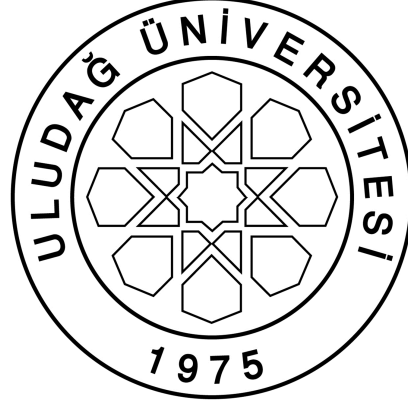


**MEKATRONİK BİR SİSTEMİN MODELLENMESİ,  
BENZETİMİ, GERÇEKLENMESİ VE GERÇEK  
ZAMANLI DENETİMİ**

**Ahmet DEMİRKESEN**



T.C.  
ULUDAĞ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**MEKTRONİK BİR SİSTEMİN MODELLENMESİ, BENZETİMİ,  
GERÇEKLENMESİ VE GERÇEK ZAMANLI DENETİMİ**

**Ahmet Demirkesen**

Prof. Dr. İbrahim YÜKSEL

(Danışman)

YÜKSEK LİSANS TEZİ  
MAKİNE MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2012

**Her Hakkı Saklıdır**

## TEZ ONAYI

Ahmet Demirkeseu tarafından hazırlanan ‘‘Mekatronik Bir Sistemin Modellenmesi, Benzetimi, Gereklenmesi ve Gerek Zamanlı Denetimi’’ adlı tez alıřması ařađıdaki juri tarafından oy birliđi/oy okluđu ile Uludađ Üniversitesi Fen Bilimleri Enstitüsü Makine Mühendisliđi Anabilim Dalı’nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Danışman** Prof. Dr. İbrahim Yüksel

**Başkan:** Prof. Dr. İbrahim YÜKSEL İmza  
Uludađ Üniversitesi Mühendislik-Mimarlık Fakültesi  
Makine Mühendisliđi Anabilim Dalı

**Üye :** Yrd. Do. Dr. Gürsel ŞEFKAT İmza  
Uludađ Üniversitesi Mühendislik-Mimarlık Fakültesi  
Makine Mühendisliđi Anabilim Dalı

**Üye :** Yrd. Do. Dr. Zeliha KAMIŞ KOCABIAK İmza  
Uludađ Üniversitesi Mühendislik-Mimarlık Fakültesi  
Otomotiv Mühendisliđi Anabilim Dalı

**Yukarıdaki Sonucu Onaylarım**

**Prof. Dr. Kadri ARSLAN**  
**Enstitü Müdürü**

..../..../.....

**U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;**

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

**beyan ederim.**

**07.08.2012**

**Ahmet Demirkesen**



## ÖZET

Yüksek Lisans Tezi

### MEKATRONİK BİR SİSTEMİN MODELLENMESİ, BENZETİMİ, GERÇEKLENMESİ VE GERÇEK ZAMANLI DENETİMİ

**Ahmet DEMİRKESEN**

Uludağ Üniversitesi

Fen Bilimleri Enstitüsü

Makine Mühendisliği Anabilim Dalı

**Danışman:** Prof. Dr. İbrahim YÜKSEL

Bu tez çalışmasında, model-tabanlı tasarım ile mekatronik sistem tasarımı süreçlerini kapsayan örnek bir mekatronik sistem geliştirilmiştir.

“Nesne Takip Sistemi” olarak isimlendirilen bu sistem, belirli bir renge sahip bir nesneyi görme alanı içerisinde algılayabilen, kameranın açısal hareketini denetleyerek nesneyi görüntü içerisinde merkezleyen bir servo sistemdir.

Model-tabanlı tasarımın, denetim sistemlerinin tasarımına sağladığı kolaylıklardan faydalanarak, bu sistem için kararlı gömülü algoritmalar geliştirilmiş, hem benzetim ortamında hem de deney düzeneği üzerinde gerçekleştirilmiştir. Geliştirilen sistemin nesne-takibi performansı test edilmiş ve sonuçlar tartışılmıştır. Tez çalışması sonucunda, benzer uygulamalara faydalı olabilecek sonuçlar çıkarılmıştır.

**Anahtar Kelimeler:** Makine ile görme, görüntü tabanlı görsel servolar, gömülü denetleyiciler, hızlı denetim prototipleme, gerçek-zamanlı sınamalar, model-tabanlı tasarım

**2012, xiv + 127 sayfa.**

## **ABSTRACT**

MSc Thesis

### **MODELING, SIMULATION, IMPLEMENTATION OF A MECHATRONIC SYSTEM AND ITS REAL-TIME CONTROL**

**Ahmet DEMİRKESEN**

Uludag University

Graduate School of Natural and Applied Sciences

Department of Mechanical Engineering

**Supervisor:** Prof. Dr. İbrahim YÜKSEL

In this thesis, a study case which covers stages of mechatronic system development with model based design was developed.

The system, that is called "object tracking system", is an image-based visual servo which can detect objects that have a specified color and area in field of view and control angular movement of camera to adjust position of the object in the center of the image.

By taking advantage of model-based design, stable embedded algorithms were developed and implemented on both simulation and test bench environment. A number of object tracking tests were performed and results were discussed. As a result of this study, facts that could be applicable and are useful for similar systems were presented.

**Key words:** Machine vision, image-based visual servo, embedded controllers, rapid control prototyping, real-time testing, model-based design

**2012, xiv + 127 pages.**

## ÖNSÖZ VE TEŞEKKÜR

Mekatronik, birden fazla mühendislik alanını içeren, sürekli gelişen ve heyecan veren bir mühendislik dalıdır. Bu tezin gelişimini ve yazılmasını kapsayan bir yılı aşkın süre içerisinde, bu genç mühendislik dalının inceliklerini keşfetme şansını buldum.

Tezin içerisine, ülkemizde henüz yaygınlaşmayan yenilikçi teknolojileri olabildiğince dahil etmeye çalıştım. Bundaki amaç, ülkemizde mekatronik üzerine çalışan kurumların (hem ticari hem de akademik), dünyada bu alandaki yeni eğilimlere olan uyumuna katkı sağlamaktır. Ayrıca teze başlamadan önce, kullanılacak teorilerin hayata geçirilmeleri de hedeflenmişti. İlerleyen sayfalar boyunca görüldüğü gibi, teorik esaslarla tasarlanmış mekatronik bir sistem, geliştirme aşamaları süresince deneylerle doğrulanmış, son aşamada “elle tutulur” bir ürün haline gelmiştir.

Uludağ Üniversitesi’ndeki yüksek lisans öğrenimim süresince, emeği geçen tüm öğretim görevlilerine ayrı ayrı teşekkür ederim. Danışmanlığımı üstlenen, sarfettiği emek ve aktardığı deneyimler ile daha iyi bir mühendis olmamı sağlayan Sayın Prof. Dr. İbrahim YÜKSEL’e teşekkür ederim.

**Ahmet Demirkesen**

07.08.2012

## İÇİNDEKİLER

	Sayfa
ÖZET .....	i
ABSTRACT .....	ii
ÖNSÖZ VE TEŞEKKÜR .....	iii
SİMGELER VE KISALTMALAR DİZİNİ.....	vi
ŞEKİLLER DİZİNİ .....	xi
ÇİZELGELER DİZİNİ .....	xiv
1. GİRİŞ .....	1
2. KAYNAK ARAŞTIRMASI .....	4
3. MATERYAL VE YÖNTEM .....	13
3.1. Materyal .....	13
3.1.1. Çalışmada Faydalanılan Disiplinler.....	13
3.1.1.1. Mekatronik Sistemler ve Mekatronik Sistem Tasarımı .....	13
3.1.1.2. Gömülü Denetim Sistemleri.....	14
3.1.1.3. Gömülü Yazılımlar ve V-Modeli .....	15
3.1.1.4. Model Tabanlı Tasarım.....	18
3.1.1.5. Gerçek Zamanlı Test ve Uygulamaları.....	22
3.1.2. Nesne Takip Sistemi .....	26
3.1.3. Nesne Takip Sistemi Deney Düzeneği .....	31
3.1.4. Nesne Takip Sisteminin Bileşenleri ve Özellikleri .....	34
3.1.4.1. DC Motor Grubu .....	34
3.1.4.2. Motor Sürücü Devresi.....	36
3.1.4.3. xPC Target Yazılımı ve Hedef/Yönetici PC .....	38
3.1.4.4. Çok Amaçlı I/O Kartı .....	40
3.1.4.5. Web Kamerası ve Örnek Nesne Fotoğrafları .....	42
3.1.5. Kullanılan Geliştirme Araçları .....	43
3.1.5.1. Sistem Modelleme ve Benzetim Araçları .....	43
3.1.5.2. Denetim Sistemi Tasarımı Araçları .....	45

3.1.5.3.	Görüntü ve Video İşleme Araçları .....	46
3.1.5.4.	Otomatik Kod Oluşturma Araçları .....	46
3.2.	Yöntem.....	48
3.2.1.	Görme Altsistemi.....	49
3.2.2.	Hareket Altsistemi .....	57
3.2.2.1.	Denetlenen Sistemin (Elektromekanik sistem) Modellenmesi .....	60
3.2.2.2.	DC Motorun Hız Denetleyicisinin Tasarımı .....	77
3.2.2.3.	DC Motorun Konum Denetleyicisinin Tasarımı .....	96
3.2.3.	Durum Geçişleri Denetimi Altsistemi .....	99
4.	BULGULAR VE TARTIŞMA .....	103
4.1.	Takip Performansı Testleri.....	103
4.1.1.	Hareketsiz Nesne Testleri .....	105
4.1.1.1.	“PI-1x” Hız Denetimi ile Hareketsiz Nesne Takibi Testi.....	107
4.1.1.2.	“PI-4x” Hız Denetimi ile Hareketsiz Nesne Takibi Testi.....	109
4.1.1.3.	Hareketsiz Nesne Takibi Deneyi Sonuçları .....	110
4.1.2.	Hareketli Nesne Testleri .....	111
4.1.2.1.	“PI-1x” Hız Denetimi ile Hareketli Nesne Takibi Deneyi .....	111
4.1.2.2.	“PI-4x” Hız Denetimi ile Hareketli Nesne Takibi Deneyi .....	115
4.2.	Model-Tabanlı Tasarımın Mekatronik Tasarım Süreçlerine Katkısı .....	120
4.3.	İleriye Dönük Çalışmalar .....	121
5.	SONUÇ .....	122
	KAYNAKLAR.....	125
	ÖZGEÇMİŞ.....	127

## SİMGELER VE KISALTMALAR DİZİNİ

### Simgeler

Simge	Açıklamalar
$m$	Piksel cinsinden görüntü genişliği
$n$	Piksel cinsinden görüntü yüksekliği
$r$	Bir matrisin herhangi bir satırı
$c$	Bir matrisin herhangi bir sütunu
$R$	Kırmızı görüntü kanalı matrisi
$G$	Yeşil görüntü kanalı matrisi
$G^*$	Belirginleştirilmiş yeşil görüntü kanalı matrisi
$G_{BW}$	Yeşil kanala ilişkin eşik değerlendirilmiş siyah-beyaz görüntü matrisi
$K_T$	Görüntüye ilişkin eşik değer
$B$	Mavi görüntü kanalı matrisi
$K_G$	Yeşil kanalını belirginleştirme ölçüğü
$v$	1 yada 0 değerini alabilen keyfi bir değer
$R$	Görüntüye ilişkin ROI matrisi
$\bar{r}$	R matrisinin ağırlık merkezinin satır bileşeni
$\bar{c}$	R matrisinin ağırlık merkezinin sütun bileşeni
$e_{px}$	Blob merkezi ile görüntü merkezi arasındaki konum farkı (piksel cinsinden)
$\Delta\bar{c}$	Konum hatası toleransı (piksel cinsinden)
$A$	Görüntüye ilişkin ROI alanı
$G_L$	Etiketlendirilmiş yeşil görüntü kanalı matrisi
$W_x$	Yatay eksendeki piksel cinsinden görüntü genişliği
$t$	Zaman, s
$T_s$	Örnekleme Zamanı, s
$T_{sc}$	Gözlem zaman-dilimi, s
$\Delta\omega$	Açısal hıza ilişkin niceleme hatası, rad/s
$N_p$	Bir encoderin bir tur başına ürettiği darbe sayısı, darbe/s

$\Delta N$	Bir $T_{sc}$ zaman-dilimi içerisinde gözlemlenen darbe sayısı
$i_a(t)$	Armatür akımı, $A$
$\theta_M$	Motor milinin açısai konumu, $rad$
$\Theta_M(s)$	Motor milinin açısai konumu (Laplace dönüşümü alınmış)
$\dot{\theta}_M = \frac{d\theta_M}{dt}$	Motor milinin açısai hızı, $rad/s$
$\ddot{\theta}_M = \frac{d^2\theta_M}{dt^2}$	Motor milinin açısai ivmesi, $rad/s^2$
$\theta_L, \theta$	Yük milinin açısai konumu, $rad$
$\Theta(s)$	Yük milinin açısai konumu (Laplace dönüşümü alınmış)
$\omega = \dot{\theta}_L = \frac{d\theta_L}{dt}$	Yük milinin açısai hızı, $rad/s$
$\Omega(s)$	Yük milinin açısai hızı ((Laplace dönüşümü alınmış)
$\ddot{\theta}_L = \frac{d^2\theta_L}{dt^2}$	Yük milinin açısai ivmesi, $rad/s^2$
$L_a$	Armatür Endüktansı, $H$
$R_a$	Armatür Direnci, $Ohm$
$K_b$	Zıt EMF katsayısı, $volt \cdot s/rad$
$K_t$	EMC Tork dönüşüm katsayısı, $Nm/A$
$\omega_M$	Motor milinin açısai hızı, $rad/s$
$M_M$	Motorun ürettiği moment, $Nm$
$M_1$	Küçük dişliye aktarılan moment, $Nm$
$M_2$	Büyük dişliye aktarılan moment, $Nm$
$B_M$	Motor milinin viskoz sürtünme katsayısı, $Nm/(rad/s)$
$B_L$	Yük milinin viskoz sürtünme katsayısı, $Nm/(rad/s)$
$J_M$	DC motor ve motor miline bağlı dişlinin eylemsizlik momenti, $kgm^2$
$J_L$	Yük ve yük miline bağlı dişlinin eylemsizlik momenti, $kgm^2$
$J_{eq}$	Motor-dişli kutusunun bileşke eylemsizlik momenti, $kgm^2$
$B_{eq}$	Motor-dişli kutusunun bileşke viskoz sürtünme kat sayısı, $Nm/(rad/s)$
$T_f$	Sürtünme torku, $Nm$

$\frac{N_2}{N_1} = n$	Dişli kutusu indirgeme oranı,
$p_1, p_2, \dots, p_n$	Bir sistemin kutupları
$c(t)$	Denetleyicinin çıkışı
$C(s)$	Denetleyicinin çıkışı (Laplace dönüşümü alınmış)
<i>duty</i>	PWM sinyaline ilişkin görev çevrimi (doluluk oranı)
<i>direction</i>	Yön sinyali
$K_{supply}$	Besleme gerilimi
$K_p$	Oransal Kazanç
$K_i$	İntegral Kazanç
$K_d$	Türev Kazancı
$\omega_n$	Doğal frekans
$K_{sys}$	Sistemin kazancı
$\tau$	Sistemin zaman sabiti
$GH(s)$	Sistemin açık döngü transfer fonksiyonu
$e_{ss}$	Sistemin kalıcı durum hatası



## Kısaltmalar

Kısaltma	Açıklamalar
TTL	Transistor-to-transistor Logic. Transistör-Transistör mantığı
CMOS	Complementary Metal-oxide-semiconductor. Tümler metal-oksit yarı-iletken
BJT	Bipolar junction transistör. İki kutuplu bağlantı transistörü
PWM	Pulse width modulation. Darbe genişlik modülasyonu
D/A	Digital/Analog. Sayısal/Analog
A/D	Analog/Digital. Analog/Sayısal
DC	Direct Current. Doğru Akım
EMF	Electromotor force. Elektromotor kuvvet.
FPGA	Field Programmable Gate Arrays. Alan programlanabilir kapılar dizisi
DSP	Digital Signal Processor. Sayısal İşaret İşleyici
CCD	Charge-coupled device. Dolgu-çiftli aygıt.
SLAM	Static Look-and-Move. Durgun izle ve hareket et.
LTI	Linear Time Invariant. Doğrusal ve Zamandan Bağımsız
RCP	Rapid Control Prototyping. Hızlı Kontrol Prototipleme
HIL	Hardware-in-the-Loop. Döngüde Donanım
PIL	Processor-in-the-Loop. Döngüde İşlemci
SIL	Software-in-the-Loop. Döngüde Yazılım
CAN	Control Area Network. Denetim Alanı Ağı
ECU	Electronical Control Unit. Elektronik denetim birimi
FDIR	Fault Detection Isolation and Recovery. Arıza Algılama Ayırma ve Kurtarma
EML	Embedded MATLAB Language
PCB	Printed circuit board. Baskı devre kartı
I/O	Input/Output. Giriş/Çıkış
PID	Proportional Integral Derivative. Oransal İntegral Türev
LED	Light-Emmiting Diot. Işık yayan diyot
3B	3 boyutlu

2B	2 boyutlu
PCI	Peripheral Component Interconnect
GND	Ground. Toprak
VIN	Input Voltage. Giriş Gerilimi
DIRA	Direction A. A yönü
DIRB	Direction B. B yönü
PC	Personal Computer. Kişisel bilgisayar
TCP/IP	Transmission Control Protocol/Internet Protocol. Gönderim Denetim Protokolü / İnternet Protokolü
USB	Universal Serial Bus. Evrensel Seri Veri yolu
LTI	Linear Time-Invariant. Doğrusal ve zamanla değişmeyen
EML	Embedded MATLAB Language
GUI	Graphical User Interface. Grafiksel Kullanıcı Arayüzü
ROI	Region of Interest. İlgilenilen bölge.

## ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1. V-Modeli İş Akışı .....	17
Şekil 3.2. Model-tabanlı tasarım iş akışı.....	19
Şekil 3.3. Model-tabanlı tasarım ve öğeleri .....	20
Şekil 3.4. Bilgisayar Benzetimleri ile Test ve Doğrulama.....	20
Şekil 3.5. Ayrıntılandırılmış Bir Tasarım .....	21
Şekil 3.6. Gerçek-Zamanlı Test Uygulamaları.....	22
Şekil 3.7. HIL yapılanması.....	24
Şekil 3.8. SIL Yapılanması .....	25
Şekil 3.9. Nesne Takip Sisteminin Yapısal Şeması.....	26
Şekil 3.10. Nesne Takip Sisteminin Fiziksel Bileşenleri .....	27
Şekil 3.11. Makinenin Çalışmasına Bir Örnek.....	28
Şekil 3.12. Görüntü üzerindeki geometrik gösterimler.....	29
Şekil 3.13. Durum geçiş denetimleri .....	29
Şekil 3.14. Gerçek-zamanlı programın çalışma prensibi .....	30
Şekil 3.15. Nesne takip sistemi deney düzeneğinin çalışma prensibi .....	31
Şekil 3.16. Deney düzeneğine ilişkin donanımsal ve yazılımsal bileşenler.....	32
Şekil 3.17. Deney Düzeneğinin Bir Görüntüsü.....	33
Şekil 3.18. Deney düzeneği elektriksel bağlantı şeması .....	33
Şekil 3.19. NXT motorun CAD montaj görüntüsü ve motora tümleşik optik encoder ..	34
Şekil 3.20. Çalışmada kullanılan RJ-12 konektörü .....	35
Şekil 3.21. Genel bir H-Köprüsü devre şeması (Bishop 2007).....	36
Şekil 3.22. Sürücü devre şeması.....	37
Şekil 3.23. Bir L298 tüm devresinin iç şeması.....	38
Şekil 3.24. xPC Target ile Gerçek Zamanlı Test yapılanması .....	39
Şekil 3.25. Çalışmada kullanılan xPC Target hedef bilgisayar .....	40
Şekil 3.26. Humusoft MF624 I/O kartı.....	41
Şekil 3.27. Simulink ve SimScape'in karşılaştırılması.....	44
Şekil 3.28. Gerçek-zamanlı programın Simulink modeli .....	48
Şekil 3.29. Görme Altsisteminin İşlevsel Şeması .....	49
Şekil 3.30. Örnek nesne fotoğrafı.....	50
Şekil 3.31. Özgün Görüntü (Solda) ve Yeşil Kanal (Sağda).....	50
Şekil 3.32. Farklı ölçekleme katsayıları altında yeşil kanal.....	51
Şekil 3.33. Parlak piksele ait okunan değerler .....	52
Şekil 3.34. Seçilen parlak noktanın yatay hizasındaki piksellerin renk şiddetleri .....	53
Şekil 3.35. Özgün resim ve eşik değerleme işlemi sonucu .....	53
Şekil 3.36. Hesaplanan blobun yakından görünümü .....	53
Şekil 3.37. Herhangi bir ikili (binary) görüntüye ait matris ve etiketlenilmiş bağlı bileşenler matrisi (Shapiro 2000) .....	54

Şekil 3.38. Merkez noktası hesaplanmış blob .....	55
Şekil 3.39. Görme altsisteminin testi .....	56
Şekil 3.40. Elektromekanik sisteme ilişkin denetim sisteminin modeli .....	57
Şekil 3.41. Denetleyici alt sistemi .....	58
Şekil 3.42. DC motor sürücüsünün Simulink/SimScape modeli.....	61
Şekil 3.43. DC motor ve dişli grubunun şeması.....	61
Şekil 3.44. DC motor ve dişli grubunun blok diyagramı .....	65
Şekil 3.45. Blok diyagramının indirgenmiş hali.....	65
Şekil 3.46. Elektromekanik altsistemin Simulink Modeli .....	67
Şekil 3.47. DC motor elektriksel kısmın Simulink modeli .....	68
Şekil 3.48. DC motor mekanik kısmın Simulink modeli.....	68
Şekil 3.49. Motorun gerçek zamanlı sınanmasına yönelik Simulink ortamında geliştirilen uygulama.....	69
Şekil 3.50. Parametre kestirimi deneyleri için geliştirilen GUI ve prensip şeması .....	70
Şekil 3.51. Giriş test sinyaline karşılık elektromekanik sistemin yanıtı .....	71
Şekil 3.52. Parametre kestirimi için oluşturulan Simulink modeli.....	71
Şekil 3.53. Optimizasyon öncesi ve sonrası benzetim/gerçek sistem yanıtlarının karşılaştırılması.....	73
Şekil 3.54. Denetlenen sistemin birim basamak yanıtı .....	75
Şekil 3.55. Hız denetim sistemi blok diyagramı.....	77
Şekil 3.56. P denetimli denetim sistemi (doğrusal model) .....	78
Şekil 3.57. P denetimli sistemin açık döngü kök-yer eğrileri grafiği .....	79
Şekil 3.58. P denetleyici ile oluşturulan denetim sisteminin farklı orantı kazançlarına göre birim basamak yanıtı .....	80
Şekil 3.59. PI denetimli denetim sistemi (doğrusal model) .....	82
Şekil 3.60. PI denetimli sistemin (başlangıç ayarları ile) açık-döngü kök-yer eğrileri grafiği .....	84
Şekil 3.61. Başlangıç ayarları ile PI denetimli sistemin birim basamak yanıtı .....	85
Şekil 3.62. Düzenlenmiş ayarlar ile PI denetimli sistemin açık-döngü kök-yer eğrileri grafiği .....	86
Şekil 3.63. Düzenlenmiş ayarlar ile PI denetimli sistemin birim basamak yanıtı .....	87
Şekil 3.64. Çeşitli PI yapılanmaları ile doğrusallaştırılmış denetim sisteminin birim basamak yanıtı.....	89
Şekil 3.65. Çeşitli PI yapılanmaları ile doğrusal-olmayan denetim sisteminin birim basamak yanıtı.....	89
Şekil 3.66. Niceleme hatalarının darbe/devir sayısı ve zaman-dilimi ile ilişkisi (Petrella ve ark. 2007).....	91
Şekil 3.67. Farklı gözlem zaman-dilimlerinde ölçülen hızlar .....	92
Şekil 3.68. "PI 1x" Denetimli Sistemin Birim Basamak Yanıtı (Gerçek Zamanlı Test) 93	
Şekil 3.69. "PI 3x" Denetimli Sistemin Birim Basamak Yanıtı (Gerçek Zamanlı Test) 94	
Şekil 3.70. "PI 4x" Denetimli Sistemin Birim Basamak Yanıtı (Gerçek Zamanlı Test) 94	
Şekil 3.71. "PI 4x" Denetimli Sistemin Birim Basamak Yanıtı (Gerçek Zamanlı Test) 95	

