

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

**SAHA VERİYOLU İLE  
ÜÇ EKSENLİ HAREKETİN KONTROLÜ**

**Yılmaz GÜVEN**

**YÜKSEK LİSANS TEZİ  
ELEKTRİK EĞİTİMİ ANABİLİM DALI  
ELEKTRİK EĞİTİMİ PROGRAMI**

**DANIŞMAN  
Yrd. Doç. Dr. Selçuk ATIŞ**

**İSTANBUL 2010**

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

**SAHA VERİYOLU İLE  
ÜÇ EKSENLİ HAREKETİN KONTROLÜ**

**Yılmaz GÜVEN  
(522308013)**

**YÜKSEK LİSANS TEZİ  
ELEKTRİK EĞİTİMİ ANABİLİM DALI  
ELEKTRİK EĞİTİMİ PROGRAMI**

**DANIŞMAN  
Yrd. Doç. Dr. Selçuk ATIŞ**

**İSTANBUL 2010**

# TEŐEKKÜR

Bu tezin ön hazırlık, uygulama ve yazım aŐamalarında bana danıŐmanlık yapan saygıdeđer hocam Yrd. Doç. Dr. Selçuk ATİŐ'e teŐekkür ederim. Ayrıca laboratuvar, donanım ve tüm malzeme imkânlarından yararlanmama izin veren başta Yrd. Doç. Dr. Nazmi EKREN olmak üzere tüm Teknik Eđitim Fakóltesi Elektrik Teknolojisi bölümü hocalarına minnet ve saygılarımı sunarım.

**TEMMUZ, 2010**

Yılmaz GÜVEN

# İÇİNDEKİLER

	SAYFA
TEŞEKKÜR.....	i
İÇİNDEKİLER.....	ii
ÖZET.....	v
ABSTRACT.....	vi
SEMBOLLER.....	vii
KISALTMALAR.....	ix
ŞEKİLLER.....	xi
TABLolar .....	xiv
<b>BÖLÜM I. GİRİŞ ve AMAÇ.....</b>	<b>1</b>
I.1 GİRİŞ.....	1
I.2 AMAÇ.....	3
<b>BÖLÜM II. HAREKET KONTROL SİSTEMLERİ HAKKINDA GENEL BİLGİLER.....</b>	<b>4</b>
<b>II.1 KONTROL SİSTEMLERİ.....</b>	<b>4</b>
<b>II.2 SERVO SİSTEMLER.....</b>	<b>5</b>
<b>II.2.1 Servo Motorlar.....</b>	<b>7</b>
<b>II.2.1.1 Adım Motorlar.....</b>	<b>9</b>
<b>II.2.1.2 Fırçalı DC Motorlar.....</b>	<b>9</b>
<b>II.2.1.3 Fırçasız Motorlar.....</b>	<b>11</b>
<b>II.2.1.4 Hibrid Servo.....</b>	<b>13</b>
<b>II.2.1.5 Direk Sürücülü Motorla.....</b>	<b>13</b>
<b>II.2.2 Komitasyon.....</b>	<b>14</b>
<b>II.2.2.1 Fırçasız Motorlarda Komitasyon.....</b>	<b>15</b>
<b>II.2.2.2 Trapezoidal Komitasyon.....</b>	<b>16</b>
<b>II.2.2.3 Sinüzoidal Komitasyon.....</b>	<b>19</b>
<b>II.2.2.4 Trapezoidal ve Sinüzoidal komitasyonun karşılaştırılması.....</b>	<b>21</b>
<b>II.2.3 Sürücü Devreleri.....</b>	<b>23</b>

II.2.3.1 Adım Motor Sürücülerini.....	24
II.2.3.2 Mini Adımlı Sürücüler.....	25
II.2.3.3 Fırçalı DC Motor Sürücülerini.....	26
II.2.3.4 Analog ve Dijital Servo Sürücüler.....	27
II.2.3.5 Fırçasız Motor Sürücülerini.....	29
II.2.4 Servo Sistem Ayarlamaları.....	30
II.2.5 Geri Besleme Elemanları.....	34
<b>II.3 VERİ İLETİŞİMİ.....</b>	<b>36</b>
II.3.1 Saha Veri Yolu Yapısı.....	37
II.3.2 Saha Veri Yolu Protokolleri.....	40
II.3.2.1 AS-Interface.....	42
II.3.2.2 Profibus.....	42
II.3.2.3 Can Bus.....	43
<b>BÖLÜM III SAHA VERİ YOLU İLE ÜÇ EKSENLİ HAREKETİN KONTROLÜ.....</b>	<b>48</b>
<b>III.1 SİSTEM BİLEŞENLERİ.....</b>	<b>48</b>
III.1.1 Doğrusal hareket modül.....	48
III.1.2 Tahrik mekanizması.....	51
III.1.2.1 Motor seçimi.....	51
III.1.2.2 Enerji bağlantı konektörleri.....	56
III.1.2.3 Geri besleme elemanları.....	57
III.1.3 Sürücüler.....	60
III.1.3.1 Bassoon dijital servo sürücü.....	60
III.1.3.2 Maestro çok eksenli kontrolör.....	62
<b>III.2 SİSTEM TASARIMI.....</b>	<b>65</b>
III.2.1 Gövde profili.....	66
III.2.2 Kontrol panosu.....	67
III.2.2.1 Güç ve kontrol bağlantıları.....	68
III.2.2.2 Haberleşme bağlantıları.....	70
III.2.3 Kurulum ve ayarlar.....	72
III.2.3.1 Sürücü kurulumu ve ayarları.....	73
III.2.3.2 Çok eksenli kontrolör kurulumu ve ayarları.....	78
<b>III.3 PROGRAMLAMA.....</b>	<b>83</b>

III.3.1 İfade ve operatörler.....	83
III.3.2 Kullanıcı değişkenleri.....	86
III.3.3 Hazır fonksiyon çağırma.....	87
III.3.4 Kullanıcı programlama dili özellikleri.....	87
III.3.5 Kod yazım standardı.....	89
III.3.6 Sistem komutları.....	90
III.3.7 Program akış komutları.....	91
III.3.8 Fonksiyonlar.....	91
III.3.9 Otomatik alt rutinler.....	92
III.3.10 Haberleşme bağlantı ayarları.....	94
III.3.11 Eksen özellikleri.....	95
III.3.12 Vektör özellikleri.....	96
III.3.13 Programlama örnekleri.....	97
III.4 SİSTEM MONTAJI.....	101
<b>BÖLÜM IV. SONUÇLAR ve TARTIŞMA.....</b>	<b>105</b>
IV.1 SÜRÜCÜ MODLARI.....	105
IV.2 UYGULAMA AYARLARI.....	112
IV.3 PROGRAM.....	113
IV.4 SONUÇLAR.....	114
<b>BÖLÜM V. SON DEĞERLENDİRMELER ve ÖNERİLER.....</b>	<b>116</b>
<b>KAYNAKLAR.....</b>	<b>118</b>
<b>EKLER.....</b>	<b>120</b>
<b>ÖZGEÇMİŞ.....</b>	<b>131</b>

# ÖZET

## SAHA VERİ YOLU İLE ÜÇ EKSENLİ HAREKETİN KONTROLÜ

Günümüzde kontrol teknolojileri hızla gelişmektedir. Bu gelişmeler sonucunda sistem gereksinimleri de artmaktadır. Bununla birlikte hızlı, verimli ve uygulanabilir sistemler için yüksek teknolojiye sahip sürücüler ve motorlar üretilmektedir. Ayrıca kontrol denince akla veri iletişimi de gelmektedir. Sistemlerin güvenilirliği ve hızı veri iletişim tekniği ile yakından ilgilidir.

Bu tezin amacı çok eksenli bir sistemin eş zamanlı olarak elektronik sürücülerle kontrol edilmesidir. Sistemde tahrik mekanizması olarak küçük yapıları ve elektriksel üstünlüklerinden dolayı fırçasız servo motorlar tercih edilmiştir. Dijital sürücülerle sürülen motorlar çok eksenli bir kontrolör üzerinden kontrol edilmektedir. Sürücüler ve kontrolör arasındaki iletişim ise saha veri yolu üzerinden sağlanmaktadır. Saha veri yolu klasik haberleşme yöntemlerine göre daha az kablolu masrafı ve montaj kolaylığının yanı sıra hız ve farklı cihazlara bağlanabilirliği gibi üstünlüklerinden dolayı tercih edilmiştir.

Tezin ilk bölümünde tanımlanan problemin çözümü için gerekli olan teorik bilgiler ikinci bölümde ayrıntılı olarak ele alınmıştır. Sürücü ve motor çeşitlerinin yanı sıra farklı saha veri yolu protokollerine yer verilmiştir. Üçüncü bölümde ise uygulamada kullanılan araç-gereç ve malzemelere tanıtılmıştır. Tasarım, montaj, kurulum ve programlama aşamaları ayrı ayrı ele alınmıştır. Dördüncü bölümde uygulamada alınan sonuç ve grafiklere yer verilmiş, karşılaşılan sorunlar hakkında bilgi verilmiştir. Son bölümde ise ileriki uygulamalarda izlenecek adımlar ve karşılaşılabilecek sorunlara yaklaşımlar ele alınmıştır.

**Temmuz, 2010**

**Yılmaz GÜVEN**

# **ABSTRACT**

## **THREE AXIAL MOTION KONTROL WITH FIELDBUS**

Nowadays, control technologies are improving rapidly. As a result of this improvement, system requirements are increasing as well. Along with it, high technologic drivers and motors are being produced for fast, effective and applicable systems. Besides, data communication is occurred where control is mentioned. Reliability and speed of systems are closely relevant with data communication technique.

Aim of this thesis is controlling a multi axis system with electronic drivers synchronously. Because of their compact size and electrical superiority, brushless servo motors are chosen as impellent. Digitally driven motors are being controlled by multi axis controller. Communication between drivers and controller is being established over filed bus. Field bus is preferred because of its connectivity, assembly and cost.

In second chapter of this thesis, theoretical information is given to solve the problem which is pointed out in first chapter of this thesis. Beside different drivers and motors, different field bus protocols are discussed. In third chapter, tools, materials, hardware and software are handled which have used for application. Planning, assembly, setup and programming stages are handled separately. In fourth chapter, application result and graphics are presented. Information is given about problems and their solutions. For further application, itinerary steps and approaches are handled in last chapter.

**July, 2010**

**Yılmaz GÜVEN**

# SEMBOLLER

<b>D(s)</b>	: Bozucu giriş
<b>E</b>	: Elektro motor kuvvet (V)
<b>E<sub>a</sub></b>	: Stator birinci sargısı elektro motor kuvveti (V)
<b>E<sub>b</sub></b>	: Stator ikinci sargısı elektro motor kuvveti (V)
<b>E<sub>c</sub></b>	: Stator üçüncü sargısı elektro motor kuvveti (V)
<b>E<sub>f-f</sub></b>	: Fazlar arası elektro motor kuvvet (V)
<b>E(s)</b>	: Hata-Sapma
<b>F</b>	: Kuvvet (N)
<b>G(s)</b>	: İleri yol transfer fonksiyonu
<b>I</b>	: Akım (A)
<b>I<sub>a</sub></b>	: Stator birinci sargı akım (A)
<b>I<sub>b</sub></b>	: Stator ikinci sargı akım (A)
<b>I<sub>c</sub></b>	: Stator üçüncü sargı akım (A)
<b>I<sub>d</sub></b>	: Reaktif akım (A)
<b>I<sub>q</sub></b>	: Aktif akım (A)
<b>K<sub>e</sub></b>	: Elektromanyetik sabit
<b>K<sub>t</sub></b>	: Moment sabiti
<b>KI</b>	: İntegral kazanç
<b>KP</b>	: Oransal kazanç
<b>L</b>	: Endükstans (H)
<b>M</b>	: Moment (Nm)
<b>N<sub>p</sub></b>	: Motor kutup sayısı
<b>R</b>	: Omik direnç ( $\Omega$ )
<b>R(s)</b>	: Referans girişi
<b>T</b>	: Moment Kuvveti (Nm)
<b>V</b>	: Doğrusal hız (m/s)
<b>V<sub>b</sub></b>	: Doğru akım bus gerilimi (V)
<b>Y(s)</b>	: Sistem çıkışı
<b><math>\beta</math></b>	: Sönüm derecesi
<b><math>\theta</math></b>	: Manyetik açı ( $^{\circ}$ )

$\Sigma$  : Toplam  
 $\omega$  : Açısal hız (rad/sn)

# KISALTMALAR

<b>AC</b>	: Alternatif akım (Alternative Current)
<b>ASI</b>	: Aktuatör ve sensor arayüzü (Actuator sensor interface)
<b>AWG</b>	: Amerikan kablo ölçüsü (American Wire Gauge)
<b>BA</b>	: Bit bit karşılaştırma (Bitwise Arbitration)
<b>BLDC</b>	: Fırçasız doğru akım motoru (Brushless Direct Current Motor)
<b>BJT</b>	: Çift kutuplu transistor (Bipolar Junction Transistor)
<b>BUS</b>	: Veri yolu
<b>CAN</b>	: Kontrolör saha iletişim ağı (Controller Area Network)
<b>CPU</b>	: Merkezi İşlem Birimi (Central Processing Unit)
<b>CRC</b>	: Döngüsel atık denetimi ( <i>Cyclic redundancy check</i> )
<b>CSMA</b>	: Taşıyıcı duyumlu çoklu geçiş ( <i>Carrier sense multiple access</i> )
<b>DC</b>	: Doğru akım (Direct Current)
<b>DLC</b>	: Veri uzunluk kodu (Data Length code)
<b>DP</b>	: Merkezi yönetimli cihazlar (Decentralized Periphery)
<b>DSP</b>	: Dijital sinyal işlemcisi (Digital Signal Processor)
<b>ECU</b>	: Elektronik kontrol ünitesi (Electronic Control Unit)
<b>EHL</b>	: Elmo yüksek seviye dili (Elmo high level language)
<b>EMC</b>	: Elektromanyetik uyumluluk (Electromagnetic compatibility)
<b>E.M.K.</b>	: Elektro motor kuvvet
<b>GTO</b>	: Kapı sönmürlü tristör (Gate Turn-Off Thyristor)
<b>HMI</b>	: İnsan makine arayüzü (Human Machine Interface)
<b>HTD</b>	: Yüksek tork dişlisi (High Torque Domatic)
<b>IC</b>	: Bütünleşik devre (Integrated Circuit)
<b>ID</b>	: Kimlik bilgisi (Identifier)
<b>IDE</b>	: Genişletilmiş kimlik bilgisi (Identifier Extend)
<b>IEEE</b>	: Uluslararası elektrik-elektronik mühendisleri enstitüsü
<b>IEC</b>	: Uluslararası elektroteknik komisyonu (International Electrotechnical Commission)
<b>IGBT</b>	: Yalıtılmış kapı çift kutuplu transistor (Insulated-gate Bipolar Transistor)

<b>IP</b>	: İnternet Protokolü (Internet Protocol)
<b>LAN</b>	: Yerel alan ağı (Local Area Network)
<b>MOSFET</b>	: Metaloksit alan etkili transistor (Metal Oxide Semiconductor Field Effect Transistor)
<b>OS</b>	: Operasyon sistemi (Operating System)
<b>PA</b>	: Süreç kontrol otomasyonu (Process Automation)
<b>PC</b>	: Kişisel bilgisayar (Personal Computer)
<b>PI</b>	: Oransal-integral kontrolör (Proportional-integral controller)
<b>PMAC</b>	: Daimi mıknatıslı alternatif akım motoru (Permanent magnet AC motor)
<b>PVT</b>	: Konum, hız, zaman tablosu (Position-Velocity-Time table)
<b>PWM</b>	: Darbe genliği modülasyonu (Pulse Width Modulation)
<b>RAM</b>	: Rastgele erişim belleği (Random Access Memory)
<b>RMS</b>	: Ortalama eşdeğer (Root Mean Square)
<b>RTR</b>	: Uzak ileti çağrısı (Remote transmitter request)
<b>SMPS</b>	: Anahtar modu güç kaynağı (Switch Mode Power Supply)
<b>SRR</b>	: Yedek uzak çağır (Sub remote request)
<b>TCP</b>	: İletim kontrol protokolü (Transmission Control Protocol)
<b>VSI</b>	: Gerilim kaynaklı invertör (Voltage Source Inverter)

# ŞEKİLLER

	<u>SAYFA NO</u>
Şekil II.1 Kontrol Blok Şeması .....	5
Şekil II.2 Hareket kontrolü bileşenleri .....	6
Şekil II.3 Kapalı çevrim servo sistem.....	6
Şekil II.4 Motor seçiminde izlenecek yönteme ait blok şeması .....	8
Şekil II.5 Adım motor yapısı.....	9
Şekil II.6 DC motor yapısı .....	10
Şekil II.7 DC motor tipleri .....	10
Şekil II.8 Fırçasız motor yapısı .....	11
Şekil II.9 Farklı faz akımları için stator alanı konumu.....	12
Şekil II.10 Komitasyon noktalarında rotorun konumu.....	13
Şekil II.11 Fırçasız motor eşdeğer devresi .....	15
Şekil II.12 Hall sensörü ile pozisyon ölçümü.....	16
Şekil II.13 Moment dalgalanması.....	18
Şekil II.14 Fırçasız motor trapezoidal sürücü devresi.....	18
Şekil II.15 Fırçasız motor sinüzoidal sürücü devresi .....	20
Şekil II.16 Gerilim kaynaklı inverter ile sabit mıknatıslı motor sürülmesi.....	21
Şekil II.17 Trapezoidal komitasyon .....	22
Şekil II.18 Sinüzoidal komitasyon .....	22
Şekil II.19 Adım motor sürücü bileşenleri .....	24
Şekil II.20 Basit tek kutuplu sürücü devresi.....	25
Şekil II.21 Basit çift kutuplu sürücü devresi .....	25
Şekil II.22 Microstep akım profilleri.....	26
Şekil II.23 Doğrusal servo yükselteç.....	26
Şekil II.24 Analog servo sürücü sistemi.....	27
Şekil II.25 Dört bölgeli çalışma .....	28
Şekil II.26 Dijital Servo Sürücü .....	28
Şekil II.27 Trapezoidal Fırçasız Servo Sürücüsü .....	29
Şekil II.28 İki faz sinüzoidal fırçasız servo sürücü .....	30

Şekil II.29 Sistem cevabı karakteristiği.....	31
Şekil II.30 Kapalı ve açık çevrim karakteristik ölçümü .....	32
Şekil II.31 Açık çevrim kazanç ve faz karakteristiği .....	32
Şekil II.32 Sönümlü sistem cevabı .....	33
Şekil II.33 Servo sistem açık çevrim kazancı ve faz karakteristiği.....	33
Şekil II.34 Yük altında sistem karakteristiği .....	34
Şekil II.35 Hall sensörü .....	35
Şekil II.36 Resolver yapısı .....	35
Şekil II.37 Dairesel optik encoder .....	36
Şekil II.38 Seri ve paralel iletişim basit şeması.....	37
Şekil II.39 Saha veri yolu piramit hiyerarşisi .....	38
Şekil II.40 Çeşitli ağların topolojisi .....	39
Şekil II.41 Veri aktarım teknikleri.....	40
Şekil II.42 CAN verisinin genel yapısı.....	43
Şekil II.43 CAN modülü blok şeması .....	44
Şekil II.44 Bit zamanlaması .....	45
Şekil III.1 Doğrusal Hareket Modülleri .....	49
Şekil III.2 Doğrusal eksenler.....	50
Şekil III.3 HR92 serisi motor yan görünüm şeması .....	53
Şekil III.4 HR92 serisi motor ön görünüm şeması.....	53
Şekil III.5 HR55 serisi motor yan görünüm şeması .....	55
Şekil III.6 HR55 serisi motor ön görünüm şeması.....	55
Şekil III.7 Motor enerji besleme bağlantı uçları .....	56
Şekil III.8 Resolver çalışma prensip şeması .....	58
Şekil III.9 Resolver sargı uçları.....	59
Şekil III.10 Geri besleme bağlantı konektörü .....	59
Şekil III.11 Bassoon sistem blok şeması.....	61
Şekil III.12 Basson bağlantı konektörleri.....	61
Şekil III.13 Maestro sistem blok şeması .....	64
Şekil III.14 Maestro bağlantıları konektörleri.....	65
Şekil III.15 45x45mm Profil kesit görünümü .....	66
Şekil III.16 45x90mm Profil kesit görünümü .....	66
Şekil III.17 Kontrol panosu yerleşim şeması .....	67
Şekil III.18 Haberleşme bağlantıları .....	70

<b>Şekil III.19</b> Saha veri yolu bağlantısı .....	72
<b>Şekil III.20</b> Motor bilgilerinin girilmesi.....	74
<b>Şekil III.21</b> Sistem limitlerinin ayarlanması .....	75
<b>Şekil III.22</b> Hız döngüsü ayarlamaları .....	76
<b>Şekil III.23</b> Konum döngüsü ayarlamaları .....	77
<b>Şekil III.24</b> Ayar parametrelerinin sürücüye kaydedilmesi .....	78
<b>Şekil III.25</b> Ethernet üzerinden TCP/IP bağlantısının gerçekleştirilmesi .....	79
<b>Şekil III.26</b> Kompozitör üzerinden CAN nesne bağlantıları .....	79
<b>Şekil III.27</b> İnternet protokol numarasının girilmesi.....	80
<b>Şekil III.28</b> Elmo Studio çalışma alanı bağlantısı .....	80
<b>Şekil III.29</b> Kaynak bilgileri eşlemesi.....	81
<b>Şekil III.30</b> Eksen ve vektör bilgilerinin girilmesi .....	82
<b>Şekil III.31</b> Eksen ve vektör bilgileri .....	83
<b>Şekil III.32</b> CAN bilgilerinin düzenlenmesi .....	94
<b>Şekil III.33</b> Telnet bağlantısı.....	95
<b>Şekil III.34</b> İki boyutlu doğrusal hareket çıkış grafiği .....	97
<b>Şekil III.35</b> Üç boyutlu doğrusal hareket çıkış grafiği .....	100
<b>Şekil III.36</b> Mekanik sistem .....	102
<b>Şekil III.37</b> Kapalı tip mekanik kablo taşıyıcı .....	102
<b>Şekil III.38</b> Optik sensör yerleşimleri .....	103
<b>Şekil III.39</b> Pano görünümü .....	104
<b>Şekil IV.1</b> Akım kontrolör sistem şeması.....	106
<b>Şekil IV.2</b> PI akım kontrolör şeması.....	106
<b>Şekil IV.3</b> Akım komutu ve sistem cevabını gösteren grafik .....	107
<b>Şekil IV.4</b> Hız modu sistem cevabını gösteren grafik .....	108
<b>Şekil IV.5</b> Hız modu sistem şeması.....	109
<b>Şekil IV.6</b> Üç eksen için hız komutu ve sistem cevabını gösteren grafik .....	109
<b>Şekil IV.7</b> Pozisyon modu sistem cevabını gösteren grafik .....	110
<b>Şekil IV.8</b> Pozisyon modu sistem şeması.....	111
<b>Şekil IV.9</b> Üç eksen için pozisyon komutu ve sistem cevabını gösteren grafik.....	111
<b>Şekil IV.10</b> Uygulama limitleri .....	112

# TABLolar

## SAYFA NO

<b>Tablo II.1</b> Pozisyon bilgisi bit deęerleri.....	16
<b>Tablo II.2</b> Rotor pozisyonuna gre faz çiftleri.....	17
<b>Tablo II.3</b> Yarı iletken karakteristikleri .....	23
<b>Tablo II.4</b> Veri aktarım ortamları genel özellikleri .....	38
<b>Tablo II.5</b> Çeşitli saha veri yolu (Fieldbus) protokolleri ve temel özellikleri .....	41
<b>Tablo II.6</b> Bazı açık ve özgün mimari protokoller.....	42
<b>Tablo II.7</b> Standart ifade formatı .....	46
<b>Tablo II.8</b> Genişletilmiş ifade formatı .....	47
<b>Tablo III.1</b> Doğrusal eksen dinamik hareket deęerleri .....	50
<b>Tablo III.2</b> Motor koruma sınıfı bilgileri .....	52
<b>Tablo III.3</b> Motor tipi tanımlama kodları.....	52
<b>Tablo III.4</b> HR92C4-64S Teknik verileri .....	54
<b>Tablo III.5</b> HR55C4-32S Teknik verileri .....	56
<b>Tablo III.6</b> Inter konektr enerji besleme baęlantı uçları .....	57
<b>Tablo III.7</b> Encoder ve Resolver standart özellikleri.....	58
<b>Tablo III.8</b> Geri besleme baęlantı uçları.....	59
<b>Tablo III.9</b> BAS-3/230R sürücü genel özellikleri.....	62
<b>Tablo III.10</b> Maestro çok eksenli kontrolr genel özellikleri.....	63
<b>Tablo III.11</b> Sürücü güç baęlantıları.....	68
<b>Tablo III.12</b> Harici yardımcı güç baęlantıları.....	68
<b>Tablo III.13</b> Geri besleme elemanları baęlantısı .....	69
<b>Tablo III.14</b> RS-232 baęlantısı için RJ-45 ataması .....	71
<b>Tablo III.15</b> CAN baęlantısı için RJ-45 ataması .....	71
<b>Tablo III.16</b> Matematiksel ve mantıksal operatrler .....	84
<b>Tablo III.17</b> Program komutları.....	87
<b>Tablo III.18</b> Program akış komutları .....	90
<b>Tablo III.19</b> Otomatik alt rutinler .....	93
<b>Tablo IV.1</b> Birim modları .....	105
<b>Tablo IV.2</b> Uygulama ayarları .....	113

# BÖLÜM I

## I.1 GİRİŞ

Günümüzde teknolojinin her alanında hızlı bir değişim yaşanmakta ve bu değişim ile birlikte ortaya yeni ihtiyaçlar çıkmaktadır. Teknolojinin hangi alanında olursa olsun hızlı ve kullanışlı sistemlere olan ihtiyacımız sürekli kendini yenilemektedir. Teknolojinin insan gereksinimlerini karşılaması noktasında ise otomasyon ve kontrol önemli bir yere sahiptir.

Hareket kontrolü ve otomasyonunda genellikle step veya fırçasız servo motorlar kullanılmaktadır. Step motorun yapısı ve sürücüsü oldukça basittir. Sistemlerdeki hız ve moment gibi ihtiyaçlar artmaya başladığından beri step motorların kullanımı azalmaya başlamıştır. Step motorların yerine hız ve moment talebini karşılayan servo sistemler yaygın olarak kullanılmaya başlanmıştır. Bir elektrikli servo sistem genellikle bir motor ve sürücüden oluşmaktadır. Motorlar yapısında kolektör ve fırçalar gibi elemanlar bulunup bulunmaması, sargıların alternatif akım (AC) veya doğru akım (DC) ile sürülmesi gibi farklılıklar göre farklı isimler alırlar. Sürücüler ise yüksek işlevsellik ve hız içeren dijital veya analog olarak tasarlanmış elektronik sistemlerdir.

Step motorlarda adım için geri besleme alınmaması ve düşük moment en büyük olumsuzluklardır. Fırçalı motorlarda ise kolektör ve fırçalardan kaynaklanan birçok olumsuzluklar mevcuttur. Bu dezavantajlar sebebiyle servo sistemlerde en çok tercih edilen motor türü fırçasız servo motorlardır. Fırçasız servo motorların; hızlı kalkış ve duruş tepkileri, düşük atalet momenti, darbe kaybının olmaması, yüksek hızlarda yüksek moment alma imkânı, hassas hareket kontrolü, yüksek verimlilik ve gürültü azlığı (fan olmaması) diğer motorlardan ayıran önemli üstünlükleridir. Günümüzde servo sistemler için birçok motor türü denenmeye devam etmektedir. Bugün için hem maliyet hem de uygulanabilirlik açısından en uygun motor türü fırçasız motor olarak görünmektedir.

Servo sistemlerin diđer önemli bir parçası sürücüdür. Motordan alınan geri besleme bilgileri ve sistemin başka noktalarından alınan bilgileri burada işlenerek sistemin sürekliliđi sağlanmaktadır. Bu noktada sürücünün işlem kapasitesi ve hızı kadar sistemin veri iletişiminin önemi de oldukça büyüktür. Günümüzde endüstriyel uygulamalar için birçok farklı veri iletişim tekniđi olmasına rağmen her gün daha hızlı ve verimli yeni iletişim teknikleri geliştirilmektedir. Geniş kapsamda seri iletişimi avantajları farklı protokol ve donanımlar yardımı ile endüstriyel uygulamalarda hareket kontrolünde yaygın olarak kullanılmaktadır. Saha veri yolu iletişimi (Bit Serial Fielbuses – Fieldbus) bunlardan biridir.

Saha veri yolu modern tesislerdeki üretim bileşenlerinin bütünleşmesi ve birbirlerine koşut çalışabilmesi için saha ve otomasyon seviyelerinde tanımlanmış endüstriyel iletişim ağlarına verilen genel bir isimdir. Saha veri yolundan önce geleneksel metotla özellikle endüstriyel tesislerde kontrolörlerin giriş ve çıkış kartlarına sinyallerin paralel kablolarla taşıma zorunluluđu ve sinyal sayılarının artmasına paralel artan kablo sayısı sistemlerin geliştirilmesine engel oluşturmaktaydı. Bu dezavantajları ortadan kaldırmak için alternatif bir iletişim yönteminin kullanılması gerekliliđini doğurmuştur. Bu noktada kontrolörlerin CPU'ların yanına veya CPU içine yerleştirilebilir bir başka iletişim kartı üretilerek sinyaller özellikle uzaktaki panolara seri protokoller üzerinden iletilmiştir. Sahalardaki bağlantı kutularının yerine saha veri yolu haberleşme sisteminin dağınmık giriş-çıkış modülleri yer almıştır.[1]

Geleneksel iletişim metotlarında yüksek projelendirme ve devreye alma masrafları, yüksek arıza oranları, sabit ve karışık yapı, sistemi geliştirmenin zor ve masraflı olması, yüksek kablo ve kablolama aksesuarı ve diđer donanım malzemelerinin kullanımının yanı sıra bakım zorluđu, dış etkenlere karşı duyarlılık ve yalıtım problemleri gibi birçok dezavantaj bulunmaktadır. Saha veri yolunun geleneksel sistemlere göre avantajları ise azalan projelendirme ve devreye alma masrafları, basit montaj ve devreye alma, kablolama masrafları ve zamandan tasarruf, hata teşhis imkânları ve kolay arıza takibi, panolarda yerden kazanım, dış etkenlerden etkilenme düzeyinin çok düşük olması, genişleme ve sistem durdurulmadan bakım imkânıdır.[1]

Tüm bu avantaj ve dezavantajlar göz önüne alındığında iyi bir hareket kontrolü için yüksek verimli motorlara, fonksiyonel sürücülere ve hızlı olduđu kadar güvenilir bir iletişim sistemini ihtiyaç duyulmaktadır.

## **I.2.AMAÇ**

Üç eksenli hareketin kontrolü için tasarlanacak sistemde tahrik mekanizması olarak yüksek çözünürlüklü geri beslemeye sahip fırçasız servo motorlar kullanılacaktır. Fırçasız servo motorların sürülmesinde kompakt dijital servo sürücüler ve bu sürücülerin üç eksenli hareketini koordine etmek içinde CPU tabanlı çok eksenli bir hareket kontrolörü kullanılacaktır. Sürücülerin birbiriyle ve merkezi kontrolör ile haberleşmesi saha veri yolu (Fieldbus) teknolojisi üzerinden CANopen protokolü kullanılarak gerçekleştirilecektir.

Ayrıca tasarlanacak bu sistemin, mevcut diğer sistemlere göre avantajları ve dezavantajları tartışılacaktır. Tasarlanacak sistem aynı zamanda lisans ve lisansüstü eğitimde Endüstriyel Kontrol ve Otomasyona yönelik derslerinin teorik ve uygulama çalışmasında kullanılacaktır.

## **BÖLÜM II**

### **HAREKET KONTROL SİSTEMLERİ HAKKINDA GENEL BİLGİLER**

Bu bölümde hareket kontrol sistemlerinde kullanılan motor ve sürücü türleri ile ilgili genel bilgiler verilecektir. Ayrıca bu sistemlerde kullanılan haberleşme teknikleri ve protokolleri de incelenecektir. Farklı motor ve sürücüler arasındaki avantaj ve dezavantajlar karşılaştırılarak, farklı uygulamalar için farklı seçenekler ele alınacaktır.

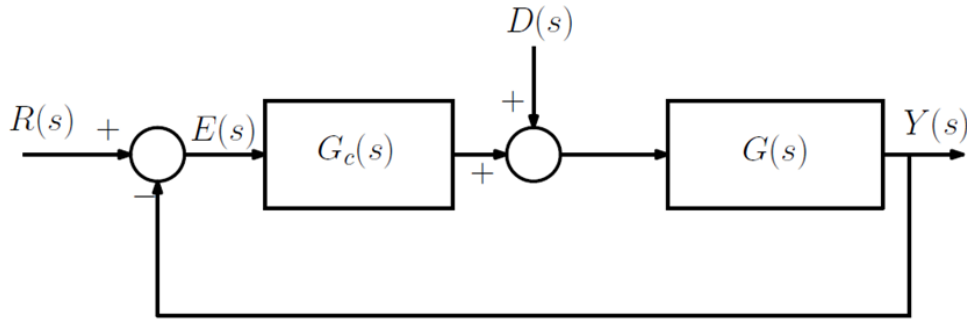
#### **II.1 KONTROL SİSTEMLERİ**

Otomasyon, bir işin insan ile makine arasında paylaşılmasıdır. Toplam işin paylaşım yüzdesi otomasyonun düzeyini belirler. İnsan gücünün yoğun olduğu otomasyon sistemleri yarı otomasyon, makinenin yoğun olduğu sistemlerde tam otomasyon olarak adlandırılırlar. Otomasyonda veri girişi ve programlama gibi düşük düzey işler insanlar tarafından yapılırken, kontrol ve geri besleme gibi rutin ancak üst düzey işlemler makineler tarafından yapılır [2].

Otomatik kontrol fiziksel kontrol sistemlerinin matematiksel olarak ifade edilerek, çözümlendikten sonra fiziksel olarak uygulanması esasına dayanır. Tıpkı alternatif akımın matematiksel bir ifade ile tanımlanarak, aynı esasa dayalı olarak trigonometrik ifadeler yardımı ile üretilmesi olayında olduğu gibi sistemi analiz edebilmek için basitleştirmek ve modellemek gerekir. Otomasyon sistemlerinde ise önemli olan giriş ve çıkış bilgileri arasında matematiksel bir eşitlik yazarak, eşitliği istediğimiz çıkış bilgisine ulaşana kadar düzenlemektir. Kontrolün temelinde giriş ve

çıkışlar arasındaki neden-sonuç ilişkisine dayalı mantıksal bir etkileşim vardır. Bu etkileşim matematiksel olarak ifade edildiği gibi grafiksel olarak da gösterilebilir[2].

Bir sistemin değişkenleri ve işlevlerini tanımlarken öncelikle konu ile ilgili bazı genel ifadelerin açıklanması gerekir. **Sistem**, genel olarak birbiri ile etkileşim içerisinde bir bütün oluşturacak şekilde çalışan elemanlar topluluğudur. **Kontrol** ise düzenleme, ayarlama ve yönetme anlamına gelmektedir. Bu denetimde giriş ve çıkış bilgilerinin doğru biçimde okunması önemlidir. Bir dış enerji kaynağından sisteme uygulanan uyarıların tümüne **Giriş**, girişe karşılık verilen cevaba ise **Çıkış** diyebiliriz. Ayrıca tüm sistemlerde denetimin kesin bir yargı ile yapılabilmesi için bir  $R(s)$  **referans** noktasına ihtiyaç duyulur. Bu referans bir giriş sinyali yahut bir eleman olabilir. Çıkış ile girişteki referans arasındaki fark  $E(s)$  **hata** veya **sapma** olarak adlandırılır. Giriş ile çıkış arasındaki bu fark sistemden kaynaklanabildiği gibi bozucu diğer etkenlerden de kaynaklanabilir. Fakat kontrolün amacı da farklı etkenlerden kaynaklanan tüm hataları ayıklamak ve sistemi ona göre ayarlamaktır[2].



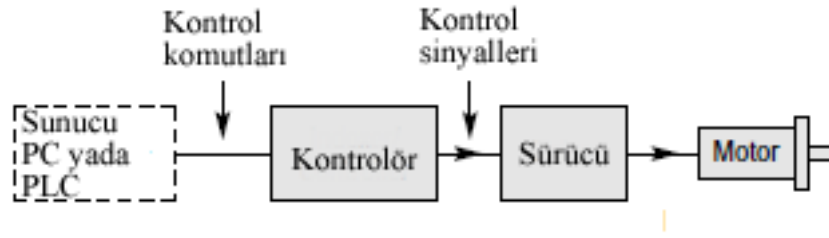
$$Y(s) = E(s)G_c(s)G(s) + D(s)G(s)$$

**Şekil II.1** Kontrol blok şeması

**Geri besleme** elemanları kontrol sistemlerinde en önemli elemanlardan biridir. Kontrol edilen çıkış bilgisini sisteme geri veren ve hata sinyallerinin elde edilmesini sağlayan elemanlardır. Kontrol sistemlerinde arzu edilen çıkışın sağlanmasında farklı elemanlar sisteme eklenebilir. Mesela geri beslemeden elde edilen hatanın çıkışta anında giderilmesi için **ileri besleme** elemanları da kullanılabilir. Geri beslemeli bir kontrol sistemi kapalı bir döngü içerisinde çalışır. Açık döngülü sistemlerde çıkış bilgisi girişe etki etmez. Bu yüzden geri beslemeli kontrol sistemleri daha kararlı ve hızlı sistemlerdir[2].

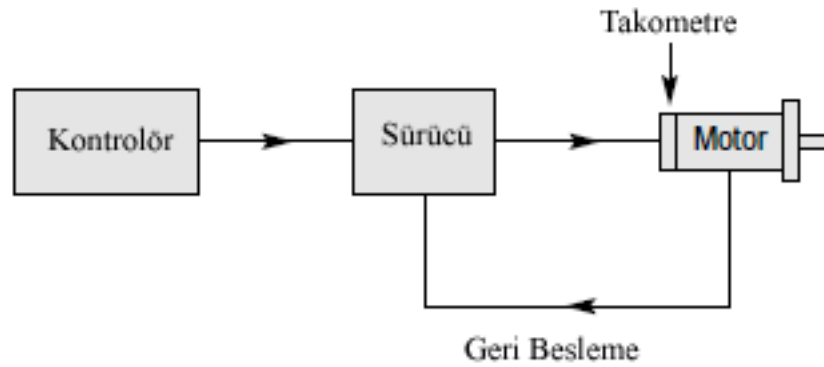
## II.2 SERVO SİSTEMLER

Hareket kontrolünde önemli olan doğru mesafe ve hız kontrolüdür. Genelde bu ikisi bir arada olmakla beraber, moment ve ivme gibi diğer yan etkenlerde kontrolün değişkenleri olarak nitelendirilebilir. Tüm bu değişkenlerin en önemli ortak noktası geri besleme bilgisi gerektirmesidir. Bu yüzden hareket kontrolü dediğimizde sürücü devresi ile eşgüdümlü çalışan geri beslemeli bir hareket mekanizması akla gelmektedir. Hareket kontrolü bileşenleri Şekil II.2’de görülmektedir[3].



Şekil II.2 Hareket kontrolü bileşenleri

Şekil II.2 de motor; adım, fırçalı DC ya da fırçasız DC servo motor olabilir. Adım motoru hariç diğer motorlar kesin ve doğru bir kontrol için bir geri besleme elemanına ihtiyaç duyar. Bu tür geri beslemeli sistemlere kapalı çevrimli sistemler denir. Şekil II.3’de kapalı çevrimli sistem verilmiştir.



Şekil II.3 Kapalı çevrim servo sistem

Şekil II.3’de ki motor için özel sürücü devresi kullanılır. Örneğin adım motor için yapılmış bir sürücü devresini fırçalı DC motor için kullanılamaz. Kontrol sisteminde hız, mesafe, yön ve ivme ölçümü gibi işler birimler tarafından paylaşılarak

yapılabilir. Bu işlemleri bir veya bir kaç sunucu bir bilgisayar tarafından yapılabildiği gibi, çok eksenli hareket kontrolünde olduğu gibi sürücülerini kontrol eden gelişmiş bir kontrolör tarafından da yapılabilir[3].

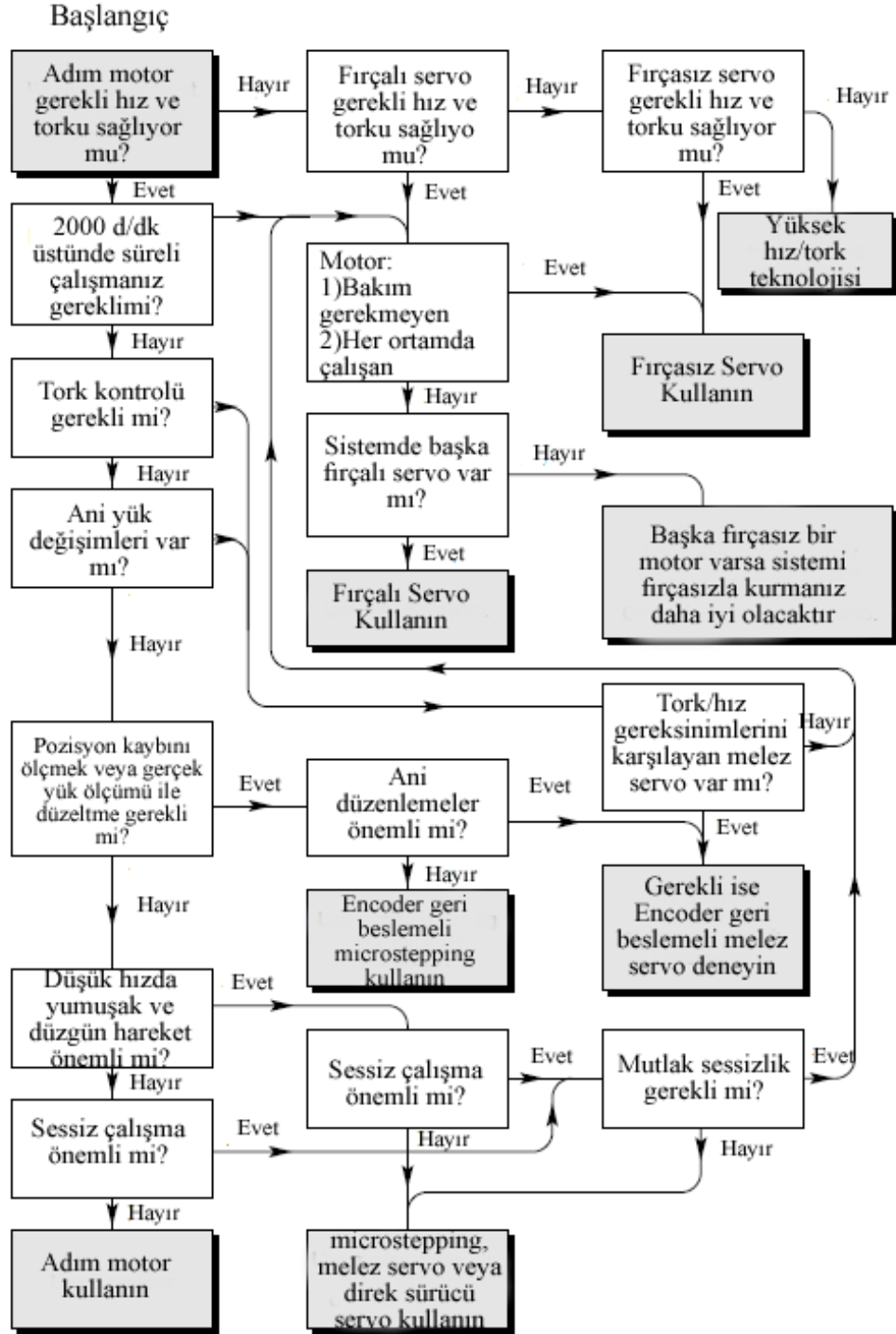
### II.2.1 Servo Motorlar

Servo motorlar diğer motorlara göre küçük boyutlarda olmalarına rağmen yüksek performanslı motorlardır. Hızlı kalkınma ve durma kabiliyetinin yanı sıra hız kontrolü de mükemmeldir. Oldukça gürültüsüz çalışan servo motorlar maliyet ve bakım masrafları açısından da diğer motorlara göre avantajlıdır. Tüm bu avantajları yüzünden günümüzde servo motorların geri besleme elemanları ve sürücü sistemleri geniş bir yelpazede farklı firmalar tarafından geliştirilmiştir. Hareket kontrolünde hızlı tepki zamanı önemli olduğundan, düşük atalet momenti ve yüksek çözünürlüklü geri besleme elemanları yüzünden servo motorların kullanım alanları oldukça genişlemiştir. Çeşitli dişli veya kayışlı mekanik sistemlerle beraber kullanılabilen servo motorların doğrusal hareket edebilen türleri de mevcuttur. Her uygulama için uygun farklı bir motor tipi mevcut olduğu gibi sürücü devresi ve geri besleme elemanı seçimi gibi başka değişkenlerde mevcuttur[3].

Adım motorları ile kurulan sistemlerin maliyeti düşük olmasına rağmen dinamik gereksinimlerin düşük olması gerekir. Yüksek hız, moment ve sürekli çalışma gibi etkenler göz önüne alındığında servo motorların kullanıldığı sistemler diğerlerine göre daha uygundur. Ayrıca düşük hızda yüksek moment kabiliyetleri yüzünden de akış kontrolü gibi birçok uygulamada tercih edilmektedirler. Kısa, ani ve kendini tekrar eden hareketlerde ise adım ve hibrid servo motorlar tercih edilmektedir. DC motorlarda fırçaların neden olduğu bazı problemlerle karşılaşmaktadır. Sık durup, kalkma ve yön değişiklikleri fırçalı DC motorlarda aşınmaya neden olduğundan fırçasız servo motorlar bu tür uygulamalarda daha etkin sonuçlar vermektedir. Bu yüzden yüksek dinamik iş ve yüksek hız gereksinimi olan sistemlerde fırçasız motorlar tercih edilmektedir. Düşük hız ve hassas hareket gerektiren uygulamalarda ise adım motor (*microstepping*) veya direk sürücülü (*direct drive*) servo sistemler tercih edilir[3].

Öte yandan tehlikeli ve havasız ortamlarda fırçalı motorlar kullanılmayacağı için işin türüne göre adım veya fırçasız motorlar kullanılmaktadır. Özellikle hava akışının olmadığı ortamlarda yüke bağlı olarak oluşan ısının yayılımı sorun olarak ortaya çıkacaktır. Bu sebeplerden dolayı motor seçiminde sadece yükün

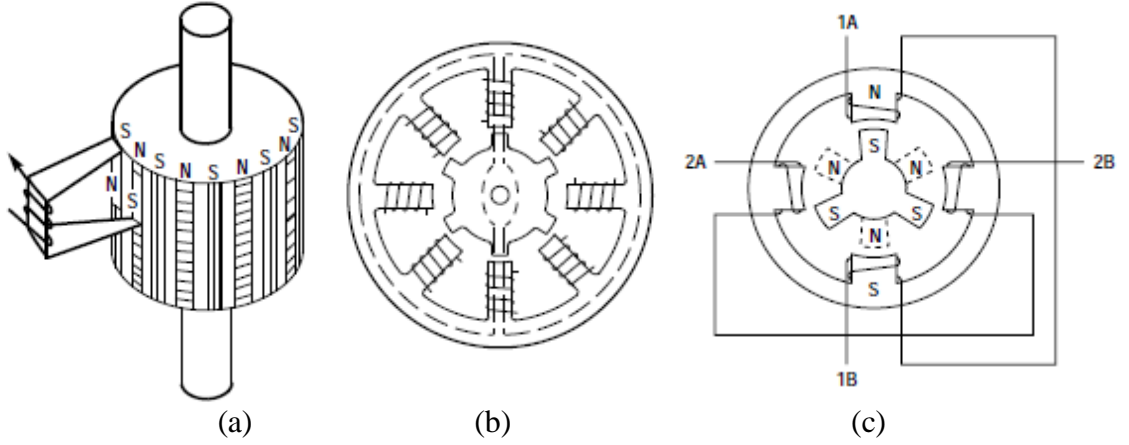
karakteristikleri değil çevre şartlarını da kapsayan birçok değişken göz önünde bulundurulmalıdır. Aynı şekilde endüstriyel uygulamalarda uygulanabilirlik ve maliyetlerde önem kazandığından sistem tasarımda sistem elemanlarının doğru seçimi büyük önem kazanmaktadır. Motor seçiminde izlenecek yöntem Şekil II.4'de görülmektedir [3].



