

**GÖREREK İŞLEM YAPABİLEN DELTA ROBOTUN
TASARIMI VE PERFORMANS KARAKTERİSTİKLERİNİN
ARAŞTIRILMASI**

İsmail Hakkı ŞANLITÜRK

**Yüksek Lisans Tezi
Makine Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Ali İNAN
MART-2012**

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

**GÖREREK İŞLEM YAPABİLEN DELTA ROBOTUN TASARIMI VE PERFORMANS
KARAKTERİSTİKLERİNİN ARAŞTIRILMASI**

YÜKSEK LİSANS TEZİ

İsmail Hakkı ŞANLITÜRK

08220102

**Tezin Enstitüye Verildiği Tarih : 15 Şubat 2012
Tezin Savunulduğu Tarih : 05 Mart 2012**

**Tez Danışmanı : Prof. Dr. Ali İNAN (F.Ü)
Diğer Jüri Üyeleri : Prof. Dr. Hasan ALLİ (F.Ü)
Yrd. Doç. Dr. Burhan ERGEN (F.Ü)**

MART-2012

ÖNSÖZ

Bu çalışma, kapağında kendi adım yazıyor olmasına rağmen birçok kişinin katkısı olmadan tamamlanamazdı.

Tez konusu seçiminde beni özgür bırakan ve tezin tamamlanmasındaki tüm katkılarından dolayı danışman hocam Prof. Dr. Ali İNAN'a, robotik konusunda yardımını gördüğüm Prof. Dr. Hasan ALLİ'ye, ders dönemindeki katkılarından dolayı Yrd. Doç. Dr. Cihan ÖZEL'e görüntü işleme ve sınıflandırma konularında yardımını gördüğüm Yrd. Doç. Dr. Burhan ERGEN'e, mekanik ve ANSYS'deki katkılarından dolayı Yrd. Doç. Dr. M. Yavuz SOLMAZ ve Yrd. Doç. Dr. Mete O. KAMAN'a, tez aşamasındaki ve düzenindeki desteklerinden dolayı Arş. Gör. Yusuf BİLGİÇ'e, robotun imalatındaki desteklerinden dolayı başta Mustafa TURAN olmak üzere atelye personeline teşekkür ederim.

Mekanik kısmı bir yana, çalışmanın elektronik ve yazılım kısımları olmasaydı eksik kalacaktı. Bu konulardaki tüm emeği ve katkıyı veren, babam Halil İbrahim ŞANLITÜRK ve kardeşim Süleyman ŞANLITÜRK'e teşekkür ederim.

Son olarak, manevi ve besleyici desteklerinden ötürü annem Melek ŞANLITÜRK'e teşekkür ederim.

Arş. Gör. İsmail Hakkı ŞANLITÜRK
ELAZIĞ - 2012

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	II
İÇİNDEKİLER.....	III
ÖZET.....	V
SUMMARY.....	VI
ŞEKİLLER LİSTESİ.....	VII
TABLolar LİSTESİ.....	X
SEMBOLLER LİSTESİ.....	XI
1 GİRİŞ.....	1
1.1 Paralel Robotlar ve DELTA.....	2
1.2 Görüntü İşleme.....	4
1.3 Araştırma Konusu Olan Robotun Gelişim Süreci.....	6
1.4 Sistem Bileşenleri.....	12
1.4.1 Kamera.....	17
1.4.2 Görüntü Alıcı.....	17
1.4.3 Görüntü İşleyici.....	18
1.4.4 Sınıflandırma.....	20
1.4.5 Denetleyici.....	23
2 MATERYAL ve METOT.....	25
2.1 Kinematik Tasarım.....	26
2.1.1 İleri Kinematik.....	27
2.1.2 Ters Kinematik.....	30
2.1.3 Jakobiyen.....	31
2.1.4 Kinetostatik.....	34
2.2 Mekanik Tasarım.....	35
2.2.1 Eleman Boyutları.....	37
2.3 Gerilme Analizi.....	38
2.3.1 Robot Kolunun Gerilme Analizi.....	39
2.3.2 Sabit Platformun Gerilme Analizi.....	40
2.3.3 Taşıyıcı Şasinin Gerilme Analizi.....	40
2.4 Hassasiyet.....	41
2.5 Görme Sistemi.....	41
2.5.1 Görüntü Alıcı.....	41
2.6 Görüntü İşleme Sistemi.....	42

2.7	Sınıflandırma Sistemi	42
2.8	Denetim Sistemi	44
3	BULGULAR.....	46
3.1	Eleman Boyutları.....	46
3.2	CAD Modeli	47
3.3	Ters ve İleri Kinematik.....	48
3.4	Çalışma Hacmi	49
3.5	Hareketli Platformun Konum, Hız ve İvme Değerleri	50
3.6	Tahrik Torkları	51
3.7	Mafsal Kuvvetleri.....	52
3.8	Mukavemet Analizleri	53
3.9	Görme Sistemi	61
3.10	Sınıflandırma Sistemi	61
3.11	Tekrarlanabilirlik.....	63
3.12	İmâl Edilen Sistem	66
4	SONUÇLAR ve TARTIŞMA.....	67
5	ÖNERİLER.....	69
6	KAYNAKLAR	70
EK – 1:	Optimizasyon Programı	74
EK – 2:	Amaç Fonksiyonu	75
EK – 3:	Sınırlayıcılar	76
EK – 4:	Görüntü Alıcı	77
EK – 5:	Görüntü İşleme Programı.....	78
EK – 6:	Sınıflandırma Sistemi.....	79
EK – 7:	Öznitelik Çıkarma Programı	80
EK – 8:	Denetim Sistemi Bilgisayar Kısımı.....	81
EK – 9:	Denetim Sistemi Mikroişlemi Kısımı	83
EK – 10:	İleri Kinematik Programı	85
EK – 11:	Ters Kinematik Programı.....	86
EK – 12:	Jakobiyen Programı	87
EK – 13:	Homojen Dönme Matrisi Hesaplama Programı.....	88
EK – 14:	Öteleme Matrisi Hesaplama Programı.....	89
EK – 15:	Robot Çizdirme Programı	90
EK – 16:	Motor Özellikleri.....	91
EK – 17:	Elektronik Devre	93
EK – 18:	PIC16F877 Bacak Bağlantıları	94
ÖZGEÇMİŞ	95

ÖZET

Bu çalışmada görerek işlem yapabilen Delta robotun tasarımının yapılması ve performans karakteristiklerinin araştırılması amaçlanmıştır. Bunun için robotun analitik modeli oluşturulmuş, bu model kullanılarak boyutları optimize edilmiştir. Sonra gerilme analizleri yapıp istenen şartlarda çalışıp çalışamayacağı belirlenmiştir. Atalet ve yerçekimi kuvvetleri de bu modele etkiletilmiştir. Robotun görme, sınıflandırma ve kontrol yazılımları hazırlandıktan sonra sistemin imalatı yapılmıştır. Son olarak deneysel çalışmalar tamamlanmıştır. Sonuçta görerek işlem yapabilen bir sistem hazırlanmıştır.

Anahtar kelimeler: Görme Güdümlü, Paralel Robot, Dinamik Gerilme

SUMMARY

Design and Investigation of Performance Characteristics of a Vision Guided Delta Robot

In this study, design and investigation of performance characteristics of a vision guided Delta robot has been aimed. The robot's analytical model has been made, and by using that model in a written optimization script, dimensions of the robot's elements has been determined. After that, to observe, if the robot have capabilities to suffer given conditions, such as payload and safely running, a stress analysis has been run. Inertial and gravity loads has been applied to the model. After the robot's vision, classification and control scripts have had written, it has been manufactured. At last, experimental studies has been completed. As a result, a vision guided system has been designed and manufactured.

Keywords: Vision Guided, Parallel Robot, Transient Stress

ŞEKİLLER LİSTESİ

	<u>Sayfa No</u>
Şekil 1.1 J.E. Gwinnet tarafından 1928'de patenti alınan küresel mekanizma [1]	2
Şekil 1.2 Gough platformu (1947) [1]	2
Şekil 1.3 Minnesota Üniversitesinde deprem simülasyonunda kullanılan sarsma masası [1]	3
Şekil 1.4 Stewart platformu (1965). Platformun hareketi, tahrik edilen 6 adet lineer eyleyici tarafından sağlanmaktadır [1]	3
Şekil 1.5 EPFL'nin Delta robotu gerçekleştirilmesi, ABB Flexpicker [2]	3
Şekil 1.6 İnsan hareketi çalışması (Eakins Thomas, 1844-1916) [5]	4
Şekil 1.7 1921'de elde edilen dijital fotoğraf [6]	5
Şekil 1.8 Ay'ın bir U.S.A. uzay aracı tarafından çekilen ilk fotoğrafı [6]	5
Şekil 1.9 KARES'in blok diyagramı [24]	13
Şekil 1.10 Dinamik hedef takibi yapabilen bir robot bileşenleri [23]	14
Şekil 1.11 Sınıflandırıcı robot bileşenleri [27]	14
Şekil 1.12 Görerek montaj yapabilen bir sistemin bileşenleri [29]	14
Şekil 1.13 İnceleme birimi [30]	15
Şekil 1.14 Lazerle hizalama kontrolü yapan bir inceleme sistemi [35]	15
Şekil 1.15 Duvardaki çatlakların tespiti için geliştirilen bir robotun bileşenleri [33] ...	16
Şekil 1.16 Sınıflandırma yapan bir sistemin bileşenleri [37]	16
Şekil 1.17 Uzay çalışmaları için geliştirilen, görebilen bir robotun blok diyagramı [38]	17
Şekil 1.18 Et görüntüsü üzerinde yapılan işlemler a)2 bitlik görüntü, b)Üzerinde işlem yapılacak bölge, c)Bölge büyütülmüş halde, d)Kenar yumuşatma, e)Kenar belirleme, f)Gürültü yok etme[31]	18
Şekil 1.19 Nesne merkezinin algılanması [29]	18
Şekil 1.20 Yüzeyin biçimini fotoğraftan belirleyebilen sistem: a)Çekilen fotoğraf, b)Kenar belirleme, c)Görüntünün tersini alma, d)Daireye tamamlama, e)İçini doldurma, f)Lazer bilgisinden oryantasyonu elde etme [32]	19
Şekil 1.21 Yalnızca istenen cismin görüntüden ayrılması [34]	19

Şekil 1.22 Lazer tarama yöntemiyle alınan görüntü [35]	19
Şekil 1.23 Otomatik çatlak inceleme sisteminin görüntü işleme algoritması [33].....	20
Şekil 1.24 Tüp sınıflandırma sisteminde yapılan görüntü işleme [37].....	20
Şekil 1.25 SOM ağıının temel yapısı [42]	21
Şekil 1.26 Karakter tanıma için giriş matrisi [43]	22
Şekil 1.27 Karakter tanıma özdüzenleyici haritası [43]	22
Şekil 1.28 Doku sınıflandırma blok şeması [45]	23
Şekil 1.29 Hareketli hedefi takip eden sistemin kontrolü [23]	24
Şekil 1.30 KARES sisteminin kontrolü [24]	24
Şekil 1.31 Yapay Sinir Ağı ile kontrol [26].....	24
Şekil 1.32 Hareketli nesne yakalayabilen bir robot kontrol şeması [34]	24
Şekil 2.1 Kinematik çiftler ve simgeler.	26
Şekil 2.2 Tahrik edilen mafsal [3]	27
Şekil 2.3 Robotun elemanları ve tahrik mafsallarının pozitif yönleri	35
Şekil 2.4 Tork optimizasyonu algoritması.....	38
Şekil 2.5 Esnek elemanların ağ yapısı (a) ve analizin sınır şartları (b)	39
Şekil 2.6 Sabit platform için ağ yapısı (a) ve sınır şartları (b).....	40
Şekil 2.7 Taşıyıcı şasi için ağ yapısı (a), ve sınır şartları (b).....	40
Şekil 2.8 Görüntü alma algoritması	41
Şekil 2.9 Görüntü işleme algoritması	42
Şekil 2.10 Nesne sınıflandırma algoritması.....	43
Şekil 2.11 Öznitelik çıkartma algoritması	44
Şekil 2.12 Robotun kontrol algoritması.....	45
Şekil 3.1 Eleman boyutu optimizasyonu	46
Şekil 3.2 Sistemin CAD modeli	47
Şekil 3.3 Verilen konumdan hesaplanan ters kinematik çözüme göre çizdirilen ileri kinematik çözüm	48
Şekil 3.4 Kinematik ifadelerin CAD ortamında doğrulanması	48
Şekil 3.5 Robotun çalışma hacmi	49
Şekil 3.6 Hareketli platformun konumunun zamana göre değişimi	50
Şekil 3.7 Hareketli platform hızının zamana göre değişimi	50
Şekil 3.8 Hareketli platform ivmesinin zamana göre değişimi	51

Sayfa No

Şekil 3.9 Robotu tahrik etmek için gerekli tork.....	52
Şekil 3.10 1. Kolun tahrik edilen mafsalsına gelen kuvvetler.....	53
Şekil 3.11 Mil için Gerilme-Zaman grafiği	53
Şekil 3.12 Kama için Gerilme-Zaman grafiği	54
Şekil 3.13 Kol için Gerilme-Zaman grafiği.....	55
Şekil 3.14 Üst Mafsal için Gerilme-Zaman grafiği	55
Şekil 3.15 Üst Mafsal Mili için Gerilme-Zaman grafiği	56
Şekil 3.16 Paralel Kol için Gerilme-Zaman grafiği.....	56
Şekil 3.17 Alt Mafsal için Gerilme-Zaman grafiği.....	57
Şekil 3.18 Alt Mafsal Mili için Gerilme-Zaman grafiği.....	57
Şekil 3.19 Bilek Mafsalsı için Gerilme-Zaman grafiği.....	58
Şekil 3.20 Bilek için Gerilme-Zaman grafiği	58
Şekil 3.21 Sabit platform için Gerilme-Zaman grafiği.....	59
Şekil 3.22 Taşıyıcı şasi için Gerilme-Zaman grafiği	59
Şekil 3.23 Taşıyıcı şasinin mod biçimleri ve ilgili doğal frekansları	60
Şekil 3.24 Görüntü işleme adımları	61
Şekil 3.25 Ağın eğitimi için kullanılan resim.....	61
Şekil 3.26 Eğitim sonucu yapılan sınıflandırma.....	62
Şekil 3.27 Nesnelerin sınıf aitlikleri	62
Şekil 3.28 Sınıflandırma için ağa gönderilen resim ve sınıflandırma sonuçları.....	62
Şekil 3.29 9° (30 darbe) için darbe sayısında meydana gelen hata miktarları.....	63
Şekil 3.30 $-13,5^{\circ}$ (45 darbe) için darbe sayısında meydana gelen hata miktarları	64
Şekil 3.31 9° için eksen bazında konumda meydana gelen hata miktarları.....	64
Şekil 3.32 $-13,5^{\circ}$ derece için eksen bazında konumda meydana gelen hata miktarları. 65	
Şekil 3.33 CAD modellerine uygun olarak imâl edilen robot	66

TABLolar LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1 Tahrik edilen elemanın uç konumları.....	28
Tablo 2.2 Motor konumlarının zamana göre aldıkları değerler.....	36
Tablo 2.3 Malzeme özellikleri.....	39
Tablo 3.1 Elemanların boyutları	47

SEMBOLLER LİSTESİ

R_b	: Sabit platformun yarıçapı (mm)
R_p	: Hareketli platformun yarıçapı (mm)
l_1	: Tahrik edilen kolun uzunluğu (mm)
l_2	: Paralel kolun uzunluğu (mm)
d_b	: Sabit platform ile tahrik mafsalı arasındaki mesafe (mm)
d_p	: Hareketli platform ile tutucu uç arasındaki mesafe (mm)
B_i, B'_i	: Tahrik edilen kolun sabit eksen takımına göre başlangıç ve bitiş koordinatları (mm)
P_i	: Hareketli platformun sabit eksen takımına göre mafsal koordinatları (mm)
O	: Sabit eksen takımı
P	: Hareketli eksen takımı
x, y, z	: Sabit platform eksen adları
x', y', z'	: Hareketli platform eksen adları
b_{ix}, b_{iy}, b_{iz}	: i. bacak mafsalinın başlangıç noktasının sabit eksen takımına göre koordinatları (mm)
$b'_{ix}, b'_{iy}, b'_{iz}$: i. bacak mafsalinın bitiş noktasının sabit eksen takımına göre koordinatları (mm)
$b''_{ix}, b''_{iy}, b''_{iz}$: i. bacak mafsalinın bitiş noktasının sabit eksen takımına göre R_p kadar ötelenmiş koordinatları (mm)
i	: Bacak indisi
θ_{il}	: Tahrik edilen mafsalin x eksenine göre z eksenine doğrultusundaki açısı ($^{\circ}$)
β_i	: Tahrik edilen mafsalin x eksenine göre y eksenine doğrultusundaki açısı ($^{\circ}$)
$h_1, h_2, j_1, j_2, k, w_i$: İleri kinematik ifadesindeki kısaltma sembolleri
A, B, C	: İleri kinematik denklem katsayıları
p_x, p_y, p_z	: Tutucu ucun sabit eksen takımına göre koordinatları (mm)

E_i, F_i, G_i, H_i, K_i	: Ters kinematik denklem katsayıları
Q_{ii}	: Tahrik edilen elemanın sabit eksen takımına göre dönme matrisi
Q	: Hareketli eksen takımının sabit eksen takımına göre dönme matrisi
r'_i	: Tutucu ucun hareketli eksen takımına göre koordinatları (mm)
\mathbf{t}	: Hareketli platformun sabit eksen takımına göre hız vektörü
\mathbf{A}	: Kartezyen uzayı katsayı matrisi
\mathbf{B}	: Eklem uzayı katsayı matrisi
J	: Jakobiyen matris
\mathbf{n}^T	: Tutucu uca etkiyen tork vektörü (Nmm)
\mathbf{f}^T	: Tutucu uca etkiyen kuvvet vektörü (N)
τ	: Tahrik edilen mafsallarda meydana gelen tork vektörü (Nmm)

1 GİRİŞ

Endüstride insanların yapmaktan sıkılacağı rutin işler robotlara yaptırılmaktadır. Bu robotların her farklı iş için programlanması gerekmektedir. Bu farklı işler için gereken programları robot kendi hafızasında tutabilmektedir. Bu yüzden istendiği anda aynı robot farklı bir işlemde, programı değiştirilmek suretiyle kullanılabilir. Bu tip robotlar daha çok yapılacak işin kesin belli olduğu, rutin veya insanların ya çok yavaş ya da tehlikeye maruz kalacağı durumlarda kullanılır.

Günümüz endüstri şartlarında ise esnek üretimin gereği olarak sürekli değişken ve dinamik durumlarla başa çıkabilecek robotlar gerekmektedir. Bu sayede robot ek programlama, ayar ve tanımlama işlemlerine gerek kalmadan sorunsuz olarak kullanılabilir. Bu yetenekler robotlara kamera entegre ederek onları görebilir hale getirmek suretiyle kazandırılmaktadır. Bunun için kameradan gelen görüntüyü işlenebilir bir fotoğrafa dönüştürebilen bir dönüştürücü, bu görüntülerden istenen özellikleri çıkarabilen bir görüntü işleyici ve robottan beklenen işlemleri yapabilecek şekilde kumanda edilmesi gereken denetim birimine komutları ileten bir merkezi denetim yazılımı gerekmektedir.

Bu çalışmada tut-yerleştir (Pick&Place) işlemlerinde kullanılan, görerek işlem yapabilen bir Delta robotun tasarımı yapılmış ve performans karakteristikleri araştırılmıştır. Delta robot paralel yapıda olan bir robottur. İlerleyen bölümde tanıtılmıştır.

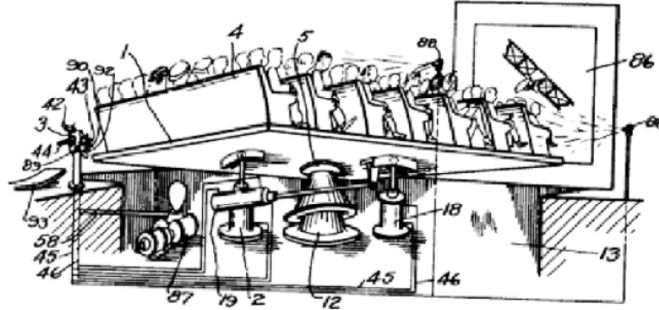
Tut-yerleştir işlemleri bir taşıma bandı üzerinde ilerleyen cisimleri özelliklerine göre ayırma, dizme, konumlarını veya oryantasyonlarını değiştirme olarak tanımlanabilir.

Araştırmanın sonucunda sanayide; işçi gücüyle yapılan tut-yerleştir işlemlerindeki açığı kapatabilecek bir robotun endüstri ortamında çalışabilmesi için gerekli altyapı hazırlanmıştır.

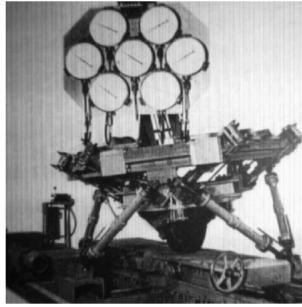
1.1 Paralel Robotlar ve DELTA

Merlet (2006), şöyle demektedir [1]:

“Paralel robotlar, birbirine seri olarak bağlanan uzuvların, biri sabit diğeri hareketli olmak üzere iki gövdeye bağlanıp, bir simetri ekseninde 2 veya daha fazla sayıda yerleştirilmesiyle elde edilen kapalı tip paralel mekanizmalardır. Seri manipülatörlerde tutucu uç yere ek elemanlarla bağlanıp yükün bir kısmının taşınmasını sağlayabilmektedir. Böylece kapalı tip bir mekanizma elde edilmektedir. Kapalı tip mekanizmalara ait teorik çalışmalar 1645’te Christopher Wren’in yaptığı çalışmalara dayanır. Daha sonra sırasıyla, 1813’te Cauchy, 1867’de Lesque ve 1897’de Bricard’ın çalışmaları vardır. Bu alandaki esas problemlerden biri “küresel hareket problemi”dir. Bu problem 1900’lerin başında Fransa’da Academie des Sciences tarafından düzenlenen Le Prix Vaillant adlı ödüllü yarışmayı kazanan Borel ve Bricard tarafından çözülmüştür. 1928’de J.E. Gwinnett’in aldığı patentteki sinema sahnesinde, küresel bir mekanizma kullanılmıştır. 1947’de Gough lastik aşınmasında kullanılmak üzere tasarladığı, Hexapod isimli 6 serbestlik dereceli mekanizmada kapalı tip mekanizmaların hesaplama temellerini ortaya koymuştur. İşlevsel bir hexapodun prototipini ilk Gough’un yapmasına rağmen, hexapodlar MAST (Multi-Simulation Table) adıyla kullanılan, iyi bilinen sistemlerdi. Bu tip mekanizmalar yaygın olarak ilk defa 1960’larda havacılık endüstrisinin gelişmesine paralel olarak artan pilot eğitim masrafları ve yerde test edilmesi gereken uçak parçalarının geliştirilmesinde kullanılmaya başlandı. 1965’de Stewart, bu amaçla bugün de Stewart Platformu olarak bilinen uçak simülatörünü geliştirmiştir.”



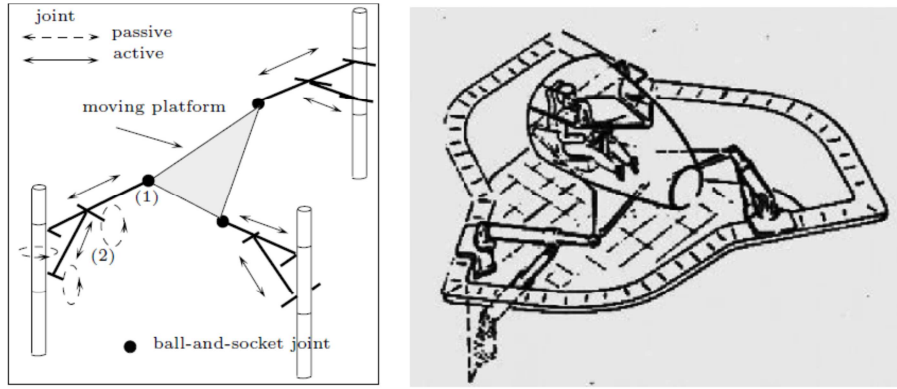
Şekil 1.1 J.E. Gwinnet tarafından 1928'de patenti alınan küresel mekanizma [1]



Şekil 1.2 Gough platformu (1947) [1]



Şekil 1.3 Minnesota Üniversitesinde deprem simülasyonunda kullanılan sarsma masası [1]



Şekil 1.4 Stewart platformu (1965). Platformun hareketi, tahrik edilen 6 adet lineer eyleyici tarafından sağlanmaktadır [1]

Angeles (2006), şöyle demektedir [2]:

“Hız amaçlı, endüstriye yönelik paralel robotlar da geliştirilmiştir. Örneğin 1988’de Clavel tarafından Lausanne Federal Polytechnic Institute (EPFL) ‘de Delta robotu geliştirilmiştir. Delta robotu 3D uzayda öteleme hareketleri yapmak üzere tasarlanmıştır.”



Şekil 1.5 EPFL'nin Delta robotu gerçekleştirilmesi, ABB Flexpicker [2]

Bu tip robotların, endüstride taşıma bantlarında ilerleyen ürünlerin tut-yerleştir işlemlerinde, otomobillerin fabrika ortamındaki yol simülasyonlarında ve takım tezgahı olarak da kullanımları mevcuttur [2]. Savunma ve uçak sanayinde ise, uçak simülatörlerini bu tip robotlar hareket ettirmektedir [2]. Eğlence sektöründe ise 3 duyuya hitap edebilen sinemalarda kullanılmaktadır [3].

Khatib ve Siciliano (2008), şöyle demektedirler [4]:

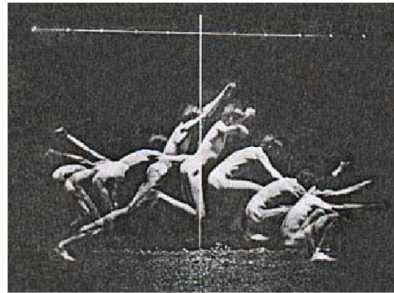
“Paralel robotlar seri manipülatörlerle karşılaştırıldıklarında rijitlikleri ve müsaade edilen atalet kuvvetleri daha yüksektir. Dolayısıyla konstrüksiyonları daha hafif olmakta ve süratli çalışabilmektedirler. Ancak; bütün seri uzuvların ortak bir gövdeye bağlanmaları, imalât toleransları, montaj düzeni ve denetimde bazı zorluklar getirmektedir.”

1.2 Görüntü İşleme

Zhou ve diğerleri (2010), şöyle demektedirler [5]:

“Dijital görüntü işleme, dijital resimler üzerinde bilgisayar algoritmalarının çalıştırılmasıyla yapılan işlemlerdir. Bu işlemlerin çıktıları yine resimler veya resimlere ait bazı karakteristik özellikler olabilmektedir. Dijital görüntü işlemenin uygulama alanları çoğunlukla robotik/akıllı sistemler, tıbbi görüntüleme, uzaktan algılama, fotoğrafçılık ve adli vakalardır. Dijital görüntü işleme direkt olarak çoklu resim noktalarından oluşan resmin kendisiyle ilgilenir. Benek (pixel) isimli bu noktalar resimdeki koordinatları belirtirler ve yoğunluk (veya gri seviye) değerlerini tutarlar.”

Bilimsel olarak görüntü işlemenin başlangıç tarihi 1844’e uzanmaktadır [5]. Eakins Thomas’ın insan hareketlerini incelemek için faydalandığı bir görüntü Şekil 1.6’da görülmektedir.



Şekil 1.6 İnsan hareketi çalışması (Eakins Thomas, 1844-1916) [5]

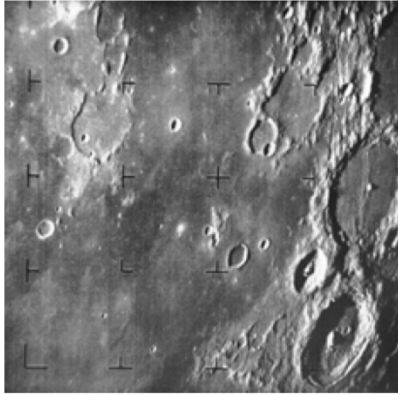
Gonzalez ve Woods (2002), şöyle demektedirler [6]:

“Dijital görüntülerin ilk uygulamaları, resimlerin deniz altından kablo vasıtasıyla Londra’dan NewYork’a gönderilmesiyle gazetecilik endüstrisinde olmuştur. 1920’lerin başında Bartlane kablodan resim alış-verişi sisteminin kurulmasıyla 1 haftadan uzun süren Atlantik üzerinden resim gönderme işlemi 3 saatten az bir sürede tamamlanır hale gelmiştir. Özel bir ekipman resimleri kablodan transfer için kodlamış ve alan tarafta yeniden oluşturmuştur. Şekil 1.7’de bu yolla gönderilmiş bir fotoğraf görülmektedir.”

“Bu yolla elde edilen görüntüler dijital olmalarına rağmen oluşturulmalarında bilgisayar içermedikleri için dijital görüntü işleme olarak değerlendirilmezler. Resimler üzerinde bilgisayar kullanılan çalışmalar ilk olarak 1964’de Jet Propulsion Laboratory’de (Pasadena, California, USA) Ranger 7 uzay aracının gönderdiği fotoğrafları düzenlemek amacıyla yapılmıştır. Şekil 1.8’de 31/07/1964 tarihinde 9:09 (EDT) saatinde Ranger 7 uzay aracından, Ay’ın yüzeyine çarpmadan 17 dakika önce gönderilen fotoğraf görülmektedir. Fotoğraf üzerindeki kareler alınan farklı fotoğrafların hizalanmasında kullanılan işaretleme noktalarını göstermektedir.”



Şekil 1.7 1921’de elde edilen dijital fotoğraf [6]



Şekil 1.8 Ay’ın bir U.S.A. uzay aracı tarafından çekilen ilk fotoğrafı [6]

1.3 Araştırma Konusu Olan Robotun Gelişim Süreci

Miller (1995), Delta robotun Hamilton prensibiyle elde edilen dinamik modeliyle deneysel tork ölçümlerini karşılaştırmış ve birbirlerine sürtünmesiz eklemlerle bağlı kabul ettiği rijit elemanlarla modellenen Delta robot modelinin doğru olduğunu ortaya koymuştur [7].

Vischer ve Clavel (1998), Delta robotun kinematik kalibrasyonu için, biri konum diğeri oryantasyon olmak üzere iki model geliştirmişler, bu modellere göre kalibre edilen robotların, kalibrasyonsuz robotlara göre konum doğrusallığında 12,7, oryantasyon doğrusallığında 3,7 kat artış gözlemlemişlerdir [8].

Staicu ve Carp-Ciocardia (2003), bir çalışmalarında konumu ve hareketi bilinen bir Delta robotun dinamik modelini iteratif matris eşitlikleriyle elde etmişler ve elde ettikleri sayısal sonuçların deneysel verilerle uyum içinde olduğunu ortaya koymuşlardır [9].

Angel ve diğerleri (2004), bir çalışmalarında Delta robottan esinlendikleri bir paralel robota bir görme sistemi entegre etmişler; geliştirdikleri algoritmayla robotun masa tenisi oynayabilmesini sağlamışlardır [8]. Bu robotun ADAMS kullanarak dinamik analizini yapmışlar ve elde ettikleri sonuçlara göre eleman uzunluklarını ve yeterli güçte motorları seçebilmişlerdir [10].