Şekil 3.72. (a) ve (b). Konum denetim sisteminin sınır kararlıdaki yanıtı ve yakından görünümü .....	97
Şekil 3.73. Ayarlanmış katsayılar ile konum denetim sisteminin yanıtı.....	98
Şekil 3.74. Durum geçiş denetimi ve Denetleyici Altsistemleri .....	100
Şekil 3.75. Durum geçiş denetimine ilişkin Stateflow şeması .....	101
Şekil 3.76. Denetleyici Altsistemi .....	102
Şekil 4.1. Deney Izgarası .....	103
Şekil 4.2. Deney masası ve işaretlenmiş noktalar .....	104
Şekil 4.3. Test sehpası ve hareketsiz nesnenin konumu .....	105
Şekil 4.4. Test süresince deney düzeneğinin görünümü .....	106
Şekil 4.5. “PI-1x” hız denetimi ile hareketsiz nesne takibi deneyi ve deneyin <i>zamana bağlı ölçüm verileri</i> grafiği .....	107
Şekil 4.6. “PI-4x” hız denetimi ile hareketsiz nesne takibi deneyi ve deneyin <i>zamana bağlı ölçüm verileri</i> grafiği .....	109
Şekil 4.7. Test sehpası ve nesnenin hareket yörüngesi .....	111
Şekil 4.8. “PI-1x” Hız Denetimi ile Hareketli Nesne Takibi Deneyi .....	112
Şekil 4.9. “PI-1x” hız denetimi ile hareketli nesne takibi deneyi ve deneyin <i>zamana bağlı ölçüm verileri</i> grafiği .....	113
Şekil 4.10. Nesne B ve A noktasında iken yakınlaştırılmış grafik .....	114
Şekil 4.11. “PI-4x” Hız Denetimi ile Hareketli Nesne Takibi Deneyi .....	116
Şekil 4.12. “PI-4x” hız denetimi ile hareketli nesne takibi deneyi ve deneyin <i>zamana bağlı ölçüm verileri</i> grafiği .....	117
Şekil 4.13. Nesne B ve A noktasında iken yakınlaştırılmış grafik .....	118

## ÇİZELGELER DİZİNİ

### Sayfa

Çizelge 3.1. V-Modeline Göre Geleneksel İş Akışı .....	17
Çizelge 3.2. Geleneksel Yaklaşımındaki Güçlükler .....	18
Çizelge 3.3. Kullanılan DC motorun parametreleri.....	35
Çizelge 3.4. DC Motor Konektöründeki Kablolar .....	35
Çizelge 3.5. Tez çalışması için MF624 kartında kullanılan I/O kapıları.....	41
Çizelge 3.6. MATLAB Coder ve Simulink Coder İşlevleri.....	47
Çizelge 3.7. DC motorun denklemleri ve transfer fonksiyonu temsilleri.....	65
Çizelge 3.8. DC motorun bilinmeyen parametreleri ve yaklaşık hesaplanan değerler...	74
Çizelge 3.9. P denetimli sistemin birim basamak yanıtı .....	80
Çizelge 3.10. Aday Denetleyiciler ve Özellikleri.....	88
Çizelge 3.11. Farklı denetim yapılanmaları ile sistemin gerçek-zamanlı test sonuçları.	95
Çizelge 3.12. Makinenin Alabileceği Durumlar ve Geçiş Koşulları .....	99
Çizelge 3.13. Farklı denetleyici yapılanmasının genel performans üzerine etkisi .....	110

## 1. GİRİŞ

Mekanik ve elektronik bileşenler, akıllı bilgisayar denetimi ile birlikte mekatronik sistemleri oluştururlar. Bundan dolayı mekatronik sistemler bilgi teknolojilerinden çok şey ödünç almıştır ve gömülü yazılımların tasarımı da mekatronik sistemlerin tasarımında çok önemli bir rol oynamaktadır. Gömülü yazılımlar, bir ürünü rakiplerinden ayırarak ona rekabet üstünlüğü sağlar, güç ve esneklik kazandırır. Modern gömülü denetim sistemlerinin günümüzdeki karmaşıklığı, tasarımlarında sistematik yöntemleri de gerektirmiştir. Bu çalışmada, gömülü yazılımların ve karmaşık kontrol sistemlerinin tasarımındaki güçlüklerin üstesinden gelmek için kullanılan matematiksel ve görsel bir yöntem olan Model-Tabanlı Tasarım süreçleri örnek bir çalışmaya uygulanmıştır.

Tüm bu süreçler, bu tez için geliştirilen ve “nesne takip sistemi” olarak adlandırılan mekatronik sisteme uygulanmıştır ve geleneksel geliştirme süreçlerine olan katkıları da ayrıca görülmüştür. Nesne takip sistemi, belirli bir renge sahip bir nesneyi görme alanı içerisinde algılayabilen, kameranın açisal hareketini denetleyerek nesneyi görüntü içerisinde merkezleyen bir servo sistemdir. Bu sistem, mekatronik bir sistemin bir çok unsurunu içermektedir.

Geliştirilen sistem, görme ve hareket alt sistemlerinden oluşmaktadır. Görme altsistemi donanımsal olarak bir adet kameradan, yazılımsal olarak ise belirli bir nesnenin algılanmasına ve konumunun hesaplanmasına yönelik algoritmalarından oluşmaktadır. Hareket alt sistemi ise donanımsal olarak konum geri beslemeli bir DC motordan, yazılımsal olarak da DC motorun açisal konum ve hız denetimini gerçekleştiren denetim algoritmalarından oluşmaktadır. Ayrıca, makinenin durumları arasındaki geçişler ve görüntü-hareket altsistemleri arasındaki bütünleşme, “durum geçiş denetimleri” ile sağlanmaktadır.

Görüntü içerisindeki nesnelerin algılanması ve izlenmesine yönelik teknikler bir çok alanda kullanılmaktadır. Bilgisayarla görme, kontrol teorisi, makine mühendisliği, optik ve endüstriyel otomasyon gibi alanların bir arada kullanıldığı makine ile görme sistemleri (*machine-vision*) bir çok uygulamada yer alır: Endüstriyel robotik, otonom taşıtlarda navigasyon, görsel gözleme dayalı olayların algılanması, üretim bantlarında ürünlerin otomatik incelenmesi gibi uygulamalar örnek olarak verilebilir.

Makine ile görme sistemlerinin endüstrideki kullanımını aşağıda özetlenmiştir:

- **Otomotiv endüstrisinde:** *Machine-vision*, bu endüstride özellikle aşağıdaki amaçlar için kullanılır:
  - o *Robotlara rehberlik:* Nesnelerin konumlarını bulmak, raflardan ve sandıklardan parçaları almak ve montaja yönelik parçaları konumlandırmak amaçlı kullanılabilir.
  - o *Muayene:* *Machine-vision* sistemleri ile, üretilen parçaların yüzeyleri incelenerek estetik ve işlevsel kusurlar algılanabilir. Ayrıca üretim hattındaki parçaların varlığını, şekil düzgünlüğü denetlenebilir, birleştirilmiş bir üründeki montaj kusurları incelenebilir.
  - o *Parça tanılama:* Üretim bandındaki parçaların tanınmasında ve sınıflandırılmasına kullanılabilir. Örnek olarak, farklı türlerde üretilen otomobil tekerlekleri, ait olduğu aracın üretildiği banda kendiliğinden yönlendirilebilir.
  - o *Boyutsal ölçümler:* Hassas işlenmiş parçaların (kavramalar, vites kutusu ve diğer alt bileşenler) boyutsal ölçümlerinde kullanılmaktadır. Böylece parçaların boyutsal toleranslar içerisinde olup olmadığı belirlenebilir, olmayanları üretim bandından uzaklaştırılabilir.
  - o *2B matris verilerinin okunması:* Parçaların üzerine lazerle işlenen operasyon bilgilerinin okunmasında ve ilgili parçaya doğru operasyonun uygulanmasında kullanılmaktadır.
- **Otonom taşıtlarda;** insan müdahalesi olmaksızın ilerleyebilen otonom araçlar iyi bir *machine-vision* uygulamasıdır. Bu araçlar trafik kurallarına uymak, engellerden ve diğer araçlardan kaçınmak, gideceği yolları planlamak ve yönlendirmek gibi işlemleri gerçek zamanlı olarak gerçekleştirebilir.



- **Tıp sektöründe;** *machine-vision* sistemleri sayısal radyografi ve kan/ilaç taraması gibi tanı aracı olarak kullanılabileceği gibi hassas tıbbi aygıtların üretiminde de kullanılabilir.
- **Paketleme hatlarında** kullanılmaktadır ve yaygın kullanımlarına örnekler aşağıdaki gibi verilebilir:
  - o Etiketlerin incelenmesi (varlığı ve yokluğu, yırtıklar, baskı kalitesi, optik karakter tanılama ve doğrulama vs.),
  - o tenekelerin, kutuların ve kartonların göçüklere ve kapanmamış kapaklara yönelik incelenmesi,
  - o robotların tutma ve yerleştirme işlemlerine yönelik rehberlik edilmesi,
  - o şişelerdeki sıvı seviyelerinin doğrulanması gibi örnekler verilebilir.
- **Diğer uygulamaları** ise aşağıdaki listelenebilir:
  - o Baskı ve renk incelemede,
  - o barkod ve karakter tanılamada,
  - o yüzey analizinde,
  - o 2B/3B konumlandırma ve nesne tanılamada,
  - o 3B-kalite, 3B-geometri ve 3B-biçim denetiminde,
  - o gıda kalite denetiminde kullanılmaktadır.

Sonuç olarak, bu tez kapsamında gerçekleştirilmesi hedeflenen çalışmalar aşağıdaki gibi özetlenmiştir:

- Belirli bir alana ve renge sahip hareketli bir nesnenin takibini gerçekleştiren bir servo sisteme yönelik kararlı algoritmaların Model-tabanlı tasarım ile oluşturulması,
- geliştirilen bu algoritmaların bir hızlı denetim prototipleme sistemi ile gerçeklenip doğrulanması,
- model-tabanlı tasarımın, geleneksel geliştirme süreçlerine katkılarının incelenmesi,

hedeflenmiştir.

## 2. KAYNAK ARAŞTIRMASI

Bu kısımda makine ile görme (*machine-vision*) ve görme-tabanlı görsel servolara ilişkin kaynak araştırmalarına yer verilmiştir ve model-tabanlı tasarımın gömülü denetim sistemleri tasarımına olan katkıları araştırılmıştır.

**Chaumette ve Hutchinson (2006)**, görüntü işlemenin (bilgisayarla görme) robotların servo denetiminde kullanıldığı sistemleri, diğer bir deyişle görsel servo denetimini genel olarak incelemiştir. İncelemede *görsel servo denetimi*, bir robotun hareketli bir uzvuna yada mobil bir robotun üzerine bağlı bir kameradan alınan görüntü verilerinin, robotun hareket denetiminde kullanıldığı sistemler olarak tanımlanmıştır. İncelemede bu sistemler, gereksinim duyulan unsurlar görüntü içerisinde bulunuyorsa “görüntü-tabanlı servo denetimi”; görüntü ölçümlerinden tahmin edilmek zorunda ise “konum-tabanlı servo denetimi” olarak adlandırılmıştır.

Nesne gözlem ve nesne takip sistemleri ve diğer çeşitli servo sistemlere ilişkin kaynaklar araştırılmıştır. Çeşitli araştırmacıların bu konuda gerçekleştirdikleri çalışmalar özetlenmiştir. Kaynaklar incelenirken kullanılan algoritmalar, yöntemler ve donanımlar özellikle göz önünde bulundurulmuştur.

**Lee ve ark. (1988)**, hareketli bir kamera ile hareketli nesnelerin takibi üzerine bir çalışma gerçekleştirmiştir. Bir robotun koluna bağlı bir kamera ile çeşitli izleme yöntemleri kullanılarak yoğun dokulu bir arka plandaki hareketli ve tekli bir nesnenin takibi gerçekleştirilmiştir. Nesnenin takibi, sadece x ve y yönlerinde öteleme hareketi yapabilen PUMA 560 model robot kolu ve bu kola bağlı bir CCD kamera ile gerçekleştirilmiştir. Başlangıçta hareketli ve tek bir nesnenin takibi gerçekleştirilmiştir. Nesnelerin takibi; piramit tabanlı işleme algoritmaları ve görüntülerin farkının hesaplanması ile gerçekleştirilmiştir. Çalışmada kullanılan piramit-tabanlı görüntü işleyicisi (PVM-1), Anderson ve ark. (1985) tarafından tanıtılmıştır. PVM-1, görüntüdeki değişimleri gerçek zamanlı olarak algılamaya ve izlemeye olanak tanıyan işlevlere sahiptir. PVM-1'in yaklaşımında, hareketin algılanması görüntünün zamana göre farkı alınarak gerçekleştirilir. Çalışmada kullanılan piksellerin farkının alınması işlemi geçici bir yüksek geçirgen filtre işlevi görür ve böylece daha hızlı hareket eden nesne, fark görüntüde daha fazla enerji bırakır. Daha sonra farkı alınmış bu görüntü,

Laplace piramitlerinin oluşturulması ile uzamsal frekans bantları kümelerine ayrıştırılmıştır. Bu Laplace piramitin her seviyesi, hareketli nesnenin geçici ve uzamsal bilgilerine ilişkin farklı bilgileri içermektedir. Yavaş hareket eden nesne çoğunlukla en üst seviyede ( $L_0$ ) yer almaktadır. Böylece  $L_0$  seviyesindeki görüntü seçilmiş ve piksellerinin mutlak değeri hesaplanmıştır ( $|L_0|$ ). Çalışmada bu mutlak değer, nesnenin hareketliliğinin bir ölçüsü olduğu belirtilmektedir.  $|L_0|$  ile çeşitli çözünürlüklerdeki hareket enerjisine yönelik yerel olarak integre edilmiş ölçümleri temsil eden Gauss piramiti oluşturulmuştur. Nihai Gauss piramiti ile hareketli nesnenin konumu yüksek bir verimle hesaplanabilmiştir. Çalışmada, görüntülerin farkını alma işlemi, kameranın görüntülerin alımı sırasında tamamen durgun olmasını gerektirmiştir. Öbür türlü, hareketli bir arka plandaki hareketli bir nesne algılanacağı belirtilmiştir. Bu çalışma durumu, “durgun izle ve hareket et” (SLAM – Static Look-and-Move) olarak isimlendirilmiştir. SLAM yapılanmada, görsel gözlem sırasında robot hareket etmemekte ve nesneyi belirli bir zaman aralığında izlemektedir. Sabit arka plandan hareketli bir nesneyi ayırabilmek için kamerayı tutan eyleyici görsel gözlem süresince tamamen hareketsiz kalmıştır. Çalışmaya göre, SLAM yapılanmasının en büyük eksisi bu kaçınılmaz gecikme zamanlarıdır ve bu gecikme kayıplarının giderilmesine yönelik “dinamik olarak izle ve hareket et” (Dynamic Look-and-Move) yapılanmasının ilkel bir hali ayrıca geliştirilmiştir. Bu çalışmada, en fazla saniyede  $10^\circ$  görüntüleme açısına sahip nesnelere takip edilebilmiştir. Bu da yaklaşık olarak, 60 cm mesafedeki ve 15 cm/s hızındaki bir nesneye karşılık gelir.

**Denzler ve ark. (1994)**, görüntülerdeki hareketli nesnelerin algılanmasına yönelik, sabit bir kameradan alınan görüntülerin farkının alınmasına dayalı hızlı bir algoritmayı tanıtmıştır.

**Briz ve ark. (1994)**, artımsal *encoder*lerden hızın hesaplanmasına yönelik iki farklı yöntemi tanıtmış ve bu yöntemlerin artılarını, eksilerini ortaya koymuştur, benzetimlerle başarımlarını karşılaştırmıştır. Hızın hesaplanmasına yönelik tanıtılan “frekans ölçümü” ve “periyod ölçümü” yöntemleri bir uyarımlı motorun denetiminde kullanılarak, deneysel sonuçları elde etmiştir. Hız ölçümü için ilave bir donanım kullanmaksızın tüm hız aralıklarında, hem geçici durum hem de kalıcı durumda, hassas bir hız ölçümü gerçekleştirilmiştir.

**Kim ve ark. (2005)**, hareket algılama ve izlemeye yönelik bir yöntem önermiştir ve bu yöntemi kamera kullanan bir mobil robot üzerinde gerçekleştirmiştir. Hareketli kameralarla nesne takibindeki zorluk, alınan görüntünün hem nesnenin hem de kameranın hareketini içermesidir. Bu çalışma kapsamında bu zorluğun üstesinden gelen bir yöntem geliştirilmiştir. Geliştirilen hareket algılama sistemi, şu adımları takip etmektedir: i) *Pan-tilt-zoom* türünde bir kameradan görüntüler alınır. ii) Mobil robot hareketli nesneyi algılayabilmek için kamera ve nesnenin bileşik hareketini ayrıştırabilmesi gerekir. Kamera hareketi ayrıştırıldıktan sonra nesnenin hareketi algılanabilir. Bu ayrıştırma işlemi, ardışık iki görüntünün farkı alınarak gerçekleştirilir.  $k$  ve  $k-1$ ,  $k+1$  adımlarındaki görüntülerin farkı belirli bir eşik değerinin üzerinde ise kamera hareketli anlamına gelir. Kameranın hareketi ise bir takım *blob* analizi yöntemleri ve çalışmada sunulan bir denklem ile hesaplanır. Bu işlem ile kameradan kaynaklanan hareketler telafi edilir. iii) Telafi işleminden sonra nesneye ilişkin aday bölgeler belirlenir. iv) Kenar, renk ve şekil gibi unsurlar hesaplanır ve v) hareketli nesne algılanır. vi) Nesne algılandığında nesneye ilişkin konum, boyut, şekil ve renk dağılımı gibi *blob* bilgileri hesaplanır ve nesne, ilgilenilen alan içerisinde izlenir. Denemeler kameralı mobil bir robot üzerinde yapılmıştır ve sözü edilen bu algoritmanın, robotun içerisinde bulunduğu ortamda kararlılığını koruduğu, hareketin algılanmasında ve nesnenin takibinde (önü kapalı olsa yada ortadan kaybolursa bile) iyi bir başarıyı sağladığı ortaya konmuştur.

**Takahashi ve ark. (2005)**, görüntü bilgilerini kullanarak izleme denetimi gerçekleştiren bir aygıt geliştirmişlerdir. CCD kameradan belirli bir mesafede bulunan hareketli bir nesnenin konumunu görsel geri beslemeler ile algılayabilen bir yöntem kullanılmış ve buna ilişkin konum denetçisini geliştirilmiştir. Geliştirilen cihaz kamera, servo motor ve *encoder* bileşenlerine sahiptir. Hareketli nesnelere, PID denetimi ve “genişletilmiş Kalman filtresi sistemi” ile takip edebilmiştir. Nesnenin konumunu ölçebilmek için nesnenin üzerine bir LED yerleştirilmiştir; böylece nesne bir LED’e indirgenmiştir. Deney sonuçları, 20 m uzaktaki bir LED’in konumunu  $\pm 1$  mm’den daha az bir sapma ile algılabildiğini göstermiştir.

**Mir-Nasiri (2006)**, nesne takibini 3B olarak gerçekleştiren kamera tabanlı bir mobil izleme robotunu geliştirmiştir. 2 tekerlekli platforma sahip bu kameralı izleme robotu belirlenen nesneye her zaman ilerleyecek şekilde tasarlanmıştır. Bu izleme robotu görme ve hareket olmak üzere iki ana altistemden oluşmaktadır. Görme altistemi; dahili potansiyometreli ve iki motorlu *pan-tilt* kameradan, PCI tabanlı görüntü yakalama kartından ve PWM tabanlı DC motor sürücü kartından oluşmaktadır. Hareket altistemi ise 2 tekerlekli bir platform ve yükselticilere sahip 2 adet DC servomotordan oluşmaktadır. Çalışmada, görüntü altisteminde 3B nesnelerin algılandığı ve hareket altisteminde ise bu tanımlanan nesneye doğru ilerlenildiği belirtilmiştir. Bu çalışmadaki görüntü denetimindeki amaç, nesnenin daima kamera merkezinde olmasını sağlamaktır. Buradaki yaklaşım, insan gözünün daima hareketli nesneye odaklanmasına benzetilmiştir. Çalışmadaki geliştirilen sistem, görüntüleri sürekli gri olarak alıyor ve 3B uzayda belirli bir şekle sahip nesnelere tanımlayabiliyor. Her bir resim karesinde hareketli nesnenin merkezinin koordinatları piksel cinsinden hesaplanmakta ve kamera sürüş mekanizmasına gönderilecek gerekli denetim sinyalleri hesaplanmaktadır. Robotun tekerleklerinin sürücülerine gönderilen denetim sinyalleri ile robotun yüzünün daima nesneye doğru dönmesi ve robotun nesneye doğru ilerlemesi sağlanmaktadır. Çalışmada kameranın izleme hareketleri *quadrant* yaklaşımı ile gerçekleştirilmiştir. Nesne konumlandırılır ve tanımlanırken; hareketli nesnenin sürekli takibi insan gözünün çalışmasına benzer bir şekilde gerçekleştirilmiştir. Tanımlama sürecini kolaylaştırmak ve zıtlığı artırmak için nesne siyah bir top olarak seçilmiştir. Nesne tanımlanmadan önce alınan görüntüler iyileştirilmiş ve önışlenmiştir. Bunun için çalışmada, alınan görüntülerde arkaplan ile nesne arasındaki zıtlığı artırmak ve gri farklılıklarını azaltmak için *ters gama doğrulaması* uygulanmıştır ve karanlık nesnelere daha karanlık, aydınlık nesnelere de daha aydınlık yapmak için *doğrusal doğrulama fonksiyonu* uygulanmıştır. Çalışmada nesnelere tanımlamak için akan görüntü bilgileri önce ikili (*binary*) biçime dönüştürülmesi gerekeceğinden, gri tonlarda ölçeklenmiş renkleri, belirlenmiş bir eşik değeri ile karşılaştırılarak siyah ve beyaz görüntü elde edilmiştir. Elde edilen görüntüdeki istenmeyen nesnelere ise bazı *blob* teknikleri ile elenmiştir. İşlenmiş bu siyah ve beyaz görüntüde aranan nesne tanımlanırken önce bloblar ve parçacıklar etiketlenmiş, sınır nesnelere kaldırılmış ve 50 piksel ve altındaki parçacıklar gürültü olarak tanımlanıp elenmiş, nihai olarak da nesnenin dairesel özellikleri hesaplanmıştır.