Şekil II.4 Motor seçiminde izlenecek yöntemle ait blok şeması

### II.2.1.1 Adım motorlar

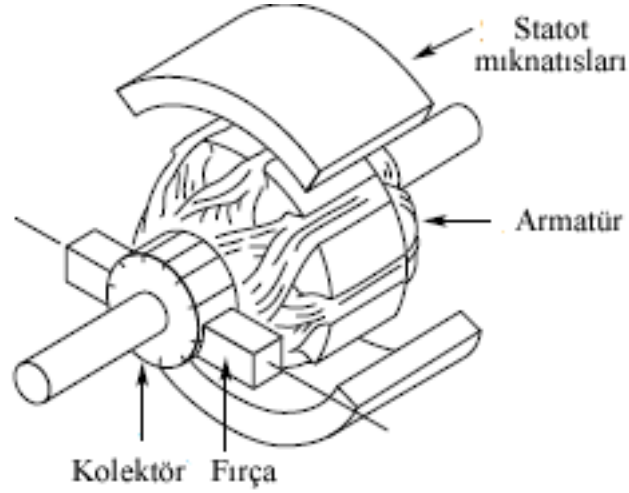
Düşük maliyetli, basit ve bakım gerektirmeyen adım motorlarının kullanım alanları oldukça geniştir. Geri besleme elemanı olmadan çalışabilen adım motorları her türlü ortamda rahatlıkla çalışabilmektedir. Bunun yanında düşük hızda kötü bir performans sergileyen adım motor için *microstep* sürücü devresi geliştirilmiştir. Bu sürücü ile düşük hızlarda daha iyi performans alınabilmektedir. Diğer taraftan yükün durumunda bağımsız çalıştığı için aşırı yükte hız kaybı ve yanında ısınma gibi sorunlarda ortaya çıkmaktadır. Şekil II.5'de görüldüğü gibi daimi mıknatıslı, değişken relüktanslı ve hibrid türde adım motorları mevcuttur. Ayrıca doğrusal (Liner) adım motorlarda kullanılmaktadır[3].



Şekil II.5 a) Daimi mıknatıslı adım motor b) Değişken relüktanslı adım motor  
c) Hibrid adım motor yapısı

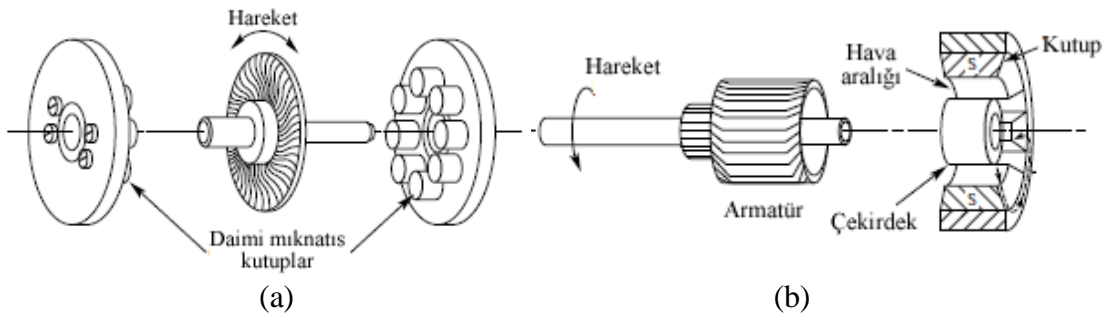
### II.2.1.2 Fırçalı DC motorlar

DC motorların tarihçesi 1830'lara kadar uzanmaktadır. AC sistemlerin gelişmesiyle popülerliğini yitirse de, otomasyon ve kontrol sistemlerinin gelişmesiyle yeniden gündeme gelmiş ve prensipte pek değişmeden günümüze kadar gelmiştir. Çok bilinen indüksiyon kanunu prensiplerine göre çalışan DC motorların en büyük olumsuzluğu bakım gerektirmeleridir. Şekil II.6'da DC motorun yapısı görülmektedir[3].



**Şekil II.6** DC motor yapısı

DC motorlardaki bir diğer önemli etken ise komitasyon olarak bilinen ve basitçe bobinlerdeki akımın yön değiştirmesi olarak tanımlanabilecek olan olaydır. Burada dönme olayını etkileşim içinde bulunan iki manyetik alan (stator ve armatür manyetik alanı) gerçekleştirir. Daimi bir moment elde etmek için bu iki alan arasındaki ilişkide daimi ve uyumlu olmalıdır. Bu durum armatürün yapısındaki kolektör dilimlerinin yapısıyla gerçekleştirilir. Komitasyon diğer bir deyişle değişim etkisi akımın armatür içerisinde dolaşımını sağlar ve armatürün konumunu düzenler. Bu sayede armatür milinden daimi tork üretimi sağlanır. Günümüzde birkaç DC motor tipi kullanılmaktadır. Statorunda daimi mıknatıs olan tür en çok kullanılan olmakla beraber fırçalı yapısı ve demir kayıpları yüzünden pek verimli değildir. Demir kaybını önlemek için farklı yapılarda DC motorlar geliştirilmiştir. Disk ve kabuk şeklinde armatürlü motorlar bunlardan en çok bilinenleridir. Şekil II.7’de disk ve kabuk şeklinde armatürlü motor yapısı görülmektedir[3].



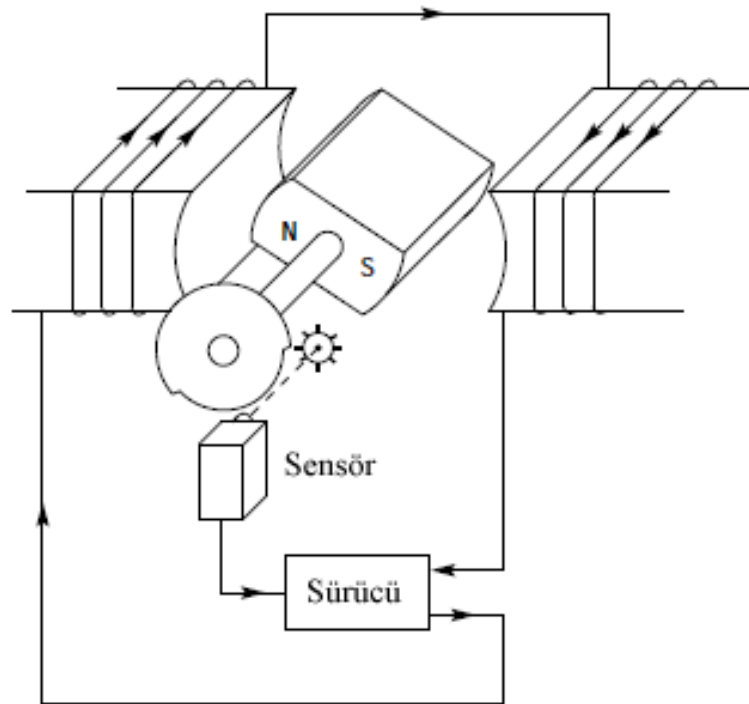
**Şekil II.7 a)** Disk (Printed) tip DC motor **b)** Kabuk (Shell) tipi DC motor

DC motorlarda kayıpları en aza indirmek için bazı geliřtirmeler yapılmıřtır. Bunlardan en önemli yapının ters yüz edilmesiyle fırçasız DC motorların üretilmesidir. Bu sayede sabit mıknatıs armatür ile sürtünme kayıpları azaltılmıř, fırçalardaki kayıplar engellenmiř ve ayrıca demir kayıpları da azaltılmıřtır. Bunların yanında fırçasız motorlarda kullanılan daimi mıknatıslar daha güçlü ve dolayısıyla daha pahalı olması ve sürücülerin fırçalı motorlara göre daha kompleks olmasından dolayı maliyetleri daha yüksektir.

### II.2.1.3 Fırçasız Motorlar

Fırçasız motor ismi altında birçok motor incelenebilir. Örneğın, adım motor AC indüksiyon ile çalıştığı için fırçasız bir motordur. Fırçasız motor başlığı altında; DC fırçalı motorların performansına benzer bir performans sergileyen, bunun yanında fırçalar ve kolektörün olumsuz etkilerinden arındırılmıř motorlar incelenecektir.

Fırçalı DC motorda sargılar armatür üzerinde olduđu için fırçalara ve kolektörlere ihtiyaç duyulur. Fırçasız motorlarda bu ihtiyaç tam tersi bir tasarım ile giderilmiřtir. Armatür yani dönen kısım sabit mıknatısla yapılmıř ve sargılar statora yani sabit olan kısma sarılmıřtır. Bu sayede hem fırça ve kolektörlere olan ihtiyaç ortadan kalkmıř hem de kayıplar azaltılmıřtır. řekil II.8'da fırçasız motor yapısı verilmektedir [3].

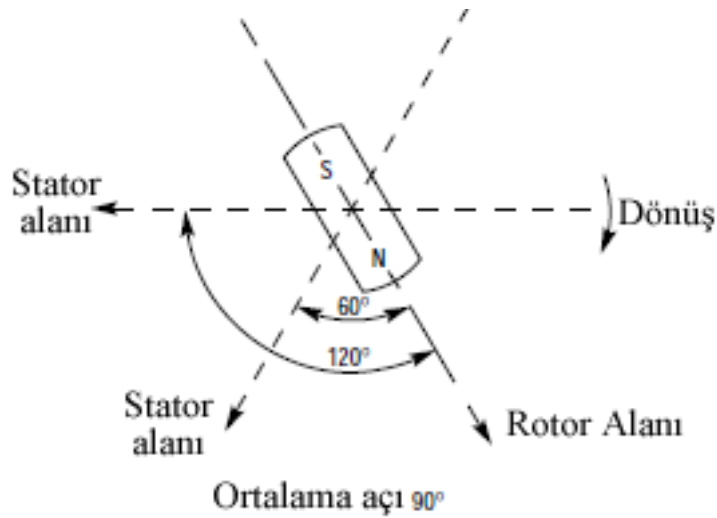


řekil II.8 Fırçasız motor yapısı

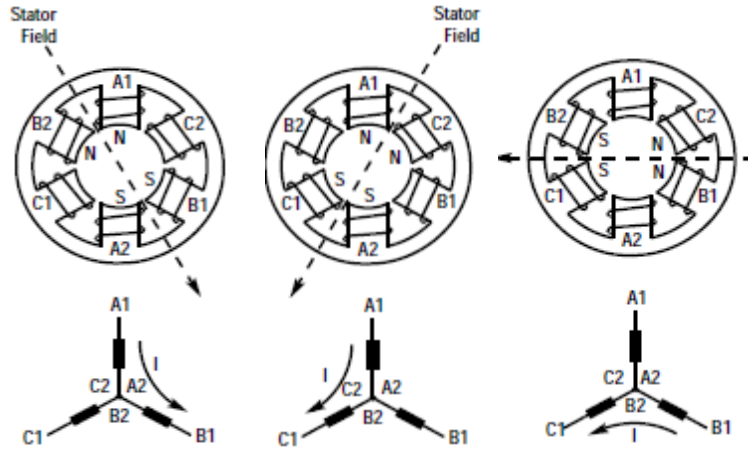
Fırçasız motorlarda, fırçalı DC motorlarda akım yönünü değiştiren kolektör ve fırçaların görevini elektronik bir sürücü devralmıştır. Burada sensör armatürün konumunu sürücüye bildirerek komitasyon için gerekli bilgiyi sağlar. Motor DC gerilimle besleniyor gibi görünse de, elektronik sürücü sayesinde sürekli yön değiştirdiği için motor AC akımla sürülmüş olur [4].

Fırçalı DC motorlarda bobinler endüvide bulunduğu için moment üretimi değişken olacaktır ve bu karakteristik sinüzoidaldir. Endüvi alanı stator alanına dik konumda iken maksimum moment, komitasyon noktalarında ise hiç moment üretilmeyecektir. Pratikte bunu engellemek için endüvideki bobinler kolektör üzerinden birbirileri ile ilişkilendirilmiştir. Bu sayede gerekli moment maksimum moment düzeyine en yakın bobin tarafından sağlanır. Böylece tork dalgalanması en aza indirgenmiş olur. Bu işleme benzer bir işlemi fırçasız motorlarda gerçekleştirmek için stator çevresine dağıtılmış çok sayıda bobin gereklidir, bu durum birçok bakımdan engelleyici bir unsurdur. Bu yüzden fırçasız motorlarda stator sargıları genelde iki ya da üç bobin grubundan oluşur. Rotor ise genellikle dört veya altı kutupludur. Şekil II.10'da iki kutuplu üç fazlı bir dizayn görülmektedir [3].

Daha önce belirtildiği gibi en yüksek torku sağlamak için stator ve rotor alanları birbirine dik konumda yani aralarında  $90^\circ$  olmalıdır. Burada motor üç fazla sınırlandırılmış olduğundan stator alanında sadece  $60^\circ$ 'lik değişim yapılabilir. Bu nedenle  $90^\circ$ 'lik moment elde edilmesi mümkün değildir ama  $60^\circ$  ile  $120^\circ$  arasında çalıştırılarak ortalama bir değer elde edilebilir. Bu işlem moment açısı  $60^\circ$  düştüğünde stator alanının değişimi ile açının  $120^\circ$  yükseltilmesiyle gerçekleştirilir[4].



Şekil II.9 Farklı faz akımları için stator alanı konumu



**Şekil II.10** Komitasyon noktalarında rotorun konumu

#### II.2.1.4 Hibrid Servo

Adım motorlar ve fırçasız servo motorlar temelde birbirine benzedikleri için ufak değişikliklerle birbirlerinin yerine kullanılabilirler. Aradaki en büyük fark kutup sayıları ve adım motorunda geri besleme elemanının olmamasıdır. Adım motorunda kutup sayısı 50'ye kadar çıkabilirken, fırçasız servoda iki veya üçtür. Bu sebeple fırçasız servonun adım motoru olarak kullanılması pek kullanışlı olmaz. Adım motoruna bir geri besleme elemanı eklenerek fırçasız servo olarak kullanılması daha iyi sonuçlar verebilir. İşte bu tür motorlar hibrid (Hybrid) servo olarak bilinir [3].

Hibrid servolar fırçasız servolarla aynı yöntemlerle sürülebilir. Fakat yüksek hızda demir kayıpları yüksektir. Maliyeti step motora göre pahalı fırçasız servolara göre ucuzdur. Sessiz ve ısınmadan çalışan melez servolar, düşük hızlarda step motor özelliğinden dolayı oldukça yumuşak ve düzgün çalışırlar.

#### II.2.1.5 Direkt sürücülü motorlar (Direct Drive)

Direkt sürücü motorlarda yük direkt olarak motora bağlıdır. Herhangi kayışlı veya dişli sistem kullanılmaz. Bazı durumlarda fırçalı ve fırçasız servo motorlar gerekli moment ve çözünürlüğe yanıt veremediğinde kullanılırlar. Küçük yapıları ve step motor gibi kolay kullanımları ile birçok performans sorununu çözebilirler. Hassasiyet, hızlı tepki zamanı, yüksek hızda yüksek moment, ayarlanabilirlik ve yumuşak çalışma gibi üstün özellikleri vardır[3].

## II.2.2 Komitasyon

DC motorlarda akım bir bobinden diğerine fırçalar ve kolektörler aracılığı ile geçer. Rotorun dönüşü ile paralel olarak işleyen bu süreçte amaç en iyi momenti elde etmek için kutup alanı ile rotor manyetik alanı arasındaki ideal açığı korumaktır. Fırçasız motorlarda ise kolektör ve fırçalar bulunmadığı için bu işlemin sürücü tarafından elektronik olarak gerçekleştirilmesi gerekir. Bu olaya elektronik komitasyon denir[4].

Sabit mıknatıs sargı alan çizgileri boyunca yönlendirilmişse, bu duruma kararlı hal denir ve sargılar mıknatıs üzerine hiçbir güç uygulamaz. Eğer mıknatıs sargı alan çizgileri boyunca sıralanmamış ise sıralanmak için girişimde bulunacaktır. Matematiksel olarak bu ifade Denklem I.1'de verilmiştir.

$$T = K_e I \cos \theta \quad (I.1)$$

$\theta$  : sargı alanı ile sabit mıknatıs arasındaki açı ( $^{\circ}$ )

$K_e$  : mıknatıs gücüne bağlı sabit,

$T_m$  : Motor sargılarının oluşturduğu dönme momenti (Nm)

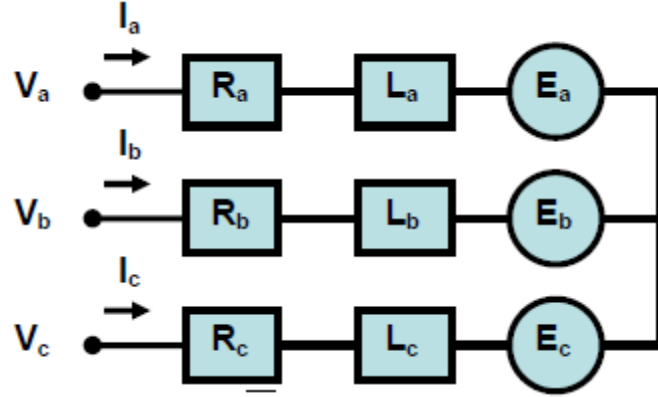
$I$  : Motor akımı (A)

Eğer mıknatıs hareket edebiliyorsa sargı alanı ile aynı dizilime gelene kadar dönecek ve o pozisyonda sabit kalacaktır. Sabit mıknatıslı ve birden fazla sargılı motorlarda sargılar farklı doğrultularda alanlar oluştururlar. Bu sargıları sıra ile enerjilendirerek döner bir manyetik alan oluşturulur ve bu sayede rotor tahrik edilir. Sargıların bu şekilde sıra ile enerjilenmesine komitasyon denir[5].

Fırçalı DC motorlarda hangi sargının ne zaman enerjileneceğini belirleyen mekanik parçalar vardır. Fırçasızlarda ise bu mekanizma elektrondur. Komitasyon işlemi genel olarak iki farklı yöntemle yapılabilir; adımlayıcı (stepper) ve fırçasız DC. Adımlayıcı komitasyon yönteminde sargı alanı ile rotor pozisyonu arasında  $90^{\circ}$  vardır ve komitasyon cihazı rotor pozisyonunu bilmelidir. Fırçasız komitasyonun bu yöntemle göre en büyük avantajı ise aynı motor akımında en yüksek torkun elde edilmesidir. Ayrıca daha düzgün ve yumuşak bir çalışmanın yanında torkun kontrol edilmesini de olanak sağlar. Servo uygulamalar için daha uygun olan bu yöntemde gerçek zamanlı hesaplamalara ve yüksek çözünürlüklü pozisyon sensörlerine ihtiyaç vardır[5].

### II.2.2.1 Fırçasız Motorda Komitasyon

Fırçasız bir motorda yıldız veya üçgen bağlı üç sargı ve rotorda sabit mıknatıs bulunur. Genellikle 2,4,6 ve 8 kutuplu olmakla beraber daha yüksek kutup sayısına sahip motorlarda mevcuttur. Kutup sayısı motorun bir mekanik turdaki elektriksel döngüsünü belirtir. 2 kutupluda bir turda tek elektriksel döngü ve 4 kutupluda iki elektriksel döngü gerçekleşir. Şekil II.11'de fırçasız motorun elektriksel eşdeğer devresi görülmektedir[5].



Şekil II.11 Fırçasız motor eşdeğer devresi

$R_a, R_b, R_c$  : Stator sargı direnci ( $\Omega$ )

$L_a, L_b, L_c$  : Stator sargı endüktansı (H)

$E_a, E_b, E_c$  : Stator zıt E.M.K. (V)

$I_a, I_b, I_c$  : Stator sargı akımı (A)

Tipik üç fazlı bir fırçasız motorda her faza ait sargılar simetriktir ve sargılarda oluşan zıt E.M.K.'lar arasında  $120^\circ$  faz farkı mevcuttur. Stator sargılarına verilen 3 fazlı sinüzoidal akım ile motorun sürülmesinde en yüksek momenti almak için rotor manyetik alanı ile stator sargılarının oluşturduğu alan arasındaki açının takip edilip düzenlenmesi gerekir[3].

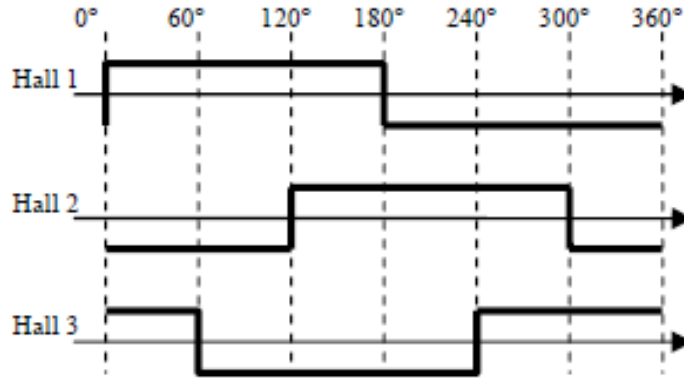
Bunu gerçekleştirmek için rotor manyetik alanının pozisyonu bilmek gerekir. Bu bilgi genel olarak iki farklı tip komitasyon tekniği ile elde edilir. Kare ve trapez dalga komitasyonu eski teknikler olmakla beraber halen kullanılmaktadır. Fakat yeni sürücüler sinüzoidal komitasyon ve değişken vektörlü (flux-vector) gibi daha gelişmiş yüksek verimlilikli farklı teknikler kullanılmaktadır[4,5].

### II.2.2.2 Trapezoidal Komitasyon

Altı adımlı (6-step) ve Fırçasız DC komitasyon olarak da bilinen bu yöntem daha çok pozisyon kontrolünün gerekmediği hız veya torkun kontrol edildiği düşük maliyetli uygulamalarında kullanılır. Bu komitasyon tekniğinde rotor ve stator alanları arasındaki  $90^\circ$  fark sürekli olarak kontrol edilmek yerine tork üretimindeki değişimler takip edilir. Rotorun elektriksel pozisyonu  $60^\circ$ 'lik bölümlere ayrılmıştır ve bu  $360^\circ$ 'lik bir turda altı farklı pozisyon anlamına gelir. Bu sayede  $90^\circ$ 'lik ideal açılı farkı  $\pm 30^\circ$ 'lik bir farkla sağlanmış olur. Bu fark torkta %13'lük bir değişime tekabül eder. Bu pozisyon ölçümü motorun içerisine yerleştirilmiş üç adet Hall sensörü ile yapılabilir. Bu üç sensör rotor manyetik alanının konumu belirleyerek aşağıdaki grafikte görüldüğü gibi rotor pozisyonunu kabaca belirler. Sensör çıkışlarındaki bilgi 0 ve 1 ile ifade edildiğinde rotorun pozisyonunu her  $60^\circ$  için belirten 3 bitlik değerler elde etmiş oluruz. Tablo II.1'de sensör çıkış bilgilerinin binary karşılıkları ve Şekil II.12'de de bu üç bitlik verilerin rotor pozisyonuna göre karşılıkları gösterilmiştir[5].

**Tablo II.1** Pozisyon bilgisi bit değerleri

Segment	Hall States
0- $60^\circ$	101
60- $120^\circ$	100
120- $180^\circ$	110
180- $240^\circ$	010
240- $300^\circ$	011
300- $360^\circ$	001



**Şekil II.12** Hall sensörü ile pozisyon ölçümü

En iyi torku elde etmek için Hall sensörleri motorun zıt E.M.K.'sına göre yerleştirilir. Nitekim zıt E.M.K. rotorun manyetik alanının konumuna göre oluşmaktadır. Kısacası zıt E.M.K.'nin sıfır noktasından geçişlerinde Hall sensörlerinin çıkış bilgisi değişir. Hall sensörlerinden alınan bilgiler doğrultusunda sürücü iki faz üzerinden motoru sürerken üçüncüyü boşa bırakır. Rotor pozisyonuna göre hangi iki fazın akımlarının devrede olacağına karar verilir. Tablo II.2'de rotor konumuna göre hangi faz çiftinin devrede olacağı verilmiştir[4,5].

**Tablo II.2** Rotor pozisyonuna göre faz çiftleri

Hall State	Phase Pair
100	A→B
101	C→B
001	C→A
011	B→A
010	B→C
110	A→C

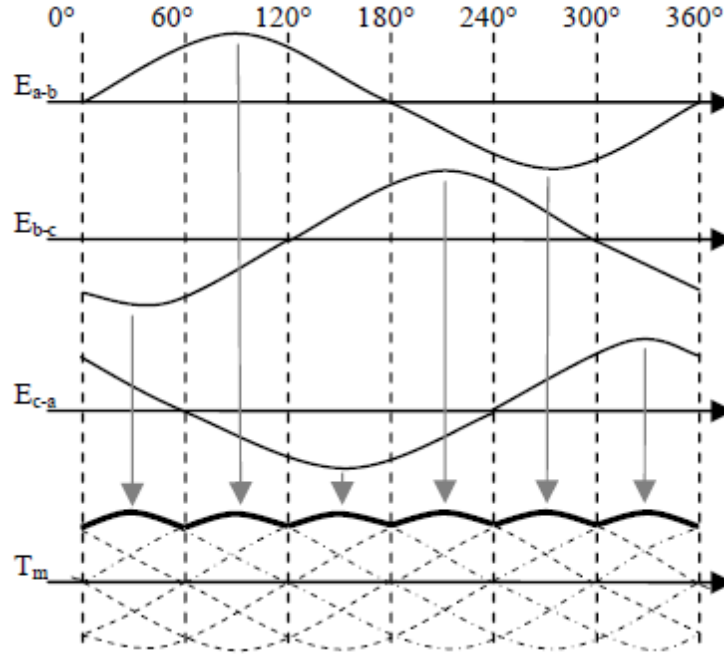
Tablo II.2'de görüldüğü gibi faz çiftlerinde her zaman bir faz ortaktır. Bu sayede akım sürekliliği sağlanmaktadır. Moment üretimini daha iyi ifade edebilmek için elektromanyetik kayıpları ihmal edilerek enerji dönüşümünü gözlemlemek gerekir. Denklem I.2 de bu dönüşüm matematiksel ifadesi verilmiştir.

$$E_{f-f} \cdot I_m = T_m \cdot \omega_m \quad (I.2)$$

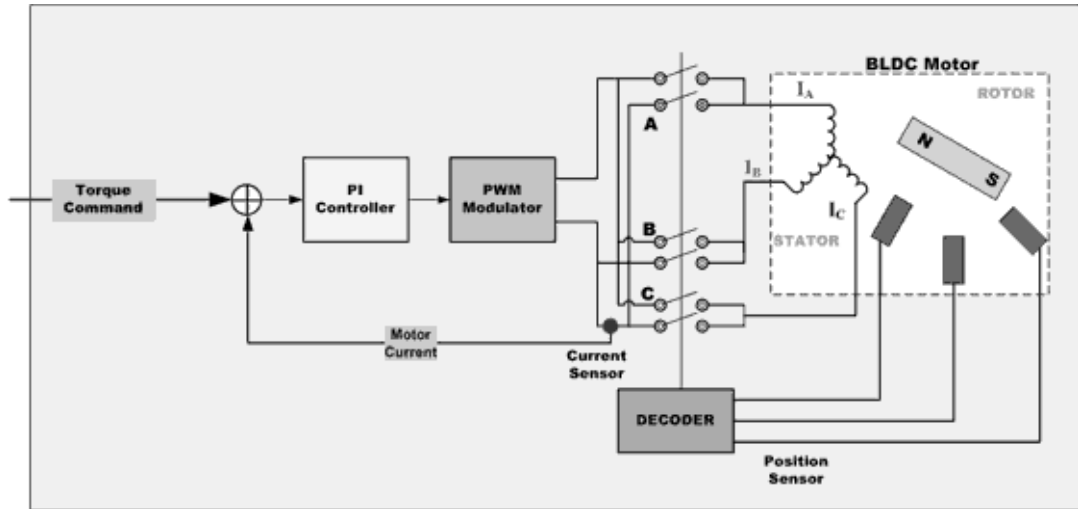
$E_{f-f}$  : İki faz arası zıt E.M.K (V)

$I_m$  : Faz akımı (A)

Sabit faz akımında motorun hızının da sabit olduğu kabul edilirse motorun ürettiği moment Şekil II.13'de ki grafikte olduğu gibidir. Grafikte görüldüğü gibi çıkış momentinde hafif bir dalgalanma vardır. Bu dalgalanma daha önce belirtildiği gibi %13'lük bir değişim ile ortalamada en yüksek moment değerinin kabaca %95'ne tekabül eden bir moment üretilmesine neden olur. Bu moment dalgalanması düşük güçlü sistemlerde görmezden gelinbilir. Ayrıca bu tür sistemlerde mantıksal kontrol ünitesi ve akım kontrolü analog olarak yapılabileceğinden sistem kurulumu basit ve maliyeti düşük olacaktır[5].



Şekil II.13 Moment dalgalanması



Şekil II.14 Fırçasız motor trapezoidal sürücü devresi

Şekil II.14’de fırçasız bir motorun trapezoidal kontrolör ile sürülmesine ait temel yapı verilmiştir. Burada motor daima bir çift bobin üzerinden sürülürken üçüncü bobin boşta. Motor içerisine yerleştirilmiş olan hall sensörleri 60°’lik parçalar halinde rotor konumunu belirten sinyaller üretmektedir. Bu sayede rotorun pozisyonuna göre hangi bobin çiftinin enerjileneceğine karar verilmiş olur. Motorun bir turundaki hareket altı farklı bölümde ele alınarak sistemden alınan moment değeri maksimize

edilirken sürekli ve kararlı çalışma sağlanmış olur. Geri beslemede veya kontrol ünitesindeki herhangi uyumsuzluk sistemin moment üretimini olumsuz etkileyeceği için sistemin kullanılacağı uygulamaya uygun geri besleme elemanı ve kontrol ünitesi tercih edilmelidir[6].

### II.2.2.3 Sinüzoidal Komitasyon

Sinüzoidal komitasyon için yeterli rotor pozisyonu bilgisi gereklidir. Bu da trapezoidal komitasyondaki pozisyon bilgisinden daha yüksek çözünürlüklerde mümkündür. Nitekim burada faz akımı sinüzoidal olduğundan buna uygun olarak rotor hareketinin de mutlak pozisyonunun bilinmesi gerekir. Fakat mutlak pozisyon için gerekli geri besleme maliyeti nedeni ile her zaman uygulanabilir bir yöntem değildir. Bunun yerine rotorun mutlak pozisyonunu belirlemek için iki farklı yöntem daha mevcuttur[5].

Bu yöntemlerden ilki başlangıçta kabaca elde edilen bir mutlak pozisyon bilgisi üzerine kullanılan artımlı geri besleme elemanı ile mümkündür. Burada yüksek çözünürlükte geri besleme bilgisi elde edilmiş olur. Diğer yöntem ise özel bir faz belirleme tekniği ile artımlı bir geri besleme elemanı üzerinden rotorun elektriksel açısının belirlenmesidir. İlk yöntem için hall sensörleri ve artımlı bir encoder yeterlidir. İkinci yöntem içinse başlangıçta izin verilecek kontrolsüz bir harekete ihtiyaç vardır. Yani motor başlangıçta geri besleme bilgisinin elde edilebilmesi için kontrolsüz çalışmalıdır. Trapezoidal komitasyon özellikle düşük hızlarda doğruluk ve yumuşak sürekli çalışmada uygunsuz sonuçlar verebilir. Sinüzoidal komitasyonda, fırçasız motor üç fazlı sinüzoidal akımla sürüldüğü için bu sorunlar giderilmiş olur. 3 fazlı sinüzoidal akım ve zıt E.M.K. ile motor momenti Denklem I.3'de verilmiştir[5,6].

$$T_m = N_p * (E_a * I_a + E_b * I_b + E_c * I_c) / \omega_m \quad (I.3a)$$

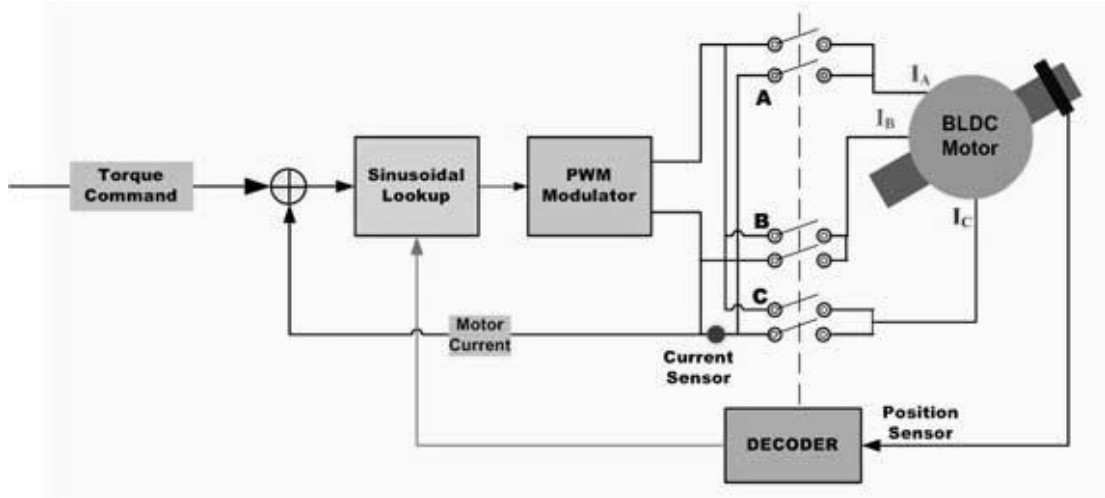
$$T_m = K_{t,p} * (\sin(\theta) * \sin(\theta) + \sin(\theta + \frac{2\pi}{3}) * \sin(\theta + \frac{2\pi}{3}) + \sin(\theta - \frac{2\pi}{3}) * \sin(\theta - \frac{2\pi}{3})) \quad (I.3b)$$

$$T_m = K_{t,p} * (\sin^2(\theta) + \sin^2(\theta + \frac{2\pi}{3}) + \sin^2(\theta - \frac{2\pi}{3})) \quad (I.3c)$$

$$T_m = K_{t,p} * \frac{3}{2} \quad (I.3d)$$

- $K_{t,p}$  : Moment sabiti  
 $I_{a-b-c}$  : Stator sargı akımları (A)  
 $N_p$  : Motor kutup sayısı

Denklem I.3a'de elektriksel güçten mekaniksel güce çevrim yer almaktadır.  $K_{t,p}$  her faz için moment sabitidir. Akım ve zıt E.M.K. sinüzoidal ve  $120^\circ$  faz farklıdır. Bu sayede moment daima sabittir. Sinüzoidal komitasyonla fırçasız motor üç faz ile sorunsuz sürülür. Bu işlemi rotor pozisyonuna bağlı olarak ilgili faz akımını seçerek yapar. Bu sayede moment dalgalanmaları engellenmiş olur. Şekil II.15'de fırçasız motor sinüzoidal sürücü devresi görülmektedir. [6].



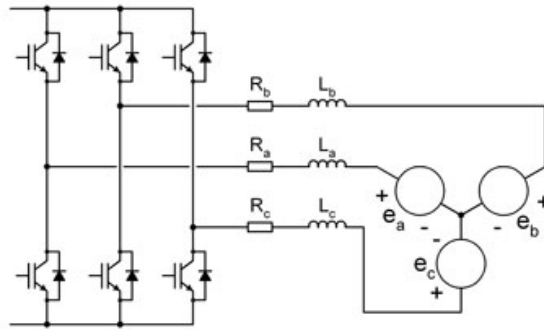
Şekil II.15 Fırçasız motor sinüzoidal sürücü devresi

Statordaki sargılar birbirine  $120^\circ$  açı farkı ile yerleştirilmiş olduğundan sargılardaki akımlarında sinüzoidal ve birbirinden  $120^\circ$  açı farklı olmalıdır. Bu eşleşme ise encoder tarafından sağlanan konum bilgisi ile yapılır. Bu sinyaller istenen moment değerinin sağlanması için moment komutu ile çarpılarak sinüs dalganın genliği belirlenir. Sinüzoidal akım komut sinyali PI (*proportional-integral controller*) kontrolör sayesinde iki motor sargısındaki akımı düzenler. Üçüncü sargıdaki akım ise kontrol edilen sargılardaki akımların negatif yönlü toplamına eşit olduğundan ayrı olarak kontrol edilemez. PI kontrolörlerin her birinin çıkışı PWM modülatörünü ve devamında çıkış köprüsünü besler [6].

Sinüzoidal komitasyonla elde edilen düzgün ve yumuşak kontrol trapezoidal komitasyon ile genellikle elde edilemez. Buna rağmen düşük hızlarda oldukça etkili olan sinüzoidal komitasyon yüksek hızlarda sorun oluşturabilir. Bunun nedeni hız arttıkça kontrol döngüsünün sinüzoidal sinyali aynı oranda hızlı yani yüksek frekansta tutması gerekmesidir. Bu aynı zamanda zıt E.M.K.'nın da genlik ve frekans bakımından yükselmesi demektir. PI kontrolörlerin kazanç ve frekans aralığı sınırlığı olduğundan akım kontrol döngüsündeki zaman değişkeni kararsızlığı faz gecikmelerine ve kazanç hatalarına neden olacaktır. Bunun sonucu olarak rotor ideal moment üretim aralığından çıkacak ve daha az moment üretilecektir. Bir noktadan sonrada motor moment üretimi sifıra düşecektir[6].

#### II.2.2.4 Trapezoidal ve Sinüzoidal komitasyonun karşılaştırılması

Hem trapezoidal (Six-step) hem de sinüsoidal (Continuous) komitasyonla sürülen motorlar genellikle statorunda yıldız bağlı üç sargı bulunan ve rotoru daimi mıknatıslı motorlardır. Her iki yöntemde de motor sargıları üç fazlı gerilim kaynaklı (VSI) bir invertör tarafından kontrol edilir. Şekil II.16 gerilim kaynaklı invertör ile sabit mıknatıslı motor sürme devresi verilmiştir[4-6].

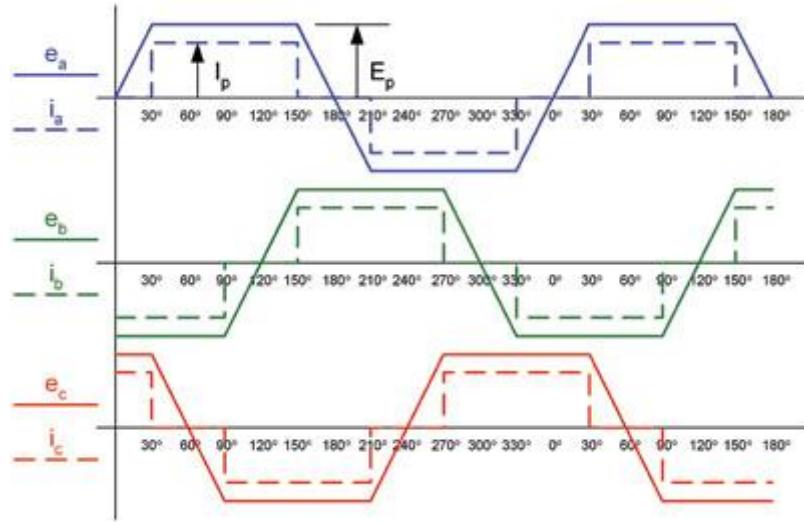


Şekil II.16 Gerilim kaynaklı invertör ile sabit mıknatıslı motor sürülmesi

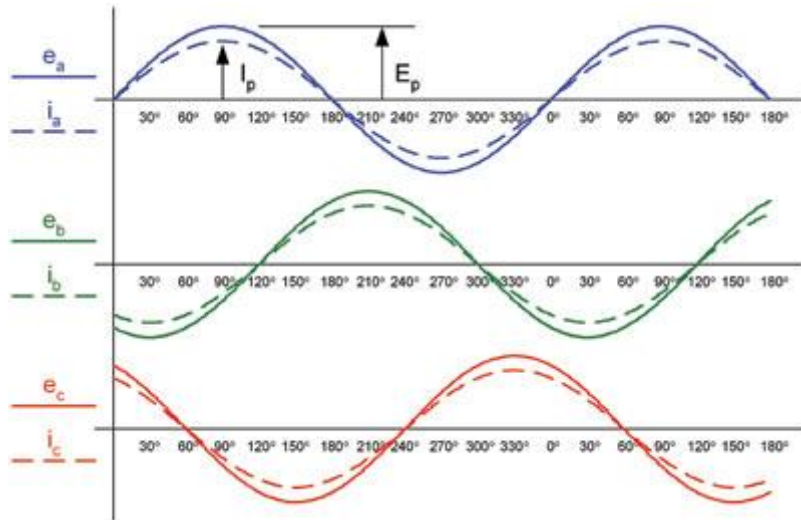
Gerilim uygulanan her fazın akımları her iki yöntemde de darbe genliği modülatörü (PWM) ile kontrol edilir. Bu sebeple fırçasız DC ve sabit mıknatıslı AC motorların birçok benzerlikleri vardır. Yüzeysel bir gözlemlerle ikisi arasında yapısal olarak bir fark olmayabilir. Her iki motor arasındaki asıl fark moment işlevi ve akımın sargılarda sürülüş biçimidir. Her iki motorda dışarıdan tahrik edilerek zıt E.M.K. değerleri osilaskop ile ölçüldüğünde fırçasız DC (BLDC) motorun düz zirveli trapezoidal ve daimi mıknatıslı AC (PMAC) motorun ise sinüzoidal dalga şekilli

olduğu görülür. Düzgün ve sürekli bir çalışma için motora uygulanan akımında zıt E.M.K. ile aynı dalga şekline sahip olması gerekir[5,6].

Fırçasız servo motorda akım sadece iki sargıya uygulanır. Bir komitasyon aralığında akım bir sargıya verilir ve diğerinden geri döner. Üçüncü sargı ise boşta. Motor mili döndükçe mümkün altı kombinasyona göre sargılar enerjilenir. Bu yöntemin en önemli özelliği herhangi bir anda altı inverter anahtarından sadece ikisi devrededir ve akımı sadece iki motor sargısı taşır. Şekil II.17’de trapezoidal komitasyonda üç faza ait dalga şekilleri görülmektedir[6].



Şekil II.17 Trapezoidal komitasyon



Şekil II.18 Sinüzoidal komitasyon

Diğer yandan sinüzoidal komitasyonla sürülen PMAC motorda her faz inverteri sürekli modülasyon (PWM) ile her sargıyı sinüzoidal akımla sürer. Uygulama açısından trapezoidal ve sinüzoidal komitasyonun en büyük fark, sinüzoidal komitasyonla herhangi konumda veya hızda alınan mutlak moment değeridir. Şekil II.18’de sinüzoidal komitasyonda üç faza ait dalga şekilleri görülmektedir[5,6].

BLDC motorlardaki komitasyondan kaynaklanan moment dengesizliği miknatıssal büzülme ile birlikte statorda oluşan akustik komitasyon parazitlerinin bir sonucudur. Bu etkiler sinüzoidal komitasyon ile sürülen PMAC motorlarsa görülmez. Bazı PWM parazit sinyalleri dışında pürüzsüz ve gürültüsüz bir çalışma sergilerler[6].

### II.2.3 Sürücü devreleri

Günümüzde artık birçok elektrik makinesi sürücü devreleri ile optimize edilmiş sistemlerde kendilerine uygulama alanı bulmaktadır. Bu gelişimde yarı iletken teknolojisinde gelinilen noktanın önemi büyüktür. Yarı iletkenler genel olarak iki çeşit negatif ve pozitif madde olarak bilinen silisyum ve germanyumun alaşımlarından meydana gelmektedir. Bu iki maddenin farklı dizilimleri ile yarı iletken malzemeler oluşturulur. Temelde bu yarı iletken malzemelerin görevi hızlıca iletim ve kesime giderek devre akımını kesmek ve iletmektir.

Sürücü devrelerinin gelişmesi tristör, triyak, GTO, BJT, MOSFET ve IGBT gibi yarı iletken anahtarlama elemanlarının gelişmesiyle olmuştur. Tüm bu yarı iletkenlerin kendine has özellikleri farklı uygulamalarda kullanılmasına olanak sağlamıştır. Tablo II.1 yarı iletken karakteristikleri verilmiştir.

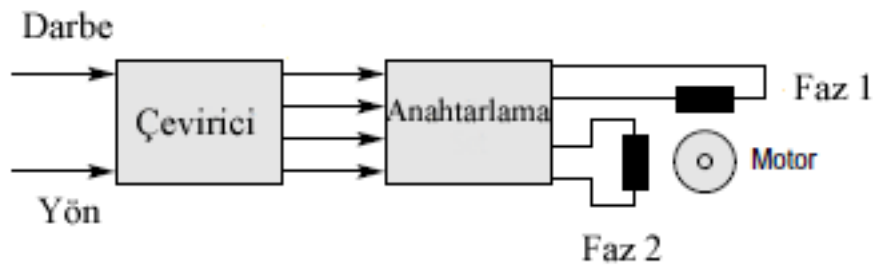
**Tablo II.1.** Yarı iletken karakteristikleri

	Tristör	GTO	BJT	MOSFET	IGBT
Gerilim (V)	4000	4000	1000	600	1500
Akım (A)	3000	3000	500	50	1000
Frekans (KHz)	1	2	30	1000	30
Gerilim Düşümü (V)	1,5-2,5	1,5-2,5	2-3	3-4	2-3
Kesim süresi (µs)	10-100	5-10	<2	<0,1	<2

Yarı iletkenler inverterlerde gerilimin dalga şeklini, genliğini ve frekansını değiştirmek için kullanılırlar. İnverterlerde DC kaynaktan alınan doğru gerilim istenilen dalga şeklinde ve frekansta üretilebilir. Genel olarak gerilim ve akım kaynaklı olmak üzere iki çeşit inverter mevcuttur. Gerilim kaynaklı inverter doğru gerilim kaynağından alınan gerilimi kare dalga gerilime, akım kaynaklı ise doğru akım kaynağından aldığı akımı kare dalga akıma dönüştürür. Akım kaynaklı inverterler daha iyi yükselme ve düşüş karakteristiğine sahip olmasına rağmen sadece özel yüklerde kullanılabilir. Bu yüzden gerilim kaynaklı inverterler daha çok kullanılmaktadır[4-7].

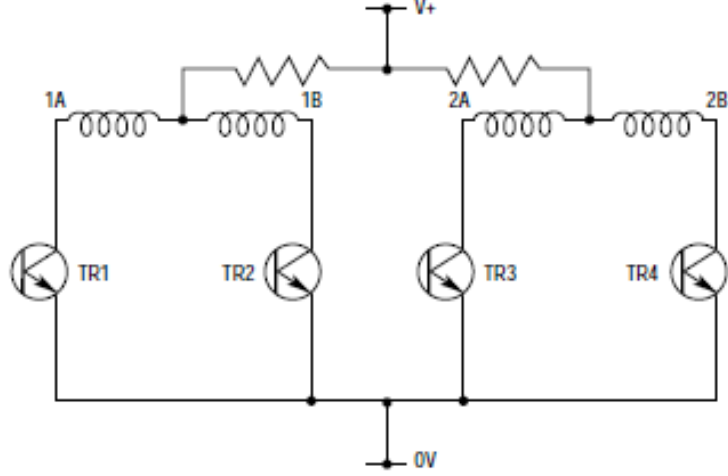
### II.2.3.1 Adım motor sürücüleri

Adım motor sürücüleri kontrol sisteminden aldığı düşük seviyeli sinyallerle motora elektrik enerjisi sağlar. Motorun çalışmasının arkasında sargılardan sırasıyla geçen akımların oluşturduğu manyeto motor kuvvet vardır. Sürücü burada akım kaynağı olarak davranır ve uygulanan gerilim sadece kontrol akımının ayarlanmasında önemlidir. Adım motor sürücüsüne giren sinyaller adım darbeleri ve yön sinyalidir. Motorun her bir adım hareketi için bir adım darbesi girilir. Motorun bir turu için sürücü 200 ila 101,600 arası darbe üretir. Endüstride en çok kullanılan adım motor sürücü tipi motorun bir turu için 400 darbe üreten yarım adım metodudur. Motor dakikada 1800 d/dk ile dönerken darbelerin frekansı 20 kHz'dir. Adım sayısı arttıkça darbe frekansı da artacağına sürücünü bunu karşılayabilecek kapasitede olmalıdır. Adım motor sürücü bileşenleri Şekil II.19'da görülmektedir[3].