Zsombor-Murray (2004), bir çalışmasında Delta robotun ileri ve ters kinematik ifadelerini merkez koordinatı sabit olan bir daire ve merkez koordinatı hareketli bir kürenin kesişim noktasını analitik olarak modelleyerek elde etmiş ve bu hesaplamaları yapabilen bir bilgisayar programı geliştirmiştir [11].

Li ve Xu (2005), bir çalışmalarında kalp masajı yapabilen bir Delta robot geliştirmişler ve robotların bu tür işlerde doktorların işlerini kolaylaştırabildiğini ortaya koymuşlardır [12].

Staicu ve Carp-Ciocardia (2005), bir çalışmalarında konumu ve hareketi bilinen paralel eyleyicilerle tahrik edilen bir Delta robotun dinamik modelini iteratif matris eşitlikleriyle elde etmişlerdir [13].

Merlet ve Donelan (2006), bir çalışmalarında paralel robot jakobiyeni tekilliğini incelemişler, bu tekilliklerin robotun ulaşabileceği uç konumlarla ilgili olduğunu belirlemişlerdir [14].

Nabat ve diğerleri (2006), bir çalışmalarında Delta robotun hareketli platformunu şekil değiştirebilir hale getirip, robota dördüncü bir kol eklemişler ve performans karakteristiklerini incelemişlerdir. Newton-Euler yaklaşımıyla modelledikleri dinamik modelin deneysel sonuçlarla uyum içinde olduğunu ortaya koymuşlardır [15].

Güneri ve diğerleri (2007), Stewart platformunu ele aldıkları bir çalışmalarında eleman uzunluğu, eklem açıları sınırları ve eklem konumunu belirleyen açıları parametre olarak çalışma düzlemi, eyleyici kuvvetleri ve eklem konumlarını incelemişler ve optimum tasarımı bu parametreleri baz alarak tartışmışlardır [16].

Charfia ve diğerleri (2007), bir çalışmalarında pasif eklem monte ettikleri bir paralel robotun kartezyen ve eklem uzayındaki hız karakteristiklerini ters jakobiyen matrisini elde ederek incelemişler, çalışma hacmini elde etmişler ve imal ettikleri prototipten elde edilen sonuçlarla sayısal sonuçların uyum içinde olduğunu ortaya koymuşlardır [17].

Laribi ve diğerleri (2007), bir çalışmalarında çalışma hacmi belirli olan bir Delta robotun optimum boyutlarının genetik algoritma temelli bir metot kullanarak elde etmişlerdir [18].

Selvakumar ve diğerleri (2009), bir çalışmalarında takım tezgâhı olarak kullanılabilir 3 eksenli bir paralel manipülatörü, hareketi vida mafsalından küresel mafsala ileten elemanlar kullanarak modellemişler, ADAMS'da çalışma hacmini analiz etmişler, aldıkları sonuçları MATLAB'a giriş olarak verip Pappus-Guldinus teoremiyle elde ettikleri robotun çalışma hacminin ADAMS'da elde edilen hacimle uyum içinde olduğunu görmüşlerdir. Ayrıca hareketli platformun yarıçapı sabitken eleman uzunluğunun

çalışma hacmiyle; eleman uzunluğu sabitken hareketli platform yarıçapının çalışma hacmiyle direkt olarak orantılı olduğunu elde etmişlerdir [19].

Rueda ve Angel (2009), bir çalışmalarında bir Delta robot için belirledikleri kritik yörünge üzerinde dinamik gerilme analizi yapmışlar, bu analizin sonunda robot konstrüksiyonunu optimize etmişler ve sonuçta ilk modele oranla robotun kütesinin 2/3 oranında; robotu hareket ettirmek için gerekli torkların da %50 oranında düştüğünü ortaya koymuşlardır [20].

Staicu (2009), bir çalışmasında dinamik karakteristikleri bilinen ve döner eyleyicilerle tahrik edilen 3 serbestlik dereceli bir paralel robotun lineer eyleyicilerle tahrik edilmesi durumunda gerekli tork ve kuvvetlerin karşılaştırmasını yapmış; en düşük güç ihtiyacının hareketli platformun saf dikey hareketi esnasında gerektiğini ortaya koymuştur [21].

Erzincanlı ve Sharp (1997), gıda maddelerinin geometrik özelliklerini sınıflandırabilen bir kodlama sistemi geliştirmişler ve her bir gıda maddesinin fiziksel özelliklerini de dikkate alarak, bu maddelere zarar vermeden tutulup taşınmalarını sağlayabilecek robot tutucu özelliklerini listeleyp bir tutucu tasarımı ortaya koymuşlardır [22].

Lee ve Qian (1998), bir çalışmalarında görme güdümlü bulanık mantık kontrol sisteminin, hareket eden nesnelerin takip edilmesinde kullanılabileceğini, sistemin ayarlanmasının PID kontrolde olduğu gibi yapılabileceğini ortaya koymuşlardır [23].

Song ve diğerleri (1999), ellerini kullanamayan insanlara yardımcı olması için renkli olarak görebilen bir robot kol tasarlamışlar ve teste tabi tutarak; tutma, kaldırma, taşıma ve anahtar açıp kapatma işlemlerinin % 100 başarıyla yapıldığını ortaya koymuşlardır [24].

Bone ve Capson (2003), fikstürsüz montaj yapabilen bir sistem geliştirmişlerdir. Sistem montajı yapılacak parçalardan ilkinin 2 boyutlu görüntü işlemeyle $\pm 1\text{mm}$ konum ve $\pm 0,2^\circ$ oryantasyon hatasıyla yerleştirip kalan parçaları da bu parçaya 3 boyutlu görüntü işlemeyle monte edilebileceğini göstermişlerdir [25].

Lopez ve diğeri (2004), nöro-bulanık denetim tekniği ile robot tutucusunun eğitici ve eğitici olmayan öğrenme kabiliyetlerini incelemişler ve eğitici öğrenmenin daha kısa sürede tamamlanıp daha başarılı sonuçlar verdiğini ortaya koymuşlardır [26].

Mitziyaz ve Mertziyos (2004), yapay sinir ağlarını kullanarak NEMUS (NEural MUlticlassifier System) adında görüntüsü alınan nesnelerin sayısı, konum ve açılarını manipülatöre kılavuzluk edebilecek şekilde kesin olarak belirleyen bir sınıflandırıcı geliştirmişlerdir [27].

Edinbarought ve diğeri (2005), elektronik bileşenlerin lehimlerinin kalite kontrolünü, geliştirdikleri yapay sinir ağı algoritmasını görebilen bir sistemde kullanarak yapmış oldukları çalışmalarında hatalı üretimleri %100 oranında belirleyebilmişlerdir [28].

Juarez ve diğeri (2005), 6 eksenli robot, taşıma bandı, montaj tablası, kamera ve bilgisayardan oluşan, kameranın DSP kartı aracılığıyla bilgisayara, bilgisayarın da RS232 portundan robota bağlı olduğu, kameranın çektiği nesne fotoğraflarının ART (Adaptive Resonance Theory) tekniğiyle oluşturulan yapay sinir ağından geçirilip monte konumlarının belirlendiği, fikstüre ihtiyaç duymayan, ortalama hesaplama süresi 1,23ms olduğu için gerçek zamanlı uygulamalarda da kullanılabilen, yardımcı bilgi olarak sensörlerden elde edilen temas kuvvetlerinin kullanıldığı bir montaj algoritması geliştirmişlerdir [29].

Pires ve Paulo (2005), klozet ve lavabo gibi düzlemsel olmayan seramik ürünlerin, üretim hattının sonunda insanlar tarafından yapılan kalite kontrolünün otomatik olarak yapılabilmesi için, iki manipülatör, iki taşıma bandı, bir görme sistemi ve bir bilgisayardan oluşan bir sistemin otomatik olarak çalışmasını sağlayacak bir yazılım geliştirmişler ve geliştirdikleri yazılımın denetlediği sistemin sorunsuz bir şekilde çalıştığını günde iki vardiyaya çalıştırıp, her vardiyada yaklaşık 1400 parçanın incelenmesini %100 başarıyla yapabildiğini ortaya koymuşlardır [30].

Du ve Sun (2006), gıda kalite kontrolünü bilgisayar görmesiyle sağlamak amacıyla kullanılan teknikleri incelemişler, Yapay Sinir Ağları ve İstatistiksel Öğrenmenin temel öğrenme metodu olduğunu, ileriki çalışmalarda ise Yapay Sinir Ağları, İstatistiksel

Öğrenme, Bulanık Mantık ve Genetik Algoritma tekniklerinin her birinin güçlü yönlerinin birleştirilmesi suretiyle daha gelişmiş öğrenme sistemlerinin geliştirilebileceğini ortaya koymuşlardır [31].

Tsai ve diğerleri (2006), fotoğrafını çekmek suretiyle kuadratik yüzeylerin tanınabildiği bir sistem geliştirmişlerdir. Yüze lazer ışını yansıttıktan sonra elde edilen noktalardan yüzeyi elde edebilmek için ayrıca yüzeyin fotoğrafını çekmişler, lazer ve görüntü verisini çakıştırıp, yüzeyin sonradan da düzenlenebilmesi için IGES formatında 3 boyutlu görüntüsünü elde etmeye çalışmışlardır. Yaptıkları deneylerde ortalama ölçme hatasını 0,2685mm, en yüksek doğruluğu, uzunluğu 100mm genişliği 50mm olan hiperbol tipi yüzeylerde 0,284157mm, en yüksek hatayı ise çapı 100mm olan dairesel yüzeylerde 5,647792mm olarak elde etmişlerdir [32].

Yu ve diğerleri (2007), karayolları tünellerindeki çatlakları ve diğer hasarları otomatik olarak tespit edebilmek için tasarladıkları hareketli robot sisteminde bu hasarları %85 doğrulukla tespit edebilmişler ve çatlakların boyutlarını tam olarak ölçmüşlerdir [33].

Freda ve Oriolo (2007), hareket eden bir hedefin yakalanmasında kullanılan bir robot geliştirmişlerdir. Geliştirdikleri sistem tahmine dayalı Tahmin et-Planla-Uygula (Prediction-Planning-Execution) yönteminden farklı olarak herhangi bir tahmin yapmadan yalnızca hareket eden cismin görüntü verisiyle işlem yaparak, hareketli cismin konumunu hesaplayıp robotu belirlenen çakışma noktasına hareket ettirebilmiştir [34].

Jayaweera ve Webb (2007), uçak parçalarının montajını istenen tolerans sınırları içerisinde ve doğrulukta yapabilecek, lazer yansıtıcıyla algılanan aynı ölçüdeki deliklerin konumlarını denk gelecek şekilde ayarlamak suretiyle montaj işlemini tamamlayan bir sistem geliştirmişlerdir [35].

Golnabi ve Asadpour (2007), görebilen sistemlerden beklenenleri, bu sistemlerin yeteneklerini, kullanılan bileşen ve alt bileşenleri, ayrıca çalışma yöntemlerini detaylı olarak sundukları çalışmalarında endüstride kullanılabilecek genel bir görme sistemi yapısını, bu sistemin kullanılabileceği alanları, ayrıca ölçme ve inceleme işlemlerinde

kullanılmak üzere 3 boyutlu görüntü işleme yapabilen bir sistemin çalışmasını ortaya koymuşlardır [36].

Eberhardt ve diğerleri (2008), medikal tüplerin sınıflandırılması amacıyla geliştirdikleri MILL (Machine In the Loop Learning) adını verdikleri bir altyapıyla çalışan görebilen robot sistemini farklı ortamlarda denemişler ve robotun dünya çapında 150 farklı konumda 3 ay boyunca günlük 5000 civarı işleminde %0,06 lık bir hata oranıyla çalıştığını ortaya koymuşlardır [37].

Nickels ve diğerleri (2010), NASA'nın uzay programında kullanılacak bir robot için yapay sinir ağları kullanarak geliştirdikleri görme güdümlü konum kontrolü tasarımında, 3 boyutlu görme sistemi ile arasında 10cm olan bir cismin uzaklığını 0,06mm hatayla ölçebilmişlerdir [38].

Taner (1997), SOM'daki nöronların kazanma oranlarına bakıp, baskın çıkmaya başlayan bir nöron varsa diğer nöronların da öğrenme işleminde bir şanslarının olmasını sağlayan, böylece bir SOM ağındaki bütün hücreleri kullanacak ve sınıf başına düşen eleman sayısı birbirlerine yakın olacak şekilde sınıflandırma yapabilen "Conscience" isimli bir algoritma geliştirmiştir [39].

Lindsey ve Strömberg (2000), boyutlandırmadan etkilenmeyen p-gramları kullanarak resimlerden elde edilen 512 elemanlı karakter vektörlerini 8x8 hücreli SOM'a giriş olarak vererek başarılı şekilde karakter tanıma yapabilen bir sistem geliştirmişlerdir [40].

Kurdthongmee (2008), aynı renkte ahşap levhaların sınıflandırmak amacıyla, ahşap fotoğraflarını çekip histogramlarını SOM ağına vererek %95 başarıyla çalışan bir sistem geliştirmiştir [41].

Ghaseminezhad ve Karami (2011), SOM algoritmasının ayrık verileri sınıflandırmada sahip olduğu zayıflıkları gidermek için birinci adımı "ikinci kazanan", ikinci adımı "sıralı öğrenme" ve üçüncü adımı etkin olmayan nöronların silindiği bir algoritma geliştirmişler ve klasik SOM algoritmasının sınıflandırmada başarısız olduğu hem gerçek dünya, hem de sentetik verileri sınıflandırmada başarılı olmuşlardır [42].

1.4 Sistem Bileşenleri

Otomasyon sistemlerinin tamamı, bir makine-insan arabirimi, sayısal kontrol ünitesi, giriş-çıkış ünitesi ve kontrol edilecek donanım bileşenlerini içerir.

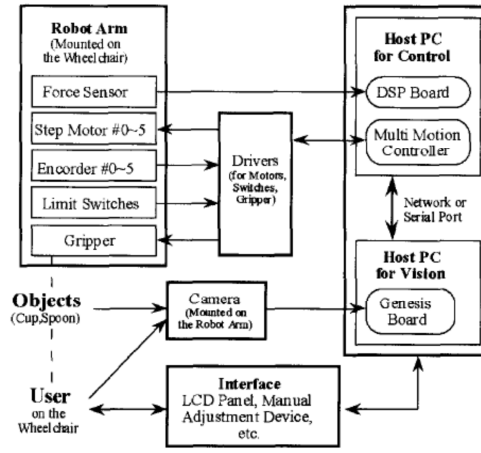
Erzincanlı ve Sharp (1997), şöyle demektedirler [22]:

“Bir otomasyon sisteminin maliyetini düşürmede farklı ürünlerin verimli bir şekilde sınıflandırılması önem taşımaktadır.”

Bunlardan makine-insan arabirimi makineden yapılması istenen komutların verilebildiği, makinedeki değişikliklerin takip edilebildiği birimdir. Otomasyon sistemini kullanacak operatörün kolayca öğrenebileceği, hatasız çalışabileceği bir yazılım şeklinde tasarlanır. CNC’lerde kullanılan paneller buna örnek olarak verilebilir; sayısal kontrol ünitesi ve giriş-çıkış ünitesi ile bağlantılıdır.

Sayısal kontrol ünitesi, makine-insan arabiriminden gelen komutların, donanımı çalıştıracak sinyallere çevirmekten sorumludur. Kontrol algoritmaları bu kısımda bulunur. Boya robotlarının yörüngelerinden sapmadan ilerlemesi, paketleme robotlarının hızlı ve yumuşak bir şekilde ürünü banttan alıp, pakete yerleştirip, tekrar geri dönmesi, kaynak robotlarının dikişi tam olarak doğru ayarlarda ve konum hatası olmadan yapabilmesi bu birimin işlevlerindedir.

Giriş-çıkış ünitesi otomasyon sisteminde yapılması gereken her seferinde aynı olan işlemlerin yapılmasından sorumludur. Bir sınıflandırma robotunda tutucu ucun açılıp kapanması, bir CNC tezgahında takım değiştirilmesi gibi işlemler bu birim aracılığıyla yapılır.



Şekil 1.9 KARES'in blok diyagramı [24]

Şekil 1.9'da görülen şemada “Robot Arm” kısmında nesnelere tutulup, bir yerden diğer bir yere taşıyan robotun bileşenleri gösterilmiştir. Bunlardan “Force sensor” nesnenin tutulup tutulmadığını algılamak, “Step motor #0-5” robotun eklemlerini hareket ettirmek, “Encoder #0~5” eklemlerin istenen konuma gidip gitmediğini geri bildirmek, “Limit switches” robotun belirlenen sınırlar dışında hareketini engellemek ve son olarak da “Gripper” nesnelere kavramak için kullanılmaktadır.

Robot koluna monte edilmiş olan bir kamerayı gösteren “Camera” kısmında görüntü alındıktan sonra “Host PC for vision” kısmına gönderilip görüntü işleme işlemleri yapılmaktadır.

Golnabi ve Asadpour (2007), şöyle demektedirler [36]:

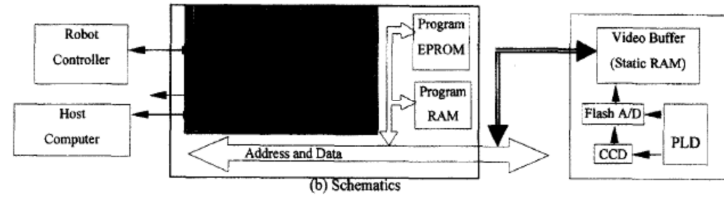
“Görebilen sistemler; karmaşık ölçme ve inceleme işlemleri için akıllı sensör sistemleri olarak değerlendirilebilir.”

Burada istenen nesnelere elde edildikten sonra kontrolden sorumlu “Host PC for control” kısmına gönderilir. Bu kısım bir bilgisayardır. “Multi Motion Controller” kısmında robot tutucu ucunu algılayan nesneye yönlendirmek için gerekli konum bilgileri üretilir. Bu bilgiler “Drivers” kısmına gönderilip robot eklemlerini hareket ettirmek için gerekli sinyallere dönüştürülür. Son olarak “Force sensor” ile nesnenin tutulup tutulmadığı anlaşılır; gerekirse düzeltme yapılır.

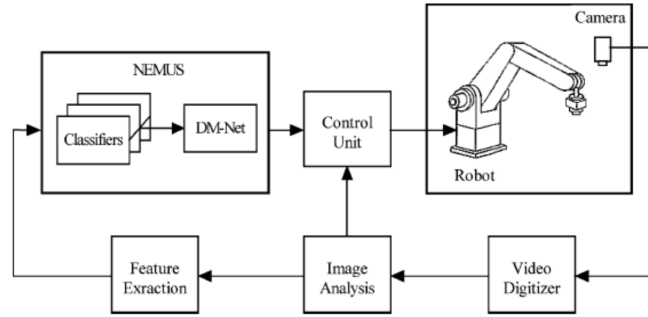
Eberhardt ve diğerleri (2008), şöyle demektedirler [37]:

“Sensörlerin ve görüntü işleme donanımlarının sürekli artan güçleri daha karmaşık sinyal işleme ve tanıma işlemlerini mümkün kılmaktadır. Ancak bu gelişme tasarım ve uygulamada belirlenmesi gereken yeni parametreleri ortaya çıkarmıştır. Ekipman, sensör ve algoritma bolluğundan seçim yapılırken sistemin makul ve iyi çalışması aynı zamanda fiyat ve güç gereksinimlerinin karşılanmasına dikkat edilmelidir.”

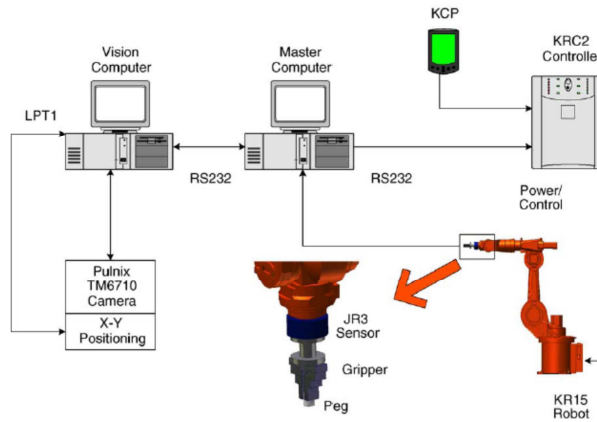
“Interface” kısmı ise kullanıcının özel ayarlar yapabileceği bölümdür.



Şekil 1.10 Dinamik hedef takibi yapabilen bir robot bileşenleri [23]

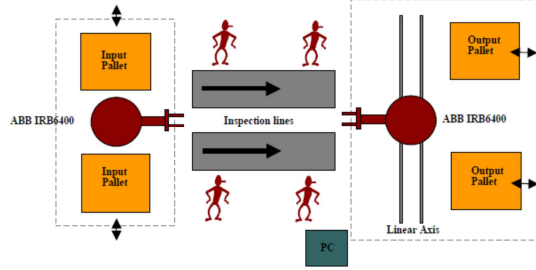


Şekil 1.11 Sınıflandırıcı robot bileşenleri [27]



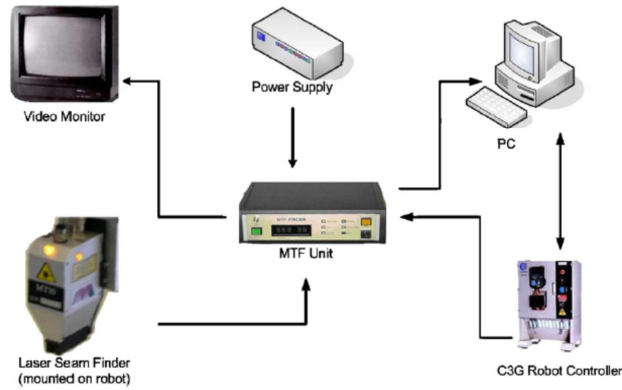
Şekil 1.12 Görerek montaj yapabilen bir sistemin bileşenleri [29]

Şekil 1.12’de görevi görüntü bilgisine göre sac parçaları monte etmek olan bir robotun şematığı verilmiştir. Şekil 1.9’da verilen sistemle şematik olarak benzer olsa da burada kullanılan robot 6 eksenlidir.



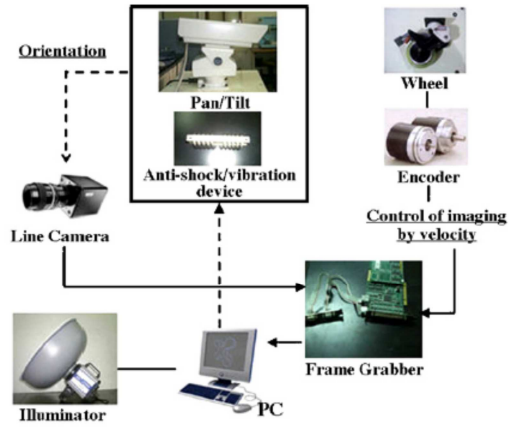
Şekil 1.13 İnceleme birimi [30]

Şekil 1.13’te sembolik insanların görüldüğü konumlara kameradan aldığı görüntüyle inceleme yapabilen bir sistem yerleştirilmiş, inceleme işlemlerinin tamamının bu sistemin yapması sağlanmıştır.



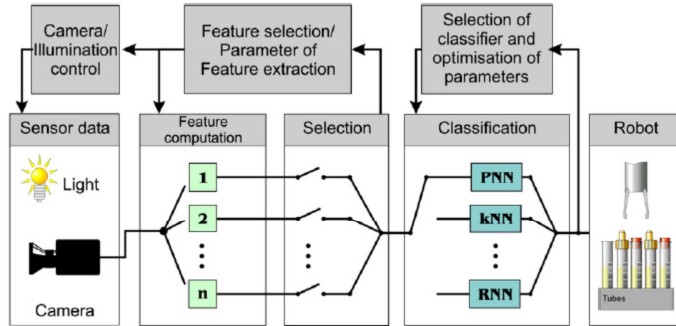
Şekil 1.14 Lazerle hizalama kontrolü yapan bir inceleme sistemi [35]

Kamera yerine lazer yansıtıcı kullanarak, uçak parçalarındaki eşit büyüklükteki delikleri algılayıp, bu bilgilere göre fikstürsüz montaj yapabilen bir sistem Şekil 1.14’te gösterilmiştir.



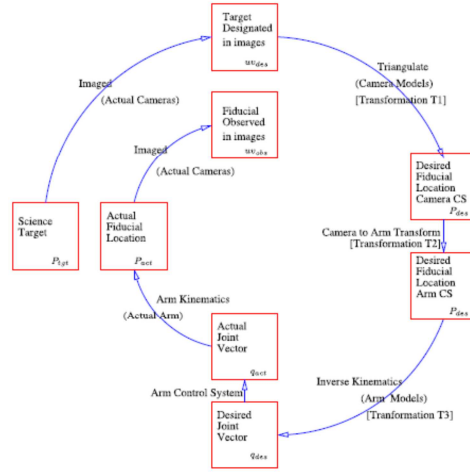
Şekil 1.15 Duvardaki çatlakların tespiti için geliştirilen bir robotun bileşenleri [33]

Görüntüleri matris yerine çizgi olarak algılayan bir kameranın kullanıldığı, duvardaki çatlakları, hareket halindeyken, otomatik olarak tespit edebilen bir sistem Şekil 1.15’de görülmektedir.



Şekil 1.16 Sınıflandırma yapan bir sistemin bileşenleri [37]

Tüp sınıflandırmasında kullanılan bir sistem Şekil 1.16’da görülmektedir.



Şekil 1.17 Uzay çalışmaları için geliştirilen, görebilen bir robotun blok diyagramı [38]

Şekil 1.17’de uzay çalışmaları için geliştirilen, görebilen bir robotun blok diyagramı görülmektedir.

1.4.1 Kamera

Görebilen sistemlerinde kullanılan kameralar 360K pikselden (752x480) başlayıp 5MP’e kadar geniş bir çözünürlük aralığında temin edilebilmektedir. Görüntü alıcı sisteme bağlantısı USB veya yüksek hızlı ağ bağlantısıyla olabilir. Bu kameralar renkli veya gri seviyede görüntü alabilir.

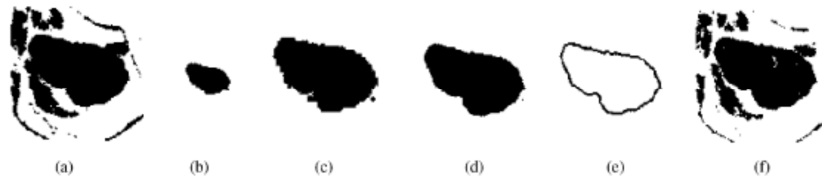
1.4.2 Görüntü Alıcı

Görüntü alıcı, görüntü işleme tekniklerini de içeren gömülü sistem şeklinde elektronik bir bileşen şeklinde temin edilebilmektedir.

Bu bileşenin görevi kameradan analog sinyal şeklinde gelen görüntüyü dijital fotoğrafa çevirmektir.

1.4.3 Görüntü İşleyici

Alınan görüntü üzerinde kenar belirleme, gürültü yok etme, nesne ayıklama, segmantasyon, filtreleme gibi işlemlerin yapıldığı bileşendir. Görüntü alıcıyla beraber gömülü sistem halinde olabildiği gibi bilgisayarda çalışan bir yazılım şeklinde de olabilir. Bu işlemleri yapabilmek için görüntü üzerinde önileme, özellik çıkarma ve sınıflandırma işlemleri yapılır. Et görüntüleri üzerinde işlem yapan örnek bir sistem Şekil 1.18’de, yüzeyin biçimini algılayabilen bir sistem Şekil 1.20’de ve resimden istenen biçimdeki nesneyi ayırabilen bir sistem Şekil 1.21’de görülmektedir.

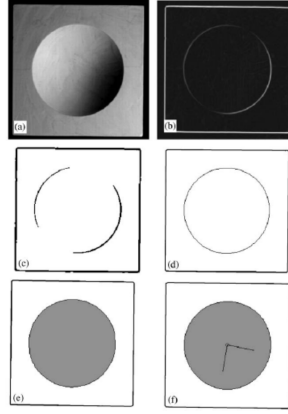


Şekil 1.18 Et görüntüsü üzerinde yapılan işlemler a)2 bitlik görüntü, b)Üzerinde işlem yapılacak bölge, c)Bölge büyütülmüş halde, d)Kenar yumuşatma, e)Kenar belirleme, f)Gürültü yok etme[31]

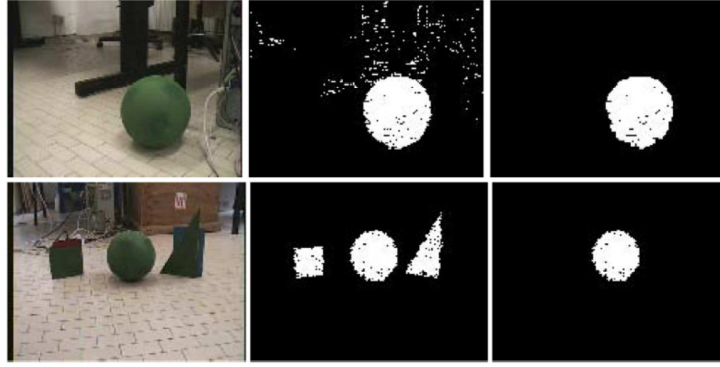


Şekil 1.19 Nesne merkezinin algılanması [29]

Görüntüsü alınan nesnenin kenar uzaklıklarını hesaplamak suretiyle merkez koordinatının hesaplandığı bir programın ekran görüntüsü Şekil 1.19’da görülmektedir.