Nesne izleme stratejisi olarak *quadrant* yaklaşımı kullanıldığı için her bir görüntü karesindeki blobların merkezi hesaplanmış, sonra insan gözünün nesnelere takip etmesine benzer şekilde görüntü düzlemi ile nesne arasındaki konum hatası hesaplanmış ve hatanın telafisine yönelik doğrulama ölçümleri gerçekleştirilmiştir. X eksenindeki sapmalar (hatalar ( $dx$ )) kameranın yatay (*pan*) motoru ile, Y eksenindeki hatalar ( $dy$ ) dikey motorun hareketleri ile telafi edilmiştir. Kamera hareket altsistem tasarımında, kameranın dönme miktarı yerine dönme hızı ile ilgilenilmiştir (DC servomotorun hız denetimi). Çalışmadaki hareket denetim stratejisi hareketli nesnenin hızı ile kameranın hızının eşleştirilmesine dayalıdır ve kameranın hızı PWM tekniği ile ayarlanmıştır. Çalışmada motorun besleme gerilimi ise  $dx$  ve  $dy$  sinyalinin mutlak değeri ile orantılı olması sağlanmış ve motorun ivmesi ve yavaşlatılması sikloid ve rampa fonksiyonları ile sağlanmıştır. Kamera hareket altsisteminin donanımsal kurulumunda ise bir tek renkli analog kamera, NI marka PCI görüntü toplama kartı, L298D sürücülü 2 adet DC motora sahip dikey-yatay kamera mekanizması ve LabView 8 yazılımı kullanılmıştır. Çalışmadaki akıllı robotun, nesnelere izleyebilmesi ve nesnelere müdahale edebilmesi amaçlanmıştır ve deneysel sonuçlar, sistemin iyi bir başarıya ve kararlılığa sahip olduğunu göstermiştir.

**Ding ve ark. (2006)** yayınladıkları makalede, geleneksel pasif video izleme yöntemlerinin yerini alacak yeni ve akıllı bir video takip yöntemini tanıtmıştır. Gerçekleştirdikleri çalışmada yerden düşük bir yükseklikteki hareketli nesnelere gerçek zamanlı olarak izleyebilen başlıca bir teknik üzerinde çalışmışlardır. Yayınlarında çalışmayı, görsel gözetim sistemlerinin üç önemli aşaması göz önünde bulundurularak incelemişlerdir: Çevrenin modellenmesi, hareketin bölümlenmesi (*segmentation*) ve izleme. Çalışmada *çevre modelleme* ile arka plan görüntüleri oluşturulmuş; *hareket bölümlenme* ile taşıtlar gibi hareketli nesnelere karşılık gelen ve sonra izlemeye alınacak olan hareketli bölgeler algılanmıştır. Hareket bölümlenirken, “arkaplan ile fark alma”, “geçici fark alma” ve “optik akış” yöntemlerinden faydalanılmıştır. İzleme aşamasında ise unsur-tabanlı izleme yöntemi kullanılmıştır. Geliştirilen algoritmaları sınamak için bir nesne gözlem sistemi geliştirilmiştir. Çalışmada geliştirilen nesne gözlem sisteminin çalışma prensibi şu şekilde özetlenmiştir: Hareketli bir platforma bağlı kameradan görüntüler alınmıştır, bölümlenmiş ve bu görüntülerin unsurları çıkarılarak nesnenin görüntü merkezine göre bağıl konum bilgisi elde edilmiştir. Bu konum bilgisi hareketli

bölgeden hesaplanmıştır. Kameranın otomatik yatay-dikey hareketi ile nesne takip edilmiştir. Bu çalışmanın deneyleri ise Canon marka ve VC-C50iR model bir *pan-tilt* kamera ve Matrox Meteor II model bir görüntü yakalama kartı ile gerçekleştirilmiştir. Kamera, motor tahrikli ve tavana asılmış bir raylı kamera taşıyıcısına bağlanmış ve takip amaçlı olarak uzaktan kumandalı bir oyuncak araba kullanılmıştır. İzlemenin başında kullanıcı bir fare yardımı ile nesneye konumlanmış ve bir alt-pencere oluşturmuştur. Nesne üzerine tıklayarak bölümlene eşığı olarak nesnenin gri-seviye değerlerini çıkarmıştır. Laboratuvar ortamındaki temiz arkaplan görüntüsü ve ışıklandırmanın sabit olmasından dolayı deney sonuçları oldukça olumludur. Diğer bir deney ise bina dışarısında trafik gözlenerek gerçekleştirilmiştir. Kamera sabit bir platforma bağlanmış ve sadece pan-tilt hareketi yapmasına izin verilmiştir. Nesneye alt-pencere ile kilitlenildiğinde taşıtı, yolun sonuna kadar kendiliğinden takip edebilmiştir.

**Petrella ve ark. (2007)**, düşük çözünürlükü ve ucuz artımsal encoderlerden hız ölçümüne ve tahminine yönelik algoritmaları karşılaştırmalı olarak tanıtmıştır. Briz ve ark. (1994) çalışmasına ilave olarak frekans/periyod ölçümü yöntemlerinin beraber kullanıldığı bir yöntemi (karışık mod yöntemi) ve “gözleyici tabanlı” (*observer-based*) yöntemleri tanıtmıştır. Karışık mod yönteminde gözlenen hız ölçümü hatalarının, frekans ve periyod ölçümü yöntemlerindeki göre çok düşük seviyede olduğu sonucuna varmıştır.

**Loh ve ark. (2007)**, bir elektronik kısma valfi denetimi sistemi üzerinde sistem modelleme, parametre kestirimi, denetleyici tasarımı ve gerçek-zamanlı denetim çalışması gerçekleştirmiştir. Çalışmada Bosch tarafından üretilen bir kısma valfi sistemi kullanılmıştır ve bu sistemin bilinmeyen parametreleri hesaplanmış ve bu sisteme doğrusal ve doğrusal olmayan denetleyiciler geliştirilmiştir. Gerçek-zamanlı denetleyici xPC Target ile gerçekleştirilmiştir. Benzetim ortamında ve deneysel elde edilen sonuçların mükemmel eşleştiği sonucuna varılmıştır. Ayrıca mevcut tasarım, bağımsız uyarımlı doğrusal olmayan bir DC motor ile genişletilmiştir. Benzetim sonuçları bu yeni tasarımın, denetim gereksinimlerini mükemmel karşıladığı sonucuna varmıştır. İçten yanmalı motorlara gönderilen yakıt/hava karışımının debisini ayarlama kullanılan bu elektronik kısma valfi sistemleri aslında birer servo denetim sistemidir ve bu tez çalışmasında sözü geçen kuramlardan faydalanılmıştır.

**Lang ve ark. (2010)**, nesnelere tanımlayabilen ve izleyebilen görmeye dayalı bir mobil robot üzerinde çalışmıştır. Çalışmayı gerçekleştirebilmek için nesnenin konumunu piksel koordinatları cinsinden hesaplayacak kararlı bir nesne takip algoritmasına ve şimdiki konum ile hedef konum arasındaki piksel cinsinden hatayı sıfırlayacak bir *görüntü-tabanlı görsel servo* denetim stratejisine gereksinim duyulmuştur. Bu çalışmada kamera yardımıyla sağlanan görme, denetim sisteminin parçasıdır. Görme ile alınan ortam bilgisi bir geri besleme bilgisi olarak denetim döngüsüne sağlanmıştır. Çalışmada nesnelere tanımlanması ve takibinde unsurlar, *ölçekten-bağımsız unsur dönüşümü* algoritması ile sağlanmıştır. Çalışmada kullanılan bu algoritma, Lowe (1999) ve Lowe (2004) tarafından öne sürülmüştür ve kullanılan algoritmanın ışık miktarı, gürültü, dönme ve ölçeklemelerden bağımsız olduğu ve kararlılığın yüksek olduğu belirtilmiştir. Çalışmada mobil robotun kamera görüntüsündeki nesneyi takip edebilmesi, ortam bilgisini geri beslemesi ve robotun hedef konuma yönelik denetlenebilmesi için *ölçekten-bağımsız unsur dönüşümü* algoritması kullanılmıştır. Çalışmaya ilişkin deneyler bütünleşik kameralı bir Pioneer DX3TM mobil robotu üzerinde denenmiştir. Robot, görüntü içerisindeki istenilen konuma erişene kadar hedefe ilerleyebilmesi için bir *görüntü-tabanlı görsel servodan* faydalanılmıştır. Çalışma sonunda şimdiki konum ile hedef konum arasındaki piksel cinsinden hatayı sıfıra yaklaştırarak robotun sürüldüğü sonucuna varılmıştır.

**Kim ve ark. (2011)**, uyarlamalı geri-adım denetimli (*adaptive backstepping control*) bir otonom *quadrotor* bir *görüntü-tabanlı görsel servo* ile bütünleştirmiştir. Bu *görüntü-tabanlı görsel servo*, görüntüden referans hızları ve görüntü düzlemindeki konum hatasını Jacobian matrisleri ile hesaplamaktadır. Belirsiz bozucu büyüklüklere rağmen referans hızın takip edilebilmesini sağlamak için, iticilerin giriş sinyalleri, uyarlamalı geri-adım denetimi ile oluşturulmuştur. Çalışma sonunda bütünleştirilen bu sistem ile görüntüden çıkarılan unsurların, robotun en az konum hatası ile hedefe ulaşabildiği gözlemlenmiştir.

**Zhou ve ark. (2011)**, çalışmasında stereo görmeye dayalı robot denetim teknolojilerini incelemiştir. Stereo görmeye dayalı robotların çalışma koşullarına daha iyi uyum sağladıkları; insan müdahalesi olmaksızın konumlandırma, tutma gibi işlevleri daha iyi yerine getirdiğini ortaya koymuştur.



*Shu ve ark. (2011)*, çalışmalarında bir kameralı servo sistem tasarlayıp uygulamıştır. Çalışmada kullanılan algoritma görüntülerin farkını almaya dayalıdır ve bu algoritma çeşitli dış ortamlarda kullanılmıştır. Çalışma sonucunda hareketli nesnenin görüntünün sürekli merkezinde kalması sağlanmıştır.

Model-tabanlı tasarım yönteminin tez konusu çalışmaya olan katkıları araştırılmıştır ve bu yöntem ile gerçekleştirilen çalışmalar incelenmiştir. Araştırmalar sonucunda, model-tabanlı tasarımın karmaşık denetim sistemlerinin tasarımına sağladığı artılar ortaya konmuştur. Model-tabanlı tasarım ile gerçekleştirilmiş çalışmalara ilişkin araştırmalar aşağıda özetlenmiştir:

*Oherek ve Robl (2001)*, ağır taşıt üreticisi MAN AG ile gerçekleştirdiği bir çalışmada; MAN marka bir kamyonun hidrostatik öntekerlek sürücü deneçisini model-tabanlı tasarım yaklaşımı ile tasarlamıştır. Geliştirilen elektronik denetim birimi (ECU), gerçek bir taşıt üzerinde gerçeklenip sınanmıştır. Geliştirilen gerçek zamanlı uygulama, VxWorks işletim sistemi ile çalışan bir dizüstü bilgisayarda gerçeklenip, I/O bağlantıları CAN veri yolu ile sağlanmıştır. Çalışma sonucunda model-tabanlı tasarımın bu çalışmaya katkıları ortaya konmuştur: i) Modellenen sistem, erken tasarım safhasında benzetimlerle doğrulanabilmiştir. ii) Doğrulanmış aynı denetim algoritmaları farklı hedef ortamlarda kolayca gerçekleştirilebilir. iii) Model tabanlı tasarımın görsel bir yöntem olması tasarım süreçlerinin okunabilirliğini artırmıştır. Tasarımın elle yazılmış kodlara göre daha okunaklı olması hatalara karşı çok daha az eğilimli olmasını sağlamıştır.

*Oherek ve Robl (2001)*, Model-tabanlı tasarım yaklaşımı ile yakıt hücrelerinde kullanılan ve elektrikle ısıtılan buharlaştırıcı geliştirmiştir. Buharlaştırıcının tasarımı ve optimizasyonunun yanı sıra denetçi sistemi de geliştirilmiştir. Denetim algoritmaları MPC555 mikrodeneçiyicisi üzerinde gerçekleştirilmiştir. Çalışma neticesinde, model-tabanlı tasarım yaklaşımının geleneksel iş akışına olan üstünlükleri ortaya konmuştur.

**Ahmadian ve ark. (2005)**, yazılım tanımlı bir telsizin sisteminin başlıca geliştirme aşamalarını, model-tabanlı tasarımın unsurlarını (tasarım, benzetim, kod oluşturma ve doğrulama) temel alarak incelemiştir. Bu çalışmada bir örnek üzerinden, model-tabanlı tasarım yaklaşımı ile bu başlıca aşamaların tek bir sürekli tasarım döngüsü içerisinde bütünleştirilmesini ele almıştır. Benzetim, kod oluşturma, HIL testleri gibi ileri seviye tasarım kavramları tanıtılmıştır. Bir “*Future Multi-Band, Multi-Waveform, Modular, Tactical Radio*” modeminin gerçekleşmesi tartışılmış ve gösterilmiştir. SMT6050 (Sundance firması tarafından geliştirilen ve Sundance donanımlarına yönelik kod oluşturmaya olanak tanıyan bir MATLAB eklentisi) ile oluşturulan kod, HIL benzetimleri ve sınamalarına yönelik kullanılmıştır. Bu çalışma ile yazılım-tanımlı radyo uygulamalarının model-tabanlı tasarım ile, geleneksel iş akışında karşılaşılan problemler olmaksızın çok daha kolayca modellenebildiği ve geliştirilebildiği sonucuna varılmıştır.

**Ghidella ve Mosterman (2005)**, sivil uçaklarda yaygın olarak rastlanabilen yedek asansör sistemlerine yönelik bir hata algılama, ayırma ve kurtarma (FDIR) sistemi tasarlamıştır. Tanıtılan çalışmada her bir asansör, aynı anda sadece biri etkin olabilen iki ayrı eyleyici tarafından konumlandırılmaktadır. Çalışmadaki hata algılama, ayırma ve kurtarma sistemi model-tabanlı tasarım iş akışına göre geliştirilmiştir ve model-tabanlı tasarımın daha çok sınama ve doğrulama unsurları üzerinde durulmuştur. Oluşturulan bilgisayar benzetiminde, proje isterlerine dayalı sınama senaryoları oluşturulmuş ve tasarımın tamamen umulduğu gibi davrandığından emin olmak için “kapsama ölçümleri” (*covarage metrics*) gerçekleştirilmiştir. Çalışma sonucunda model-tabanlı tasarımın, sistemin hayata geçirilmeden sınanmasına ve oluşabilecek hataları daha erken aşamalarda bulunabilmesine olanak tanıdığı ortaya konmuştur.

**Kirby ve Kang (2008)**, güç sistemleri koruma rölelerine yönelik bir çalışma gerçekleştirmiş ve model-tabanlı tasarım ile tasarım döngüsünü iyileştirmeye yönelik yenilikçi çalışma yöntemlerini tanıtmıştır. Projelerin tamamlanma sürelerini uzatan ve son aşamada farkedilen hataların güçlülere neden olduğu geleneksel iş akışına karşılık model-tabanlı tasarım yaklaşımını önermiştir. Model-tabanlı tasarım yaklaşımı ile zaman ve proje maliyetlerinden önemli miktarda kazanılabileceği sonucuna varmıştır.

### 3. MATERYAL VE YÖNTEM

Tez konusu çalışmanın, mekatronik bir sistemin tüm unsurlarını kapsadığından söz edilmiştir. Çalışma, tamamen mekatronik disiplini çerçevesi içerisinde gerçekleştirilmiştir ve bağlı kalınan bu disiplinler bu kısımda kapsamlıca anlatılmıştır.

Geliştirilen sistem aslında bir “*gerçek-zamanlı test*” uygulamasıdır; sistemin sahip olduğu tüm donanımsal/yazılımsal bileşenler ve sistemin çalışma ilkeleri de bu kısımda ayrıntılı bir şekilde verilmiştir.

Tasarım ve doğrulama süreçlerini daha akıcı ele alabilmek için sistem, alt-sistemlere ayrıştırılmıştır. Her bir alt-sisteme ilişkin matematiksel modeller kurulmuş ve bu matematiksel modeller deney düzeneğinden alınan veriler ile doğrulanmıştır. Sistem genel hatları ile “*görme*” ve “*hareket*” alt sistemlerinden oluşmaktadır. Video işleme algoritmaları, hareket denetimi ve mantıksal durum denetimleri içeren bu alt sistemler deney düzeneği üzerinde gerçek-zamanlı olarak test edilmiştir.

#### 3.1. Materyal

##### 3.1.1. Çalışmada Faydalanılan Disiplinler

###### 3.1.1.1. Mekatronik Sistemler ve Mekatronik Sistem Tasarımı

Bir çok teknik alanda ürünlerin yada süreçlerin elektronik ile bütünleşik olarak kullanıldığı gözlemlenebilir. Özellikle de bu 1980’lerden itibaren geliştirilen mekanik sistemler için geçerlidir. Elektriksel ve mekanik parçaları birbirinden ayrık olan bu sistemler zamanla algılayıcılar, eyleyiciler ve sayısal mikroelektronik bileşenlere sahip bütünleşik elektromekanik sistemler haline almıştır. Bu bütünleşik sistemlere, mekanik ve elektronik sözcüklerinden türetilen ***mekatronik sistemler*** denilmiştir.

Mekatronik sözcüğü ilk defa 1969 yılında Japon elektrik firması Yasakawa’nın tanımında geçmiştir. Yasakawa, teknolojiler ve ürünlerde elektroniğin hergün daha fazla mekanik ile iç içe geçeceğini ve hangisinin nerede başlayıp nerede biteceğini artık kesin olarak ayırt edemeyeceğimizi söylemiştir. Bugünkü anlamda eksik kalan bu tanım yıllarca evrim geçirerek günümüzdeki tanımlarına ulaşmıştır. Bu tanımlardan birine göre mekatronik; endüstriyel ürün ve süreçlerin tasarımı ve üretiminde, elektronik ve

akıllı bilgisayar kontrolünün makine mühendisliği ile beraber sinerjik bir bütünlüğüdür (Bishop 2002). Bu tanımlardan yola çıkılarak mekatroniğin disiplinler arası bir alan olduğu sonucuna varılabilir. Mekanik sistemler, elektronik sistemler (güç elektroniği, algılayıcı ve eyleyici teknolojileri vs.) ve bilgi teknolojileri (sistem ve kontrol kuramı, otomasyon, yazılım mühendisliği vs.) birlikte rol alarak bir mekatronik sistemi oluştururlar.

Bu tez çalışması kapsamında geliştirilen nesne gözlem sistemi uygulaması, mekatronik sistem geliştirme unsurlarının bir çoğunu kapsamaktadır.

### 3.1.1.2. Gömülü Denetim Sistemleri

Bir denetim sistemi, fiziksel bir sistemin (*plant*) istenilen bir şekilde davranmasını sağlamak üzere uygulanan bir stratejidir. Yaygın olarak iki farklı denetim stratejisi mevcuttur (Mosterman ve ark. 2005):

- ***Kapalı-döngü denetim:*** Denetlenen sistemin (*plant*) yanıtı ile referans girişi (istenilen davranış) arasındaki hatayı düzeltmeye yönelik geri-beslemeli bir denetimdir.
- ***Mantıksal durum denetimi:*** Mantıksal durum denetimi, *olay tabanlı* bir denetimdir ve denetlenen sistemi durum geçişleri ile yönetir.

Tez çalışması kapsamında geliştirilen nesne gözlem sisteminde kamera hareketini sağlayan DC motorun açısal konum ve hız denetimi, *kapalı-döngü bir denetime* iyi bir örnektir. Nesne takip sistemi, nesnenin görüş alanı içerisinde olmasına (*olay*) göre “hız denetimi”, “konum denetimi” ve “beklemede kal” gibi farklı çalışma durumlarına sahiptir ve sahip olduğu anlık duruma göre farklı şekilde çalışacaktır. Böyle bir denetim de *mantıksal durum denetimine* iyi bir örnektir.

Günümüzde denetim sistemleri tamamen sayısal olarak gerçekleştirilir ve mikrodenetleyici tabanlıdır. Çünkü geri beslemeli bir denetimin stratejisi karmaşıklıktıkça, sistemin analog bileşenler kullanarak hayata geçirilmesi giderek zorlaşır. Analog bir geri beslemeli denetimin dinamikleri herşeye tepki verdiği için yada sınırlı bir çalışma aralığına sahip olduğundan, istenilen denetçi özyapısına ulaşmak zordur. Bu nedenlerden dolayı, bilgisayarların yada mikrodenetleyicilerin kullanıldığı

modern denetim sistemlerinde, denetim stratejileri yazılımsal olarak gerçekleştirilirler. Bir başka deyişle, denetlenen sistemi düzenleyecek olan giriş sinyali bir mikrodenetleyici tarafından hesaplanır. Bir mikrodenetleyici yardımı ile yazılımsal olarak gerçekleştirilen ve gerçek-zaman kısıtlarına sahip denetim sistemlerine *gömülü denetim sistemleri* denilir.

Bu tez çalışmasındaki sistemin denetim algoritmaları tamamen gerçek-zamanlı bir çekirdeğe sahip (*xPC Target Real Time Kernel*) bir hedef bilgisayar üzerinde yazılımsal olarak gerçekleştirilmiştir.

### **3.1.1.3. Gömülü Yazılımlar ve V-Modeli**

Mekatronik sözcüğünü tanımlanırken; mekanik, elektronik ve bilgi teknolojilerinin bir arada kullanıldığı sistemlerden söz edilmişti. Bir mekatronik sistemde; mekanik ve elektronik donanımlar, mikroişlemci tabanlı denetim yazılımları ile yönetilir. Bu denetim yazılımlarını tutarlı bir şekilde geliştirebilmek için denetlenen sistemin davranışlarını doğru bir şekilde tanımlamak gerekir. Tersten gidilirse, denetlenen sistemin dinamik davranışları hakkında fikir sahibi olunduğu sürece bu sisteme yönelik denetim yazılımları geliştirilebilir.

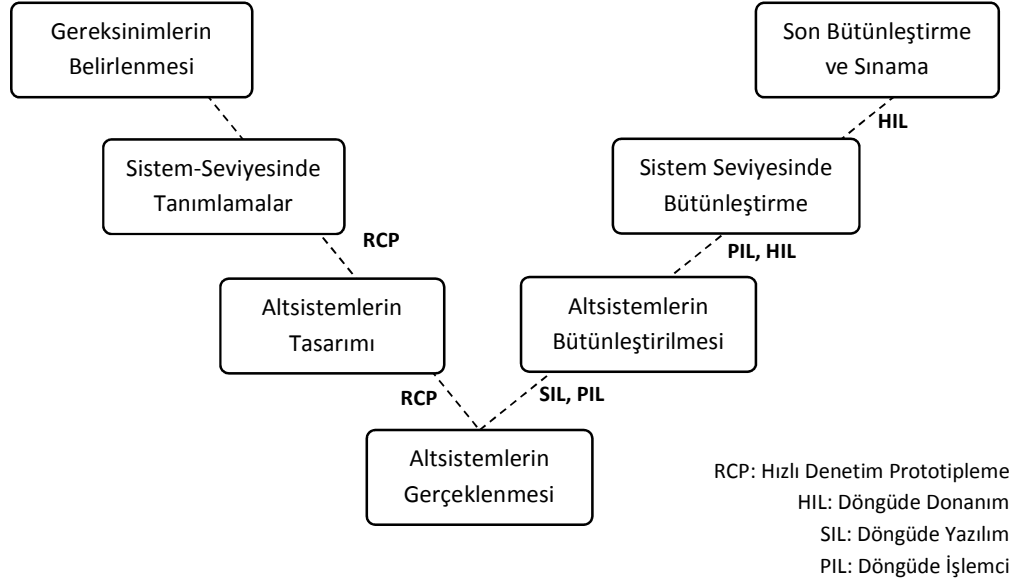
Mikrobilgisayarların yaygın olmadığı dönemlerde, denetlenen sistemlerin davranışlarını tanımlayan matematiksel ifadelerin çözümlenmesi kağıt üzerinde gerçekleştiriliyordu. Tasarlanan denetleyiciler mekanik ya da analog elektronik tabanlı yapıdaydı. Günümüzde ise elle çözümlenmelerin yerini bilgisayar benzetimleri; mekanik/analog denetleyicilerin yerini ise mikrodenetleyiciler almıştır. Bu sayede, mikroişlemci tabanlı elektronik donanımlara yönelik geliştirilen yazılımlara gereksinim duyulmuştur ve “gömülü yazılımcılık” kavramı ortaya çıkmıştır.