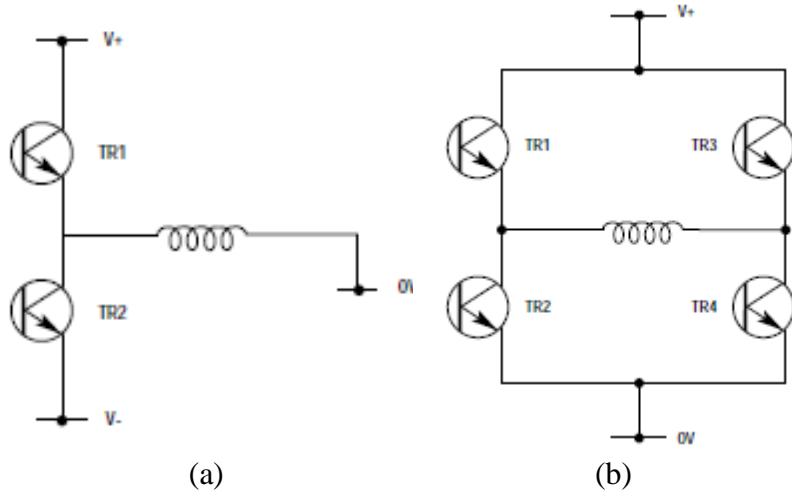
Şekil II.19. Adım motor sürücü bileşenleri

Sürücünün mantıksal çevirici kısmı adım ve yön bilgilerini kontrol sinyallerine çevirerek anahtarlama elemanlarına iletir. Çevirici kısmı her sürücüde hemen hemen aynı olmakla beraber performansı belirleyen asıl etken anahtarlama sistemidir. En basit anahtarlama tekniği Şekil II.20'de görüldüğü gibi tek kutuplu düzendir.



**Şekil II.20** Basit tek kutuplu sürücü devresi

Tek kutuplu sürücü devresinin olumsuz yönü motordaki sargıların tamamını kullanmakta yetersiz kalmasıdır. Sargıların sadece yarısından geçen akım performans düşümüne neden olmaktadır. Sargıların diğer yarısını da devreye alarak aynı güç tüketimi ile performansı arttırmak mümkündür. Bunu başarmak için Şekil II.21.b'de görüldüğü gibi çift kutuplu sürücü devresi kullanmak daha iyi bir çözüm olabilir. Bu sayede bobindeki akımın yönü değiştirilirken bobinin tamamı kullanılmış olur[3].



**Şekil II.21 a)** Basit çift kutuplu sürücü **b)** Çift kutuplu köprü

### II.2.3.2 Mini adımlı sürücüler (Micro-stepping)

Basit motor adımları daha küçük parçalara bölünmek suretiyle sargılardaki akımın miktarı değiştirilebilir. Motorun çalışması esnasında bu küçük parçaların birbirine eşit aralıklarla ve eşit moment oluşturacak şekilde olması esastır. Bu yöntem özellikle

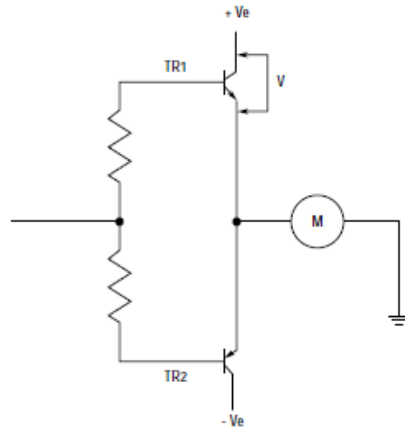
düşük akımlarda sürücüye daha doğru bir kontrol yapmasında yardım olur. Fazlardaki en ufak dengesizlik *microstep* sistemlerde kabul edilemez konum hatalarına neden olur. Bu yüzden darbe genişliği modülasyonu (PWM) hata payını en aza indirmek için genel olarak kullanılır. Sürücü tarafından üretilen faz akımı ardı ardına gelen minik adımların sinüs şeklini alması ile oluşturur. Fakat sinüs dalga şekli her motor için en ideal dalga şekli değildir. Böyle durumlarda akım dolgu (Filled out) ve budaklı (Trimmed) dalga şekillerini alabilir. *Microstep* akım profilleri Şekil II.22’de verilmektedir[3].



Şekil II.22 Microstep akım profilleri

### II.2.3.3 Fırçalı DC Motor Sürücüleri

Motor sürücülerinde doğrusal ve anahtarlamalı olmak üzere iki çeşit yükselteç devresi kullanılır. Doğrusal yükselteçlerde transistörler üzerinde sürekli gerilim olduğunda büyük güçte transistörler ve soğutucu mekanizmalara ihtiyaç vardır. Bu yüzden çok yüksek güçlü sistemlerde pek kullanılmazlar. Fakat elektriksel parazitlerin azlığından dolayı düşük güçlü sistemlerde tercih edilirler. Şekil II.23’de doğrusal servo yükselteç devresi verilmektedir.



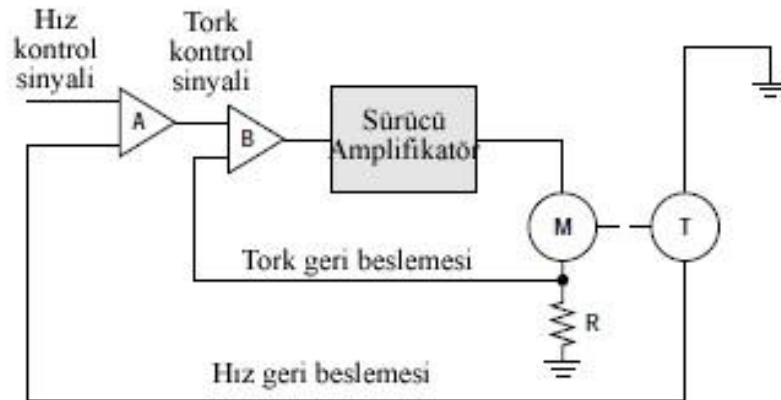
Şekil II.23 Doğrusal servo yükselteç

Anahtarlamalı yükselteçler ise düşük güçlü sistemler haricinde en çok kullanılan yükselteç türüdür. Bu yükselteçlerde çıkış genellikle darbe genişliği modülasyonu (PWM) ile kontrol edilir. Yükselteçteki transistörler kapalı konumdayken uçlarındaki gerilim 1-2 voltu aşmadığı için çok yüksek güçlü sistemlerde dahi rahatlıkla kullanılabilirler. Anahtarlama işlemleri adım motor sürücülerine benzemekle beraber sadece tek anahtar düzeni gereklidir, bu sürücüyü daha basit hale getirir. Buna rağmen DC sürücünün görevi akım değişikliği ile motor momentini kontrol etmektir. Bu işlem için analog ve dijital olmak üzere iki yöntem mevcuttur[3].

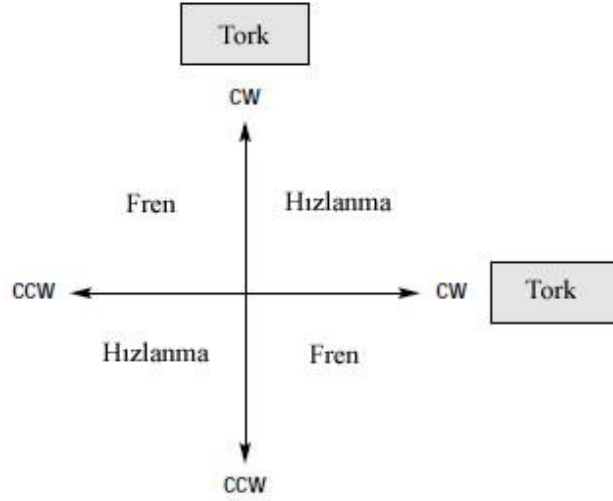
#### II.2.3.4 Analog ve Dijital Servo Sürücüler

Adım motor sürücülerinin aksine fırçalı ve fırçasız servo sürücülerini analog veya dijital olabilir. Geleneksel analog sürücülerde motor hızı +10V ila -10V arası analog bir girişle belirtilmiştir. Analog sürücülerde ayarlar potansiyometre kullanarak yapıldığı için rutin ayarlamalarda zorluk çekilebilir. Dijital sürücülerde ise ayarlamalar bir terminal veya bilgisayar aracılığıyla dijital olarak gönderilir. Bu rutin işlemlerde kesinlik ve kolaylık sağlar. Fırçasız motorların sürülmesinde dijital sürücülerin kullanılması daha uygundur. Encoder ve resolver gibi geri besleme elemanlarında alınan bilgiler dijital bilgilerdir [3,7].

Analog sürücüde hız ve tork bilgisi ölçülerek giriş sinyali ile analog olarak karşılaştırılır. Sürücü amplifikatör bu bilgiler doğrultusunda motorun hızını ve torkunu ayarlar. Sürücü hem hızlanma hem fren torkunu her iki yönde de gerçekleştirebildiği için bu tür amplifikatörler dört bölgeli olarak çalışır. Şekil II.24 analog servo sürücü sistemi, Şekil II.25’de dört bölgeli çalışma görülmektedir.

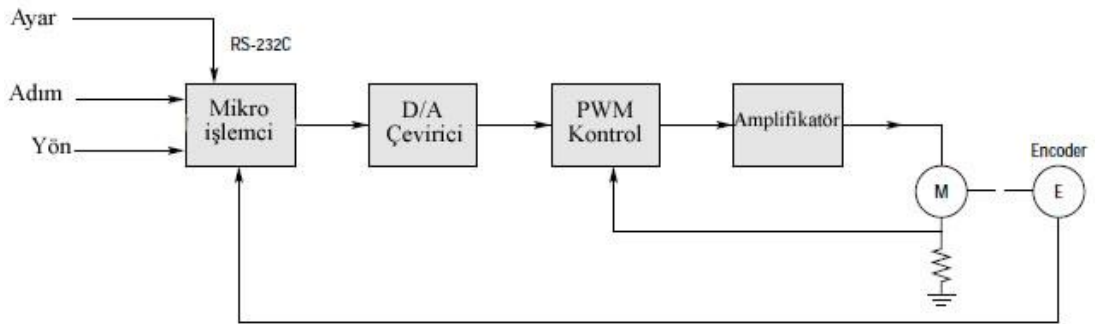


Şekil II.24 Analog servo sürücü sistemi



Şekil II.25 Dört bölge çalışması

Dijital servo sürücülerde tüm işlemler bir mikro işlemci aracılığıyla yürütülür. Mikro işlemciden alınan dijital veriler Dijital-Analog dönüştürücü ile analog veriye dönüştürülür. Dijital sürücü ile analog sürücü aynı işi yapmakla beraber mikro işlemci ile matematiksel algoritmalar kullanarak analog sistemin modellenmesi yapılır. Bu model sistemin davranışlarını tahmin ederek gerekli cevap işaretlerini üreterek sisteme müdahale eder. Dijital servo sistemlerde bu işlemlerin yapılma süresi sistemin performansını doğrudan etkiler. Şekil II.26'da dijital servo sürücü blok şeması verilmiştir[3,7].



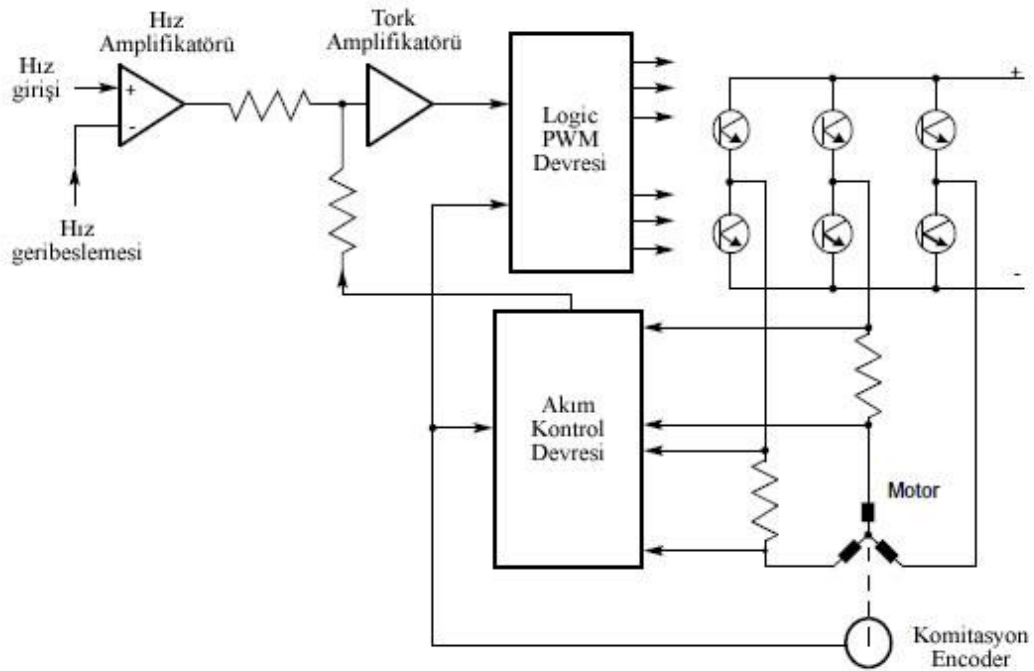
Şekil II.26 Dijital Servo Sürücü

Dijital sürücü sistemlerinde tüm ayarlamalar buton veya sayısal verilerle yapılır. Herhangi ayarlı direnç veya potansiyometre kullanımı gerekmez. Bazı uygulamalarda ani ve yüksek değişimler olduğunda sürücünün sistemde gerekli değişimi yaparak

performansı düzenlemesi gerekir. Dijital sürücülerde rutin işlemlerde veriler yedeklenerek sistemin düzenlenme hızı artırılabilir.

### II.2.3.5 Fırçasız Motor Sürücüleri

Fırçasız motorlarda da genelde iki çeşit sürme metodu kullanılmaktadır. Üç fazlı trapezoidal motor sürücüsünde akım yıldız bağlı sargılar üzerinden istenen anda istenen iki sargı üzerinden geçirilerek motor sürülür. Bu sıralamada motorun konum ve yön bilgisi, mantıksal bir sürücü üzerinden, geri besleme elemanlarıyla okunmasıyla yapılır. Normal bir H-köprüsü üzerinden üç bacaklı bir anahtarlama ünitesi ile sürülen motor sargılarının yıldız noktasının anahtarlama elemanları ile bağlantısı yoktur. Sistemde bir PWM kontrolörü fırçalı DC sürücülerde olduğu gibi akımı kontrol eder. Gerekli akım bilgisi sargılara seri bağlanmış dirençler üzerinden alınır. Buradan elde edilen bilgiler işlenip referans bilgileri ile uyarlanarak, geri besleme elemanlarından alınan veri ile sistemin istenilen şekilde sürülmesine karar verilir. Şekil II.27'de trapezoidal fırçasız servo sürücüsü devresi görülmektedir[3,4].

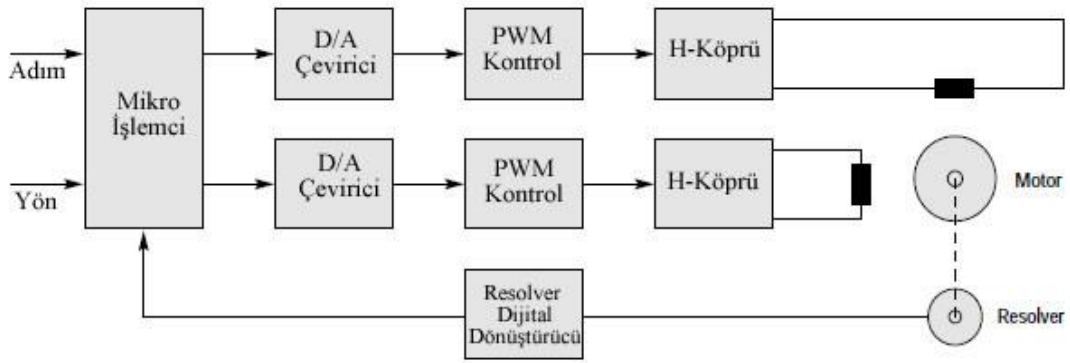


Şekil II.27 Trapezoidal Fırçasız Servo Sürücüsü

Fırçasız servo sürücülerinin giriş kısmı analog fırçalı servo sürücülerle aynıdır. Hız geri besleme bilgisi birçok farklı yöntemle elde edilebilir. Örneğin, bazı tako

jeneratörlerin ürettiği AC gerilim, akım geri besleme bilgisi ile aynı yöntemle elde edilir. Bu sebeple daha sağlıklı ve güvenilir bir geri besleme bilgisi için yüksek çözünürlüklü optik bir encoder çıkış bilgisi analog gerilim bilgisine dönüştürülebilir.

Sinüzoidal sürücüler ise iki yada üç fazlı olabilirler. Şekil II.28’de iki faz sinüzoidal fırçasız servo sürücü blok şeması görülmektedir. İki fazlı sürücü iki bacaklı bir H-köprüsü kullanarak motor sargılarındaki akımı kontrol eder. Trapezoidal sürücünün aksine sinüs dalga ile çalışan sistemlerde mikro işlemci ile alınan adım ve yön bilgileri kullanılır. Burada motor dönerken sargılarda sin ve cos şeklinde akımlar oluşturmamız gereklidir. Bunun için geri besleme bilgileri dijital bilgiye dönüştürülerek mikro işlemciye ulaştırılır. Mikro işlemci bu bilgi ile motorun konumu ve yönüne göre hangi sargıda hangi akımın dolaşması gerektiğine karar verir ve dijital-analog dönüştürücülerle bu bilgiler PWM kıyıcı amplifikatörüne gönderilir[3].



Şekil II.28 İki faz sinüzoidal fırçasız servo sürücü

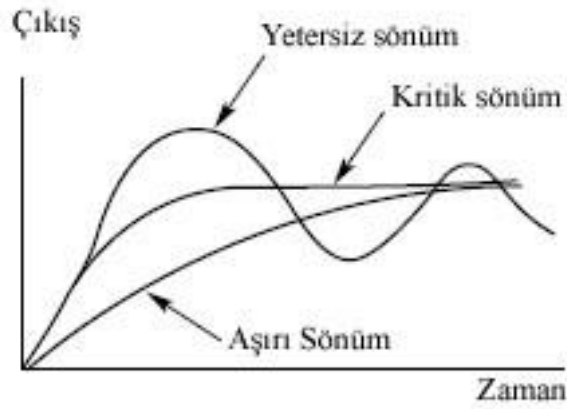
Sinüzoidal sürücülerde komitasyon bilgisi mutlak veya artımlı encoder ile de sağlanabilir. Artımlı encoder daha ucuz bir çözüm olmakla beraber enerjinin verildiği anlarda 90°'lik tork üretimini sağlamak için bazı ayarlamalara ihtiyaç duyacaktır.

#### II.2.4 Servo sistem ayarlamaları

Analog veya dijital kapalı çevrimli servo sisteminde bazı ayarlamalar gereklidir. Ayar işlemlerinde servo sistemin çalışma karakteristiğini belirlediği için giriş sinyallerinin sürekli takip edilmesidir. Servo sistemlerde çıkış ile giriş arasında mutlaka bir fark vardır. Servo çalışmaya başladığında bu farkı, diğer bir ifadeyle hatayı, en aza indirmeye çalışır. Sistemin kazancı (*gain*) hatanın ne derecede düzleneceğine karar verir. Yüksek kazanç en ufak hatalarda bile yüksek düzenleme

momentine neden olabilir. Eğer çıkışta en ufak hata bile önemli ise kazancın yüksek olması beklenir.

Hem servo motorun hem de yükünün eylemsizliği, sistemdeki hızlanma veya durma girişimleri sırasında giriş bilgilerinin amplifikatör tarafından değiştirilmesini gerektirir. Eylemsizlik, sistemde aşırı düzeltmeye neden olduğunda sistemin titreşmesine yani sarkaç gibi sallanmasına neden olur. Bu sallantının bastırılması gerekir fakat aşırı bastırmada sistem cevabını zayıflatabilir. Bu yüzden servo sistemler ayarlanırken en hızlı cevabın en az aşım ile elde edilmesi önemlidir. Şekil II.29'da sistem cevabı karakteristiği verilmiştir[3].

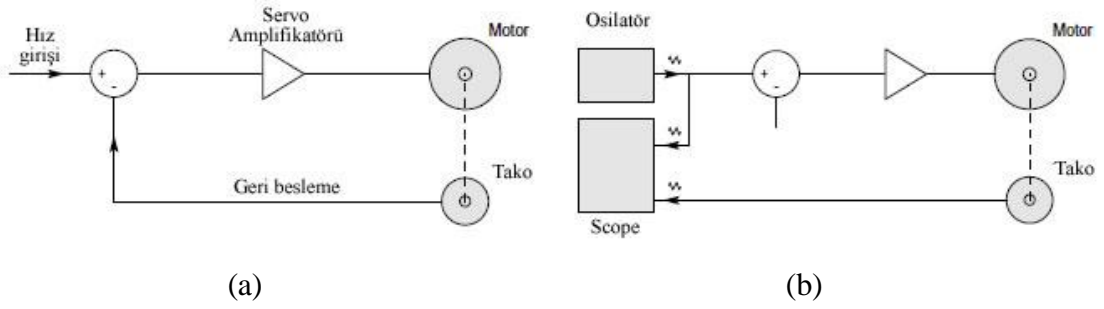


Şekil II.29 Sistem cevabı karakteristiği

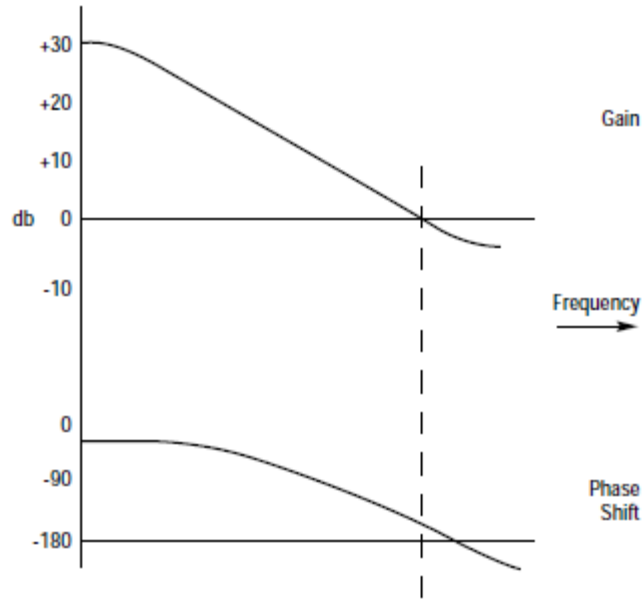
Uygulamada ayarlama; analog sistemlerde potansiyometre ayarlaması, dijital sistemlerde ise sayısal değerlerin değiştirilmesidir. Bu işlemin etkili bir biçimde yapılabilmesi için sürücüde neler olduğunun anlaşılması gerekir. Servo sistem negatif geri beslemeli kapalı çevrimli bir sistemdir. Geri besleme pozitif yapılırsa sistemin titreşmesine neden olur. Dolayısıyla geri besleme sürekli olarak negatif kalmalıdır. Geri beslemenin negatif kalması; giriş düşük frekanslı bir sinüs dalgası ise geri beslemede  $180^\circ$  farkla yine sinüs şeklinde fakat negatif olacaktır. Servo sistemlerin karakteristiği, frekans değişimleri ile değişir. Bu duruma faz karakteristikleri de dâhildir. Bu nedenle düşük frekanslı negatif bir geri besleme, yüksek frekanslı pozitif bir sinyale dönüşebilir. Bunun sonucu aşım veya sürekli sallantılı bir durum oluşabilir[3].

Kapalı çevrimli bir sistemde her birim birbiri ile etkileşim içinde olduğundan kapalı çevrimli bir sistemi analiz etmek zordur. Çıkış girişe negatif geri besleme yaptığı için kendi kendini iptal eden bir durum vardır. Kapalı çevrimin çalışma şeklinin

anlaşılmasının en iyi yolu döngünün açık çevrime dönüştürmektir. Böylece açık çevrim karakteristiği çıkışın belirli bir girişe verdiği yanıtı görme şansı verir. Kazancı ve faz değişimini farklı frekanslarda ölçerek sonucu grafiğe dökülebilir. Şekil II.30'da kapalı çevrimin açık çevrime dönüşümü verilmiştir. Şekil II.31'de açık çevrim kazanç ve faz karakteristiği görülmektedir.



**Şekil II.30 a)** Kapalı çevrim servo hız sistemi **b)** Açık çevrim karakteristik ölçümü



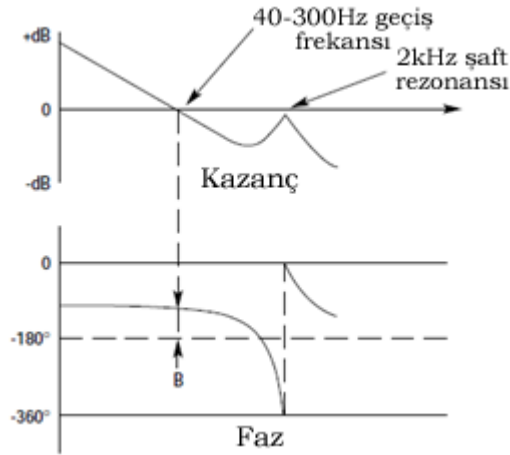
**Şekil II.31** Açık çevrim kazanç ve faz karakteristiği

Şekil II.31'de kazanç desibel (dB) logaritmik olarak skala edilmiştir. Sıfır dB noktası kazancın 1 olduğunu göstermektedir ve giriş ile çıkış genlikleri aynıdır. Faz skalası derece ile gösterilmiş ve giriş ile çıkış arasındaki faz farkını ifade etmektedir. Geri besleme sinyalinin düşük frekansta negatiftir. Faz farkı  $180^\circ$  ulaştığında geri besleme sinyali  $360^\circ$  atlar ve giriş ile aynı faza gelmiş olur. Geri besleme sinyali hiçbir zaman girişten fazla ve aynı fazda olmamalıdır. Pozitif geri besleme sürekli artan bir

salınıma neden olabilir. Matematiksel olarak ve sistem verileri doğrultusunda sistemin kazanç ve faz karakteristiklerini tahmin etmek mümkündür. Pratikte belirli bir problem ile karşılaşılmadıkça bu karakteristikleri ölçmek gereksizdir. Yukarıda belirtildiği gibi bir sistemin istikrarlı çalışması için açık çevrim kazancının birden küçük ve  $180^\circ$  faz farklı olmalıdır. Bu şartlar oluştuğunda sistem girişteki hızlı bir değişime cevap (ring) verecektir. Şekil II.32’de sönümlü sistem cevabı görülmektedir[3].



Şekil II.32 Sönümlü sistem cevabı

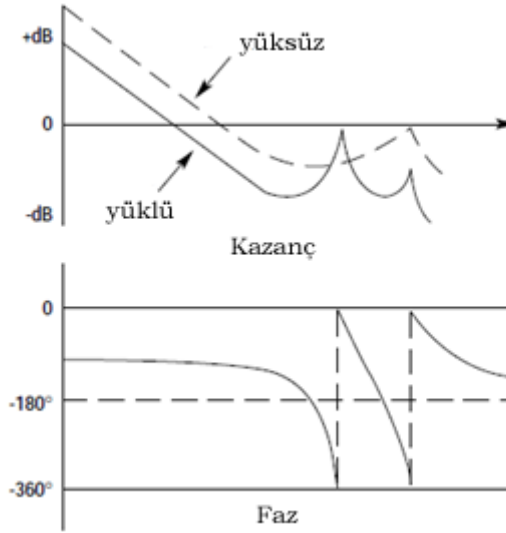


Şekil II.33 Servo sistem açık çevrim kazancı ve faz karakteristiği

Yüksüz bir servo sistemde tipik açık çevrim kazancı ve faz karakteristiği Şekil II.33’de ki gibidir. Şekil II.33 incelendiğinde 2 kHz seviyesindeki ani yükselmenin nedeni şaft rezonansı denen takometre ile motor arasındaki saftaki dairesel salınımdır. Bu noktada fazın  $180^\circ$  kritik hattı aştığı görülmektedir. Bu noktada kazanç birden küçük (0dB) olmalıdır. Zaman sabiti hangi amplifikatörün kazancının düzenleneceğini frekansı kontrol ederek karar verir. Zaman sabiti kontrolü ayarlandığında, yüksek frekans kazancını ani yükselme anı 2 kHz’de sıfırın altında tutulur. Yüksek kazanç ve düşük zaman sabiti ile motor 2 kHz’de vızıldaama şeklinde ses çıkarır. Bir diğer önemli nokta ise geçiş frekansı noktasıdır. Bu noktada kazanç eğrisi 0dB çizgisinden geçer.

Bu frekans genellikle 40 ila 300 Hz civarındadır.  $\beta$  (Beta) noktası geçiş frekansındaki faz farkıdır. Eğer  $\beta$  çok küçük ise sistem aşım ve dalgalanma yapacaktır.  $\beta$  sönüm derecesini belirtir[3].

Sönüm kontrolü geçiş frekansında faz farkını artırır. Bunun için ivme geri beslemesi kullanılır. Faz farkı artırılarak sistemin kararlılığı sağlanır. Sönümü arttırmak aynı zamanda 2 kHz noktasındaki aşımın kazancı düşürmeye yarar. Dışarıdan verilen bir yük hem kazanç hem de faz karakteristiğini değiştirecektir. Yüksek eylemsizlik sadece kazancı düşürmekle kalmaz, kazancın anlık yükselmesine de neden olur. Bu kazanç yükselmesi 2 kHz noktasındakinden daha büyük olabilir. Bu anda motor düşük frekansta zaman sabiti ayarlandığında vızıldaşma şeklinde ses çıkarmaya başlar. Şekil II.34’de yük altında sistem karakteristiğı görülmektedir[3].



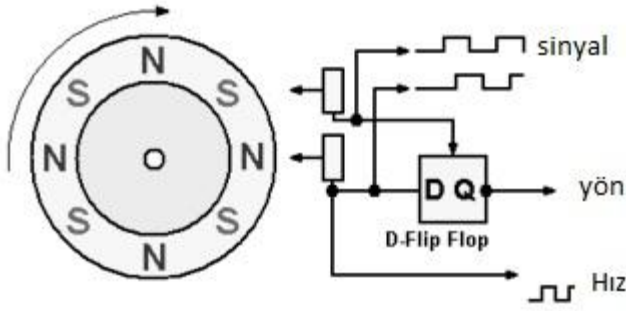
Şekil II.34 Yük altında sistem karakteristiğı

Bu saniyelik yükselmenin genliğı motor ile yük arasındaki bağlantının uyum ve sertliğıne bağılı olarak değışir. Yaylı bir bağlantı yüksek kazanç yükselmesine neden olurken, salınımı engellemek için sistem kazancı düşürmeye çalışırken, sistemin yanıt süresini yavaşlatır. Bu sebeple hızlı bir sistem performansı için motor ile yük arasında dairesel bükümsüz bir kublaj bağlantısı yapılmalıdır.

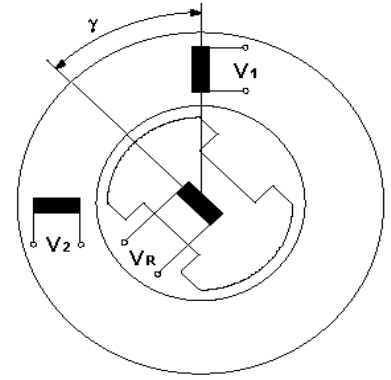
### II.2.5.Geri Besleme Elemanları

Servo mekanizmasının bir diğeri önemli elemanı da geri besleme elemanıdır. Geri besleme elemanı olarak takogeneratör, hall sensörü, resolver ve encoder gibi

elemanlar kullanılır. Hız, doğruluk, çalışma ısı ve konum bilgisi gibi etkenlere göre hangi geri besleme elemanının tercih edileceği belirlenebilir. Geri besleme elemanının temel görevi motorun hareketine bağlı olarak bir sinyal üretmek ve bu sinyali kontrolöre göndererek motorun hızı ve konumu hakkında bilgi vermektir. Takogeneratör sabit mıknatıslı bir DC jeneratördür. Doğrusal çıkış yapısı nedeni ile genellikle hız bilgisini okumada kullanılır. Hall sensörü ise manyetik alandaki veya akımdaki değişimleri algılayan bir sensördür. Hız ölçümü ve pozisyon belirlemesi için kullanılabilir. Hız ve yön belirlemek için basit ve kesin bir çözüm olmakla beraber çözünürlüğü oldukça düşüktür. Şekil II.35’de hall sensörü görülmektedir[4].



Şekil.II.35 Hall sensörü

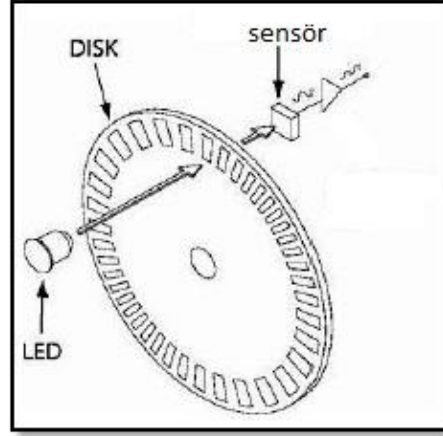


Şekil II.36 Resolver yapısı

Resolver mutlak pozisyon ölçümü yapmakla beraber sağlam ve gürültüye karşı dayanıklıdır. Tek kötü yanı analog çalışmasıdır. Yapısında birbirine dik iki bobin mevcuttur. Bazı durumlarda çoklu biçimlerde de imal edilebilir. Çözünürlüğü hall sensörüne göre daha yüksektir. Yüksek sıcaklıklarda çalışabilir ve motor hareket etmezse bile konum bilgisi verebilir. Şekil II.36’da resolver yapısı verilmektedir[3].

Encoder ise yüksek çözünürlükleri ile diğer geri besleme elemanlarından oldukça üstündür. Hız ve konum ölçebilen encoderlar mekanik hareketi digital bir sinyale çevirmek sureti ile çalışırlar. Encoderlar doğrusal veya dairesel çalışabildikleri gibi mutlak ve artımlı olmak üzere iki farklı çeşidi vardır. Mutlak encoderlarda her konum için belirlenmiş kendine has bir sinyal düzeni varken, artımlılarda sinyal hareket tarafından üretilir. Bu yüzden artımlı encoderlar enerji verildiği andaki konumu sıfır olarak kabul ederken mutlak encoder gerçek anlamda saydığı için enerjisi

kesilse dahi kaldığı yerden devam edebilir. Şekil II.37’de dairesel optik encoder verilmektedir[4].

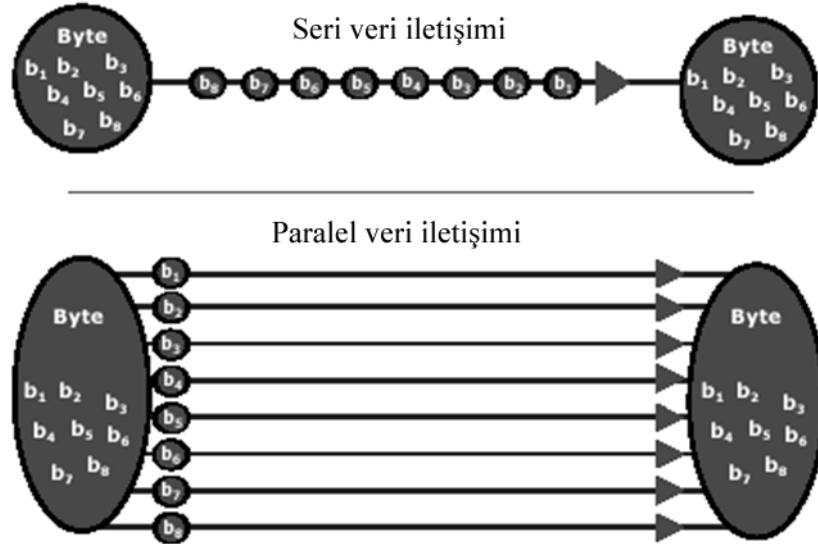


Şekil II.37 Dairesel optik encoder

Ayrıca daha yüksek çözünürlük ve doğruluk gerektiren sistemlerde sin-cos encoderlar da kullanılmaktadır. Encoderlar geri besleme sinyali olarak kare dalga üretirken, sin-cos encoderlar sinüsoidal sinyal üretirler. Bu sinüsoidal sinyal işlenerek istenilen çözünürlükte kare sinyal üretilebilir.

### II.3 VERİ İLETİŞİMİ

Otomasyon sistemlerinde tüm cihazlar arasında paralel veya seri haberleşme altyapısı mevcuttur. Paralel bir iletişim hattında 1 baytlık (1bayt = 8bit) bir veriyi taşımak için 8 paralel kabloya ihtiyaç vardır. Ayrıca hatta aynı anda sadece bir veri sinyali bulunabilir. Seri bağlantıda ise veri tek bir hat üzerinden gönderilse de paralel iletişimdeki gibi bitler halindedir. Veri sadece bir kodlama mekanizması ile kodlanmıştır ve hattın diğer ucunda aynı tür bir kod çözücü tarafından çözülür. Bu sayede aynı veriyi iletmek için daha az iletişim kablosu gerekir ve tek bir hat üzerinden birden fazla veri gönderebilir. İlk bakışta verinin kodlanması, iletilmesi ve kodun çözümlenmesi uzun bir süre gerektirir gibi görünse de BUS sisteminin mimarisine ve eleman sayısına göre bu gecikme mili saniyeler mertebesinde olduğu için pratikte ihmal edilebilir. Şekil II.38’de seri ve paralel iletişimin basit şemaları görülmektedir[2,7].



**Şekil II.38** Seri ve paralel iletişim basit şeması

Alansal Veri Yolu (Fieldbus) kavramında Alan/Saha olarak belirtilen bölge üretim sahası anlamını taşımaktadır. Modern dijital haberleşmenin de sağladığı avantajlar sayesinde Fieldbus sistemleri günümüzde uyarlanabilir ve montaj kolaylığı sağlayan alternatif çözümler olarak karşımıza çıkmaktadır. Temel olarak BUS haberleşmesi farklı cihazlar arasında veri iletişiminin ortak veri hattı üzerinden gerçekleştirilmesidir. Sistemin parçası olan alansal cihazlar tahrik elemanları, algılayıcı ve diğer otomasyon donanımlarıdır.

Fieldbus gerçek zamanlı kontrol yapılan endüstriyel bir iletişim ağıdır. Zincir, yıldız, halka, dal ve ağaç tipi iletişim ağlarında kullanılabilir. İlk bilgisayarlar sadece iki bilgisayarın haberleşmesine izin veren RS-232 (seri bağlantı) ile birbirine bağlanırlardı. Bu 4-20 mA aralığında sinyaller kullanarak noktadan noktaya veri iletişimi sağlamak demektir. Günümüzde ise bilgisayarlar bir fieldbus protokolü olan LAN tipi bağlantılar ile bir iletişim noktasından birden çok analog ve dijital noktaya bağlanabilmektedir. Bu sayede kablo ve bağlantı noktası sayısı azaltılmış olur. Saha veri yolu üzerinden haberleşen cihazlarda olması gereken mikroişlemciler sayesinde birden fazla iletişim noktası aynı cihaz tarafından sağlanmış olur[8].

### **II.3.1 Saha veri yolu (Fieldbus) yapısı**

Bir BUS haberleşme sisteminin verimliliği uygulama alanına ve özellikle transfer edilecek verinin büyüklüğüne bağlıdır. Sistemde doğrudan süreç verileri ve algılayıcılardan gelen bilgiler dolaşabilir ve verinin büyüklüğü 100 byte ölçeğine kadar

ulaşabilir. Sistemin haberleşme hızı ve bazı analog kontrol ünitelerinin cevap verme süreleri çeşitli uygulamalarda çok önem kazanabilmektedir. Bus sistemlerinde haberleşme süresi milisaniyeler mertebesinde olduğu için bu durum belirli bir seviyeye kadar problem teşkil etmez. Ancak haberleşme hiyerarşisi modelinde üstlere doğru çıkıldığında bu süre artma eğilimi gösterir[8,9].

Şekil II.39’da görüldüğü gibi Fieldbus sistemlerini sınıflandırmanın en kolay yolu piramit türü hiyerarşik sınıflandırma referansını kurmaktır. En alt seviye, tahrik ve algılama elemanlarını içerir. Bir üst kontrol seviyesinde algılayıcılardan gelen bilgileri derleyen ve tahrik elemanlarını denetleyen elemanlar bulunur. Sistem veya hücre seviyesi olarak adlandırılan bölgede birden fazla alt kontrol sisteminin düzenli çalışmasını sağlayan elemanlar bulunur ve üst seviye kontrol işlevleri bu düzeyde gerçekleştirilir. Daha üst seviyelerde aktif olarak iş gören ve haberleşen eleman sayısı azalmakta, ancak buna karşın aktarılan veri miktarı ve içerik bilgisi artmaktadır[8].



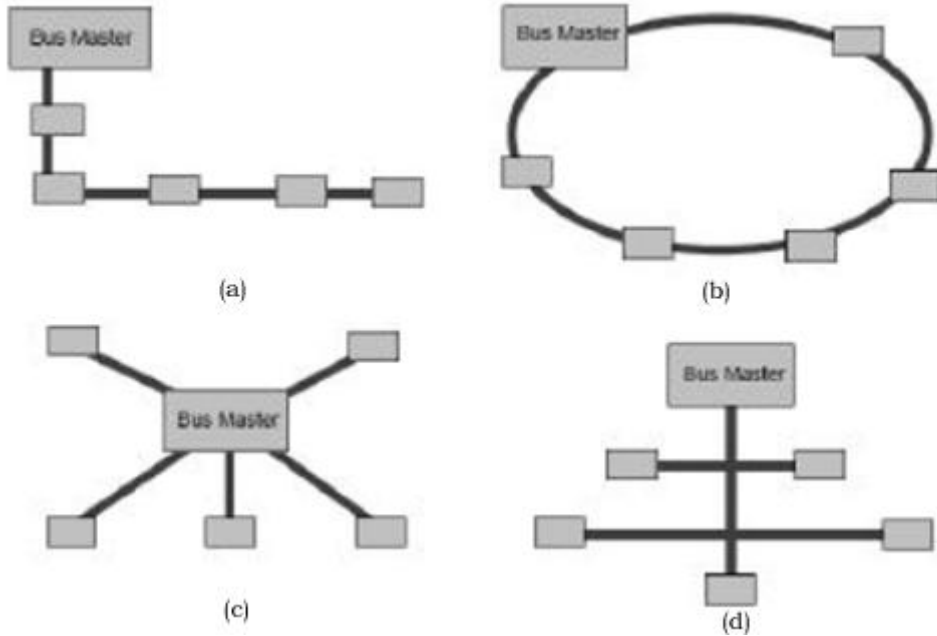
**Şekil II.39** Saha veri yolu piramit hiyerarşisi

Tüm veri iletişimi sistemlerinde olduğu gibi fieldbus sisteminde de fiziksel bir katman yani fiziksel bir verinin var olması gerekir. Bu veri herhangi bir seviyede gerilim, akım veya 1 ve 0’lardan oluşan mantıksal bir yapıda olabilir. Daha sonra birbiri ile ilişkilendirilecek parçaların ne tür bir bağlantı katmanı ile birbirine bağlanacağı belirlenmelidir. Bu katman genelde optik veya bakır kablolardan oluşabilir. Gönderilecek verinin türüne, eleman sayısına ve haberleşme mesafesine göre farklı iletim ortamları tercih edilebilir. Tablo II.4’de veri aktarım ortamları genel özellikleri verilmiştir[10].

**Tablo II.4** Veri aktarım ortamları genel özellikleri

	Veri Aktarımı	Kablo Uzunluğu	EMC	Uygulama
Paralel Kablo	< 100 kbit/s	<1m	Çok Düşük	Bilgisayar Bağlantı Kabloları (Yazıcı)
Burgulu Kablo	<10Mbit/s	<1000m	Düşük	Telefon / Fieldbus Sistemleri
Koaksiyal Kablo	<100Mbit/s	<1000m	İyi	Radyo ve Televizyon
Fiberoptik Kablo	<100Mbit/s	<1000m	Çok İyi	Telefon / Fieldbus Sistemleri

Fiziksel katmanlardan sonra ağ ve transfer katmanları gelir. Ağ katmanı verinin sistemde nasıl dolaştırılacağını belirlemek olarak tanımlanabilir. Bunun için farklı tipte ağ bağlantıları kullanılabilir. BUS ağı içerisinde yer alan cihazların konumu ve birbirleriyle olan mantıksal ilişkileri ağın topolojik yapısı ile ilgili olan kavramlardır. Tüm ağ topolojilerinin temelinde noktadan noktaya bağlantısı olan sadece iki cihaz arasındaki bağlantı yatar. Ancak ikiden daha fazla cihazın bulunduğu durumlarda doğrusal, dairesel, yıldız ve ağaç tipi gibi çok çeşitli ağ yapıları mevcuttur. Şekil II.40'da çeşitli ağların topolojisi verilmiştir[9,10].

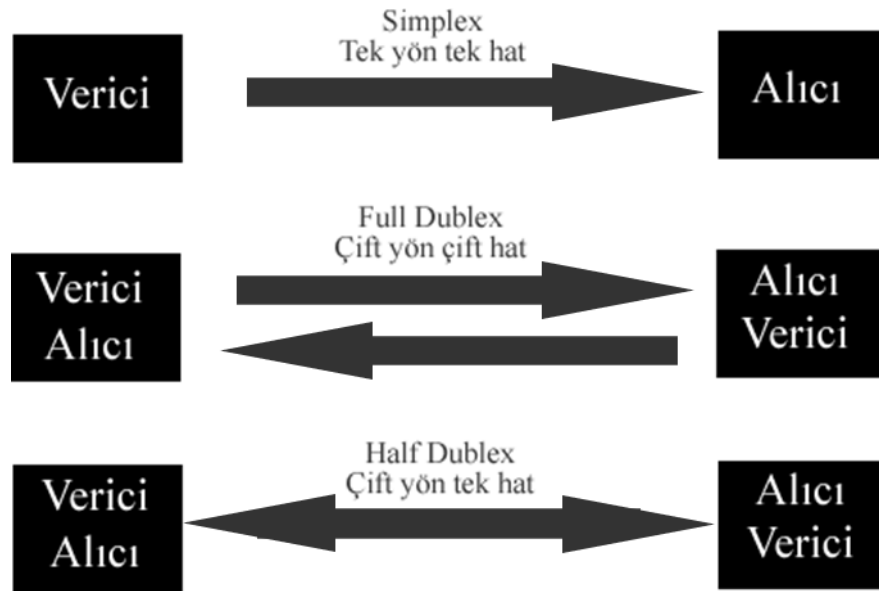


**Şekil II.40** a) Doğrusal b) Dairesel (ring) c) Yıldız d) Ağaç tipi

Transfer katmanında ise gönderilen verinin ilgili alıcı tarafından alınması ve işlenmesi sağlanır. Veri aktarımı esnasında gönderilen bilginin hangi yönde ve hangi öncelikle gönderileceği de önemli bir kavram olarak karşımıza çıkmaktadır. Bu öncelik

birbiriyle haberleşen cihazlar arasında tanımlanmalıdır. Aksi takdirde haberleşme önceliğinde karışıklıklar ve veri kaybı gibi sorunlar yaşanabilir.

Temel olarak tek yönlü ve çift yönlü olmak üzere iki alternatif veri iletişim tekniği mevcuttur. Şekil II.41’de görüldüğü gibi üç ana haberleşme yönü vardır. Noktadan-Noktaya (Simplex), Çift Yönlü (Full Duplex) ve Kısmi Çift Yönlü (Half Duplex) tercih edilebilecek yöntemlerdir. Noktadan-Noktaya veri yönünde verici ve alıcı cihaz önceden belirlenmiş olup sadece tek bir yönde veri aktarımı mümkündür. Bu tip bağlantı genelde PC – yazıcı gibi alıcının geri besleme göndermediği durumlarda kullanılır. Ancak alıcı tarafın geri besleme sinyali gönderebildiği uygulamalarda diğer çift yönlü yöntemler kullanılmalıdır. Bu durumda alıcı cihazın aynı zamanda verici niteliğini de taşıması gerekir. Son olarak ise uygulama katmanı gelir ve bu katmanda gönderilen verinin işlenerek yorumlanması işlemi gerçekleştirilir[10].