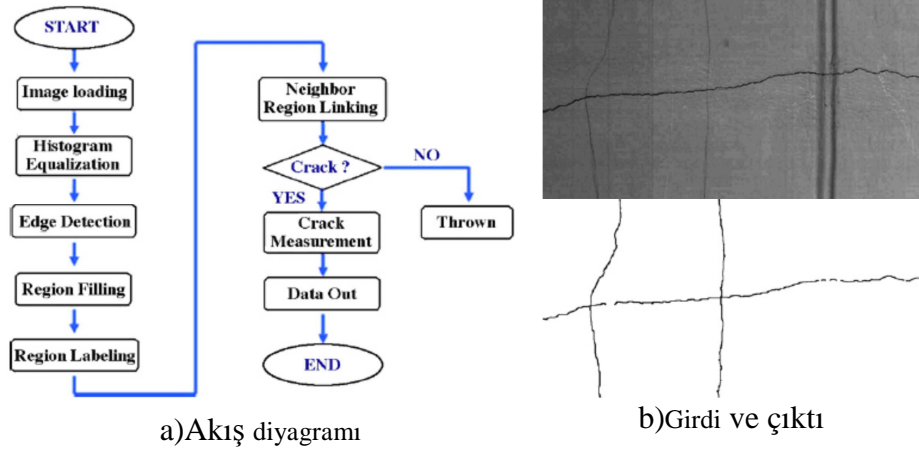
Şekil 1.20 Yüzeyin biçimini fotoğraftan belirleyebilen sistem: a)Çekilen fotoğraf, b)Kenar belirleme, c)Görüntünün tersini alma, d)Daireye tamamlama, e)İçini doldurma, f)Lazer bilgisinden oryantasyonu elde etme [32]



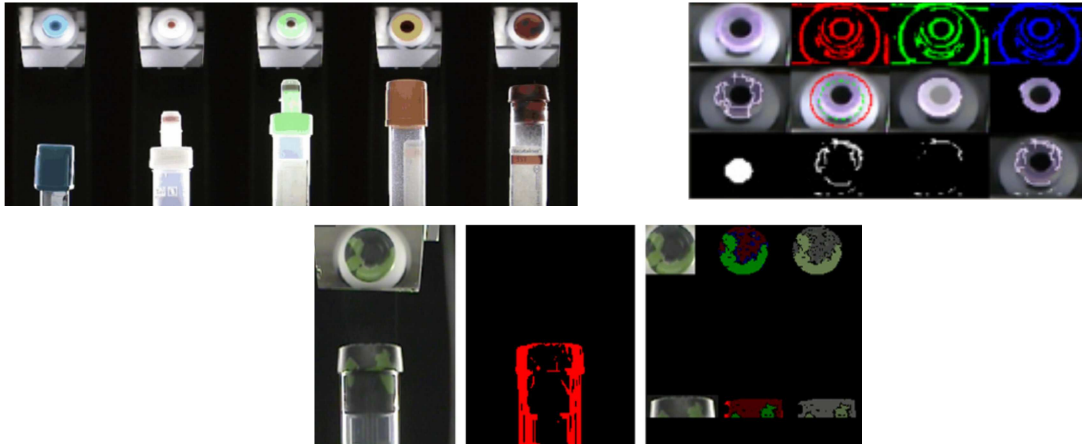
Şekil 1.21 Yalnızca istenen cismin görüntüden ayrılması [34]



Şekil 1.22 Lazer tarama yöntemiyle alınan görüntü [35]



Şekil 1.23 Otomatik çatlak inceleme sisteminin görüntü işleme algoritması [33]



Şekil 1.24 Tüp sınıflandırma sisteminde yapılan görüntü işleme [37]

Duvardaki çatlakları görerek algılayabilen bir sistemin görüntü işleme algoritması Şekil 1.23'te görülmektedir.

5 farklı tüp çeşidinin farklı özelliklerinin ayırıcı olarak belirlenip, alınan fotoğrafta bunların ayıklanarak tüpü uygun yere yerleştirip, kapağını açabilen bir sistemde yapılan görüntü işleme işlemleri Şekil 1.24'te görülmektedir.

1.4.4 Sınıflandırma

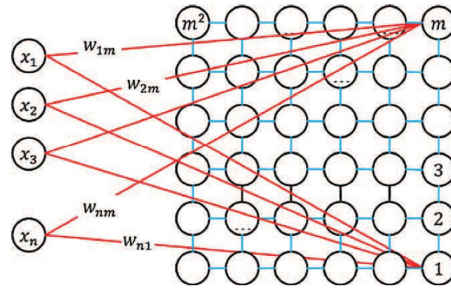
Teze konu olan robotun ayırma işlevini yerine getirebilmesi için görüntüsü alınan nesnelere sınıflandırabilmesi gerekmektedir.

Ghaseminezhad ve Karami (2011), şöyle demektedirler [42]:

“İnsan çevresini, görme, duyma, tatma vs. duyu organları aracılığıyla algılar. Algılamada bilinçli olarak ya da bilinç altında konuları sınıflandırır ve bir bütün olarak çevresindeki nesnelere ne olduklarını bilir. Dahası nesnenin bütün özelliklerini aynı anda analiz edebilir. Bu yüzden, insan bir nesne gördüğünde, bilinçaltında nesnenin yalnızca birkaç özelliğine göre karar vermez. Başka bir deyişle, insan nesnelere eğitimsiz olarak sınıflandırır. Ancak, soru şu olmaktadır: hangi sistem bu şekilde davranış gösterebilir? Bu eski bir sorudur ve cevabı Yapay Sinir Ağları (YSA) olabilir. Kohonen tarafından ortaya koyulan Self-Organizing Map (SOM) (Özdüzenleyici Harita) ağı, sınıflandırmada makul sonuçlar verebilen bir YSA'dır.”

Kohonen (1990), şöyle demektedir [43]:

“Sinir sistemlerini modellemek için kullanılan ağ yapıları ve sinyal işlemcileri, her biri farklı temellere sahip olmak üzere, kabaca 3 sınıfa ayrılabilir. İleri beslemeli ağlar kümelenmiş giriş sinyallerini kümelenmiş çıkış sinyallerine dönüştürür. İstenen giriş-çıkış dönüşümü genellikle sistem parametrelerinin harici bir eğitici tarafından ayarlanmasıyla belirlenir. Geri beslemeli ağlarda, giriş bilgisi, geri-besleme sisteminin temel aktivite durumunu belirler ve durum değişimlerinden sonra asimptotik son durum hesaplamasının çıktısı olarak tanımlanır. Üçüncü kategoride, bir sinir ağındaki komşu hücreler, komşusal etkileşimlerle kendileri arasında rekabet eder ve değişik sinyal örüntü algılayıcılarına dönüşürler. Bu kategoride öğrenme rekabetçi, eğitimsiz veya özdüzenleyici (self-organizing) adını alır.”



Şekil 1.25 SOM ağıının temel yapısı [42]

Şekil 1.25'te bir SOM ağıının temel yapısı görülmektedir. Buna göre giriş vektöründeki her eleman ağıın tamamına uygulanmaktadır.

Attribute	Item																																
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6	
a ₁	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
a ₂	0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
a ₃	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	3	3	3	3	6	6	6	6	6	6	6	6	6	6	
a ₄	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	1	2	3	4	2	2	2	2	2	2	
a ₅	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6

Şekil 1.26 Karakter tanıma için giriş matrisi [43]

B	C	D	E	*	Q	R	*	Y	Z
A	*	*	*	*	P	*	*	X	*
*	F	*	N	O	*	W	*	*	1
*	G	*	M	*	*	*	*	2	*
H	K	L	*	T	U	*	3	*	*
*	I	*	*	*	*	*	*	4	*
*	J	*	S	*	*	V	*	5	6

Şekil 1.27 Karakter tanıma özdüzenleyici haritası [43]

Şekil 1.26’da karakterlerden alınan özellik vektörleri matris formunda görülmektedir. Bu matris SOM ağına girilerek karakterler sınıflandırılmıştır. Bu sınıflandırmanın sonucu Şekil 1.27’de görülmektedir.

Taner (1997), şöyle demektedir [39]:

“Sonucun kesin belli olduğu durumlarda İleri beslemeli Sinir Ağları eğitici olarak sınıflandırma amacıyla kullanılabilir. Yeterli bilgi olmadığı durumlarda ise sınıflandırma için eğitici bir metot kullanılıp sonuçların tecrübeler göre yorumlanması gerekir.”

Kohonen (1998), şöyle demektedir [44]:

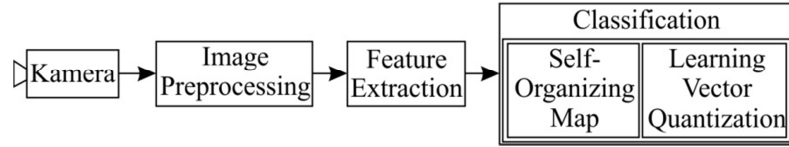
“Özdüzenleyici haritalar (SOM) yüksek boyutlu verilerin görüntülenmesi için yeni ve işlevsel bir yazılım aracıdır. Yüksek boyutlu bir yayılımı düzenli bir düşük boyutlu ızgaraya haritalar. Böylece karmaşık, nonlineer istatistiki ilişkilere sahip yüksek boyutlu verileri düşük boyutlu bir görüntüde basit geometrik ilişkiler olarak dönüştürebilir. Bilgiyi bu şekilde sıkıştırırken ana verideki en önemli yapısal ve ölçüsel ilişkileri korur; aynı zamanda bir tür soyutlaştırma gerçekleştirdiği düşünülebilir. Görüntüleme ve soyutlaştırma olarak bu iki bakış açısı, yöntem analizi, makine algılaması, denetim ve iletişim gibi farklı karmaşık işlemlerde kullanılabilir.”

Linsdey (2000), şöyle demektedir [40]:

“Görüntüdeki nesnelerin sınıflandırılması için bir resmin tamamıyla yapay sinir ağı gibi bir sınıflandırıcıya gönderilmesi nadiren uygun olmaktadır. Bunun yerine Fourier dönüşümü veya moment analizi gibi yöntemlerle temel görüntü özellikleriyle oluşturulan çok daha küçük bir yapı, sınıflandırıcı için giriş olmaktadır. Görüntü yapısı ne kadar küçük olursa sınıflandırma işlemi o kadar hızlı çalışır. Sınıflandırıcı olarak yapay sinir ağı kullanılması durumunda ise sınıflandırma eğitimi çok daha hızlı bir biçimde istenen başarı seviyesine ulaşabilmektedir. Diğer yandan, görüntü yapısının kaba olarak yapılandırılması halinde görüntüleri sınıflandırmak için gerekli olan bilgi kaybolabilmektedir.”

Kohonen ve diğerleri (1996), şöyle demektedirler [45]:

“Özdüzenleyici harita terimi hata-teorik düşüncelerle tanımlanmış bir sınıfı belirtir. Pratikte basit SOM algoritmalarıyla hesaplanan eğitimci, rekabetçi öğrenmede kullanılırlar.”



Şekil 1.28 Doku sınıflandırma blok şeması [45]

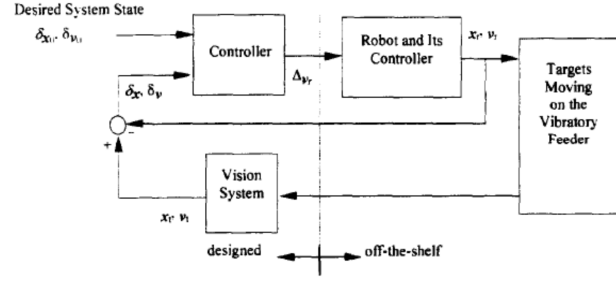
Şekil 1.28’de doku sınıflandırma için kullanılacak bir sistemin blok şeması görülmektedir. Buna göre SOM’un devreye girdiği kısım, görüntü işlenip, özellik çıkartıldıktan sonra olmaktadır.

1.4.5 Denetleyici

Görüntü işleyicide çıkartılan özelliklere göre yapılması gereken işin detaylarının belirlendiği kısımdır. Karar vericiden aldığı bilgilere göre tetiklenmesi gereken donanımların hareketinin belirlenip elektronik sinyaller halinde bu donanımlara gönderir. Yazılım ve donanım bileşenlerini bir arada bulundurur.

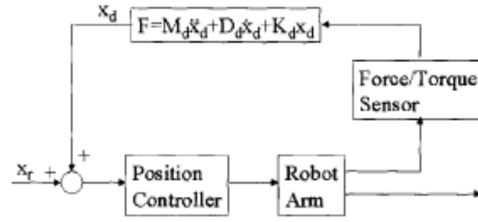
Şekil 1.29 - Şekil 1.32 arasında görerek işlem yapabilen bazı sistemlerin karar verici ve denetim sistemleri görülmektedir.

		Position_Difference						
		NL	NM	NS	ZE	PS	PM	PL
Speed_Difference	NL	NL	NL	NL	NL	NL	NM	ZE
	NM	NL	NL	NL	NL	NL	ZE	PM
	NS	NL	NL	NL	NS	ZE	PM	PL
	ZE	NL	NL	NM	ZE	PM	PL	PL
	PS	NL	NM	ZE	PS	PL	PL	PL
	PM	NM	ZE	PL	PL	PL	PL	PL
	PL	ZE	PM	PL	PL	PL	PL	PL

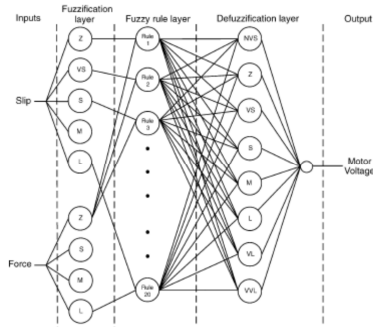


Şekil 1.29 Hareketli hedefi takip eden sistemin denetimi [23]

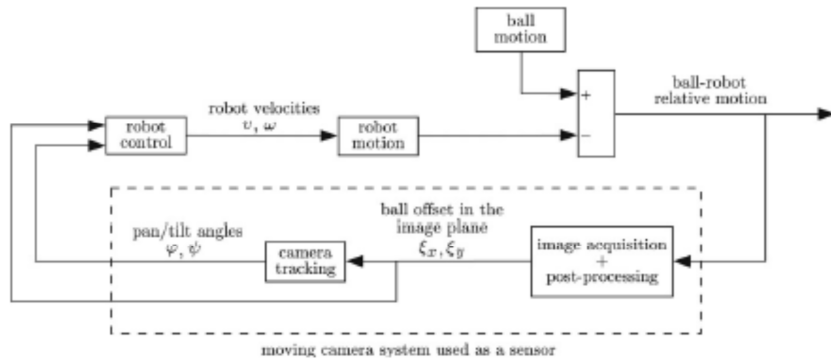
$$\Gamma = \begin{cases} \Gamma_0 & \text{if } |f_y| < f_{y0} \text{ and } f_z > f_{z0} \\ \Gamma_1 & \text{if } f_y > f_{y0} \text{ and } f_z < f_{z0} \\ \Gamma_2 & \text{if } f_y < -f_{y0} \text{ and } f_z < f_{z0} \\ \Gamma_3 & \text{if } f_y > f_{y0} \text{ and } f_z > f_{z0} \\ \Gamma_4 & \text{if } f_y < -f_{y0} \text{ and } f_z > f_{z0} \\ \Gamma_5 & \text{if } |f_y| < f_{y0} \text{ and } f_z < f_{z0} \end{cases}$$



Şekil 1.30 KARES sisteminin denetimi [24]



Şekil 1.31 Yapay Sinir Ağı ile denetim [26]



Şekil 1.32 Hareketli nesne yakalayabilen bir robot denetim şeması [34]

2 MATERYAL ve METOT

Bu çalışmada, robotun tasarımındaki tek kısıtlayıcı veri motor torkudur. Tasarım motor torkunu aşmayacak şekilde maksimum çalışma hacmini veren, tekilliklerden uzak eklem uzayında kalacak ve minimum maliyeti gerektirecek şekilde yapılmıştır.

İlk olarak robotun analitik modeli çıkarılıp MATLAB ortamına aktarılarak sayısal bir model hazırlanmıştır. Bu model robotun ileri-ters kinematik, jakobiyen matrisini içeren bir modeldir. Bu model kullanılarak tahrik torku optimizasyonu yapılmış ve böylece robotun eleman boyutları tespit edilmiştir. Sonrasında bilgisayar ortamında robotun 3 boyutlu katı modeli hazırlanmıştır. Katı model ANSYS yazılımına aktarılmış, robotun ulaşacağı uç konumlara, taşınması beklenen yük ile yüklü olduğu halde hareket etmesi halinde, tahrik mafsallarında oluşacak torklar ve parçalarda oluşan gerilmeler elde edilerek mukavemet analizleri yapılmıştır. Sonra denetim ve sınıflandırma yazılımları hazırlanmış, bu yazılımlarla bilgisayar ortamında robotun gerçekleştirilmesi yapılmıştır. Daha sonra deneysel çalışmaları tamamlamak için imalâta geçilmiştir. İmalât sürecinde katı modeller yardımıyla CAM kodları çıkarılmış ve parçalar 3 eksenli CNC frezede imâl edilmiştir. Kullanılacak standart parçalar temin edildikten sonra robotun montajı tamamlanıp deneysel çalışmalara başlanmıştır. Yapılan sayısal analizlerle deneysel çalışma sonuçları karşılaştırılmış, sonuçlar yorumlanmıştır.

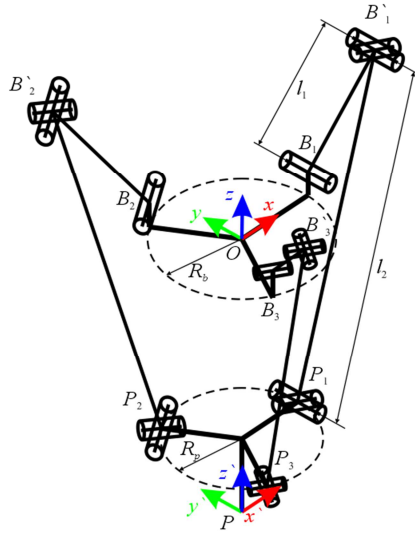
Robotun görme ve denetim sistemleri MATLAB ortamında hazırlanıp gerçek zamanlı olarak yapılmasına çalışılmıştır. Görme sistemi bir kameradan ve MATLAB ortamında çalışan bir programdan oluşmaktadır. Programın kameradan aldığı görüntülerdeki objelerin geometrik merkezlerini ve oryantasyon açılarını algılayabilmesi, ayrıca sınıflandırabilmesi sağlanmıştır. Konumlar elde edildikten sonra kullanılan motorların bu konumlara ulaşabilecek şekilde robotu tahrik etmeleri sağlanmıştır.

2.1 Kinematik Tasarım

Kinematik analiz tahrik edilen elemanların açılarına göre hareketli platformun konumunu veya hareketli platformun belirli bir konumda olmasını sağlayacak açıların belirlenmesi için yapılan analizdir. Delta robotta tahrik edilen elemanlar döner mafsallarla gövdeye bağlandıkları için bunların konumlarını belirleyen parametre sabit elemana göre olan açılarıdır.

Tahrik edilen elemanların açılarına göre hareketli platformun konumunun hesaplandığı analize “İleri Kinematik”; diğer duruma da “Ters Kinematik” analizi denir.

Kinematik tasarım kısmında kullanılan notasyon ve formüller, ileri kinematik kısmı hariç, [3]’den alınmıştır.

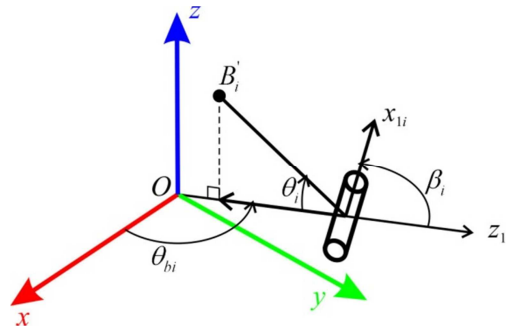


Şekil 2.1 Kinematik çiftler ve simgeler.

Şekil 2.1’de kinematik çiftler ve kullanılan simgeler verilmiştir. Robotun her 3 bacağı da aynı yapıya sahiptir; döner mafsallı bağlantı elemanı, birbirine dik iki adet döner (kardan) mafsallı bağlantı elemanı, son olarak uca bağlı kardan mafsaldan oluşmuştur. $O-xyz$ eksen takımı sabit olan platformu, $P-x'y'z'$ eksen takımı ise hareketli olan platformu temsil etmektedir.

B_i noktaları sabit eksen takımına göre, tahrik edilen döner mafsalsın geometrik konumunu, P_i noktaları sabit eksen takımına göre hareketli platformun geometrik konumunu, B'_i noktaları ise tahrik edilen elemana bağlı kardan mafsalsın sabit eksen takımına göre geometrik konumu ifade etmektedir.

2.1.1 İleri Kinematik



Şekil 2.2 Tahrik edilen mafsalsın [3]

Şekil 2.2'ye göre:

$$b'_{ix} = b_{ix} - l_{i1} \sin(\theta_{bi} + \beta_i) \cos \theta_{i1}, \quad , i = 1, \dots, 3 \quad [3] \quad (2.1)$$

$$b'_{iy} = b_{iy} + l_{i1} \cos(\theta_{bi} + \beta_i) \cos \theta_{i1}, \quad , i = 1, \dots, 3 \quad [3] \quad (2.2)$$

$$b'_{iz} = b_{iz} + l_{i1} \sin \theta_{i1}, \quad , i = 1, \dots, 3 \quad [3] \quad (2.3)$$

Şekil 2.1'e göre aşağıdaki bağıntı yazılabilir:

$$(b'_{ix} - x_i)^2 + (b'_{iy} - y_i)^2 + (b'_{iz} - z_i)^2 = l_{i2}^2 \quad (2.4)$$

x_i, y_i, z_i değerleri P_i noktasının koordinatlarını ve l_{i2} de i . bacağın ikinci elemanını ifade etmektedir.

Denklemleri basitleştirmek için [11]'de verilen yöntemi kullanarak b'_{ix} noktalarının konumları R_p kadar negatif yönde kaydırılır. Bu şekilde P_i noktalarının P noktasına göre konumu denklemlerden ayrılmış olur.

Tablo 2.1 Tahrik edilen elemanın uç konumları

b''_{ix}	b''_{iy}	b'_{iz}
$R_b - R_p + l_1 \cos \theta_1$	0	$d_b + l_1 \sin \theta_1$
$-\frac{1}{2}(R_b - R_p + l_1 \cos \theta_2)$	$-\frac{\sqrt{3}}{2}(R_b - R_p + l_1 \cos \theta_2)$	$d_b + l_1 \sin \theta_2$
$-\frac{1}{2}(R_b - R_p + l_1 \cos \theta_3)$	$\frac{\sqrt{3}}{2}(R_b - R_p + l_1 \cos \theta_3)$	$d_b + l_1 \sin \theta_3$

$$(x - b''_{1x})^2 + (y - b''_{1y})^2 + (z - b'_{1z})^2 = l_2^2 \quad (2.5)$$

$$(x - b''_{2x})^2 + (y - b''_{2y})^2 + (z - b'_{2z})^2 = l_2^2 \quad (2.6)$$

$$(x - b''_{3x})^2 + (y - b''_{3y})^2 + (z - b'_{3z})^2 = l_2^2 \quad (2.7)$$

$$w_i = b''_{ix}{}^2 + b''_{iy}{}^2 + b'_{iz}{}^2 \quad (2.8)$$

$$(2.9) = (2.5) - (2.6)$$

$$(2.10) = (2.5) - (2.7)$$

$$(2.11) = (2.6) - (2.7)$$

(2.9) ve (2.10) denklemlerinden iki bilinmeyen çözümlenmesiyle (2.12) ve (2.13) ifadeleri elde edilir.

$$x = \frac{j_1 - zh_1}{k} \quad (2.12)$$

$$y = \frac{j_2 - zh_2}{-k} \quad (2.13)$$

(2.12) ve (2.13) ifadelerinde:

$$h_1 = (b_{3y}'' - b_{1y}'')(b_{2z}' - b_{1z}') - (b_{2y}'' - b_{1y}'')(b_{3z}' - b_{1z}') \quad (2.14)$$

$$h_2 = (b_{3x}'' - b_{1x}'')(b_{2z}' - b_{1z}') - (b_{2x}'' - b_{1x}'')(b_{3z}' - b_{1z}') \quad (2.15)$$

$$j_1 = \frac{(w_2 - w_1)(b_{3y}'' - b_{1y}'') - (w_3 - w_1)(b_{2y}'' - b_{1y}'')}{2} \quad (2.16)$$

$$j_2 = \frac{(w_2 - w_1)(b_{3x}'' - b_{1x}'') - (w_3 - w_1)(b_{2x}'' - b_{1x}'')}{2} \quad (2.17)$$

$$k = (b_{3y}'' - b_{1y}'')(b_{2x}'' - b_{1x}'') - (b_{2y}'' - b_{1y}'')(b_{3x}'' - b_{1x}'') \quad (2.18)$$

(2.12) ve (2.13) ifadeleri (2.5) denkleminde yerine yazılırsa yalnızca z 'ye bağlı ikinci dereceden (2.19) denklemi elde edilir. Bu denklemin çözümünden (2.20) denklemi elde edilir. Bu z değeri (2.12) ve (2.13) denklemlerinde yerine konularak sırasıyla x ve y değerleri hesaplanabilir. $y_1 = 0$ olduğuna dikkat edilmelidir.

$$\underbrace{\left(\frac{h_1^2 + h_2^2}{g^2} + 1 \right)}_{A_k} z^2 + \underbrace{\left(\frac{h_1 j_1 + h_2 j_2}{k^2} + \frac{h_1 b_{1x}''}{k} - b_{1z}' \right)}_{B_k} 2z + \underbrace{\left(b_{1x}''^2 + b_{1z}'^2 - l_2^2 + \frac{j_1^2 + j_2^2}{k^2} + \frac{2j_1 b_{1x}''}{k} \right)}_{C_k} = 0 \quad (2.19)$$

$$z = \frac{-B_{kin} + \sqrt{B_{kin}^2 - 4A_{kin} C_{kin}}}{2A_{kin}} \quad (2.20)$$

Hesaplanan z kökleri üç adet kürenin kesiştiği ortak noktaları vermektedir. Hesaplama sonucunda kürelerin alt kesişim noktasına ihtiyaç duyulduğundan küçük reel kök alınmalıdır. Köklerin kompleks çıkması robotun belirtilen eklem açılarında monte edilemeyeceğini göstermektedir.

Sonuçta tutucu ucun konumu (2.21) denklemindeki gibi olur.

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ d_p \end{bmatrix} \quad (2.21)$$

2.1.2 Ters Kinematik

(2.1), (2.2) ve (2.3), (2.4)'de yerine yazılırsa (2.22) denklemini elde edilir.

$$E_i \cos \theta_{i1} + F_i \sin \theta_{i1} = G_i \quad [3] \quad (2.22)$$

(2.22) denklemindeki ifadeler (2.23)~(2.27) denklemlerinde verilmiştir.

$$E_i = (y_i - b_{iy}) \cos(\theta_{bi} + \beta_i) - (x_i - b_{ix}) \sin(\theta_{bi} + \beta_i) \quad (2.23)$$

$$F_i = z_i - b_{iz} \quad (2.24)$$

$$G_i = \frac{(x_i - b_{ix})^2 + (y_i - b_{iy})^2 + (z_i - b_{iz})^2 + l_{i1}^2 - l_{i2}^2}{2l_{i1}} \quad (2.25)$$

$$K_i = 1 \quad (2.26)$$

$$H_i = E_i^2 + F_i^2 - G_i^2 \quad [3] \quad (2.27)$$

$$\sin \theta_{i1} = \frac{F_i G_i + K_i E_i \sqrt{H_i}}{E_i^2 + F_i^2} \quad [3] \quad (2.28)$$

$$\cos \theta_{i1} = \frac{E_i G_i - K_i F_i \sqrt{H_i}}{E_i^2 + F_i^2} \quad [3] \quad (2.29)$$

Sonuç olarak, (2.28) ve (2.29) denklemlerinin tanjant fonksiyonunda yerine yazılmasıyla (2.30) denkleminde θ_{i1} elde edilir:

$$\theta_{i1} = \arctan[\sin \theta_{i1}, \cos \theta_{i1}] \quad (2.30)$$

2.1.3 Jakobiyen

Jakobiyen, hareketli platformun hareketiyle eklem hareketlerinin uyumlu olması için gerekli diferensiyel ilişkiyi tanımlar. Sabit bir matris değildir; her konum için yeniden hesaplanması gerekir.