Gömülü yazılımlar, mikroişlemci tabanlı ürünlerin tamamında yer alır. Gömülü yazılımlar için ilk akla gelen ürünler otomobillerdir. Bir otomobilde; elektrikli direksiyon (*electric powered steering*), elektronik kısma valfi denetimi (*electronic throttle control*), ABS, otomatik vites, hava yastıkları, motor denetimi, yolcu bilgilendirme, klima denetimleri vs. gibi altbileşenler tamamen gömülü yazılımlar ile denetlenir.

Gömülü yazılımlar, farklı mühendislik alanlarına hakimiyet gerektirdiğinden genelde takımlar halinde geliştirilir. Bu denli bir ekip çalışması ayrıca bir proje yönetim yöntemini gerektirir. Sistem mühendisliğinde ve proje yönetimlerinde “V-modeli” olarak isimlendirilen iş akışı modeli, gömülü yazılımlar için bir endüstri standardı haline gelmiştir. V-modeli, ilk defa 1980’li yıllarda eşzamanlı olarak Amerikan ve Alman savunma sanayisi tarafından geliştirilmiş ve günümüzde bir çok kurum tarafından iş akışı modeli olarak benimsenmiştir.

Şekil 3.1’de verildiği üzere V-modeli, sistem geliştirme süreçlerindeki doğrulama ve sağlama aşamalarına odaklanan bir sistem geliştirme yöntemidir. Sol taraf, gömülü yazılıma ilişkin kodun oluşturulması ile sonuçlanır. Sağ taraf ise sol taraftaki aşamaların doğrulanması ve sağlanmasına odaklanır. Genelde, sistemlerin geliştirilmesine en tepedeki gereksinimlerin (*requirement*) bir araya getirilmesi ile başlanır. Bu gereksinimler genelde metin belgeleri biçiminde saklanır. Bu gereksinimlerden sistem tanımlamaları (*specification*) elde edilir. Hayata geçirilecek ayrıntılı tasarımlar, sistem tanımlamalarından oluşturulur. Sistem tanımlamaları, ayrıntılı tasarım elde edilene kadar aşamalı olarak sadeleştirilir. (Mosterman ve Ghidella 2005)

Bu geleneksel iş akışının aşamaları Çizelge 3.1’de ve bu yaklaşımın zorlukları Çizelge 3.2’de özetlenmiştir.



**Şekil 3.1.** V-Modeli İş Akışı

**Çizelge 3.1.** V-Modeline Göre Geleneksel İş Akışı

Aşama	Açıklama
Gereksinimler ve Tanımlamalar	Sistem mühendisleri, tasarımın farklı bileşenlerine ilişkin işlevsel gereksinimleri ve tanımlamaları oluşturur.
Tasarım	Bazen ayrı takımlar halinde çalışan tasarım mühendisleri sistem bileşenlerinin modellerini oluşturur.
Gerçekleme	Tasarım mühendisleri tarafından oluşturulan modeller fiziksel ortamda gerçekleştirilir. Tasarım mühendisleri tarafından oluşturulan algoritmalarla yola çıkarak, yazılım mühendisleri tarafından gömülü yazılımlar bu aşamada gerçekleştirilir.
Sınama ve Doğrulama	Sistemin bileşenleri ve sistemin bütünü, beklenen işlevleri yerine getirdiğini doğrulamak için sınanır.

V-modeli iş akışında, Çizelge 3.1’de tanımlanan aşamalar birbirinden ayrıdır ve beraberinde Çizelge 3.2’de özetlenen bir takım güçlükler getirir.

**Çizelge 3.2.** Geleneksel Yaklaşımındaki Güçlükler

<b>Güçlük</b>	<b>Örnek Durum</b>
Takımlar arasındaki iletişim güçlüğü	Tasarımlar ve tanımlamalar genelde belgeler halinde yazılır ve tasarım mühendisleri tarafından okunmak ve anlaşılacak zorundadır.
Aynı işin tekrarlanması	Yazılım mühendisleri, tasarım mühendisinin oluşturduğu algoritmaları programlama diliyle yazabilmek için yeniden oluşturmak zorundadır.
Tasarım aşamalarının sonunda farkedilen hatalar	Bileşenler bütünleştirilirken bir hata bulunursa, bu hata tasarımın test aşamasına kadar farkedilmeyecektir.

Sağladığı bir takım araçlar ile V-modeli iş akışındaki aşamaların bir arada olmasını sağlayan ve bu süreçleri kolaylaştıran model-tabanlı tasarım yaklaşımı ortaya konmuştur.

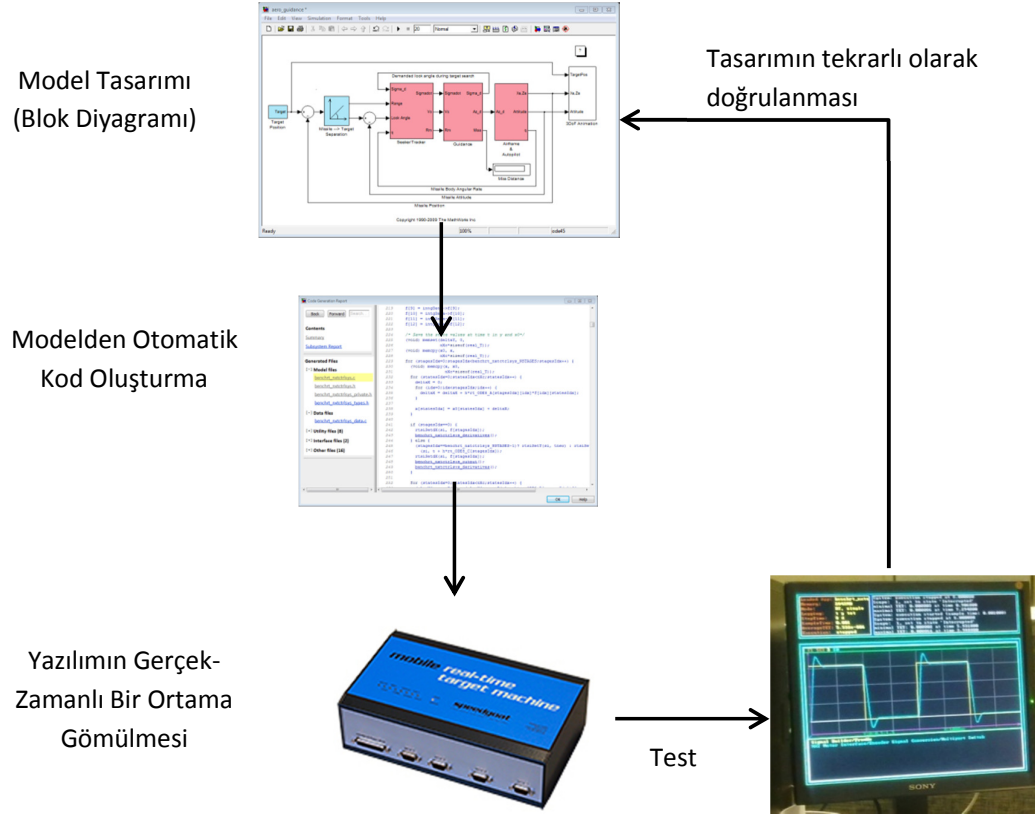
#### **3.1.1.4. Model Tabanlı Tasarım**

Otomotiv ve havacılık/uzay elektroniği uygulamalarındaki hesaplama dayalı işlevsellik, katlanarak daha karmaşık hale gelmektedir. 1970'li yıllarda ortalama bir otomobilde 100 bin gömülü program satırı bulunurken, günümüzde 100 milyon program satırına ulaşmıştır. (Popovici ve Lalo 2009). Günümüzdeki karmaşık denetim sistemlerinin tasarımı, artık yeni bir sistematik yaklaşım gerektirecek hale gelmiştir (Mosterman ve Ghidella 2005).

Model-tabanlı tasarım; gömülü yazılımların ve karmaşık kontrol sistemlerinin tasarımındaki güçlüklerin üstesinden gelmek için kullanılan ve sistemin matematiksel modelinin grafiksel bir blok diyagramı ortamında (Simulink gibi) geliştirildiği bir yöntem disiplindir. Model-tabanlı tasarımın genelleştirilmiş bir iş akışı Şekil 3.2'de verilmiştir. V-modeli, sistemlerin geliştirilmesine yönelik aşamaları tayin ederken, model-tabanlı tasarım ise bu aşamaların gerçekleştirilmesine yönelik araçlar ve

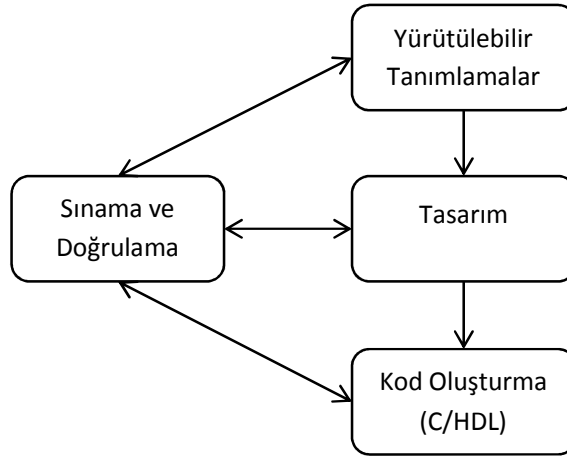


teknolojiler sunar. Model-tabanlı tasarımda, Çizelge 3.1’de verilen geleneksel iş akışının yerini, Şekil 3.2’de verilen iş akışı almıştır.



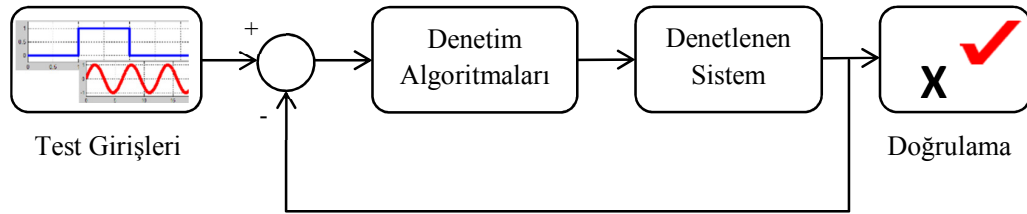
Şekil 3.2. Model-tabanlı tasarım iş akışı

Model tabanlı tasarımda; fiziksel prototipler yada metinsel tanımlamalar yerine, *yürütülebilir tanımlamalar* yerine geçen sistem modeli vardır ve bu model herşeyin merkezidir. Model-tabanlı tasarım, aşağıda özetlenen ve Şekil 3.3’de verilen dört unsuru içerir (Popovici ve Lalo 2009):



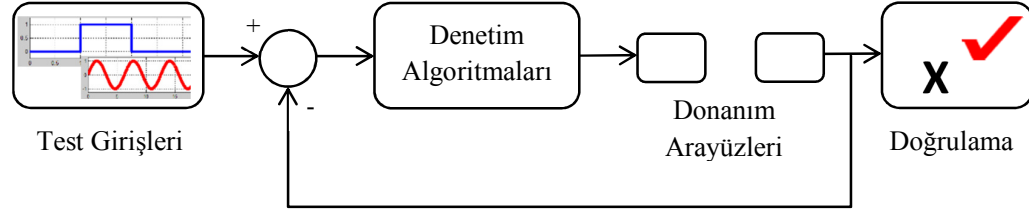
**Şekil 3.3.** Model-tabanlı tasarım ve öğeleri

- **Yürütülebilir tanımlamalar:** Yürütülebilir bir tanımlama (*executable specification*), bilgisayar ortamında benzetimi gerçekleştirilebilir bir model algoritması anlamına gelir. *Yürütülebilir tanımlamalar*, geleneksel iş akışında metinler şeklinde saklanan sistem *tanımlamalarının* ve *gereksinimlerin* yerini alır. Sistem modelleri yürütülebildiğinden dolayı, metin şeklinde saklanan sistem tanımlamalarına göre çok daha az belirsizlikler içerir. Bu tanımlamalar yürütülebilir bir model olduğu için, gerçekleştirme aşamasına daha geçilmeden erken aşamalarda sınav ve doğrulamalar gerçekleştirilebilir (Şekil 3.4). Erken aşamalarda, yürütülebilir tanımlamalar; yazılım/donanım uyumu gibi gerçekleştirme ayrıntılarını içermeyebilir. Daha çok, tasarlanan sistemin davranışlarını yansıtan algoritmalara sahiptir.



**Şekil 3.4.** Bilgisayar Benzetimleri ile Test ve Doğrulama

- **Benzetimlerle tasarımlar:** Model-tabanlı tasarımda sonraki adım, yürütülebilir tanımlamaları ayrıntılı tasarımlara dönüştürmektir (*model elaboration*). Tasarım; bellek yada işlem gücü gibi sınırlı kaynaklara sahip gömülü sistemlere uygun hale getirmek için ayrıntılandırılır (Şekil 3.5). Uygun bir gerçeklemeye yönelik bu tasarım değişiklikleri yapılırken, asıl modele aşağıdaki gibi ayrıntılar eklenir:
  - o Yazılım/donanım uyumu,
  - o hedef donanıma göre sabit-noktalı hassasiyetin eklenmesi,
  - o zamanlama bilgileri ve
  - o bileşen arayüzleri.



Şekil 3.5. Ayrıntılandırılmış Bir Tasarım

- **Kod dönüşümü ile gerçekleştirme:** Otomatik kod dönüşümü teknolojisi ile tasarımlar, donanımlara gömülebilir C yada HDL kodlarına dönüştürülebilir. Eğer elde edilmek istenen, gömülü bir ortamda (DSP yada mikrodenetçiler gibi) çalışacak bir yazılım ise C kodu oluşturulur.
- **Kesintisiz sına ve doğrulama:** Model tabanlı tasarımda sına ve doğrulama, tasarımın ilk aşamalarında başlayan kesintisiz bir süreçtir.

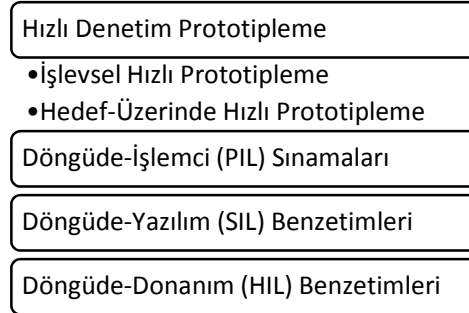
Model tabanlı tasarım bir bilgisayar destekli tasarım teknolojisidir ve model-tabanlı tasarımı temel alan ticari paket yazılımlar mevcuttur. MathWorks'ün Simulink yazılımı ve NI'nın LabView yazılımı buna örnek verilebilir. Bu çalışma kapsamında Simulink yazılımı tercih edilmiştir.

### 3.1.1.5. Gerçek Zamanlı Test ve Uygulamaları

Denetim sistemleri üzerinde geleneksel yöntemler ile çalışan kontrol mühendisleri, son ürünün çalışan bir prototipini bile sürecin son aşamalarında sahip olabilmektedir. Ancak sistem bütünleştirildikten sonra sistemin beklenildiği gibi çalışıp çalışmadığı anlaşılmaktadır. Sistemin tasarlandığı, gerçekleştirildiği ve bütünleştirildiği aşamalardaki olası hatalar ancak son aşamada farkedilmektedir. Model-tabanlı tasarımın sağladığı en büyük artılardan biri, sadece son aşamada değil, tasarım çevriminin her aşamasında test ve doğrulamaya olanak sağlanmasıdır (Şekil 3.3). Bu kesintisiz doğrulama ve sına, hataların erken aşamalarda tespit edilmesine ve düzeltilmesine olanak sağlamaktadır.

Model-tabanlı tasarımda, Şekil 3.2'den (sayfa 19) de görüldüğü gibi, sistemin matematiksel modelini temsil eden blok-diyagramlarından, otomatik olarak kod oluşturmak mümkündür. Bu oluşturulan kod, gerçek-zamanlı bir hedef bilgisayar üzerine, mikrodenetleyici birimlerine, DSP yada FPGA gibi donanımsal ortamlara gömülmeye ve gerçek zamanlı olarak test edilmeye hazırdır.

Sonuç olarak gerçek-zamanlı test, benzetim modellerinin kendi normal çalışma koşulları altında çalıştırılması, test edilmesi ve doğrulanması işlemlerinin bütünüdür ve yaygın karşılaşılan uygulama biçimleri Şekil 3.6'da verilmiştir.



Şekil 3.6. Gerçek-Zamanlı Test Uygulamaları

Bir denetim sisteminde, denetim algoritmaları *denetlenen sistemin* modeline göre tasarlanır ve analiz edilir (Şekil 3.4). Bu denetim sistemi içerisindeki denetim algoritmalarının gerçek bir *denetlenen sisteme* bağlanması ve bu *denetlenen sistem* üzerinde sınanması gerekir (Şekil 3.5)

**Hızlı denetim prototipleme**; denetim algoritmalarının denetlenen gerçek sistem ile çalıştırılarak hızlıca doğrulanmasına olanak tanıyan bir yöntemdir (Mosterman 2005). Hızlı denetim prototipleme, denetçinin işlevsel bir prototipinin hızlıca oluşturulmasını ve denetim algoritmalarının gerçek I/O aygıtlarına sahip bir gerçek-zamanlı bir ortam üzerinde yinelemeli olarak test edilmesini kapsar (Abourida ve ark. 2009). Hızlı prototiplemenin amacı, tasarım tamamlanmadan önce kusurları ve hataları tamamen belirlemek ve güvenilirliği kazanmaktır (Mosterman 2005). Şekil 3.5 tekrar göz önünde bulundurulduğunda, denetim algoritmaları (test girişleri ve doğrulamalar ile beraber) gerçek-zamanlı bir hedef ortama (bir hedef bilgisayar yada mikrodenetleyici) otomatik kod dönüşümü ile gömülür. Bu sefer denetlenen sistem, bir bilgisayar benzetim modeli yerine, gerçek bir sistemdir.

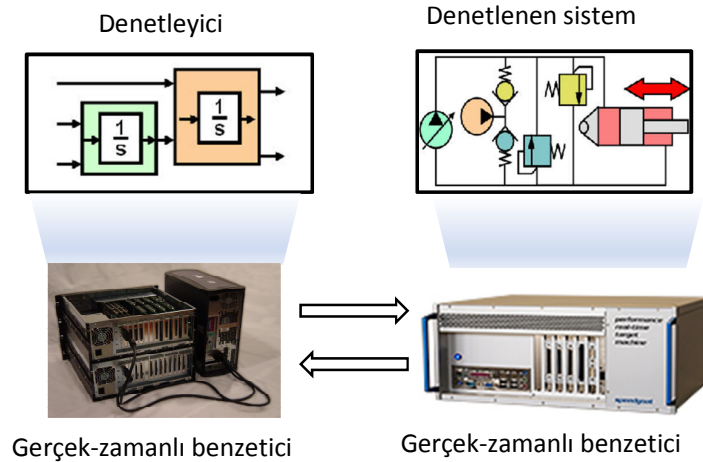
Hızlı denetim prototiplemenin yaygın olarak iki farklı türü vardır.

- **İşlevsel hızlı prototipleme**: Tasarlanan denetçi henüz başlangıç aşamasındayken (ileri denetim stratejilerini desteklemeyecek kadar ilkelken) kavramlar geliştirmek ve yeni düşünceler ortaya koymak için kullanılan hızlı protitipleme yöntemidir. İşlevsel hızlı prototiplemede kullanılan donanım çoğunlukla PC-tabanlıdır ve son ürüne yönelik denetçi uygulamalarına yönelik değildir. Burada odaklanılan şey kavramdır: Oluşturulan kodun performansı, işlem gücü, hafıza tüketimi ve sabit-nokta hassasiyetleri vs. gibi son ürüne yönelik nitelikler göz önünde bulundurulmaz. İşlevsel hızlı prototiplemeye yönelik ticari paket yazılımlar mevcuttur. MathWorks firmasının xPC Target yazılımı, dSpace firmasının TargetLink yazılımı ve Opal-RT firmasının RT-Lab yazılımı işlevsel hızlı prototiplemeye yönelik ortam sağlamaktadır.
- **Hedef-üzerinde hızlı prototipleme**: Denetim algoritmaları, doğrudan son ürün işlemcisi üzerinde prototiplenir ve tüm I/O sınırlandırmalarını kapsayacak şekilde denetlenen sisteme doğrudan bağlanır. Bu yaklaşımda, son ürün kodu oluştururken göz önünde bulundurulan tüm kısıtlar hesaba katılır.

Bilimsel ve akademik araştırmalarda, tasarlanan sistem çok nadir olarak bir son ürüne dönüştürüldüğünden dolayı hızlı prototipleme bu alanda önemli bir rol oynar. Endüstrideki uygulamalarda ise, denetim sistemi henüz bir son ürün haline getirilmeden önce, denetim algoritmalarının test edilmesinde kullanılır (Mosterman 2005).

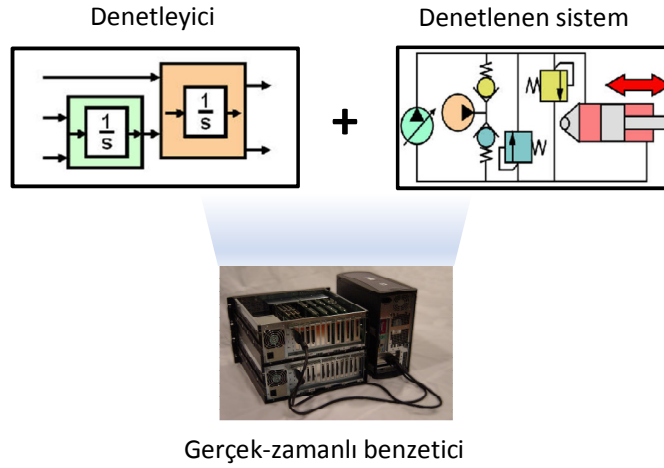
Bu tez çalışmasında ele alınan “nesne takip sistemi”, işlevsel hızlı denetim prototiplemeye bir örnektir. Geliştirilen tüm denetim algoritmaları xPC Target ortamında bir PC üzerinde gerçekleştirilmiştir ve denetlenen gerçek sisteme bağlanmıştır. Tez çalışmasının amacı bir son ürün elde etmek olmadığından, hedef-üzerinde hızlı prototipleme ele alınmamıştır ve gelecekteki çalışmalarda göz önünde bulundurulmak üzere bırakılmıştır.

**Döngüde-donanım (HIL)** uygulamalarında (bu tez çalışmasında kullanılmamakla beraber) gerçek denetleyici, denetlenen gerçek sistem yerine, gerçek zamanlı bir benzetici üzerinde çalışan sanal sisteme bağlanır (Şekil 3.7). Hızlı kontrol prototiplerden farklı olarak bir *döngüde-donanım* uygulamasında, gerçek fiziksel sistem henüz ortada yokken denetçileri erkenden sınamak mümkün olur. Denetlenen sanal sistemler daha az maliyetlidir ve daha sabittir. Böylece daha tekrarlanabilir sonuçlar alınabilir ve gerçek donanım üzerinde uygulanamayan test koşulları (sıra dışı durum testleri gibi) sağlanabilir.



**Şekil 3.7.** HIL yapılanması

Döngüde-yazılım (SIL) ise, hızlı kontrol prototipleme ve HIL'in birleşimini temsil eder. Yeterince güçlü bir hedef bilgisayar ile, hem denetçi hem de denetlenen sistem aynı benzetici içerisinde gerçek zamanlı olarak çalıştırılabilir (Şekil 3.8).