Şekil II.41 Veri aktarım teknikleri

### II.3.2 Saha veri yolu protokolleri

Verici ve alıcı arasında hızlı, güvenli ve anlaşılabilir veri iletişimi gerçekleştirilebilmesi için kullanılan yazılım ve donanıma göre bazı unsurların belirlenmesi gerekir. Haberleşme ile ilgili kuralların belirlenmesi ile haberleşme protokolü belirlenmiş olur. Herhangi bir protokolde bilgi aktarımı ve kontrolü ile ilgili tüm detaylar belirlenmiş ve sabitlenmiştir. Alıcı ve verici cihazların belirli bir protokol üzerinden haberleşecek şekilde üretilmiş olması gerekmektedir. Bazı BUS sistem üreticileri sadece kendi haberleşme protokolleri için kullanılmak üzere bütünleşmiş

devreler (IC) üretirler. Bu protokolleri kullanacak olan sistem üreticileri bu elektronik devreleri kullanarak kendi ürünlerinin bu protokol üzerinden haberleşebilmesini temin edebilir. Protokol, kontrol edilen sistemin tüm elemanlarının haberleştiği ve birbiriyle anlaştığı ortak konuşma dili olarak değerlendirilebilir[10].

Geleneksel veri taşıma tekniklerine göre daha avantajlı olan saha veri yolu sistemi çok yeni olmakla beraber günümüzde **IEC 61158** standardına göre yaygın olarak kullanılmaktadır. Bu standart kapsamında sekiz farklı protokol tipi tespit edilmiş ve daha sonra bunlara bazı yeni protokoller daha eklenmiştir. Fieldbus teknolojisinde kullanılan protokoller açık ve lisanslı (Özgün) protokoller olmak üzere ikiye ayrılabilir. Lisanslı protokoller belirli bir firma veya kişi tarafından geliştirilmiş olup, sadece lisansın size sunduğu haklardan yararlanabilirsiniz. Açık kaynaklı protokoller ise farklı firma ve üreticilerin cihazları birbiri ile haberleşebildiği gibi farklı markalardaki muadil cihazlar birbiri ile yer değiştirebilir. İlk fieldbus sisteminin geliştirilmesinden bu yana birçok firma kendi uygulamalarında kullanılmak üzere farklı özelliklere sahip protokoller geliştirmiştir. Tablo II.5’de çeşitli saha veri yolu (Fieldbus) protokollerinin temel özellikleri, Tablo II.6’da açık ve özgün mimari protokolleri gruplandırılmıştır[8].

**Tablo II.5** Çeşitli saha veri yolu (Fieldbus) protokolleri ve temel özellikleri

Saha veri yolu (Fieldbus)	En fazla eleman sayısı	Terminal Gücü	Veri aktarım Hızı
As-Interface	62	Var	<1ms
CANOpen	127	Yok	<1ms
ControlNET	99	Yok	<1ms
CC-Link	64	Yok	<1ms
EtherCAT	65536	Yok	>1ms
Interbus	511	Yok	<1ms
Profibus	126	Yok	<1ms
Sercos III	511	Yok	>1ms
DeviceNET	64	Var	<1ms

**Not:** Eleman sayısı sınırsız olarak gösterilen protokoller çok fazla eleman kullanılabildiği için sınırsız olarak belirtilmiştir.

**Tablo II.6** Bazı açık ve özgün mimari protokoller

PROTOKOL	TANIMLAMA	ÜRETİCİ FIRMA
AÇIK MIMARI	INTERBUS-S PROFIBUS CANOPEN DEVICENET AS-INTERFACE FIP CAN LON	PHOENIX PNO/SIEMENS CIA ALLEN-BRADLEY ASI VEREIN/SIEMENS AEG-SCHNEIDER AUTOMATION INTEL/MOTOROLLA/PHILLIPS ECHELON/MOTOROLLA
ÖZGÜN (SPESİFİK) MIMARI	BECKHOFF I/O MELSEC SUCONET K ET 100 CS 31 REMOTE I/O MODNET	BECKHOFF MITSHUBISHI KLOCKNER MOELLER SIEMENS ABB ALEN BRADLEY AEG

### II.3.2.1 AS-Interface

Aktüatör-Sensör ara yüzü olarak adlandırılan bu sistem en alt düzeyde oldukça basit bir altyapı ile uygulanmaktadır. Paralel veri iletişime göre en basit alternatif olan sistem birçok farklı firma tarafından finanse edilen bir araştırma gurubu tarafından geliştirilmiştir.

ASI oldukça basit ve ucuz olmanın yanında aynı zamanda güvenli bir sistemdir. Temelde merkezi bir kontrol ve buna bağlı 31 alt düzey kontrol sistemi içerebilir. Her alt düzeyde 4 bit giriş ve 4 bit çıkış olmak üzere 8 bit veri işlenebilir. Günümüzde çeşitli çalışmalarla bu kapasiteler bir miktar arttırılmıştır. ASI sistemlerinde kablo boyun 100 metre ile sınırlı olsa da tekrarlayıcılar sayesinde 300 metreye kadar iletişim gerçekleştirilmektedir. ASI için özel olarak geliştirilen yassı kablolar mevcuttur. Veri ve voltaj aynı kablo üzerinden taşınabilir. Bu voltaj 24-30V seviyesinde olduğundan daha yüksek gerilim gereksinimi olan yerlere ayrı bir besleme kablosu çekilmelidir. Ayrıca bu kablolar sayesinde bağlantı noktalarında kabloları kesmeden kablo ısırma yöntemi ile gerekli bağlantılar oluşturulabilmektedir[11].

### II.3.2.2 Profibus (Process Fieldbus)

Profibus sistemleri alt seviyeli otomasyon cihazlarına ve kontrol elemanlarına bağlanabildiği gibi süreç kontrol elemanlarına da bağlanabilmektedir. Genel olarak kullanılan **DP** (Decentralized Periphery) ve **PA** (Process Automation) olmak üzere iki varyasyonu bulunmaktadır. Profibus-DP sensör ve aktüatörlerin merkezi bir sistem ile kontrol edilmesi, Profibus-PA ise süreç işlevlerinin ölçümü ve takip edilmesi için kullanılır. Lineer topoloji üzerinden iki hatlı kablo ile haberleşen Profibus sisteminde kullanılacak maksimum kablo mesafesi taşınacak olan verinin büyüklüğü ile ters

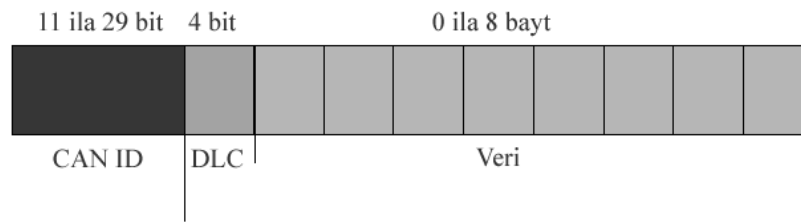
orantılıdır. 12Mbaud seviyesinde 100 metre sınırı varken 9.6kbaud seviyesinde 1200 metre kablo kullanılması mümkündür[8].

Profibus sisteminde üst ve alt düzey kontrol elemanlarının erişim ve kontrol hakları belirginleştirilerek birbirinden ayrılmıştır. Tüm alt seviye cihazların ana veri hattına erişimi “master” cihazlar tarafından denetlenir. Çevresel aygıtların ayrı ayrı kontrolü, sistem çalışırken aktif konumdan çıkarılması ve yine sistem çalışırken yeni cihazların eklenmesi mümkündür. Sistemin hata algılama özellikleri son derece gelişmiştir ve ağın herhangi bir düzeyinde ve noktasındaki arıza belirlenebilir. Profibus ile tasarlanan kontrol sistemin PC veya programlama ünitesi ile bölgesel olarak çalıştırılabilir ve ölçeklenerek devreye alınabilir[8].

### II.3.2.3 CAN Bus

CAN (Controller Area Network) otomotiv teknolojileri için geliştirilmiş, mikro kontrolörlerin sunucu bir bilgisayara gereksinim duymadan haberleşmesine olanak sağlayan bir standarttır. Özellikle otomotiv sanayinde kullanılmak üzere geliştirilmiş olmasına rağmen otomasyon ve tıbbi cihazlarda da yaygın olarak kullanılmaktadır.

CAN elektronik kontrol birimlerinin (ECU) birbiri ile haberleşmesini sağlayan ve birden çok üst seviye kontrol elemanı (multi-master) bulundurabilen seri bir haberleşme standardıdır. Her bağlantı noktasından sinyal alınabildiği gibi gönderilebilir de. Bir mesaj öncelikle gönderen ve mesaj tipini belirten bir kimlik (ID), verinin uzunluğunu belirten bir koddan (DLC) ve sekiz bayta kadar veriden oluşur. Seri olarak iletilen bu veri tüm bağlantı noktalarınınca algılanır. Şekil II.42’de CAN verisinin genel yapısı görülmektedir[8,9].

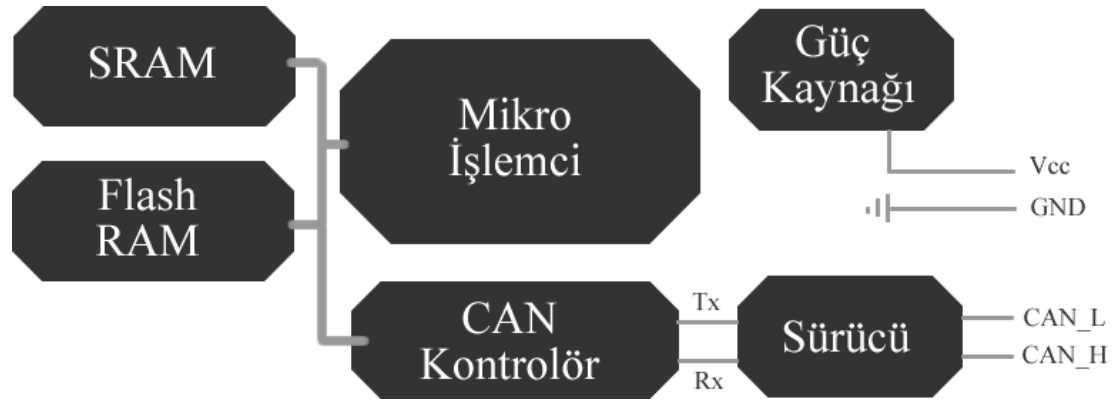


Şekil II.42 CAN verisinin genel yapısı

Hat üzerine bağlı olan sensör veya aktuatör gibi elemanlar hat üzerinde dâhili bir işlemci ve CAN kontrolörü ile bağlıdır. Dâhili işlemci hangi mesajın ne anlama geldiğini ve iletilip iletilmeyeceğine karar verir. Hatta birden fazla sinyal varsa baskın kimlik (ID) bilgisi olan diğerlerini bastırır. Bu işlemin gerçekleşmesi sürecinde hat

üzerindeki veriyi bünyesinde toplaması için CAN kontrolörü kullanılır. Aynı şekilde hatta gönderilecek veri yine bu kontrolör üzerinden bitler halinde seri olarak iletilir.

CAN üzerinde iletişim 40 metrelik bir mesafede 1Mbit/s kadar mümkündür. Ancak mesafe arttıkça bu hız düşer. Örneğin, 500 metrelik bir sistemde hız 125kbit/s ve 5km ise 10kbit/s kadar düşer. CAN sistemlerinde sadece gerekli olduğu anlarda veri aktarımı yapıldığı için diğer sistemlere göre iletişim yükü daha azdır. Ayrıca verinin birden fazla kontrol mekanizmasından geçerek iletilmesi, bu sistemlerdeki veri akışını daha güvenli kılmaktadır. Aynı şekilde hataların tespiti ve giderilmesi de daha hızlı gerçekleşmektedir. Şekil II.43’de CAN modülü blok şeması görülmektedir[8,9].

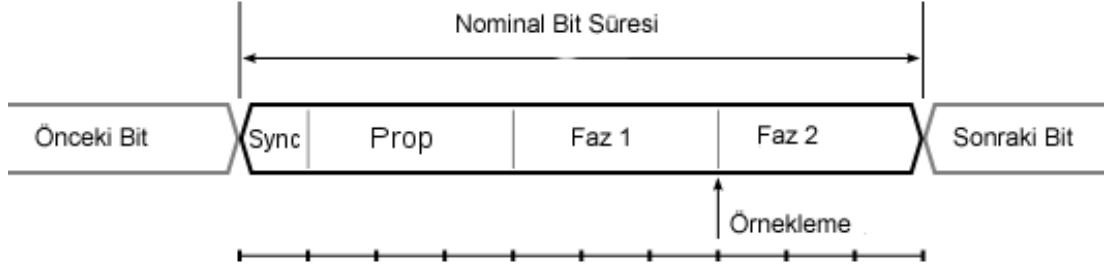


Şekil II.43 CAN modülü blok şeması

CAN sisteminde mesajların öncelik sırasına göre iletilmesi için baskın bit değeri mantıksal “0” ve çekinik bit değeri ise mantıksal “1” olarak belirlenmiştir. Mantıksal VE kapısında olduğu gibi her iki bit değeri birbiri ile çarpıldığı için mantıksal “0” her zaman baskındır. CAN sisteminde çekinik bir bit gönderilirken, aynı anda baskın bir bit gönderilirse baskın bit iletilir. Öncelikli verilerde gecikme olmazken, daha düşük öncelikli veriler baskın verinin gönderilmesi bitince iletilirler[9].

Taşıyıcı duyumlu çoklu geçişli bit bit karşılaştırmalı (CSMA/BA) diferansiyel bir terminal ile bir öncelik planı oluşturulabilir. Burada veri iletimini yapacak eleman hatta dinleyerek iletim için en uygun anı bekler. Ayrıca öncelik planına göre hangi verinin daha önce iletileceği belirlenmiş olduğundan çakışmalar engellenmiş olur. Diğer yandan öncelik kimlikleri (ID) bit bit karşılaştırılarak en baskın veri tespit edilir ve diğerleri bir sonraki iletme kadar geri çekilirler. CAN haberleşme ağında her düğüm

elemanının kendi eşzamanlı saati bulunur ve iletişim sırasında herhangi bir zaman bilgisi gönderilmez. Senkronizasyon mesaj bitlerinin bölünerek numaralandırılması ile yapılır. İlk olarak eşleme bölümü ve ardından sırası ile propaganda, faz 1 ve faz 2 bölümleri gelir. Her faz evresinin uzunluğu sistem ve düğüm elemanlarının durumuna bağlı olarak ayarlanabilir. Örnekleme noktası ise iki faz arasında bulunur. Bu sayede sürekli bir eşleme sağlanmış olur. Şekil II.43’de bit zamanlaması verilmiştir[8,9].



Şekil II.43 Bit zamanlaması

CAN ağları iki farklı tip mesaj formatına göre ayarlanabilir. Bunlardan ilki standart (CAN 2.0A) ikincisi ise genişletilmiş (CAN 2.0B) ifadeleridir. Bu iki format arasındaki tek fark standart CAN ifadesi kimlik bilgisi için 11 bit barındırırken, genişletilmiş CAN ifadesinin 29 bit barındırmasıdır. İki ifade arasındaki ayırım genişleme kimlik biti (IDE) ile yapılır. Genişletilmiş ifadeleri destekleyen CAN kontrolörleri aynı zamanda standart ifadeleri de gönderip alabilir.

CAN üzerinden dört değişik ifade çerçevesi bulunur. Bunlar:

- **Data Frame** iletilen veriyi taşıyan
- **Remote Frame** arzu edilen veriyi çağıran
- **Error Frame** hata kontrolü yapan
- **Overload Frame** aşırı yükü bildiren

Veri taşıyan ifadenin 11 veya 29 bit kimlik taşıyan iki farklı türü olduğunu daha önce belirtmiştik. Veri çağıran ifade ise onun bir veri ifadesi olmadığını belirten bir bit bulunur (RTR) ve bu bitin değeri “1” olarak çekiniktir. Ayrıca veri çağrısı yapan ifade herhangi veri alanı bulunmaz. Hata ifadesi iki bölümden oluşur. İlk bölüm farklı istasyonlar tarafından sağlanan hatanın hangi düğümde olduğunu belirten işaret, ikinci bölüm ise hata sınırlamasıdır. Aşırı yük ifadesi de hata ifadesi gibi iki bölümden

oluşur. İlki aşırı yük işareti, ikincisi ise aşırı yük sınır belirteçidir. Tablo II.7’de Standart ifade formatı, Tablo II.8’de genişletilmiş ifade formatı görülmektedir[8,9].

**Tablo II.7** Standart ifade formatı

Bölüm adı	Uzunluk	Açıklama
Başlangıç (Start-of-frame)	1 bit	İfade çerçevesinin başlangıcını bildirir
Kimlik (Identifier)	11 bit	Veriye has standart kimlik bilgisi
Uzak ileti çağrısı (RTR)	1 bit	Veri ve çağır ifadesi ayrımını yapar
Kimlik uzantısı (IDE)	1 bit	Standart “1” ve genişletilmiş “0” kimlik belirtir
Rezerve (r0)	1 bit	Rezerve biti
Veri uzunluğu kodu (DLC)	4 bit	Verinin uzunluğunu belirtir
Veri alanı	0-8 bayt	İletilecek veri
Döngüsel atık denetimi (CRC)	15 bit	Hata kontrolü yapar
CRC limit	1 bit	Çekinik “1” olarak seçilir
ACK	1 bit	Verici “1” ve Alıcı “0” arasındaki kontrol
ACK limit	1 bit	Çekinik “1” olarak seçilir
Son (End-of-frame)	7 bit	Çekinik “1” olarak seçilir

Veri ve uzak çağrı ifade çerçeveleri önceki çerçevelerden bir bit alanı ile ayrılır. Bu alana *interframe space* denir. Aşırı yük ve hata çerçevelerinde böyle bir ayrım yoktur. Ayrıca CAN mesajlarında bit doldurma (bit stuffing) denilen ve beş ardıl aynı bitin ardından gelen karşıt bir bitle yapılan bir teknikle kullanılır. Doldurulmuş veriler alıcı tarafından eski haline getirilir. Sistemde tüm veriler doldurulmuş olduğundan ardıl altı aynı tür ifade (111111 ya da 000000) hata olarak yorumlanır[11].

CAN standardı uygulama katı görev protokolleri içermediği için akış kontrolü, cihaz adresleme ve veri bloklarının taşınması gibi işlemler için üst katman eklemeleri yapılması gerekir. DeviceNet, CANopen, SDS, CANaerospace, CAN Kingdom gibi birçok üst katman protokolleri bu tür amaçlar için özel olarak hazırlanmıştır.

**Tablo II.8** Genişletilmiş ifade formatı

Bölüm Adı	Uzunluk	Açıklama
Başlangıç (Start-of-frame)	1 bit	İfade çerçevesinin başlangıcını bildirir
Kimlik A (Identifier)	11 bit	Veriye has standart kimlik bilgisi
Yedek uzak çağrısı (SRR)	1 bit	Çekinik “1” olarak seçilir
Kimlik uzantısı (IDE)	1 bit	Standart “1” ve genişletilmiş “0” kimlik belirtir
Kimlik B (Identifier)	18 bit	Kimlik uzantısı
Uzak ileti çağrısı (RTR)	1 bit	Veri “0” ve çağır “1” ifadesi ayrımını yapar
Rezerve (r0, r1)	2 bit	Rezerve bitler
Veri uzunluğu kodu (DLC)	4 bit	Verinin uzunluğunu belirtir
Veri alanı	0-8 bayt	İletilecek veri
Döngüsel atık denetimi (CRC)	15 bit	Hata kontrolü yapar
CRC limit	1 bit	Çekinik “1” olarak seçilir
ACK	1 bit	Verici “1” ve Alıcı “0” arasındaki kontrol
ACK limit	1 bit	Çekinik “1” olarak seçilir
Son (End-of-frame)	7 bit	Çekinik “1” olarak seçilir

## **BÖLÜM III**

### **SAHA VERİYOLU İLE ÜÇ EKSENLİ HAREKET KONTROLÜ**

Bu bölümde tasarlanan sistem için gerekli motor, sürücü, hareket modülü, sensör ve benzeri elemanlar birer birer ele alınarak incelenecektir. Sistem bileşenlerinin incelenmesinden sonra sistemin bütünlüğünü sağlayan iletişim ve geri besleme elemanları ele alınacaktır. Mekanik ve elektriksel sistem tasarımının yanı sıra ayrıca üç eksenli hareketin bilgisayar ortamında programlanması için gerekli yazılım ve yazılım dili incelenerek çeşitli uygulamalar için örnek programlar yazılacaktır.

#### **III.1 SİSTEM BİLEŞENLERİ**

Sistem, Latince ve Yunanca birleşme, oluşma ve bir araya gelme anlamına gelen kelimelerin birleşmesinden meydana gelen bir kelimedir. Sistem dendiğinde akla birbiri ile etkileşim içerisinde bir bütün oluşturacak şekilde çalışan elemanlar topluluğu gelmektedir. Bu anlamda sistem birbiriyle madde, enerji veya bilgi alışverişinde bulunan elemanlar veya parçaları anlamına gelebilir. Bu durumda bu elemanlarını ve onların eylemlerini kapsayan matematiksel veya mantıksal bir model oluşturulabilir. Bir sistem birden çok alt sistemi içinde barındırabildiği gibi kendisinden daha büyük bir sistemin parçası da olabilir[2].

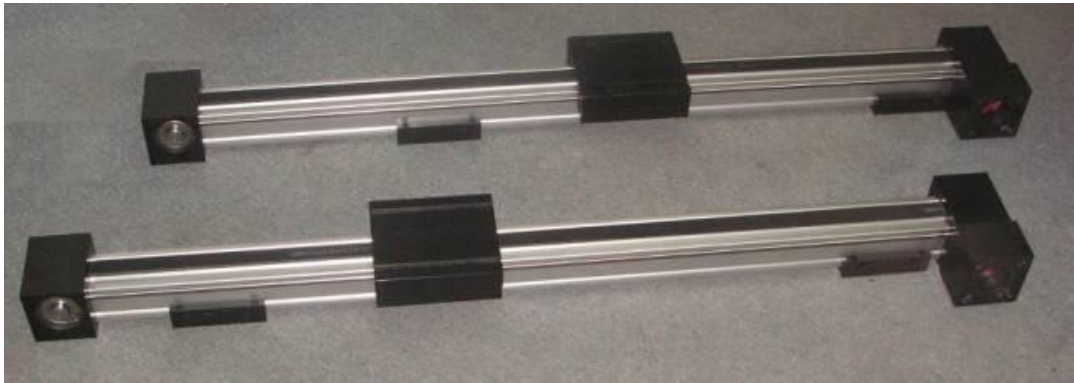
Sistem geliştirme sürecinde ilk adım ihtiyacın belirlenmesi ve bu ihtiyacın giderilmesine yönelik gerekli analizin yapılmasıdır. Bu analizin sonuçlarına göre gerekli sistem tasarımı yapılır. Hareket kontrolünde en önemli ihtiyaç sistemin kararlılığı ve hızıdır. Öte yandan maliyet ve uygulama kolaylığı da diğer değişkenlerdir. Hareket kontrolüne yönelik bir sistem tasarlanırken ilk adım çalışma ortamına uygun elemanların seçimidir. Yanıcı maddelerin olduğu bir ortamda çalışacak elemanlarla, nemli ve ıslak bir ortamda çalışacak olanlar birbirinden farklı özelliklere sahip olmalıdır. Eleman seçiminde bir diğer faktörde sistemin yapacağı işin niteliğidir. Bu sistemin ne kadar hassas, hızlı veya güçlü olması gerektiğinin belirlenmesidir.

Doğru elemanların seçimiyle tasarım süreci başlar ve seçilen bu elemanların birbiri ile doğru etkileşim içerisinde çalışmasını sağlayacak sistemin tasarımı yapılır. Tasarım sürecinde planlama ve analizde hesaba katılmayan sorunlarla karşılaşılabilir. Hatta tüm tasarım bittikten sonra bile en başa dönerek bazı değişiklikler yapmak gerekebilir. Bu yüzden bir sistem tasarlanırken her aşama dikkatle ele alınarak incelenmelidir. Hareket kontrolüne yönelik bir sisteminin tasarımı en önemli aşamalar tahrik ve sürücü mekanizmasının seçimi, haberleşme sisteminin seçimi ve mekanik sistemin tasarımı olarak özetlenebilir.

### III.1.1 Doğrusal hareket modülü

Mekanik sistem tasarlanırken dikkat edilmesi gereken ilk nokta hareketin niteliğidir. Hareket doğrusal veya dairesel olabildiği gibi sürekli veya kesik de olabilir. Ayrıca mekanik sistemin tahrik mekanizması ve sensörler ile uyum içinde tasarlanması gerekir. Mekanik sistemin bileşenleri yükün ve hareketin niteliğine uygun biçimde seçilmelidir.

Üç eksenli hareket için doğrusal ve dairesel parçaların bir arada kullanılması mümkündür. Bu parçalar hareketin ve yükün özelliklerine göre değişiklik gösterebilir. Ayrıca tahrik mekanizmasından alınan hareketin kayış, bant, dişli, zincir veya temassız manyetik mekanizmalarla sistemin gerekli yerlerine iletilmesi mümkündür. Tasarımımızda ağırlık ve sessiz çalışma önemli olduğunda alüminyum gövdeli ve kayışlı doğrusal modüller tercih edilmiştir.



Şekil III.1 Doğrusal hareket Modülleri

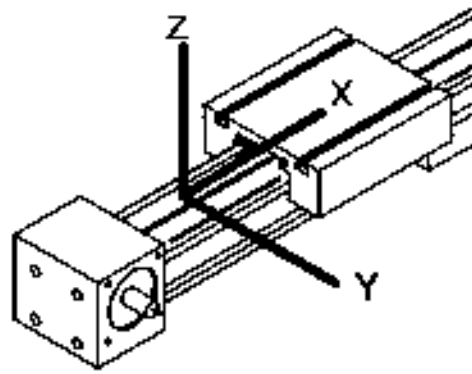
Bu modüller farklı kombinasyonlar kullanılarak çok eksenli hareketlerin yapılması mümkün olmaktadır. Hareket modüllerinin ağırlığı ve hareketi iletmek için kullanılan mekanizmanın çeşidi hareket için harcanacak güç miktarı üzerinde etkilidir.

Ayrıca bu değişkenler sistem cevabının hızını da etkiler. Tablo III.1’de üç eksen için uygulanması gereken güç miktarı ve moment değerleri verilmiştir[12].

**Tablo III.1.**Doğrusal eksen dinamik hareket değerleri

	<b>Açık Kayış</b>	<b>Gizli Kayış</b>	<b>Bilyeli Vida</b>
<b>F<sub>x</sub></b>	650N	650N	*
<b>F<sub>y</sub></b>	400N	400N	350N
<b>F<sub>z</sub></b>	1000N	1000N	850N
<b>M<sub>x</sub></b>	45Nm	45Nm	35Nm
<b>M<sub>y</sub></b>	65Nm	65Nm	45Nm
<b>M<sub>z</sub></b>	90Nm	90Nm	65Nm
<b>V</b>	3m/s	3m/s	*

\*Kullanılan bilyeli vidaya göre değişir



**Şekil III.2.**Doğrusal eksenler

Doğrusal modüller açık kayış, gizli kayış ve bilyeli vida tahrikli olmak üzere üç çeşittir. Yataklama hassas taşlanmış miller ve rulmanlı makaralarla yapılmıştır. Özel sıyırıcı-yağlayıcılar kullanılarak kirli ve tozlu ortamlardan kaynaklanacak olumsuzluklar önlenmiştir. Kullanılan HTD serisi dişli kayışlarla pek çok endüstriyel uygulama için gereken doğrusal kuvvet, tekrarlanabilirlik ve hassasiyet sağlanmıştır. Bilyeli vida tahrikli doğrusal modüller ile yüksek eksenel kuvvet ve çok küçük tekrarlanabilirlik değerlerine ulaşmak mümkündür[12].

### **III.1.2 Tahrik Mekanizması**

Bir sistemi elektrik enerjisi vererek harekete geçirmek elektriksel tahrik olarak tanımlanır. Elektriksel tahrikte genelde elektrik motorları kullanılır. Günümüzde motor teknolojileri yüksek verimlilikli materyaller sayesinde oldukça ilerlemiştir. Bu ilerlemenin son adımlarından biri olan daimi mıknatıslı ve fırçasız motorlar otomasyon teknolojilerinde sıklıkla kullanılmaktadır. Boyutlarına görece ile oldukça verimli ve kararlı çalışan bu motorlar, sürücü teknolojilerindeki gelişmelere paralel olarak yayın olarak kullanılmaya başlamıştır.

#### **III.1.2.1 Motor seçimi**

Motor seçiminde ilk dikkat edilmesi gereken hususlar motorun çalışacağı ortama ve sistemle uyumlu olmasıdır. Elektromanyetik uyumluluk, çalışma sıcaklığı, ortamın nem seviyesi gibi bazı değişkenler üretici firmalar tarafından belirtilen sınırlar içerisinde olmalıdır. Ayrıca motorun azami akımı, mekaniksel hız limiti ve çalışma gerilim değerleri tasarladığımız sistemle uyum içinde olmalıdır. Motor seçiminde dikkat edilmesi gereken bir diğer önemli etkende motorun fiziksel boyutları ve tasarlanan sisteme uygulanabilirliğidir. Bunların dışından motorun türüne ve nasıl sürüleceğine bağlı olarak geri besleme, frenleme ve koruma elemanları da doğru seçilmelidir.

Tasarladığımız sistemde birçok üstün özelliğinden dolayı daimi mıknatıslı fırçasız motorlar tercih edilmiştir. Bu motorlar küçük yapılarına rağmen ihtiyaç duyulan moment kuvvetini düşük hızlarda dahi sağlamaktadır. Üç eksen kontrolü için üç motor kullanılmış olup, bu motorlardan ilki ana eksen hareket ettirdiği için diğerlerine nazaran daha büyük yapıda ve yüksek moment kuvvetine sahiptir. Diğer iki motor ise daha küçük olmalarına rağmen ana eksen motoruna nazaran daha yüksek hızlara çıkabilmektedir. Sistem için kullanılacak olan motorlar SEM kontrollü motor teknolojilerine ait fırçasız AC servo motorlardır. Bu motorlar hem mekanik sisteme hem de kullanılacak sürücüye uyumluluk açısından seçilmiştir. Nitekim geri besleme elemanı olarak yüksek çözünürlüklü resolver kullanan bu motorlar maliyet bakımından dolayı da tercih edilmiştir. Aşırı ısınmalara karşı termistör korumasının yanı sıra çevresel etkilere karşı koruma faktörü olarak tüm SEM servo motorlar mil haricinde IP65 ve milde ise IP64 koruma sınıfına sahiptir. Bu koruma sınıfları ile ilgili ayrıntılı bilgiler Tablo III.2’de verilmiştir[13].

**Tablo III.2** Motor koruma sınıfı bilgileri

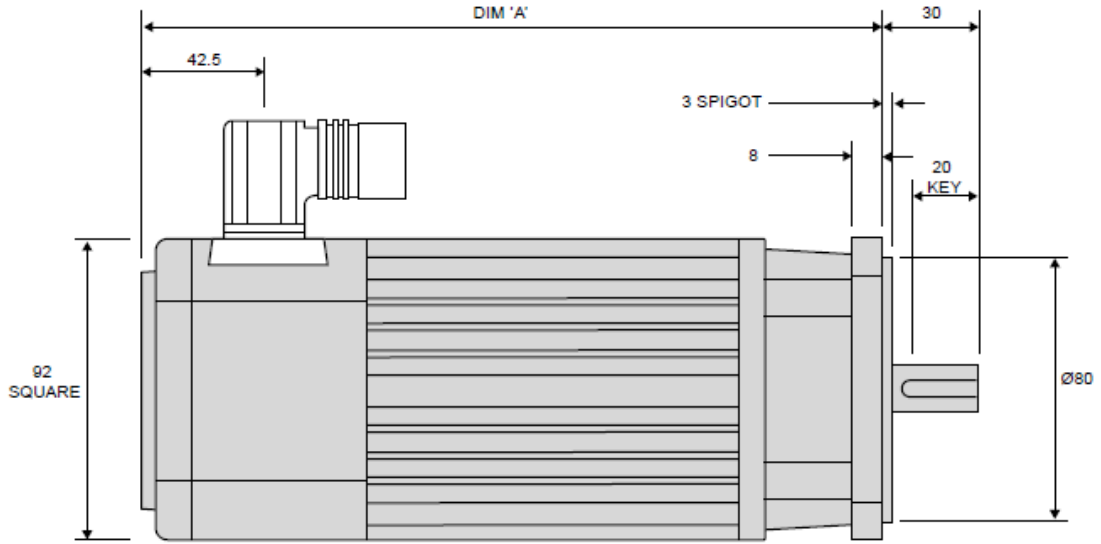
Koruma sınıfı	Açıklama
<b>IP65</b>	Toza karşı tam koruma. Her yönden gelebilecek düşük basıncı sıvılara karşı sınırlı girişim koruması.
<b>IP64</b>	Toza karşı tam koruma. Her yönden gelebilecek sıvı püskürmelerine karşı sınırlı girişim koruması.

Kullanılacak motorlar seçilirken koruma sınıfı ve elektromanyetik uygunluk gibi konular ne kadar önemliyse, motorların çalışma gerilimi, kutup sayısı ve atalet momenti gibi teknik özellikleri de önemlidir. Üretici firmalar genelde ürettikleri motorlara bu gibi teknik özellikleri özetleyen ürün kodları belirlerler. Mesela ürün kodunun başındaki bir harf motorun türünü, sonundaki bir rakamda çalışma gerilimi veya kutup sayısını belirtebilir. SEM motorlarda ilk iki harf motor türünü, sonraki iki rakam milimetre kare olarak çerçeve alanını, bu iki rakamdan sonra gelen harf motorun uzunluğunu, daha sonra gelen rakam ise kutup sayısını belirtir. Tire ile ayrılmış olan sondaki rakam iki faz arasında 1000 devirde tepe gerilim değerini gösterir. Son olarak bu rakamın sağındaki harf geri besleme ve komitasyon ile ilgili bilgiyi verir. Kullandığımız motorlar **HR92C4-64S** ve **HR55C4-32S** olduğuna göre bu motorların tiplerini Tablo III.3’de görüldüğü gibi özetleyebiliriz[13].

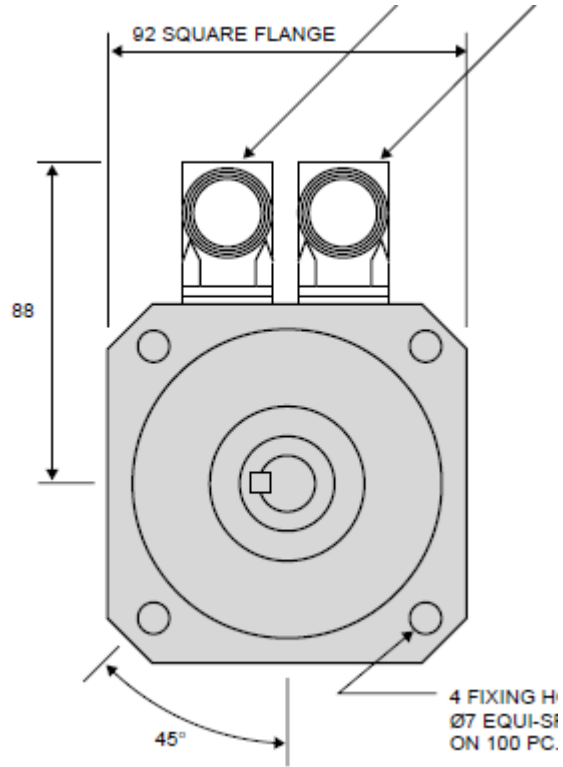
**Tablo III.3** Motor tipi tanımlama kodları

Kod	Açıklaması	Kod	Açıklaması
<b>HR</b>	Yüksek tepkili fırçasız servo motor (High Response)	<b>HR</b>	Yüksek tepkili fırçasız servo motor (High Response)
<b>92</b>	92mm <sup>2</sup> çerçeve genişliği	<b>55</b>	55mm <sup>2</sup> çerçeve genişliği
<b>C</b>	Motor uzunluğu (En kısa A)	<b>C</b>	Motor uzunluğu (En kısa A)
<b>4</b>	Motorun kutup sayısı	<b>4</b>	Motorun kutup sayısı
<b>64</b>	Aşık voltaj değeri (Her 1000 devir için iki faz arası değer)	<b>32</b>	Aşık voltaj değeri (Her 1000 devir için iki faz arası değer)
<b>S</b>	Sinüzoidal (resolver ile)	<b>S</b>	Sinüzoidal (resolver ile)

Diğer yandan motor seçiminde motorun boyutları ve bağlantı noktalarının konumunda dikkat edilmelidir. Motor seçilirken motorun boyutlarının ve bağlantı noktalarının konumlarının sistemin çalışmasını herhangi bir şekilde etkilemeyecek şekilde seçilmeli veya düzenlenmelidir. HR92 serisi motorun önden ve yandan görüşleri ait şemalar Şekil III.3 ve Şekil III.4’de verilmiştir.



Şekil III.3 HR92 serisi motor yan görünüm şeması



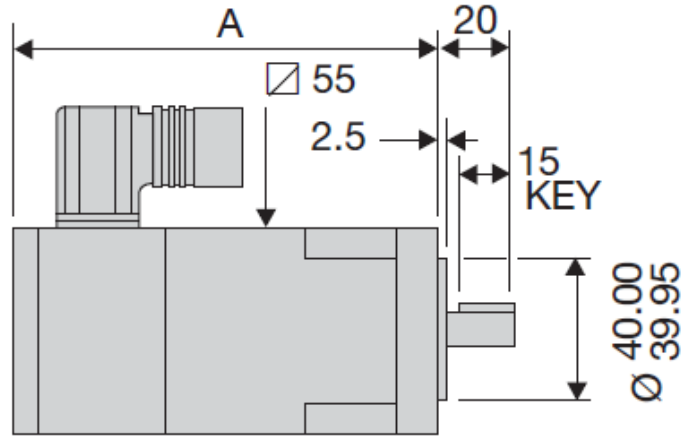
Şekil III.4. HR92 serisi motor ön görünüm şeması

**HR92** serisi motor ana eksenini kontrol edeceği için moment kuvveti diğer motorlara göre daha fazladır. Ana eksen aslında diğer iki eksenin farklı olmamakla beraber diğer eksenlerin mekanik sistemlerinin ve motor ağırlıklarının da bu eksen üzerine binmesinden ötürü ana eksen olarak tanımlanmıştır. Kısaca hareketli sistemin tüm yükünü bu motor taşıyacağı için moment kuvveti yüksek ve dolayısıyla fiziksel boyutları da büyüktür. **HR92** serisi motorlar genel olarak birbirine benzemekle beraber kullanılan geri besleme elemanı ve frenleme sistemine göre boyutları değişebilmektedir. **HR92C** modelinde uzunluk 230mm'dir. Birde çalışma gerilimleri ve dolayısıyla akımları birbirinden farklılık göstermektedir. **HR92C4-64S**'e ait teknik özellikler Tablo III.4'de verilmiştir[13].

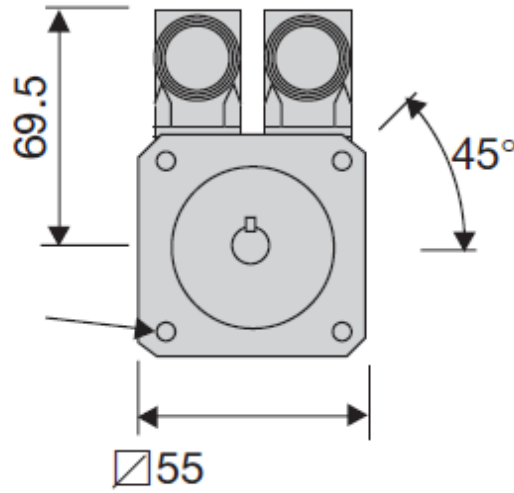
**Tablo III.4** HR92C4-64S Teknik verileri

Değişken	Birim	Değer
Max. Akım	A	11
Daimi Akım (rms)	A	2
Max. E.M.K.	V	380
Max. Tork	Nm	4.6
Daimi Tork	Nm	1.5
Max. Mekanik hız	d/dk	6000
Atalet momenti	kgcm <sup>2</sup>	0.94
Sürtünme	Nm	0.04
Sargı direnci	Ω	10.4
Sargı endüktansı	mH	43
Çalışma sıcaklığı	°C	40
Motor ağırlığı	kg	4.1

Daha önce belirtildiği gibi diğer iki eksen için kullanılacak olan motorlar hareketlerini ana eksen üzerindeki modüller aracılığı ile gerçekleştireceği için daha küçük olarak seçilmiştir. Ayrıca taşıyacakları yük ana eksen üzerine binen yüke nazaran daha az olacağı için fazla moment kuvvetine de ihtiyaç duyulmamaktadır. HR55 serisi motorlar ufak yapılarına rağmen yüksek hızlara çıkabilmektedir. HR55 serisi motorun önden ve yandan görünüşleri ait şemalar Şekil III.5 ve Şekil III.6'da verilmiştir.



Şekil III.5 HR55 serisi motor yan görünüm şeması



Şekil III.6 HR55 serisi motor ön görünüm şeması

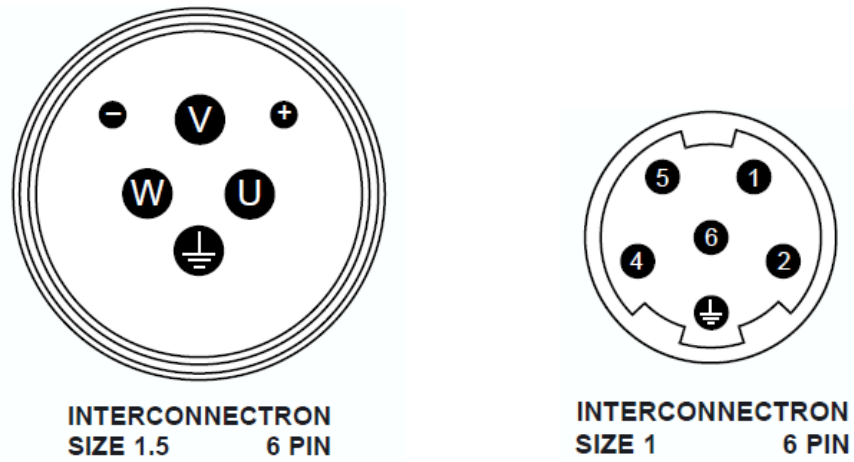
**HR55** serisi motorlarda **HR92** serisi gibi geri besleme elemanına ve frenleme sistemine bağlı olarak farklı uzunluklara sahiptir. **HR55C** modelinde uzunluk frenleme istemi eklendiğinde 176mm, frenleme sistemi olmadan ise 140mm'dir. Sistemde kullanılan tüm motorlar dört kutuplu ve sinüzoidal olarak çalışmaktadır[13].

**Tablo III.5** HR55C4-32S Teknik verileri

Değişken	Birim	Değer
Max. Akım	A	6.1
Daimi Akım (rms)	A	1.07
Max. E.M.K.	V	260
Max. Tork	Nm	1.4
Daimi Tork	Nm	0.4
Max. Mekanik hız	d/dk	8000
Atalet momenti	kgcm <sup>2</sup>	0.19
Sürtünme	Nm	0.0015
Sargı direnci	Ω	20
Sargı endüktansı	mH	17
Çalışma sıcaklığı	°C	40
Motor ağırlığı	kg	1.4

### III.1.2.2 Enerji bağlantı konektörleri

Servo motorlarda enerji bağlantılarının yanı sıra geri besleme elemanı içinde bağlatın uçları mevcuttur. Normal motorlara göre bağlantı kablo sayısı fazla olduğundan servo motorlarda bağlantı kutuları kullanıldığı gibi özel bağlantı konektörleri de kullanılabilir. Bu konektörler bağlantı kolaylığı sağladığı gibi güvenlik açısından da kullanışlıdır. Ayrıca hareketli sistemlerde özel bağlatın kabloları ve bağlantı konektörleri kullanmakta gerekebilir.

**Şekil III.7** Motor enerji besleme bağlantı uçları

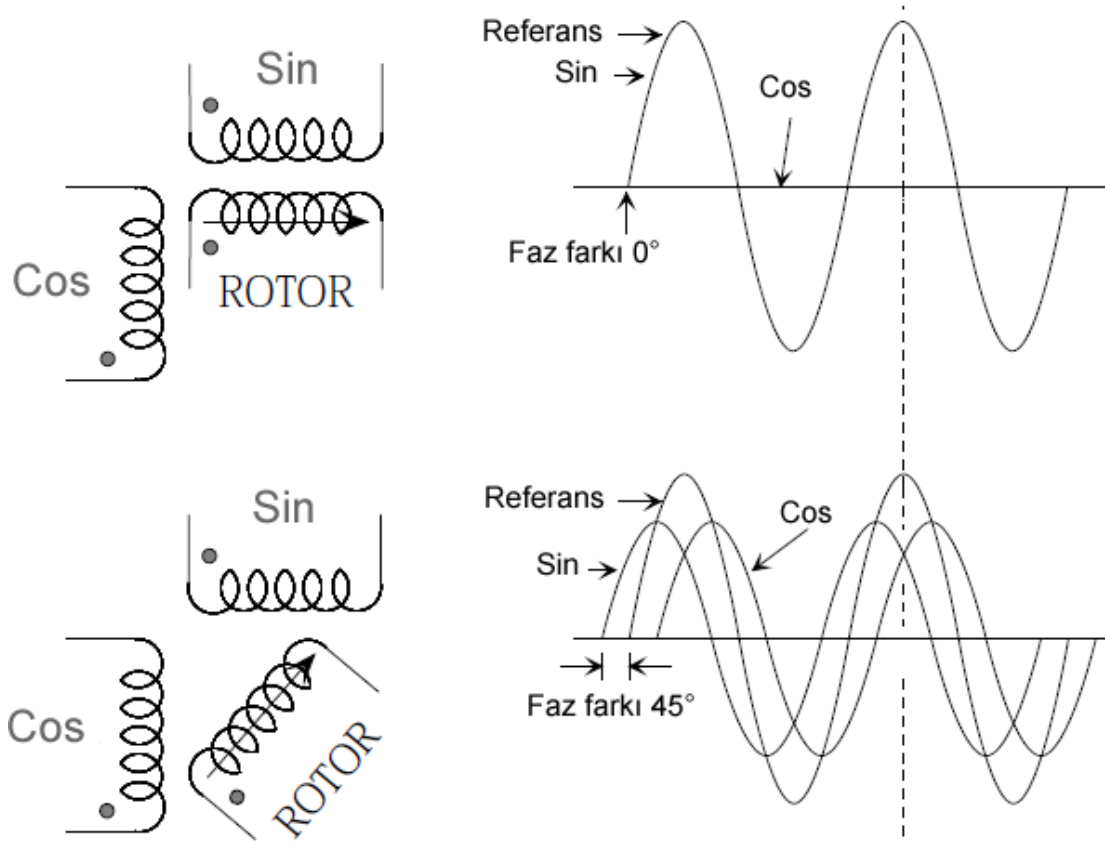
SEM motor teknolojileri servo motorların bağlantıları için farklı seçenekler sunmaktadır. Bunlardan hareketli bir sistemin tasarımına en uygun olanı Inter konektör bağlantı tipidir. Şekil III.7’de gösterilen bağlantı konektörleri çerçeve genişliği 115mm ve daha küçük olan motorlar için “size 1”, 142mm ve daha büyük çerçeveli motorlar için “size 1.5” tiplerini kullanılmaktadır. Konektör üzerinde motor sargı uçlarının yanı sıra dinamik frenleme için motora uygulanacak DC gerilim için iki adet bağlantı daha yer almaktadır. Bu konektörlerin bağlantı uçları Tablo III.6’da gösterilmiştir[13].

