K=0
    LCDOUT $FE,1
    LCDOUT "KNTRL:BILGISAYAR"

```

```

B1:
k=k+1
    if (K=1) then
        lcdout $fe,$c0
        lcdout "B1 -> DEGISTIR "
    endif
    if (k=21)then
        lcdout $fe,$c0
        lcdout "GAZ ORANI -> %",DEC2 GAZORANI
    endif
    if (k=41)then
        GOSUB ANALOGOKU
        IF (DURUM=0) THEN
            K=0
            SARI=0
        ENDIF
        IF (DURUM=1) THEN
            lcdout $fe,$c0
            lcdout "ALCAK GERILIM "
        ENDIF
        IF (DURUM=2) THEN
            lcdout $fe,$c0
            lcdout "YUKSEK GERILIM "
        ENDIF
    endif
    IF ((DURUM<>0)aND(k//3=1)) THEN
        toggle SARI
    ENDIF
    IF (K>60) THEN
        GOSUB SES
        K=0
    ENDIF
    IF (DURUM>2) THEN
        GOSUB SES
        GOTO ACIL
    ENDIF

```

```

MAVI=1
FOR i=1 TO 120
    if (BUTON1=1) THEN
        GOTO B2
    ENDIF
    if (BUTONACIL=1) THEN
        DURUM=10
    endif

```

```

        GOSUB SES
        GOTO ACIL
    ENDIF
NEXT

MAVI=0
FOR i=1 TO 255
    if (BUTON1=1) THEN
        GOTO B2
    ENDIF
    if (BUTONACIL=1) THEN
        DURUM=10
        GOTO ACIL
        GOSUB SES
    ENDIF
    pause 1
NEXT
GOTO B1

B2:
    GOSUB SES
    LCDOUT $FE,1
    LCDOUT "B2:PEDAL"
    lcdout $fe,$c0
    lcdout "B3:POT B4:IPTAL"

B3:
    FOR i=1 TO 255
        IF (BUTON2=1) THEN
            GOSUB SES
            GOTO A0
        ENDIF
        IF (BUTON3=1) THEN
            GOSUB SES
            GOTO C0
        ENDIF
        IF (BUTON4=1) THEN
            GOSUB SES
            GOTO B0
        ENDIF
    PAUSE 1
    if (BUTONACIL=1) THEN
        DURUM=10
        GOSUB SES
        GOTO ACIL
    ENDIF
NEXT
toggle MAVI
GOTO B3

```

```
C0:
  KONTROLAD2=0
  KONTROLBC2=1
  KONTROLAD1=0
  KONTROLBC1=1
  BEYAZ=0
  K=0
  LCDOUT $FE,1
    LCDOUT "KNTRL:POT"
C1:
  k=k+1
  GOSUB ANALOGOKU
  if (K=100) then
    lcdout $fe,$c0
    lcdout "B1 -> DEGISTIR  "
  endif
  if (k//2=0) then
    lcdout $fe,$c0
    IF (GAZORANI<100) THEN
      lcdout "GAZ ORANI-> %",DEC2 GAZORANI," "
    ELSE
      lcdout "GAZ ORANI-> %",DEC3 GAZORANI
    ENDIF
  endif
  if (k=41)then
    GOSUB ANALOGOKU
    IF (DURUM=0) THEN
      K=0
      SARI=0
    ENDIF
    IF (DURUM=1) THEN
      lcdout $fe,$c0
      lcdout "ALCAK GERILIM  "
    ENDIF
    IF (DURUM=2) THEN
      lcdout $fe,$c0
      lcdout "YUKSEK GERILIM  "
    ENDIF
  endif
  IF ((DURUM<>0)aND(k//3=1)) THEN
    toggle SARI
  ENDIF
  IF (K>60) THEN
    GOSUB SES
    K=0
  ENDIF
  IF (DURUM>2) THEN
    GOSUB SES
```

```

        GOTO ACIL
    ENDIF

MAVI=1
    FOR i=1 TO 255
        IF (BUTON1=1)THEN
            GOTO C2
        ENDIF
        if (BUTONACIL=1) THEN
            DURUM=10
            GOSUB SES
            GOTO ACIL
        ENDIF
    NEXT
MAVI=0
    FOR i=1 TO 255
        IF (BUTON1=1)THEN
            GOTO C2
        ENDIF
        if (BUTONACIL=1) THEN
            DURUM=10
            GOSUB SES
            GOTO ACIL
        ENDIF
        pauseus 200
    NEXT
MAVI=1
    FOR i=1 TO 255
        IF (BUTON1=1)THEN
            GOTO C2
        ENDIF
        if (BUTONACIL=1) THEN
            DURUM=10
            GOSUB SES
            GOTO ACIL
        ENDIF
    NEXT
MAVI=0
    FOR i=1 TO 255
        IF (BUTON1=1)THEN
            GOTO C2
        ENDIF
        if (BUTONACIL=1) THEN
            DURUM=10
            GOSUB SES
            GOTO ACIL
        ENDIF
        pause 1
    NEXT
BEYAZ=1

```