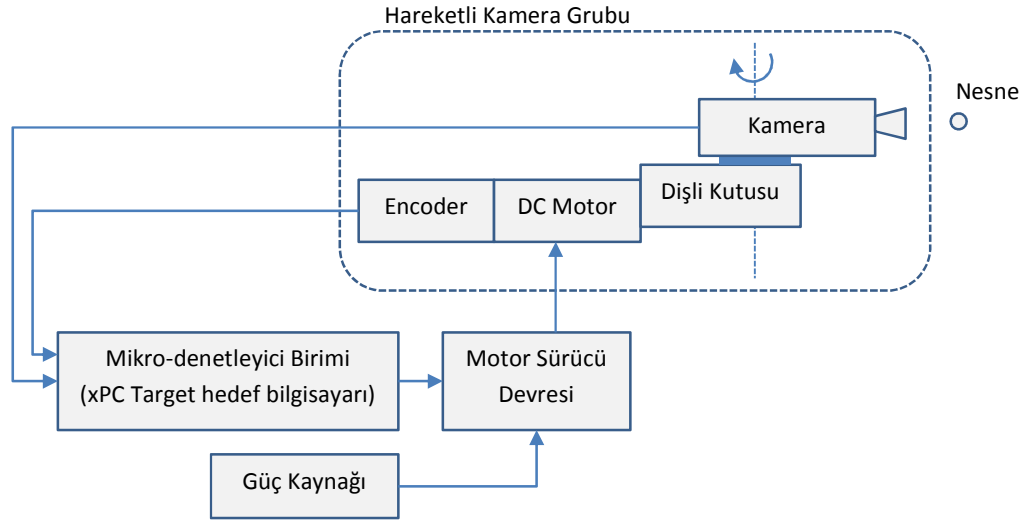
**Şekil 3.8.** SIL Yapılanması

*Döngüde-yazılımın döngüde-donanım* ve hızlı denetim prototiplemeye göre artışı, fiziksel giriş ve çıkış bağlantılarının kullanılmayıdır ve böylece sinyal bütünlüğü sağlanabilir. Buna ilave olarak, hem denetçi hem de denetlenen sistem aynı benzeticide çalıştığından dolayı, fiziksel dış ortam ile ilgili zamanlamalar artık önemli değildir. Sonuçlar üzerinde etkisi olmaksızın benzetimler, gerçek zamandan yavaş ya da hızlı olabilir. Bu da *döngüde-yazılımı*, bir benzetim türü olan “hızlandırılmış benzetim”e (*rapid simulation*) elverişli kılar. Hızlandırılmış durumda bir benzetim gerçek zamandan daha hızlı çalışır. Böylece, kısa bir zaman diliminde çok fazla testin gerçekleştirilmesi mümkün olur. Bu nedenden dolayı *döngüde-yazılımı*, model kapsama (*model coverage*) analizleri, Monte-Carlo benzetimleri gibi istatistiksel sınamalar için çok uygundur. Eğer gerçek zamanlı hedef bilgisayarın işlem gücü, gerçek zamana ulaşmak için yetersizse, yine de bir masaüstü bilgisayardan hızlı olacak şekilde, gerçek zamandan bir oranda daha az hızda olacaktır.

Tez çalışması kapsamında döngüde-yazılım, hareketli kameranın (DC motorun) konum denetleyicisinin tasarımında kullanılmıştır. Benzetim ortamında denetçi algoritmaları doğrulandıktan sonra, bu algoritmaların performansını sınamak üzere xPC Target hedef bilgisayarı üzerinde gerçek zamanlı olarak (hem denetleyici hem de denetlenen sistem) çalıştırılmıştır ve sonuçlar, gerçek sistemin performansı ile karşılaştırılmıştır.

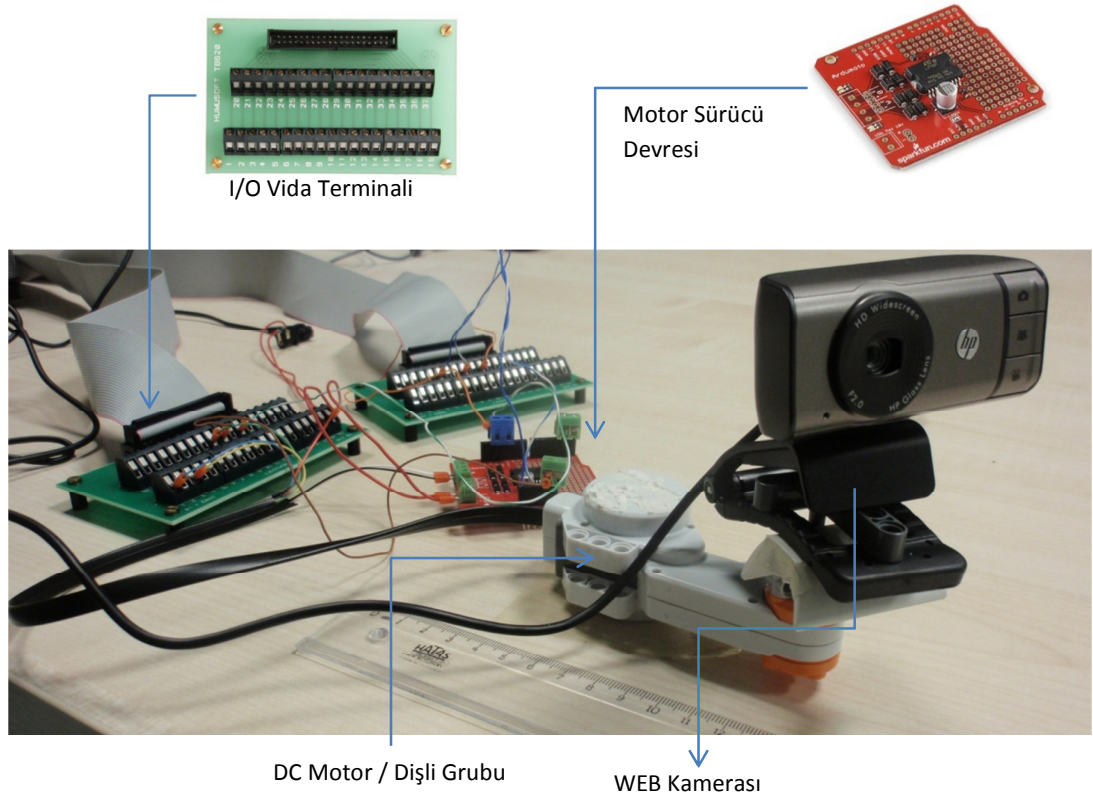
### 3.1.2. Nesne Takip Sistemi

Tezin başlığına uygun olarak, bir mekatronik sistem tasarımının tüm süreçlerini içeren bir örnek uygulama tasarlanmış ve gerçekleştirilmiştir. Bu örnek uygulama “nesne takip sistemi” olarak isimlendirilmiştir. Nesne takip sistemi, yatay düzlemde dönme hareketi yapabilen ve hareketli/durgun bir nesnenin takibini gerçekleştirebilen bir hareketli-kameradır. Belirli bir renge ve alana sahip bir nesneyi görüntü içerisinde algılayabilen, nesnenin merkezi ile görüntü merkezini eşleştirecek şekilde yatay kamera konumunu güncelleyebilen bir görsel-servodur. Sistemin yapısal şeması Şekil 3.9’da, yakından bir görünümü Şekil 3.10’da verilmiştir. Sistemin çalışmasına bir örnek ise Şekil 3.11’de verilmiştir.



Şekil 3.9. Nesne Takip Sisteminin Yapısal Şeması

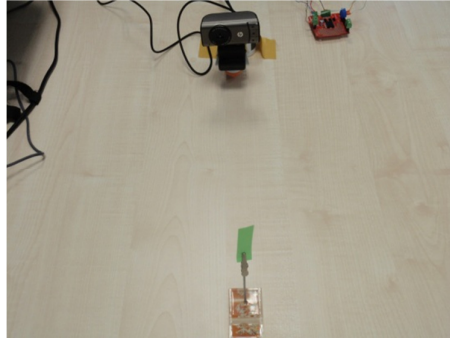




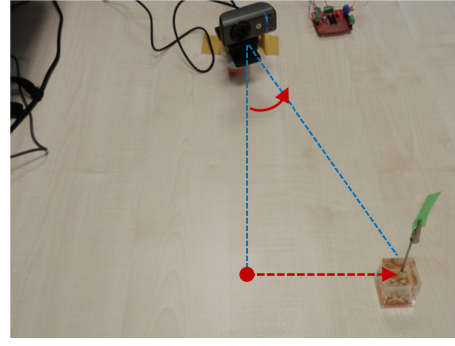
**Şekil 3.10.** Nesne Takip Sisteminin Fiziksel Bileşenleri

Şekil 3.9 ve Şekil 3.10'dan anlaşılacağı üzere hareketli kamera grubu, DC motor/dişli grubu ve bir kameradan oluşmaktadır. Kameranın yatay hareketi DC motor tarafından sağlanmakta ve motor, yönetici bilgisayardan gelen denetim sinyallerine uygun olacak şekilde sürücü devre tarafından sürülmektedir. Kameranın mevcut konumu (ve dolaylı olarak hızı), motor üzerinde bütünleşik olarak bulunan konum algılayıcısı (encoder) ile algılanmaktadır. Motor sürücü devresi, güç kaynağından sağlanan sabit gerilimle beslenmektedir. Görüntü işleme ve hareket denetim algoritmaları, hedef bilgisayar üzerinde gerçek-zamanlı olarak çalışmaktadır. Gerekli sinyal alışverişleri hedef bilgisayar üzerinde bulunan bir I/O kartı tarafından sağlanmaktadır.

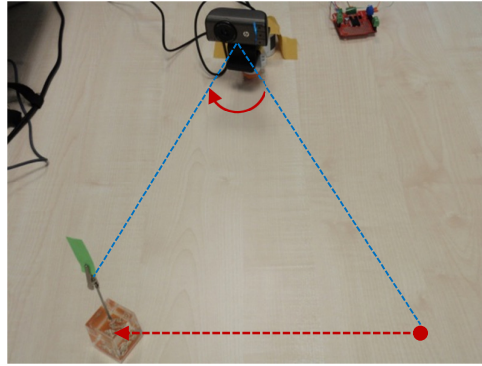
Sistemin çalışma sırasındaki davranışına bir örnek Şekil 3.11'de verilmiştir.



(a) Nesne Başlangıç konumunda



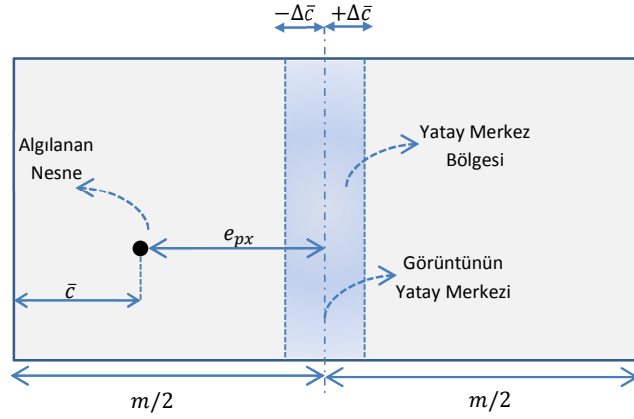
(b) Nesne +x yönünde ilerletilmiş



(c) Nesne -x yönünde ilerletilmiş

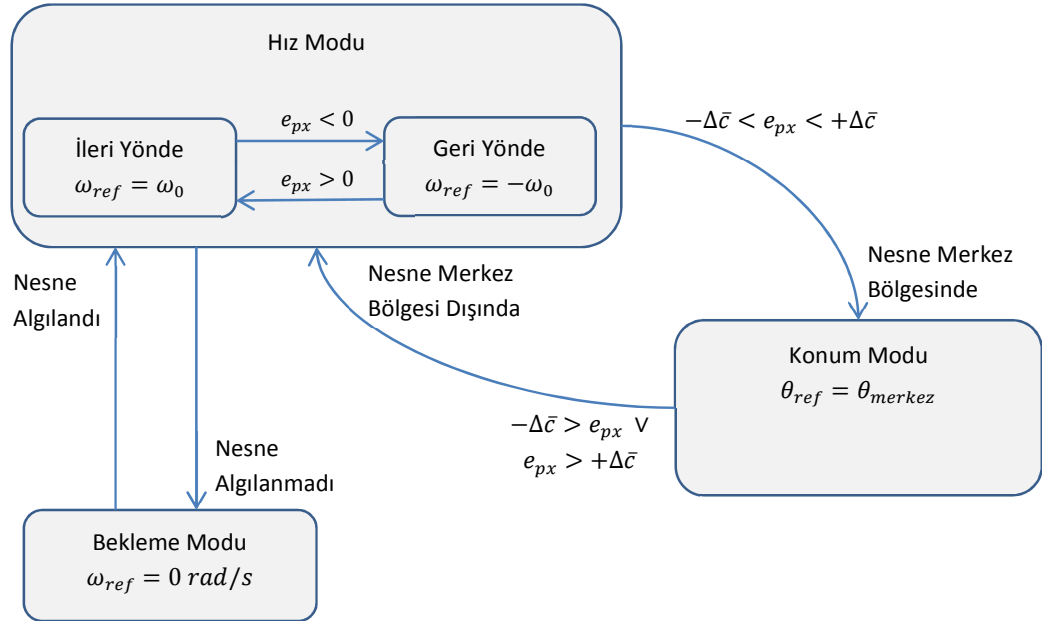
### Şekil 3.11. Makinenin Çalışmasına Bir Örnek

Sistem nesnenin takibini, nesnenin görüntü içerisindeki yatay merkezi ( $\bar{c}$ ) ile görüntü merkezi ( $m/2$ ) arasındaki farkı en aza indirerek gerçekleştirir. Piksel cinsinden hesaplanan bu  $e_{px} = \bar{c} - m/2$  hatası, kameranın yatay açısal hareketi ile telafi edilmektedir. Pratikte, görüntü ve nesne merkezlerini eşleştirmek tam mümkün olmadığından bu işlem, belirli bir  $\mp\Delta\bar{c}$  toleransı ile gerçekleştirilir. Alınan görüntü üzerindeki geometrik gösterimler Şekil 3.12’de verilmiştir.



**Şekil 3.12.** Görüntü üzerindeki geometrik gösterimler

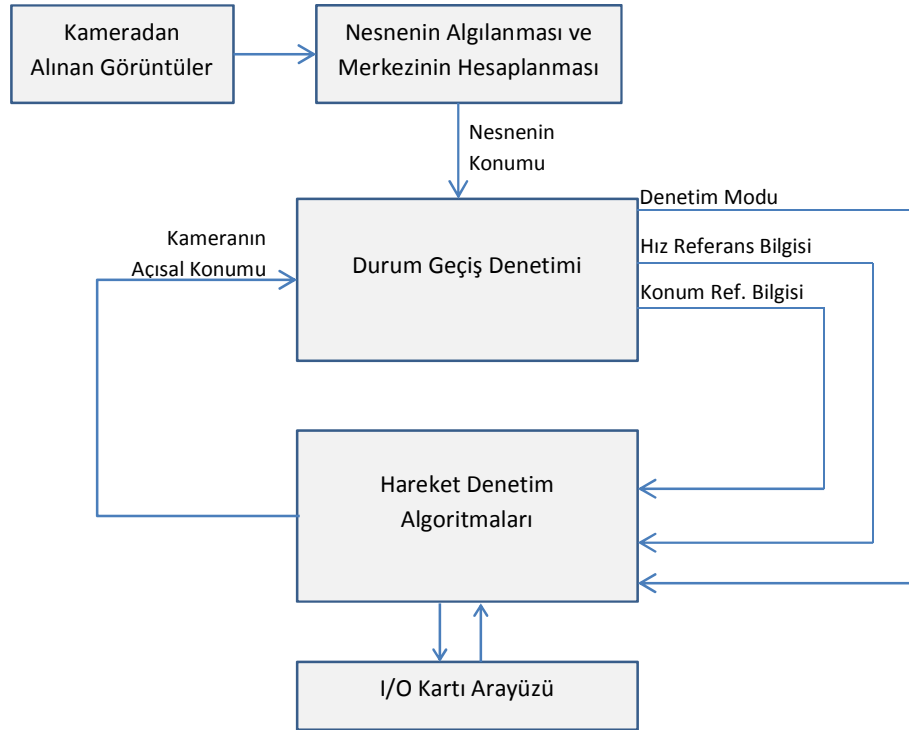
Nesne görüş alanı içerisinde (ve görüntü merkezinin dışında) algılandığında, kamera nesneye doğru sabit bir hızla dönecektir. Nesne ile görüntü merkezleri eşleştiğinde ise kamera o andaki ( $t$  anı), açısal konumunda ( $\theta$ ) kalacaktır. Böylesine bir denetim, Şekil 3.13’de tarif edilen durum geçişleri ile sağlanır.



**Şekil 3.13.** Durum geçiş denetimleri

Şekil 3.13’den görüldüğü üzere sistemin açısal hareketi ya hız denetimi yada konum denetimi ile gerçekleştirilir. Nesne yatay görüntü merkezi dışındayken, sistem hız moduna geçer ve sabit hızla nesneye yaklaşılır (hız denetimi). Nesne yatay görüntü merkezi içerisindeyken, sistem konum moduna geçer ve kamera o anki konumunu korur (konum denetimi).

Sözü geçen bu çalışma prensibi, hedef bilgisayar üzerinde çalışan gerçek-zamanlı bir yazılım ile gerçekleştirilmiştir. Bu program, MATLAB&Simulink paket programı ortamında tasarlanmıştır ve yine bir MATLAB&Simulink eklentisi olan xPC Target ortamında gerçekleştirilmiştir. Geliştirilen programın çalışma akışı Şekil 3.14’de verilmiştir.



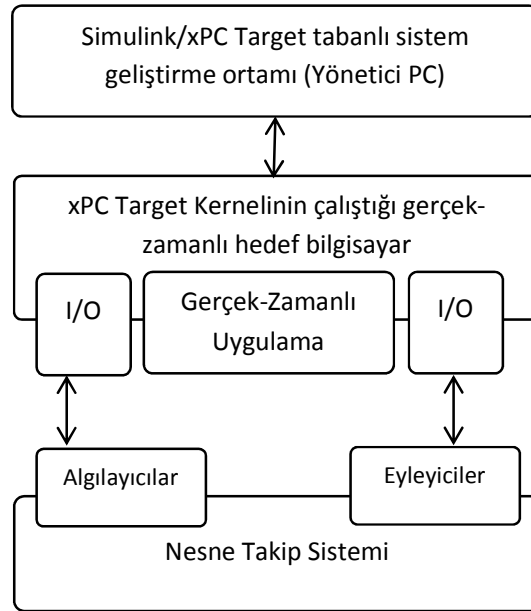
**Şekil 3.14.** Gerçek-zamanlı programın çalışma prensibi

Şekil 3.14’de gösterilen alt-işlevler ileriki başlıklarda ayrıntılı bir şekilde anlatılsa da bu kısımda bütünsel olarak bahsedilebilir: Kameradan alınan görüntüler işlenerek ilgili nesne algılanır ve nesnenin görüntü içerisindeki konumu hesaplanır. Nesne konum bilgisine ve nesnenin varlığına/yokluğuna dayalı olarak makine, hangi “denetim

modu”nda çalışacağına karar vermektedir. Karar verilen denetim moduna göre, hız denetimi yada konum denetimi gerçekleştirilir.

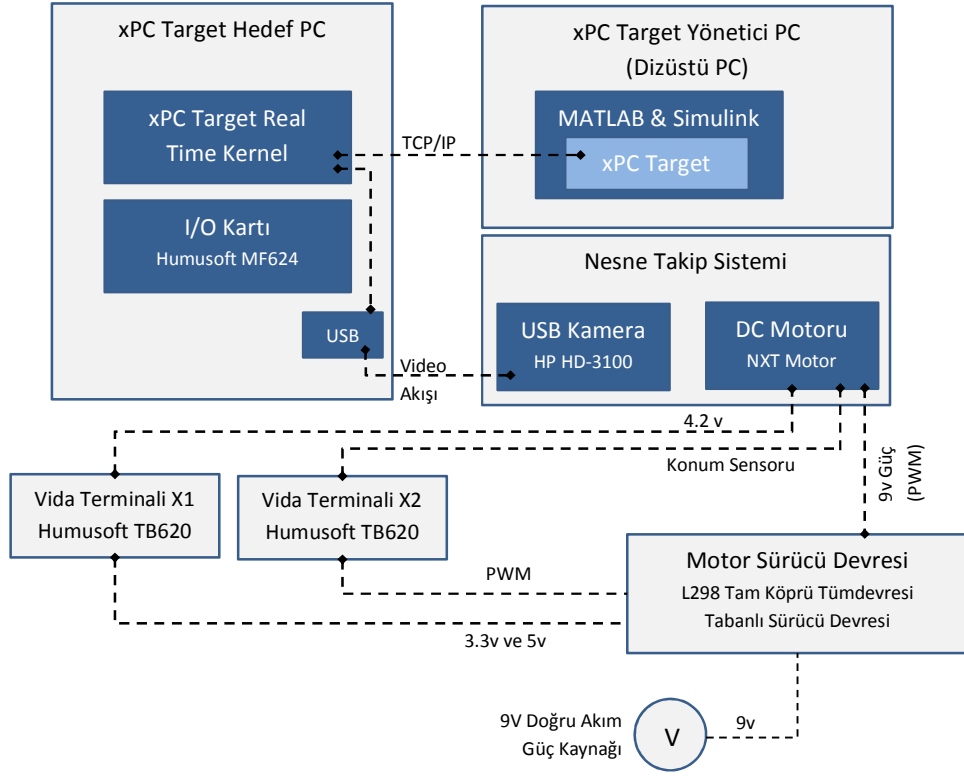
### 3.1.3. Nesne Takip Sistemi Deney Düzeneği

Tez konusu çalışma için bir deney düzeneği kurulmuştur. Deney düzeneğinin çalışma prensip şeması Şekil 3.15’de verilmiştir. Donanımsal, yazılımsal bileşenler ve fiziksel bağlantılar Şekil 3.16’da verilmiş ve 3.1.4. numaralı (sayfa 34) başlıkta ayrıntılı bir şekilde anlatılmıştır. Oluşturulan deney düzeneğinin fotoğrafı Şekil 3.17’de verilmiştir. Deney düzeneğinin bileşenler arası elektriksel bağlantı şeması Şekil 3.18’de verilmiştir.



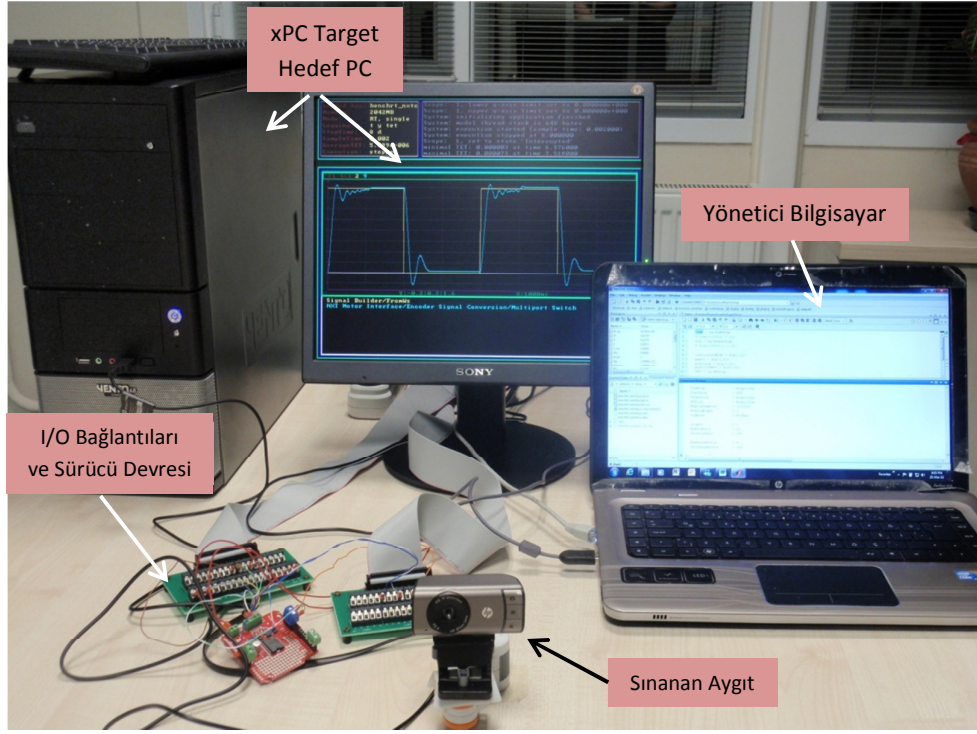
Şekil 3.15. Nesne takip sistemi deney düzeneğinin çalışma prensibi

Şekil 3.15, sayfa 23’de anlatılan *işlevsel hızlı denetim prototipleme* uygulamasının “nesne takip sistemine” uyarlanmış biçimidir. Hızlı denetim prototipleme düzeneği, sayfa 38’de anlatılan xPC Target yapılanmasına uygun oluşturulmuştur.