**Tablo III.6.** Inter konektör enerji besleme bağlantı uçları

PIN	Fonksiyon	PIN	Fonksiyon
U	MOTOR U	1	MOTOR U
V	MOTOR V	2	MOTOR V
+	Fren B+	4	Fren B+
-	Fren B-	5	Fren B-
W	MOTOR W	6	MOTOR W
⏚	MOTOR Toprak ⏚	⏚	MOTOR Toprak ⏚

### III.1.2.3 Geri besleme elemanları

Sistemde kullanılan tüm motorlar geri besleme elemanı olarak resolver kullanılmaktadır. Resolver küçük bir döner transformatör olarak tanımlanabilir. Bu transformatörün birincil ve ikincil sargıları arasındaki oran ve polarite farkı rotor pozisyonuna bağlıdır. İkincil sargılar birbirine 90°’lik konumlarda resolver statoruna yerleştirilmiştir ve birincil sargı ise resolver rotorunun üzerine monte edilmiştir. Aralarında 90° açı farklı bulunan ikincil sargılar sinüs ve kosinüs olarak adlandırılabilir. Birincil sargıya uygulanan referans AC gerilimi ile bu iki sargı arasındaki faz farkı rotorun pozisyonunu bildirir. Burada önemli olan sinüs ve kosinüs gerilimlerinin tepe değerleridir. Tepe değeri rotorun açısı ile değişir. Sonuçta sinüs ve kosinüs sargılarındaki değerler karşılaştırılarak rotor konumu belirlenmiş olur. Şekil III.8 resolver sargılarını ve bu sargılarda indüklenen gerilimleri göstermektedir[14].

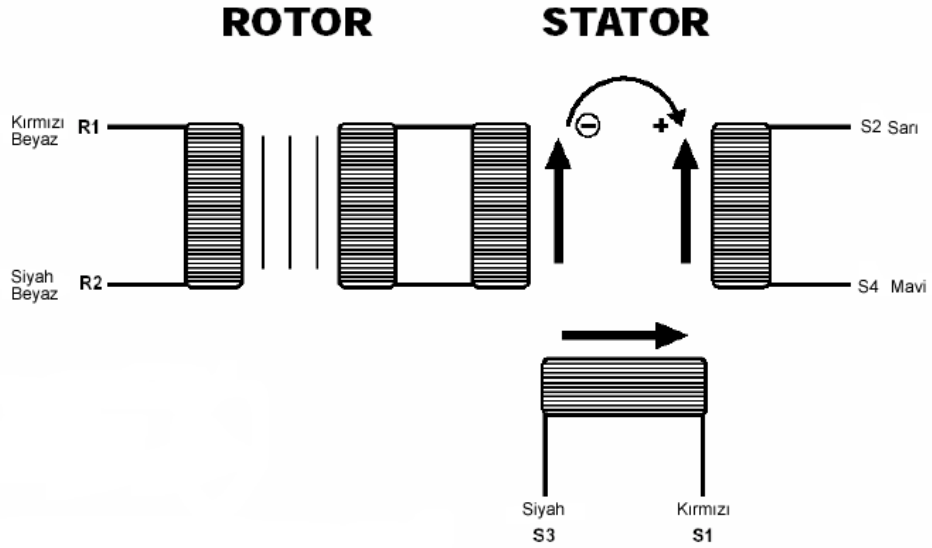


**Şekil III.8** Resolver çalışma prensip şeması

Kullandığımız motorlarda geri besleme elemanı olarak resolver seçilmesinin nedeni hem maliyet olarak encoder'a göre daha ucuz olması hem de çalışma ortamından daha az etkilenmesi olarak sıralanabilir. Encoderlar her ne kadar daha yüksek doğruluk seviyesine ve çözünürlüğe sahip olsa da çalışma sıcaklığı ve elektromanyetik uyumluluk gibi ortam faktörlerinden daha kolay etkilenebilmektedir. Tablo III.7'de resolver ve encoder standart özelliklerinin karşılaştırılması verilmiştir.

**Tablo III.7** Encoder ve resolver standart özellikleri

	Encoder	Resolver	Birim
<b>Standart çözünürlük</b>	24,000	16,384	darbe/devir
<b>Maksimum hız</b>	1.000.000	500,000	darbe/sn
<b>Doğruluk seviyesi</b>	2.5/1.0	15.0/7.0	arcminute
<b>Cevap süresi</b>	<1	50	ms
<b>Çalışma sıcaklığı</b>	0-100	-55-175	°C

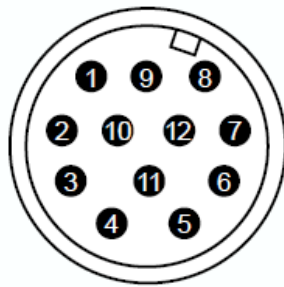


**Şekil III.9** Resolver sargı uçları

SEM motorlarda kullanılan Resolver bağlantıları için kullanılan konektörler enerji bağlantısı için kullanılanlardan farklı olarak 12 pinli ve standart büyüklüktedir. Ayrıca motorun termal korumasını sağlayan ısı sensör bağlantıları da geri besleme konektörü üzerinde bulunmaktadır. SEM motorlar için geri besleme bağlantı konektörü şekil III.10'da ve konektöre ait uç bağlantıları Tablo III.8'de verilmiştir.

**Tablo III.8** Geri besleme bağlantı uçları

PIN	Fonksiyon
1	STATOR 2. faz S2
2	STATOR 2. faz S4
3	
4	
5	
6	
7	ROTOR R2
8	ISIL SENSÖR
9	ISIL SENSÖR
10	ROTOR R1
11	STATOR 1. faz S1
12	STATOR 2. faz S3



**INTERCONNECTRON  
SIZE 1 12 PIN**

**Şekil III.10** Geri besleme bağlantı konektörü

### III.1.3 Sürücüler

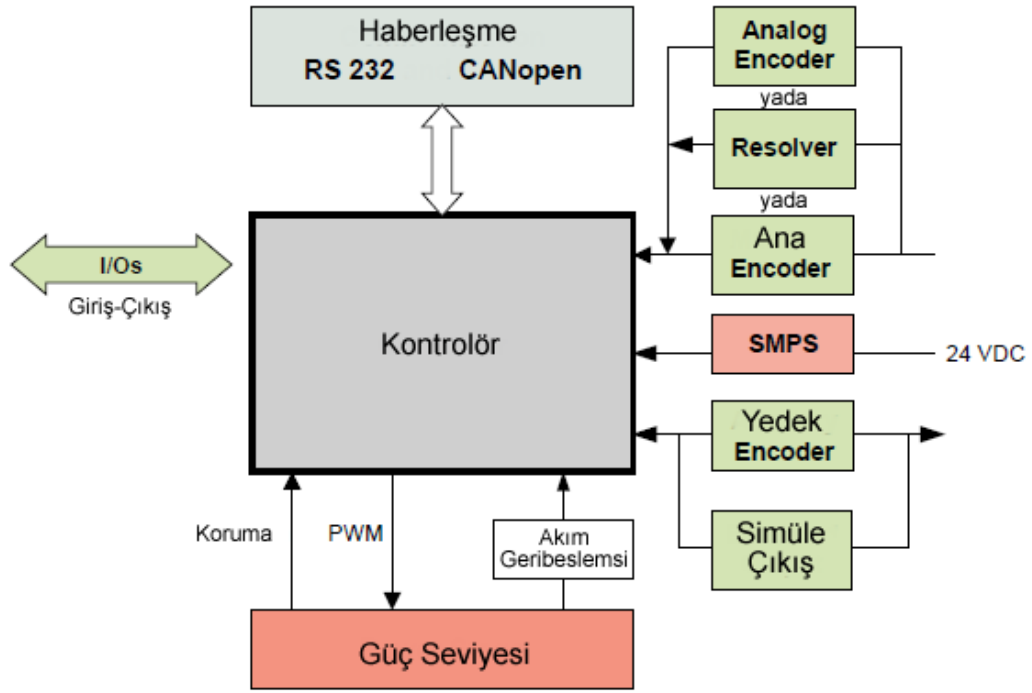
Kontrol teknolojilerinde kullanılan servo motorların harici bir sürücü devresi ile birlikte kullanılmaktadır. Bu sürücü devreleri komitasyon, hız ve konum kontrolü gibi görevlerin yanında haberleşme ve koruma gibi görevlerde üstlenmektedir. Yarı iletken teknolojilerindeki hızlı gelişmelere paralel olarak sürücü teknolojileri de hızla değişmiştir. Günümüzde en etkili ve hızlı sürücüler DSP ortamında geliştirilmektedir. Piyasada bulunan dijital servo sürücüler temelde aynı özellikleri paylaşmakla beraber kompakt yapı, hız ve etkinlik olarak birbirinden farklı özellikler taşımaktadır.

Bizim sistemimizde kullanılan ELMO hareket kontrol teknolojilerine ait SimplIQ sürücüler küçük yapıları ve gelişmiş ayar seçeneklerinden dolayı tercih edilmiştir. Ayrıca bu sürücüleri kontrol ederek çok eksenli hareket kontrolüne olanak sağlayan Maestro çok eksenli kontrolörde aynı ürün ailesinin bir parçasıdır.

#### III.1.3.1 Basson dijital servo sürücü

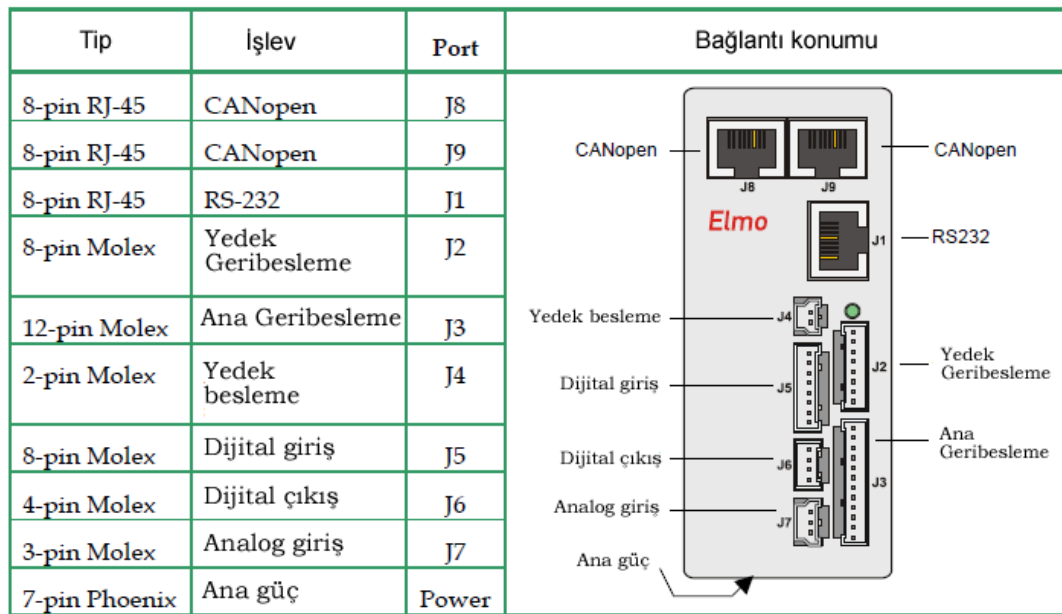
SimplIQ sürücü ailesinde tüm özellikler DSP ortamında geliştirilmiştir ve içerdiği yazılım üç parçaya bölünmüştür. Başlangıç önyükleme yazılımı (Boot software), DSP'nin dahili flaş hafızasına kalıcı olarak yerleştirilen yazılımdır. Sürücünün kullanımı süresince hiçbir şekilde değiştirilemez. Operasyonel yazılım (firmware), gerektiğinde üretici firmanın internet sayfasından yenilenebilir veya değiştirilebilir. Destek veritabanı (Supportive database), seri flaş hafıza birimine yüklenmiştir. Bu bilgi bankası kullanıcı tarafından veya fabrika üreticisi tarafından sağlanan programların ve uygulama bilgilerinin depolandığı, karakteriksel açıklamalar için kullanılan bir dosyalama sistemi olarak hizmet verir. Bu bölümdeki yazılımın düzenlenmesinde windows tabalı bir sistemde çalışan Elmo kompozitör yazılımı kullanılır. Şekil III.11'de sürücünün blok şeması verilmiştir[15].

Tasarlanan sistemde kullanılan sürücüler SimplIQ ailesinden **BAS-3/230R** katalog numarasına sahip dijital servo sürücüsüdür. Burada ilk üç harf sürücünün modelini, tirden sonraki rakam sürekli akımını, ayraçtan sonraki rakam nominal çalışma gerilimini ve gerilim değerinin sağındaki harfte geri besleme elemanı türünü belirtmektedir. Bassoon sürücüler direk olarak 110/230 VAC tek faz ile beslenebilir. Ayrıca sürücünün çalışması için 24 VDC bir güç kaynağı gereklidir. Sürücü üzerinde dijital giriş ve çıkışlarının yanı sıra birde analog giriş bulunmaktadır.



Şekil III.11 Bassoon sistem blok şeması

Sürücü tamamen dijital akım ve hız kontrolünün yanı sıra pozisyon kontrolü de yapmaktadır. Ayarlamalar ve program yüklemeleri için RS-232 ara yüzünün yanı sıra çok eksenli hareket kontrolünde kullanılmak üzere CANopen ara yüzü de yer almaktadır. Sürücünün üzerinde yer alan bağlantı noktaları şekil III.12’de gösterilmiştir.



Şekil III.12 Bassoon bağlantıları konektörleri

Bassoon sürücüler IP20 koruma sınıfına sahiptir. Yani 12.5mm'den daha büyük objelere karşı korumalı ama sıvılara karşı korumalı değildir. 0-40°C arasında çalışabilen sürücü 350gr ağırlığı ve 105x44x76mm boyutlarıyla hareket kontrolü için oldukça uygundur. Sürücü ile ilgili genel özellikler aşağıdaki Tablo III.9'da ayrıntılı olarak verilmiştir.

**Tablo III.9** BAS-3/230R sürücü genel özellikleri

<b>Özellik</b>	<b>Birim</b>	<b>Değer</b>
<b>En düşük besleme gerilimi</b>	VAC	30
<b>Nominal besleme gerilimi</b>	VAC	230
<b>En yüksek besleme gerilimi</b>	VAC	255
<b>En yüksek güç çıkışı</b>	W	710
<b>Verim</b>	%	>97
<b>DC sürekli akım(Trapezoidal)</b>	A	3
<b>AC sürekli akım(Sinüzoidal)</b>	A	2.1
<b>Akım limiti(RMS)</b>	A	2xIc
<b>PWM anahtarlama frekansı</b>	KHz	22 ±5%
<b>Anahtarlama tekniği</b>	-	Unipolar PWM
<b>Dijital giriş sayısı</b>	-	6
<b>Dijital çıkış sayısı</b>	-	2
<b>Analog giriş sayısı</b>	-	1

### III.1.3.2 Maestro çok eksenli kontrolör

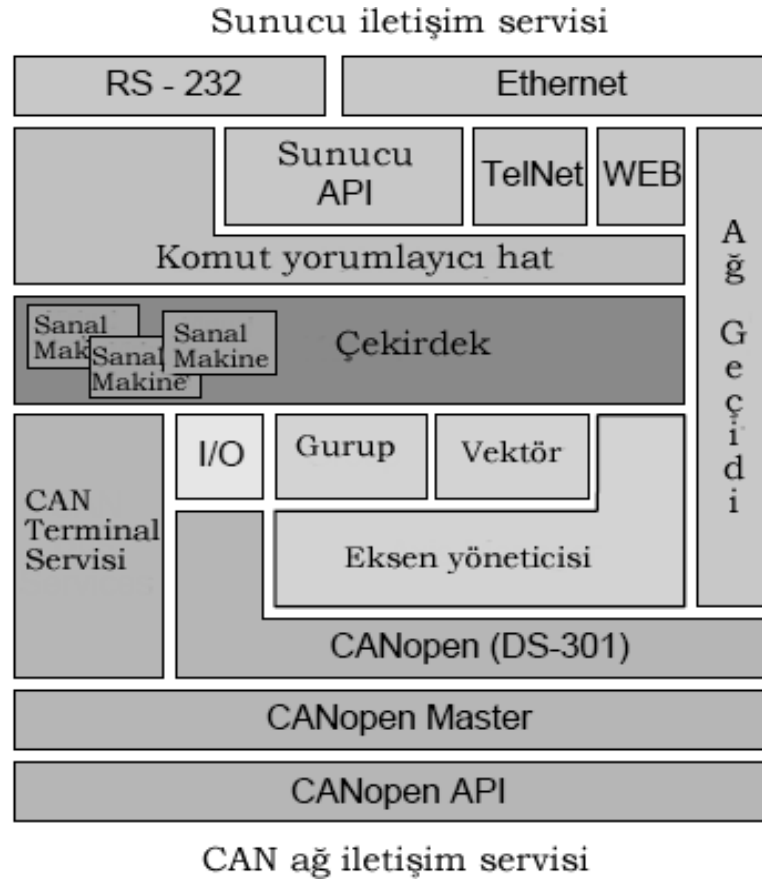
Elmo Maestro iletişim ağı tabanlı çok eksenli bir hareket kontrolörüdür. Maestro ve SimplIQ sürücüler hareket kontrolü sürecini paylaşarak iş yükünü dağıtır. Bu sayede hızlı ve etkili bir kontrol yapıldığı gibi 128 eksene kadar kontrole imkân vermektedir. SimplIQ sürücülerde olduğu gibi RS-232 ve CANopen ara yüzlerinin yanı sıra Ethernet üzerinde TCP/IP bağlantı olanağı da sunan Maestro işletim ağına ana cihaz olarak eş zamanlama, yönetim ve ayarlamalar gibi birçok işlev gerçekleştirebilmektedir. Ayrıca Ethernet-CAN geçişine olanak sağlayarak CAN üzerindeki sürücülere ve düğüm noktalarına direk erişim sağlamaktadır. Yine SimplIQ sürücülerde olduğu gibi Windows tabanlı bir bilgisayarda Elmo yazılımları ile her türlü düzenleme ve ayarlama yapılabilmektedir[16].

SimplIQ sürücülerde olduğu gibi Maestro çok eksenli kontrolörde de ürün kodu ürün hakkında kilit özelliklerle ilgili bilgi vermektedir. Kullandığımız çok eksenli kontrolör **MAS-10048-2C** modeli 48 eksene kadar kontrol ve 2 ayrı CAN bağlantı noktasına sahiptir. Kontrolörün diğer özellikleri Tablo III.10’da verilmiştir.

**Tablo III.10** Maestro çok eksenli kontrolör genel özellikleri

<b>Bileşen</b>	<b>Özellik</b>
<b>Hardware</b>	Bağımsız
<b>İşletim sistemi</b>	Gerçek zamanlı Elmo Kernel
<b>Programlama</b>	SimplIQ genişletilmiş dil Elmo Studio
<b>Eksen tipi</b>	Akıllı servo sürücüler CANopen DSP 402
<b>İşlem süresi</b>	1ms
<b>Protokoller</b>	CANopen: DS 301,DSP 401,DSP 402 Telnet
<b>Giriş-Çıkışlar</b>	8 Dijital giriş ve 8 Dijital çıkış 4 difrensiyal analog giriş Artımlı encoder 40 mega-counts/sn
<b>Ara yüzler</b>	RS-232 Fast Ethernet (10/100Mbps) 2 CAN bus
<b>Besleme gerilimi</b>	24 VDC $\pm$ 20%
<b>İşlemci</b>	300MHz Pentium uyumlu 16 bit ISA
<b>Hafıza</b>	64MB dâhili flaş hafıza 64MB RAM (128’e kadar arttırılabilir)
<b>Boyutlar</b>	153mm x 107mm x 51mm
<b>Ağırlık</b>	400 gram
<b>Çalışma sıcaklığı</b>	0-40°C

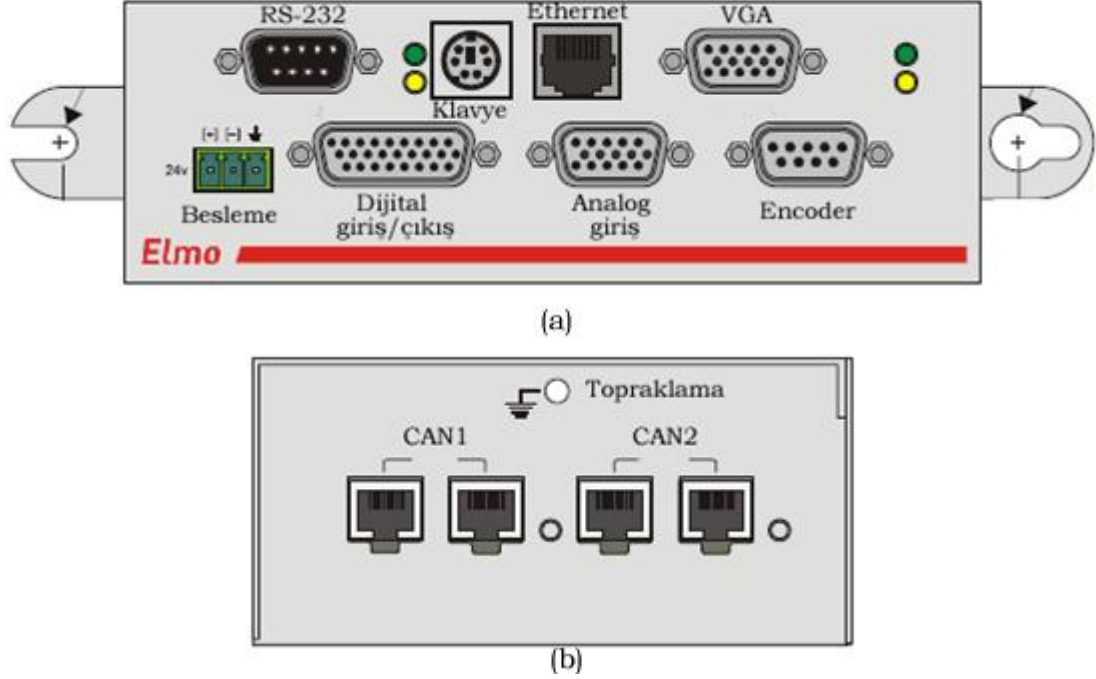
Maestro'nun sistem mimarisi beş ana grupta incelenebilir. İlk grup sunucu iletişim servisi (*Host Communication Services*) olarak adlandırılır ve içerisinde standart ara yüz ve protokolleri barındırır. Bu kısım Maestro'nun gerçek dünya ile iletişimini sağlar. İkinci grup komut yorumlayıcı hatır (*Command Line Interpreter*) ve sürücü veya kontrolöre ait değişik komutların anında yürütülmesini sağlar. Üçüncü grup ise çekirdek (*kernel*) bölümüdür. Bu bölüm kullanıcı programları çalıştırır. Dördüncü grup olan hareket yöneticisi (*Motion Manager*) tüm eksenlere gerekli bilgileri gönderip, eksenlerden gelen cevapları toplar ve bu sayede tüm eksenler arasında koordinasyonu sağlar. Son grup CAN iş ağının iletişim servisidir (*CAN Network Communication Server*) ve Maestro'nun çoklu eksenlerle haberleşmesini sağlar[16].



**Şekil III.13** Maestro sistem blok şeması

Sunucu bağlantısı TCP/IP veya RS-232 ara yüzleri ile sağlanabilir. Bu sayede gerekli operasyon komutları, veriler, durum bilgileri alınıp gönderilebildiği gibi komut yorumlayıcı, dosya yükleme ve indirme, komut işleme ve CAN ağ geçiş kontrolü gibi farklı işlemlerde gerçekleştirilebilir. Komut yorumlayıcı hat tüm eksen, vektör ve

grup komutlarını çalıştırabilir ve bu komutların durumları hakkında bilgi toplayabilir. Çekirdek ise kullanıcı programlarının çalıştırılması için farklı sanal makineler yaratarak çok eksenli kontrole olanak sağlar. Kontrolörün bağlantılarına ait şekiller Şekil III.14’de verilmiştir.



Şekil III.14 Maestro bağlantı konektörleri a) Üst görünüm b) Yan görünüm

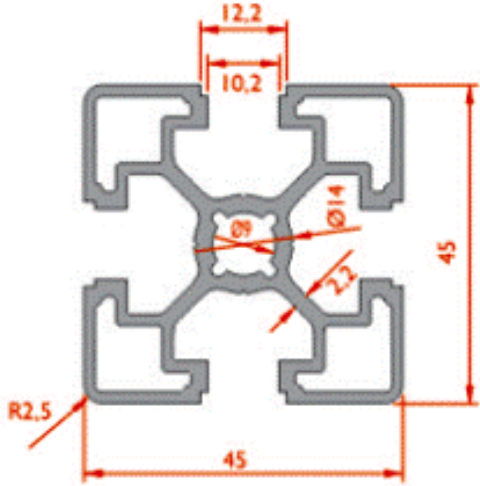
### III.2 SİSTEM TASARIMI

Üç eksenli hareketin kontrolü için ilk önce mekanik sistemin tasarlanması gerekir. Mekanik sistemde dairesel motor hareketini doğrusal harekete çevirecek olan doğrusal modüller yahut dairesel hareketin iletimi ve düzenlenmesi için çeşitli mekanik parçalar kullanılmalıdır. Daha sonra bu mekanik sisteme tahrik mekanizmasının montajı gerçekleştirilir. Mekanik sistem ve tahrik mekanizması uyum içinde olmalıdır. Son olarak tahrik mekanizmasını kontrol edecek olan sürücü sistemi kurulumu gerçekleştirilir. Bu sistem genellikle ayrı bir pano içerisinde hareketli sistemin uzağında yer alır. Fakat bazı uygulamalarda sürücü sistemi ve mekanik sistem iç içe olabilir. Ayrıca sürücü sistemi ve tahrik mekanizması arasındaki bağlantıların mekanik sistemin çalışmasını etkilemeyecek şekilde tasarlanmalıdır. Hareketli sistemlerde genelde mekanik kablo taşıyıcılar veya sürtünme temaslı konektörler

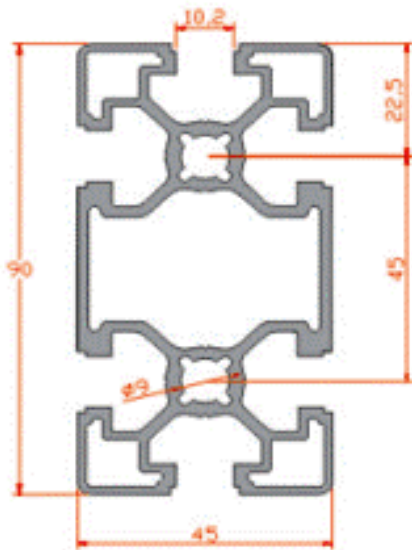
kullanılır. Bazı özel uygulamalar içinse radyo frekanslı veya optik kablosuz bağlantı yöntemleri de kullanılabilir.

### III.2.1 Gvde profili

Doğrusal hareket modllerinin sabitleneceđi iskelet iin alminyum profiller tercih edilmiřtir. Alminyum hem hafif hem sađlam olmasının yanı sıra yumuřak yapısı ile montaj sırasında da kolaylık sađlamaktadır. Kullanılan profiller 45x45mm ve 45x90mm olmak zere iki eřitirdir. Her iki profil zerinde de montaj kolaylıđı sađlaması iin her yzeyde oluklar aılmıřtır. Profillerin ađrılıđının azaltılması iin ii boř olacak řekilde ince et yapısıyla imal edilmiřtir. İstenildiđi takdirde dolgu malzemeleri ile profil ađrılıđı arttırılabilmektedir. Kullanılan profillerin gvde kesitleri řekil III.15 ve řekil III.16'da gsterilmiřtir.



řekil III.15 45x45mm profil kesit grnm

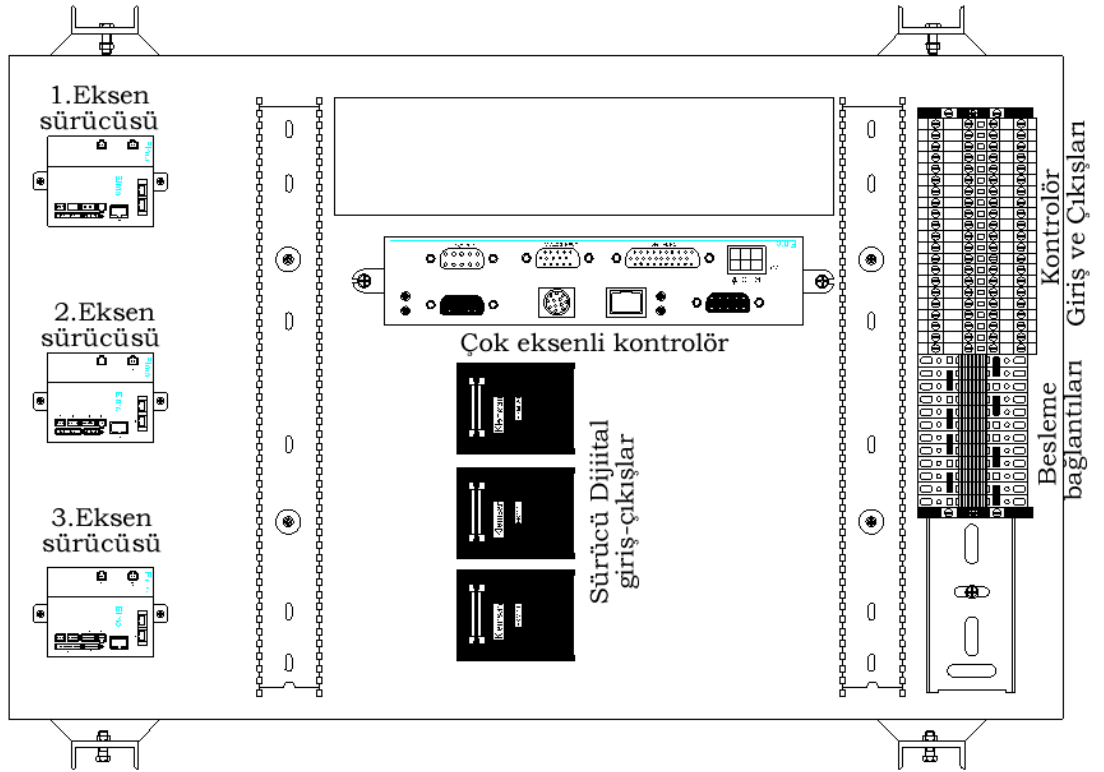


řekil III.16 45x90mm profil kesit grnm

### III.2.2 Kontrol panosu

Sistemde kullanılan SimplIQ sürücüler ve çok eksenli kontrolör mekanik sistemden ayrı olarak bir pano içerisinde yer almaktadır. Bu panonun içerisinde sürücüler ve kontrolörler arası saha veri yolu (fieldbus) iletişim ağı, dijital giriş ve çıkışlara ait konektörler, harici DC güç kaynağı ve koruma elemanları yer almaktadır. Sürücü ve kontrolör sistem arasındaki haberleşmeyi sağlayan saha veri yolunun mümkün olduğunca kısa tutulması ve yapının az yer kaplaması için gerekli düzenlemelerin yapılması gerekir.

Dijital servo sürücüler ve çok eksenli kontrolör uygun haberleşme ağı kurulabilecek şekilde pano içerisine yerleştirilirken aynı zamanda motorlara gidecek olan besleme kabloları da dikkate alınmalıdır. Kontrol katmanı ile enerji katmanını birbirinden olabildiğince ayrı tutmak önemlidir. Diğer yandan sürücü ve kontrolörün çalışması için gerekli olan DC gerilimin ve bu gerilimi sağlayan güç kaynağının da güvence altına alınması gerekir. Sistemde birçok sinyal aynı anda taşınacağı için kablolama ve yalıtıma dikkat edilmelidir. Cihazların ve panonun metal kısımlarının topraklanması da güvenlik açısından bir diğer önemli noktadır. Şekil III.17’de kontrol panosunun üstten görünüşü verilmiştir.

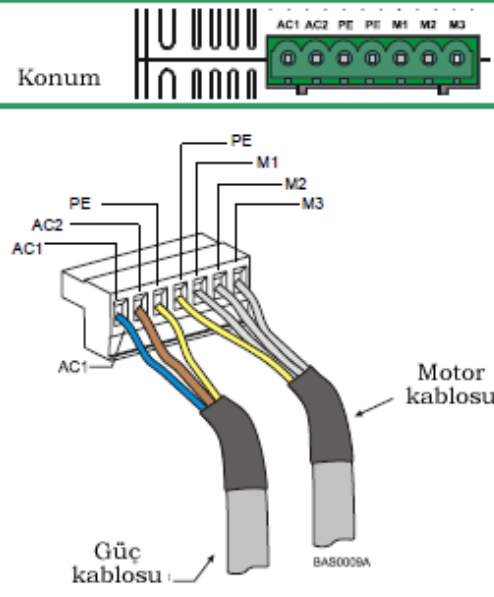


Şekil III.17 Kontrol panosu yerleşim şeması

### III.2.2.1 Güç ve kontrol bağlantıları

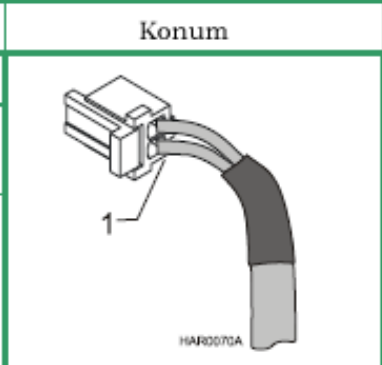
Sürücü ve kontrolörün pano yerleşimi yapıldıktan sonra gerekli tüm bağlantıların uygun kablo ve konektörler ile yapılması gerekir. Bassoon dijital servo motor sürücülerini tek faz üzerinden 230 VAC gerilim ile beslenir. Bu gerilim gerekli işlemlerden geçerek motor sargılarına verilecek olan akıma kaynak oluşturur. Sürücü güç bağlantı konektörü uç bağlantıları Tablo III.11’de gösterilmiştir.

**Tablo III.11** Sürücü güç bağlantıları

Pin	İşlev	Kablo	Konum	
AC1	Ana gerilim faz 1	Güç		
AC2	Ana gerilim faz 2	Güç		
PE	Toprak	Güç		
		AC Motor		DC Motor
PE	Toprak	Motor		Motor
M1	Motor fazı	Motor		N/C
M2	Motor fazı	Motor		Motor
M3	Motor fazı	Motor		Motor

Tablo III.12’de sürücünün çalışması için gereken 24VDC harici güç kaynağı bağlantısı gösterilmiştir. Aynı şekilde çok eksenli kontrolör içinde 24VDC bir gerilim gerektiğinden kullanılacak güç kaynağı tüm sistemi besleyebilecek güçte seçilmelidir.

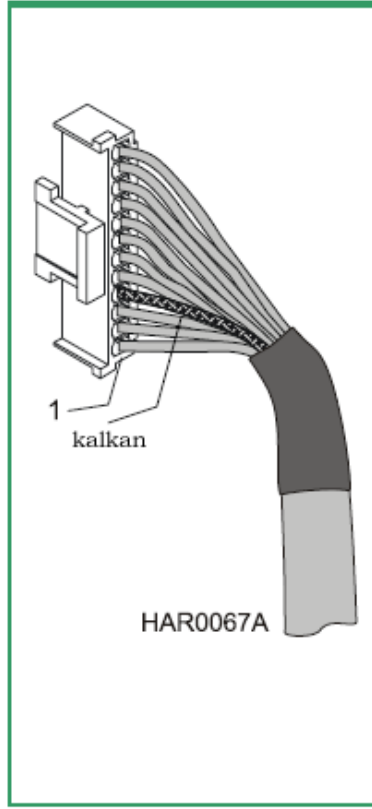
**Tablo III.12** Harici yardımcı güç bağlantıları

Pin	Sinyal	İşlev	Konum
1	+24VDC	+24 VDC harici güç kaynağı	
2	RET24VDC	Harici güç kaynağına geri dönüş ucu	

Sürücü ana ve harici yardımcı güç bağlantıları yapıldıktan sonra motordan gelen geri besleme bilgisinin sürücüye aktarılması için gerekli olan geri besleme bağlantısı yapılır. Bu bağlantı motorda kullanılan geri besleme elemanına ve sürücünün özelliğine göre yapılmalıdır. Daha önce resolverın bir birincil ve iki ikincil sargısı olduğundan bahsetmiştik. İkincil sargılardan biri Sine A diğeri de Cosine B olarak adlandırılmıştır. Birincil sargı ise referans bilgisinin geldiği bölümdür. Resolver bağlantıları Tablo III.13’de gösterilmiştir.

**Tablo III.13** Geri besleme elemanları bağlantısı

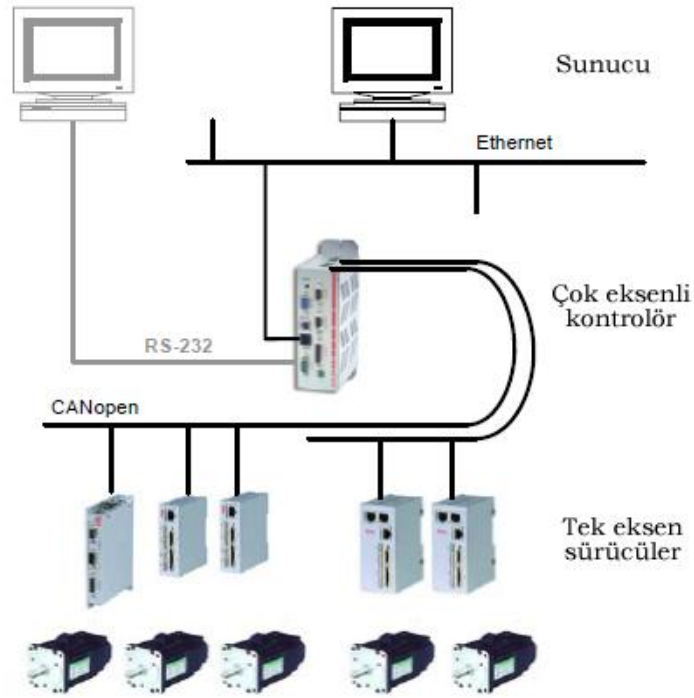
		Resolver	
		BAS XX/230R	
	Sinyal	İşlev	
1	Boş	-	
2	Boş	-	
3	Boş	-	
4	SUPRET	Kaynak dönüş	
5	SUPRET	Kaynak dönüş	
6	Boş	-	
7	R2	Vref Dönüş f= 1/TS, 50mA Max	
8	R1	Vref f=1/TS, 50mA Max.	
9	S4	Cosine B Dönüş	
10	S2	Cosine B	
11	S3	Sine A Dönüş	
12	S1	Sine A	



Sürücü giriş ve çıkışlarının yassı kablolarla bağlantı klemenslerine bağlanması ilerde olabilecek değişiklikler için oldukça yararlı olabilir. Bunun için pano üzerine yerleştirilen küçük bir bağlantı kutusu kullanılabilir. Sürücü üzerinde altı adet dijital giriş ve iki adet dijital çıkış ile bir adet analog giriş bulunmaktadır. Aynı şekilde çok eksenli kontrolör üzerinde bulunan giriş ve çıkışlarında harici bağlantı noktalarına taşınması uygulamada büyük kolaylık sağlayacaktır.

### III.2.2.2 Haberleşme bağlantıları

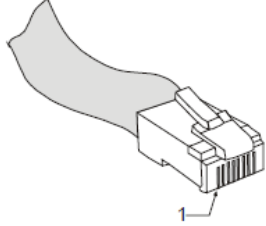
Dijital servo sürücü ve çok eksenli kontrolörün programlama ve ayarlamaları için RS-232 üzerinden yapılabildiği gibi saha veri yolu ve Ethernet üzerinden de cihazlara ulaşmak mümkündür. Çok eksenli kontrolör ve sunucu bilgisayar arasında direk veya bir iş ağı üzerinden Ethernet bağlantısı kurulabilir. Direk bağlantıda çapraz Ethernet kablosu, iş ağı üzerinden bağlantıda ise standart kablo kullanılır. Dijital servo sürücüler ise direk olarak RS-232 üzerinden sunucu bilgisayara bağlanabildiği gibi çok eksenli kontrolör üzerinden saha veri yolu ile de iletişim sağlanabilir. Şekil III.18'de haberleşme bağlantılarının genel bağlantı şeması gösterilmiştir[15].



Şekil III.18 Haberleşme bağlantıları

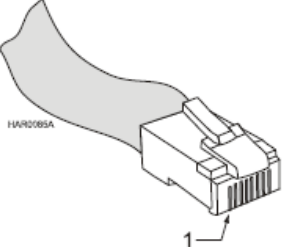
RS-232 bağlantısı için 26 veya 28 AWG veri kablosu kullanılabilir. En iyi sonuç için kablonun alüminyum kılıflı olmasına dikkat edilmelidir. Aynı şekilde kablonun ucuna takılacak olan RJ-45 türü konektöründe koruma kalkanı olmalıdır. RJ-45 bağlantı konektörünü RS-232 bağlantısında kullanmak için gerekli kablo ataması Tablo III.14'de verilmiştir[15].

**Tablo III.14** RS-232 bağlantısı için RJ-45 ataması

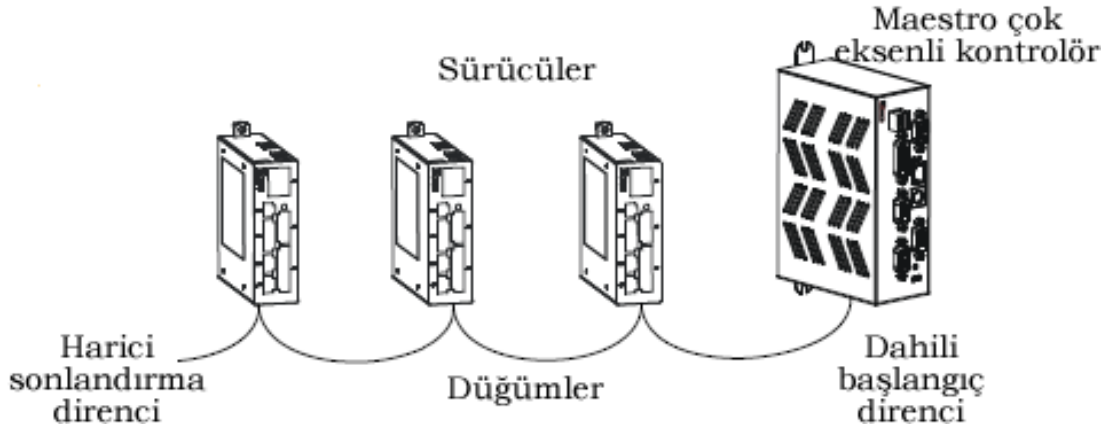
Pin	Sinyal	İşlev	Konum
1	—	—	
2	—	—	
3	Tx	RS-232 gidiş	
4	—	—	
5	COMRET		
6	Rx	RS-232 geliş	
7	—	—	
8	—	—	

Çok kesenli kontrolör ve sürücüler arasındaki haberleşme saha veri yolu üzerinden CANopen protokolü ile gerçekleştirilmektedir. Bu yüzden bu cihazlar arasındaki bağlantı yine aynı tür kablo ve konektörlerle fakat farklı sinyal atamaları ile yapılır. Saha veri yolu bağlantılarında cihazlar topolojilerle yerleştirilebilir. Eksen sayısı veya saha büyüklüğüne göre alternatif bağlantı türleri denenebilir. Önemli olan düğüm noktalarının, başlangıç ve bitiş noktalarının belirlenmesidir. RJ-45 konektörünü CAN bağlantısında kullanmak için gerekli kablo ataması Tablo III.15’de gösterilmiştir.

**Tablo III.15** CAN bağlantısı için RJ-45 konektör ataması

Pin	Sinyal	İşlev	Konum
1	CAN_H	CAN baskın hat bağlantısı	
2	CAN_L	CAN çekinik hat bağlantısı	
3	CAN_GND	CAN toprak bağlantısı	
4	—	—	
5	—	—	
6	CAN_SHLD	CAN kalkan bağlantısı (RJ dış koruma kılıfına)	
7	CAN_GND	CAN toprak bağlantısı	
8	—	—	

Tasarladığımız sistemde saha veri yolunun başlangıcı çok eksenli kontrolör olduğundan sistem kendi iç direnci ile başlangıç noktasını Maestro cihazı olarak belirlemiştir. Her bir eksenli kontrol eden sürücüler ise saha veri yolu üzerinde her bir düğüm noktasına yerleştirilmiştir ve saha veri yolunun son noktası için 120Ω’luk bir sonlandırma direnci kullanılmıştır. Sürücüler ve kontrolör arasındaki CAN bağlantısının basit görünümü Şekil III.19’da gösterilmiştir[16].



Şekil III.19 Saha veri yolu bağlantısı

### III.2.3 Kurulum ve ayarlar

Elektrikli servo sistemlerde kullanılan sürücüler yüksek işlem kabiliyetleri ve hızları ile sistemin kalbini oluştururlar. Sistemin verimi, hızı, hareket kabiliyeti, hata tanımlaması ve sürekliliği gibi unsular sürücü sisteminin özelliklerine bağlıdır. Küçük güçlerde analog sürücü sistemleri kullanılabilirken, yüksek güçlerde dijital sürücüler tercih edilmektedir. Ayrıca dijital sistemlerin kararlılık ve doğrulukları daha yüksektir. Sürücünün dijital olması beraberinde karmaşık elektronik devreler ve yazılımlar getirmektedir. Bu yüzden servo sistemlerin ayarlamaları ve programlaması önemlidir.

Üretici firmalar genellikle ürünleri için kendi özel programlama dillerini geliştirseler de, bu diller programcılıkta kullanılan popüler dillerin birer türevidir. Birçok firma kullanıcıların sistemlerini daha rahat tasarlayabilmesi için sürücülerin içine standart özellikler eklerler. Bu özellikler kullanıcının ihtiyacına ve bilgi seviyesine göre değiştirilebilir veya geliştirilebilir. Bu düzenlemeler için genellikle üretici firma tarafından özel yazılımlar hazırlanır. Bu yazılımlar aracılığı ile bilgisayar veya HMI sistemleri ile istenilen ayar ve programlamalar yapılır. Elmo dijital sürücü ve çok eksenli kontrolör Windows ortamında gerekli ayar ve programlamaları yapmak için bir kompozitöre sahiptir. Elmo kompozitör birçok aracın yanında şunları da barındırır[18]:

- Kurulum ve ayar araçları
  - Basit uygulama verileri ve sınırların girilmesi için bölümler
  - Giriş/Çıkış bağlantı noktalarıyla işlevlerin ilişkilendirilmesi için araçlar
  - Otomatik akım kontrolör ayarları
  - Otomatik komitasyon ayarları

- Elle ve otomatik hız kontrolör ayarları
- Elle ve otomatik pozisyon kontrolör ayarları
- RS-232 ve CAN arabağlantı üzerinden akıllı kullanıcı termili
- Sekiz sinyale kadar birebir izleme yapabilen gelişmiş faliyet kontrol kaydedicisi
- Uygulama veri tabanı bakım ünitesi
- Yazılım geliştirici kullanıcılar için gelişmiş bütünleşik gelişim ortamı
  - Editör
  - Derleyici
  - Kullanıcı yazılımı yükleme ve geriyükeleme
  - Hata ayıklayıcı

SimplIQ sürücü ve kontrolör sisteme ait değişkenleri ölçerken geri besleme elemanı tarafından üretilen darbeleri kullanır. Bu darbeleri kullanarak elde edilen zaman, konum ve hız gibi bilgiler ayarlamalarda kullanılır. Bu bilgiler konum kod çözücü gibi çevre birimleri tarafından sağlanır. SimplIQ sürücü ana ve yardımcı olmak üzere birbirinin aynısı iki konum kod çözücüsüne sahiptir. Her ikisi de doğru hız bilgisi için (Zamanlayıcı baştanbaşa A ve B ayarlanır) eş zamanlı olarak ayarlanmıştır. A konum kod çözücüsü kareleme yahut darbe/yön bilgisini ölçer. Çözücünün maksimum sayma oranı giriş filtresiz olarak 20MHz'dir. Eğer bir giriş filtresi uygulanmış ise maksimum pulse oranı azaltılır[17].