```

FOR i=1 TO 255
  IF (BUTON1=1)THEN
    GOTO C2
  ENDIF
  if (BUTONACIL=1) THEN
    DURUM=10
    GOSUB SES
    GOTO ACIL
  ENDIF
NEXT
BEYAZ=0
FOR i=1 TO 255
  IF (BUTON1=1)THEN
    GOTO C2
  ENDIF
  if (BUTONACIL=1) THEN
    DURUM=10
    GOSUB SES
    GOTO ACIL
  ENDIF
  pauseus 200
NEXT
BEYAZ=1
FOR i=1 TO 255
  IF (BUTON1=1)THEN
    GOTO C2
  ENDIF
  if (BUTONACIL=1) THEN
    DURUM=10
    GOSUB SES
    GOTO ACIL
  ENDIF
NEXT
BEYAZ=0
FOR i=1 TO 255
  IF (BUTON1=1)THEN
    GOTO C2
  ENDIF
  if (BUTONACIL=1) THEN
    DURUM=10
    GOSUB SES
    GOTO ACIL
  ENDIF
  pause 1
NEXT
GOTO C1

C2:
GOSUB SES
LCDOUT $FE,1

```

```

        LCDOUT "B2:BILGISAYAR"
        lcdout $fe,$c0
        lcdout "B3:PEDAL B4:IPTAL"
        BEYAZ=1
MAVI=0
C3:
    FOR i=1 TO 255
    IF (BUTON2=1) THEN
        GOSUB SES
        GOTO B0
    ENDIF
    IF (BUTON3=1)THEN
        GOSUB SES
        GOTO A0
    ENDIF
    IF (BUTON4=1)THEN
        GOSUB SES
        GOTO C0
    ENDIF
    PAUSE 1
    if (BUTONACIL=1) THEN
        DURUM=10
        GOSUB SES
        GOTO ACIL
    ENDIF
    NEXT
    toggle MAVI
    toggle BEYAZ
    GOTO C3

'*****
'*****

ANALOGOKU:
    adcin 0,GIRISVOLT
    adcin 1,GAZTAM
    adcin 2,FRENKONUM
    adcin 3,GAZYARIM
    adcin 4,FRENACIL

    DURUM=0
    GAZORANI=((GAZTAM-205)*10)/71
    FRENORANI=(FRENKONUM*10)/82
    BESLEME=(GIRISVOLT*16)/100
    ACILORANI=(FRENACIL*100)/593

    If ((GIRISVOLT<703) AND (GIRISVOLT>640)) THEN
        DURUM=1
    ENDIF
    If ((GIRISVOLT<961) AND (GIRISVOLT>832)) THEN

```

```

        DURUM=2
    ENDIF
    if (GIRISVOLT<640) THEN
        DURUM=3
    ENDIF
    if (GIRISVOLT>961) THEN
        DURUM=4
    ENDIF
    if (((gaztam/2)-GAZYARIM)>4) OR ((GAZYARIM-(gaztam/2))>4)) THEN
        'DURUM=5
    ENDIF
    if (gaztam>924) THEN
        'DURUM=6
    ENDIF
    if (gaztam<456) THEN
        'DURUM=7
    ENDIF
    if (FRENKONUM>804) THEN
        'DURUM=8
    ENDIF
    RETURN

```

```

*****
*****

```

ACIL:

```

    KONTROLBC2=0
    KONTROLAD2=1
    KONTROLAD1=0
    KONTROLBC1=1
    KIRMIZI=1
    GOSUB SES
    LCDISIK=0
    FOR i=0 TO 255
        IF (BUTON3=1)AND (BUTON4=1)THEN
            GOTO BAS
        ENDIF
        PAUSE 2
    NEXT

    SELECT CASE DURUM

    CASE 3
        LCDOUT $FE,1
        LCDOUT "DUSUK GERILIM"
        lcdout $fe,$c0
        lcdout "BESLEME  ",DEC(BESLEME/10),".",DEC (BESLEME//10),"V"

    CASE 4
        LCDOUT $FE,1

```

```

        LCDOUT "YUKSEK GERILIM"
        lcdout $fe,$c0
        lcdout "BESLEME  ",DEC(BESLEME/10),".",DEC (BESLEME//10),"V"