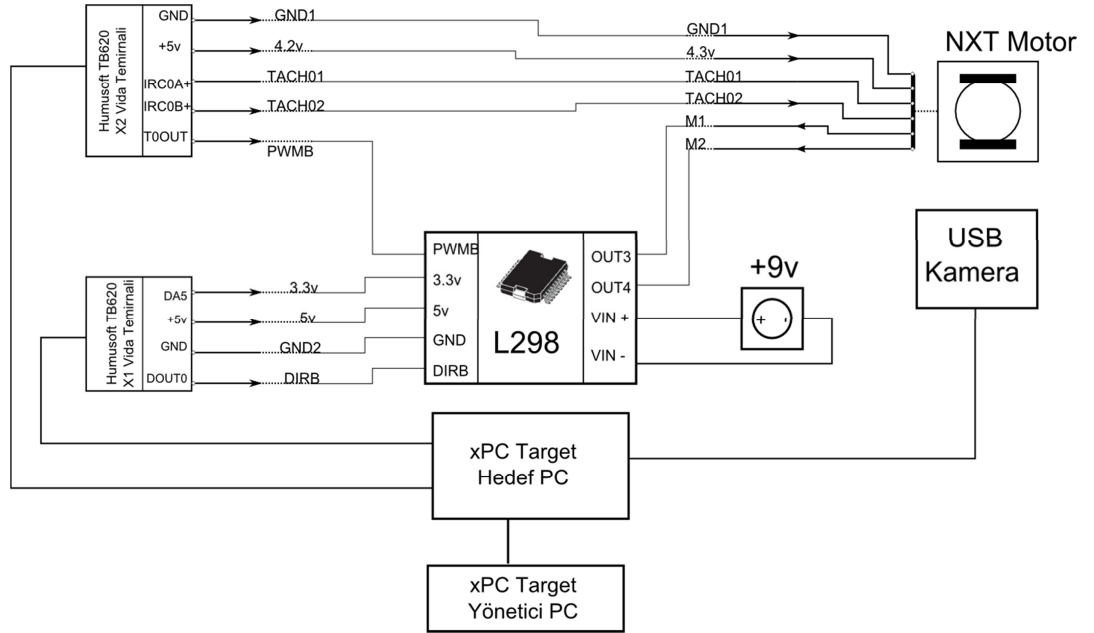
Şekil 3.16. Deney düzeneğine ilişkin donanımsal ve yazılımsal bileşenler

Şekil 3.17’de, Şekil 3.16’da tarif edilen yapılanmaya uygun olarak oluşturulan deney düzeneğinin bir fotoğrafı verilmiştir.



Şekil 3.17. Deney Düzeneginin Bir Görüntüsü

Şekil 3.18’de deney düzeneginin elektriksel bağlantı şeması verilmiştir. Şekil 3.18’de verilen bağlantı kapıları ve kısaltmalar, 3.1.4 başlığında ayrıntılı olarak verilmiştir.



Şekil 3.18. Deney düzenegi elektriksel bağlantı şeması

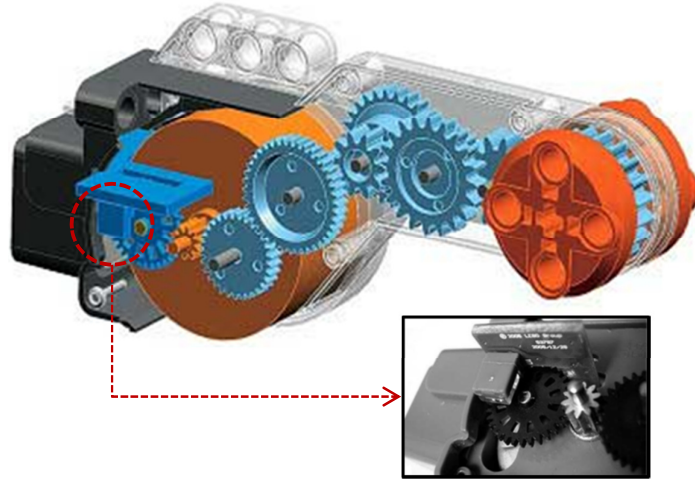
### 3.1.4. Nesne Takip Sisteminin Bileşenleri ve Özellikleri

#### 3.1.4.1. DC Motor Grubu

Nesne takip sisteminde, kameranın açısal hareketi bir DC motor ile sağlanmaktadır. DC motor olarak, Lego tarafından üretilen NXT Motor olarak bilinen bir fırçalı doğru akım motoru seçilmiştir. Bu motorun tercih nedenleri aşağıdaki şekilde sıralanabilir:

- Bütünleşik konum algılayıcısı (encoder), bu çalışmada gerekli olacak açısal konum denetimini mümkün kılmıştır,
- Sahip olduğu bütünleşik indirgeme dişlisi konum denetimini kolaylaştırmıştır,
- düşük güç gereksinimleri, piyasada bulunabilirliği ve ekonomik oluşu diğer nedenlerdir.

Şekil 3.19'da motor ve dişli kutusu grubunun 3B çizimi ve tümleşik konum algılayıcısına ait bir fotoğraf verilmiştir.



Şekil 3.19. NXT motorun CAD montaj görüntüsü ve motora tümleşik optik encoder



Çalışmada kullanılan DC motorun parametreleri Çizelge 3.3’de verilmiştir:

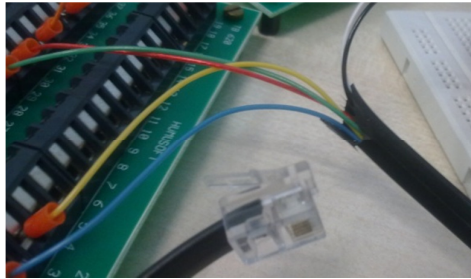
**Çizelge 3.3.** Kullanılan DC motorun parametreleri

Parametre	Değer
Armatür Direnci	$R_a = 6.69 \text{ ohm}$
Armatür Endüktansı	$L_a = 1.4 \cdot 10^{-4} \text{ Henry}$
Eşdeğer zıt EMF katsayısı	$\overline{K_b} = 0.468 \text{ volt} \cdot \text{s/rad}$
Eşdeğer tork sabiti	$\overline{K_t} = 0.317 \text{ Nm/A}$

NXT Motoruna güç ve veri giriş çıkışları, 6 ayrıklı iletkenli bir kablo ile sağlanmaktadır. “NXT kablo”, RJ-12 standartındaki bir konektör (Şekil 3.20) ile sonlanmaktadır. Bu konektörün kablolama yapısı Çizelge 3.4’de verilmiştir (Gasperi 2007).

**Çizelge 3.4.** DC Motor Konektöründeki Kablolama

Pin Numarası	Renk	İsim	Açıklama
1	Beyaz	M1	Motora güç sağlar. En yüksek gerilim 9v olabilir. Doğrudan sürücü elektroniği devresine bağlanır.
2	Siyah	M2	
3	Kırmızı	GND	Toprak ucudur.
4	Yeşil	4.3V GÜÇ	Algılayıcı (encoder) beslemesidir. Sabit 4.3v veren bir gerilim kaynağına bağlanır.
5	Sarı	TACH01	NXT Motorun optik encoder çıkışlarıdır.
6	Mavi	TACH02	



**Şekil 3.20.** Çalışmada kullanılan RJ-12 konektörü

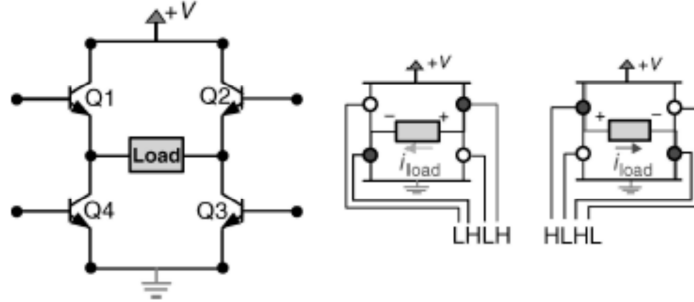
Açısal konum, dahili optik enkoder ile algılanır. Enkoder, kare biçiminde sinyal üretir ve böylece motorun yönü ve hızı belirlenebilir. Bu iki sinyal, kaydırılmış dikdörtgenel

darbelerdir ve bu kayıklık bir fazın çeyreğini temsil eder. Motorun dönüş yönü, bu iki sinyal karşılaştırılarak belirlenir (Gasperi 2007).

### 3.1.4.2. Motor Sürücü Devresi

Denetleyiciden gelen sinyalle motorun çalıştırılabilmesi için yüksek güç sağlayan bir arayüze gereksinim vardır. Bunun için çoğunlukla, ana elemanları güç transistörlerinden oluşan bir akım sürücü devre kullanılır. İleri ve geri yönde çalıştırılabilen bir DC motor için en uygun sürücü devre, H köprüsü biçiminde bağlanmış 4 transistörlü bir devredir.

H-köprüsü devre yapılanması, tek-kutuplu kaynak kullanarak iki-kutuplu uygulamalar gerçekleştirmenin elverişli bir yoludur. Şekil 3.21’de bir yükün sürüldüğü basit bir H-köprüsü devresi verilmiştir.



Şekil 3.21. Genel bir H-Köprüsü devre şeması (Bishop 2007)

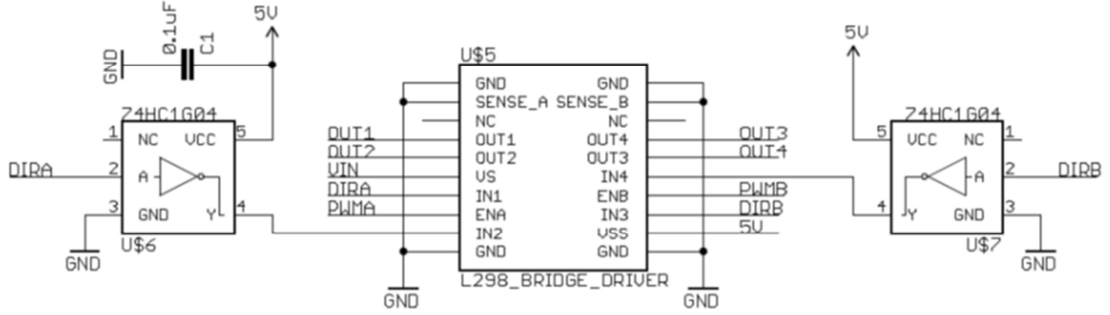
Bir H-köprüsü, Wheatstone köprüsü yapılanmasına uygun dört adet transistörden oluşur. Tek kutuplu bir kaynak kullanarak çift kutuplu bir yükü sürülebilmek için Şekil 3.21’de görülen Q1-Q3 ve Q2-Q4 transistör çiftleri açılıp kapatılır.

Bir çok uygulamada H-köprüsü üzerindeki transistör çiftleri, sayısal bir aygıtın çıkışları (TTL yada CMOS) ile doğrudan sürülebilir. n-p-n ya da n-kanallı transistörler, yüksek seviye (yüksek-TTL) bir sinyal ile açılabilir ya da düşük seviye bir sinyal ile (düşük-TTL) kapatılabilir.

Motoru sürmek için gerekli mantıksal (*logic*) sinyal, H-köprüsüne gönderilen darbe genişlik modülasyonu (PWM) ile sağlanır. Bu teknikte, motora sağlanan akım çok kısa

aralıklarla açılır ve kapatılır. Motorun hızı, kendisine uygulanan ortalama gerilime bağlıdır ve PWM yöntemi bu ortalama gerilimi ayarlamanın bir yoludur.

Bu çalışmada Sparkfun firması tarafından üretilen ve Ardumoto olarak isimlendirilmiş hazır bir PWM sürücü devresi kullanılmıştır. Bu sürücü devrenin şeması Şekil 3.22’de verilmiştir.

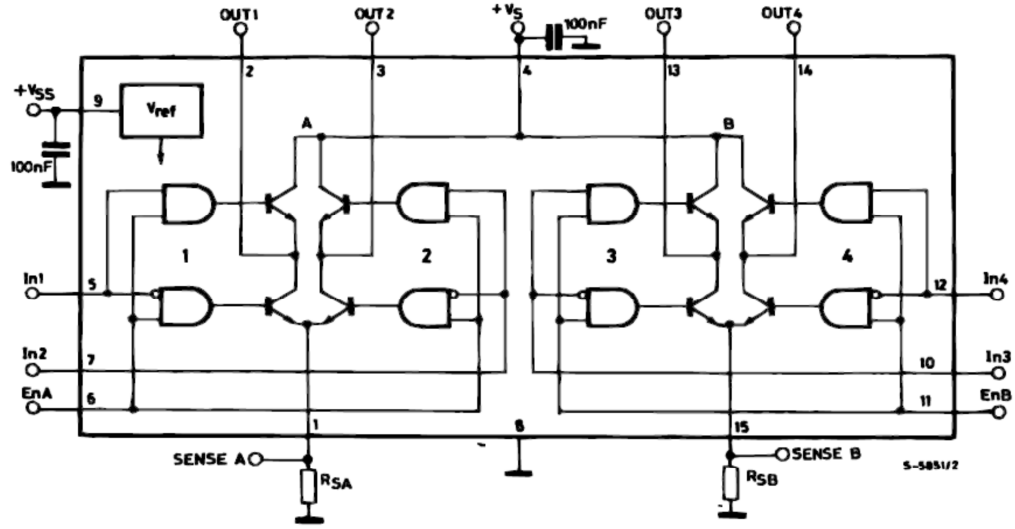


**Şekil 3.22.** Sürücü devre şeması

Şekil 3.22’de verilen OUT1, OUT2 ve OUT3, OUT4 çıkışları, sırasıyla 1. ve 2. motorun gerilim beslemesi olarak kullanılır. Bu çıkışlar VIN giriş beslemesinden elde edilirler. VIN giriş beslemesi, PWMA ve PWMB mantıksal sinyalleri ile anahtlanır ve çıkış OUT1-4 elde edilir.

Yine Şekil 3.22’de verilen DIRA ve DIRB girişleri, OUT1-2 ve OUT3-4 çıkış gerilimlerinin kutuplarını değiştirmek için kullanılır. Bu şekilde motor ileri yada geri yönde sürülebilir. Eğer DIRA ve DIRB girişlerine yüksek-TTL değerinde sinyal gönderilirse motor ileri yönde; düşük-TTL değerinde sinyal gönderilirse motor geri yönde sürülmektedir.

Sürücü devre, ST tarafından üretilen L298 H-köprüsü tümdevresine sahiptir. Bu tüm devrenin şeması Şekil 3.23’de verilmiştir.



Şekil 3.23. Bir L298 tüm devresinin iç şeması

L298 tek yongalı bir tümleşik devredir. Röleler, selenoidler, DC motorlar ve adım motorlar gibi eyleyicileri sürmek için tasarlanmış bir tam-köprü sürücüdür.

Şekil 3.23'den de görüldüğü gibi bu sürücü devre 2 ayrı H-köprü devresine sahip olup, 2A kapasiteli iki ayrı kanala sahiptir ve böylece iki motor aynı anda sürebilir. Yine iki ayrı PWM girişlerine sahiptir ve bu girişlere gönderilen mantıksal sinyaller ile güç kaynağından alınan doğru akım anahtarlanabilir. Devrenin en yüksek anahtarlama gerilimi 20 voltur.

Bu sürücü devre, NXT motorun akım ve gerilim gereksinimlerine uygundur. Ayrıca, kullanılan I/O kartının sayısal çıkışları ve sürücü kartının girişleri (PWM ve DIR) TTL seviyesindedir. Bu da I/O kartı ile sürücü devreyi birbirine uyumlu kılmaktadır.

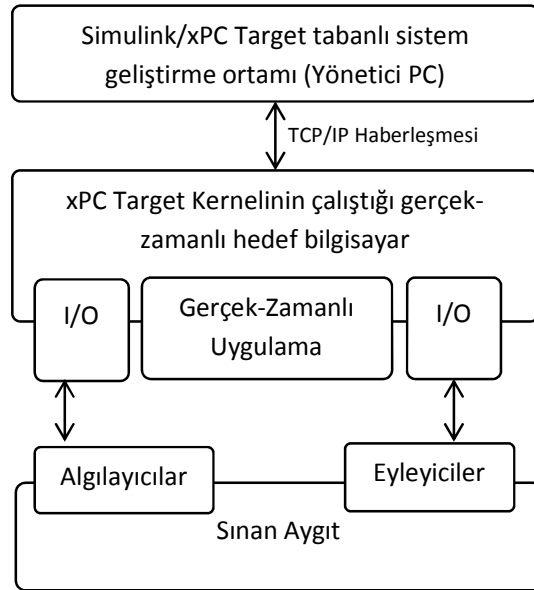
### 3.1.4.3. xPC Target Yazılımı ve Hedef/Yönetici PC

Tezde konusu olan makineye ilişkin denetim sistemlerinin tasarlanması, gerçek-zamanlı yazılımların geliştirilmesi ve çalışma sonrası analizler MATLAB&Simulink ortamında gerçekleştirilmiştir. Bu geliştirme ortamı 3.1.5 (sayfa 43) numaralı başlıkta ayrıntılı olarak verilmiştir. Tez çalışması için tasarlanan ve çalışma prensibi Şekil 3.14'de verilen (sayfa 30) gömülü algoritmaların gerçek-zamanlı olarak test edilmesi ve

çalıştırılmasına olanak tanıyan bir ortama gereksinim vardır. Gerçek-zamanlı test ve çalışma ortamı olarak bir Simulink eklentisi olan xPC Target yazılımı tercih edilmiştir.

xPC Target, Simulink ortamında geliştirilen algoritmaların gerçek-zamanlı olarak test edilmesine olanak tanıyan bir yazılım ortamıdır. 3.1.1.5 numaralı başlıkta ve Şekil 3.6'da (sayfa 22) özetlenen gerçek-zamanlı test uygulamaları xPC Target ile gerçekleştirilebilir.

xPC Target uygulamasında, birbirleri ile haberleşen hedef-yönetici (*host-target*) PC yapılanması mevcuttur. Bu yapılanma genel bir biçimi Şekil 3.24'de verilmiştir.



**Şekil 3.24.** xPC Target ile Gerçek Zamanlı Test yapılanması

Hedef bilgisayar, gerçek zamanlı bir işletim sistemi çekirdeği<sup>1</sup> (*kernel*) ile çalışır ve Windows, Linux, DOS vs. gibi herhangi bir işletim sistemine gereksinim duymaz. Bu *kernel* üzerinde, Simulink modellerinden oluşturulan *gerçek-zamanlı uygulamalar*<sup>2</sup> çalışır. Hedef bilgisayar çevresel fiziksel bileşenlerle (algılayıcılar ve eyleyiciler) sinyal alışverişlerini sağlayan I/O kartlarını da üzerinde bulundurur.

<sup>1</sup> xPC Target Real-Time Kernel. xPC Target yazılımı tarafından sağlanır ve hedef bilgisayar bu gerçek-zamanlı işletim sistemi ile başlatılır.

<sup>2</sup> Bir *gerçek zamanlı uygulama* (*real-time application*) xPC Target Real-Time Kernel üzerinde doğrudan çalışır ve Simulink modellerinden otomatik olarak üretilen C kodlarının derlenmesi ile oluşturulur. Derleyici olarak Microsoft Visual C++ yada openWATCOM C derleyicisi kullanılabilir. Uygulama, geliştirilen gömülü algoritmaları ve kullanılan I/O kapılarının sürücülerini içerir.

Yönetici PC, Simulink modellerinin geliştirildiği kullanıcı bilgisayardır. Hedef bilgisayar tamamen yönetici PC tarafından kontrol edilir. Çalışma öncesinde, *gerçek-zamanlı uygulamalar* yönetici PC tarafında oluşturulur ve hedef bilgisayara yüklenir. Çalışma sırasında ve öncesinde gerçek-zamanlı uygulamanın parametreleri yönetici PC üzerinde canlı olarak değiştirilebilir (*online parameter tuning*). Çalışma sırası ve sonrasında hedef bilgisayar tarafından sağlanan sinyal verileri işlenmek üzere yönetici PC'ye aktarılabilir.

Yönetici-hedef PC arasındaki haberleşme, TCP/IP protokolü ile LAN üzerinden sağlanır.

Tez çalışmasında kullanılan hedef bilgisayar, Intel Core2Duo mikroişlemcisine ve 2 GB RAM'e sahiptir (Şekil 3.25). xPC Target yazılımının 5.2 (R2012a) sürümü kullanılmıştır. Yönetici PC olarak HP Pavilion dv6-3102et dizüstü PC kullanılmıştır. I/O kartı olarak Humusoft MF624 çok amaçlı I/O kartı kullanılmıştır.

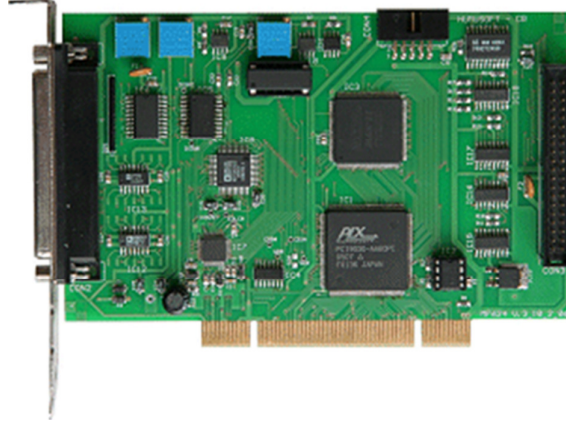


Şekil 3.25. Çalışmada kullanılan xPC Target hedef bilgisayar

#### 3.1.4.4. Çok Amaçlı I/O Kartı

Hızlı denetim prototipleme uygulamalarında, gerçek-zamanlı hedef bilgisayarın algılayıcılardan verileri alabilmesi ve eyleyicilere hesaplanan denetim sinyallerini gönderebilmesi gerekir. Çevresel birimlerle sinyal alışverişlerinin gerçekleştirilebilmesi için bir I/O kartına arayüz elemanı olarak gereksinim duyulur. Bu tez çalışmasında, xPC Target hedef bilgisayar üzerinde Humusoft firması tarafından geliştirilen çok amaçlı I/O kartı (Şekil 3.26) kullanılmıştır. MF624 kartı servo denetimi uygulamalarına uygun PCI tabanlı bir karttır. Bu tez çalışmasında gerekli olan kanallara -bir başka deyişle- A/D ve

D/A dönüştürücülerine, encoder girişlerine, sayısal I/O ve zamanlayıcı/sayıcı kapılarına yeterli sayıda sahiptir.



**Şekil 3.26.** Humusoft MF624 I/O kartı

MF624 kartının, bu tez çalışmasında tercih edilmesinin nedenleri aşağıdaki şekilde sıralabilir:

- MATLAB ortamı ve xPC Target yazılımı ile tam uyumludur,
- çalışmada kullanılan DC motorun optik encoder geri beslemesine uygun encoder girişlerine sahiptir,
- sürücü devre kartı girişi için gerekli sinyallere uygun PWM çıkışı sağlar,
- konum algılayıcısını destekleyecek güçte akım çıkışına sahiptir,
- diğer yardımcı birimlere gönderilen sayısal çıkışlara sahiptir.