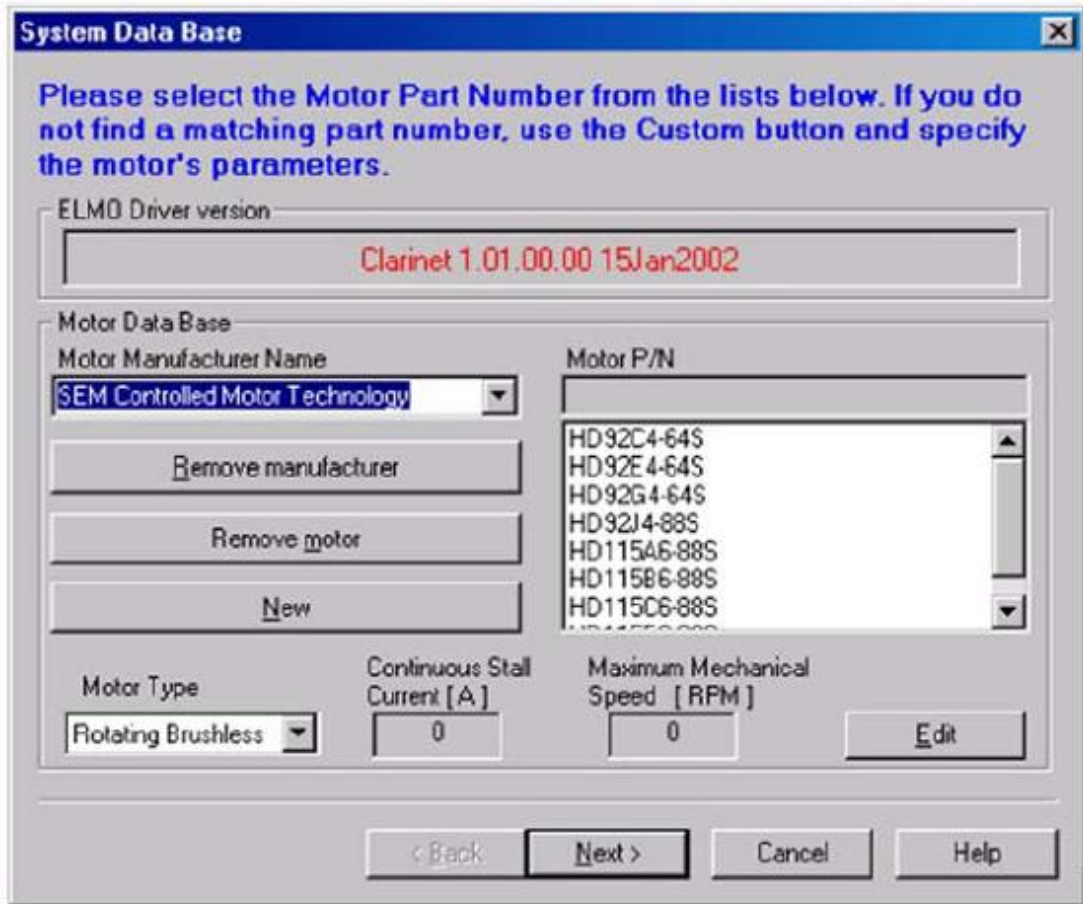
Akım ve gerilim gibi bilgiler içinse analog-dijital dönüştürücüler kullanılır. Tüm ölçümlerin çözünürlüğü 12 bit'dir. Pratikte son bit gürültüdür. Motor akımları bir ölçüm düzeneğinin sonucu olarak sapmasız biçimde ölçülür. SimplIQ sürücü devrelerinde elektronik hatalardan dolayı analog girişler sapmadan kaçınmaz. Bu sapma yazılım kullanarak yaklaşık 5 mV bir çözünürlükle düzeltilebilir. Yazılımsal olarak 5-10mV çözünürlük aralığındaki sapmaları düzeltilebilir[17].

### **III.2.3.1 Sürücü kurulumu ve ayarları**

Sürücünün ve motorun tüm bağlantıları gerçekleştirildikten sonra konum bilgisi, hız bilgisi ve komitasyon ayarları gibi bazı değişkenlerin sürücü tarafından motorun geri besleme elemanı aracılığı ile ayarlanması gerekir. Bu ayarlar yazılım tarafından otomatik olarak yapılabildiği gibi istenilen noktalarda kullanıcı müdahalesi ile istenilen

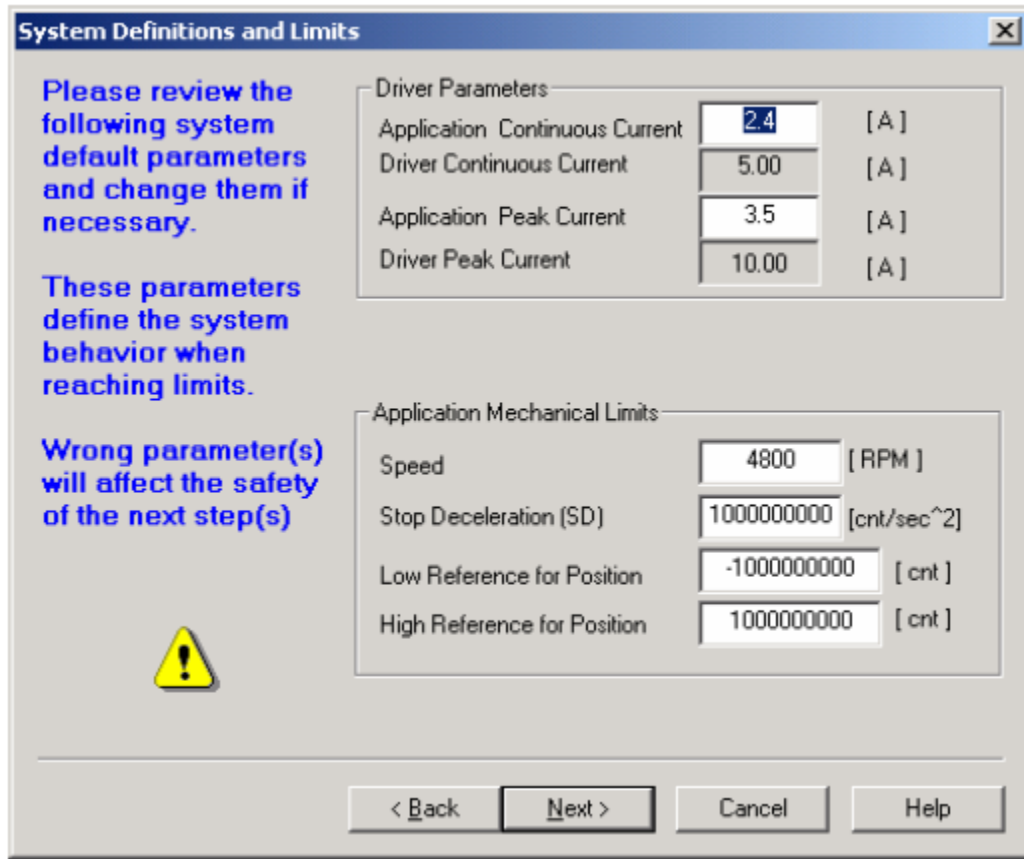
özellikler değiştirilebilir. Basitçe bu ayarlamalar motorun sürücüyü tanımlaması olarak tanımlanabilir.

Sürücü ile bilgisayar arasındaki haberleşme bağlantısı RS-232 veya CAN üzerinden gerçekleştirilebilir. Her iki seçenekte de bilgisayar üzerinde gerekli haberleşme ayarlamalarını yapmak gerekir. RS-232 için hız 19200 bps ve eşleşme kapalı konumda olmalıdır. CAN bağlantısı içinse düğüm numarası ve bağlantı hızı sistemde kullanılan değerlere ayarlanmalıdır. Bağlantı gerçekleştirildikten sonra yazılım veri bankasından kullandığınız motoru üretici firma ve ürün numarasını kullanarak seçebilirsiniz. Bu durumda motora ait tüm bilgiler otomatik olarak atanacaktır. Kullandığınız motor veri bankasında mevcut değilse düzenleme komutu ile yeni bir motora ait bilgileri kendinizde girebilirsiniz. Kompozitör üzerinden açılan pencerede motorun tipi, sürekli akım değeri ve maksimum mekanik hızı gibi bilgiler girilebilir. Motor bilgilerinin sürücüyü girilmesine ait ekran görüntüsü Şekil III.20'de görülmektedir[18].



Şekil III.20 Motor bilgilerinin girilmesi

Daha sonra kullanılan geri besleme elemanı türü ve bu elemanın çözünürlük bilgisinin girileceği ekran karşımıza gelecektir. Bu bilgiler motorda kullanılan geri besleme elemanının türüne göre değişiklik gösterir. Resolver için standart çözünürlük 16384 darbe/devir olarak belirlenmiştir. Bu işlemde sonra kompozitör sistemin limit bilgilerini girmemizi isteyecektir. Bu limitler sürücü için en yüksek sürekli ve tepe akım değerlerini geçemez. Ayrıca hız ve ivme gibi mekanik limitlerde aynı pencereden ayarlanabilir. Bu bilgiler sistemin güvenliği için önemli olduğundan dikkatle belirlenmelidir. Sistem limitlerinin belirlenmesine ait ekran görüntüsü Şekil III.21’de verilmiştir.

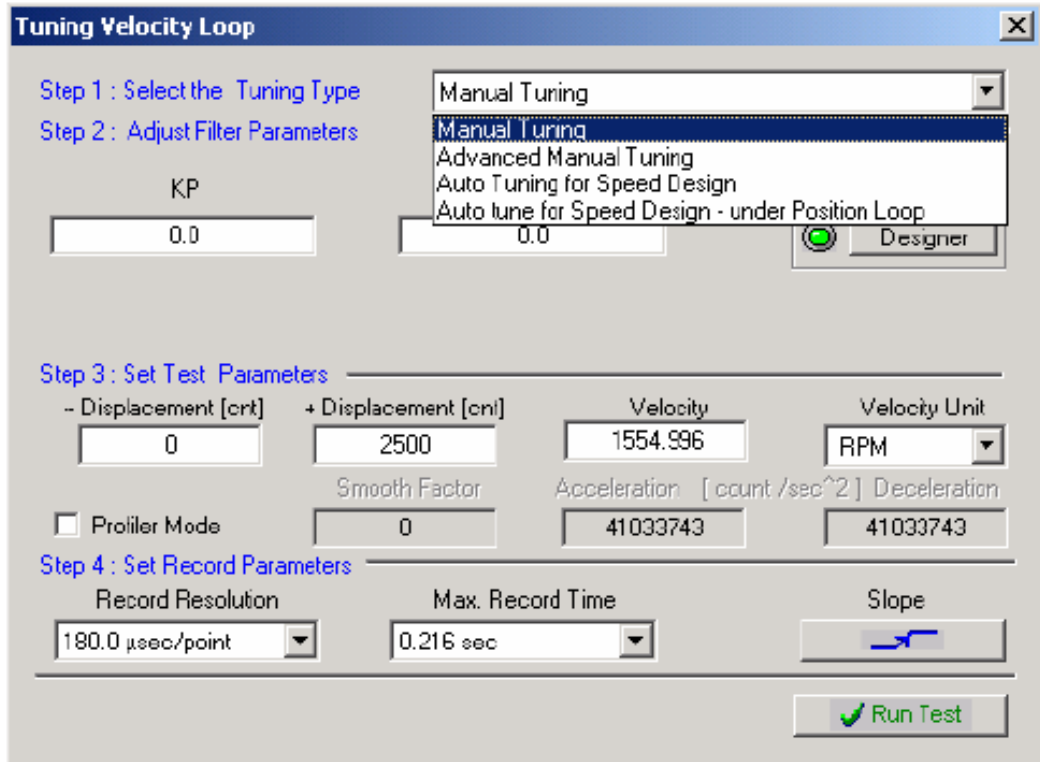


Şekil III.21 Sistem limitlerinin ayarlanması

Sürücü üzerindeki giriş ve çıkışların hangi amaçla kullanılacağı da bu ayarlamalar sırasında belirtilebilir. Dijital girişlere ani durdurma, başlatma, durdurma, yumuşak durdurma, yön belirleme ve başlangıç konumuna dön gibi görevler atanabilir. Aynı şekilde dijital çıkışlarda belirli durumlarda dışa sinyal vermek için ayarlanabilirler. Tüm bu veriler başarı ile girildikten sonra sırası ile akım döngüsü, komitasyon, hız döngüsü ve konum döngüsü ayarları yapılır[18].

Akım döngüsü ayarlamasında sürücünün motor akımını kontrol edebilmesi için gerekli ölçümleri yapar. Bunu motor sargılarına yüksek frekanslı akım göndererek omik ve endüktif dirençleri ölçerek ve dinamik geri dönüş bilgilerini kaydederek yapar. Komitasyon ayarları için motor ileri ve geri yönde düşük hızda bir tur atarak sürücüye gerekli bilgileri verir. Bu bilgiler fırçasız motorlardaki elektronik komitasyon için gereklidir. Akım döngüsü ve komitasyon ayarları sürücü tarafından otomatik olarak gerçekleştirilir ve kullanıcının müdahalesine izi verilmez. Fakat hız ve konum ayarlarında kullanıcı tasarladığı sistemin özellikleri doğrultusunda bazı değişiklikler yapabilir[18].

Hız döngüsü ayarlamalarında kullanıcı sistem cevap süresi, yumuşatma faktörü, hızlanma ve yavaşlama ivmesi gibi değerleri kendi ayarlayabildiği gibi sürücünün otomatik olarak da bu değerleri atamasını sağlayabilir. Ayrıca farklı tasarımlar için farklı otomatik ayarlamalarda yapılabilir. Hız döngüsü ayarlarına ait ekran görüntüsü Şekil III.22’de görülmektedir.



Şekil III.22 Hız döngüsü ayarlamaları

Konum döngüsü ayarlarında ise cevap hassasiyetini etkileyen kazanç, adım aralığı, hız, yumuşatma faktörü ve ivme gibi değerler girilebilir. Hız döngüsü

ayarlarında olduğu gibi burada da istenilen bilgiler kullanıcı tarafından girilebildiği gibi sürücü tarafından otomatik olarak belirlenebilir. Başlangıç olarak otomatik ayarlar bizim için yeterli olabilir fakat doğruluğun ve hızın önemli olduğu ileri düzeyde sistem tasarımlarında bu değerlerden bazılarının gerekli hesaplamalar sonucunda kullanıcı tarafından girilmesi gerekebilir. Konum döngüsüne ait ayar parametreleri Şekil III.23'de gösterilmiştir.

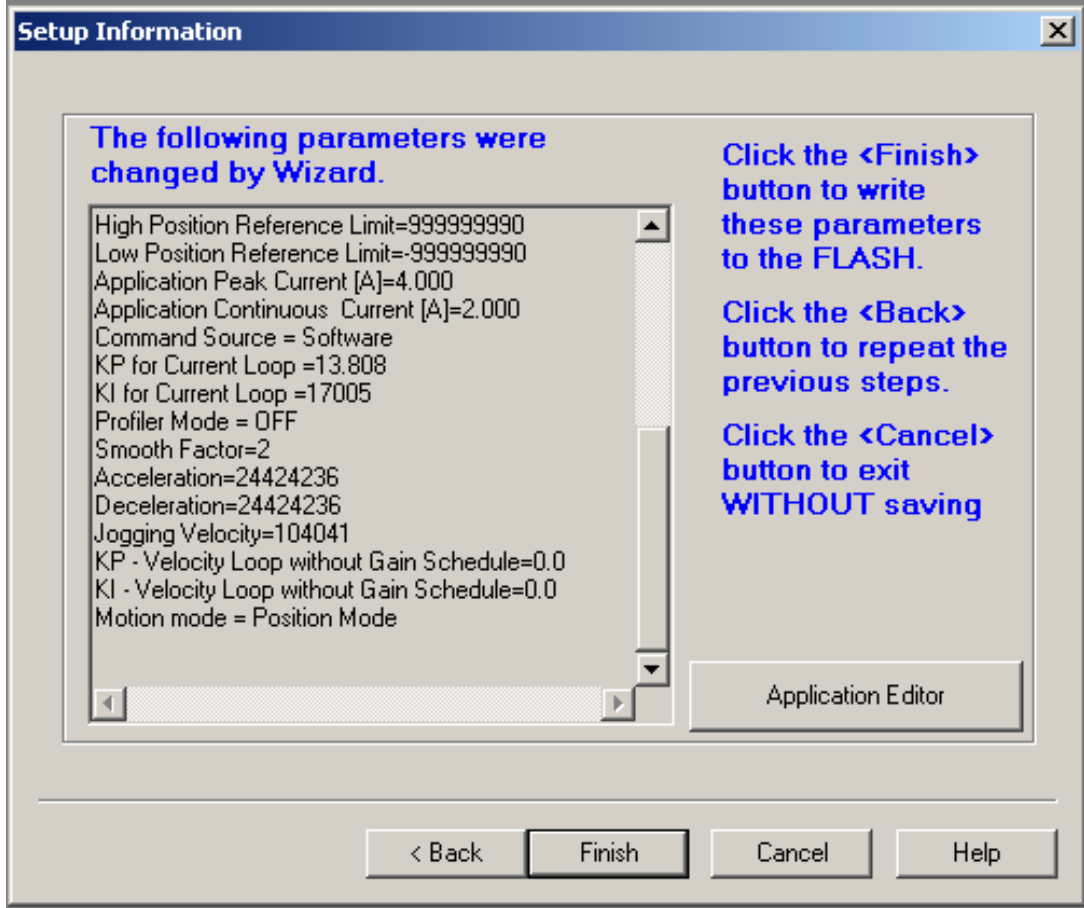
The image shows a software window titled "Tuning Position Loop" with a close button in the top right corner. The window is organized into four steps:

- Step 1: Select the Tuning Type**: A dropdown menu is set to "Manual Tuning".
- Step 2: Adjust Filter Parameters**:
  - Input fields for KP, KI, and KD are all set to "0.000".
  - An "Advanced Filter" section has a radio button labeled "ON" which is selected, and a "Designer" button next to it.
- Step 3: Set Test Parameters**:
  - "Step [cnt]" is set to "1000".
  - "Speed" is set to "3.750".
  - "Speed Unit" is set to "RPM".
  - "Smooth Factor" is set to "1".
  - "Acceleration [count/sec<sup>2</sup>]" is set to "4517172".
  - "Deceleration" is set to "4517172".
- Step 4: Set Record Parameters**:
  - "Record Resolution" is set to "250.0 μsec/point".
  - "Max Record Time" is set to "0.300 sec".

At the bottom right, there is a "Run Test" button. At the very bottom, there are four buttons: "< Back", "Next >", "Cancel", and "Help".

Şekil III.23 Konum döngüsü ayarlamaları

Tüm bu test ve ayarlamalar sürücünün motorun çalışma karakteristiği hakkında bilgi sahibi olması için önemlidir. Bu yüzden tüm ayarlar sürücünün belleğine kaydedilerek daha sonraki uygulamalarda hep bu bilgiler kullanılır. Eğer sistemde herhangi bir değişiklik gerçekleşirse bu değişkenlerinde değiştirilmesi veya otomatik olarak yenilenmesi gerekir. Ayrıca bu parametre değerleri Elmo uygulama editörü ile yeniden düzenlenebilir. Tüm sistem parametreleri ve parametrelerin sürücü hafızasına kayıt edilmesine ait ekran görüntüsü Şekil III.24'de görülmektedir.

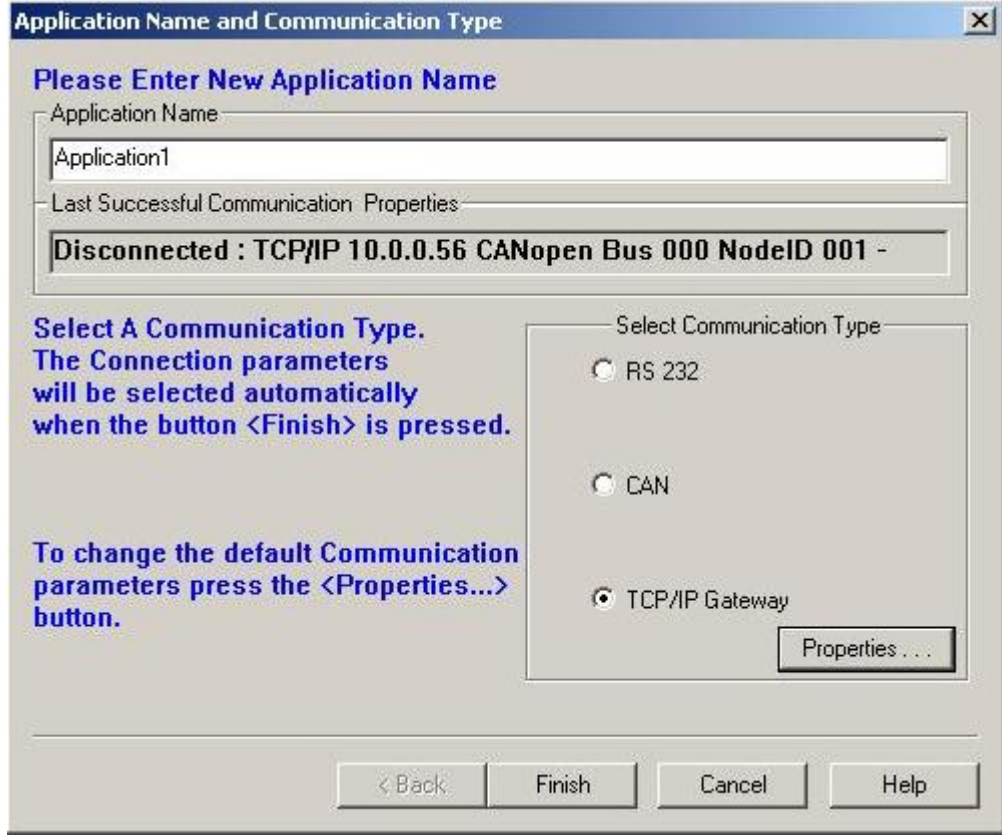


Şekil III.24. Ayar parametrelerinin sürücüye kaydedilmesi

### III.2.3.2 Çok eksenli kontrolör kurulumu ve ayarları

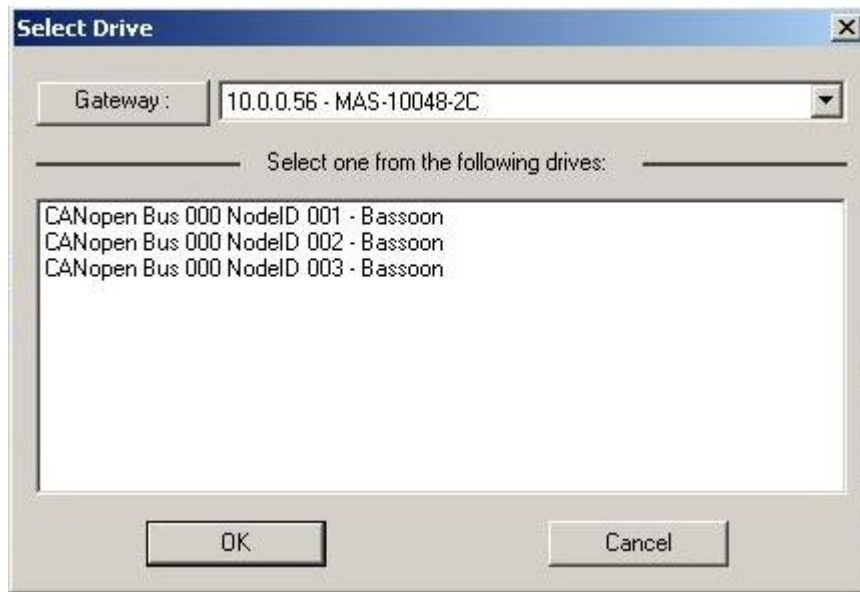
Maestro çok eksenli kontrolör sunucu bilgisayarla RS-232 veya Ethernet üzerinden TCP/IP protokolü ile iletişim kurabilir. Birçok bileşene sahip bir iş ağında Ethernet bağlantısı hız ve bağlanabilirlik açısından daha verimlidir. Kontrolör ile sunucu bilgisayar birbirine direk olarak bağlanabildiği gibi bir Ethernet bağlantı kutusu üzerinden de bağlanabilir. Direk bağlantıda çapraz kablo kullanılırken, dağıtıcı veya çoklayıcı üzerinden yapılan bağlantıda standart Ethernet kablosu kullanılır.

Kontrolörü sunucu bilgisayara bağlamak için sürücüler için kullanıla kompozitör kullanılabilir. Yeni bir uygulama başlatıldığında açılan iletişim menüsünden TCP/IP bağlantısı seçilerek bu işlem gerçekleştirilebilir. Bu işlem sonunda kontrolör üzerinden CAN üzerinde bulunan tüm düğümlere ve bu düğümlerde tanımlı olan cihazlara erişmek mümkündür. Haberleşme bağlantısı seçim ekranı Şekil III.25'de görülmektedir.



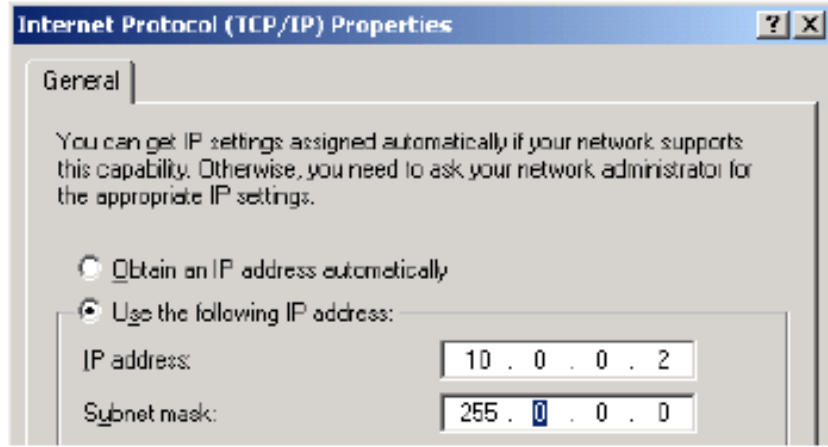
Şekil III.25 Ethernet üzerinden TCP/IP bağlantısının gerçekleştirilmesi

Bağlantı sağlandıktan sonra kompozitör üzerinden istenilen düğüm ve o düğümde tanımlı cihaza ulaşılabilir. CAN üzerindeki cihazlar Şekil III.26’da gösterilen menüden seçilebilmektedir.



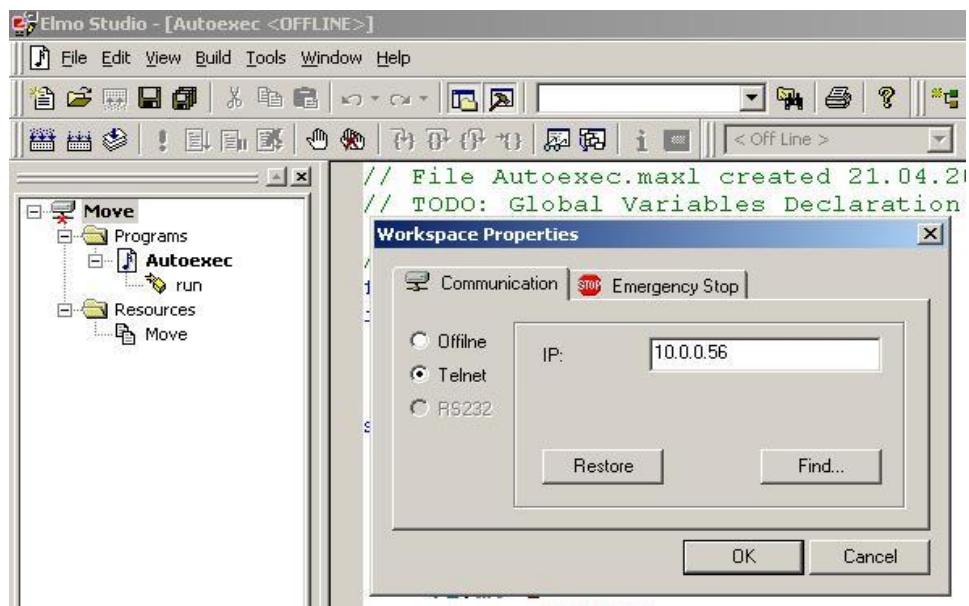
Şekil III.26 Kompozitör üzerinden CAN nesne bağlantıları

Çapraz kablo kullanılarak yapılan direkt bağlantıda sunucu bilgisayarın internet protokolüne ait IP değerini değiştirmek gerekebilir. Maestro cihazlar için fabrikada tanımlanmış olan IP adresi 10.0.0.56 olarak belirlenmiştir. Sistemde birden fazla cihaz varsa bu adreslerin son hanesindeki rakam değişebilir[16].



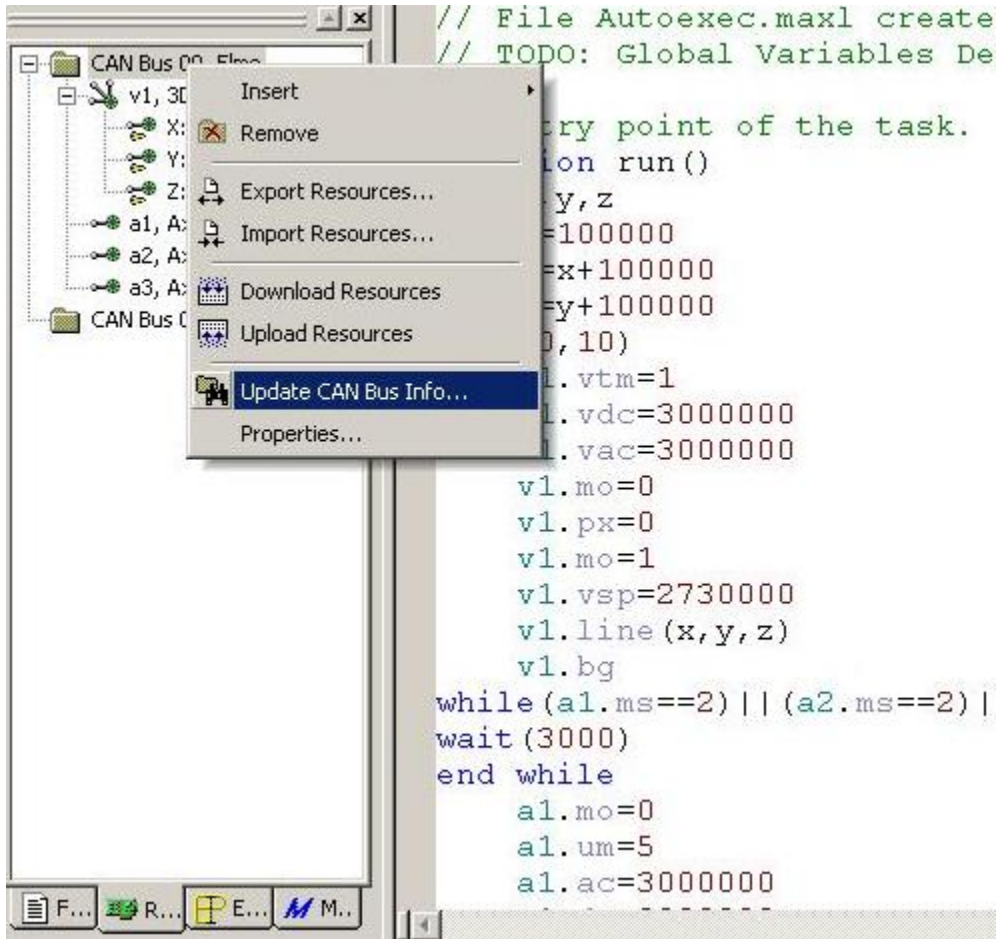
Şekil III.27 İnternet protokol numarasının girilmesi

Sunucu bilgisayar ile kontrolör arasındaki bağlantı sağlandıktan sonra kompozitörün araçlar menüsünde yer alan Elmo Studio kullanılarak gerekli tüm ayar ve değişiklikler yapılabilir. Çalışmanın düzenlenmesi ve programların kontrolöre yüklenebilmesi için telnet veya RS-232 bağlantısının sağlanmış olması gerekir. Bağlantı kontrolü Şekil III.28’de görüldüğü gibi çalışma alanı ayarlarından gerçekleştirilir.



Şekil III.28 Elmo Studio çalışma alanı bağlantısı

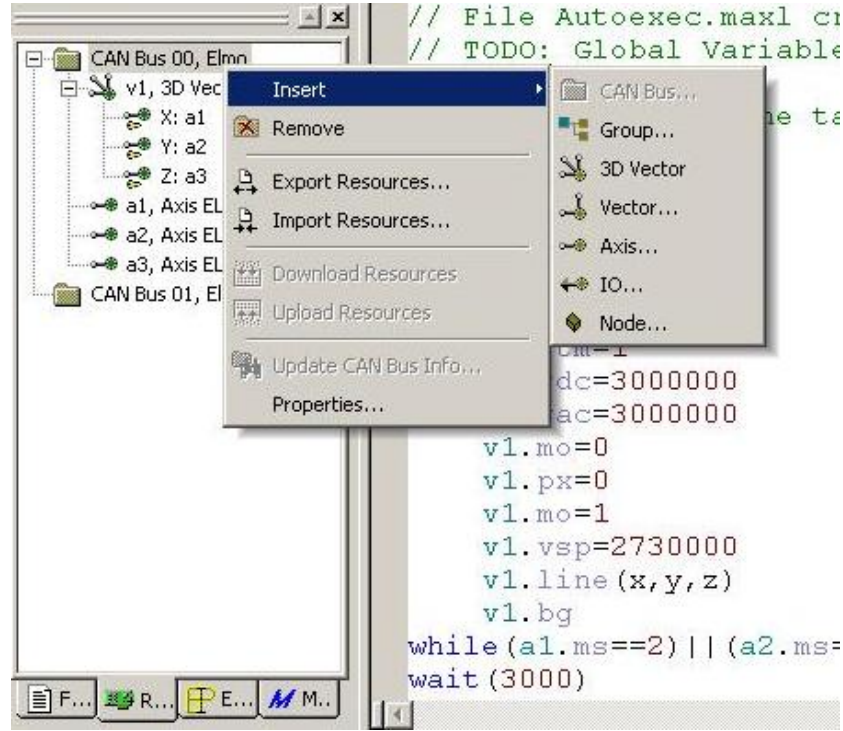
Studio'da yeni bir çalışma alanı başlatıldığında ilk olarak kontrolör ve bilgisayardaki kaynakların eşlenmesi gerekir. Bu kaynak kontrolör üzerindeki bağlı durumda bulunan elemanların bilgilerinden oluşur. Bu bilgiler eşleşmediği takdirde herhangi bir düzenleme veya programlama yapılamaz. Bilgiler sistemden indirilebildiği gibi kullanıcı tarafından da ayarlanarak sisteme yüklenebilir. Önemli olan kaynak bilgilerinin uyuşmasıdır. Kaynak bilgileri Şekil III.29'da görüldüğü gibi CAN Bus üzerinden eşlenebilir.



Şekil III.29 Kaynak bilgileri eşlenmesi

Kaynak bilgileri kullanıcı tarafından elle girilirken her bir düğümün kimlik bilgisi (Node ID), eksen adı ve iletişim tipi belirtilmelidir. Vektör bilgilerinde ise vektöre ait her bir eksen daha önceden tanımlanan eksenlerden seçilerek yeni bir vektör oluşturulur. Bu bilgiler girilirken dikkat edilmesi gereken birbirini tekrar eden isimler kullanmamak ve CAN kimlik numaralarının doğru girilmesidir. Eğer sistemde çok fazla eksen ve vektör kullanılıyorsa gruplara bölünerek kontrol edilebilirler.

Kaynak bilgilerini düzenlemek için Şekil III.30'da görüldüğü gibi ekle (Insert) seçeneği kullanılır.



Şekil III.30 Eksen ve vektör bilgilerinin girilmesi

Kaynak bilgileri girildikten ve güncellendikten sonra Elmo Studio'nun solundaki kaynak görünümü menüsünden takip edilebilir. Programlama sırasında komutlar bu kaynaklara göre belirlenir. Eğer sisteme yeni bir eksen veya vektör dâhil edilmişse kaynaklara da eklenmesi gerekir ki yazılan programda tanımsız olarak hata vermesin. Tanımlı CAN'ler üzerine sağ tıklanarak bu bilgiler kontrolöre yüklenebildiği gibi kontrolörde var olan kaynak bilgileri de aynı şekilde Elmo Studio'ya aktarılabilir. Bağlantı gerçekleştirilip kaynak bilgileri yüklendiğinde Maestro programlama için hazırdır. Programlama için SimplIQ'nun ortak yorumlayıcı dili kullanılır.

Şekil III.31 a'da eksen bilgilerinin girilmesi için açılan pencerenin, b'de ise vektör bilgilerinin girildiği pencere görülmektedir. Bu bilgilerin doğru girilmesi programlama aşamasında karşılaşılabilecek sorunları engellediği gibi program satırlarının okunmasını da kolaylaştırabilir.

(a)

(b)

Şekil III.31 a) Eksen bilgileri b) vektör bilgileri

### III.3.PROGRAMLAMA

SimplIQ sürücü kullanıcıya kurulum, komut gönderme ve durum sorgulama gibi işlemleri gerçekleştirmesini sağlayan bir iletişim dili kullanır. Daha öncede bahsedildiği gibi sürücü ile haberleşmek için iki method kullanılabilir. Birincisi, sürücüden doğrudan bir cevap almak veya sürücüye komutları transfer etmek için bir haberleşme arayüzü (Rs-232 veya CANopen) kullanmak. Bu method on-line iletişim gerektirir ve sunucu ile sürücü arasında sıkı bir ilişki kurulur. RS-232 ve CANopen haberleşme standartları ve fiziksel yapısı farklı komut sözdizimini gerektirir. Bu bölüm RS-232 ve CAN OS sözdizim temeline dayalı sürücü dilin tanımlamaktadır. İkincisi, sürücü dilini kullanarak bir program yazmak ve sürücü hafızasına yüklemektir. Sürücü daha sonra sunucu desteksiz veya minimum destekle programı çalıştırabilir[18].

#### III.3.1 İfade ve Operatörler

Sürücü dili matematiksel, mantıksal ya da koşullu ve bağlı işlemleri iki ya da daha fazla değişken arasında destekleyebilir. Değişken ve operatörler birçok değişik biçimde bir araya getirilerek ifadeler yaratılır. SimplIQ sürücü 32 bit tamsayıları (integer) ve 32 bit floating-point (floats) sayıları kullanır. Yazım girişlerinde virgüllü ve üslü ifadeler içeren sayılar “floats” olarak yorumlanır. Diğer sayılar tamsayı olarak yorumlanır. Tam sayılar için aralık  $[-2,147,483,648...2,147,483,647]$ 'dir. Eğer bir tam sayı aralık dışında ise, hata olarak yorumlanır. (Örnek olarak 2,147,483,648, sayısı SimplIQ sürücü tarafından “Bad Command Format error” olarak yorumlanır.) En düşük integer -2,147,483,648 komut yorumlayıcısında doğrudan girilemez. Bu sayı

anacak onaltılık sayı tabanında (hexadecimal) girilebilir. (0x80,000,000.) Pozitif tam sayılar, ondalık (decimal) veya onaltılık (hexadecimal) olarak yazılabilir. Örneğin; 16 sayısı 0x10'a eşittir. Bir tam sayı değeri her zaman alta yuvarlanır. Mesala,  $5/2=2,5$  ve bu değer 2 olarak alınır[18].

Ondalık ve üslü sayılar (floats) için aralık  $[-1e20...1e20]$  dir. Sayılar üslü veya üstsüz olarak yazılabilir.  $2.5e4$  sayısı 25,000.0 eşittir. Yazarken 25,000 yazılamaz çünkü bu sayı tam sayı olarak algılanır. SimplIQ ondalıklı ve üslü sayıları IEEE standartlarına göre altı ondalık rakam olarak düzenler. Örnek olarak, 12,345.0 tam bir "floating" ifadedir. 1,234,568.0 sayısı SimplIQ sürücü tarafından 12,345,680.0 olarak algılanır. SimplIQ sürücü  $10^{20}$  den daha büyük sayıları değerlendiremez[18].

İfadeler mantıksal, ilişkisel ve aritmetik operatörlerin kombinasyonlardan oluşabilir. İfade değerlendirilmesi öncelik seviyesine göre gerçekleştirilir. Her öncelik seviyesinde, operatörler eşit önceliğe sahiptir ve soldan sağa değerlendirilir. Örneğin  $a*b/c$  hesaplanırken  $(a*b)/c$  şeklinde hesaplanır. Tablo III.16'da SimplIQ sürücü dilinde kullanılan matematik ve mantık operatörleri listelenmiştir. Tabloda ayrıca en yüksekten en düşük öncelik seviyelerine doğru düzenlenen işlem öncelikleri de belirtilmiştir.

**Tablo III.16** Matematiksel ve mantıksal operatörler

Operator	Tanım	Öncelik
~	Bir değişkenin bit bit değili	17
!	Mantıksal inkar	17
-	Birli eksi	17
%	İki tam sayının bölümünden kalan	16
*	İki değişkenin çarpımı	16
/	Soldaki değişkenin sağdakine bölümü	16
+	İki değişkenin toplamı	15
-	Sağdaki değişkenin soldakinden çıkarılması	15
<<	Bit bit sola kaydırma	14
>>	Bit bir sağa kaydırma	14
<	Mantıksal den küçük	13
<=	Mantıksal den küçük veya eşit	13
>	Mantıksal den büyük	13
>=	Mantıksal den büyük veya eşit	13

**Tablo III.16 (Devam)** Matematiksel ve mantıksal operatörler

==	Mantıksal eşitlik	12
!=	Mantıksal eşitsizlik	12
&	İki değişken arasında bit bit AND	11
	İki değişken arasında bit bit OR	9
&&	Mantıksal AND	8
	Mantıksal OR	7
=	Görev atama	
()	İfade belirtmek ve işlev çağırma için	
[]	İçerik dizilimi ve işlev döngüsü katı	

Birçok aritmetik operatör hem tamsayı hem de ondalıklı olarak kullanılır. Tamsayılar arasındaki bir aritmetik operasyonda tamsayı üretilir. Ondalıklı sayılar arasında veya bir tamsayı ile bir ondalıklı sayı arasında bir ondalıklı sayı üretilir.

**Örnek:**

1+2	(Sonuç 3, Tamsayı.)
1+0x10	(Sonuç 17, Tamsayı. Hexadecimal 0x10 sayısı tamsayı olarak algılanmıştır.)
1+2.0	(Sonuç 3.0, Ondalıklı.)
2.1+3.4	(Sonuç 5.5, Ondalıklı.)

İki tamsayı arasındaki toplama ve çıkartma işlemlerinin sonuçları tamsayı aralığını [-2,147,483,648...2,147,483,647] aşıyorsa, sonuç bir tamsayı olacak şekilde kırılır. Örneğin; 2,147,483,647 + 10'un sonucu -2,147,483,639'dır. Eğer sonuç bir kalana sahipse, 2 tamsayı arasındaki bir bölme işlemi bir ondalıklı sonucu verebilir. Örneğin 8/2'nin sonucu 4, tamsayı 9/2.0'ın sonucu 4.5, ondalıklıdır. 2 tamsayı arasındaki bir çarpma işlemi tamsayı aralığını aşıyorsa, sonuç bir ondalıklı sayıya dönüştürülür ve yuvarlama yapılmaz. Örneğin; (100,000 x 100,000) sonuç 1.0e+10, ondalıklı olur[18].

Bit operatörleri bir tamsayı giriş gerektirir. Bit operatörleri için ondalıklı girişler tamsayılara yuvarlanır. Örneğin; 7.9 & 3.4 ----- 7 & 3' eşittirler. Operatör uygulanmadan (örneğin bitwise AND) ondalıklı sayı 7.9 tamsayı 7'ye ve 3.4 ise 3'e

yuvarlanır. Minimum tamsayı değeri için birli negatif işleminin bir sonucu, tamsayı aralığını aşarsa sonuç maksimum tamsayı değerine yuvarlanır:  $-0x80,000,000$  işlemi  $2,147,483,647$  veya  $0x7FFFFFFF$  ile sonuçlanır[18].

Operatörlere ait detaylı açıklamalar EK.A Tablo.1’de ve SimplIQ yorumlayıcı dilinde listelenmiş hazır matematiksel fonksiyonlar EK.B Tablo.1’de verilmiştir.

### III.3.2 Kullanıcı Değişkenleri

Kullanıcı değişkenleri kullanıcı yazılımı ile belirlenebilir. Kullanıcı program değişkenleri başarılı bir şekilde derlenip sürücüye indirilmiş ise komut satırı içinde kullanılabilir. Kullanıcı, bir kullanıcı değişkeni değerini sorgulamak veya değiştirmek için yorumlayıcıyı kullanabilir. Kullanıcı bir değişkenin kapsamına dikkat etmelidir. Bir değişken yerel veya küresel seviyede tanımlanabilir. Yerel (**Local**) değişkenler sadece tanımlandıkları fonksiyon içinde kullanılabilir. Küresel (**Global**) değişkenler ise bir program dışında ve herhangi bir fonksiyon içinde kullanılabilirler. Program çalışır haldeyken veya durdurulduğunda kullanıcı değişkeni sorgulanabilir veya değiştirilebilir. Örneğin derlenmiş bir programın küresel seviyede aşağıdaki satırları içerdiğini varsayalım[18].

#### Örnek:

```
int    ZEBRA,GIRAFFE[3];
float  GNU;
```

$GNU=ZEBRA * GIRAFFE[1] + 2 * \sin(GIRAFFE[2])$  İfadesi geçerlidir. Kullanıcı program değişkenleri bu duruma duyarlıdır. Kullanıcı değişkenleri program içinde *int* veya *float* ile tanımlanır. Değişken uzunluğu maksimum 12 karakterdir ve sözdizimi aşağıdaki gibidir:

```
int    int_var1, int_var2[12], ..., int_varN;
float flt_var1[13], flt_var2, flt_varN;
int    a;
float  b;
```

Bir fonksiyon daimi bir değişkeni referans alabilir. Bu durumda değişken fonksiyon içerisinde **global** olarak deklare edilip üstte fonksiyon tanımında

belirtilmelidir. Küresel olarak tanımlanmayan değişkenler otomatik değişken olarak referans alınır. Daimi bir değişkenin boyutları program içerisinde bir kez belirtildikten sonra kullanılan tüm fonksiyonlar için aynıdır. Daimi bir değişkenin boyutlarını deklare etmenin geçerli yolu ya köşeli parantez içerisinde yazılmasıdır ya da hiç köşeli parantez kullanılmamasıdır[18].

Sadece küresel değişkenler bir boyutlu dizi (array) olabilir. Bir fonksiyonun ne bir yerel değişkeni ne de giriş/çıkış değişkeni (bağımsız değişken) bir vektör olabilir. Yerel değişkenler fonksiyon bloğunun başlangıcında tanımlanmalıdır. Bir değişken ismi program içerisinde kullanılan komutlardan biri olamaz. Program komutları Tablo III.17’de gösterilmiştir.

**Tablo III.17** Program komutları

int	otherwise	wait	break	until	if
float	end	goto	else	return	nargin
elseif	function	nargout	for	global	XOR
While	exit	quit	switch	virtual	case
reset	try	clear	catch	leave	cd
pw	save	continue	this	label	dummy

### III.3.3. Hazır Fonksiyonları Çağırma

Hazır fonksiyonu çağırma işlevi tek ifade için kullanılabilir. EK.B Tablo.1’de matematiksel hazır fonksiyonların listesi verilmiştir. Matematiksel olmayan hazır fonksiyonlar EK.B Tablo.2’de görülmektedir.

#### Örnek:

$\sin(3.14/3)$

$AC=\text{abs}(DC)$

$SP=SP+\sin(3.14/2)$

### III.3.4 Kullanıcı programlama dili özellikleri

*SimplIQ* servo sürücü derleyici tarafından çevrilmiş *Elmo High-level language* (EHL) uzantısı ile yazılmış bir kullanıcı programını okur. *Elmo Studio IDE*’nin bir

parçası olan derleyici (Complier), kompozitör (Composer) içinde bütünleşiktir. Derleme işlemi PC’de çevrimdışı (off-line) olarak çalıştırılır. Kompozitör, SimplIQ bellek yazılımının (firmware) bir parçası değildir. SimplIQ sürücü bir kullanıcı programını yürütmeden önce, kullanıcı programını ilk önce derlemeli ve derleme kodları daha sonra SimplIQ sürücünün hafızasına (Serial flash memory) yüklenmelidir. Derlenmiş kayıtlı program belli bir düzen içinde yerleştirilmiş bir komut listesidir[18].

Bir kullanıcı programını aşağıdaki gibi organize edilir;

- Tam ve ondalık sayı değişkenleri tanımlanır
- İfadeler, komutlar, etiketler ve açıklamaları içeren program yazılır
- Programdan çıkış için bir *exit* direktifi kullanılabilir.

Program tekstinde pek çok yorumlayıcı komutlar kullanılabilir. Program teksinde kullanılmayan yorumlayıcı komutları aşağıda verilmiştir:

2. Sürücü ve sunucusu arasındaki *upload* ve *download* verisi

Örneğin; VR, versiyonu tanımlamak için kullanılmaz (upload işlemi).

3. Flash hafızasına veri depolamak veya flash hafızasında veriyi geri almak.

Örneğin; CD komutu flash hafızasından parametreleri yeniden yüklemek için kullanılmaz.