Jakobiyen matris konum ifadelerinin bağlı oldukları değişkenlere göre türetilmesiyle elde edilir. (2.1), (2.2) ve (2.3) ifadeleri bir matrisde birleştirilip yazılırsa (2.31) denklemi elde edilir. (2.32) ve (2.33) denklemlerindeki ifadeler, (2.31) denklemindeki ifadeleri tanımlar.

$$\mathbf{b}'_i = \mathbf{b}_i + l_{i1} \mathbf{Q}_{i1} \mathbf{d}_i \quad [3] \quad (2.31)$$

$$\mathbf{Q}_{i1} = \begin{bmatrix} \cos(\theta_{bi} + \beta_i) & -\sin(\theta_{bi} + \beta_i) & 0 \\ \sin(\theta_{bi} + \beta_i) & \cos(\theta_{bi} + \beta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad [3] \quad (2.32)$$

$$\mathbf{d}_{i1} = \begin{bmatrix} 0 \\ \cos \theta_{i1} \\ \sin \theta_{i1} \end{bmatrix} \quad [3] \quad (2.33)$$

Ayrıca,

$$l_{i2}^2 = (\mathbf{p}_i - \mathbf{b}_i')^T (\mathbf{p}_i - \mathbf{b}_i') \quad [3] \quad (2.34)$$

(2.35) denklemi, (2.32), (2.33) ve (2.34) ifadelerinin çarpımının zamana göre türetilmesiyle elde edilir.

$$\dot{\mathbf{b}}_i' = l_{i1} \mathbf{Q}_{i1} \dot{\mathbf{d}}_i \quad [3] \quad (2.35)$$

$$\dot{\mathbf{d}}_i = \begin{bmatrix} 0 \\ -\sin \theta_{i1} \\ \cos \theta_{i1} \end{bmatrix} \dot{\theta}_{i1} \quad [3] \quad (2.36)$$

$$(\mathbf{p}_i - \mathbf{b}_i')^T \dot{\mathbf{b}}_i' - (\mathbf{p}_i - \mathbf{b}_i')^T \dot{\mathbf{p}}_i = 0 \quad [3] \quad (2.37)$$

(2.37) ifadesindeki $\dot{\mathbf{p}}_i$ (2.38) denkleminde verilmiştir.

$$\dot{\mathbf{p}}_i = \dot{\mathbf{p}} + \dot{\mathbf{Q}} \mathbf{r}_i' \quad [3] \quad (2.38)$$

(2.39), (2.40), (2.41) ve (2.42)'de verilen tanımlamaları kullanarak (2.37) yeniden yazılırsa (2.43) denklemi elde edilir.

$$\mathbf{e}_i = l_{i1} \mathbf{Q}_{i1} \begin{bmatrix} 0 \\ -\sin \theta_{i1} \\ \cos \theta_{i1} \end{bmatrix} \quad [3] \quad (2.39)$$

\mathbf{Q} : $P(x', y', z')$ koordinat sisteminin, $O(x, y, z)$ eksen takımına göre dönme matrisi

\mathbf{r}_i' : P_i noktasının $P(x', y', z')$ eksen takımına göre koordinatları

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.40)$$

$$\mathbf{r}_i' = \begin{bmatrix} x_i' \\ y_i' \\ z_i' \end{bmatrix} [3] \quad (2.41)$$

$$\dot{\mathbf{b}}_i' = \mathbf{e}_i \dot{\theta}_{i1} [3] \quad (2.42)$$

$$(\mathbf{p}_i - \mathbf{b}_i')^T \mathbf{e}_i \dot{\theta}_{i1} = (\mathbf{p}_i - \mathbf{b}_i')^T \dot{\mathbf{p}} + [(\mathbf{Q}\mathbf{r}_i') \times (\mathbf{p}_i - \mathbf{b}_i')]^T \boldsymbol{\omega} [3] \quad (2.43)$$

$$\mathbf{A}\mathbf{t} = \mathbf{B}\dot{\boldsymbol{\theta}} [3] \quad (2.44)$$

Olduğu göz önüne alınıp $\dot{\boldsymbol{\theta}}$ ve \mathbf{t} aşağıdaki şekilde tanımlanır.

$$\dot{\boldsymbol{\theta}} = [\dot{\theta}_1 \dots \dot{\theta}_n]^T \quad n=3 [3] \quad (2.45)$$

$$\mathbf{t} = [\omega_1 \quad \omega_2 \quad \omega_3 \quad \dot{x} \quad \dot{y} \quad \dot{z}]^T [3] \quad (2.46)$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} [3] \quad (2.47)$$

$$\mathbf{B} = \text{diag} \left[(\mathbf{p}_1 - \mathbf{b}_1')^T \mathbf{e}_1, \dots, (\mathbf{p}_n - \mathbf{b}_n')^T \mathbf{e}_n \right] [3] \quad (2.48)$$

$$\mathbf{a}_i = \begin{bmatrix} (\mathbf{Q}\mathbf{r}_i') \times (\mathbf{p}_i - \mathbf{b}_i') \\ (\mathbf{p}_i - \mathbf{b}_i') \end{bmatrix} \quad i=1\dots3 [3] \quad (2.49)$$

(2.49) ve (2.47)'de yerine yazılırsa (2.50) denklemi elde edilir.

$$A_{3 \times 6} = \begin{bmatrix} z_i' \{b_{iy} - y_i + l_{i1} \cos(\theta_{bi} + \beta_i) \cos \theta_{i1}\} \\ x_i' (b_{iz} - z_i + l_{i1} \sin \theta_{i1}) + z_i' \{x_i - b_{ix} + l_{i1} \sin(\theta_{bi} + \beta_i) \cos \theta_{i1}\} \\ -x_i' \{b_{iy} - y_i + l_{i1} \cos(\theta_{bi} + \beta_i) \cos \theta_{i1}\} \\ x_i - b_{ix} + l_{i1} \sin(\theta_{bi} + \beta_i) \cos \theta_{i1} \\ y_i - b_{iy} - l_{i1} \cos(\theta_{bi} + \beta_i) \cos \theta_{i1} \\ z_i - b_{iz} - l_{i1} \sin \theta_{i1} \end{bmatrix}^T, i = 1, 2, 3 \quad (2.50)$$

(2.48) denklemindeki ifadeler açılarak yerlerine yazılırsa (2.51) denklemi elde edilir.

$$\begin{aligned} B_{i,i} &= l_{i1} \sin(\theta_{bi} + \beta_i) \sin \theta_{i1} (x_i - b_{ix} + l_{i1} \sin(\theta_{bi} + \beta_i) \cos \theta_{i1}) \\ &\quad - l_{i1} \cos \theta_{i1} (b_{iz} - z_i + l_{i1} \sin \theta_{i1}) \\ &\quad + l_{i1} \cos(\theta_{bi} + \beta_i) \sin \theta_{i1} (b_{iy} - y_i + l_{i1} \cos(\theta_{bi} + \beta_i) \cos \theta_{i1}) \end{aligned} \quad (2.51)$$

(2.44)'ü kullanarak Jakobiyen matris (2.52)'deki gibi elde edilir.

$$J = B^{-1}A \quad (2.52)$$

2.1.4 Kinetostatik

Jakobiyen matrisin bir uygulaması da robotun statik analizidir. Bu analizde hareketli platforma uygulanan kuvvet ve torklara tepki olarak tahrik edilen elemanlarda oluşan kuvvet veya torklar elde edilir.

$$J' = A^{-1}B \quad (2.53)$$

$$\mathbf{n}^T : \text{Bileğe uygulanan tork} \quad (2.54)$$

$$\mathbf{f}^T : \text{Bileğe uygulanan kuvvet} \quad (2.55)$$

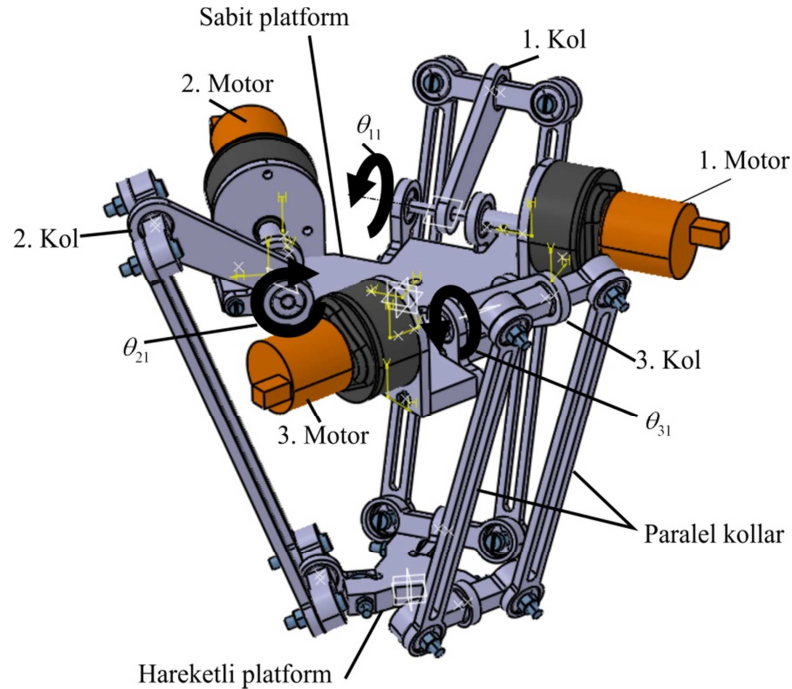
$$\tau : \text{Mafsal torqları} \quad (2.56)$$

$$w = \begin{bmatrix} \mathbf{n}^T & \mathbf{f}^T \end{bmatrix}^T \quad [3] \quad (2.57)$$

$$\tau = J^T w \quad (2.58)$$

2.2 Mekanik Tasarım

Mekanik tasarımda mafsal torqları ve mafsallara gelen kuvvetlerin belirlenmesi, ayrıca mukavemet analizleri yapılmıştır. Bu işlemler ANSYS ortamında sayısal olarak yapıldığından öncelikle eleman boyutlarının belirlenmesi gerekmektedir. Eleman boyutları motor torkunu aşmayacak şekilde belirlenmiş, CAD modelleri oluşturulmuş ve mukavemet analizleri tamamlanarak robota son şekli verilmiştir. Şekil 2.3'te robotun elemanları görülmektedir.



Şekil 2.3 Robotun elemanları ve tahrik mafsallarının pozitif yönleri

Robot kollarının hareket sınırı $-\pi/4 : \pi/6$ aralığında belirlenmiştir. Motor kodlayıcısı bir turda 24 darbe sinyali üretmektedir (EK-16). Motor ayrıca 1:50 oranında bir redüktöre sahiptir (EK-16). Buna göre bir turda 1200 darbe sinyali üretilmektedir. Hareketin 1byte'lık değişkenle kontrol edilebilmesi için en fazla 255 darbe kullanılabilir. 250 darbe sinyali 75° 'lik bir harekete izin vermektedir.

$$\theta_{i,j} = \begin{bmatrix} -\pi/4 & -\pi/4 & -\pi/4 \\ -\pi/4 & -\pi/4 & \pi/6 \\ -\pi/4 & \pi/6 & \pi/6 \\ -\pi/4 & \pi/6 & -\pi/4 \\ \pi/6 & -\pi/4 & -\pi/4 \\ \pi/6 & -\pi/4 & \pi/6 \\ \pi/6 & \pi/6 & -\pi/4 \\ \pi/6 & \pi/6 & \pi/6 \end{bmatrix} \quad (2.59)$$

$$(\max) f = \sum_{i=2}^8 \sqrt{(p_{xi} - p_{x(i-1)})^2 + (p_{yi} - p_{y(i-1)})^2 + (p_{zi} - p_{z(i-1)})^2} \quad (2.60)$$

Motor devri $92,6d/dk$ 'dir; 75° 'lik açığı $0,13499s$ 'de katedebilmektedir (EK-16). Buna göre zamana bağlı olarak tahrik edilen mafsalların açıları Tablo 2.2'de verilmiştir. Robotun gerilme analizinde Tablo 2.2'de verilen hareket sınır şartları kullanılmıştır.

Tablo 2.2 Motor konumlarının zamana göre aldıkları değerler

Adım	Zaman (s)	1. Motor ($^\circ$)	2. Motor ($^\circ$)	3. Motor ($^\circ$)
0	0	0	0	0
1	0,13499	45	45	45
2	0,26998	45	45	-30
3	0,40497	45	-30	-30
4	0,53996	45	-30	45
5	0,67495	-30	45	45
6	0,80993	-30	45	-30
7	0,94492	-30	-30	45
8	1,0799	-30	-30	-30

Robotun kendi ağırlığına ek olarak $0,5kg$ yük taşıyabilmesi amaçlanmıştır. Bu da yük sınır şartı olarak uygulanmıştır.

2.2.1 Eleman Boyutları

Sabit ve hareketli platform en küçük boyutlara, kollar ise en büyük çalışma hacmini sağlamak amacıyla, mümkün olan en büyük boyutlara sahip olacak şekilde tasarlanmıştır. Robotun çalışma hacminde robotun ihtiyaç duyabileceği maksimum torklar hesaplanmış; eleman boyutları motorun torkundan daha fazla tork gerektirmeyecek şekilde boyutlandırılmıştır.

Sabit ve hareketli platformun boyutları CAD ortamında belirlendikten sonra MATLAB'da optimizasyon yapılmıştır.

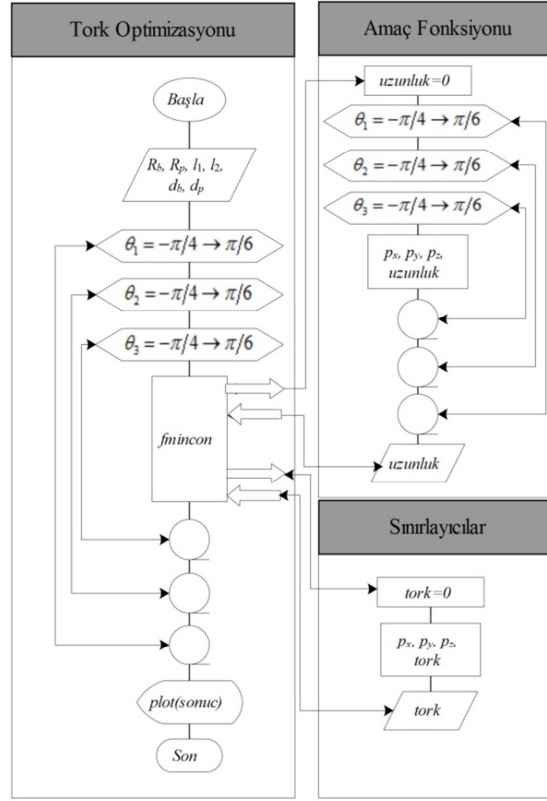
Amaç fonksiyonu robot eklemlerinin sınırlara ulaştığı açı değerlerinde alınan yol olarak belirlenmiştir. Bu yol ne kadar uzun olursa robotun çalışma hacmi de o kadar büyük olacaktır. Bu yolu elde etmek için robotun izlemesi gereken konumlar (2.59) denkleminde. Amaç fonksiyonu ise (2.60) denkleminde verilmiştir.

Sınırlayıcılar, kolların imal edilebilir boyutlardan küçük; eldeki imkanlarla imal edilebilecek boyutlardan büyük olmaması ve motor torkunun aşılmamasıdır. Buna göre en küçük değer, en küçük rulmanların kullanılması halinde elemanın sahip olması gereken minimum boydur. En büyük boyut ise mevcut CNC tezgahta işlenebilecek en büyük boydur. Tork kısıtı ise her bir konumda (2.58) denkleminde hesaplanmıştır.

Optimizasyon, MATLAB Optimization Toolbox kullanılarak yapılmıştır. Bu araç non-linear sınırlayıcıların kullanılmasına imkan sağladığı için tercih edilmiştir.

Optimizasyonu yapabilmek için 3 adet program hazırlanması gerekmiştir. Birincisi amaç fonksiyonunu (EK-2), ikincisi non-linear sınırlayıcıları (EK-3) ve üçüncüsü de optimizasyonu çalıştırmak (EK-1) içindir.

Optimizasyon algoritması Şekil 2.4'te verilmiştir.



Şekil 2.4 Tork optimizasyonu algoritması

2.3 Gerilme Analizi

Gerilme analizinde robotun önce bir kolu esnek elemanlarla modellenip analiz edilmiştir. Bu analizden alınan mafsallara göre sabit platformun statik analizi yapılmıştır. Daha sonra bu analizden alınan sabit mesnet tepkilerine göre de robotu taşıyan iskeletin statik, burkulma ve modal analizleri yapılmıştır. Malzeme özellikleri ise Tablo 2.3'te verilmiştir.

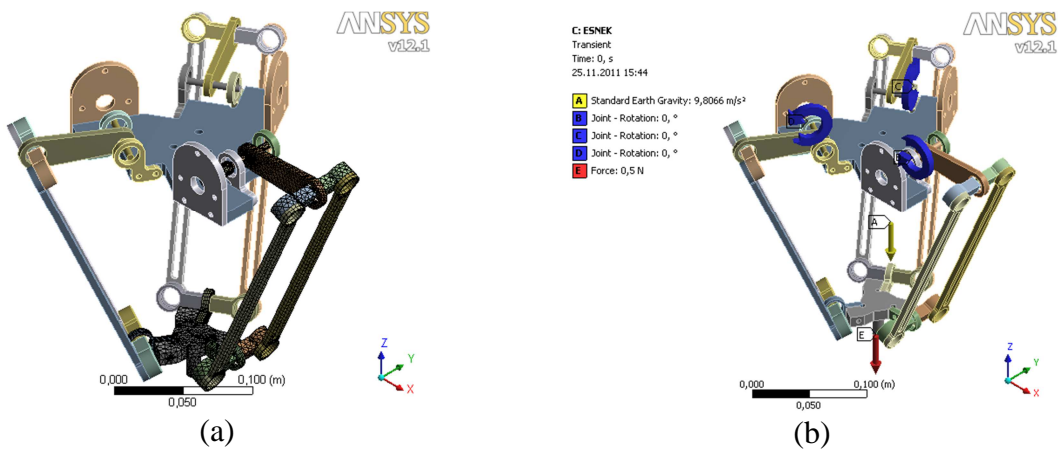
Konum analizi Rigid Body Dynamics modulünde tamamlanmıştır. Bunun sebebi Transient Analysis modulünde konum bilgisinin esnek elemanlar için alınamamasıdır.

Tablo 2.3 Malzeme özellikleri

Malzeme	$E (N/mm^2)$	$\sigma_{ak} (N/mm^2)$	ν	$\rho (kg/m^3)$
Alucobond	70.000	90	0,33	1.220
PMMA	2.243	54	0,33	1.200
Çelik	215.000	475	0,3	7.850

2.3.1 Robot Kolunun Gerilme Analizi

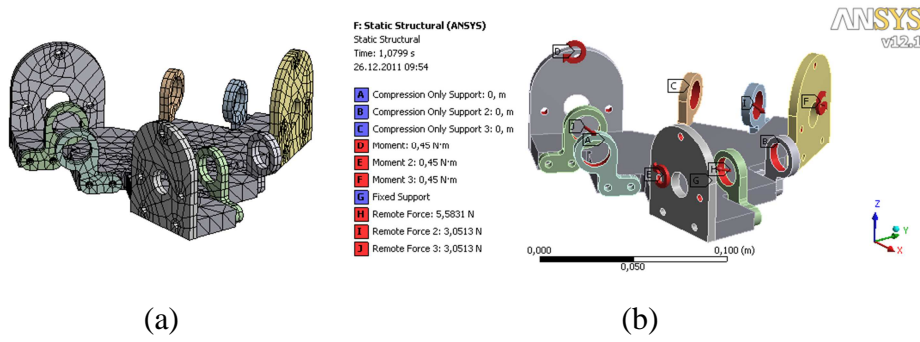
Analizde geometrik non-lineeriteler dikkate alınmış; malzemenin lineer elastik davranış gösterdiği durum için çözüm yapılmıştır. Robot, sabit platformdan “Fixed” mesnetiyle sabitlenmiştir. Tahrik mafsallarına Tablo 2.2’deki hareket sınır şartları uygulanmış, hareketli platforma ise dikey yönde -5N’luk yük etki ettirilip yerçekimi ve atalet etkileri de hesaba katılmıştır. Ağ yapısı ANSYS’teki otomatik ayarlar kullanılarak belirlenmiştir. Buna göre ağın kalitesini belirleyen bütün parametreler “Orta” ayarda olacak şekilde belirlenmiştir. Ağ yapısı ve sınır şartları Şekil 2.5’de, Modelde analize dahil olan düğüm sayısı 70716; eleman sayısı ise 36374’dür.



Şekil 2.5 Esnek elemanların ağ yapısı (a) ve analizin sınır şartları (b)

2.3.2 Sabit Platformun Gerilme Analizi

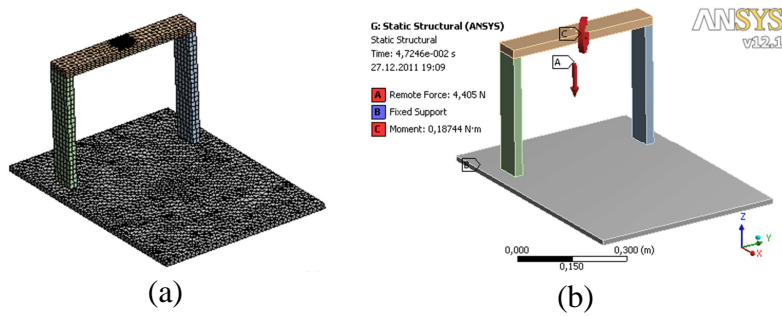
Sabit platform hareket etmeyeceği için Transient Analysis modulünde analize dahil edilmemiştir. Tahrik mafsallarındaki tepki kuvvetleri belirlendikten sonra Static Analysis modulünde bu kuvvetlerin ve motorların uygulayacağı torqların etkisinde analiz edilmiştir. Sabit platform için yapılan analizin ağ yapısı ve sınır şartları Şekil 2.6’da görülmektedir. Ağ örülürken “Orta” kalitedeki ayarlar seçilmiştir. Ağdaki eleman sayısı 6477, düğüm sayısı ise 18335’dir.



Şekil 2.6 Sabit platform için ağ yapısı (a) ve sınır şartları (b)

2.3.3 Taşıyıcı Şasinin Gerilme Analizi

Taşıyıcı şasinin gerilme analizi Static Analysis, burkulma analizi Linear Buckling ve modal analizi de Modal Analysis modulünde gerçekleştirilmiştir. Burkulma ve modal analiz için ağ yapısı kritik olduğundan ağın kalitesini belirleyen parametreler “iyi” ayarda seçilmiştir. Ağdaki eleman sayısı 32045, düğüm sayısı da 70290’dır.



Şekil 2.7 Taşıyıcı şasi için ağ yapısı (a), ve sınır şartları (b)

2.4 Hassasiyet

Motor 250 darbe sinyali kullanıldığında 75°'lik bir harekete izin vermektedir. Buna göre robotun hassasiyeti (2.61) denkleminde verilmiştir.

$$\frac{75^\circ}{250darbe} = 0,3\%_{darbe} \quad (2.61)$$

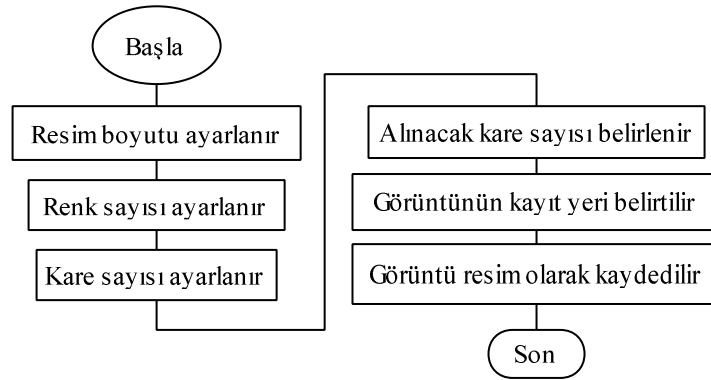
2.5 Görme Sistemi

Görme sisteminde resimler 256 renk gri-seviye olacak şekilde alınmıştır. Bu resimlere sırasıyla kontrast ayarlama, eşik seviyesi belirleme, gürültüleri silme, nesnelere tanımlama ve bu nesnelere konum ve oryantasyonlarını elde etme işlemleri uygulanmıştır.

2.5.1 Görüntü Alıcı

360K çözünürlüğündeki bir kamera bilgisayara bağlanıp, MATLAB ImageAcquisition Toolbox yazılımı kullanılarak istenen sıklıkta görüntü almak için kullanılmıştır.

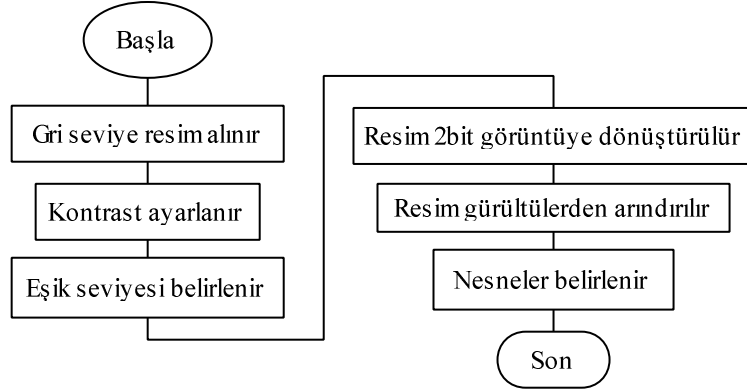
Görüntü almak için kullanılan yazılımın (EK-4) algoritması Şekil 2.8'de verilmiştir.



Şekil 2.8 Görüntü alma algoritması

2.6 Görüntü İşleme Sistemi

Görüntü işleyici olarak MATLAB ImageProcessing Toolbox kullanılmıştır. Kullanılan yazılımın (EK-5) algoritması Şekil 2.9’de verilmiştir.



Şekil 2.9 Görüntü işleme algoritması

2.7 Sınıflandırma Sistemi

Sınıflandırma için MATLAB’ın Neural Network araç kutusundaki Neural Clustering Tool uygulaması kullanılmıştır. Bu uygulamada ağ SOM algoritmasıyla eğitilmektedir. Nöronlar 2 boyutlu olarak istenilen boyutta ayarlanabilmektedir. SOM ağına gönderilecek özellikler MATLAB ImageProcessing Toolbox aracında elde edilmiştir.

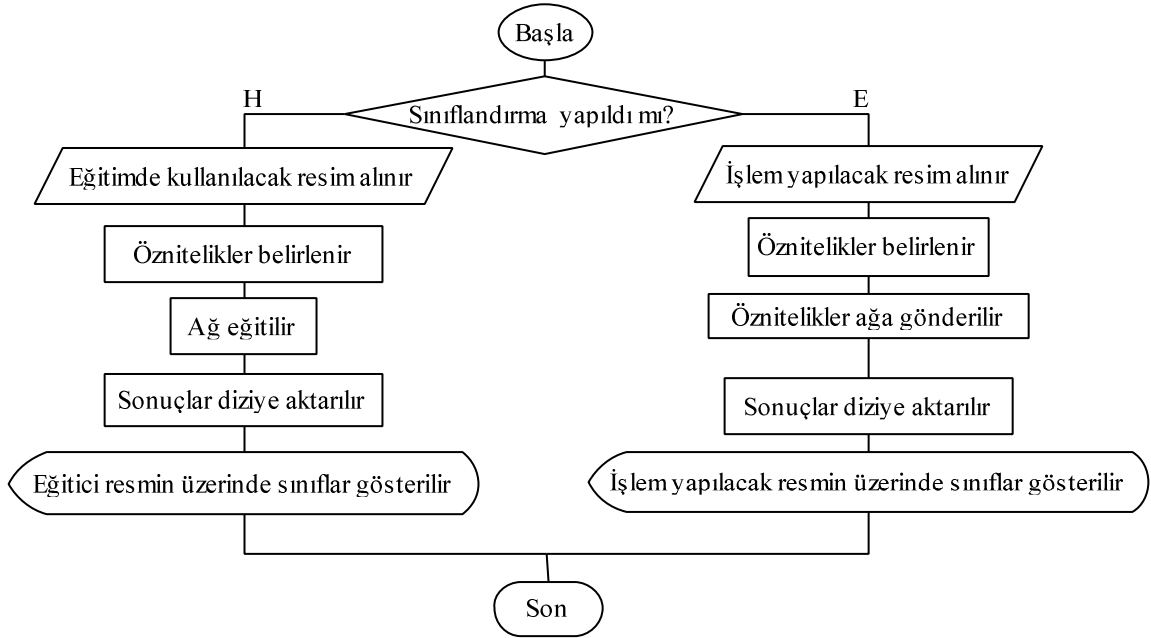
Boyut farklılıklarının etkisini ortadan kaldırmak için algılanan nesnelerin en ve boy değerlerinin orta değeri alınmış, resimdeki bütün nesneler bu en ve boyya sahip olacak şekilde yeniden boyutlandırılmıştır.

Oryantasyonun etkisini ortadan kaldırmak için bütün nesneler yatay pozisyona getirilmiş ve özellikleri bu aşamadan sonra hesaplanmıştır.

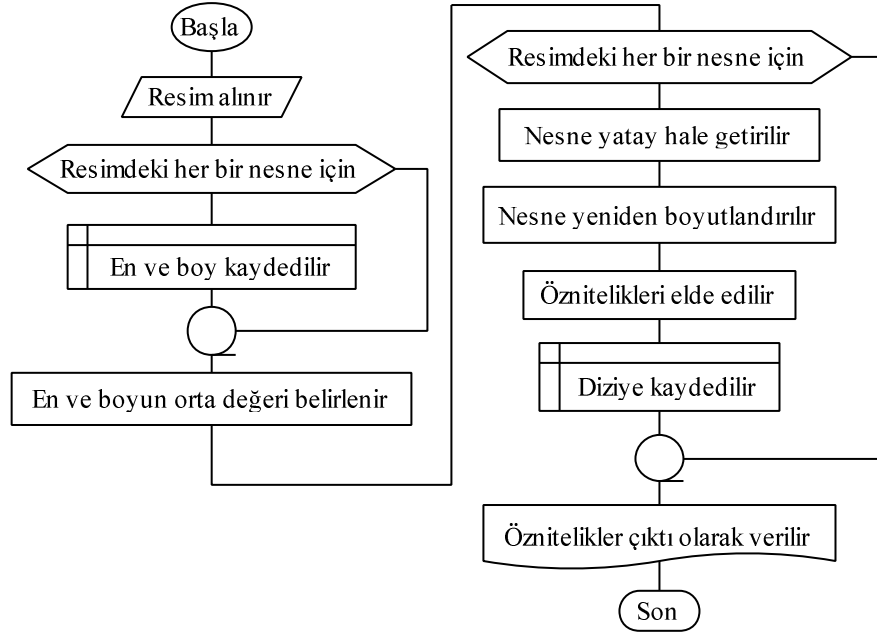
Alınan fotoğraflarda belirlenen objelerin asal eksen uzunlukları, egzantiriklikleri ve alanları sınıflandırma kriteri olarak belirlenmiştir.

Sınıflandırma için giriş vektörü 4 elemanlı olduğundan, en fazla 16 farklı sınıf oluşabilecektir. Bu yüzden ağın hücre sayısı 16 olarak belirlenmiştir.

Resimlerin öznitelikleri hem sınıflandırma yapılırken, hem de robotun çalışması esnasında fotoğrafı çekilen nesnelerin hangi sınıfa ait olduğuna karar verilirken kullanılmaktadır. Bu nedenle öznitelikler ayrı bir programda hesaplanmış, sınıflandırma ve kontrol için gerektiği yerlerde ayrı ayrı çağırılmıştır. Sınıflandırma yapan (EK-6) ve öznitelikleri hesaplayan (EK-7) programların algoritmaları sırasıyla Şekil 2.10 ve Şekil 2.11’de verilmiştir.



Şekil 2.10 Nesne sınıflandırma algoritması



Şekil 2.11 Öznitelik çıkartma algoritması

2.8 Denetim Sistemi

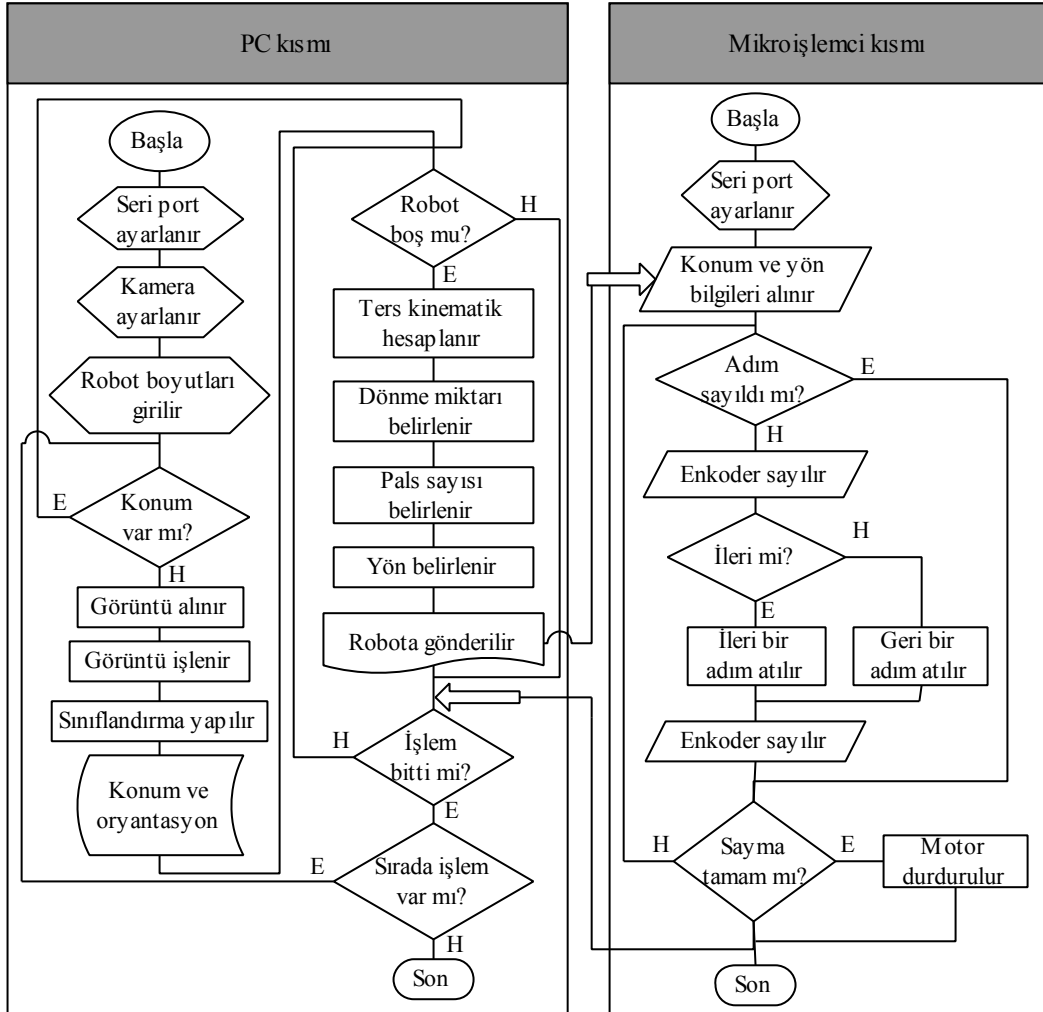
Robotun denetim sistemi bir bilgisayar kamerası, MATLAB ortamında çalışan bir program (EK-8), üzerinde PIC16F877 (EK-18) mikroişlemcisi bulunan bir elektronik devre (EK-17), bu devreye programlanan bir C yazılımı (EK-9), bu devreye bağlı 3 adet kodlayıcılı Japan Servo DM34E50G DC motordan (EK-16) oluşmaktadır.