CASE 5
    LCDOUT $FE,1
    LCDOUT "GAZ PEDALI"
    lcdout $fe,$c0
    lcdout "ORAN HATASI"

CASE 6
    LCDOUT $FE,1
    LCDOUT "GAZ PEDALI"
    lcdout $fe,$c0
    lcdout "YUKSEK GERILIM "

CASE 7
    LCDOUT $FE,1
    LCDOUT "GAZ PEDALI"
    lcdout $fe,$c0
    lcdout "ALCAK GERILIM "

CASE 8
    LCDOUT $FE,1
    LCDOUT "FREN PEDALI"
    lcdout $fe,$c0
    lcdout "YUKSEK GERILIM "
CASE ELSE
    GOTO BAS

END SELECT

LCDISIK=1
GOSUB ANALOGOKU
FOR i=0 TO 255
    IF ((BUTON3=1)AND (BUTON4=1)) THEN
        GOTO BAS
    ENDIF
    PAUSE 2
NEXT
goto ACIL

```

EK B : Dizel Motor Özellikleri

1.3 Multijet 16v 90 Beygir

Silindir sayısı, konumu 4, tek hat üzerinde sıralı, ön tarafta çapraz

Supap sayısı 4

Çap x strok (mm) 69,6 x 82,0

Silindir hacmi (cm³) 1248

Azami güç: 90 HP

Devir / dakika 4000

Azami tork: 200 Nm

Devir / dakika 1750

EK C : Matlab/Simulink Fonksiyon Kodları

Sifirlama

```
function out = fcn(in)
%#eml
a=in;
b=0;
if (1<a)
    b=a;
end
out = b;
```

Vites_Degistir

```
function [v_artir,v_azalt] = fcn(devir,vites)
%#eml
artir=0;
azalt=0;
if ((devir>3600)&&(vites<5))%% 43 devir/sn = 2600 devir/dakika
    artir=1;
    azalt=0;
end
if ((devir<1000)&&(vites>1)) %% 16 devir/sn = 2600 devir/dakika
    artir=0;
    azalt=1;
end
v_artir=artir;
v_azalt=azalt;
```

Sayici_Resetle

```
function [reset1,vites,reset2] = fcn(u1,u2)
%#eml
b=0;
c=0;
a=(u1-u2);
if (u1==100)
    b=1;
end
if (u2==100)
    c=1;
end
reset1=b;
reset2=c;
vites=mod(a,6);
```

Kayma_Hesabi

```
function f_out = fcn(kuvvet,ivme)
%#eml
a=kuvvet;
b=0;
fz1=(751.9)*(9.81);
mu=1;
fmax=fz1*mu;
if (fmax<abs(a))
    if(a>0)
        a=fmax;
    else
        a=(-1)*fmax;
    end
end
if ((kuvvet>0)||(kuvvet<0))
    b=0.01+abs(ivme/100);
end
f_out = a*(1-b);
```

Yuvarlanma_Direnci

```
function ft = fcn(V_teker,kuvvet)
%#eml
p=2.2; %% Lastik Basinci (yaklasik 2.2 bar)
l1=1.019;
l2=1.586;
m=1235;
g=9.80665;
a=kuvvet;
b=0;
fz1=(m*l2*g)/(l1+l2);
fz2=(m*l2*g)/(l1+l2);
v=V_teker*3.6;
c=0.005+((0.01+(0.0095*(v/100)*(v/100))))/p;
fy1 = fz1*c;
fy2 = fz2*c;
fy=fy1+fy2;
if (V_teker>0) %% arac ileri gidiyorsa
    b=a-fy;
elseif (V_teker<0) %% arac geri gidiyorsa
    b=a+fy;
else %% arac duruyorsa
    if(kuvvet>fy)
        b=a-fy;
    elseif (abs(kuvvet)>fy)
```

```

        b=a+fy;
    else
        b=0;
    end
end
ft=b;

```

Yon_2

```

function isaret = fcn(fren,hiz, abs)
%#eml
a=fren;
b=hiz;
c=abs;

if (b>0.1)
    a=c;
end
if (b<(-0.1))
    a=c;
end

if(hiz>0)
    a=a*(-1);
end
isaret = a;

```

Yon2

```

function y = fcn(u1,u2)
%#eml
if(u1<0)
    u2=u2*(-1);
end
y = u2;

```

EK D : Bilgisayar Ortamındaki Kontrol Algoritmasına Ait Yazılım