Örneğin; XC## durdurulmuş bir kullanıcı programını kaldığı yerden devam ettirir.

4. Kullanılan programın yürütülmesine karışılması

Yorumlayıcı komutlarına ilave olarak bir program işleyişi yönlendiren akış ifadelerini içerebilir. Bunlardan bazıları:

-Tekrarlar, Örneğin; for I=1:10:100

-Alt program, yürütme komutları, örneğin; If(I>=100)

-Şartlar; örneğin; while(I<1000)

Program metninde noktalı virgüller, komutlar, satırbaşları veya paragraf başları komutları ayırır. Program x ve k gibi 2 değişkeni tanımlar. ##func bir giriş noktasıdır. Derlemeden sonra, kodun bu parçası, XQ##Func komutunun yorumlayıcıya gönderilmesiyle çalıştırılabilir. Program ##Func de başlağında, kullanıcı değişkeni x’i temizler. Bu otomatik olarak yapılmaz ve başlangıç değeri her ilgili değişken için manuel olarak ayarlanmalıdır. O zaman program, her iterasyonla x artacak şekilde 11 kez iterasyon yapar. Son olarak, x=11 olur. Çıkış komutu programın yürütülmesini durdurur. Diğer kod bölümü XQ##Lab komutunun gönderilmesiyle yürütülebilir[18].

**Örnek:**

```
switch (IP & 3)      İki düşük giriş bitine göre seç.
  case 1             (IP&3) nümerik değeri 1.
PR=1000;
  case 2             (IP&3) nümerik değeri 2.
PR=500;             Değer 2 ise
  otherwise
PR=100;             Değilse. . . (Son iki bir 0 veya 3).
end
BG                 Harekete başla
```

**Not:** Bu örnekte eksen digital girişe bağlı olarak hareket etmektedir.

**III.3.5 Kod Yazım Standardı**

Kullanıcı programı tekst kodlu satırlardan oluşur. Kodları derleyicinin tanınması için belirli bir düzende tanımlı sözdizimi kullanılmalıdır. Bu bölüm program tekstlerinin genel elemanlarını açıklamaktadır. Bir satır şu sonlandırıcılara sahip olabilir; satır başı, satır atlatma veya her ikisi bir arada. Bir satır tek bir ifade veya ifade dizisinden oluşur. Aynı satırdaki bir dizi içindeki ifadeler “;” veya “,” ile ayrılabilir (Parantez veya köşeli parantez içinde olmamalıdır)[18].

**Örnek:**

```
a = 3 , b = 2 , c = a + b ,   Bir satırdaki üç ifade virgülle ayrılmış
a = 3 ; b = 2 ; c = a + b ;   Bir satırdaki üç ifade noktalı virgül ile ayrılmış
a = 3 , b = 2 ; c = a + b     Virgül, noktalı virgül ve satır atlanarak ayrılmış
[a,b] = func (23, c, 3.14)    Virgüller parantez içinde olduğundan ayrılmamıştır
```

Kullanıcı programı bir satırdan uzun olabilir ve ekranda çok iyi bir şekilde okunamayabilir. Okunabilirliği arttırmak için üç nokta (...) ile devam ettirilebilir.

**Örnek:**

```
c = 12 * a + sqrt(2) - sin(3.14 / 2) + 7 ^ 3 * ...
(6 + b) * 34
```

Kullanıcı programının tekst bölümlerinde her satırında maksimum 128 karakterden oluşabilir. Bu değer aşılsa derleyici hata verir. İfadelerde de çeşitli sınırlamalar vardır. İzin verilebilen maksimum bir ifade uzunluğu 512 sembolden oluşur. Program birden fazla satıra uzanan karmaşık bir ifade içeriyorsa, açıklamalar ve üç noktalar hariç, 512 karakteri geçerse derleyici hata verir[18].

Kullanıcı programında limitlendirilmiş ayar parametrelerinin listesi:

5. Kullanıcı programın maksimum uzunluğu
6. Maksimum Döngü sayısı, fonksiyon içeren maksimum sayı, maksimum etiket sayısı, maksimum otomatik döngü sayısı
7. Maksimum global ve local değişken sayısı
8. Global değişkenleri depolamak için veri kesimi maksimum uzunluğu
9. Derlenmiş kodlar için kod kesim maksimum uzunluğu
10. Maksimum yığın derinliği

### III.3.6 Sistem komutları

SimplIQ sistem komutları iki harfle tanımlanmıştır. (ac = 100,000 ve AC = 100,000). Her bir komut 16 bit işaret içerir. Örneğin dördüncü bit komutunu bir değere ayarlamak için kullanılıp kullanılmadığı tanımlar[18].

#### Örnek:

- |        |  |
|--------|--|
| a = AC | Bu ifade sistem komutu AC'nin değerini a değişkenine atamaktadır. AC komutu "get a value" yani değerlendirmeye açı ise geçerlidir. |
| AC = a | Bu ifade a değişkenin değerini AC sistem komutuna atar. AC komutunun "set a value" yani ayarlanmaya açık olması gerekir.           |
| BG     | Bu çalıştırılabilir bir komuttur. Atanamaz. BG=1 ve a=BG geçerli ifadelerdir.  |
| LS     | Geçersiz bir komuttur. Kullanıcı programı içerisinde izin verilmesede, yorumlayıcı dilde kullanılır.                               |

Tüm sistem komutlarına ait açıklamalı tablo EK.C'de verilmiştir.

### III.3.7 Program akış Komutları

*SimplIQ* sürücü kullanıcı program akışını yönetmek için bir komutlar dizisi kullanır. Bu komutlarla; kullanıcı programı belirli olayları otomatik olarak cevaplamak veya tekrarlama kararlarını yapabilir. Program akış komutları, sıralı çalışan bir komut dizisinden daha karışık fonksiyonları icra etmek için kullanıcı programlarını aktif hale getirir. Program akış komutları Tablo III.18’de sıralanmıştır.

**Tablo III.18** Program akış komutları

Komut	İşlev
<b>While-end</b>	Şart sağlanana kadar tekrar eder
<b>until</b>	Şart sağlanana kadar programı askıya alır
<b>wait</b>	Belirtilen süre kadar programı bekletir
<b>For-end</b>	Belirli kez programı tekrar eder
<b>break</b>	Yinelemeyi durdurur veya değiştirir
<b>İf-elseif-else-end</b>	Şart ifade eder
<b>Switch-case-otherwise-end</b>	Durum seçimi
<b>goto</b>	Programın belirli bir noktasına gider
<b>reset</b>	İşletilen programın mevcut halini iptal eder
<b>Function-return</b>	İşlev ve dönüş noktası belirtir
<b>##</b>	Etiket veya otomatik rutin belirtir
<b>#@</b>	Etiket veya otomatik rutin belirtir
<b>exit</b>	Programın işletilmesini durdurur
<b>continue</b>	Bir sonraki tekrar döngüsüne geçiş yapar
<b>Try-catch</b>	Beklenmeyen hatalara karşı tepki göstermek için kullanılır.

### III.3.8 Fonksiyonlar

Fonksiyonlar parametreler tarafından belirlenen ve programın herhangi konumunda çağrılabilen program bölümleridir. Bir fonksiyon deklarasyonu aşağıdaki bölümlerden oluşur:

- Ayrılmış fonksiyon anahtar sözcüğü
- Çıkış argümanlarının köşeli parantez içerisinde yazılmış listesi (İsteğe bağlı olarak)
- Atama işareti, sadece çıkış argümanı mevcut ise
- Fonksiyon ismi
- Giriş argümanlarının köşeli parantez içerisine yazılmış listesi (Boş bir liste olarak yazılabilir)
- Fonksiyon kapsamında önceden ayrılmış bir yanıt anahtar sözcüğü

#### Örnek 1:

```
function [int y1, int y2] = func1 (float x1, int x2)
```

**func1** isimli fonksiyonun iki giriş değeri (x1 ve x2) vardır. Yanıt olarak yine iki değer alınır. Çıkış değerlerinin listesi boş olabilir. Bu durumda iki ya da üç öge mevcut değildir. Sadece çıkış değerleri mevcutsa köşeli parantez kullanılmayabilir. Çıkış ve giriş değerleri sayısı en fazla 16 olabilir[18].

#### Örnek 2:

```
function func2 (float x1)
```

Burada **func2** isimli fonksiyonun tek giriş değeri vardır ve çıkış yanıtı yoktur.

#### Örnek 3:

```
[int y1] = func3 (int x1)
```

Bu fonksiyon deklarasyonu fonksiyon anahtar sözcüğü eksik olduğundan geçersizdir.

### III.3.9 Otomatik alt rutinler

Bir otomatik alt rutin (auto-routine) sistem akışı tarafından otomatik olarak işletilen özel tip bir rutindir. Bu rutinler ancak istenen şartlar gerçekleştiğinde işletilir ve giriş veya çıkış değişkenleri yoktur[18].

#### Sözdizimi:

Bir otomatik rutin fonksiyon ya da altyordam olarak tanımlanabilir:

- Fonksiyon olarak tanımladığında fonksiyonlar için geçerli tüm söz dizim kuralları geçerlidir.
- Altyordam (subroutine) olarak tanımlandığıdaysa tanım satırında ## veya #@ sonra yazılan altyordam isimleri gelir. Altyordam yapısı tanım satırını takip eder ve **return** anahtar sözcüğü ile tamamlanır. (**return flow** kontrol bloğu içinde yer almadığı sürece)

Otomatik altyordamlar için herhangi varsayılan denetleyici bulunmaz. Kullanıcı tanımlanmamış ise altyordam işlenirken herhangi bir denetim gerçekleşmez. Tablo III.19'da tüm altyordamlar özetlenmiş ve en yüksek öncelikten en düşüğe göre sıralanmıştır.

**Tablo III.19** Otomatik alt rutinler

Rutin adı	Öncelik	Çalışma koşulu	Mask (MI)
<b>AUTOEXEC</b>	0	Enerji verildiğinde çalışır. İstenen zamanda ayrı yeten çağrılabilir.	1 (0x1)
<b>AUTO_PERR</b>	1	Bir yürütme hatası oluştuğunda çalışır.	32768 (0x8000)
<b>AUTO_ER</b>	2	Motor hatasında MO=0 iken çalışır.	2 (0x2)
<b>AUTO_STOP</b>	3	Dijital giriş sert durdurma moduna ayarlı ve aktif ise çalışır	4 (0x4)
<b>AUTO_BG</b>	4	Dijital giriş başlatma moduna ayarlı ve aktif ise çalışır	8 (0x8)
<b>AUTO_RLS</b>	5	Dijital giriş geri yön moduna ayarlı ve aktif ise çalışır	16 (0x10)
<b>AUTO_FLS</b>	6	Dijital giriş ileri moduna ayarlı ve aktif ise çalışır	32 (0x20)
<b>AUTO_ENA</b>	7	Dijital giriş yetki atama moduna ayarlı ve aktif ise çalışır	64 (0x40)
<b>AUTO_I1</b>	8	Dijital giriş #1 genel kullanım için ayarlanmış ve aktif ise çalışır.	128 (0x80)
<b>AUTO_I2</b>	9	Dijital giriş #2 genel kullanım için ayarlanmış ve aktif ise çalışır.	256 (0x100)

**Tablo III.19 (Devam)** Otomatik alt rutinler

<b>AUTO_I3</b>	10	Dijital giriş #3 genel kullanım için ayarlanmış ve aktif ise çalışır.	512 (0x200)
<b>AUTO_I4</b>	11	Dijital giriş #4 genel kullanım için ayarlanmış ve aktif ise çalışır.	1024 (0x400)
<b>AUTO_I5</b>	12	Dijital giriş #5 genel kullanım için ayarlanmış ve aktif ise çalışır.	2048 (0x800)
<b>AUTO_I6</b>	13	Dijital giriş #6 genel kullanım için ayarlanmış ve aktif ise çalışır.	4096 (0x1000)
<b>AUTO_HM</b>	14	Homing yani başlangıç noktasına dönüş işlevi bitişinde çalışır.	8192 (0x2000)
<b>AUTO_HY</b>	15	Yardımcı homing işlevi bitişinde çalışır.	16384 (0x4000)

**Not:** AUTOEXEC hariç tüm altyordamlar program çalışırken otomatik olarak aktive edilirler.

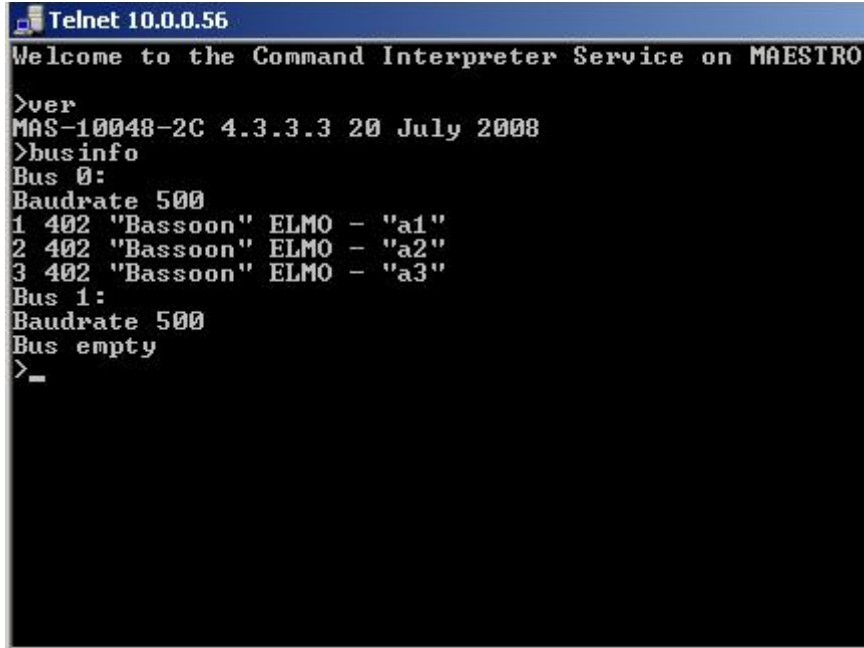
### III.3.10 Haberleşme bağlantı ayarları

Programlamaya geçmeden önce her bir eksenin saha veri yolu üzerinde kendine has bir kimlik numarasına ait olduğundan emin olmak gerekir. Düğümler üzerindeki cihazların CAN kimlik numaraları PP[N] sistem komutu kullanılarak kontrol edilebilir veya değiştirilebilir[15].



**Şekil III.32** CAN bilgilerinin düzenlenmesi

CAN üzerindeki her bir cihaza kimlik ataması yapıldıktan sonra bu bilgiler *CANopen configurator* ile kontrol edilebilir veya değiştirilebilir. *CANopen configurator* ekran görüntüsü şekil III.32’de görülmektedir. Bizim sistemimizde kontrolörün iki CAN bağlantısından birinin üzerinde üç adet dijital sürücü yer almaktadır. CAN iletişim hızı 500kbit ayarlanmıştır. CAN bağlantıları ayrıca Elmo Studio üzerinde yer alan Telnet bağlantısı üzerinden de düzenlenebilir veya kontrol edilebilir. Kontrolöre ait bir çok sistem komutu Telnet üzerinden kullanılabilir.



```
Telnet 10.0.0.56
Welcome to the Command Interpreter Service on MAESTRO

>ver
MAS-10048-2C 4.3.3.3 20 July 2008
>businfo
Bus 0:
Baudrate 500
1 402 "Bassoon" ELMO - "a1"
2 402 "Bassoon" ELMO - "a2"
3 402 "Bassoon" ELMO - "a3"
Bus 1:
Baudrate 500
Bus empty
>_
```

Şekil III.33 Telnet bağlantısı

Kontrolör Ethernet üzerinden sunucu bilgisayara bağlı ve Telnet online modun da iken CAN üzerindeki her bir düğüm elemanı görülebilmektedir. Telnet bağlantısı üzerinden CAN bilgilerinin görünümü Şekil III.33’de görülmektedir. Bu noktadan sonra Elmo Studio aracılığı ile program yazımı ve eş zamanlı olarak sürücü veya kontrolöre yükleme işlemleri gerçekleştirilebilir.

### III.3.11 Eksen özellikleri

Eksenlerin Elmo Studio üzerinde kaynaklar bölümden ayarlanması daha önce açıklamıştır. Programlama sırasında bu eksenlere ayrı ayrı komut göndermek istendiğinde ilgili komutun başına eksen adının eklenmiş olması yeterlidir.

**Sözdizimi:**

Eksen adı.Komut=Değer      Değer atamak için  
Değer=Eksen adı.Özellik      Değer almak için

**Örnek:**

A1.UM=2  
A1.MO=1  
A1.BG

Eksenler vektörlerin bir parçası olabildiği gibi tek başlarına gerçekleştirebilecekleri işlevlerde mevcuttur. Fakat genelde çok eksenli hareket için vektör alt bileşeni olarak kullanılırlar. Eksen ayarlarında kullanılan tüm komutlar EK.D'de Tablo I'de verilmiştir.

**III.3.12 Vektör özellikleri**

Tasarlanan hareket iki veya üç boyutlu olabilir. Her iki şekilde de eksenleri belirli bir düzen içinde hareket etmesi gerekir. Bu yüzden eksenler bir vektör ile tanımlanır ve gerekli işlemler vektör üzerinden yapılır. Vektörlerde de eksenlerde olduğu gibi bir sözdizimi kullanılır.

**Sözdizimi:**

Vektör adı.Komut=Değer

**Örnek**

V1.AC=10000000;  
V1.MO=1;  
V1.BG;

Bir vektör iki ve üç boyutlu geometrik şekilleri, trigonometrik fonksiyonları veya belirli bir tabloya bağlı kalacak şekilde ardışık hareketleri gerçekleştirebilir. Üç eksenli hareket için tanımlanan üç boyutlu bir vektör üç farklı eksen oluşturur. Bu eksenlerin aynı CAN üzerinde tanımlanmış olması gerekir. Üç boyutlu ve iki boyutlu vektörler interpolasyon ile oluşturulduğu için temel özellikleri hemen hemen aynıdır. Diğer yandan birden fazla eksen üzerinden gerçekleştirilen hareketlerde tüm eksenlerin

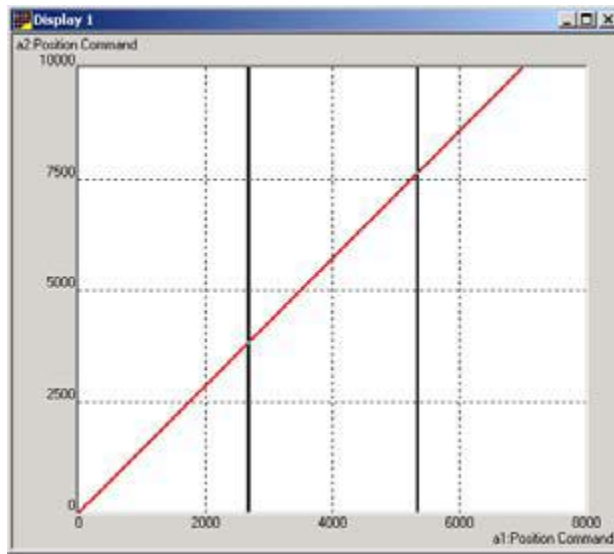
aynı hızda hareket etmesi veya aynı anda bitiş noktasında olması gibi bazı özel sınırlamalar mevcuttur. Vektör özelliklerine ait tüm komutlar EK.D Tablo II'de verilmiştir[16].

### III.3.13 Programlama örnekleri

Herhangi programlama öncesi eksen ve vektör bilgileri daha önceki bölümlerde gösterildiği gibi donanıma göre ayarlanmış olmalıdır. Burada önemli olan program içerisinde kullanılan eksen, vektör ve CAN giriş-çıkışlarına ait bilgilerin eş olmasıdır. Aksi takdirde Elmo Studio yazılan programın derlemesinde hata verecek ve kontrolöre yüklemeyecektir.

#### Örnek1:

```
function run()
sync(0,20)           //CAN senkronizasyon süresi 20 ms olarak ayarlanmış
v1.mo=0             //Motor kapalı konumda
v1.px=0             //Bulunan konumu sıfır referans olarak ayarla
v1.mo=1             //Motor açık konumda
v1.vsp=27300        //En yüksek hız
v1.line(5000,10000) //x=5000, y=10000 koordinatlarına git
v1.bg               //Harekete başla
end function
```



Şekil III.34 İki boyutlu doğrusal hareket çıkış grafiği

Örnek 1’de iki eksenli olan v1 vektörü belirtilen koordinatlara göre doğrusal bir hareket yapmaktadır. Burada v1 vektörünün kaynaklar bölümünde iki eksenli bir vektör olarak tanımlanmış olması gerekir. Hareketin grafiksel görünümü şekil III.34’de gösterilmiştir.

### Örnek2:

```
function run()
a1.rm = 0
a2.rm = 0 //Eksenler için referans değeri kullanıcı
           tarafından sıfır olarak belirlenmiş
sync(0, 10) //CAN senkronizasyonu 10 ms olarak
            ayarlanmış
v1.mo=0 //motor kapalı
v1.px=0 //Başlangıç konumu olarak ayarla
v1.mo=1 //motor açık
v1.vum=1 //En yüksek hızda yörünge hareketi tanımla
v1.vsc=1 //En yüksek hızda dönüm noktalarını yumuşat
v1.vtm=1 //PVT hareket tablosunu etkin hale getir
v1.vac = 28000000 //Hızlanma ivmesi
v1.vdc = 28000000 //Yavaşlama ivmesi
v1.vsp=10000 // En yüksek hız
v1.circle(5000, 180, 90) // 5000 birim yarıçapında 180° dereceden
                        başlayan 90°’lik çember hareketi yap
v1.bg // Harekete başla
while (a1.ms==2)||(a2.ms==2) //İki eksenle durana kadar bekle
wait(10) //10ms bekle
end while
end function
```

Örnek 2’de referans değerleri kullanıcı tarafından belirlenmiş. Bu değer kullanıcı tarafından belirtilmediğinde sistem tarafından otomatik olarak belirlenir. İki eksenli v1 vektörüne çember yayı çizdirmek için kullanılan programda vektör hareketinin nasıl olacağı belirleyen komutlar kullanılmıştır. Bu komutlarla ilgili ayrıntılı bilgi EK.D Tablo.2’de verilmiştir.

### Örnek3:

```
function run()
float teta,pi // teta ve pi değerlerinin tanımlanması
sync(0,10)
v1.vum=1 //En yüksek hızda yörünge hareketi
```

```

v1.vac = 28000000 //Hızlanma ivmesi
v1.vdc = 28000000 //Yavaşlama ivmesi
v1.vsp = 50000 //En yüksek hız
v1.vse = 0 //Son hız
pi = 3.14159265358979 //pi sayısı tanımlanmıştır
a = 100000 //Elips a eksen
b = 50000 //Elips b eksen
Xc = 0 // Elipsin x üzerindeki merkez koordinatı
Yc = 0 // Elipsin y üzerindeki merkez koordinatı
v1.splines() // Yörüngesel harekete başla
for teta = 0:pi/18:2*pi //teta değeri için gerekli hesaplama
x = Xc + a*cos(teta) //x koordinatı için gerekli hesaplama
y = Yc + b*sin(teta) //y koordinatı için gerekli hesaplama
v1.splinep(x,y) // Hesaplanan x ve y koordinatları için yörünge
Hareketi tanımla
end for
v1.splinee(0) // Yörüngesel hareket sonu
v1.bg // Hareketi gerçekleştir
end function

```

Yukarıdaki örnek 3’de görüldüğü gibi gerçekleştirilecek hareket için yapılması gereken matematiksel işlemler program içerisinde gerçekleştirilebilir. Burada dikkat edilmesi gereken tüm değişkenlerin türleri ile birlikte fonksiyon içinde tanımlanmasıdır. Tanımlanmayan değişkenler veya farklı türlerde tanımlanan değişkenler hataya neden olacaktır.

#### Örnek4:

```

function run()
sync(0,10)

SetAxisStartPos(a1, 0) //Birinci eksen 0’a ayarla
SetAxisStartPos(a2, 0) //İkinci eksen 0’a ayarla

```

```

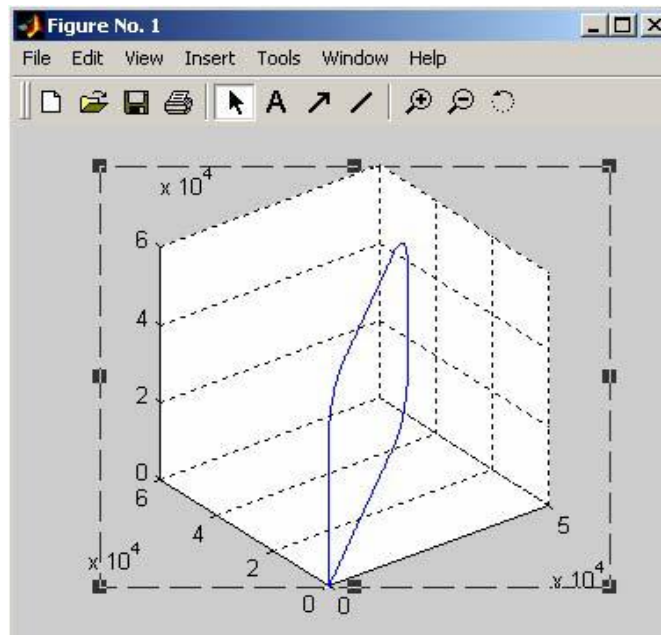
SetAxisStartPos(a3, 0)           //Üçüncü eksen 0'a ayarla

v2.vsc=2
v2.vsr=12000
v2.vsp = 70000                   //En yüksek hız
v2.vse = 70000
v2.starts()
v2.addline(50000, 50000, 0)     //V2 vektörü ile verilen koordinatlarda
                                //doğrusal hareket gerçekleştir

v2.addline(50000, 50000, 50000)
v2.addline(0, 0, 50000)
v2.vse = 0                       //Hareket sonundaki hız
v2.addline(0, 0, 0)             //Başlangıç konumuna dön
v2.ends()
end function

```

Örnek 4'de üç eksenli v2 vektörü  $x=0$ ,  $y=0$ ,  $z=0$  konumundan başlayarak  $50000:50000:0$ ,  $50000:50000:50000$  ve  $0:0:50000$  noktalarından geçen doğrusal hareketlerle bir dikdörtgen oluşturmakta ve sonunda başlangıç konumuna geri dönmektedir. Hareketin çıkış grafiği Şekil III.35'da gösterilmiştir.



Şekil III.35 Üç boyutlu doğrusal hareket çıkış eğrisi

### Örnek5:

```
function run()
int x,y,z //x,y ve z değerleri tam sayı olarak tanımlanıyor
x=100000; y=50000; z=3000 //Koordinat bilgileri
sync(0,10)
v1.vtm=1
v1.vdc=3000000
v1.vac=3000000
v1.mo=0
v1.px=0
v1.mo=1
v1.vsp=273000 //En yüksek yörünge hızı
v1.line(x,y,z) //Belirtilen koordinatlarda doğrusal hareket yap
v1.bg //Harekete başla
while(a1.ms==2)||(a2.ms==2)||(a3.ms==2)//Tüm eksenler durana kadar bekle
wait(1000) //Tüm eksenler durunca 1sn daha bekle
end while
end function
```

## III.4 SİSTEM MONTAJI

Gövde profili kullanılarak oluşturulan kaide hareket modülünün istenilen sınırlar içerisinde istikrarlı bir biçimde hareket edebilmesi için tasarlandı. Doğrusal hareket modülü bu profil üzerine özel civata ve somunlarla monte edilerek, dengeli bir sistem için doğruluk ölçümleri yapıldı. Ayrıca daha sonraki hesaplamalarda kullanılmak üzere sistemin hareketli parçalarının ağırlıkları tartıldı. Tüm bunlar mekanik sistemde oluşabilecek herhangi uyumsuzluğu engellemek için önemli değişkenlerdir.

Şekil III.36'da ki resimde görüldüğü gibi doğrusal hareket modülü üzerindeki mekik bir kayış üzerinde hareket etmektedir. İkinci eksen kontrol eden motor bu mekik üzerindeki levhaya dik açıyla monte edilmiştir. Üçüncü eksen ise ikinci eksen kontrol eden motorun miline akuple edilmiş olan bir diğer parça üzerinde hareket etmektedir. Doğrusal hareket modülünü tahrik eden motor dışındaki iki motor hareket eden mekiğin üzerinde olduğu için bu motorlara ait olan geri besleme ve enerji

kablolarının hareketi engellemeyecek biçimde yerleştirilmesi gerekti. Bunun için mekanik kablo taşıyıcısı kullanıldı. Kullanılan kablo taşıyıcısı Şekil III.37'de görülmektedir.

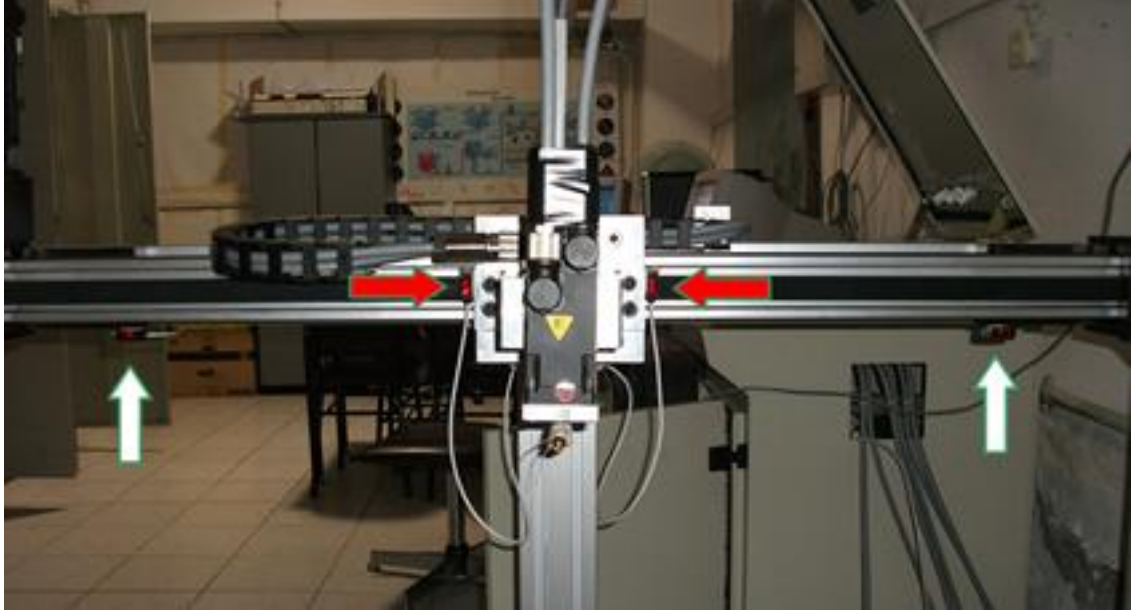


Şekil III.36 Mekanik sistem



Şekil III.37 Kapalı tip mekanik kablo taşıyıcısı

Motorların kablo bağlantıları gerçekleştirildikten sonra mekanik sistemin hareketi kontrol edildi. Motor bağlantılarının hareketi nasıl etkilediği incelendi ve bazı konumlarda kablo ağırlıklarının da hesaba katılması gerektiği görüldü. Bunun için motorun üst limit akımı değiştirilerek gerekli moment sağlandı.



**Şekil III.38** Optik sensör yerleşimleri

Sistemin güvenliği için motorların eksen hareketlerinin sınırlanması gerekmektedir. Bu yüzden şekil III.38’de görüldüğü gibi optik sensörler ile hareket sınırları belirlenmiş ve bu sensörlerin çıkışları direkt olarak sürücülerdeki dijital girişlere bağlandı. Bu sayede eksen hareketini doğrudan durdurabilecek ve diğer eksenleri etkilemeyecektir.

Mekanik sistem üzerindeki motor ve sensörlerin bağlantı kabloları kontrol panosuna bir kablo kanalı içerisinde taşındı. Sürücü ve kontrolörün kontrol panosu içerisinde monte edildikleri yere kabloların taşınması ve bağlantıların yapılmasından sonra kontrol panosu için gerekli olan besleme bağlantıları yapıldı. Bu bağlantılar yapılırken her bir sürücüye giden faz ayrı ayrı sigortalanarak sistem güvenliği artırıldı.

Ayrıca sürücü ve kontrolörün çalışması için gerekli olan 24VDC kaynaktan şebekeden yalıtılmış ve ayrı bir sigorta üzerinden beslenmiştir. Kontrol panosunun görünümü Şekil III.39’da gösterilmiştir.



Şekil III.39. Pano görünümü

## BÖLÜM IV

### SONUÇLAR VE TARTIŞMALAR

Bu bölümde çalışma süresince elde edilen sonuçlar ve bu sonuçlara bağlı olarak yapılan değişiklikler konu edilmiştir. En iyi sonucu elde etmek ve problemin çözümüne yönelik tartışmalara yer verilmiştir. Dijital servo sürücülerin farklı çalışma modları ve modların motor üzerindeki etkileri tartışılarak, farklı değişkenlerin sistem çalışma karakteristiği üzerindeki etkileri üzerinde durulmuştur. İleriki uygulamalar için karşılaşılan sorunlar ve çözüm yaklaşımları ele alınmıştır.

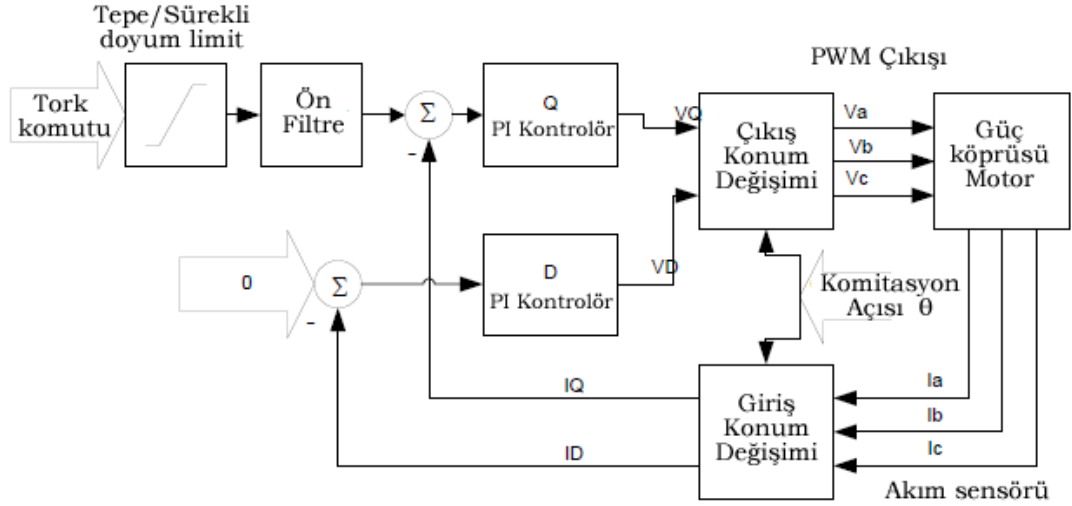
#### IV. 1 SÜRÜCÜ MODLARI

Fırçasız servo motor sürücüleri akım, hız ve pozisyon gibi farklı değişkenleri kontrol ederek en iyi sonuç için sürekli ölçümler yapan karmaşık elektronik cihazlardır. Gerçekleştirilen uygulamada ELMO dijital sürücülerde farklı modlarda motor kontrolü yapabilmektedir. Tüm bu modların çalışmasında akım kontrolünün önemi ortaya çıkmaktadır. Hız, moment ve konum gibi değişkenler akım kontrolörü üzerinden gelen ve geri besleme cihazından alınan veriler ile kontrol edilmektedir[17].

**Tablo IV.1** Birim modları (Unit mode)

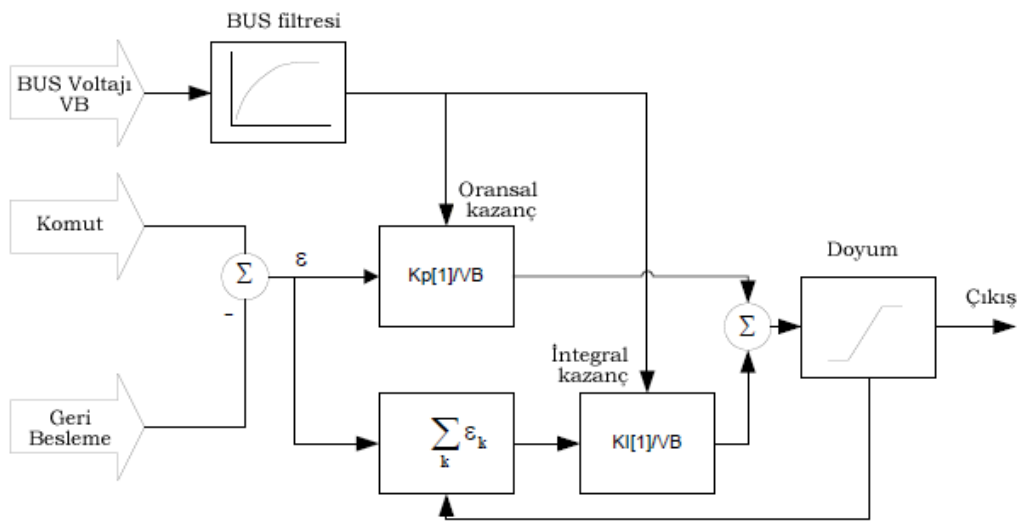
UM ye atanacak değer	Açıklama
1	Tork (akım) kontrol modu
2	Hız kontrol modu
3	Pozisyon (Micro-stepper) modu
4	Çift geri beslemeli konum kontrolü
5	Tek geri beslemeli konum kontrolü

ELMO dijital servo sürücülerde akım kontrolü aktif (IQ) ve reaktif (ID) akım olmak üzere iki değişken üzerinden kontrol edilmektedir. Akım kontrolörünün sistem mimarisi Şekil IV.1’de gösterilmiştir[17].



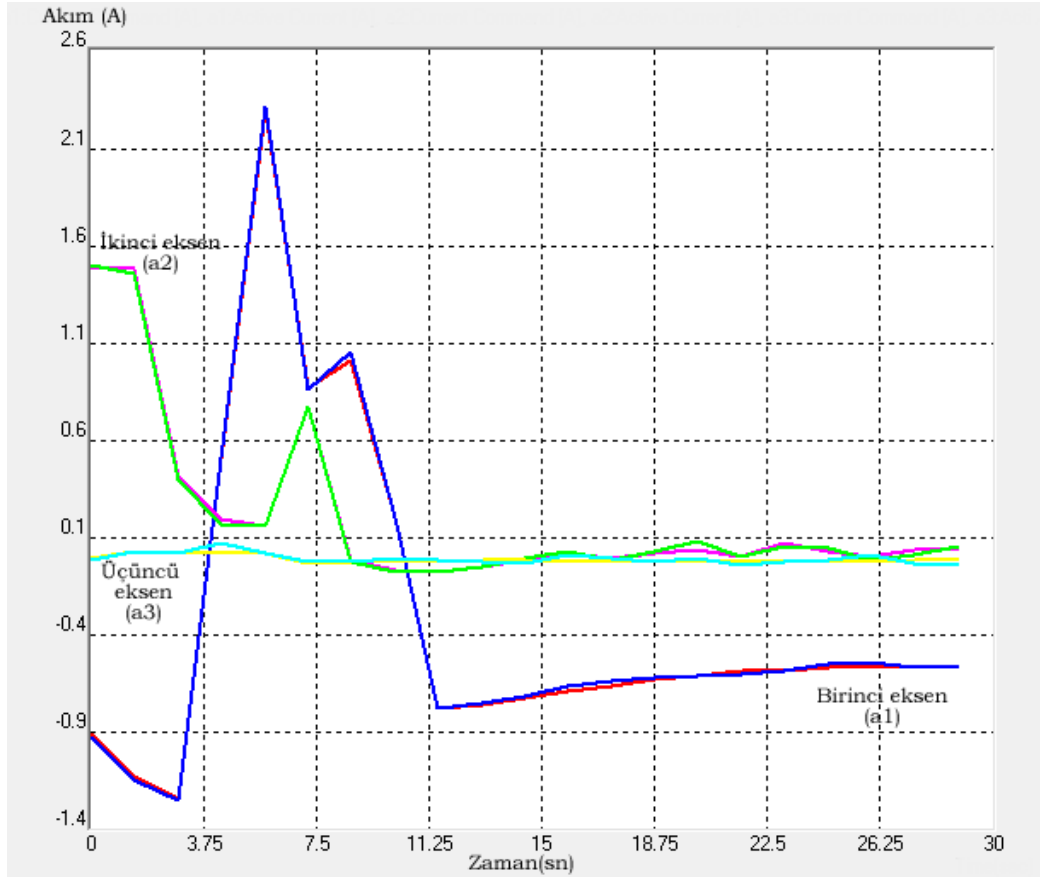
Şekil IV.1 Akım kontrolörü sistem şeması

Aktif ve reaktif akım kontrolörlerin yapısı birbirine benzerdir. Bu iki kontrolör için en önemli iki değişken KP oransal ve KI integral kazanç değerleridir. Bu iki değer akım kontrolörün çıkış PWM iş çevrim oranına uyması için BUS voltajına bölünür. Doyum ise akım kontrolörün örnekleme zamanına bağlı olarak değişmektedir. PI akım kontrolörün şeması Şekil IV.2’de verilmiştir[17].



Şekil IV.2 PI akım kontrolör şeması

Gerçekleştirilen uygulamada bu değerler  $KP=12.280$  ve  $KI=15188$  civarında atanmıştır. Bu değerler sistemin yükü ve sistem yanıtına göre belirlenmiştir. İstenildiği takdirde farklı modlar için değiştirilebilir. Şekil IV.3'de görülen akım komutu ve sistem yanıtı grafiğinde bazı uyumsuzluklar görülmektedir. Bu uyumsuzluklar dengesiz yük ve mekanik uyumsuzluklardan kaynaklanmaktadır.



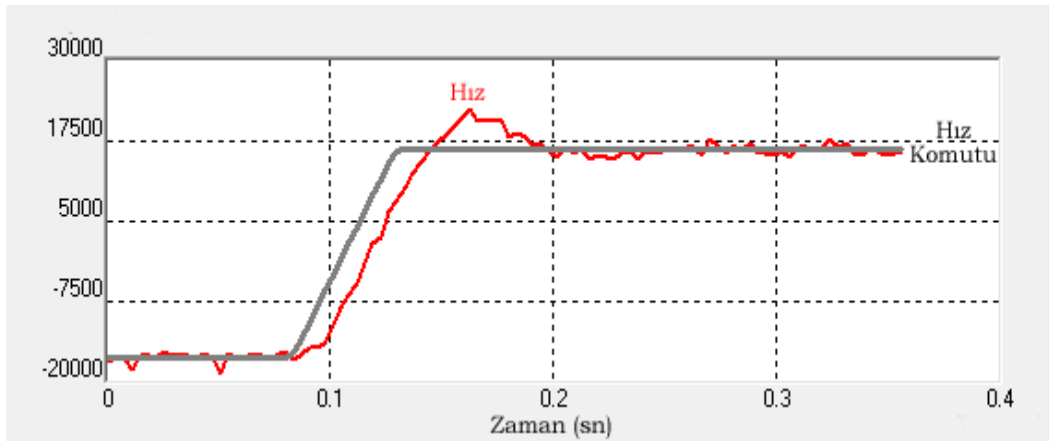
Şekil IV.3 Akım komutu ve sistem cevabını gösteren grafik

**Not:** Şekil IV.3'de lacivert 1. eksen akım komutu, kırmızı sistem cevabı; Yeşil 2. eksen akım komutu, turuncu sistem cevabı; Mavi 3. eksen akım komutu, sarı sistem cevabını göstermektedir

ELMO dijital sürücüler akım (Tork), hız, pozisyon (micro-stepper), çift ve tek geri beslemeli pozisyon kontrol modu olmak üzere beş farklı birim moduna sahiptir. Akım modunda motorun sadece momentini kontrol edilir. Hız modunda ise geri besleme elemanında gelen bilgi ile motorun hızı, hareket yumuşaklığı kontrol edilir. Bu sayede sistemin yanıt süresi ve oturma şekli kontrol edilebilir. Pozisyon modunda hız moduna

benzer şekilde çalışır fakat pozisyon geri besleme bilgisi alınmaz. Birden fazla geri besleme elemanı kullanılıyorsa çift, tüm ölçümler için tek bir geri besleme elemanı kullanılıyorsa tek geri beslemeli mod kullanılır.

Gerçekleştirilen uygulamada hız kontrolörü ile yapılan ayarlamalarda sistem yanıtı Şekil IV.4’de verilen grafikte görüldüğü gibidir. Burada yumuşatma faktörü 4 ve hız 60 devir/dk olarak ayarlandı. Hızlanma ve yavaşlama ivmeleri ise 700000 darbe/sn<sup>2</sup> olarak belirlendi. İstenildiği takdirde bu değerler değiştirilerek sistem yanıt süresi ve oturma şekli değiştirilebilir.

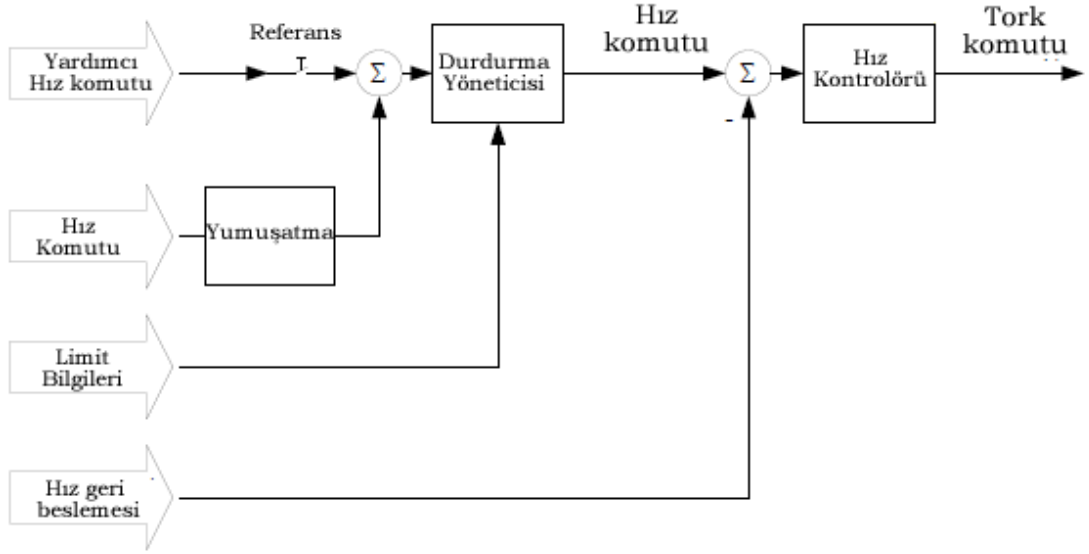


**Şekil IV.4** Hız modu sistem cevabını gösteren grafik

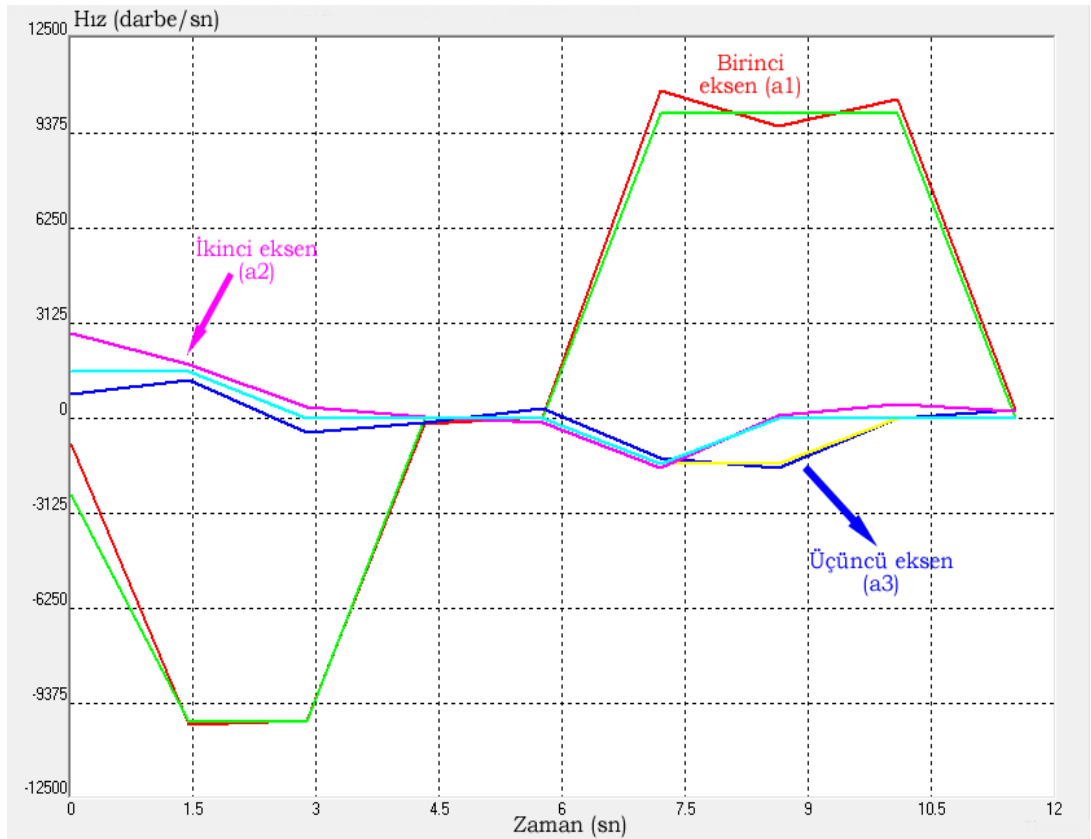
**Not:** Şekil IV.4’de gri renk hız komutu, kırmızı ise sistem cevabını göstermektedir.