Bilgisayar kamerası görüntünün alınmasından sorumludur. MATLAB ortamında çalışan yazılımda alınan görüntünün işlenmesi, nesnelerin ayrıştırılması, konum ve oryantasyon bilgilerinin elde edilmesi, bu bilgilerin motorların sürülmesinde kullanılacak adım sayılarına dönüştürülmesi ve motorları süren devreye gönderilmesi işlemleri yapılır. Devreye gömülü halde bulunan C yazılımı ise mikroişlemcinin seri portundan gelen adım sayısı ve yön bilgilerini kullanarak motorların hareketlendirilmesini sağlar. Bu adım bilgilerinin doğruluğunu motor millerine bağlı kodlayıcıları sayarak sağlar.

Robotun denetimi için yalnızca konum ve açı bilgileri yeterli olurken sınıflandırma için diğer bilgiler de gereklidir. Bu bilgiler hem ağın eğitimi için, hem de ağ eğitildikten sonra fotoğrafı çekilen diğer nesneler için elde edilmelidir. Bütün bilgilerin tek bir programda

alınması robotun denetiminde yavaşlamaya sebep olacaktır. Bu nedenle robotun denetimi için gerekli bilgileri veren programla sınıflandırmayı yapan programın ayrı ayrı yazılması gerekmiştir.

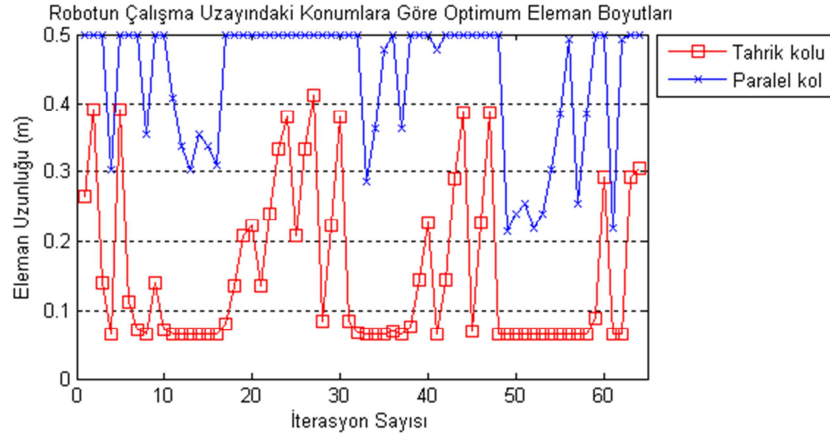
Denetim algoritması Şekil 2.12’da görülmektedir.



Şekil 2.12 Robotun denetim algoritması

3 BULGULAR

3.1 Eleman Boyutları



Şekil 3.1 Eleman boyutu optimizasyonu

Sınırlayıcılar, kolların imal edilebilir boyutlardan küçük; eldeki imkanlarla imal edilebilecek boyutlardan büyük olmaması ve motor torkunun aşılmamasıdır. Buna göre en küçük boyut $65mm$ 'dir. Bu değer en küçük rulmanların kullanılması halinde elemanın sahip olması gereken minimum boyuttur. En büyük boyut ise $500mm$ seçilmiştir. Bu değer de mevcut CNC tezgaha işlenebilecek en büyük boyuttur. Tork kısıtı ise her bir konumda (2.58) denkleminde (EK-12) hesaplanmıştır.

Sabit ve hareketli platformun boyutları $R_b = 54mm$, $R_p = 35mm$ olarak belirlenmiştir. Bu değerler mevcut imkanlarla imalatı yapılabilecek en küçük boyutlardır.

En uygun eleman boyutu Şekil 3.1'deki değerler arasından seçilmiştir. Eleman boyutunun tek bir konuma göre hesaplanması halinde daha yüksek tork gerektirecek diğer konumlara robot ulaşamayacaktır. Bu nedenle her iki elemanın da en küçük boyutlarının seçilmesi gerekmektedir. Buna göre 61. iterasyonda elde edilen 65 ve 218 değerleri seçilmiş, $l_1 = 65mm$, $l_2 = 200mm$ olarak belirlenmiştir.

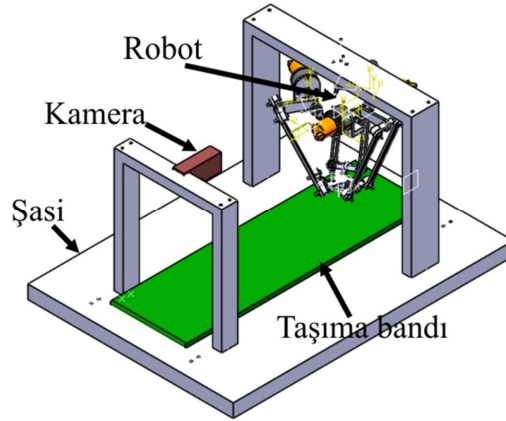
Tablo 3.1'deki d_b parametresi sabit platformun üst düzlemiyle motor millerinin eksenleri arasındaki mesafeyi, d_p parametresi ise hareketli platformun geometrik merkezi ile tutucunun uç konumu arasındaki, z eksenli boyunca olan mesafeyi göstermektedir.

Tablo 3.1 Elemanların boyutları

R_b	d_b	l_1	l_2	R_p	d_p
54mm	15mm	65mm	200mm	35mm	35mm

3.2 CAD Modeli

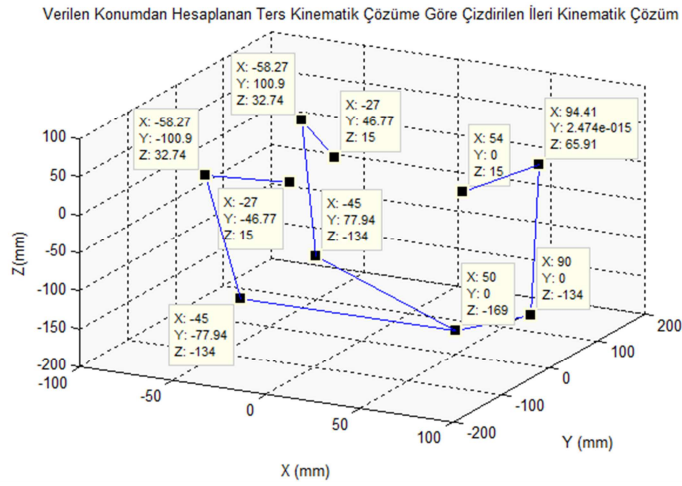
Sistemin CAD modeli CATIA yazılımında oluşturulmuştur. Parçalar teker teker modellendikten sonra montajlar haline getirilmiş ve bir ana montaj altında birleştirilmiştir. Bu model Şekil 3.2'de görülmektedir.



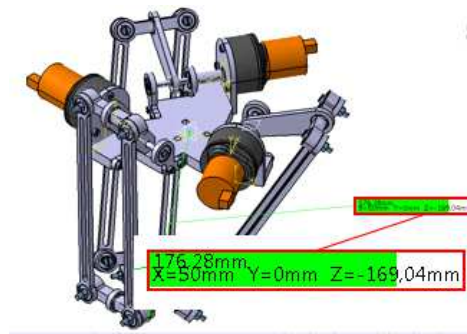
Şekil 3.2 Sistemin CAD modeli

3.3 Ters ve İleri Kinematik

$P=\{50;0;-169,04\}mm$ konumu keyfi olarak belirlenmiş ve (2.30) denkleminde (EK-11) ters kinematik çözüm hesaplanmıştır. Buna göre $\theta_1 = \{51,5593^\circ; 15,8403^\circ; 15,8403^\circ\}$ olarak elde edilmiştir. Daha sonra bu açılar (2.12), (2.13), (2.20) denklemlerinde yerine konularak (2.30) denkleminde ileri kinematik çözüm elde edilmiştir (EK-10). Son olarak Şekil 2.1'e uygun şekilde robotun pozunu (EK-15) çizdirilmiştir. Çizdirilen robot Şekil 3.3'te görülmektedir. CAD modeline de $\theta_1 = \{51,5593^\circ; 15,8403^\circ; 15,8403^\circ\}$ açıları uygulanmış ve sonuçta robotun başta verilen $P=\{50;0;-169,04\}mm$ konumunda bulunduğu tespit edilmiştir. Bu durum Şekil 3.4'te görülmektedir.



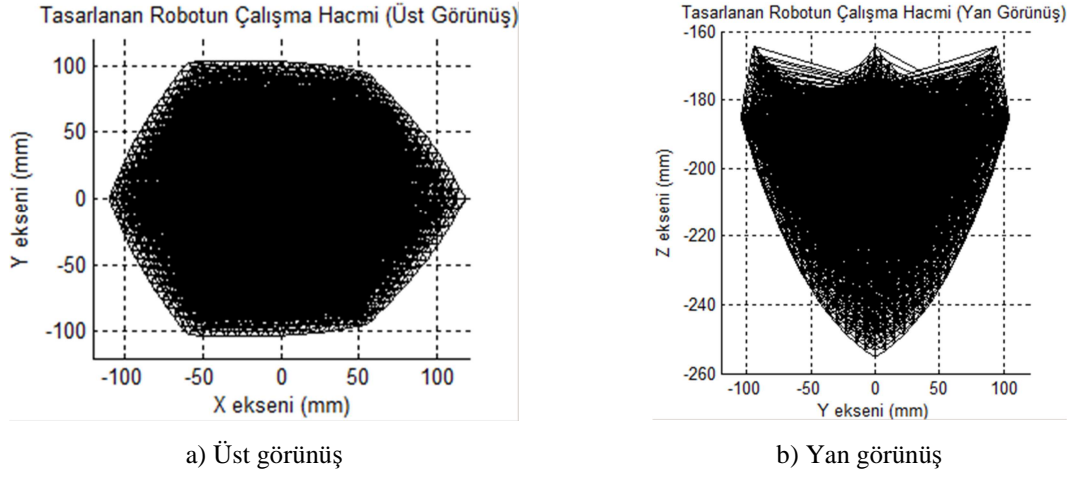
Şekil 3.3 Verilen konumdan hesaplanan ters kinematik çözüme göre çizdirilen ileri kinematik çözüm



Şekil 3.4 Kinematik ifadelerin CAD ortamında doğrulanması

3.4 Çalışma Hacmi

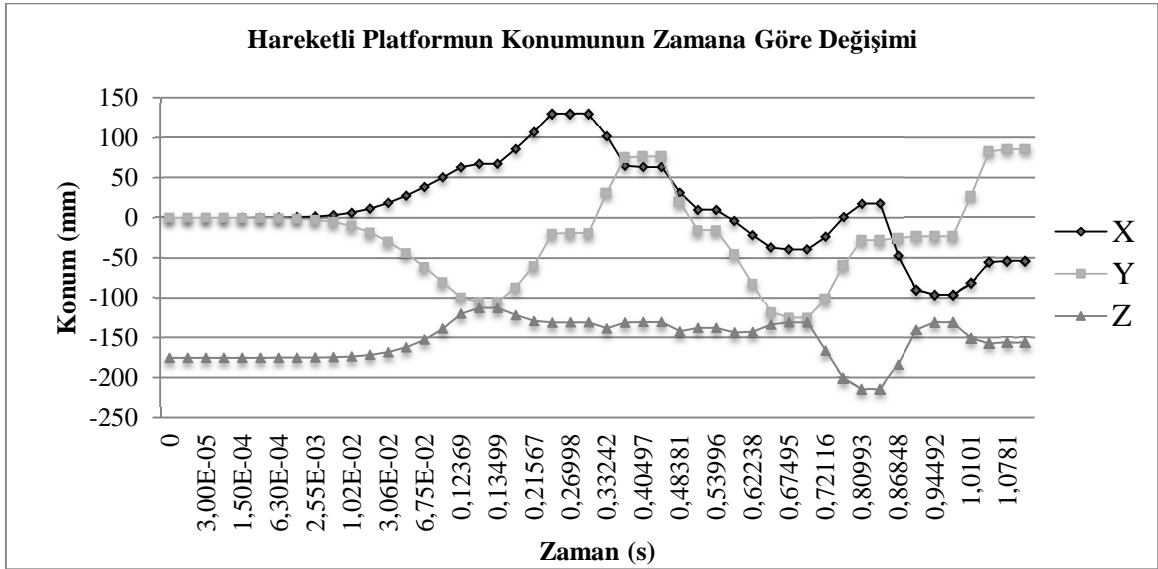
Denklem (2.21)'in robot eklemlerinin hareket aralığında hesaplanmış ve robotun ulaşabileceği bütün konumlar elde edilmiştir. Bu konumlar grafik olarak çizdirilmiştir. Elde edilen grafik Şekil 3.5'te görülmektedir.



Şekil 3.5 Robotun çalışma hacmi

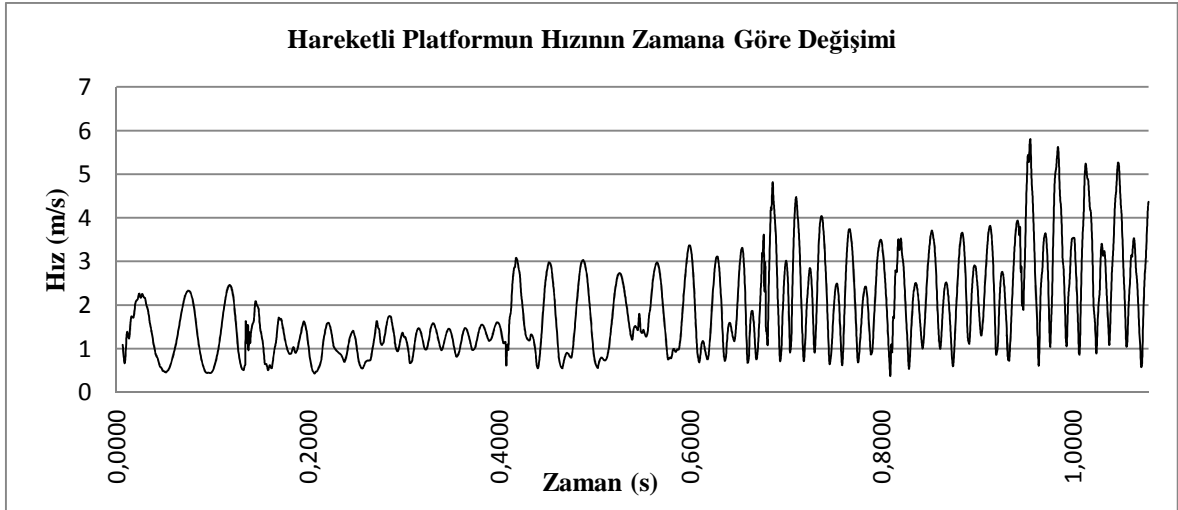
Şekil 3.5'e göre robot $150 \times 100 \times 40 \text{ mm}$ 'lik bir dikdörtgenler prizmasının içinde hareket edebilecek bir çalışma hacminde çalışabilmektedir.

3.5 Hareketli Platformun Konum, Hız ve İvme Değerleri



Şekil 3.6 Hareketli platformun konumunun zamana göre değişimi

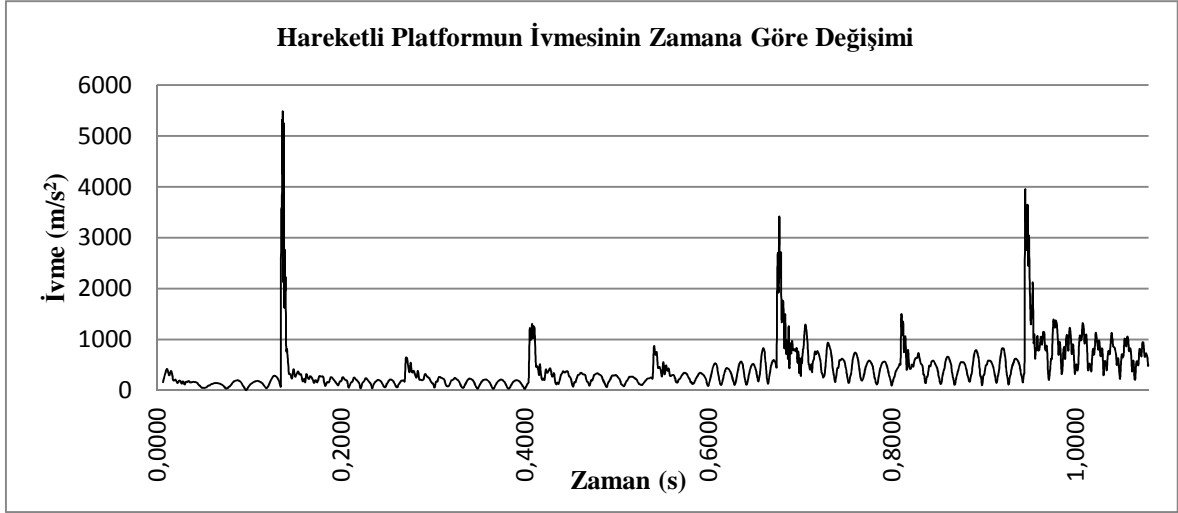
Şekil 3.6'ya göre 2,55E-03s'ye kadar konumda bir değişiklik olmamıştır. Değişiklik olmamasının sebebi analizin yakınsaması için gerekli şartın sağlanmasını gerektiren sürenin küçük olmasındandır.



Şekil 3.7 Hareketli platform hızının zamana göre değişimi

Şekil 3.7'deki değerlere göre hareketli platformun ortalama hızı 2,0235m/s olmuştur. Robotun 0,13499-0,67495s aralığında yaptığı düzlemsel hareket boyunca ortalama hızı ise 1,4252m/s olmaktadır. Tasarlanan robotun 150x100x40mm ölçülerinde bir dikdörtgenler

prizmasında hareket edebildiği göz önüne alınırsa, robotun, çalışma uzayının bir köşesinin alt noktasından uzak köşesinin alt noktasına hareket edebilmesi için gerekli süre 0,182s olmaktadır. Bu da robotun, dakikada teorik olarak 164 kez çalışma hacminin, birbirlerine en uzak noktaları arasında gidip gelebileceğini göstermektedir.

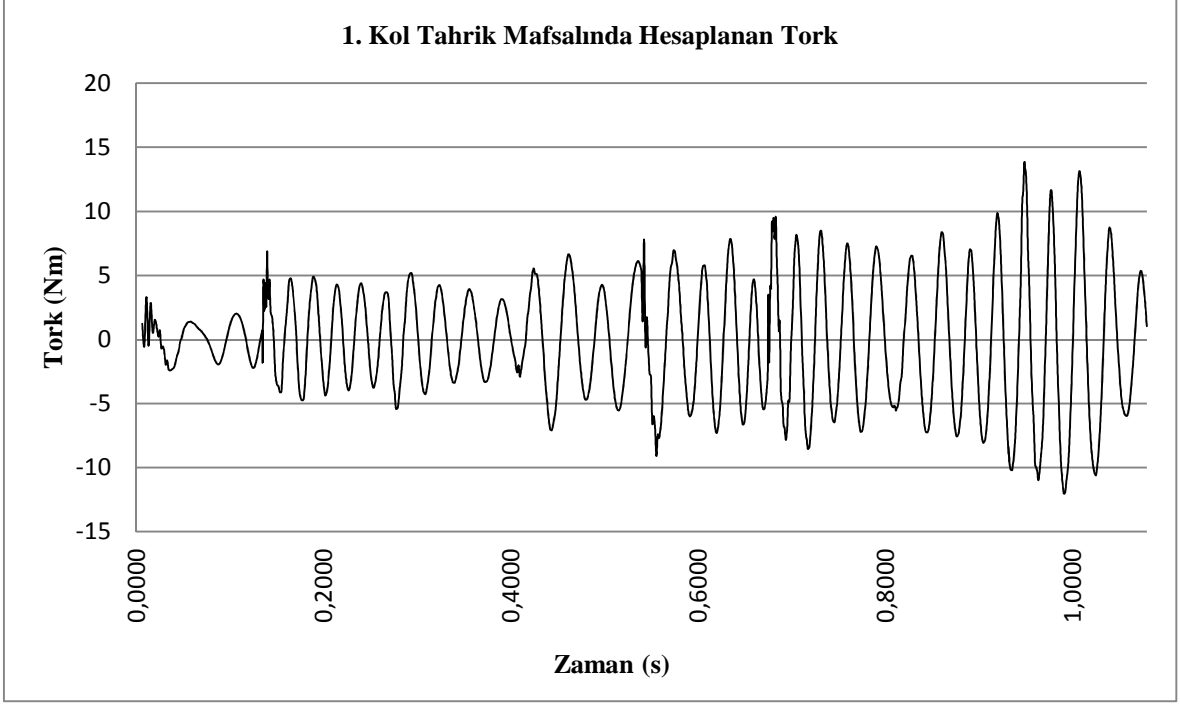


Şekil 3.8 Hareketli platform ivmesinin zamana göre değişimi

Şekil 3.8’de görülen uç noktalar Tablo2.2’de verilen adımların başlangıç noktalarıdır. Adım başlangıçlarında ivme değerleri maksimum olmuş adım süresince azalmıştır. Hareketli platformun ortalama ivmesi 524m/s^2 olmuştur. Bu da yerçekimi ivmesi cinsinden $53,49g$ ’ye denk gelmektedir. Bu değerler [7]’de verilen değerlerle uyumludur. Buna göre hareketli platform hareketi boyunca kendi ağırlığının 53,49 katında ek kuvvete maruz kalmaktadır.

3.6 Tahrik Torkları

Eldeki motorların, belirlenen eleman boyutlarıyla kullanımına yeterli olup olmayacağını görmek için tahrik tork analizi yapılmıştır. Robot, Tablo 2.2’deki hareket sınır şartlarında analiz edilmiştir. Tutucu uca z yönünde -5N ’luk yük uygulanmıştır. Buna göre elde edilen sonuçlar Şekil 3.9’da verilmiştir.

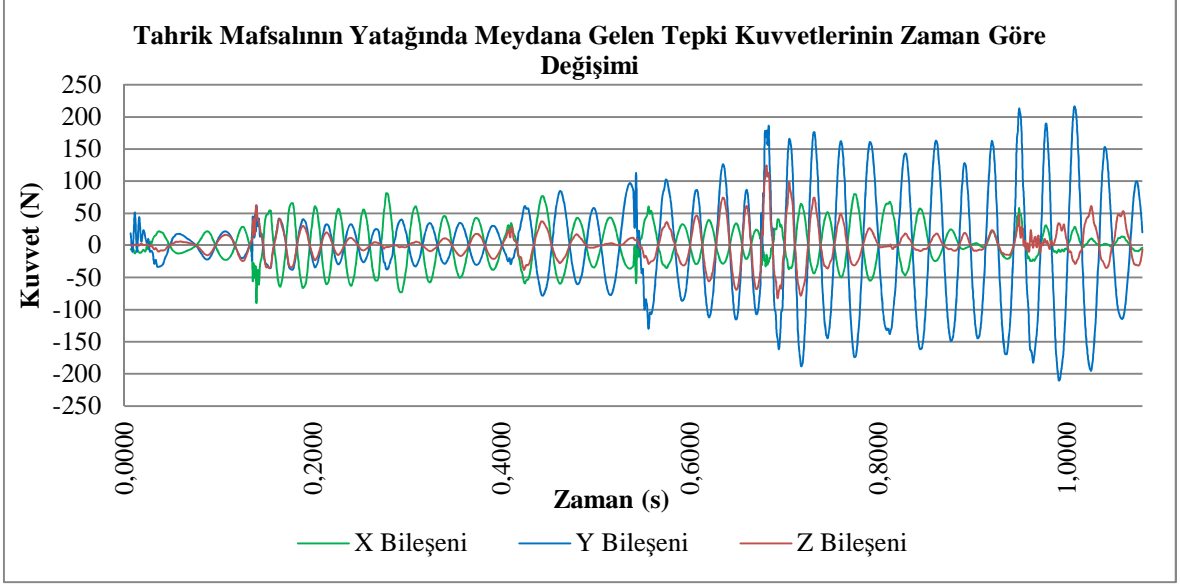


Şekil 3.9 Robotu tahrik etmek için gerekli tork

Şekil 3.9'a göre robotun hareketi esnasında en fazla $13,839Nm$ tork gereklidir. Motor torkunun teorik olarak $0,45Nm$ olduğu göz önüne alındığında motorların yetersiz olduğu ortaya çıkmaktadır. Ancak imâl edilen robot Tablo 2.2'de verilen konumlara deneysel çalışma esnasında ulaşabilmiştir. Buna göre kullanılan motorlar robotu çalışma hacminin tamamında hareket ettirebilecek kapasitedir ancak istenen hız değerlerini sağlayamamaktadırlar.

3.7 Mafsal Kuvvetleri

Mafsallarda kullanılacak rulmanların emniyetli bir şekilde seçilebilmesi için kuvvet analizi yapılmıştır. Yine bu analizde de Tablo 2.2'deki sınır şartları kullanılmıştır. Elde edilen mafsal kuvvetleri grafiği Şekil 3.10'da verilmiştir.

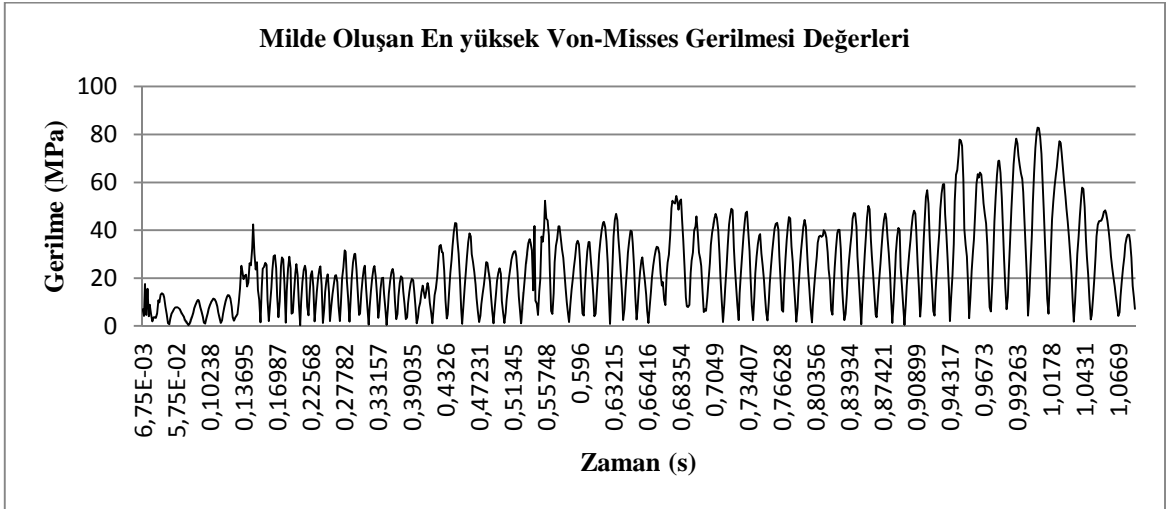


Şekil 3.10 1. Kolin tahrik edilen mafsala gelen kuvvetler

Şekil 3.10'a göre mafsal eksenine dik düzlemdeki en büyük kuvvet 226N olmaktadır.

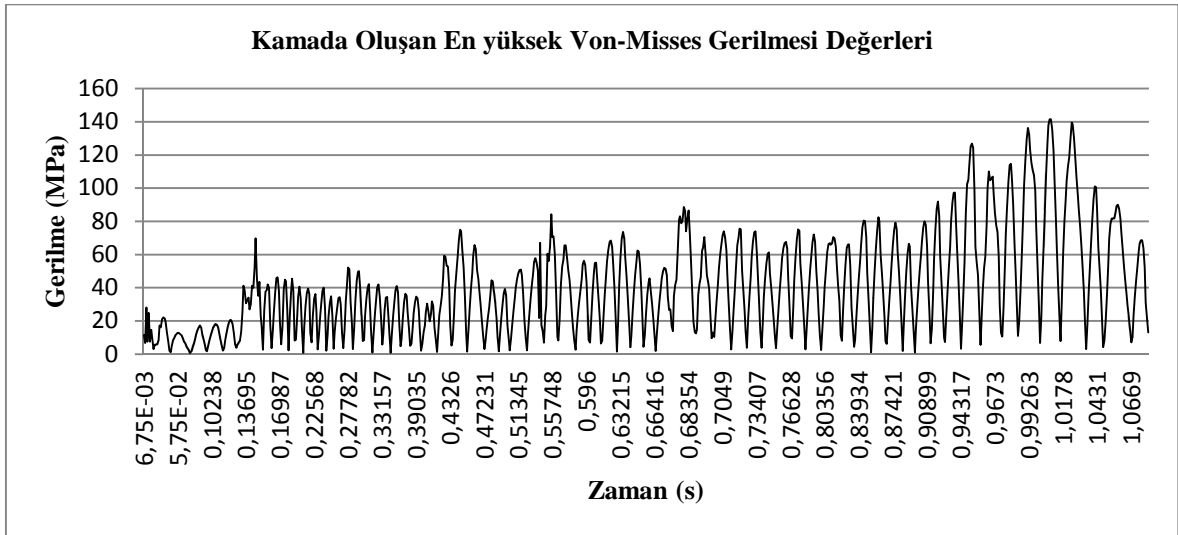
3.8 Mukavemet Analizleri

Robotun mukavemet analizleri Ansys Workbench yazılımında yapılmıştır. Atalet etkilerinin de hesaba dahil olması için Transient Analysis modülü kullanılmıştır. Elemanlar esnek kabul edilmiştir. Elde edilen sonuçlara göre parçaların Gerilme-Zaman grafikleri Şekil 3.11-Şekil 3.23 arasındaki şekillerde verilmiştir.



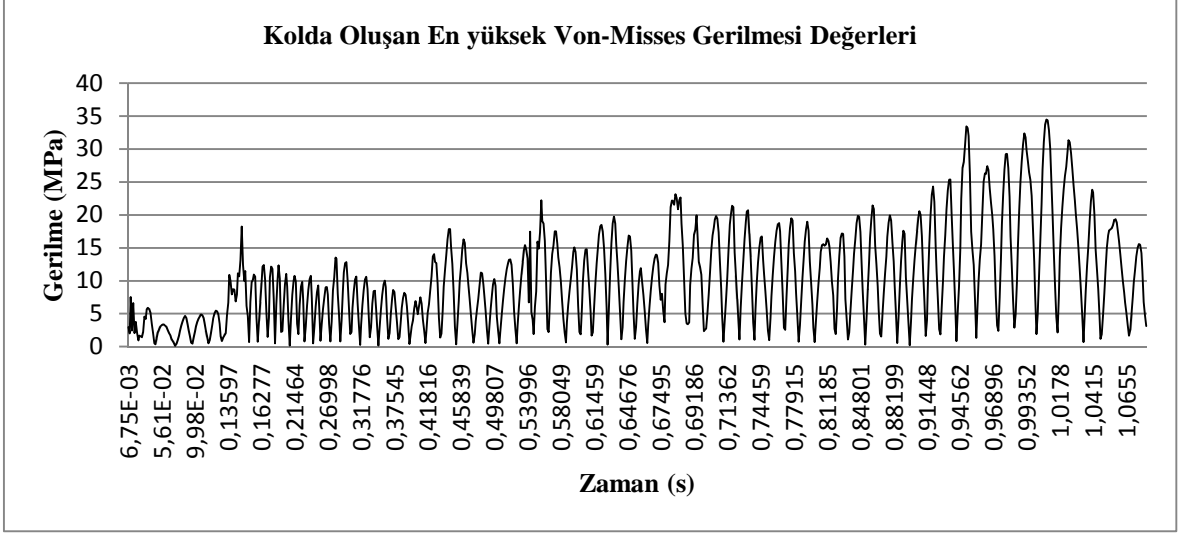
Şekil 3.11 Mil için Gerilme-Zaman grafiği

Şekil 3.11’de görülen grafikte en yüksek gerilme, kamanın bağlandığı kademeli kısımda oluşmuştur. Oluşan en yüksek gerilmenin değeri $83MPa$ ’dır ve $1,0078s$ ’de oluşmuştur. Milin malzemesi transmisyon çeliğidir (Ç1013); akma noktası $465-490MPa$ arasında olduğundan ve oluşan en yüksek gerilme bu değeri geçmediğinden mil emniyetlidir. Grafik incelendiğinde, gerilme değerlerinin, Tablo 2.2’de verilen adımların başlangıç sürelerinde anlık maksimum değerlere çıktığı, adım süresi boyunca bu gerilme değerlerinin azaldığı görülmektedir. Analiz süresi boyunca gerilme değerlerinin sürekli arttığı görülmektedir. Bunun sebebi hareketli platformun izlediği yörüngenin, robotu en yüksek gerilmeye maruz bırakacak yörüngeden geçmesidir. Hareketli platformun izlediği yörünge ile [20]’de verilen yörüngeler karşılaştırıldığında, gerilme değerlerinin uyum içinde olduğu görülmektedir. Bu durum gerilme analizi yapılan diğer parçalar için de geçerlidir.