MF624 kartı üzerinde, tez çalışmasındaki kameranın açısal hareketini sağlayan DC motorun denetiminde kullanılan I/O kapıları Çizelge 3.5’de özetlenmiştir.

**Çizelge 3.5.** Tez çalışması için MF624 kartında kullanılan I/O kapıları

I/O	Kullanım
D/A dönüştürücüler	Aygılayıcı (encoder) beslemesi
Sayısal çıkış	Motor yön denetimi
Encoder girişleri	Konum algılayıcısı (encoder) geri beslemesi
Zamanlayıcılar/Sayıcılar	PWM sinyalinin oluşturulması (DC motor sürücü devresi için)

Motor sürücüsüne sağlanacak PWM sinyali, hedef bilgisayar üzerinde çalışan gömülü yazılım tarafından hesaplanan frekansta ve doluluk oranında donanımsal olarak üretilir ve sayıcı çıkışlarına gönderilir. PWM sinyalinin doluluk oranı  $0 \leq duty \leq 1$  değeri ile belirlenir.  $duty = 0$  iken, sayıcı çıkışı %100 düşük TTL seviyesinde iken,  $duty = 1$  iken %100 yüksek TTL seviyesindedir (Anonim, 2006).

MF624 kartı, hedef bilgisayarın PCI yuvası üzerinde yer aldığı için dış ortam bağlantısı (algılayıcı ve eyleyici kabloları) ilave bir vida terminali ile sağlanmaktadır. Bu çalışmada, yine Humusoft firmasının üretmiş olduğu TB620 evrensel terminal kartı kullanılmıştır.

#### **3.1.4.5. Web Kamerası ve Örnek Nesne Fotoğrafları**

Tez çalışmasındaki nesne takip sisteminde, görüntüdeki nesneye ilişkin konum vs. gibi unsurların hesaplanabilmesi için önce görüntülerin gerçek zamanlı olarak denetim sistemine beslenmesi gerekir. Bu nedenden, görüntülerin toplanmasına yönelik bir kameraya gereksinim vardır. Bu tez çalışmasındaki denetim algoritmalarının gerçekleştirildiği xPC Target ortamı, CameraLink arayüzünü kullanan kameraları yada USB Web kameralarını desteklemektedir. Çalışmada, kolay bulunabilirliği ve ekonomik olması nedeni ile HP HD-3100 web kamerası tercih edilmiştir. HD-3100, 720 piksel ekran genişliğine ve 2.0 mm odak uzaklığına sahiptir.

Çalışmada kullanılan gerçek-zamanlı video işleme algoritmaları ilk önce sabit fotoğraflar üzerinde prototiplenmiştir. Uygun video işleme algoritmalarını tasarlayabilmek için Şekil 3.30'da verilenlere (sayfa 50) benzer görüntüler alınmıştır ve kullanılmıştır.



### 3.1.5. Kullanılan Geliştirme Araçları

#### 3.1.5.1. Sistem Modelleme ve Benzetim Araçları

Denetlenen sistemlerin modelleri oluşturulurken MATLAB & Simulink, SimScape ve SimElectronics yazılımları tercih edilmiştir.

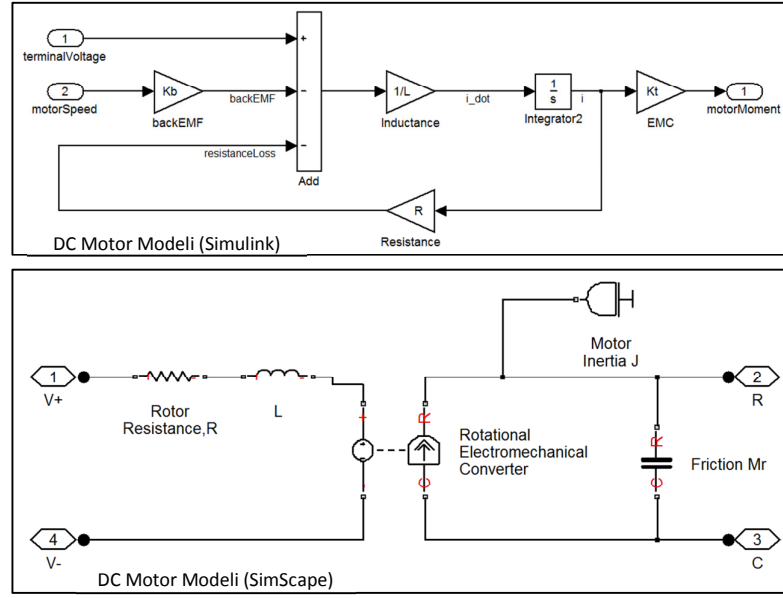
MATLAB, MathWorks firması tarafından geliştirilen bir bilgisayar yazılımıdır. Algoritma geliştirmeye, verilerin görselleştirilmesine, verilerin analizine ve sayısal hesaplamalara yönelik yüksek-seviyeli bir programlama dili ve bütünleşik geliştirme ortamıdır. Bu unsurlarından dolayı tez çalışmasında tercih edilmiştir.

Simulink bir MATLAB eklentisi olup, dinamik ve gömülü sistemlerin model tabanlı tasarımına yönelik bir ortamdır. Haberleşme, kontrol, sinyal işleme, video ve görüntü işleme gibi zamana-bağlı sistemlerin tasarımına, benzetimine, gerçekleşmesine ve sınanmasına yönelik etkileşimli grafiksel bir ortamdır. Simulink ortamında modelleme, “*sinyal akışları grafikleri*” ve “*blok diyagramları*” ilkesine benzer şekilde gerçekleştirilmektedir.

Nesne takip sisteminin sahip olduğu dinamik altsistemlerin tasarımına, benzetimine ve doğrulanmasına yönelik tercih edilmiştir.

**SimScape** bir Simulink eklentisidir ve fiziksel sistemlerin matematiksel denklemlerine gereksinim olmaksızın modellenmesine olanak tanıyan bir kütüphanedir. Simulink ortamında dinamik sistemleri *sinyal akışları* ilkesine göre modellenirken, SimScape kütüphanesi ile fiziksel sistemler, elemanlar arasındaki enerji alış/verişini temsil eden “fiziksel bağlantılar” ilkesine göre oluşturulur (Şekil 3.27). Mekanik sistemleri modellemek için kütle, eylemsizlik vs., elektriksel sistemleri modellemek için direnç, kapasite, endüktans vs. gibi alt-seviye bileşenlere sahiptir. **SimElectronics**, SimScape’in bir alt kütüphanesidir ve elektronik sistemlerin modellenmesine olanak tanıyan sürücüler ve eyleyiciler, tümleşik devreler, pasif aygıtlar, yarı iletkenler vs. gibi bileşenlere sahiptir.

Bu tez çalışmasında, DC motorun sürücüsünün modellenmesinde SimScape ve SimElectronics kütüphanesinden faydalanılmıştır.



**Şekil 3.27.** Simulink ve SimScape'in karşılaştırılması

Simulink ortamında geliştirilen modeller değeri tam olarak bilinmeyen parametreler içeriyor olabilir ve bu bilinmeyen parametreler deney verilerinden tahmin edilmesi gerekebilir. Aynı zamanda bir model yanıtının belirli kısıtlar içerisinde olması istenebilir ve bu kısıtları sağlayan model parametrelerinin hesaplanması gerekebilir. Yaygın olarak sırası ile *parametre kestirimi* ve *yanıt optimizasyonu* olarak bilinen bu uygulamaların gerçekleştirilebilmesi için MathWorks firması tarafından geliştirilen ve bir MATLAB eklentisi olan **Simulink Design Optimization** aracı kullanılmıştır.

*Parametre kestirimi*, sistemin geçici-durum yanıtını içeren ölçüm verileri ile gerçekleştirilir. Yaygın olarak, adım yada ani-darbe (*impulse*) giriş sinyaline karşılık sistem yanıtı ölçülür. Ölçüm verileri ile benzetim sonuçları, kullanıcı tarafından belirlenen maliyet (*cost*) fonksiyonu minimize edilene kadar yinelemeli olarak (*iteratively*) karşılaştırılır. Bu karşılaştırma yaygın olarak, en küçük kareler hatasının hesaplanması ile gerçekleştirilir. Bu optimizasyon tabanlı uygulama, veri-yönelimli matematiksel modellemeye yardımcı olmaktadır. Bu yöntem, elde edilen sonuçlardan ya da varılması gereken sonuçlardan yola çıkarak, model parametrelerini geriye dönük olarak hesapladığı için, bazı yazılı kaynaklarda *gri kutu modelleme* olarak da geçmektedir.

Nesne takip sistemi uygulamasında, DC motorun doğrudan hesaplanamayan bazı parametrelerin kestirilmesi gerekli olmuştur. Bu kapsamda, DC motora ait eşdeğer sönüm oranı ( $B_{eq}$ ) ve eşdeğer eylemsizlik momenti ( $J_{eq}$ ) parametreleri deney düzeneği üzerinden toplanan veriler ile *parametre kestirimi* yöntemleri ile hesaplanılmıştır.

Bu tez çalışmasında parametre kestirimi çalışması yapılırken aşağıdaki adımlar takip edilmiştir.

- Elektromekanik altsistemin (sürücü ve DC motor/dişli grubu) matematiksel modeli Simulink ortamına taşınmıştır,
- birim basamak, sinüs vs. gibi giriş test sinyalleri tanımlanmıştır,
- gerçek elektromekanik altsisteme gönderilen test sinyallerine (denetçi sinyalleri) karşılık konum verileri kaydedilmiştir,
- $B_{eq}$  ve  $J_{eq}$  için başlangıç tahmin değerleri belirlenmiştir,
- optimizasyon yöntemi, toleranslar vs. gibi çözücü ayarları belirlenmiş ve optimizasyon işlemi başlatılmıştır,
- optimizasyonun iterasyonları, aynı test sinyalleri ile çalıştırılan Simulink benzetim modelinin yanıtı ile deney düzeneğinden alınan sistem yanıtı belirli toleranslar dahilinde eşleşene kadar, bir başka deyişle *cost* fonksiyonu minimize edilene kadar tekrarlanmıştır.
- kestirilen parametreler, çalışmada kullanılmak üzere kaydedilmiştir.

### 3.1.5.2. Denetim Sistemi Tasarımı Araçları

Nesne takip sisteminin denetleyici tasarımında, bir MATLAB & Simulink eklentisi olan **Control System Toolbox** , **Simulink Control Design** ve **Stateflow** araçları kullanılmıştır.

**Control System Toolbox**, kontrol sistemlerinin tasarımına ve analizine yönelik işlemlere ve grafiksel arayüzlere sahip bir MATLAB eklentisidir. Doğrusal ve zamanla-değişmeyen (LTI) denetim sistemlerine yönelik bir çok ön-tanımlı fonksiyonlar içerir. **Simulink Control Design** ise Simulink ortamında modellenen kontrol sistemlerinin tasarımına analizine olanak tanıyan bir eklenti paketidir.

Bu tez çalışmasında, hareketli kameranın konum ve hız denetimine yönelik algoritmaların tasarımında kullanılmıştır.

**Stateflow**, durum şemalarının ve akış grafiklerinin geliştirilmesine yönelik bir yazılım ortamıdır ve mantıksal durum geçişleri içeren gömülü denetim sistemlerinin tasarımına olanak tanır. Durumların ve karmaşık mantıksal akışların modellenebilmesi için etkileşimli grafiksel bir ortam sağlar.

Şekil 3.13’de (sayfa 29) tarif edilen durumlar arasındaki geçişlerin modelleyebilmek için Stateflow tercih edilmiştir.

### 3.1.5.3. Görüntü ve Video İşleme Araçları

Makine ile görme (*machine-vision*) uygulamalarında video işleme algoritmalarının tasarımı ve gerçekleştirilmesi önemli bir rol oynar. Bu algoritmalar MATLAB, Halcon ve NeuroCheck, Cognex vs. gibi ticari paket program ortamında geliştirilebilir ve M dili<sup>3</sup>, C, C++, C#, OpenCV, ITK, VTK vs. gibi yazılım dili ve kütüphaneleri ile gerçekleştirilebilirler.

Nesne takip sistemi uygulamasında ise, görüntü içerisindeki nesneye ilişkin konum vs. gibi unsurların hesaplanması gerekli olmuştur. Bu nedenle görüntülerin işlenmesine, analizine, görselleştirilmesine ve bunlara yönelik algoritmaların geliştirilmesine yönelik bir MATLAB eklentisi olan **Image Processing Toolbox** tercih edilmiştir. Tez çalışmasında kullanılan gerçek-zamanlı video işleme algoritmalarının (Şekil 3.29, sayfa 49) tasarımında ise yine bir MATLAB eklentisi olan **Computer Vision Toolbox** kütüphanesi kullanılmıştır.

### 3.1.5.4. Otomatik Kod Oluşturma Araçları

MATLAB ve Simulink ortamında geliştirilen algoritmaların gerçek-zamanlı uygulamalara dönüştürülebilmesi için otomatik kod dönüşümü gereklidir. **MATLAB Coder** ve **Simulink Coder**, bu kod dönüşümünü sağlayan araçlardır. Elde edilen C kodu, hedef donanıma yönelik üretilir. Bu çalışmada xPC Target gerçek-zamanlı çekirdeğine yönelik kod üretimi gereklidir ve bu otomatik kod dönüşümü araçları gerekli olmuştur.

---

<sup>3</sup> MATLAB dili

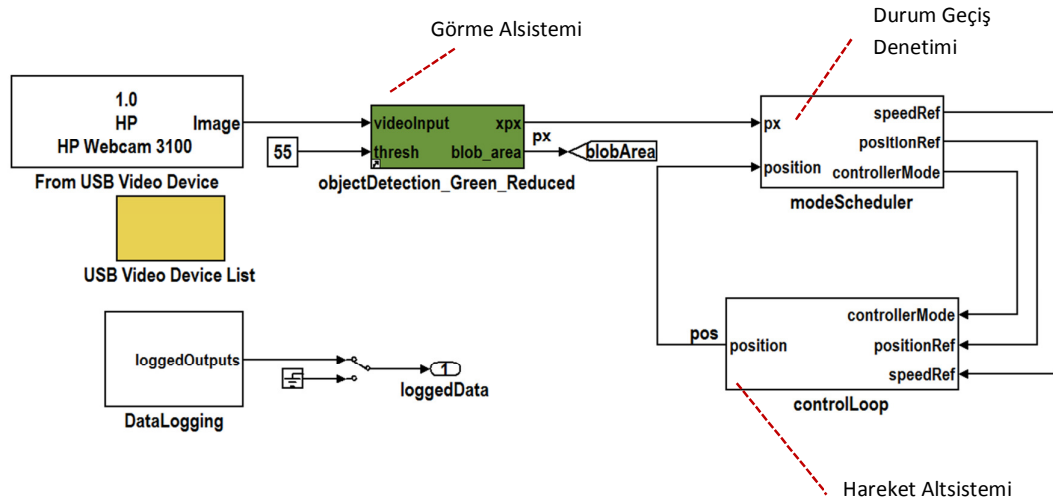
MathWorks tarafından geliştirilen **MATLAB Coder** ve **Simulink Coder** araçlarının 2011 yılı sürümünden önceki “Real-Time Workshop” olarak isimlendirilmişti. 2011 yılından itibaren Realtime Workshop, iki ayrı ürün olarak mevcuttur ve işlevleri Çizelge 3.6’da özetlenmiştir:

**Çizelge 3.6.** MATLAB Coder ve Simulink Coder İşlevleri

MATLAB Coder	Embedded MATLAB Language (EML) dili ile yazılmış fonksiyonları gömülebilir ve hedeften bağımsız C-kodlarına dönüştürür.
Simulink Coder	Simulink modellerini gömülebilir ve hedeften bağımsız C kodlarına dönüştürür.

### 3.2. Yöntem

Nesne takip sistemi için geliştirilen gerçek zamanlı programın prensip şeması Şekil 3.14’de (sayfa 30) özetlenmişti. Anlaşıldığı üzere bu program görme, hareket ve “durum geçiş denetimi” alt sistemlerinden oluşmaktadır. Bu alt sistemler matematiksel olarak ayrı ayrı tasarlanmış, Simulink ortamında modellenmiş ve daha sonra tek bir model içerisinde bütünleştirilmiştir (Şekil 3.28).



Şekil 3.28. Gerçek-zamanlı programın Simulink modeli

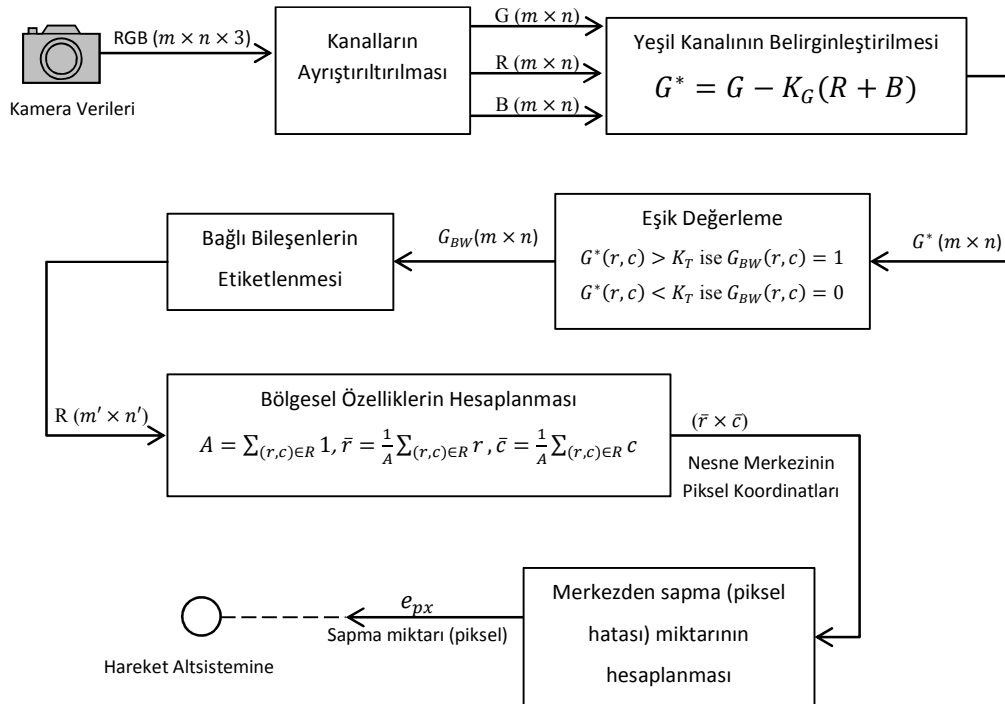
Görme alt sisteminde görüntüye sırayla eşik-değerleme ve bağlı-bileşenlerin etiketlendirilmesi yöntemleri uygulanmıştır ve daha sonra blob analizi<sup>4</sup> yöntemleri ile nesneye ilişkin unsurlar hesaplanmıştır. Hareket alt sistemi ise tek serbestlik dereceli dönel bir mekanizma ve ona ilişkin denetim algoritmalarından oluşmaktadır. Mekanizmanın dinamiği Newton’un ikinci yasası ile türetilmiş ve hareket denetimi ise PID algoritması ile tasarlanmıştır.

<sup>4</sup> Bir görüntü içerisinde hesaplanan bağlı bileşenlerin her birine blob ismi verilmektedir. Blob analizi ise, bu bağlı bileşenlerden ağırlık merkezi, alan vs. gibi unsurların hesaplanmasına yönelik yöntemler bütünüdür.

### 3.2.1. Görme Altsistemi

Bu tez çalışmasında ele alınan nesne takip sistemi, görüntü içerisindeki belirli bir renge ve alana sahip nesnelere ayırt edip algılayabilen ve kameranın açısal konumunu bu nesneyi merkezleyecek şekilde güncelleyen bir görüntü-tabanlı görsel servodur. Bunun için, görüntüleri ortama alabilecek, sözü edilen nesneyi görüntü içerisinde ayırt edip algılayabilecek, nesnenin konum bilgilerini ve konum hatasını (merkezden uzaklığını) hesaplayıp hareket altsistemine geri besleyebilecek bir **görme altsistemine** gereksinim duyulur.

**Görme altsistemi** kameradan görüntülerin alınmasına, görüntülerin yorumlanmasına ve hareket altsistemine geri beslenecek konum hatası değerlerinin hesaplanmasına olanak tanıyan donanımsal ve yazılımsal bileşenlerden oluşur. Görüntülerin yorumlanmasına olanak tanıyan yazılım xPC Target hedef bilgisayarı üzerinde gerçek zamanlı olarak çalışırken, görüntü verileri bir web kamerası ile alınmaktadır. Görme altsistemi için geliştirilen yazılımın genel algoritmik akışı Şekil 3.29'da verildiği gibidir:



Şekil 3.29. Görme Altsisteminin İşlevsel Şeması

Şekil 3.29'daki işlevsel şemaya göre video işleme yazılımı prototiplenirken, Şekil 3.30'da bir örneği verilen nesne fotoğraflarından faydalanılmıştır.



**Şekil 3.30.** Örnek nesne fotoğrafı

Kameradan giriş olarak alınan görüntüler her bir görüntü karesi için  $m \times n \times 3$  elemanlı ( $m \in Z^+$  ve  $n \in Z^+$  olmak üzere) üç boyutlu bir matris şeklinde hedef bilgisayar ortamına alınırlar. Burada  $m$  alınan görüntünün piksel cinsinden genişliği,  $n$  ise yüksekliğidir. Görüntü matrisinin üçüncü boyutuna ilişkin her bir katman ayrı bir renk kanalını temsil eder.  $m \times n$  elemanlı R, G ve B matrisleri sırası ile alınan görüntünün kırmızı (R), yeşil (G) ve mavi (B) kanallarını temsil eder. Özgün görüntü ve görüntünün yeşil kanalına bir örnek Şekil 3.31'de verilmiştir.



**Şekil 3.31.** Özgün Görüntü (Solda) ve Yeşil Kanal (Sağda)

Konum hesaplamaya yönelik algoritma belirli bir eşik üzerindeki yeşil nesnelere için tanımlandığı için, yeşil kanal (G) belirli bir eşik değeri ile karşılaştırılır.

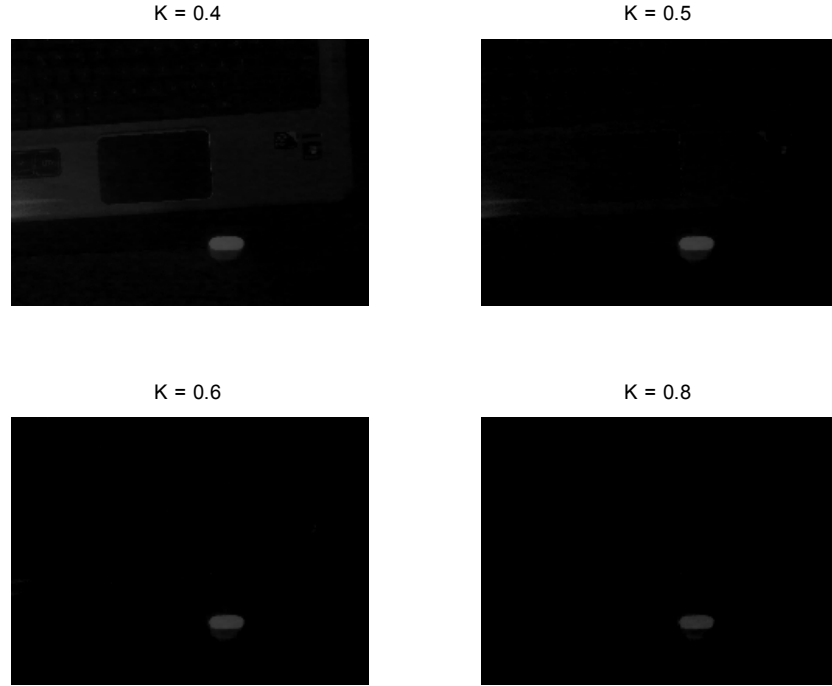


Eşik değeri ile karşılaştırmadan önce yeşil renkli pikseller basit bir aritmetik işlem ile belirginleştirilir:

$$G^* = G - K_G(R + B) \quad (3.1)$$

Yeşil (G) kanalını temsil eden matristen, kırmızı (R) ve mavi (B) kanalı temsil eden matrislerin toplamının belirli bir ölçekte ( $K_G$ ) farkı alındığında belirginleştirilmiş yeşil kanalına ilişkin matris ( $G^*$ ) hesaplanabilir.