```
// Kütüphaneler
#include <stdlib.h>
#include <stdio.h>
#include <memory.h>
#include <windows.h>
#include <conio.h>
#include <time.h>
#include "canlib.h"

union datam
{
    short int aci;
    unsigned char veri[2];
}acim;

#define idsi 1440

// GAZ KATSAYILARI
#define K_P 3.0
#define K_I 0.6
#define K_D 2.04
#define T 0.1
#define N_D 1.5
// FREN KATSAYILARI
#define K_Pf 18
#define K_If 7.3
#define K_Df 1
#define N_Df 2
#ifdef __BORLANDC__
#pragma argsused
#endif

typedef struct PID_DATA{
    double A_s;
    double B_s;
    double C_s;
    double D_s;
    double E_s;
    double u_ke_1;
    double u_ke_2;
    double e_ke_1;
    double e_ke_2;
} pidData_t;

struct PID_DATA pidData;
struct PID_DATA pidDataf;
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;
long time2=0;

void pid_Init(double p, double i, double d, double t, double n, struct PID_DATA
*pid);
double pid_Controller(double setPoint, double processValue, struct PID_DATA
*pid_st);
////////////////////////////////////
void pid_Init(double p, double i, double d,double t, double n, struct PID_DATA
*pid)
{
    pid->A_s = (4*p + 2*p*n*t + 4*n*d + 2*t*i + n*i*t*t)/(4 + 2*n*t);
    pid->B_s = (2*i*t*t*n -8*p -8*n*d)/(4+2*n*t);
    pid->C_s = (4*p -2*p*n*t + 4*n*d -2*i*t + t*t*i*n)/(4+2*n*t);
    pid->D_s = (-8)/(4+2*n*t);
    pid->E_s = (4-2*n*t)/(4+2*n*t);

    pid->u_ke_1 = 0;
    pid->u_ke_2 = 0;
    pid->e_ke_1 = 0;
    pid->e_ke_2 = 0;
}
////////////////////////////////////
double pid_Controller(double setPoint, double processValue, struct PID_DATA
*pid_st)
{
    double e_k, u_k;

    e_k = setPoint - processValue;
    u_k = ( pid_st->A_s * e_k ) + ( pid_st->B_s * pid_st->e_ke_1 ) + ( pid_st-
>C_s * pid_st->e_ke_2 ) - ( pid_st->D_s * pid_st->u_ke_1) - ( pid_st->E_s * pid_st-
>u_ke_2 );

    pid_st->e_ke_2 = pid_st->e_ke_1;
    pid_st->e_ke_1 = e_k;

    pid_st->u_ke_2 = pid_st->u_ke_1;
    pid_st->u_ke_1 = u_k;
    return((double)u_k);
}
////////////////////////////////////
void Init(void)
{
    pid_Init(K_P, K_I, K_D, T, N_D, &pidData);
}

```

```

    pid_Init(K_Pf, K_If, K_Df, T, N_Df, &pidDataf);
}
////////////////////////////////////
void Set_Input(double inputValue)
{
    ;
}
////////////////////////////////////
void Usage(int argc, char* argv[])
{
    printf("\nCANLIB Dump Program\n");
    printf("(Part of the CANLIB SDK from KVASER AB -  

http://www.kvaser.se)\n");
    printf("\n");
    printf("This is a sample program that just dumps all incoming messages on the  

screen,\n");
    printf("or to a file.\n");
    printf("\nUsage: candump [flags] [filename]\n");
    printf("  -X      Listen to CAN channel number X.\n");
    printf("  -B<value> Set the bitrate. Value is any of 1000,500,250,125.\n");
    printf("          Default bitrate is 125 kbit/s.\n");
    printf("  -h      Print this help text.\n");
    printf("  -q      Be more quiet than usual.\n");
    printf("\nIf no filename is specified, standard output is used.\n\n");
    printf("The following keys can be used during logging:\n");
    printf("  ESC     Stop logging.\n");
    printf("  Q       Be more quiet.\n");
    printf("  q       Be less quiet.\n");
    printf("\nExample:\n");
    printf("candump -B250 -0 logfile.log\n");
    printf("  would set CAN channel 0 to 250 kbit/s and log to logfile.log\n");
    exit(1);
}
////////////////////////////////////
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", (canStatus)stat);
    }

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

/////////////////////////////////////////////////////////////////
void main(int argc, char* argv[])
{
    canStatus stat;
    FILE *f;
    int i;
    int hnd;
    int Ready;
    int Overrun;
    int veri1, veri2, veri;
    unsigned long MessageCount;
    unsigned long timeOffset;
    unsigned char giden_veri[8];
    double referenceValue, measurementValue, inputValue, fark, sinir;
    double referenceDefault=10.0;
    int kontrol=1;

```