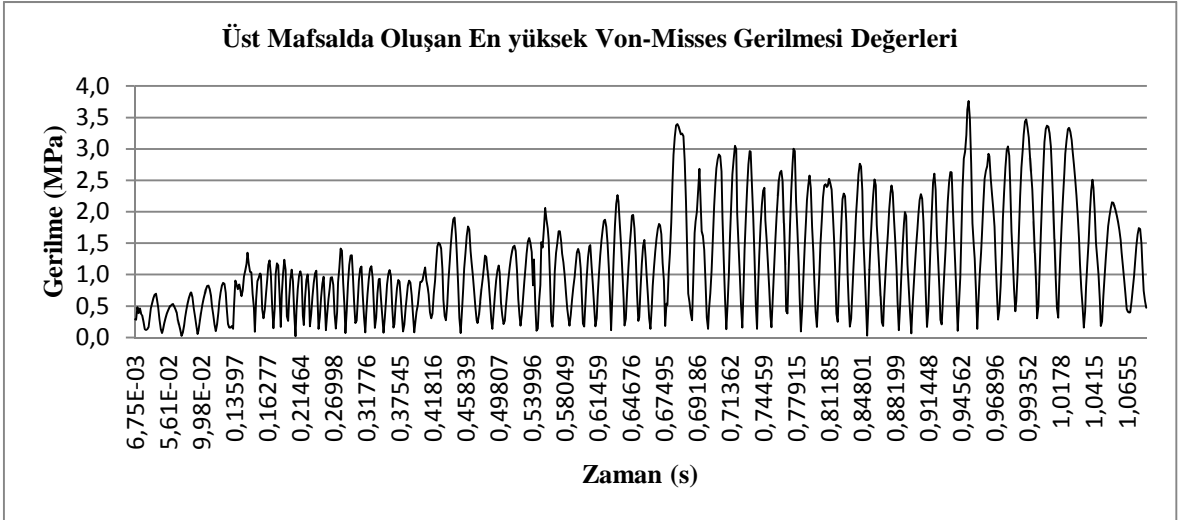


Şekil 3.12 Kama için Gerilme-Zaman grafiği

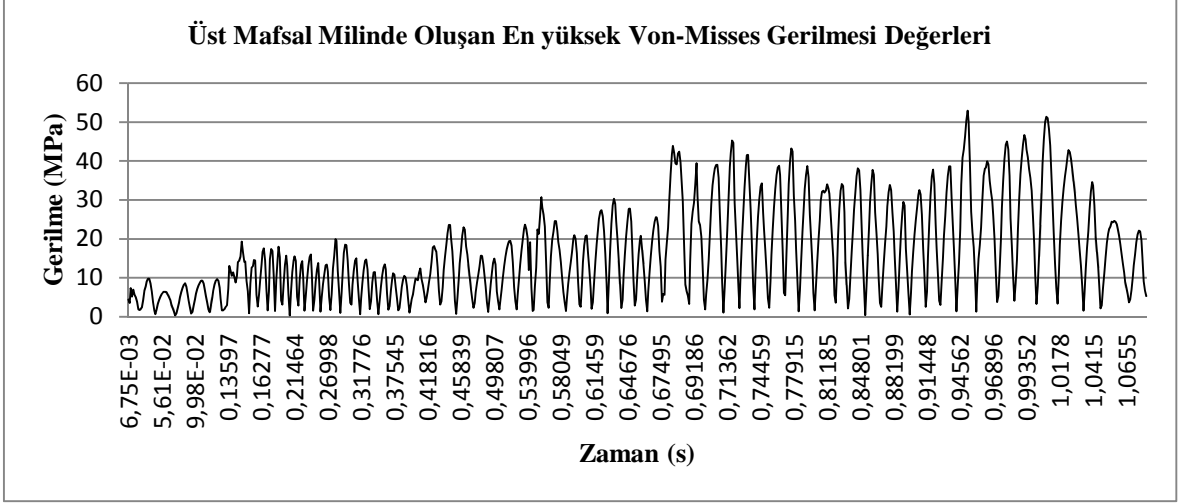
Şekil 3.12’de görülen grafikte en yüksek gerilme değeri $142MPa$ ’dır. Gerilme $1,0084s$ ’de oluşmuştur. Kama malzemesi karbon çeliği (Ç1010) seçilmiştir. Bu değer emniyetlidir.



Şekil 3.13'te grafikte en yüksek gerilme değeri 34,48Mpa'dır. Gerilme 1,0078s'de oluşmuştur. Bu değer emniyetlidir.

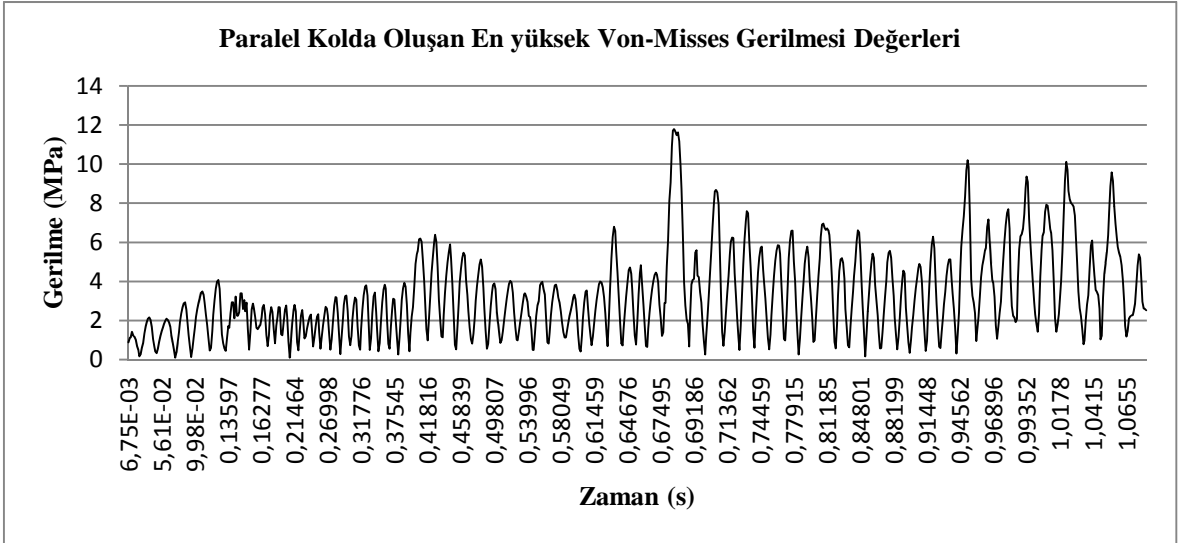


Şekil 3.14'te görülen grafikte en yüksek gerilme 0,94977s'de ve 3,76MPa olarak meydana gelmiştir. Bu değer emniyetlidir. 0,67495s'de görülen sıçrama 5. adımın başlangıç noktasıdır. Analizin devamında sönümlendiği görülmektedir.



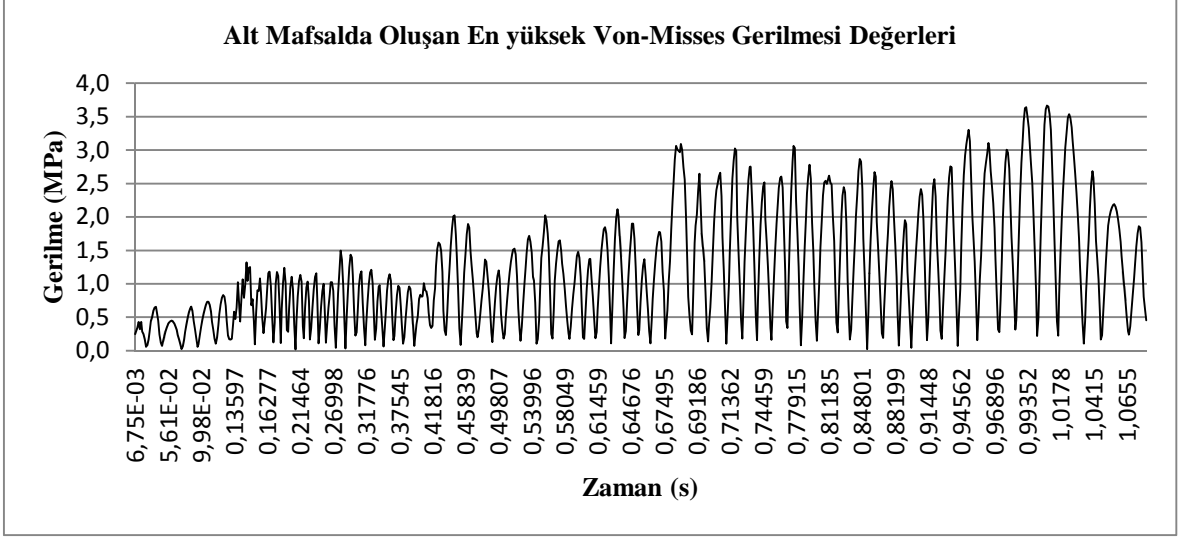
Şekil 3.15 Üst Mafsal Mili için Gerilme-Zaman grafiği

Şekil 3.15’de görülen grafikte en yüksek gerilme 0,94977s’de 53Mpa olarak meydana gelmiştir. Bu değer emniyetlidir.



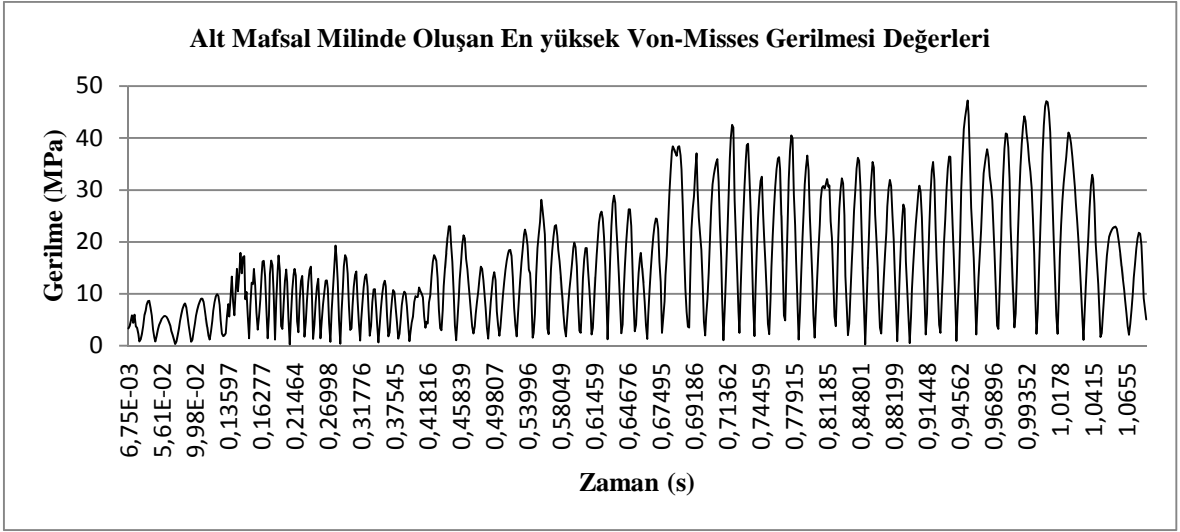
Şekil 3.16 Paralel Kol için Gerilme-Zaman grafiği

Şekil 3.16’da görülen grafikte en yüksek gerilme 0,68065s’de 11,786MPa olarak meydana gelmiştir. Bu değer emniyetlidir.



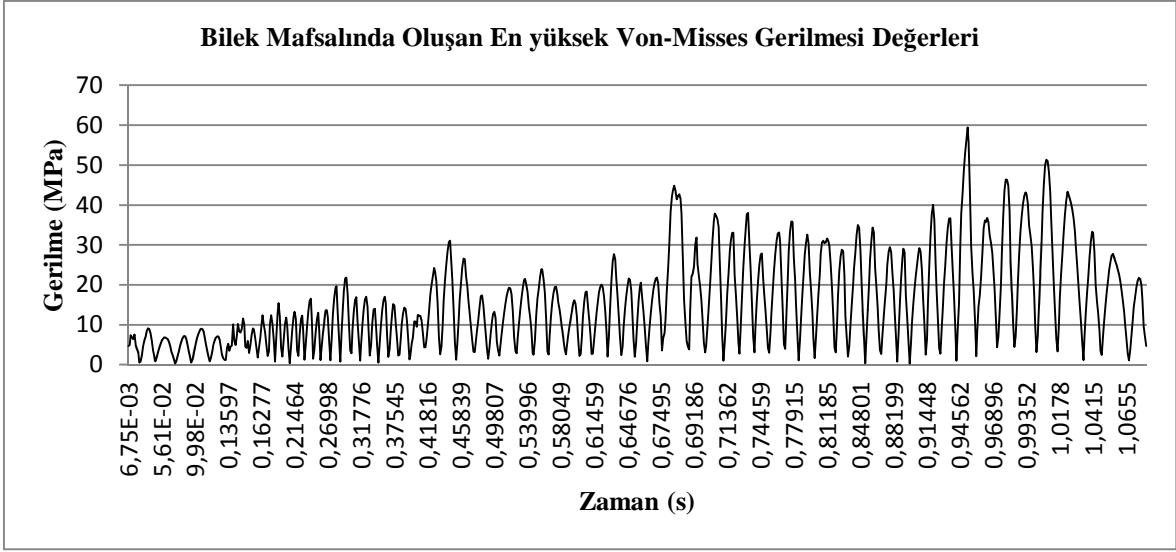
Şekil 3.17 Alt Mafsal için Gerilme-Zaman grafiği

Şekil 3.17’de görülen grafikte, oluşan en yüksek gerilme 1,0078s ‘de meydana gelmiştir. Değeri ise 3,66MPa olmuştur. Bu değer emniyetlidir.



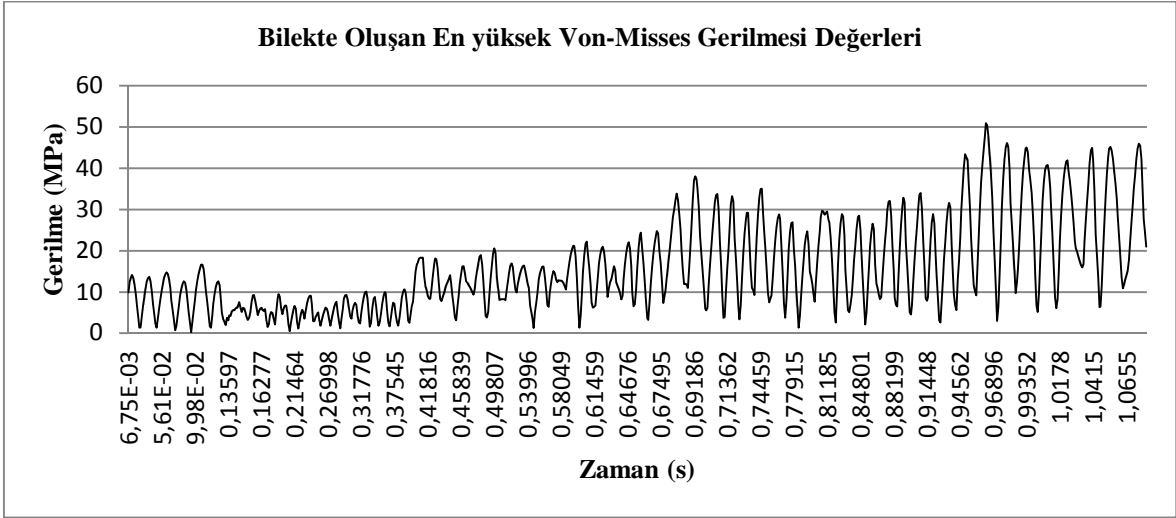
Şekil 3.18 Alt Mafsal Mili için Gerilme-Zaman grafiği

Şekil 3.18’de görülen grafikte, en yüksek gerilme 0,94977s’de 47,26MPa olarak meydana gelmiştir. Bu değer emniyetlidir.



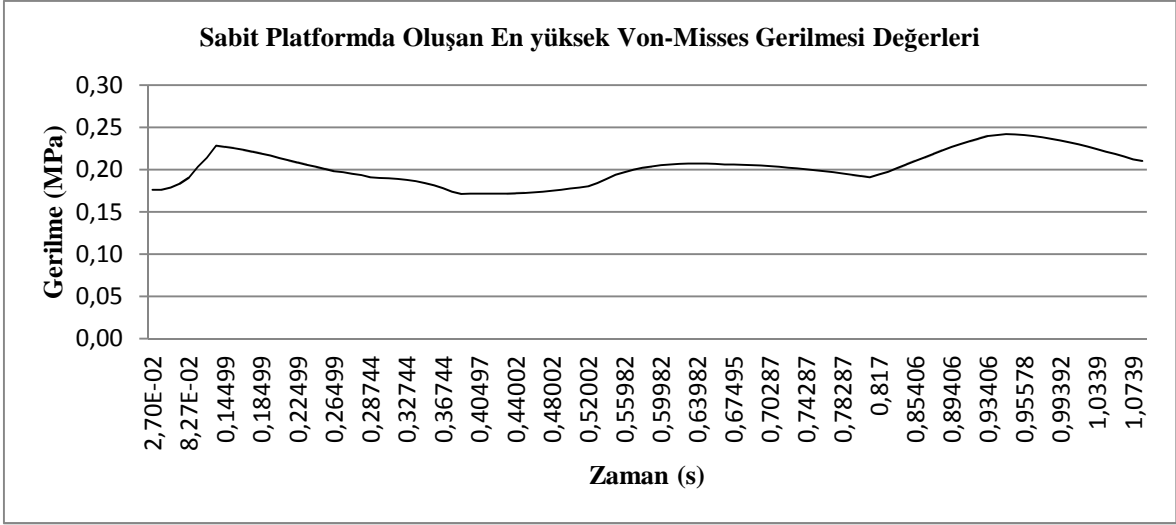
Şekil 3.19 Bilek Mafsalı için Gerilme-Zaman grafiği

Şekil 3.19’da görülen grafikte en yük gerilme 0,94977s oluşmuştur; değeri 59,419MPa olarak meydana gelmiştir. Bu değer emniyetlidir.



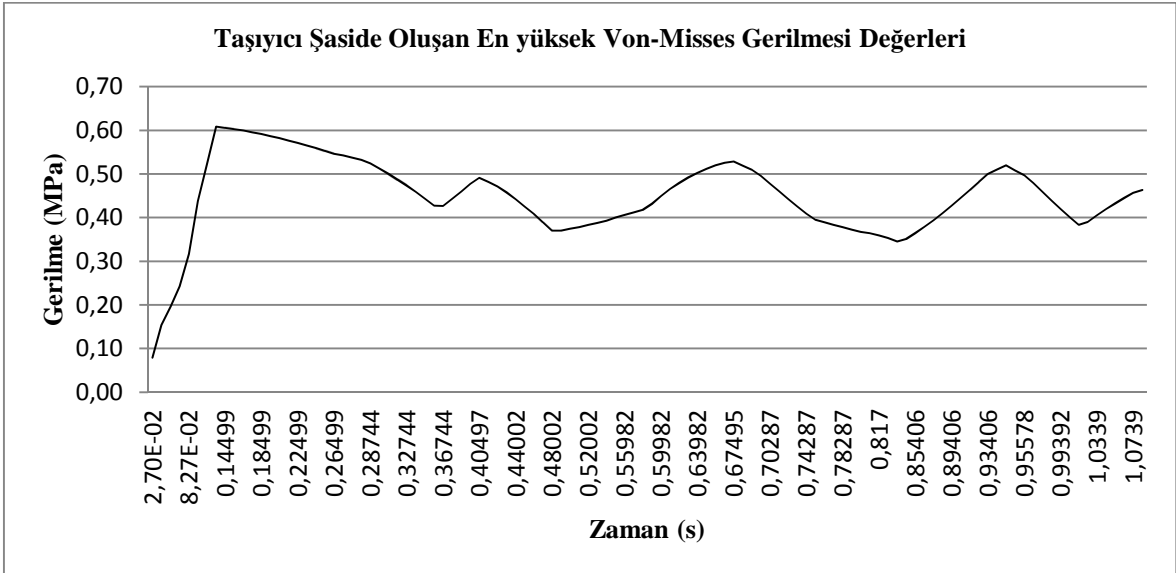
Şekil 3.20 Bilek için Gerilme-Zaman grafiği

Şekil 3.20’de görülen grafikte en yüksek gerilme değeri 0,96273s’de 50,959MPa olarak meydana gelmiştir. Bu değer emniyetlidir.



Şekil 3.21 Sabit platform için Gerilme-Zaman grafiği

Sabit platform hareket etmediğinden atalet etkileri ihmal edilip daha kısa sürede çözüm elde edilmiştir. Statik durum için çözüm yapıldığından gerilme değerlerinde bir salınım görülmemektedir. Şekil 3.21 incelendiğinde grafiğin diğer gerilme grafikleriyle uyum içinde olduğu görülmektedir. En yüksek gerilme 0,94492s'de 0,242MPa olarak meydana gelmiştir. Bu değer emniyetlidir.



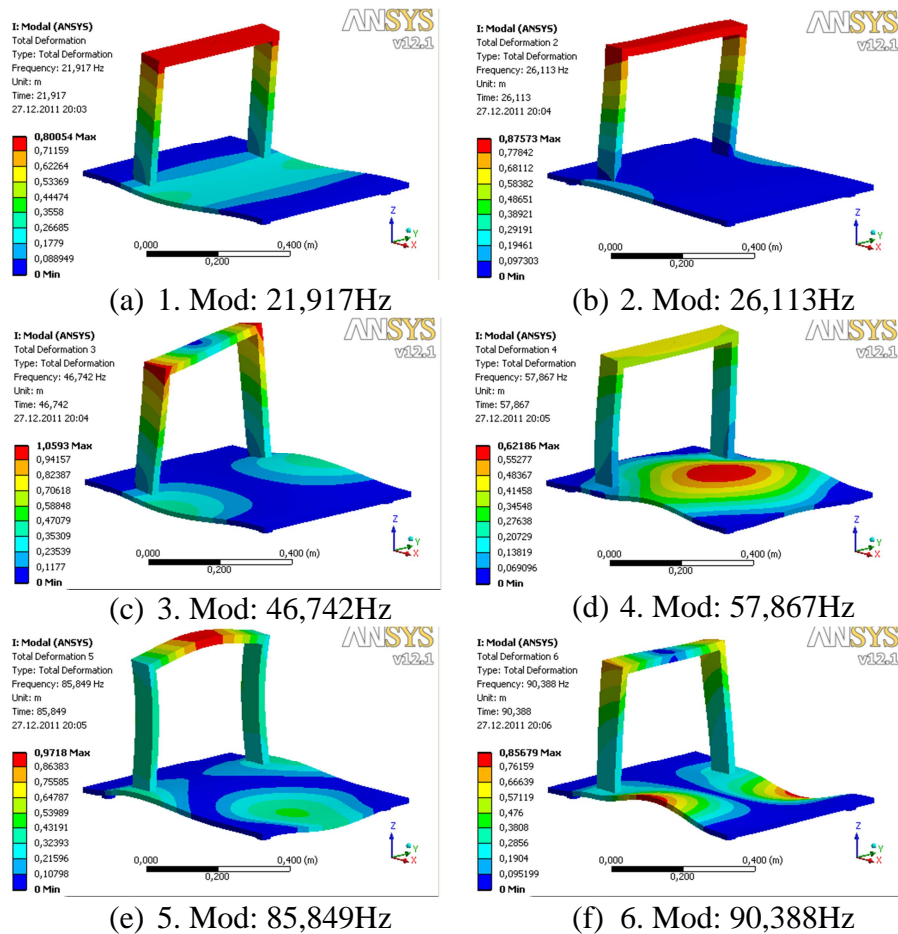
Şekil 3.22 Taşıyıcı şasi için Gerilme-Zaman grafiği

Şekil 3.22'de görüldüğü üzere en yüksek gerilme 0,13499s'de 0,608MPa olarak meydana gelmiştir. Bu değer emniyetlidir.

Şekil 3.21 ve 3.22’de oluşan gerilmeler çok düşüktür. Taşıyıcı şasideki en büyük şekil değiştirme ise yaklaşık olarak $1/10.000m$ olarak meydana gelmiştir. Robotun konum kontrolü yalnızca motor kodlayıcılarından alınan bilgiye göre yapılmaktadır. Bu yüzden taşıyıcı şaside ve sabit platformda yapılabilecek herhangi bir hafifletme veya kesit azaltma çalışması şekil değiştirme miktarını arttırabileceğinden robotun hassasiyetini olumsuz etkileyecektir. Bu nedenle oldukları gibi bırakılmışlardır.

Burkulma analizinde burkulma sabiti 796,37 olarak elde edilmiştir. Bu değer şasinin burkulmaya karşı emniyetli olduğunu göstermektedir.

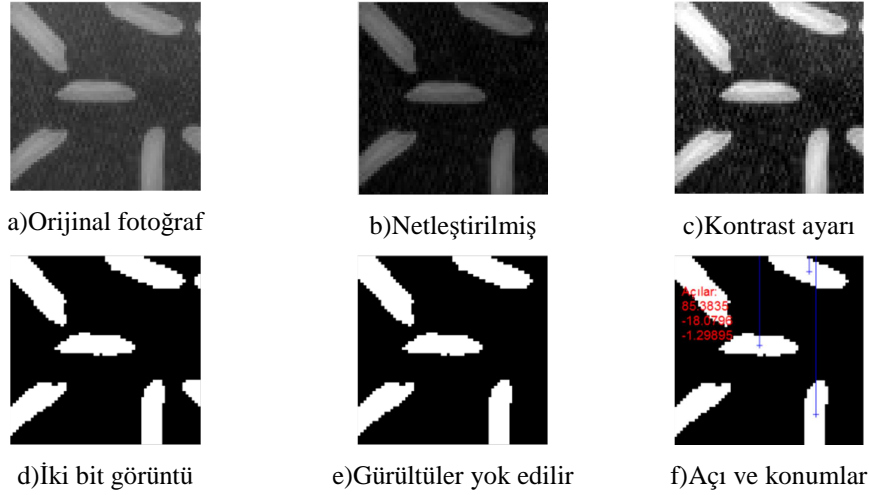
Modal analiz sonucunda elde edilen sistemin 6 mod biçimi ve rezonans frekansları Şekil 3.23’te verilmiştir.



Şekil 3.23 Taşıyıcı şasinin mod biçimleri ve ilgili doğal frekansları

3.9 Görme Sistemi

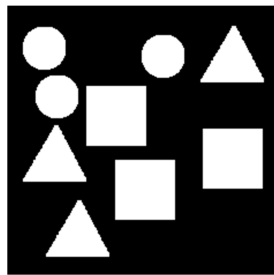
Şekil 2.8 ve Şekil 2.9'deki algoritmalar kullanılarak bir görüntüdeki nesnelerin merkez konumlarının ve oryantasyonlarının elde edilmesi Şekil 3.24'de görülmektedir.



Şekil 3.24 Görüntü işleme adımları

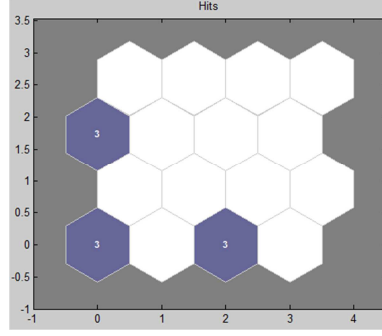
3.10 Sınıflandırma Sistemi

Şekil 2.10 ve Şekil 2.11'deki algoritmalar kullanılarak, bilgisayar ortamında hazırlanan, Şekil 3.25'te görülen resim ağın eğitiminde kullanılmıştır.



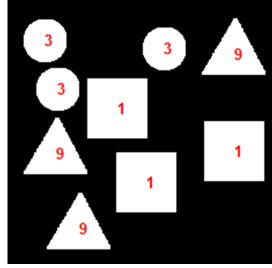
Şekil 3.25 Ağın eğitimi için kullanılan resim

Eğitim tamamlandıktan sonra sınıflar ve ağdaki konumları Şekil 3.26'da görülmektedir. Burada da 9 adet nesnenin 3 farklı sınıfa ayrıldığı görülmektedir.



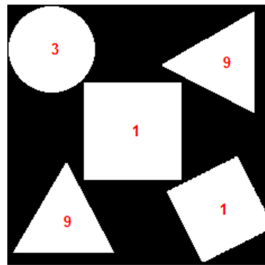
Şekil 3.26 Eğitim sonucu yapılan sınıflandırma

Şekil 3.27’de aynı sınıfa ait olan nesnelerin eşit sınıf numaralarıyla numaralandırıldığı görülmektedir. Buna göre sınıflandırma işlemi başarıyla gerçekleştirilmiştir.



Şekil 3.27 Nesnelerin sınıf aitlikleri

Şekil 3.28’de aynı sınıflara ait nesnelerin boyutları ve oryantasyonları farklı olacak şekilde hazırlanan resim görülmektedir. Bu resmin ağa gönderilmesi halinde alınan sınıf sonuçları Şekil 3.28’de görülmektedir.