Hız modunda eğer referans sıfır olarak alınmışsa yardımcı hız komutu devrede değildir. Yumuşatma faktörü sistem cevabının yükseliş grafiğinde tepe noktasının değerini belirler. Durdurma yöneticisi limit bilgileri ve limit sensörlerinden aldığı bilgileri işleyerek hız kontrolörüne gerekli bilgiyi sağlamaktadır. Hız modu sistem şeması şekil IV.5’de görülmektedir[17].

Şekil IV.6’de görüldüğü gibi sistem yanıtı ile hız komutu arasında ufakta olsa bir fark görülmektedir. Bu fark mekanik etkenlerden olduğu gibi haberleşme isteminde oluşan gecikmelerden de kaynaklanmaktadır. Bu farkın en aza indirgenmesi için sürtünme, yük dağılımı ve haberleşme bağlantısı hızı gibi değişkenler kontrol edilmiştir. Grafiğin yükselen ve düşen kenarlarını ivme değişkeni, hızın sabit kaldığı yatay bölümleri ise yumuşatma faktörünü değiştirerek değiştirmek mümkündür.



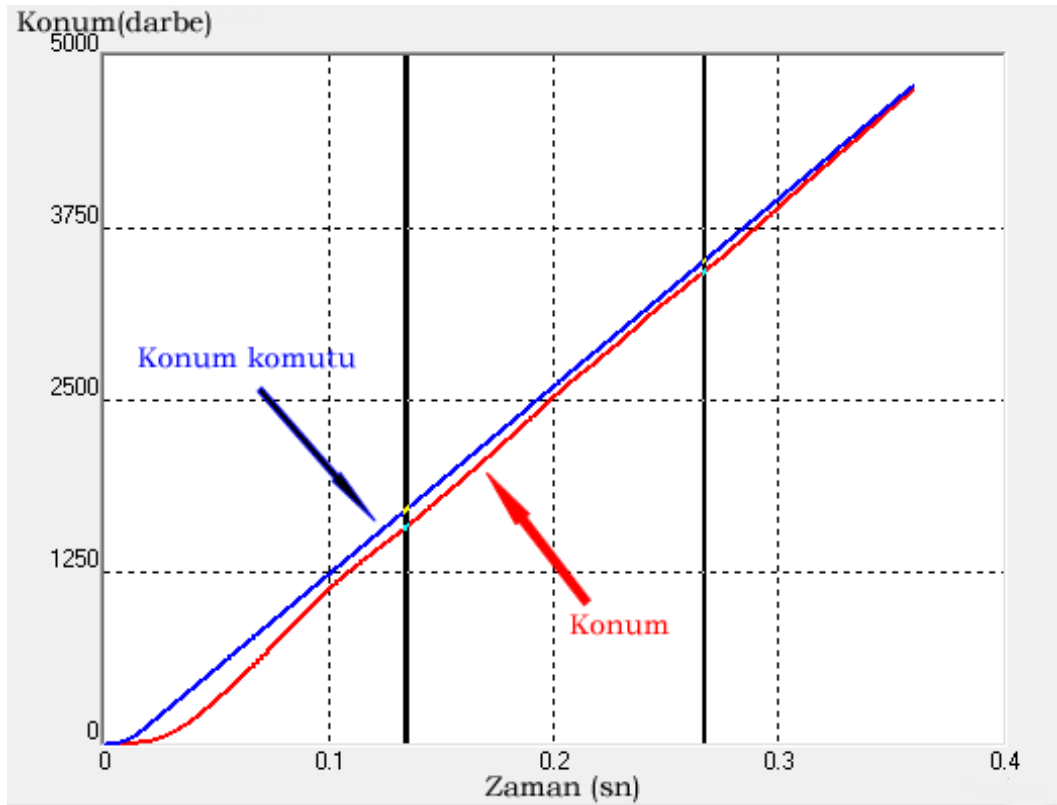
Şekil IV.5 Hız modu sistem şeması



Şekil IV.6 Üç eksen için hız komutu ve sistem cevabını gösteren grafik

**Not:** Şekil IV.6'da yeşil 1. eksen hız komutu ve kırmızı sistem cevabı; Pembe 2. eksen hız komutu ve mavi sistem cevabı; Lacivert 3. eksen hız komutu ve sarı sistem cevabını göstermektedir.

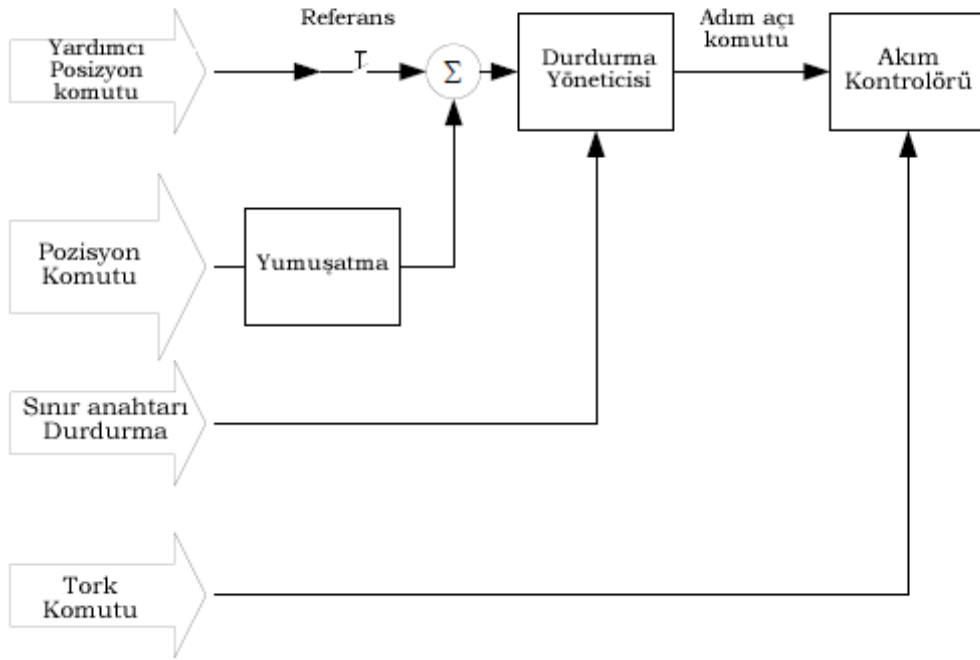
Pozisyon modunda sistemin pozisyon komutuna verdiği yanıt incelenmiştir. Burada da hız modunda olduğu gibi yumuşatma faktörü, hız ve ivme gibi değişkenler önem kazanmaktadır. Şekil IV.7’de görüldüğü gibi sistem cevabı ile pozisyon komutu arasında başlangıçta ufak bir fark bulunmaktadır. Bu pozisyon hatası kısa bir süre sonra giderilmiş ve sistem istenen konuma oturmuştur. Oturma zamanı yaklaşık olarak 350 ms olarak görülmektedir. Bu süreyi düşürmek için sistemin mekanik hassasiyeti ve veri iletişim hızı artırılmalıdır.



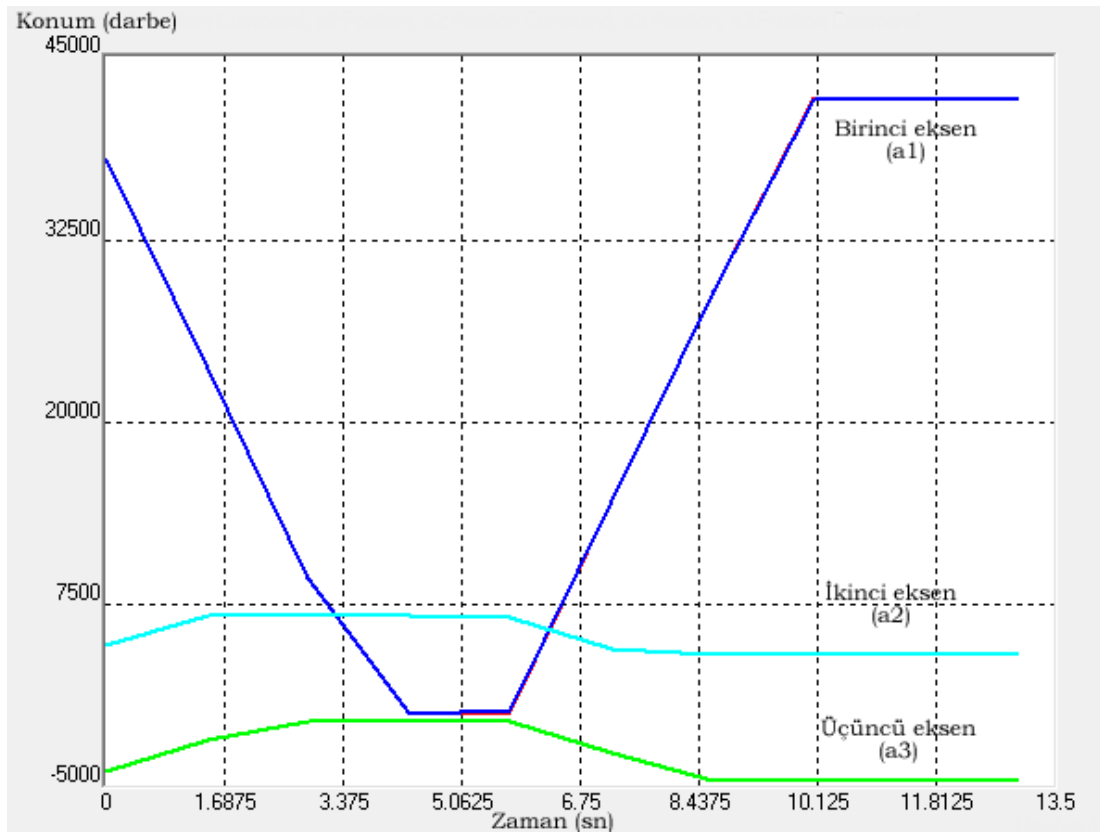
Şekil IV.7 Pozisyon modu sistem cevabı gösteren grafik

Bu moda motor geri besleme bilgisi almaz. Motor manyetik alanı istenilen konuma hareket ettirilir ve rotorda bunu takip eder. Geri besleme bilgisi olmadığı için bu durumda SimpiQ sürücü mikro adımlı bir sürücü gibi davranır ve her elektriksel tur için 1024 darbe üretir. Adım modu sistem şeması Şekil IV.8’de görülmektedir[17].

Üç eksen için pozisyon komutuna sistemin verdiği cevap Şekil IV.9’de gösterilmiştir. Pozisyon komutu ve sistem yanıtı grafikte hemen hemen üst üste oturmuş yani pozisyon hatası gözlenmemektedir. Fakat grafikte bazı doğrusal olmayan eğriler olması bazı konumlarda dengesiz yük dağılımdan olduğunu göstermektedir. Ayrıca azda olsa titreşimlerin var olması mekanik sistemden kaynaklanmaktadır.



Şekil IV.8 Pozisyon modu sistem şeması



Şekil IV.9 Üç eksen için pozisyon komutu ve sistem cevabını gösteren grafik

## IV.2 UYGULAMA AYARLARI

Kontrol uygulamalarında sürücü ve kontrolörün hareketli sistemin ihtiyaçlarını karşılayabilecek biçimde ayarlanması gerekir. Uygulama limitleri programlama aşamasından önce belirlenmelidir ki, programlama sırasında bu limitler dışına çıkılmasın. Bazı durumlarda sistem gereksinimleri limitlerin dışına çıktığı gözlemlenmiştir ve bu durumda başa dönerek gerekli değişikliklerin yapılması gerektiği. Uygulama limitleri sürücü ayarları yapılmadan önce belirlendi. Uygulama limitleri gösteren uygulama ekranı Şekil IV.10'da gösterilmiştir.

Driver Parameters		
Application Continuous Current	2	[ A ]
Driver Continuous Current	3.30	[ A ]
Application Peak Current	4	[ A ]
Driver Peak Current	6.60	[ A ]

Application Mechanical Limits		
Speed	3000	[ RPM ]
Stop Deceleration (SD)	1000000000	[ cnt/sec^2 ]
Low Reference for Position	-999999990	[ cnt ]
High Reference for Position	999999990	[ cnt ]

Şekil IV.10 Uygulama limitleri

Bu limitler motor maksimum akımı ve hızı göz önüne alınarak mekanik sistemin moment ihtiyacını karşılayacak şekilde ayarlanmıştır. Uygulamanın ihtiyaçlarına göre belirlenen tüm bu değerler ve farklı modlarda elde edilen veriler sürücü flaş hafızasına kayıt edildi. Daha sonra gerek görülen noktalarda bu veriler değiştirilerek sistemin verimi ve sürekliliği sağlandı. Gerekli test ve ayarlamalardan sonra elde edilen uygulama ayarları Tablo IV.1'de görülmektedir.

**Tablo IV.2** Uygulama ayarları

<b>Geri besleme</b>	<b>Resolver</b>
<b>Kutup sayısı</b>	2
<b>Resolver çözünürlüğü</b>	16384 darbe/devir
<b>Akım aşık değeri</b>	6.6a
<b>Hız limiti</b>	6000 devir/dk
<b>Hız hata limiti</b>	12600devir/dk
<b>Uygulama sürekli akımı</b>	3A
<b>Akım modu için KP kazancı</b>	12.880
<b>Akım modu için KI kazancı</b>	15188
<b>Yumuşatma faktörü</b>	4
<b>İvme</b>	700000devir/sn <sup>2</sup>
<b>Serbest hız</b>	16384

### IV.3 PROGRAM

Çok eksenli hareketin kontrolü için gerekli olan program birden çok şekilde yazılabilmektedir. Burada önemli olan yazılan programın en az satır sayısı ve değişkenle uygulama ihtiyacını karşılamasıdır. Program vektör komutlarıyla yazılabildiği gibi eksen komutları kullanılarak da yazılabilmektedir. Vektör komutlarıyla yazılan programda vektör için gerekli yörünge tablosunu (PVT) çıkarmak gerekir. Bu sayede eksenler eş zamanlı olarak çalışırken hız ve konum gibi bilgileri değiştirerek bir birine uyumlu hareket edebilmektedir. Program eksen komutlarıyla yazıldığında her eksen ayrı ayrı kontrol edilse de, aynı komut satırı içerisinde olduğu için yine eş zamanlı olarak hareket etmektedir. Çok karmaşık bir hareket yapılmayacaksa eksen komutlarını kullanmak daha pratik bir çözüm olabilir. Matematiksel fonksiyonlar içeren elips, yay veya dairesel bir hareket yapılacaksa vektör komutları kullanılmalıdır. Aşağıdaki Örnek 1’de vektörel bir hareket programlanmıştır.

#### Örnek1:

```
function run()  
int x,y,z //Değişken tanımlama  
x=100000;y=50000;z=25000 //Koordinat bilgileri
```

```

sync(0,10) //CAN senkronizasyonu
v1.vtm=1 // Vektör yörünge hareketi
v1.vdc=3000000;v1.vac=3000000 //Yavaşlama ve hızlanma ivmesi
v1.mo=1 //Motor açık
v1.vsp=2730000 //Vektör maksimum hızı
v1.line(x,y,z) //Doğrusal hareket
v1.bg //Harekete başla
while(a1.ms==2)||(a2.ms==2)||(a3.ms==2) //Tüm eksenler durana kadar bekle
wait(1000) //1000 ms bekle
end while
end function //Program sonu

```

Örnek1’de programlanan hareket eksen komutları kullanarak Örnek2’de de programlanmıştır.

### Örnek2:

```

function run()
int x,y,z //Değişken tanımlama
x=100000;y=50000;z=25000
a1.mo=0;a1.um=5;a1.mo=1;a1.sp=10000;a1.pa=x;a1.bg; //Koordinat bilgileri
a2.mo=0;a2.um=5;a2.mo=1;a2.sp=1500;a2.pa=-y;a2.bg; //Birinci eksen komutları
a3.mo=0;a3.um=5;a3.mo=1;a3.sp=1500;a3.pa=z;a3.bg; //İkinci eksen komutları
while (a1.ms==2)||(a2.ms==2)||(a3.ms==2) //Üçüncü eksen komutları
wait(3000) //Eksenlerin tümü durana kadar bekle sonra 1sn daha bekle
end while //Program sonu
end function

```

#### IV.4 SONUÇLAR

Tüm bu sonuçlar gösteriyor ki, bir hareket kontrol mekanizmasında mekanik sistemin tasarımı ve uyumluluğu elektriksel sistemin verimi ve sürekliliği için oldukça önemlidir. Ayrıca haberleşme sistemi ve kullanılan protokolün sistem yanıt sürelerini etkilediği gözlenmiştir. Hareket kontrolünde bir diğer önemli değişken ise motor çalışma karakteristiği ve sürücünün bu karakteristik ölçütünde ayarlanmasıdır. Bir çok sürücü bu ayarları kendi otomatik olarak yapsa da, her uygulama için farklı ihtiyaçlar doğmaktadır. Bu yüzden gerekli parametre değerlerin hesaplamalar sonucu veya deney yolu ile bulunması daha sağlıklı sonuçlar vermektedir.

Hareketli bir sistem tasarlarken karşılaşılan bir diğer problemde sürekliliktir. Süreklilik için mekanik sistem tasarlanırken ileride karşılaşılabilecek tüm sorunlar önceden hesaplanmalıdır. Örneğin, önceden hesaba katılmayan bir sensör tüm tasarımı değiştirmeye neden olabilir. Yahut sisteme yüklenen aşırı yükün moment ihtiyacı sürücünün motoru kontrol etmek için uyguladığı akım değeri ile karşılanamayabilir. Karşılaşılan böyle bir durumda sürücünün limit akım değerleri değiştirilerek giderilmiştir. Fakat önceden hesaba katılmayan böyle bir değişim sistemde uyumsuzluklara neden olabilmektedir.

## BÖLÜM V

### SON DEĞERLENDİRMELER VE ÖNERİLER

Tüm bu çalışmalar gösteriyor ki, hareket kontrolünde ilk önemli adım tasarımdır. Tasarımda karşılaşılan güçlükler kontrol sisteminde daha büyük sorunlara neden olmaktadır. Mekanik sistemin elektriksel sistemle uyumlu şekilde çalışmasını sağlamak için gerekli tüm ayarlamaların yapılması gerekmektedir. Mekanik sistemin mekanik ve dinamik denklemlerinin çıkarılması bu noktada önem kazanmaktadır.

Hareket kontrolünde yaygın olarak kullanılan elektrikli servo sistemler adım motorlu veya fırçalı doğru akım motorunun kullanıldığı sistemlere göre daha karmaşık olmasına rağmen çalışma karakteristiğinin oldukça üstün olduğu gözlemlenmiştir. Çok düşük hızlarda bile sabit moment kuvveti üretebilmesi ve yüksek hızlara çıkabilmesinin yanı sıra dinamik frenleme kabiliyetinin yüksek olması hareket kontrolünde tercih edilmesinin başlıca nedenleridir. Ayrıca boyut ve ürettiği moment kuvveti oranı bakımından diğer motorlara göre oldukça üstün oldukları söylenebilir. Diğer yandan fırçasız servo motorlar için üretilen dijital sürücülerin oldukça üstün programlanabilir özelliklere ve kararlı çalışma karakteristiğini sahip olduğu görülmüştür. Kurulumu basit fakat ayar ve programlaması karmaşık olmasına rağmen farklı uygulamalar için farklı seçenekler sunması uygulanabilirliğini artırmaktadır.

Hareket kontrolünde veri iletişiminin önemi bu tez çalışmasında öne çıkan hususlardan birisi olmuştur. Kullanılan veri iletişim protokolünün farklı cihaz ve protokollerle haberleşebilmesi önemlidir. Saha veri yolunda kullanılan düğüm stratejisi bağlantı kolaylığının yanı sıra iş ağındaki eleman sayısını da muazzam şekilde artırmaktadır. Saha veri yolunda kullanılan açık mimari protokoller farklı protokoller ile iletişim kurabilmektedir. Tasarladığımız sistem CANBUS, MODBUS veya TCP/IP üzerindeki cihazlarla iletişim kurabilmektedir. Sistemin ileri uygulamalara adaptasyonu için bu çok önemli bir özelliktir. Ayrıca kullanılan sürücü ve kontrolörün analog ve dijital giriş-çıkış bağlantı seçeneklerinin olması da bu adaptasyon için gereklilik olarak ortaya çıkmıştır.

İleriki uygulamalarda mekanik sistemin tüm hesaplamalarının eksiksiz biçimde yapılması önerilir. Bu sayede kontrol aşamasında daha az problem ile karşılaşılacaktır.

Ayrıca kullanılacak motor ve sürücülerin bir biri ile uyumlu olduğu, daha önce yapılmış uygulamalar incelenerek araştırılmalıdır. Haberleşme sisteminden kaynaklanan gecikmeler önceden hesaba katılmalı ve çevre faktörleri iyi hesaba katılmalıdır. Kullanılan elemanların elektromanyetik uygunluğu ve koruma faktörleri hakkında geniş bilgi sahibi olunmalıdır. Diğer yandan elektriksel sistemin korunması için gerekli tüm tedbirler alınmalıdır. Çalışılan elemanlar karmaşık elektronik devreler içerdiği için korumalı ve şebekeden yalıtılmış güç kaynakları kullanılmalıdır. Sistemin topraklaması iyi yapılmış olmalıdır. Kullanılan güç ve geri besleme kablolarının montajına dikkat edilmeli, veri iletişimi için uygun kablolar kullanılmalıdır.

Diğer yandan bir sistem tasarlarken sistem bileşenlerini iyi tanımak gerekir. Kullanılan motorların çalışma karakteristikleri çıkarılmalı, uygulama için doğru sürücü parametreleri belirlenmelidir. Geri besleme elemanının türü uygulamaya ve çalışma ortamına uygun seçilmelidir. Kullanılan hareket modülleri ve diğer mekanik parçaların ağırlıkları, atalet momentleri ve ağırlık merkezi gibi değişkenler göz ardı edilmemelidir.

Bu çalışma sayesinde görülmüştür ki, fırçasız servo motorlar hareket kontrolü için ideal motorlardır. Kullanılan sürücüler benzerlerine göre daha az yer kaplamakta ve birçok ayarı kendisi otomatik olarak yapabilmektedir. Veri iletişimi için kullanılan saha veri yolu kurulumu oldukça basit ve kablolama maliyeti düşük bir sistemdir. Açık mimari protokoller sayesinde farklı sistemlerle haberleşebilmektedir.

## KAYNAKLAR

[1] Sarıbay, T.; “Fieldbus Teknolojisi Nereye Gidiyor?”, *Panel- Bileşim Fuar- Endüstriyel Otomasyon 2000*, İstanbul, Mart 31, (2000).

[2] Yüksel, İ.; *Otomatik kontrol sistem dinamiği ve denetim sistemleri*, Vipaş yayınları, 4.Baskı, İstanbul, (2001).

[3] Parker Motion&Control, “Step and Servo motor systems and control”, <http://www.parkermotion.com/literature/motors.pdf> (12.11.2009).

[4] Mazurkiewicz, J.; *Motion Control Basics* Baldor Electric, Forth Smith, Arkansas, USA (2000).

[5] Bosteels J.; *Design Trends “Brushless Motor Commutation”* Motion Design inc., Mayıs, (2008).

[6] Renesas Technology, “Motor Control Algorithms” [http://www.renesas.eu/http://www.renesas.eu/applications/key\\_techonologies/motor\\_control/motor\\_algorithms/child\\_folder/bldc.jsp](http://www.renesas.eu/http://www.renesas.eu/applications/key_techonologies/motor_control/motor_algorithms/child_folder/bldc.jsp) (10.04.2010).

[7] Konrad, R.: “Motion Control for Mobile Robots” *M.sc. Thesis*, Lulea University of Technology, Kiruna, Sweden, (2007).

[8] CAN in Automation e.V., “CAN History”, <http://www.can-cia.de> (2001) <http://www.can-cia.de/index.php?id=161>. (04.02.2010)

[9] *Building Adapter for Vehicle On-board Diagnostic*, <http://www.obddiag.net/http://www.obddiag.net/prog2455.html> (09.10.2009)

[10] Meineke, M.: “Fieldbus Technology in Pneumatics”, Bosch Rexroth Pneumatics, Basic Pneumatics Trainer Vol.II, New York, USA, (2006)

- [11] Chatha, A.; *Fieldbus: The Foundation for Field Control Systems* Control Engineering, May, (1994), 47–50.
- [12] Mekatronik Robotik ve Otomasyon Teknolojileri, [www.mekatronik-robotik.com.tr](http://www.mekatronik-robotik.com.tr) (2006) <http://www.mekatronik-robotik.com.tr/dm.html> (03.11.2008).
- [13] SEM Controlled Motor Techology, *Brusless AC Servo Motors*, <http://www.sem.co.uk/> (2005)  
[http://www.sem.co.uk/download.asp?f=files/Brochure\\_HRHS\\_English\\_Issue9.pdf](http://www.sem.co.uk/download.asp?f=files/Brochure_HRHS_English_Issue9.pdf) (25.04.10)
- [14] Kessler, J.; “Synchro-Resolver conversion handbook”, *Data Device Corporation*, Fourth Edition, New York, USA, (1994).
- [15] ELMO Motion Control, “Basson Digital Servo Drive”, <http://www.elmomc.com> (2008) <http://www.elmomc.com/support/manuals/MAN-BASIG.pdf> (20.10.2009)
- [16] ELMO Motion Control, “Maestro Multi-Axis Controller”, <http://www.elmomc.com> (2008) <http://www.elmomc.com/support/manuals/MAN-MASSW.pdf> (13.02.2010)
- [17] ELMO Motion Control, “SimpliQ Software”, <http://www.elmomc.com> (2004) <http://www.elmomc.com/support/manuals/MAN-SIMSW.pdf> (11.10.2009)
- [18] ELMO Motion Control, “Composer Software Manual”, <http://www.elmomc.com> (2007) <http://www.elmomc.com/support/manuals/MAN-COMPUM.pdf> (20.12.2009)

**EK.A****Operatör özellikleri****Tablo 1.** Operatör detayları

Operatör/ Tanımı	Gösterim	İşlenen	Çıkış Tipi	Örnekler
Aritmetik toplama	+	2	Tam sayı veya ondalıklı	$4+5=9$ $3.45+2.78=6.23$
Aritmetik çıkarma	-	2	Tam sayı veya ondalıklı	$4-5=-1$ $3.45-2.78=0.67$
Aritmetik çarpma	*	2	Tam sayı veya ondalıklı	$PA=PA*2$ $5*4=20$ $1.5*2=3.0$
Aritmetik bölme	/	2	Tam sayı veya ondalıklı	$20/4=5$ $3/1.5=2.0$
İki tamsayının bölümünden kalan	%	2	32-bit tamsayı	$20\%4=0$ $5\%2=1$
Bit bit değil (NOT)	~	1	32-bit tamsayı	$\sim 3 = 0xfffffc$ , aslında $-4$ $\sim 3.2$ aynıdır !3
Bit bir veya (OR)		2	32-bit tamsayı	$PA=0x2   0x5$ eşdeğer $PA=7$ $PA=0x2   5.1$ ile aynıdır
Bit bir ve (AND)	&	2	32-bit tamsayı	$PA=0x7 \& 0x3$ eşdeğer $PA=3$ $PA=0x7 \& 3.1$ ile aynıdır
Mantıksal eşitlik	==	2	0(hayır) veya 1(evet)	If $x=3$ ve $y=3$ iken $x==y$ sonuç 1 If $x=3$ ve $y=5$ iken $x==y$ sonuç 0
Mantıksal eşitsizlik	!=	2	0(hayır) veya 1(evet)	If $x=3$ ve $y=3$ iken $x!=y$ sonuç 0 If $x=3$ ve $y=5$ iken $x!=y$ sonuç 1

Mantıksal –den büyük	>	2	0(hayır) veya 1(evet)	If x=3 ve y=3 iken x>y sonuç 0 If x=3 ve y=2 iken x>y sonuç 1 If x=1 ve y=2 iken x>y sonuç 0
Mantıksal –den büyük veya eşit	>=	2	0(hayır) veya 1(evet)	If x=3 ve y=3 iken x>=y sonuç 1 If x=3 ve y=2 iken x>=y sonuç 1 If x=1 ve y=2 iken x>=y sonuç 0
Mantıksal –den küçük	<	2	0(hayır) veya 1(evet)	If x=3 ve y=3 iken x<y sonuç 0 If x=3 ve y=2 iken x<y sonuç 0 If x=1 ve y=2 iken x<y sonuç 1
Mantıksal-den küçük veya eşit	<=	2	0(hayır) veya 1(evet)	If x=3 ve y=3 iken x<=y sonuç 1 If x=3 ve y=2 iken x<=y sonuç 0 If x=1 ve y=2 iken x<=y sonuç 1
Mantıksal VE (AND): Değişkenlerin ikisinde 0 değilse 1, birisi 0 ise 0'dır.*	&&	2	0 ya da 1	1 && 5 sonuç 1 0.21 && 2 sonuç 1 0 && 2 sonuç 0
Mantıksal VEYA (OR): Değişkenlerden biri 1 ise sonuç 1,		2	0 veya 1	1  0 sonuç 1 0  0 sonuç 0

ikisinde 0 ise 0'dır.*				
Logical DEĞİL (NOT): Değişken 0 ise 1, 0 değilse 0'dır.*	!	1	0 veya 1	!4 sonuç 0 !0 sonuç 1 !0.0004 sonuç 1
Birli eksi: Değişken pozitif ise sonuç negatif, ve tam tersi*	-	1	Değişken ile aynı	-4.5 sonuç -4.5 -4 sonuç -4 (-4) sonuç 4 -5+5 sonuç 0
Bit bit sola kaydırma: İlk değişkeni ikincinin değeri kadar sola kaydırır*	<<	2	32-bit tamsayı	8<<2 sonuç 32
Bit bit sağa kaydırma: İlk değişkeni ikincinin değeri kadar sağa kaydırır*	>>	2	32-bit tamsayı	8>>2 sonuç 2

\*Değerler değerlendirmeden önce tamsayılara yuvarlanır.

**EK.B**

Hazır fonksiyon tanımlamaları

**Tablo 1.** Hazır matematik fonksiyonları.

<i>Operatör</i>	<i>Açıklama</i>	<i>Üretilen Sayı</i>
<b>sin</b>	Sine	Ondalıklı
<b>cos</b>	Cosine	Ondalıklı
<b>abs</b>	Mutlak değer Not: Onaltılık bir girişin mutlak değeri 0x80,000,000 değerini geçerse, maksimum pozitif değerle sınırlandırılır.	Girişle aynı
<b>sqrt</b>	Karekök, değişken negatif ise sıfırdır.	Ondalıklı
<b>fix</b>	Tamsayıya kırpma: fix(3.8) eşittir 3 fix (-3.8) eşittir -3 Not: Eğer giriş limitlerin dışındaysa, pozitif sayılar için maksimum, negatif sayılar için minimum değerle sınırlanır.	Tamsayı
<b>rnd</b>	En yakın tamsayıya yuvarlama: rnd(3.8) eşit 4 rnd(-3.8) eşit -4 rnd(3.4) eşit 3 Not: Eğer giriş limitlerin dışındaysa, pozitif sayılar için maksimum, negatif sayılar için minimum değerle sınırlanır.	Tamsayı
<b>sign</b>	Değişkenin işaretini verir: Negatif sayılar için -1, pozitif sayılar için 1 ve sıfır için sıfır verir sign (-3.8) sonuç -1 sign (3.8) sonuç 1	Tamsayı
<b>real</b>	Tamsayı ondalıklı gerçekte ifadeye çevirir. Sayı ondalıklı ise hiçbirşey yapmaz. 5/2 denktir 2 real (5)/2 eşittir 2.5 5/real (2) eşittir 2.5	Ondalıklı

**Tablo 2.** Matematiksel olmayan hazır fonksiyonlar

<b>Operator</b>	<b>Açıklama</b>	<b>Döngü</b>
<b>tdif</b>	Zaman farkı x=TM x=TM örneklemeinde tdif(x) msec olarak döner	Tamsayı
<b>emit(n)</b>	n'i yayımlar TPDO (CAN ileti işlem veri objesi), n eşit 1, 3 ya da 4. (sonraki bölümde ayrıntılar verilmiştir)	Tamsayı, işlem başarılı ise 1 değilse 0
<b>emcy(n)</b>	Kullanıcı programında acil mesaj yayımlar,hata kodu (32-bit tamsayı olarak) kullanıcı tarafından belirlenmiştir	Hiçbirşey
<b>PrgErr(N)</b>	Makinenin son program hatası N'ye döner, burada N -1 ile 2 arasındadır. Bu operatör AUTO_PERR ile son hatayı sorgular. N = -1 ise, SimplIQ sürücü makinenin son hata işlevine döner. Bu yüzden yorumlayıcıda PrgErr(-1) olarak ayarlamak gereksizdir.	Tamsayı
<b>tick</b>	Sistem zamanını dahili birim üzerinden okur. Bu işlem sistem zamanından bağımsız dahili bir zamanlayıcı kullanır. TM komutu CAN senkronizasyonunda zamanlayıcı olarak ayarlanabildiği gibi, tick işlevinde zamanlama damgası harici süreçlerden etkilenmez.  Tick(x) bekleme durumundaki zaman farkını işler. Geçerli sınırlar tock işlevine bağlı olarak [0....32,000] milisaniyedir. 32,000 büyük değerler için işlem OUT_OF_RANGE hata kodunu verir. Dahili elemanlarda sistem zamanının okunması için 0 ayarlanır.	Tamsayı
<b>tock</b>	Zaman farkını rapor eder. Dahili tick(x) komutunun örneklediği zamanı milisaniye olara raporlar. Tock işlevi tdif gibi çalışır, fakat sistem zamanından bağımsız dahili bir sayaç kullanır.  Not: 32 saniyeden büyük zaman farkları hata olarak sonuç verebilir.	Tamsayı

**EK.C**  
**Sistem komutları**

**Tablo 1.** Hareket komutları

<b>Komut</b>	<b>Açıklama</b>
<b>AC</b>	Hızlanma ivmesi Darbe/saniye <sup>2</sup>
<b>BG</b>	Başlama komutu
<b>BT</b>	İstenilen zamanda başlama komutu
<b>DC</b>	Yavaşlama ivmesi Darbe/saniye <sup>2</sup>
<b>IL[N]</b>	Dijital giriş görev ataması N=dijital giriş no
<b>JV</b>	Tempo değeri Darbe/saniye <sup>2</sup>
<b>MO</b>	Motor açık/kapalı 1 veya 0
<b>PA</b>	Mutlak konum referans bilgisi Darbe (counts)
<b>PR</b>	Göreceli konum referans bilgisi Darbe (counts)
<b>SD</b>	Yavaşlamayı durdurma komutu
<b>SF</b>	Yumuşatma faktörü
<b>SP</b>	Konumdan konuma hız
<b>ST</b>	Yavaşlama ivmesi ile hareketi durdurma

**Tablo 2.**Giriş-Çıkış komutları

<b>Komut</b>	<b>Açıklama</b>
<b>AN[N]</b>	Analog giriş okuma N=analog giriş no
<b>IB[N]</b>	Bit bit dijital giriş okuma N=dijital giriş no
<b>IF[N]</b>	Dijital giriş filtresi N=dijital giriş no
<b>IP</b>	Tüm dijital girişleri oku
<b>OB[N]</b>	Bit bit dijital çıkış okuma N=dijital giriş no
<b>OC[N]</b>	Çıkış karşılaştırma N=dijital giriş no
<b>OL[N]</b>	Lojik çıkış N=dijital giriş no
<b>OP</b>	Tüm dijital çıkışları ayarla

**Tablo 3.** Durum komutları

<b>Komut</b>	<b>Açıklama</b>
<b>BV</b>	Motor maksimum DC gerilimi
<b>DD</b>	CAN kontrolör durumu
<b>DV[N]</b>	Arzu edilen referans değeri
<b>EC</b>	Hata kodu
<b>LC</b>	Akım sınırlama
<b>MF</b>	Son motor hatası
<b>MS</b>	Hareket durum raporu
<b>PK</b>	Tepe hafızası
<b>SN</b>	Seri numarası
<b>SR</b>	Nümerik bit kodlu metronom durumu
<b>TI[N]</b>	Isı gösterge dizini
<b>VR</b>	Yazılım versiyonu

**Tablo 4.** Geri besleme komutları

<b>Komut</b>	<b>Açıklama</b>
<b>AB[N]</b>	Mutlak encoder ayar parametreleri
<b>ID</b>	Aktif akım okuma
<b>IQ</b>	Reaktif akım okuma
<b>PE</b>	Konum hatası
<b>PX</b>	Ana encoder konumu bilgisi
<b>PY</b>	Yedek konum bilgisi
<b>VE</b>	Hız hatası
<b>VX</b>	Ana encoder hız bilgisi
<b>VY</b>	Yardımcı encoder hız bilgisi
<b>YA[N]</b>	Yardımcı konum algılayıcı değişkeni

**Tablo 5.** Ayar komutları

<b>Komut</b>	<b>Açıklama</b>
<b>AG[N]</b>	Analog kazancı
<b>AS[N]</b>	Analog giriş sapması
<b>BP[N]</b>	Fren parametreleri
<b>CA[N]</b>	Komitasyon parametreleri
<b>CL[N]</b>	Sürekli akım sınırlaması
<b>EF[N]</b>	Encoder filtre frekansı
<b>EM[N]</b>	ECAM parametresi
<b>ET[N]</b>	ECAM tablosu
<b>FF[N]</b>	İleri besleme
<b>FR[N]</b>	Takip oranı
<b>HM[N]</b>	Dönüş ve yakalama modu
<b>HY[N]</b>	Yardımcı dönüş ve yakalama modu
<b>MC[N]</b>	Sürücü en yüksek akım eşiği
<b>MP[N]</b>	Hareket parametreleri
<b>PL[N]</b>	Tepe limiti ve süresi
<b>PM</b>	Profil modu
<b>PT</b>	Konum zaman komutu
<b>PV</b>	Konum hız zaman komutu
<b>PW[N]</b>	PWM sinyal parametresi
<b>QP</b>	Konum
<b>QT</b>	Zaman
<b>QV</b>	Hız
<b>RM</b>	Referans modu
<b>TR</b>	Hedef yarıçapı
<b>UM</b>	Birim modu: Adım, Hız, Tork, konum kontrolü v.s.
<b>VH[N]</b>	Üst referans limiti
<b>VL[N]</b>	Alt referans limiti
<b>PP[N]</b>	CAN veya RS-232 haberleşme parametresi

**Tablo 6.** Kullanıcı program komutları

<b>Komut</b>	<b>Açıklama</b>
<b>CC</b>	Program derleme
<b>CP</b>	Uygulama programı temizleme
<b>DL</b>	Program indirme
<b>HP</b>	Programı durdur
<b>KL</b>	Hareketi durdur ve programı durdur
<b>LP[N]</b>	Değişken listesi
<b>LS</b>	Program listesi
<b>MI</b>	Maskeleme
<b>PS</b>	Program durumu
<b>XC</b>	Bir sonraki noktadan programı çalıştırmaya devam et
<b>XQ</b>	Program çalıştır

**Tablo 7.** Genel kullanım komutları

<b>Komut</b>	<b>Açıklama</b>
<b>CD</b>	CPU veri bankası
<b>DF</b>	Sürücü yazılımını indir
<b>DS</b>	Sürücü yazılımını indir
<b>EO</b>	Eko modu
<b>HX</b>	Hexadecimal veya decimal mod seçimi
<b>LD</b>	Parametreleri hafızaya yükle
<b>RS</b>	Parametreleri sıfırla
<b>SV</b>	Parametreleri hafızaya kaydet
<b>TM</b>	Sistem zamanı
<b>TP[N]</b>	Ondalıklı parametre yarıları
<b>TS</b>	Örnekleme zamanı

**EK.D**  
**Eksen ve Vektör komutları**

**Tablo 1.** Eksen komutları

<b>Komut</b>	<b>Açıklama</b>	<b>Değer</b>
<b>ADI</b>	Eksen sürücü giriş bildirimi Hareket sonunda gerçekleşecek eylem için fonksiyon çağrısı gerçekleştirir.	0 veya 1 Ör: A1.ADI=1
<b>ADT</b>	Eksen gecikme zamanı Standart değer 20 milisaniyedir.	20-254 milisaniye
<b>AEL</b>	CAN encoder alt limiti	0-0x4000 integer
<b>AEH</b>	CAN encoder üst limit	0-0x4000 integer
<b>AEM</b>	CAN encoder interpolasyon zamanı	1-64 milisaniye
<b>AFP</b>	İlk anda gönderilecek satır sayısını belirler PVT tablosu en fazla 64 satır içerir	3-10 integer
<b>AID</b>	Eksen numarası	-
<b>ALN</b>	Eksen için geçerli yörünge sayısı	0,1,...,N
<b>AMC</b>	Eksen hareket sonu eylemi Eksen hareketinin bittiğini sisteme bildirir ve bir sonraki işleme izin verir	0 veya 1
<b>APE</b>	Programdan ilgili fonksiyonu çağırarak eylemi gerçekleştirir	0-15 integer
<b>APP</b>	PVT tablosundaki bölüm sayısını belirtir Sürücüye gönderilecek PVT satır sayısını sınırlar	3-10 integer
<b>ATM</b>	Hareket sonundaki konum Eksenin son konum bilgisini okur	integer
<b>addpoint</b>	Belirlenen yörünge üzerinde hareket gerçekleştirir Eksen adı.addpoint(konum,hız,zaman)	Konum=darbe Hız=darbe/sn Zaman=milisaniye
<b>dotrj</b>	Yörüngesel hareket PVT tablosunda yer alan hareketi gerçekleştirir Eksen adı.dotrj(Yörünge adı)	-
<b>ends</b>	Yörüngesel hareketin sonunun belirtir Eksen adı.ends()	-
<b>error</b>	Son hareket hatasını okur Eksen adı.error	-
<b>isok</b>	Hareket eksenini durumunun gösterir Eksen adı.isok	-
<b>startp</b>	Eksen PVT yörünge hareketini başlatır Eksen adı.startp()	-
<b>trj</b>	PVT mekanizmasını eksene yüklemeye yarar Eksen adı.trj(yörünge adı)	

**Tablo 2.** Vektör genel komutları

<b>Komut</b>	<b>Açıklama</b>	<b>Değer</b>
<b>VAC</b>	Vektör hızlanma ivmesi	Darbe/saniye <sup>2</sup>
<b>VAE</b>	Hızlanma ivmesi aşım limiti	0-1 arası
<b>VB</b>	Başlangıç komutu gecikme süresi	2-254 milisaniye
<b>VCR</b>	PVT tablosu çözünürlük sabiti	0 (Değişken) veya 1 (Sabit)
<b>VDC</b>	Vektör yavaşlama ivmesi	Darbe/saniye <sup>2</sup>
<b>VDT</b>	Dotrj komutu için gecikme süresi	20-254 milisaniye
<b>VFP</b>	PVT tablosundan yüklenecek satır sayısı	3-10 integer
<b>VID</b>	Vektör grup numarası	-
<b>VLN</b>	Vektör için geçerli olan yörüngesel döngü sayısı	0,1.....N
<b>VMC</b>	Vektörel hareket sonunda eylem icrası	0 (eylem yok) 1 (eylem var)
<b>VNT</b>	PVT için adım süresi alt limiti	1-255 milisaniye
<b>VPE</b>	PVT tablosu için konum hatası	Darbe
<b>VPP</b>	PVT için nokta bölüm büyüklüğü	3-10
<b>VSC</b>	PVT yörüngesi kıvrım modu	0-Kıvrım yok 1-En yüksek hızda yumuşatma 2-Beli bir yarıçapta yumuşatma 3-Köşeden belli uzaklıkta
<b>VSD</b>	Köşeden belirli uzaklıkta hareket yumuşatma	Darbe
<b>VSE</b>	Hareket sonunda hız ayarı ile yumuşatma	Darbe/saniye
<b>VSM</b>	Vektör komutlarının anlık veya seri olarak gönderilmesi	0-komutlar anlık gönderilir 1-Komutlar ardışık gönderilir
<b>VSP</b>	PVT yörünge hızı ayarı	Darbe/saniye
<b>VSR</b>	Hareketi belirli bir yarıçapta yumuşatma	Darbe
<b>VTM</b>	PVT hareket başlangıcı	0 veya 1
<b>VTT</b>	PVT yörünge süresi (VUM=2 iken)	milisaniye
<b>VUM</b>	PVT yörünge modu	1-En yüksek hızda 2-Belirlenen zamanda 3-Belirli bir hızda
<b>VVE</b>	PVT hareketi için hız hatası	0-1 arası % olarak
<b>VXT</b>	PVT hareketi için adım süresi üst limiti	1-255 üst limiti

**Tablo 3.** İki ve üç boyutlu vektör komutları

<b>Komut</b>	<b>Açıklama</b>	<b>Değer</b>
<b>addcircle</b>	Dairesel yörünge hareketi yapar	v1.addcircle(yarıçap,başlangıç açısı, yay açısı) int yarıçap float açı
<b>addwell</b>	PVT yörünge hareketi sonuna gecikme ekler	addwell(gecikme süresi)  milisaniye olarak
<b>addline</b>	Doğrusal hareket yapar	addline(x konum,y konum, z konum) Üç boyutlu hareket için z konumu eklenir

<b>addpoint</b>	Vektörel harekete PVT hareket noktası ekler	v1.addpoint(x konum,x hız,y konum, y hız, zaman) Üç boyutlu hareket için z konumu ve hız bilgisi eklenir
<b>attach</b>	Vektöre ait tüm elemanların eş zamanlı çalışmasını sağlamak için kullanılır	v1.attach()
<b>detach</b>	Vektör eş zamanlamasını iptal eder	v1.detach()
<b>ends</b>	Vektörel hareketin sonlandırır	v1.ends()
<b>splinee</b>	İki ve üç boyutlu kapalı yörünge hareketi yapar	v1.splinee(x,y) Üç boyutlu hareket için z konumu eklenir
<b>starts</b>	Vektör hareketini başlatır	v1.starts()
<b>error</b>	Son hatayı tanımlar	v1.isok
<b>trj</b>	Yörünge hareketini başlatır	v1.trj(yörünge ismi)
<b>clears</b>	Tüm yörünge hareketi bilgilerini siler	v1.clears()

## ÖZGEÇMİŞ

1981 yılında İstanbul'da dünyaya geldi. Babasının vefatından sonra annesinin memleketi Gaziantep'e yerleşerek, ilk ve orta eğitimini burada Akyol ilköğretim okulunda tamamladı. 1995 yılında M.R.U. Endüstri Meslek lisesi elektrik bölümüne girdi ve buradan 1998 yılında mezun oldu. Bir süre profesyonel olarak sporla uğraştıktan sonra 2000 yılında Erciyes üniversitesi Uçak elektriği bölümüne girdi fakat özel sorunlardan dolayı hazırlık okuduktan sonra birinci sınıfta okulu bırakarak bir süre yurt dışında yaşadı. Türkiye'ye geri döndükten sonra 2004 yılında Marmara üniversitesi teknik eğitim fakültesi elektrik eğitim bölümüne girdi. 2008 yılında buradan mezun olduktan sonra yüksek lisans eğitimine aynı üniversitede devam etti. Yüksek lisanstan sonra eğitimine devam etmek ve akademik kariyer yapmayı planlamaktadır.