```

int oran=10; // % gaz için
int oran2=5; // % fren için

    Filename[0] = '\0';
    referenceValue = 0.0;
    Init();
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, "sssLogging 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_9600;//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%c %02lu ",id,flags &
                canMSG_EXT ? 'x' : '',flags & canMSG_RTR? 'R'
                : '',flags & canMSGERR_OVERRUN ? 'o' : '', flags
                & canMSG_NERR ? 'N' : '',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, "%02x ", msg[i]);

                for (; i<8; i++)
                    fprintf(f, " ");
                aci=0;
                acim.veri[0]=msg[1];
                acim.veri[1]=msg[0];
                measurementValue =acim.aci*0.0625;
                if (kontrol==1)
                {
                    fark=(referenceValue-measurementValue);

```

```

sinir=(float)((referenceValue*oran2)/100);
if(fark <= sinir)
{
    inputValue = pid_Controller(referenceValue, measurementValue,
    &pidDataf);
}
else
{
    kontrol=2;
    pid_Init(K_Pf, K_If, K_Df, T, N_Df, &pidDataf);
    inputValue = pid_Controller(referenceValue, measurementValue,
    &pidData);
}
}
else
{
    fark=(measurementValue-referenceValue);
    sinir=(float)((referenceValue*oran)/100);
    if(fark < sinir)
    {
        inputValue = pid_Controller(referenceValue, measurementValue,
        &pidData);
    }
    else
    {
        kontrol=1;
        pid_Init(K_P, K_I, K_D, T, N_D, &pidData);
        inputValue = pid_Controller(referenceValue, measurementValue,
        &pidDataf);
    }
}
fprintf(f, "hiz= %3d : %7.4lf, ref=%lf,kont=%10.6lf", acim.aci,
measurementValue,referenceValue,inputValue );
saydir++;
if(saydir==1)
{
    saydir=0;
    giden_veri[0]=0x55;
    giden_veri[1]=0xaa;
    if(deger==1)
    {
        if(kontrol==2)//(inputValue>0)
        {
            veri1=(int)(inputValue*40.0);
            if(veri1>4000)
                veri1=4000;
            if(veri1<0)
                veri1=0;
            veri2=0;
            else

```

```

        {
            veri2=(int)(-1*inputValue*40.0);

            if(veri2>4000)
                veri2=4000;
            if(veri2<0)
                veri2=0;
            veri1=0;
        }
    else
    {
        veri1=0;
        veri2=0;
    }
    fprintf(f, ": dac= %d,%d",veri1,veri2);
    giden_veri[3]=(veri2&0xf00)/256;
    giden_veri[2]=(veri2&0xff);
    giden_veri[5]=(veri1&0xf00)/256;
    giden_veri[4]=(veri1&0xff);
    veri=0x55aa+(veri1&0xfff)+(veri2&0xfff);
    giden_veri[7]=(veri&0xff00)/256;
    giden_veri[6]=(veri&0xff);
    WriteFile(hComm, giden_veri, 8, &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;
        veri1=0;
        veri2=0;
        giden_veri[3]=(veri2&0xf00)/256;
        giden_veri[2]=(veri2&0xff);
        giden_veri[5]=(veri1&0xf00)/256;
        giden_veri[4]=(veri1&0xff);
        veri=0x55aa+(veri1&0xffff)+(veri2&0xffff);
        giden_veri[7]=(veri&0xff00)/256;
        giden_veri[6]=(veri&0xff);
        WriteFile(hComm, giden_veri, 8, &dwBytesRead, NULL);
    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 's':
        deger=0;
    break;
    case 'g':
        deger=1;
        Init();
    break;
    case 'r':
        referenceValue=referenceDefault;
        Init();
    break;
    case 'y':
        referenceValue=0.0;
    break;
    case '+':
        referenceValue+=5;
    break;
    case '-':
        referenceValue-=5;
        if(referenceValue<0.0)

```

```

        referenceValue=0.0;
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);
}

```


ÖZGEÇMİŞ

Ad Soyad :Şafak GÜNER
Doğum Yeri ve Tarihi :Ardahan, 1984
Adres :Okan Üniversitesi Formula 1 Yanı Tuzla Kampüsü,
İstanbul
E-Posta :safakguner@hotmail.com
Lisans :Marmara Üniversitesi,Teknik Eğitim Fakültesi,
Bilgisayar ve Kontrol Öğretmenliği

Mesleki Deneyim ve Ödüller: Okan Üniversitesi, Mühendislik Mimarlık
Fakültesi, Bilgisayar Mühendisliği Bölümü
Araştırma Görevlisi,(2007-)