Şekil 3.28 Sınıflandırma için ağa gönderilen resim ve sınıflandırma sonuçları

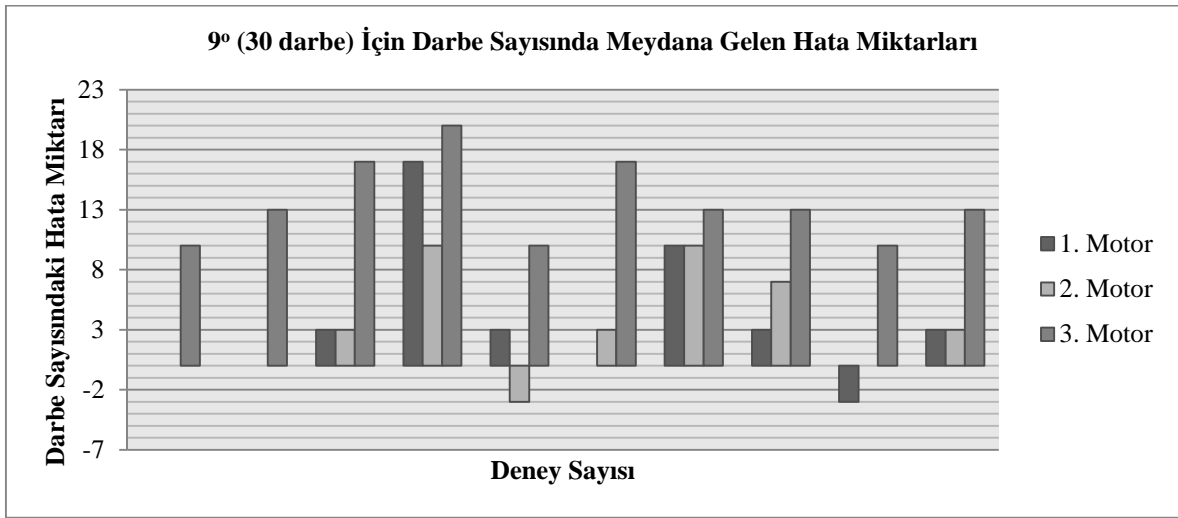
Buna göre Şekil 2.10 ve Şekil 2.11’de verilen algoritmaların kullanılmasıyla oluşturulan sınıflandırma sistemi, eğitim yapıldıktan sonra, aynı sınıflara ait olan nesnelere

boyutlarından ve oryantasyonlarından bağımsız olarak başarılı bir şekilde sınıflandırabilmiştir.

3.11 Tekrarlanabilirlik

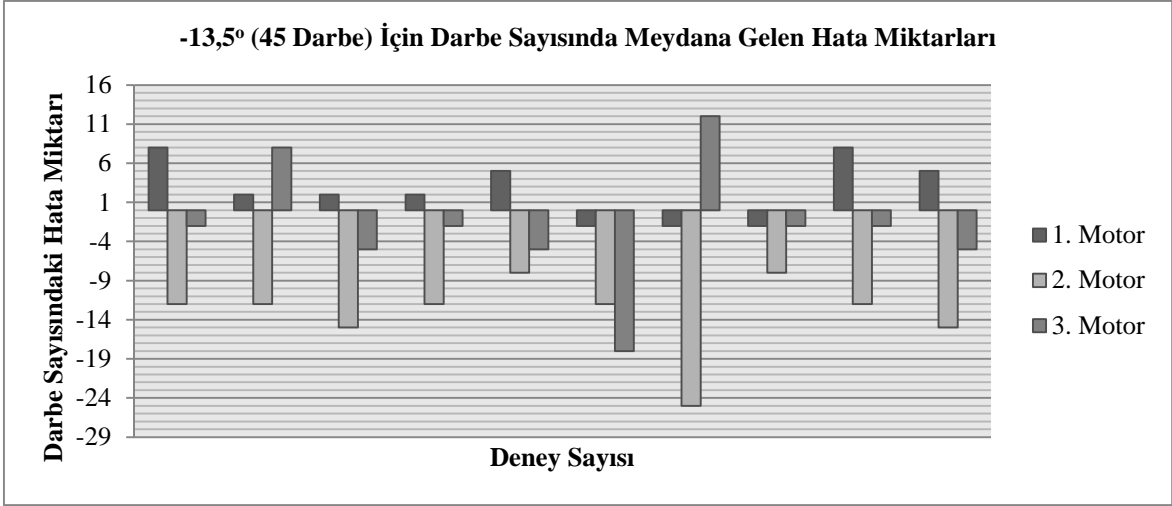
Tekrarlanabilirlik deneyleri robotu kontrol eden motorların millerine açı ölçer bağlanması suretiyle yapılmıştır. Her defasında istenen açının elde edilebileceği darbe sayısı motorlara gönderilmiş ve açı ölçerden konumlar okunmuştur. Okunan açılardan yola çıkılarak darbe sayısındaki ve robotun teorik konumundaki sapma miktarları belirlenmiş, sonuçlar Şekil 3.29 - Şekil 3.32 arasında gösterilmiştir.

Yüzde hata oranları, hata miktarlarının 0-1 arasında normalize edilip, en küçük kareler toplamı yönteminin uygulanmasıyla elde edilmiştir.



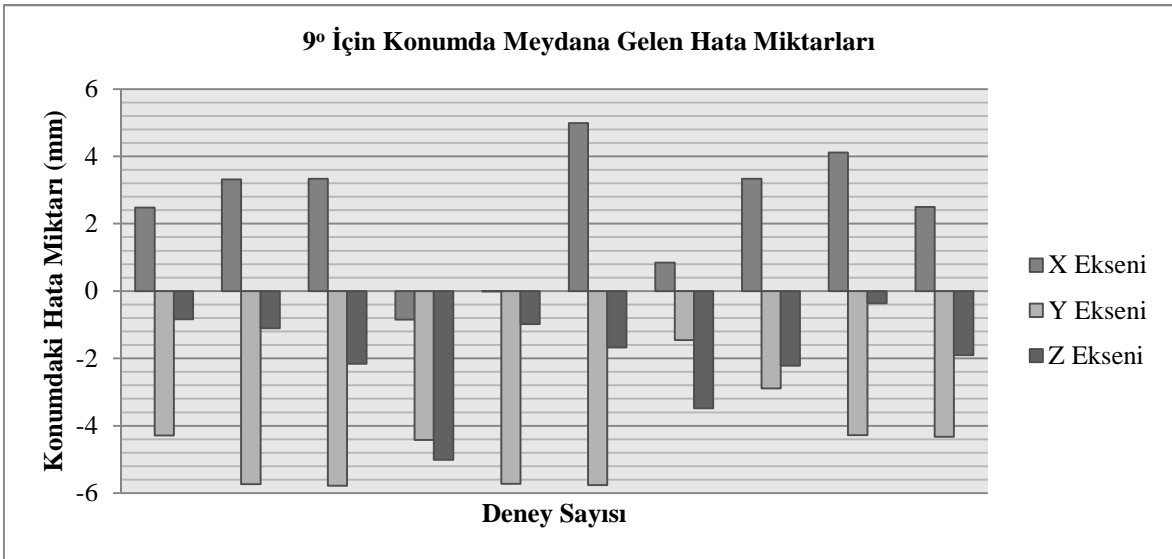
Şekil 3.29 9° (30 darbe) için darbe sayısında meydana gelen hata miktarları

Şekil 3.29'e göre 30 darbe için darbe sayısında meydana gelen en büyük hata 20 darbe olmuştur. Buna göre kodlayıcı 30 darbe sayması gerekirken 50 darbe saymıştır. İlk motorda darbe sayısındaki hata oranı %7,5 olarak gerçekleşmiştir. Darbeler ikinci motorda %14,25, üçüncü motorda ise %24,425 oranında hatalı sayılmıştır.



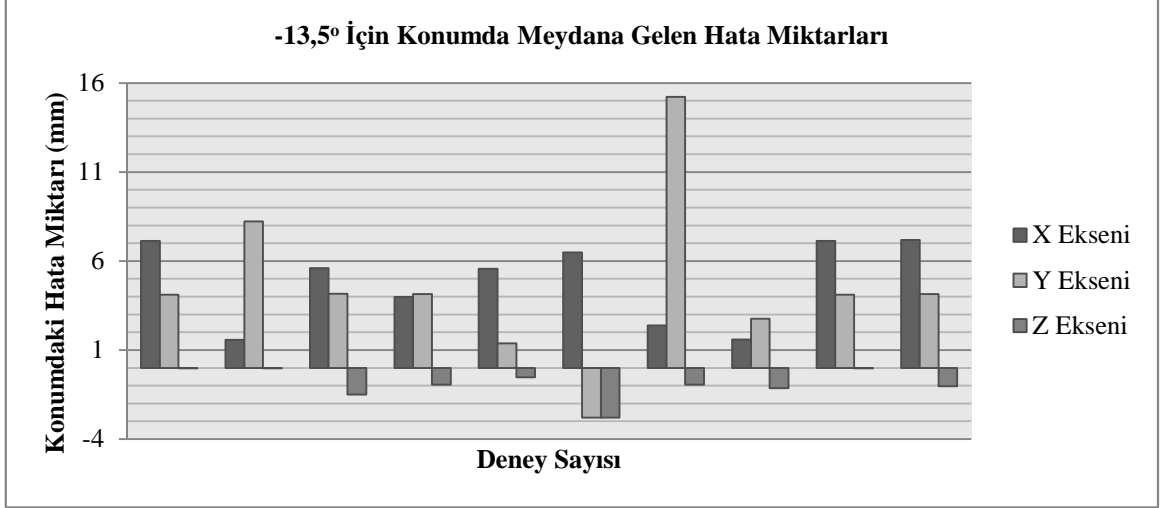
Şekil 3.30 -13,5° (45 darbe) için darbe sayısında meydana gelen hata miktarları

Şekil 3.30'a göre 45 darbe için meydana gelen en büyük sayım hatası 25 darbe olmuştur. 1. Motor için hata oranı %13,5, 2. Motor için %11,4 ve 3. Motor için de %9,1 olarak meydana gelmiştir.



Şekil 3.31 9° için eksen bazında konumda meydana gelen hata miktarları

Şekil 3.31'a göre, 9° için X ekseninde meydana gelen ortalama hata oranı %17,85, Y eksenini için %32,57 ve Z eksenini için de %11,20 olarak meydana gelmiştir.



Şekil 3.32 -13,5° derece için eksen bazında konumda meydana gelen hata miktarları

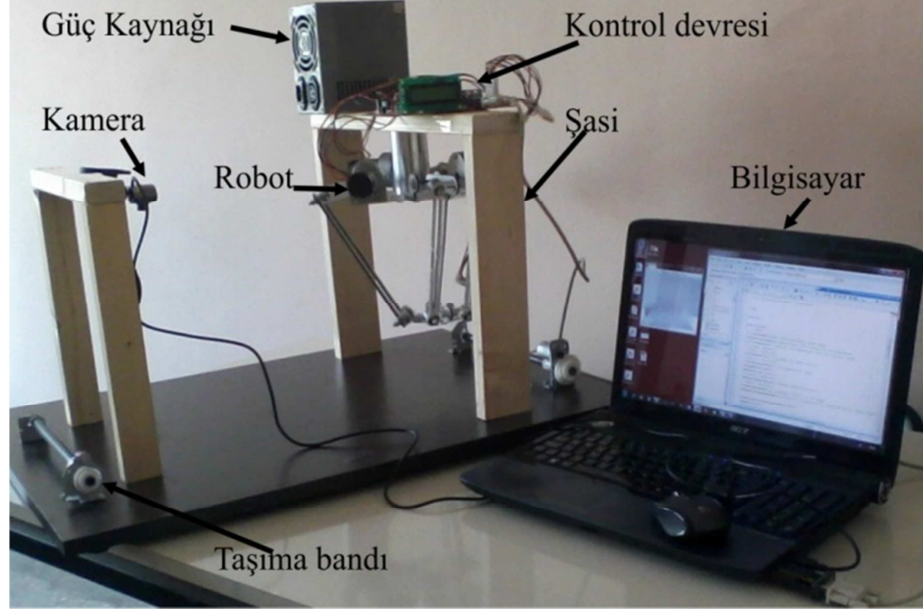
Şekil 3.32'e göre, -13,5° için X ekseninde meydana gelen ortalama hata oranı %26,37, Y eksenini için %8,33 ve Z eksenini için de %9,25 olarak meydana gelmiştir.

Şekil 3.29 - Şekil 3.32 arasındaki grafiklerde gösterilen hataların bu kadar yüksek olmasının sebebi, motorların donanım kesimleri kullanılarak, ayrı ayrı mikroişlemcilerle sayım kontrolü yerine, 3 motorun da tek bir mikroişlemciden kontrol edilmesidir. Bir motorun sayılması için harcanan sürede diğer motorun veya motorların kodlayıcılarında meydana gelen değişiklikler kaçırılabilir.

Sonuç olarak robotun tekrarlanabilirliği %0,8135 olarak elde edilmiştir.

3.12 İmâl Edilen Sistem

Robotun ölçüleri ve kameranın konumu belirlendikten sonra sistemin imal edilmiş hali Şekil 3.33'de görülmektedir.



Şekil 3.33 CAD modellerine uygun olarak imâl edilen robot

4 SONUÇLAR ve TARTIŞMA

Bu çalışmada görerek işlem yapabilen bir Delta robotun tasarımı yapılmış ve performans karakteristikleri araştırılmıştır.

Robotun tasarımı, eleman uzunluklarının, mevcut motor torkuna göre optimum boyutlarda ve eklem uzayında tekilliklerden uzak olacak şekilde yapılmıştır. Elemanların diğer boyutları mevcut imkanlarla, minimum ölçülerde ve maliyette imal edilebilecek şekilde belirlenmiştir.

Gerilme analizleri yapılırken elemanlar esnek kabul edilmiş; atalet ve yerçekimi kuvvetinin etkisi hesaba katılmıştır. Ayrıca robotun taşınması hedeflenen yük de uygulanmıştır. Mevcut motorların kullanılması halinde robot elemanlarında oluşan gerilme değerlerinin emniyetli sınır içinde kaldığı görülmüştür. Bu şekilde robotu istenen hızda çalıştırmak için kullanılacak motorun ve oluşan gerilmeleri taşıyacak parçaların boyutlarının yeterliliği tek bir modelde belirlenmiştir. Deneysel çalışmalarda, belirlenen şartlar için robotun hasara uğramadan çalıştığı görülmüştür.

Geliştirilen kontrol yazılımı, robotun gerçek zamanlı çalışmasına imkan vermiştir. Sınıflandırma işleminde eğitimin tamamlanması için geçen süre buna dahil değildir. Buna göre sınıflandırma eğitimi tamamlandıktan sonra robot gerçek zamanlı olarak çalışabilmektedir.

Geliştirilen görüntü işleme yazılımında algılanan nesnelerin boyutları, alanları, oryantasyon açıları, merkez konumları, asal eksen uzunlukları hesaplanmıştır. Bu özellikler robotun ayırma, dizme, tutma ve yerleştirme işlemleri gibi insanların ya çok yavaş veya yetersiz, ya da tehlikeye maruz kalacağı veya hijyenik risk nedeniyle istenmediği durumlarda endüstrinin ihtiyacını gidermek için kullanılabilir.

Robotun, teorik olarak dakikada 164 defa çalışma hacminin en uzak noktaları arasında gidip dönme kabiliyeti vardır. Bu hareket esnasında hareketli platform ortalama 524m/s^2 'lik doğrusal ivmelenmeye maruz kalmaktadır.

Sınıflandırma yazılımında, boyut ve oryantasyondan bağımsız olarak görüntüsü alınan nesnelerin sınıfları doğru şekilde belirlenmiştir.

Robot yalnızca görme sisteminden aldığı bilgilerle, dışarıdan müdahaleye gerek duymadan çalışabilmiştir. Ancak bu haliyle endüstride kullanımı sınırlıdır. Daha geniş çaplı uygulamalarda kullanılabilmesi için insan-makine arabiriminin uygulamaya özel olarak geliştirilmesi gereklidir.

Bu tip robotlar, literatürde çalışma hacmine göre tasarlanıp, bu amaç elde edildikten sonra mümkün olan en yüksek maliyet/performans oranına sahip bileşenlerin seçilmesiyle analiz edilirler. Mevcut imkanlara göre tasarımda gerekli adımların gösterilmesi açısından bu çalışma bir farklılık getirmiştir.

[7], [9] ve [12]'de yapılan dinamik analizlerdeki basitleştirmelerden, yalnızca eklemlerin sürtünmesiz olması bu çalışmada kullanılmıştır. [20]'deki çalışmadan farklı olarak da yalnızca tahrik kolu değil, sabit platformdan hareketli platforma kadar (hareketli platform dahil) bütün elemanlar esnek olarak modellenip analiz edilmiştir. [7], [9], [12] ve [20]'de yapılan çalışmaların her birinde robotun izleyeceği yörünge önceden bellidir. [7] ve [9]'da ise başlangıç hız ve ivmelerinin de bilinmesi gereklidir. Ancak bu çalışmada yapılan analizde yalnızca robot eklemlerinin çalışma aralığının bilinmesi yeterlidir. Bu çalışmada izlenen analiz yöntemi, robotun ulaşabileceği ve eklemlerin birbirlerine göre bütün konumlarını içerdiğinden sonuçların daha genel geçer olduğu söylenebilir. Ayrıca robotun yalnızca hareketli platformuna değil, herhangi bir bölgesine etkiyen yükler altında performansının analiz edilebilmesi sağlanmıştır.

Bu çalışmada izlenen analiz yöntemi kullanılarak yalnızca tahrik torqları değil, ayrıca robotun bağlandığı şasi kısmına etkiyen yükler de belirlenebilmiştir. Bu, konum hassasiyetinin robotu destekleyen konstrüksiyonu da içine alacak şekilde analiz edilebilmesini sağlamaktadır.

5 ÖNERİLER

Bu çalışmada robotun yalnızca noktadan noktaya konum kontrolü yapılmıştır. Motorlarda kullanılan kodlayıcıların değiştirilip yazılıma yörünge planlamanın eklenmesiyle hız ve yörünge kontrollü bir hale getirilebilir. Bu çalışma, bu açıdan bir CNC veya medikal robot tasarımına temel olabilir.

Mikroişlemci devresinde kodlayıcılar, donanım kesmeleri yerine sinyal portlarına bağlanmıştır. Bu da zaman gecikmeleri ve seri işlem nedeniyle kodlayıcı darbelerinin sayılmasında bazı hatalara neden olmuştur. Kodlayıcıların sayısınca donanım kesmesine sahip bir mikroişlemci kullanılması halinde, robotun tekrarlanabilirliği belirgin şekilde olumlu yönde etkilenecektir.

[7]'de verilen robot bu çalışmada verilen yöntemle modellenip sonuçlar yine [7]'de verilen deneysel sonuçlarla karşılaştırılırsa %10 değerinden daha yakın sonuçlar elde edileceği düşünülmektedir. [9]'da verilen robotun bu çalışmadaki yöntemle modellenip analiz edilmesi halinde ise sonuçlar arasında %10'luk bir uyumun olması beklenmektedir.

6 KAYNAKLAR

1. **Merlet, J.P.**, Paralel Robots SE, Springer, 2006.
2. **Angeles, J.**, Fundamentals of Robotic Mechanical Systems: Theory, Methods and Algorithms, Springer, New York, 1997.
3. **Zhang, D.**, Parallel Robotic Machine Tools, Springer, 2010.
4. **Khatib, O., Siciliano, B.**, Handbook of Robotics, Springer, ISBN: 978-3-540-23957-4, Berlin Heidelberg, 2008.
5. **Zhou, H., Wu, J., Zhang, J.**, Digital Image Processing Part I, ISBN: 978-87-7681-541-7, Ventus Publishing, 2010.
6. **Gonzalez, R.C., Woods, R.E.**, Digital Image Processing Second Edition, ISBN: 0-201-18075-8, Prentice Hall, 2002.
7. **Miller, K.**, 1995, Experimental Verification of Modeling of DELTA Robot Dynamics by Direct Application of Hamilton's Principle, IEEE International Conference on Robotics and Automation, 532-537.
8. **Vischer, P., Clavel, R.**, 1998, Kinematic Calibration of The Parallel Delta Robot, Robotica, **16**, 207-218.
9. **Staicu, S., Carp-Ciocardia, D.C.**, 2003, Dynamic Analysis of Clavel's Delta Parallel Robot, IEEE International Conference on Robotics & Automation, 4116-4121.
10. **Angel, L., Saltaren, R., Sebastian, J.M., Hansson, A., Aracil, R.**, 2004, Robotenis: Parallel Robot With Visual Control, IEEE, **15**, 402-412.
11. **Zsombo-Murray, P.J.**, McGill University, Department of Mechanical Engineering, Centre for Intelligent Machines, Kanada, 2004.
12. **Li, Y., Xu, Q.**, Dynamic Analysis of a Modified DELTA Parallel Robot for Cardiopulmonary Resuscitation, Department of Electromechanical Engineering, Faculty of Science and Technology University of Macau, Çin, 2005.
13. **Staicu, S., Carp-Ciocardia, D.C.**, Dynamics of Delta Parallel Robot with Linear Actuators, 2005, IEEE International Conference on Mechatronics, 870-875.
14. **Merlet, J.P., Donelan, P.**, 2006, On the Regularity of the Inverse Jacobian of Parallel Robots, Advances in Robot Kinematics, **1**, 41-48.

15. **Nabat, V., Company, O., Pierrot, F., Poignet, P.**, 2006, Dynamic Modeling and Identification of Par4 A Very High Speed Parallel Manipulator, IEEE/RSJ International Conference on Intelligent Robots and Systems, 496-501.
16. **Güneri, B., Saide Sarigül, A., Karagülle, H.**, 2007, Kinematic and Kinetic Considerations on a Stewart Platform in Terms of Workspace, The 5th IFAC Intl., 191-196.
17. **Cherfia, A., Zaatri, A., Giordano, M.**, 2007, Kinematics Analysis of a Parallel Robot With a Passive Segment, *Ingeniare. Revista chilena de ingenieria*, **Vol.15 No.2**, 141-148.
18. **Laribi, M.A., Romdhane, L., Zegloul, S.**, 2007, Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace, *Mechanism and Machine Theory*, 42, 859–870.
19. **Selvakumar, A.A., Sivaramakrishnan, R., Srinivasa, K.T.V, Ramakrishna, V.S., Vinodh., B.**, 2009, Simulation and Workspace Analysis of a Tripod Parallel Manipulator, *World Academy of Science Engineering and Technology*, **57**, 624-629.
20. **Rueda, J.D., Angel, L.**, 2009, Structural Analysis of a Delta-Type Parallel Industrial Robot Using Flexible Dynamic of ANSYS 11.0, *Industrial Electronics Society (Annual Conference IECON)*, 2247-2252.
21. **Staicu, S.**, 2009, Inverse dynamics of the 3-PRR planar parallel robot, *Robotics and Autonomous Systems*, **57**, 556-563.
22. **Erzincanlı, F., Sharp J.M.**, 1997, A classification system for robotic food handling, *Food Control*, **8**, 191-197.
23. **Lee, K., Qian, Y.**, 1998, A vision-guided fuzzy logic control system for dynamic pursuit of a moving target, *Microprocessors and Microsystems*, **21**, 571-580.
24. **Song, W., Lee, H., Bien, Z.**, 1999, KARES: Intelligent wheelchair-mounted robotic arm system using vision and force sensor, *Robotics & Autonomous Systems*, **28**, 83-94.
25. **Bone, G.M., Capson, D.**, 2003, Vision-guided fixtureless assembly of automotive components, *Robotics and Computer Integrated Manufacturing*, **19**, 79-87.
26. **Domingues-Lopez, J.A., Damper, R.I., Crowder, R.M., Harris, C.J.**, 2004, Adaptive neurofuzzy control of a robotic gripper with on-line machine learning, *Robotics and Autonomous Systems*, **48**, 93-110.
27. **Mitziás, A.M., Mertzios, B.G.**, 2004, A neural multiclassifier system for object recognition in robotic vision applications, *Measurement*, **36**, 315-330.
28. **Edinbarough, I., Balderas, R., Bose, S.**, 2005, A vision and robot based on-line inspection monitoring system for electronic manufacturing, *Computers in Industry*, **56**, 986-996.

29. **Lopez-Juarez, I., Corona-Castuera, J., Pena-Cabrera, M., Ordaz-Hernandez, K.,** 2005, On the design of intelligent robotic agents for assembly, *Information Sciences*, **171**, 377-402.
30. **Pires, J.N., Paulo, S.,** 2005, Complete robotic inspection line using PC-based control, supervision and parameterization software, *Robotics & Computer-Integrated Manufacturing*, **21**, 11-17.
31. **Du, C., Sun, D.,** 2006, Learning techniques used in computer vision for food quality evaluation: a review, *Journal of Food Engineering*, **72**, 39-55.
32. **Tsai, M.J., Hwung, J.H., Lu, T., Hsu, H.,** 2006, Recognition of quadratic surface of revolution using a robotic vision system, *Robotics & Computer-Integrated Manufacturing*, **22**, 134-143.
33. **Yu, S., Jang, J., Han, C.,** 2007, Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel, *Automation in Construction*, **16**, 255-261.
34. **Freda, L., Oriolo, G.,** 2007, Vision-based interception of a moving target with a nonholonomic mobile robot, *Robotics & Autonomous Systems*, **55**, 419-432.
35. **Jayaweera, N., Webb, P.,** 2007, Adaptive robotic assembly of compliant aero-structure components, *Robotics and Computer-Integrated Manufacturing*, **23**, 180-194.
36. **Golnabi, H., Asadpour, A.,** 2007, Design and application of industrial machine vision systems, *Robotics and Computer-Integrated Manufacturing*, **23**, 630-637.
37. **Eberhardt, M., Roth, S., König, A.,** 2008, Industrial application of machine-in-the-loop-learning for a medical robot vision system - Concept & comprehensive field study, *Computers and Electrical Engineering*, **34**, 111-126.
38. **Nickels, K., DiCicco, M., Bajracharya, M., Backes, P.,** 2010, Vision guided manipulation for planetary robotics, *Robotics and Autonomous Systems*, **58**, 121-129.
39. **Taner, T.M.,** 1997, Kohonen's Self-Organizing networks with "Conscience", *Rock Solid Images*, 1-7.
40. **Lindsey, C.L., Strömberg, M.,** 2000, Image classification using the frequencies of simple features, *Pattern Recognition Letters*, **21**, 265-268.
41. **Kurdthongmee, W.,** 2008, Colour classification of rubberwood boards for fingerjoint manufacturing using a SOM neural network and image processing, *Computer and Electronics in Agriculture*, **64**, 85-92.
42. **Ghaseminezhad, M.H., Karami, A.,** 2011, A novel self-organizing map (SOM) neural network for discrete groups of data clustering, *Applied Soft Computing*, **11**, 3771-3778.

43. **Kohonen, T.**, 1990, The Self-Organizing map, *Proceedings of the IEEE*, **78**, 1464-1480.
44. **Kohonen, T.**, 1998, The Self-Organizing map, *Neurocomputing*, **21**, 1-6.
45. **Kohonen, T., Oja, E., Simula, O., Visa, A., Kangas, J.**, 1996, Engineering applications of the self-organizing map, *Proceedings of the IEEE*, **84**, 1358-1384.

EK – 1: Optimizasyon Programı

```
clear all
clc
rb=0.01;
db=0.05;
rp=0.01;
dp=0.01;
l1=0.065;
l2=0.01;
d=[rb;db;rp;dp;l1;l2];
k=1;
a0=[0.054;0.015;0.04;0.035;0.065;0.2];
options = optimset;
options = optimset(options,'Display', 'off');
options = optimset(options,'MaxIter', 400);
options = optimset(options,'Algorithm', 'sqp');
for t2=-pi/4:0.4363:pi/6
    for t3=-pi/4:0.4363:pi/6
        for t4=-pi/4:0.4363:pi/6
            d=fmincon(@(d) deltaAmac(d), a0,-
1*eye(6),zeros(6,1),[],[],[0.054;0.015;0.04;0.035;0.065;0],[0.054;0.015;0
.04;0.035;0.5;0.5],@(d) mycon(d,t2,t3,t4),options);
            sonuc(:,k)=d;
            k=k+1;
        end
    end
end
figure
hold off
plot(sonuc(5,:), 'r-*)
hold on
plot(sonuc(6,:), 'b-s')
figure
plot(sonuc(1,:), 'r')
hold on
plot(sonuc(2,:), 'g')
plot(sonuc(3,:), 'b')
plot(sonuc(4,:), 'black')
```

EK – 2: Amaç Fonksiyonu

```
function f=deltaAmac(a)
C=zeros(1,3); S=zeros(1,3);
X=zeros(1,40);
Y=zeros(1,40);
Z=zeros(1,40);
i=2;
X(1)=0; Y(1)=0; Z(1)=0;
boy=0;
rb=a(1); rp=a(3); l1=a(5); l2=a(6); db=a(2); dp=a(4);

for p=-pi/4:1.3089:pi/6
    C(1)=cos(p);
    S(1)=sin(p);
    for q=-pi/4:1.3089:pi/6
        C(2)=cos(q);
        S(2)=sin(q);
        for r=-pi/4:1.3089:pi/6
            C(3)=cos(r);
            S(3)=sin(r);
            x=[rb-rp+l1*C(1);
                -0.5*(rb-rp+l1*C(2));
                -0.5*(rb-rp+l1*C(3))];
            y=[0;
                -(sqrt(3)/2)*(rb-rp+l1*C(2));
                (sqrt(3)/2)*(rb-rp+l1*C(3))];
            z=[db+l1*S(1);
                db+l1*S(2);
                db+l1*S(3)];
            w=[x(1)^2+y(1)^2+z(1)^2;
                x(2)^2+y(2)^2+z(2)^2;
                x(3)^2+y(3)^2+z(3)^2];
            g=(y(3)-y(1))*(x(2)-x(1))-(y(2)-y(1))*(x(3)-x(1));
            h=[(z(2)-z(1))*(y(3)-y(1))-(z(3)-z(1))*(y(2)-y(1));
                (z(2)-z(1))*(x(3)-x(1))-(z(3)-z(1))*(x(2)-x(1))];
            j=[((w(2)-w(1))*(y(3)-y(1))-(w(3)-w(1))*(y(2)-y(1)))/2;
                ((w(2)-w(1))*(x(3)-x(1))-(w(3)-w(1))*(x(2)-x(1)))/2];
            A=(h(1)^2/g^2 + h(2)^2/g^2 + 1);
            B=((2*h(1)*x(1))/g - 2*z(1) - (2*h(2)*y(1))/g -
                (2*h(1)*j(1))/g^2 - (2*h(2)*j(2))/g^2);
            CC=x(1)^2 - l2^2 + y(1)^2 + z(1)^2 + j(1)^2/g^2 + j(2)^2/g^2
            - (2*j(1)*x(1))/g + (2*j(2)*y(1))/g;
            delta=B^2-4*A*CC;
            zi=[(-B+sqrt(delta))/(2*A);
                (-B-sqrt(delta))/(2*A)];
            zii=min(zi);
            X(i)=(j(1)-zii*h(1))/g;
            Y(i)=(j(2)-zii*h(2))/(-g);
            Z(i)=zii-dp;
            i=i+1;
            boy=boy+sqrt((X(i)-X(i-1))^2+(Y(i)-Y(i-1))^2+(Z(i)-Z(i-
1))^2);
        end
    end
end
f=-boy;
end
```


EK – 3: Sınırlayıcılar

```
function [c,ceq] = mycon(a,t2,t3,t4)

Rb=a(1);
Db=a(2);
Rp=a(3);
Dp=a(4);
L=[a(5) a(6)];
i=1;jjj=1;
n=3;
Betai=-pi/2;
p=f_ilKin([t2 t3 t4],[Rb Db Rp Dp L(1) L(2)]);
P=[p(1) p(2) p(3) 0];
Pii=[Rp 0 Dp 1]';
Teta=[t2 t3 t4];
Bii=[Rb 0 Db 1]';

for i=1:n;
    Pi=P'+dondur('z',(i-1)*2*pi/n)*Pii;
    Piii=dondur('z',(i-1)*2*pi/n)*Pii;
    Bi=dondur('z',(i-1)*2*pi/n)*Bii;
    AA(i,:)=[Piii(3)*(Bi(2)-Pi(2)+L(1)*cos(Betai+(i-
1)*2*pi/n)*cos(Teta(i))),
            Piii(1)*(Bi(3)-Pi(3)+L(1)*sin(Teta(i)))+Piii(3)*(Pi(1)-
Bi(1)+L(1)*sin(Betai+(i-1)*2*pi/n)*cos(Teta(i))),
            -Piii(1)*(Bi(2)-Pi(2)+L(1)*cos(Betai+(i-
1)*2*pi/n)*cos(Teta(i))),
            Pi(1)-Bi(1)+L(1)*sin(Betai+(i-1)*2*pi/n)*cos(Teta(i)),
            Pi(2)-Bi(2)-L(1)*cos(Betai+(i-1)*2*pi/n)*cos(Teta(i)),
            Pi(3)-Bi(3)-L(1)*sin(Teta(i))];
    BB(i,i)=L(1)*sin((i-1)*2*pi/n+Betai)*sin(Teta(i))*(Pi(1)-
Bi(1)+L(1)*sin((i-1)*2*pi/n+Betai)*cos(Teta(i)))-
L(1)*cos(Teta(i))*(Bi(3)-Pi(3)+L(1)*sin(Teta(i)))+L(1)*cos((i-
1)*2*pi/n+Betai)*sin(Teta(i))*(Bi(2) - Pi(2) + L(1)*cos((i-
1)*2*pi/n+Betai)*cos(Teta(i)));
end
J=pinv(AA)*BB;
tork=J'*[0 0 0 0 0 -5]';
c = 0;
ceq = 0.45-abs(tork(1));
```

EK – 4: Görüntü Alıcı

```
vid = videoinput('winvideo', 1, 'YUY2_176x144');  
src = getselectedsource(vid);  
vid.ReturnedColorspace = 'grayscale';  
src.FrameRate = '30.0000';  
I= getsnapshot(vid);
```

EK – 5: Görüntü İşleme Programı

```
imread(resim);
background=imopen(I,strel('disk',15));
I2=I-background;
I3=imadjust(I2);
level=graythresh(I3);
bw=im2bw(I3,level);
bw=imfill(bw,'holes');
bw=bwareaopen(bw,50);
cc=bwconncomp(bw,8);
graindata=regionprops(cc,'Centroid','Orientation');
konum=zeros(length(graindata)+1,4);
konum(:,3)=190;
konum(:,4)=1;
for i=1:length(graindata)
    for j=1:2
        konum(i,j)=graindata(i).Centroid(j);
        konum2(i,j)=konum(i,j);
        if j==1
            konum(i,j)=konum(i,j)*150/176;
        else
            konum(i,j)=konum(i,j)*100/144;
        end
    end
    aci(i)=graindata(i).Orientation;
    konum(i,:)=dondur('z',pi/2)*([75 50 0 2]-konum(i,:))';
end
```

EK – 6: Sınıflandırma Sistemi

```
if sinif==0
    a=imread('egitici.bmp');
    b=regionprops(a, 'Centroid');
    obje=objeler(a);
    net=SOMagi(obje);
    cikti=sim(net,obje);
    [hangisi kacinci]=find(cikti);
    dizi=[hangisi kacinci];
    imshow(a);
    hold on
    for k = 1:numel(b)
        x = b(k).Centroid(1);
        y = b(k).Centroid(2);
        text(x, y, sprintf('%d', dizi(k,1)), 'Color', 'r', 'FontWeight',
'bold');
    end
else
    figure,
    a=imread('siniflandir.bmp');
    imshow(a);
    b=regionprops(a, 'Centroid');
    obje=objeler(a);
    cikti=sim(net,obje);
    [hangisi kacinci]=find(cikti);
    dizi=[hangisi kacinci];
    for k = 1:numel(b)
        x = b(k).Centroid(1);
        y = b(k).Centroid(2);
        text(x, y, sprintf('%d', dizi(k,1)), 'Color', 'r', 'FontWeight',
'bold');
    end
end
```

EK – 7: Öznitelik Çıkarma Programı

```
function sonuc=objeler(a)
b=regionprops(a, 'Orientation', 'BoundingBox', 'PixelIdxList');
for i=1:numel(b)
    kutu(i)=b(i).BoundingBox(1,3);
end
median_boy=median(kutu(:,1));
median_en=median(kutu(:,2));
BOYUT_BOY=round(median_boy);
BOYUT_EN=round(median_en);
for i=1:numel(b)
    c=false(size(a));
    c(b(i,1).PixelIdxList)=true;
    c=imcrop(c,b(i).BoundingBox);
    c=imrotate(c,-b(i).Orientation);
    d=regionprops(c, 'BoundingBox');
    e=imcrop(c,d(1).BoundingBox);
    e=imresize(e,[BOYUT_BOY BOYUT_EN]);
    d=regionprops(e, 'Area', 'Eccentricity', 'MajorAxisLength',
'MinorAxisLength');
    objeler(:,i)=[d(1).Area; d(1).Eccentricity; d(1).MajorAxisLength;
d(1).MinorAxisLength];
end
objeler=1000.*(objeler./max(objeler(:)));
sonuc=objeler;
end
```

EK – 8: Denetim Sistemi Bilgisayar Kısım

```
clear all
clc

ser = serial('COM4','BaudRate',9600,'DataBits',8);
fopen(ser);
r=[54;15;40;35;65;200];
vid = videoinput('winvideo', 1, 'YUY2_176x144');
src = getselectedsource(vid);
vid.ReturnedColorspace = 'grayscale';
src.FrameRate = '30.0000';
t1=1

while(1)

while(t1==1)
t1=0;
I= getsnapshot(vid);
background=imopen(I,strel('disk',15));
I2=I-background;
I3=imadjust(I2);
level=graythresh(I3);
bw=im2bw(I3,level);
bw=imfill(bw,'holes');
bw=bwareaopen(bw,50);
cc=bwconncomp(bw,8);
graindata=regionprops(cc,'Centroid','Orientation');
konum=zeros(length(graindata)+1,4);
konum(:,3)=190;
konum(:,4)=1;
for i=1:length(graindata)
    for j=1:2
        konum(i,j)=graindata(i).Centroid(j);
        konum2(i,j)=konum(i,j);
        if j==1
            konum(i,j)=konum(i,j)*150/176;
        else
            konum(i,j)=konum(i,j)*100/144;
        end
    end
    aci(i)=graindata(i).Orientation;
    konum(i,:)=dondur('z',pi/2)*([75 50 0 2]-konum(i,:))';
end
imshow(bw);
hold on
stem(konum2(:,1),konum2(:,2),'b+');
pause(0.5);
end
p=konum(:,1:3);
i=1;
joker=zeros(size(p,1)+1,size(p,2));
while(i<=size(p,1))
aci=invKinDelta(p(i,:),r);
aci2=fix(rad2deg(aci));
joker(i+1,:)=aci2;
acilar=fix((joker(i+1,:)-joker(i,:))*50*24/360);
yon=zeros(1,3);
```

```

for j=1:size(p,2)
    if acilar(j)<0
        yon(j)=49;
        acilar=acilar+1;
    else
        yon(j)=48;
        acilar=acilar-1;
    end
end
fprintf(ser,[char(yon(1)) char(abs(acilar(1))) char(yon(2))
char(abs(acilar(2))) char(yon(3)) char(abs(acilar(3)))]);
fprintf(ser, char(13));
idn=fscanf(ser,'%c',1);

if idn=='T'
    i=i+1;
else
    fclose(ser);
    break;
end
t1=1;
end
end
fclose(ser);

```

EK – 9: Denetim Sistemi Mikroişlemi Kısmı

```
#include <16F877.h>
#device adc=8
#FUSES NOWDT, HS, NOPUT, NOPROTECT, NOBROWNOUT
#FUSES NOLVP, NOCPD, NOWRT, NODEBUG
#use delay(clock=20000000)
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7,stream=COM_A)
#byte PORTC=0x07
#byte PORTD=0x08
#byte TRISC=0x87
#byte TRISD=0x88
#bit ENC1=PORTD.0
#bit ENC2=PORTD.1
#bit ENC3=PORTD.2

int16 i=0, j=0, k=0;
int8 m=0, m1=0, m2=0, m3=0;
int8 sabit=0;
char joker[7]={0,0,0,0,0,0,0};
int1 a1=0, a2=0, a3=0;

#INT_RDA
void RDA_isr(void)
{
    joker[m]=fgetc(COM_A);
    m++;
}

void main()
{
    setup_adc_ports(NO_ANALOGS);
    setup_adc(ADC_OFF);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_ccp1(CCP_OFF);
    setup_ccp2(CCP_OFF);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    TRISC=0b11000000;
    PORTC=0x00;
    TRISD=0b11111111;
    PORTD=0x00;
    while(1)
    {
        while(m>5)
        {
            a1=0; a2=0; a3=0;
            if(joker[0]==0x31)
                m1=1;
            else
                m1=2;
            if(joker[2]==0x31)
                m2=4;
            else
                m2=8;
        }
    }
}
```



```

        if(joker[4]==0x31)
            m3=16;
        else
            m3=32;
i=0; j=0; k=0;
while(1)
{
sabit=0; a1=ENC1; a2=ENC2; a3=ENC3;
    if(i<joker[1])
        sabit=m1;
    if(j<joker[3])
        sabit+=m2;
    if(k<joker[5])
        sabit+=m3;
    PORTC=sabit;
delay_us(540);
i+=(ENC1!=a1); j+=(ENC2!=a2); k+=(ENC3!=a3);
    PORTC=0; PORTD=0;
if(i>=joker[1] && j>=joker[3] && k>=joker[5])
    {
        PORTC=0; m=0; putc('T'); break;
    }
}
}
}
}

```

EK – 10: İleri Kinematik Programı

```
function sonuc=f_ilKin(Teta,L)
rb=L(1);
db=L(2);
rp=L(3);
dp=L(4);
l1=L(5);
l2=L(6);
R=[rb db rp dp l1 l2];
C=[cos(Teta(1)) cos(Teta(2)) cos(Teta(3))];
S=[sin(Teta(1)) sin(Teta(2)) sin(Teta(3))];
x=[rb-rp+l1*C(1);
   -0.5*(rb-rp+l1*C(2));
   -0.5*(rb-rp+l1*C(3))];
y=[0;
   -(sqrt(3)/2)*(rb-rp+l1*C(2));
   (sqrt(3)/2)*(rb-rp+l1*C(3))];
z=[db+l1*S(1);
   db+l1*S(2);
   db+l1*S(3)];
w=[x(1)^2+y(1)^2+z(1)^2;
   x(2)^2+y(2)^2+z(2)^2;
   x(3)^2+y(3)^2+z(3)^2];
g=(y(3)-y(1))*(x(2)-x(1))-(y(2)-y(1))*(x(3)-x(1));
h=[(z(2)-z(1))*(y(3)-y(1))-(z(3)-z(1))*(y(2)-y(1));
   (z(2)-z(1))*(x(3)-x(1))-(z(3)-z(1))*(x(2)-x(1))];
j=[((w(2)-w(1))*(y(3)-y(1))-(w(3)-w(1))*(y(2)-y(1)))/2;
   ((w(2)-w(1))*(x(3)-x(1))-(w(3)-w(1))*(x(2)-x(1)))/2];
A=(h(1)^2/g^2 + h(2)^2/g^2 + 1);
B=((2*h(1)*x(1))/g - 2*z(1) +(2*h(1)*j(1))/g^2 + (2*h(2)*j(2))/g^2);
CC=x(1)^2 - l2^2 + z(1)^2 + j(1)^2/g^2 + j(2)^2/g^2 +
(2*j(1)*x(1))/g;
delta=B^2-4*A*CC;
zi=[(-B+sqrt(delta))/(2*A);
   (-B-sqrt(delta))/(2*A)];
zii=min(zi);
X=(j(1)-zii*h(1))/g;
Y=(j(2)-zii*h(2))/(-g);
Z=zii-dp;
sonuc=[X Y Z];
end
```

EK – 11: Ters Kinematik Programı

```
function sonuc=f_tKin(p,r)
Rb=r(1);
Db=r(2);
Rp=r(3);
Dp=r(4);
L=[r(5) r(6)];
n=3;
Betail=-pi/2;
Teta=zeros(3,3);
    P=[p(1) p(2) p(3) 1];
    Pii=[Rp P(2) Dp 1]';
    Bii=[Rb 0 Db 1]';
    for i=1:n;
        Pi=P'+dondur('z',(i-1)*2*pi/n)*Pii;
        Bi=dondur('z',(i-1)*2*pi/n)*Bii;
        Ei=(Pi(2)-Bi(2))*cos((i-1)*2*pi/n+Betail)-(Pi(1)-Bi(1))*sin((i-
1)*2*pi/n+Betail);
        Fi=Pi(3)-Bi(3);
        Gi=((Pi(1)-Bi(1))^2+(Pi(2)-Bi(2))^2+(Pi(3)-Bi(3))^2+L(1,1)^2-
L(1,2)^2)/(2*L(1,1));
        Ki=1;
        Hi=Ei^2+Fi^2-Gi^2;
        sinTi=(Fi*Gi+Ki*Ei*sqrt(Hi))/(Ei^2+Fi^2);
        cosTi=(Ei*Gi-Ki*Fi*sqrt(Hi))/(Ei^2+Fi^2);

        S4=p(2)/(L(2));
        C4=sqrt(1-(p(2)^2/L(2)^2));

        S3=(cosTi*(p(3)-Db+Dp)+sinTi*(Rb-p(1)-Rp));
        C3=(cosTi*(p(1)-Rb+Rp)+sinTi*(p(3)-Db+Dp)-L(1));

        Teta(i,:)=[atan2(sinTi,cosTi) atan2(S3,C3) atan2(S4,C4)];
    end
    sonuc=Teta;
end
```

EK – 12: Jakobiyen Programı

```
function [tork, hiz]=f_Jako(p,Teta,r)
Rb=r(1);
Db=r(2);
Rp=r(3);
Dp=r(4);
L=[r(5) r(6)];
l1=L(1);
l2=L(2);
R=[Rb Db Rp Dp l1 l2];
i=1;jjj=1;
n=3;
Betai=-pi/2;
P=[p(1) p(2) p(3) 0];
Pii=[Rp 0 Dp 1]';
Bii=[Rb 0 Db 1]';
for i=1:n;
    Pi=P'+dondur('z',(i-1)*2*pi/n)*Pii;
    Piii=dondur('z',(i-1)*2*pi/n)*Pii;
    Bi=dondur('z',(i-1)*2*pi/n)*Bii;
    AA(i,:)= [Piii(3)*(Bi(2)-Pi(2)+L(1)*cos(Betai+(i-
1)*2*pi/n)*cos(Teta(i))),
        Piii(1)*(Bi(3)-Pi(3)+L(1)*sin(Teta(i)))+Piii(3)*(Pi(1)-
Bi(1)+L(1)*sin(Betai+(i-1)*2*pi/n)*cos(Teta(i))),
        -Piii(1)*(Bi(2)-Pi(2)+L(1)*cos(Betai+(i-
1)*2*pi/n)*cos(Teta(i))),
        Pi(1)-Bi(1)+L(1)*sin(Betai+(i-1)*2*pi/n)*cos(Teta(i)),
        Pi(2)-Bi(2)-L(1)*cos(Betai+(i-1)*2*pi/n)*cos(Teta(i)),
        Pi(3)-Bi(3)-L(1)*sin(Teta(i))];
    BB(i,i)=L(1)*sin((i-1)*2*pi/n+Betai)*sin(Teta(i))*(Pi(1)-
Bi(1)+L(1)*sin((i-1)*2*pi/n+Betai)*cos(Teta(i)))-
L(1)*cos(Teta(i))*(Bi(3)-Pi(3)+L(1)*sin(Teta(i)))+L(1)*cos((i-
1)*2*pi/n+Betai)*sin(Teta(i))*(Bi(2) - Pi(2) + L(1)*cos((i-
1)*2*pi/n+Betai)*cos(Teta(i)));
end
J=pinv(AA)*BB;
tork=J*[0 0 0 0 0 -5]';
hiz=J*[9.697 9.697 9.697]';
end
```

EK – 13: Homojen Dönme Matrisi Hesaplama Programı

```
function sonuc=dondur(eksen,x)
if eksen=='x' || eksen=='X'
    sonuc=[1 0 0 0;
          cos(x) -sin(x) 0 0;
          sin(x) cos(x) 0 0;
          0 0 0 1];
elseif eksen=='y' || eksen=='Y'
    sonuc=[cos(x) 0 sin(x) 0;
          0 1 0 0;
          -sin(x) 0 cos(x) 0;
          0 0 0 1];
elseif eksen=='z' || eksen=='Z'
    sonuc=[cos(x) -sin(x) 0 0;sin(x) cos(x) 0 0; 0 0 1 0;0 0 0 1];
end
end
```

EK – 14: Öteleme Matrisi Hesaplama Programı

```
function sonuc=gotur(eksen,x)
if eksen=='x' || eksen=='X'
    sonuc=[1 0 0 x;
           0 1 0 0;
           0 0 1 0;
           0 0 0 1];
elseif eksen=='y' || eksen=='Y'
    sonuc=[1 0 0 0;
           0 1 0 x;
           0 0 1 0;
           0 0 0 1];
elseif eksen=='z' || eksen=='Z'
    sonuc=[1 0 0 0;
           0 1 0 0;
           0 0 1 x;
           0 0 0 1];
end
end
```

EK – 15: Robot Çizdirme Programı

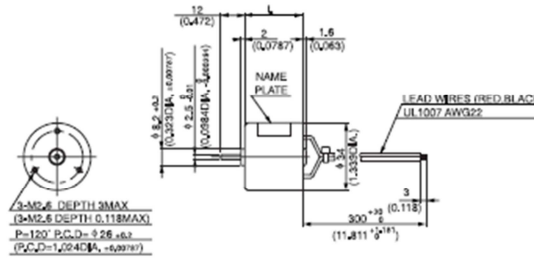
```
function cizDelta(Teta,p,r)
    Rb=r(1);
    Db=r(2);
    Rp=r(3);
    Dp=r(4);
    L=[r(5) r(6)];
    n=3;
    Betai=-pi/2;
    Bussu=zeros(n,3);
    xler=zeros(n,4);
    yler=xler;
    zler=xler;
for j=1:size(Teta,1)
    Bi=[Rb 0 Db 1]';
    P=[p(j,1) p(j,2) p(j,3) 1];
    Pi=[P(1)+Rp P(2) P(3)+Dp 1]';
    for i=1:n
        if i==1
            Ti=0;
        else
            Ti=2*pi/n;
        end
        Pi=dondur('z',Ti)*Pi;
        Bi=dondur('z',Ti)*Bi;
        Bussu(i,1)=Bi(1)-L(1)*sin((i-1)*2*pi/n+Betai)*cos(Teta(j,i));
        Bussu(i,2)=Bi(2)+L(1)*cos((i-1)*2*pi/n+Betai)*cos(Teta(j,i));
        Bussu(i,3)=Bi(3)+L(1)*sin(Teta(j,i));
        xler(i,:)=[Bi(1) Bussu(i,1) Pi(1) P(1)];
        yler(i,:)=[Bi(2) Bussu(i,2) Pi(2) P(2)];
        zler(i,:)=[Bi(3) Bussu(i,3) Pi(3) P(3)];
    end
    hold off
    for i=1:n
        plot3(xler(i,:),yler(i,:),zler(i,:));
        view([45 10]);
        hold on
    end
    pause(0.5);
end
end
```

EK – 16: Motor Özellikleri

DME34



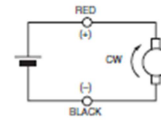
● DIMENSIONS Unit mm(inch)



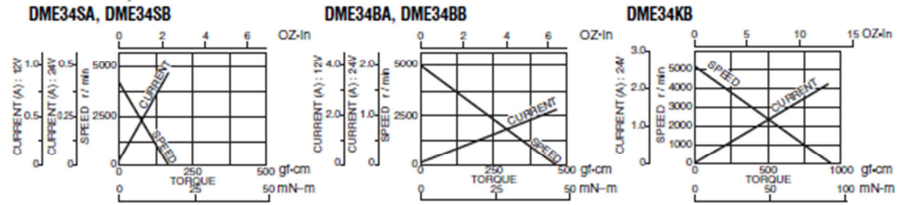
MODEL CODE	VOLTAGE	OUTPUT	CURRENT
SA	12V	1.3W	0.2A
SB	24V	1.3W	0.1A
BA	12V	4.5W	0.65A
BB	24V	4.5W	0.31A
KB	24V	7W	0.41A

Model	L	Weight	
		g	lb
DME34SA DME34SB	29.5	100	0.22
DME34BA DME34BB	35.0	110	0.24
DME34KB	45	140	0.31

● CONNECTION



● CURRENT, SPEED-TORQUE CURVE



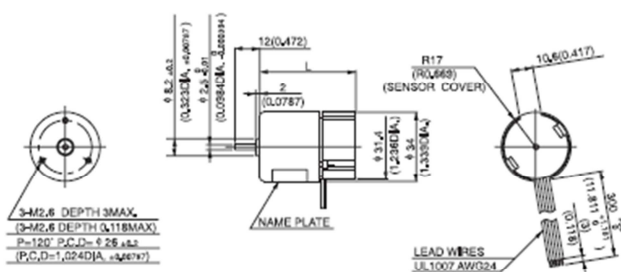
● STANDARD SPECIFICATIONS

Model	Rated				No load		Stall torque			
	Output W	Voltage V	Torque		Current A	Speed r/min	Current A	Speed r/min	mN-m	oz-in
			mN-m	oz-in						
DME34SA	1.3	12	3.9	0.56	0.2	3300	0.04	4300	17	2.36
DME34SB	1.3	24	3.9	0.56	0.1	3300	0.02	4300	17	2.36
DME34BA	4.5	12	11.8	1.67	0.65	3700	0.07	5000	45	6.39
DME34BB	4.5	24	11.8	1.67	0.31	3700	0.04	5000	45	6.39
DME34KB	7	24	14.7	2.08	0.41	4300	0.06	5100	92	13.03

DME34

● REVOLUTION SENSOR OPTICAL TYPE

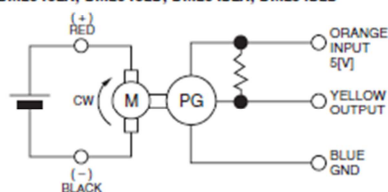
MODEL CODE	VOLTAGE	OUTPUT CURRENT	CURRENT
SA	12V	1.3W	0.2A
SB	24V	1.3W	0.1A
BA	12V	4.5W	0.65A
BB	24V	4.5W	0.31A
KB	24V	7W	0.41A



Model	L	Weight	
		g	lb
DME34SEA DME34SEB	47.1	120	0.26
DME34BEA DME34BEB	52.6	130	0.29
DME34KEB	62.6	160	0.35

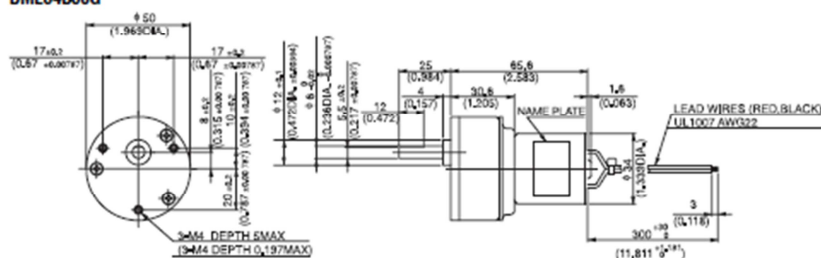
● CONNECTION OF REVOLUTION SENSOR

DME34SMA, DME34SMB, DME34BMA, DME34BMB
DME34SEA, DME34SEB, DME34BEA, DME34BEB



● DIMENSIONS Unit mm(inch)

DME34B50G



(WEIGHT 310g 0.68lb)

● with 50G TYPE GEARBOX

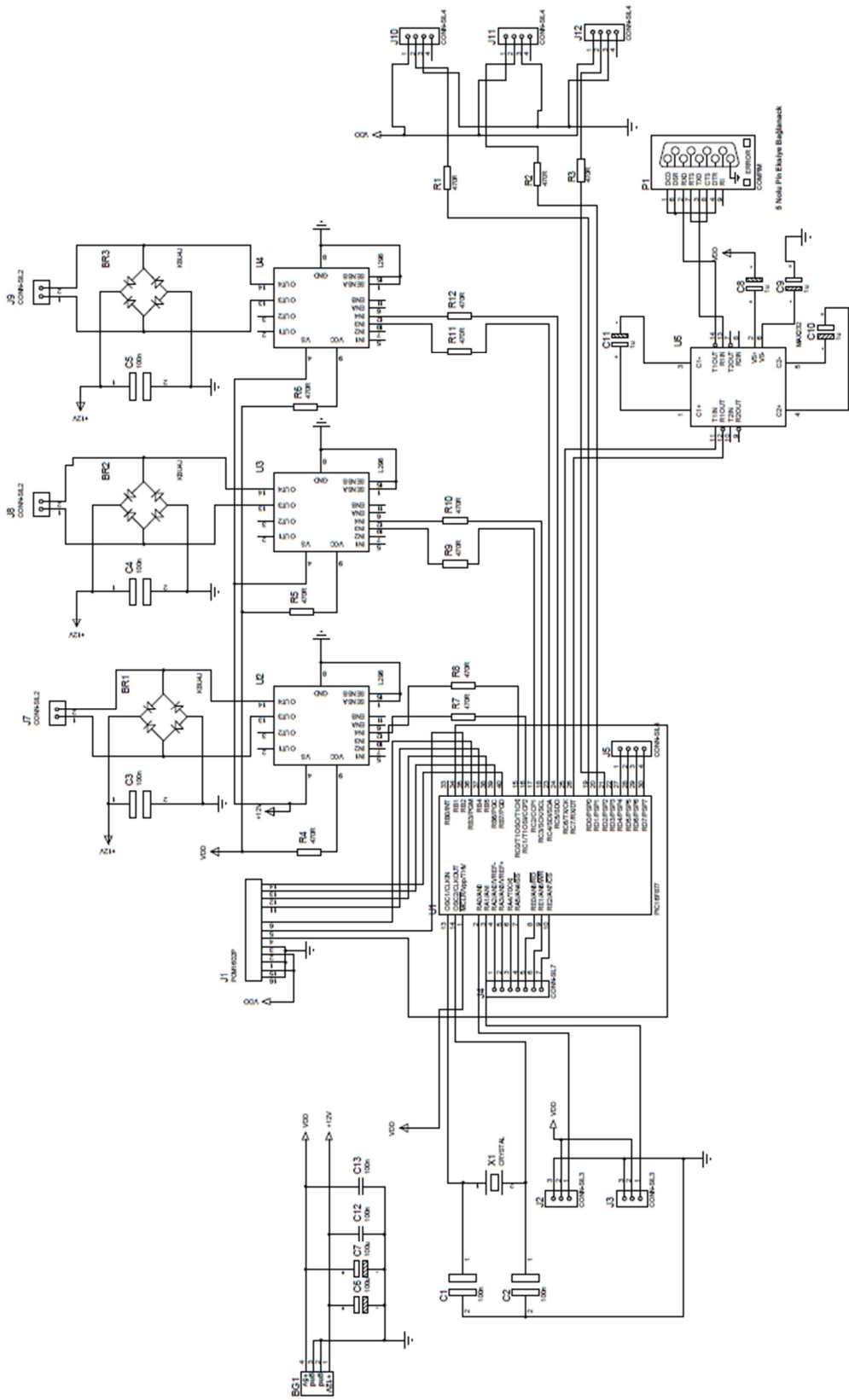
Model	Gear ratio	Rated speed	t/min									
			9	18	*27	*36	*54	*72	96	144	192	256
DME34B50G	Rated torque	N-m	0.085	0.17	0.23	0.3	0.46	0.62	0.74	0.98	0.98	0.98
			oz-in	12.08	23.61	31.94	43.05	65.27	87.49	104.15	138.87	138.87

WITH GEARBOX
50G



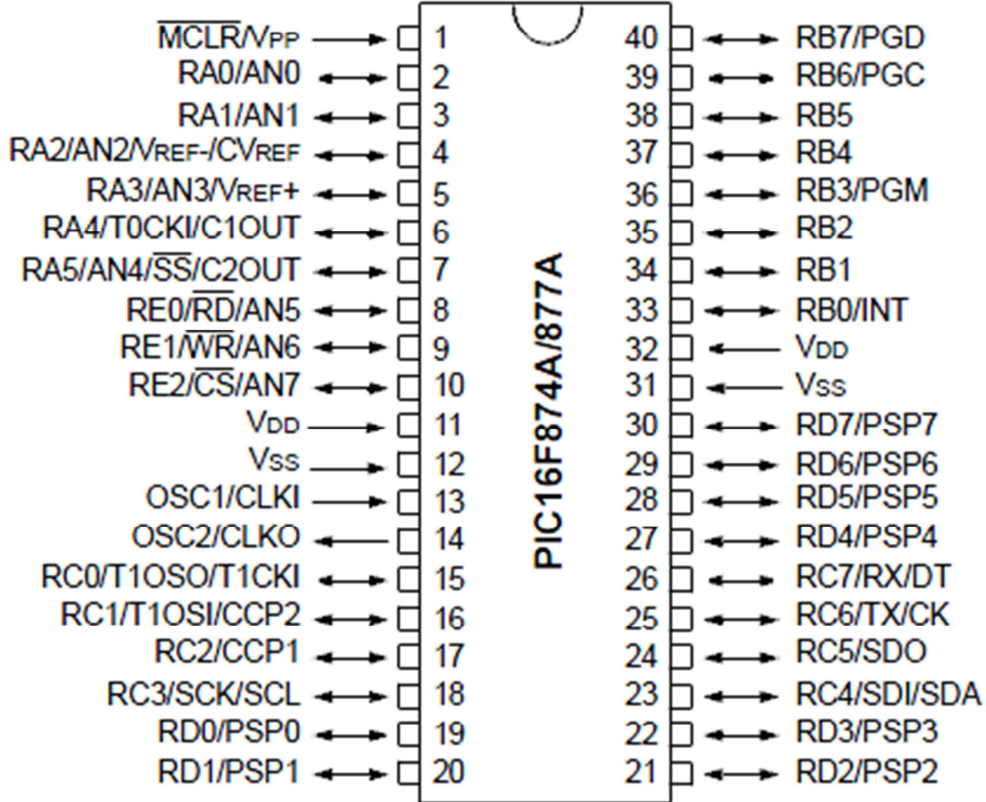
50G

EK – 17: Elektronik Devre



EK – 18: PIC16F877 Bacak Bağlantıları

40-Pin PDIP



ÖZGEÇMİŞ

Kişisel Bilgiler		
Ad Soyad	İsmail H. ŞANLITÜRK	
Adres	Fırat Üniversitesi Mühendislik Fakültesi Makine Mühendisliği ELAZIĞ	
Telefon	0424 237 00 00 - 5328	
E-Posta	ihsanliturk@firat.edu.tr	
Eğitim Bilgileri		
Lisans	Fırat Üniversitesi Mühendislik Fakültesi Makine Mühendisliği 2008	
İş Tecrübesi		
Fırat Üniversitesi Elazığ	Araştırma Görevlisi	2009-
ADEKA A.Ş. Samsun	Bakım Onarım Mühendisi	2008-2009
Alptekin A.Ş. Malatya	CAD Mühendisi	2008
Aka Kırtasiye Ltd. Samsun	Departman Sorumlusu	2001-2003