$K_G$  katsayısı deneysel olarak hesaplanmıştır. Yukarıdaki aritmetik işlem, örnek bir görüntü üzerine uygulanarak aşağıdaki görsel sonuçlar edilmiştir. Farklı  $K_G$  değerlerine ilişkin görsel sonuçlar Şekil 3.32’de verilmiştir.



**Şekil 3.32.** Farklı ölçekleme katsayıları altında yeşil kanal

Görsel bir inceleme sonucunda  $K_G = 0.6$  olarak belirlenmiştir. Yeşil nesne görüntünün geri kalanına göre belirginleştirildikten sonra **eşik değerlendirme** işlemi uygulanır.

Görüntü zıtlığı ve parlaklığın değiştirilmesi ile birlikte kullanıldığında eşik değerlendirme tekniği nesnelere arka plandan ayırmak için kullanılabilir. Eşik değerlendirme işlemi, görüntü matrisindeki her bir değeri bir eşik değeri ile karşılaştırarak gerçekleştirilir. Bu

işlem sonunda görüntü ikili (*binary*) değerlere (sadece siyah ve beyaz) sahip bir  $G_{BW}$  matrisine dönüşür.

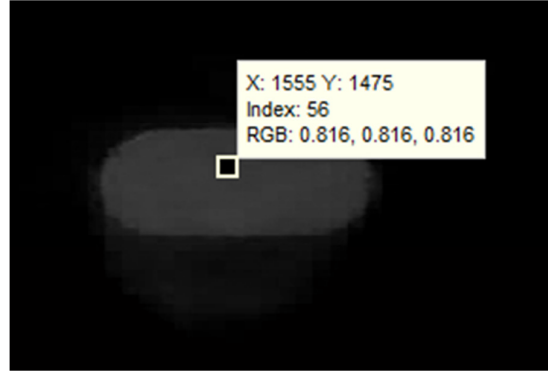
$G^*(r, c)$  ve  $G_{BW}(r, c)$  sırası ile  $G^*$  ve  $G_{BW}$  matrislerine ait  $r$  satırı ve  $c$  sütunundaki değeri temsil etsin.  $m \times n$  bir görüntü için  $r = 0 \dots (m - 1)$  ve  $c = 0 \dots (n - 1)$  aralığında olacaktır. Örneğin,  $G^*(0,0)$  en üst sol pikselin değerini temsil ederken  $G^*(m - 1, n - 1)$  en alt sağ pikselin değerini temsil edecektir.

$\forall r$  ve  $\forall c$  için

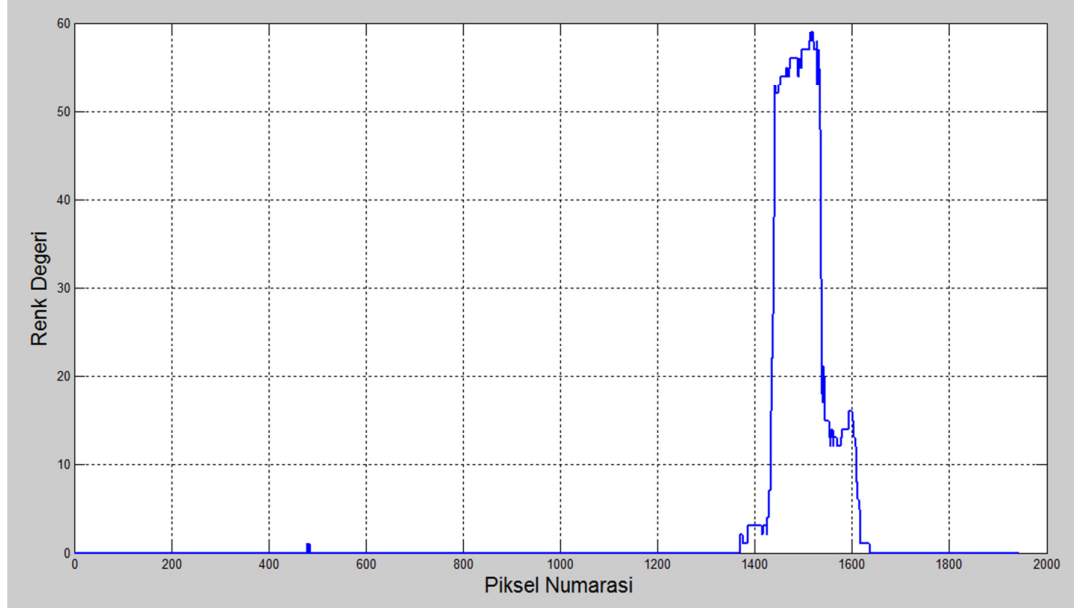
$$\begin{aligned} G^*(r, c) > K_T \text{ ise } G_{BW}(r, c) &= 1 \\ G^*(r, c) < K_T \text{ ise } G_{BW}(r, c) &= 0 \end{aligned} \quad (3.2)$$

ile  $G_{BW}$  ikili matrisi hesaplanabilir. Burada  $K_T$  eşik değeridir. Eşik değeri bir önceki örnek görüntü üzerinde deneysel olarak belirlenmiştir.

Bunun için (3.2) denklemi ile hesaplanan  $G^*$  matrisine ilişkin görselde nesne üzerinde bir nokta seçilmiştir (Şekil 3.33) ve ilgili noktanın yatay piksel numarasına karşılık değer grafiği çizdirilmiştir (Şekil 3.34).



Şekil 3.33. Parlak piksele ait okunan değerler



**Şekil 3.34.** Seçilen parlak noktanın yatay hizasındaki piksellerin renk şiddetleri Şekil 3.34’de verilen grafik üzerinde görsel bir inceleme sonucunda eşik değeri deneysel olarak  $K_T = 35$  olarak belirlenmiştir.

Eşik değeri hesaplanarak  $G_{BW}$  matrisinin görseli Şekil 3.35 ve Şekil 3.36’da gösterilmiştir.



**Şekil 3.35.** Özgün resim ve eşik değeri işlemi sonucu



**Şekil 3.36.** Hesaplanan blobun yakından görünümü

Bu binary görüntüden nesnenin merkezini elde edebilmek için,  $G_{BW}$  matrisine bazı *binary* görüntü analizi yöntemleri uygulanır. Önce “**bağlı bileşenlerin etiketlenmesi**” yöntemi ile binary görüntü içerisindeki bağlı bileşenler (*blob*) etiketlenir; daha sonra da etiketlenen bağlı bileşenlerin merkezleri hesaplanır.

**Bağlı bileşenleri** tanımlamak gerekirse:  $G_{BW}$  matrisinde iki adet  $G_{BW}(r, c) = G_{BW}(r', c') = v$  pikseli tanımlansın.  $v$  değeri ya 1 yada 0 değerini alabilir.  $(r, c)$ 'den  $(r', c')$ 'ye  $i = 1, \dots, n$  için  $(r, c) = (r_0, c_0), (r_1, c_1), \dots, (r_n, c_n) = (r', c')$  sıralı pikselleri bulunsun. Eğer her bir  $i = 1, \dots, n$  değeri için  $G_{BW}(r_i, c_i) = v$  oluyorsa ve yine her bir  $i$  değer için  $(r_i, c_i)$  ile  $(r_{i-1}, c_{i-1})$  pikseli komşu ise  $(r, c)$  pikseli ile  $(r', c')$  pikseli birbirine **bağlıdır** denir (Shapiro 2000). Böylece  $(r_0, c_0), \dots, (r_n, c_n)$  sıralı pikselleri  $(r, c)$ 'den  $(r', c')$ 'ye bağlı bir yol oluşturur. Her biri  $v$  değerini alan ve her bir piksel çiftinin birbirine göre bağlı olduğu bir kümeye “ $v$  değerinde bir **bağlı bileşen**” denir. Şekil 3.37’de 1’lerden oluşan beş adet bağlı bileşen görülüyor. Bu bileşenler gerçekten de birbirleriyle 8 yada 4-komşuluğa göre bağlıdır (Shapiro 2000).

1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

Şekil 3.37. Herhangi bir ikili (binary) görüntüye ait matris ve etiketlenilmiş bağlı bileşenler matrisi (Shapiro 2000)

**Bağlı bileşenlerin etiketlenilmesi**, binary  $G_{BW}$  matrisindeki her bir bağlı bileşenin benzersiz bir numara ile eşleştirilmesidir. Elde edilen  $G_L$  matrisi ise **etiketlenilmiş görüntüyü** temsil eder. (Şekil 3.37) Bağlı bileşenlerin etiketlenilmesine yönelik çeşitli algoritmalar vardır (Shapiro 2000).

Bağılı bileşenler bir kez etiketlendikten sonra bu bölgelere ilişkin unsurlar hesaplanabilir (*feature extraction*). Bu unsurlar bölgenin alanı, merkezi, uç noktaları gibi geometrik özellikler olabileceği gibi çembersellik yada elipsellik gibi şekilsel ölçümler de olabilir. Bir  $R$  matrisi bir bölge içerisindeki piksellere işaret etsin. Bölgenin alanı ( $A$ ) ve merkezi ( $(\bar{r}, \bar{c})$ ) hesaplanabilen en basit özelliklerdir ve aşağıdaki şekilde hesaplanırlar (Shapiro 2000):

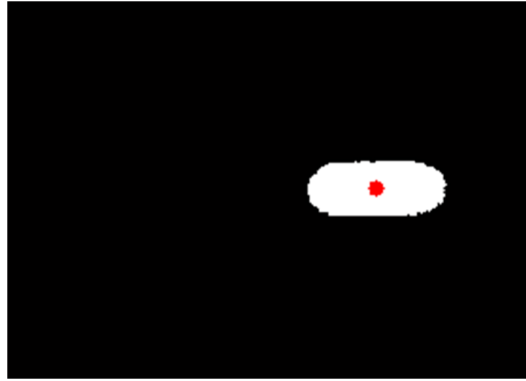
$$A = \sum_{(r,c) \in R} 1 \quad (3.3)$$

Denklem (3.3), basitce  $R$  alanı içerisindeki piksellerin sayısı anlamına gelir.

$$\begin{aligned} \bar{r} &= \frac{1}{A} \sum_{(r,c) \in R} r \\ \bar{c} &= \frac{1}{A} \sum_{(r,c) \in R} c \end{aligned} \quad (3.4)$$

Bölgenin merkezi  $(\bar{r}, \bar{c})$ , piksellerin  $R$  bölgesi içerisindeki ortalama konumudur (ağırlık merkezi). Her bir  $(r, c) \in R$  birer tamsayı çifti olsa bile,  $(\bar{r}, \bar{c})$  genelde bir tamsayı değildir fakat bir pikselin ondalık bir hassasiyeti kabul edilebilir bir miktardır.

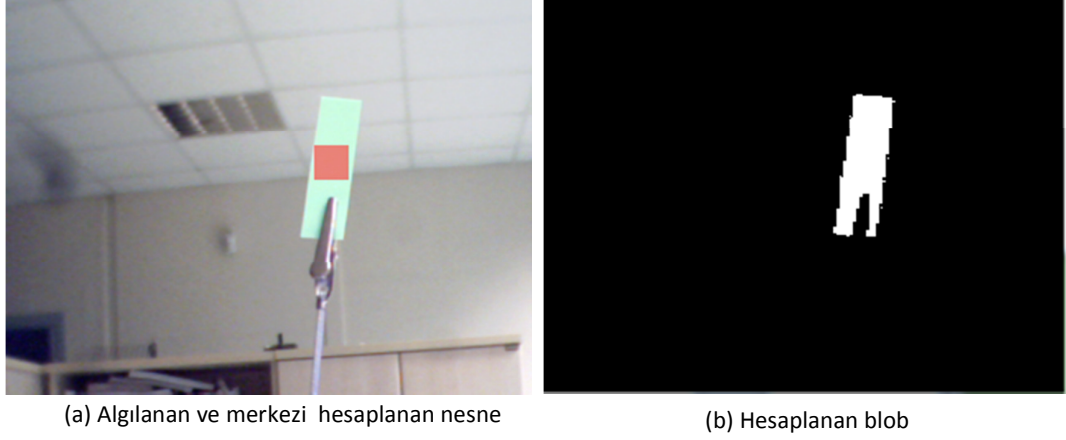
Denklem (3.4) yardımı ile, etiketlenilmiş görüntü ( $G_L$ ) matrisi içerisindeki bölgelerin merkezleri hesaplanabilir. Deney görüntüsünde hesaplanan ve işaretlenen merkez noktası aşağıdaki Şekil 3.38’de görülmektedir:



**Şekil 3.38.** Merkez noktası hesaplanmış blob

Bu şekilde video görüntüleri içerisindeki nesneye ilişkin konum ve alan unsurları çıkarılabilir. Hesaplanan bu unsurlar daha sonra hareket alt sistemine hata girişi olarak beslenmektedir.

Tasarlanan görme alt sisteminin testi sırasında elde edilen bir ekran görüntüsü Şekil 3.39'da verilmiştir.



**Şekil 3.39.** Görme alt sisteminin testi

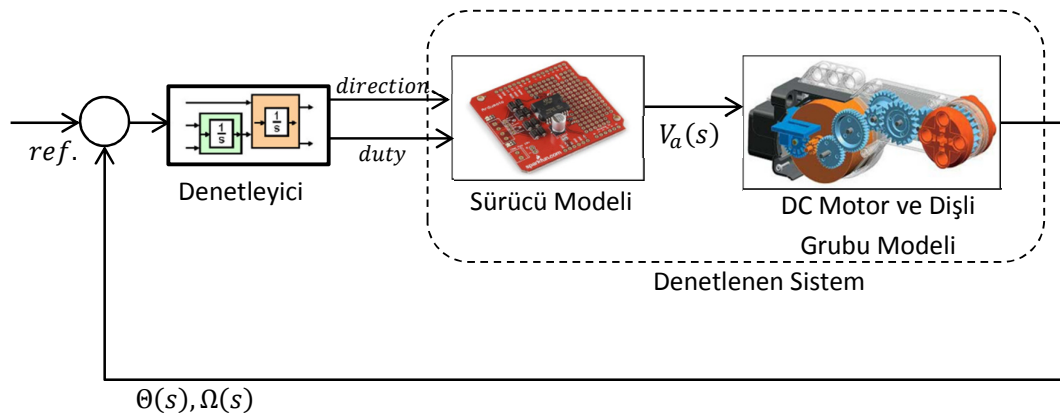
### 3.2.2. Hareket Altsistemi

Tez çalışmasında ele alınan nesne takip sisteminin, algılanan nesneyi merkezleyecek şekilde kameranın açisal konumunu güncelleyen bir görüntü-tabanlı görsel servo olduğundan bahsedilmişti. Kameranın açisal hareketi ise hareket altsistemi tarafından gerçekleştirilmektedir. Hareket altsistemi donanımsal olarak bir elektromekanik eyleyici olan DC motor-dişli grubundan ve sürücü elektroniğinden; yazılımsal olarak ise konum ve hız denetleyici algoritmalarından oluşmaktadır.

Hareket altsistemi aslında kapalı döngü bir denetim sistemidir Şekil 3.15 (sayfa 31) tekrar göz önünde bulundurulursa denetim algoritmaları gerçek-zamanlı olarak xPC Target hedef bilgisayar üzerinde çalışmaktadır. Konum geri beslemesi, DC motor üzerindeki dahili optik encoder ile sağlanmakta ve hesaplanan denetim sinyalleri ise PWM biçiminde sürücü devresine gönderilmektedir.

Sonuç olarak, kameranın açisal hareketi DC motorun açisal konum ve hız denetimi ile gerçekleştirilmektedir. Bundan dolayı, kullanılan elektromekanik eyleyiciye konum ve hız denetleyicisi tasarlamak gerekli olmuştur.

Şekil 3.40'da hareket altsistemine ilişkin denetim sisteminin prensip şeması verilmiştir.

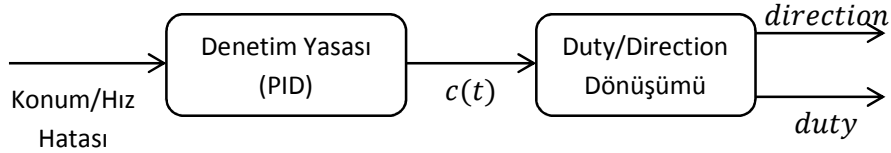


**Şekil 3.40.** Elektromekanik sisteme ilişkin denetim sisteminin modeli

Şekil 3.40'da görüldüğü üzere hareket altsistemi, kapalı döngü bir denetim sisteminin genel biçimine uygun olacak şekilde denetleyici ve denetlenen altsistemlerinden mevcuttur. Denetleyici altsistemi, referans hızın/konumun ve ölçülen hızın/konumun

farklı olan hata sinyalinden, denetlenen sistemin girişi olan denetleyici sinyallerini (*duty* ve *direction*) hesaplar.

Şekil 3.41'den tekrar görüldüğü gibi, sürücü altsisteminin girişleri (aynı zamanda denetleyicinin çıkışları) *direction* ve *duty* sinyalleridir.



**Şekil 3.41.** Denetleyici alt sistemi

*duty* ve *direction* sinyali, DC motorun sürülebilmesi için gerekli olan gerilim sinyalinin ( $V_a$ ) oluşturulması için gereklidir. *duty*, belirli bir  $t$  anı için PWM sinyalinin doluluk/boşluk oranı anlamına gelir ve DC motora sağlanan gerilim ( $V_a$ ),  $0 < duty < 1$  değeri ile kontrol edilir. *direction* değeri ise, 3.1.4.2. numaralı başlıkta (sayfa 36) tarif edilen H-köprüsünün kutuplarını değiştirmek için, başka bir deyişle, motoru ileri yada geri sürmek için gerekli olan mantıksal bir sinyaldir.

Şekil 3.41'de gösterilen ve tasarımı 3.2.2.2-3 numaralı başlıklarda (sayfa 77 ve 96) ayrıntılı bir şekilde verilen denetim yasası olarak PI algoritması tercih edilmiştir.

Şekil 3.41'deki  $c(t)$  her zaman  $-1 < c(t) < 1$  aralığındadır ve denetim yasası tarafından hesaplanan komut sinyalini temsil eder.  $c(t)$ , elektromekanik altsisteme giriş olarak gönderilmeden önce, *duty* ve *direction* sinyallerine ayrıştırılması gerekir:

$$\begin{aligned}
 duty &= |c(t)| \\
 c(t) > 0 &\rightarrow direction = 1 \\
 c(t) < 0 &\rightarrow direction = 0
 \end{aligned}
 \tag{3.5}$$

Sonuç olarak, Şekil 3.41'de gösterilen *duty/direction* dönüşümü (3.5) ile gerçekleştirilir.



Elektromekanik eyleyiciye yönelik denetim sisteminin tasarımında aşağı sıralanan iş akışı takip edilmiştir.

- 1) *Denetlenen sistemin modellenmesi:* Kapalı-döngü denetim sistemlerinde denetleyicinin tasarımı, denetlenen sistemin modeline dayalı olarak geliştirildiği için önce denetlenen sistemin modellenmesi gerekli olur. Bu çalışmada elektromekanik sistemin bileşenleri (DC motor - dişli grubu ve sürücü elektroniği) matematiksel olarak modellenmiştir ve Simulink yazılımı ortamında benzetim modeli oluşturulmuştur. Modellenen bir fiziksel sistemin bazı parametreleri kesin olarak bilinmiyor ise ölçüm verilerinden yola çıkarak *parametre kestirimi* gibi yöntemler ile bu bilinmeyen parametreler hesaplanabilir. Bu çalışmada, değeri kesin olarak bilinmeyen eşdeğer sürtünme katsayısı ( $B_{eq}$ ) ve eylemsizlik momenti ( $J_{eq}$ ) *parametre kestirimi* yöntemleri ile hesaplanıp model kalibre edilmiştir.
- 2) *Denetleyici tasarımı:* Deney verileri ile doğrulanmış bir denetlenen sistem modeli elde edildikten sonra, bu sisteme ilişkin denetleyici tasarlanabilir. Bu çalışmada elektromekanik altsistem için gerekli olan hız ve konum denetleyicisi, ilerideki başlıklarda açıklanan yöntemler ile tasarlanmıştır.
- 3) *Gerçek zamanlı testlerin gerçekleştirilmesi:* Benzetim ortamında kurulan denetim sistemi işlevsel olarak tasarlanıp doğrulandıktan sonra, denetim algoritmalarının gerçek zamanlı bir ortamda test edilmesi gerekli olur. Bu çalışmada, geliştirilen denetim algoritmaları xPC Target hedef bilgisayarı ortamında gerçek zamanlı olarak çalıştırılıp, gerçek elektromekanik sistem ile birlikte test edilmiştir. Test yöntemi olarak da, önceki başlıklarda açıklanan hızlı denetim prototipleme ve döngüde-yazılım benzetimleri kullanılmıştır.

Bu iş akışı yinelemeli bir iş akışıdır. Tasarlanan denetim sistemi, istenilen performans ölçütlerini (maksimum aşım, oturma zamanı vs. ) sağlayana kadar model parametreleri değiştirilip gerçek-zamanlı testler tekrarlanmıştır. İstenilen performans ölçütleri sağlandıktan sonra, en uygun parametreler belirlenip, çalışma içerisinde kullanılmıştır. Hemen sonraki başlıklarda, uygulanan iş akışı ayrıntıları ile tarif edilmiştir.

### 3.2.2.1. Denetlenen Sistemin (Elektromekanik sistem) Modellenmesi

Şekil 3.40'dan da görüldüğü üzere; denetlenen sistem; DC motoru, dişli kutusu ve sürücü elektroniğinden oluşan elektromekanik bir altsistemdir.

3.1.4.2 numaralı başlıkta (sayfa 36) tarif edilen motor sürücü devresi, PWM ve yön (DIR) sinyal girişlerine sahiptir. PWM sinyali fiziksel olarak, 3.1.4.4 numaralı başlıkta (sayfa 40) tarif edilen çok amaçlı I/O kartı tarafından oluşturulur. Sürücü devresi, sabit bir doğru akım kaynağından alınan besleme gerilimini ( $K_{supply}$ ) PWM ve DIR sinyaline göre anahtarlayarak, motoru sürececek olan  $V_a$  gerilimini oluşturur. Çok amaçlı I/O kartının PWM sinyalini oluşturabilmesi için belirli bir  $t$  anı için *duty* değerine gereksinim duyar ve bu sinyal denetleyici altsistem tarafından hesaplanır. Özetle, sürücü devresinin matematiksel modeli oluşturulmak istenirse, modelin girişleri *duty* ve *direction* sinyalleri olacaktır ve çıkışı ise DC motoru sürececek olan gerilim  $V_a$  olacaktır.

Sürücü devresini modellemek için iki farklı yaklaşım tercih edilmiştir. Birinci yaklaşımda, Simulink yazılımı ortamında SimScape ve SimElectronics kütüphaneleri ile tamamlanmıştır. Diğer yaklaşımda ise, sürücü devresinin basitleştirilmiş matematiksel modeli elde edilmiştir. DC motora uygulanan ortalama gerilim, PWM sinyalinin doluluk/boşluk oranı (*duty*) ile doğru orantılı olduğu kabul edilirse,

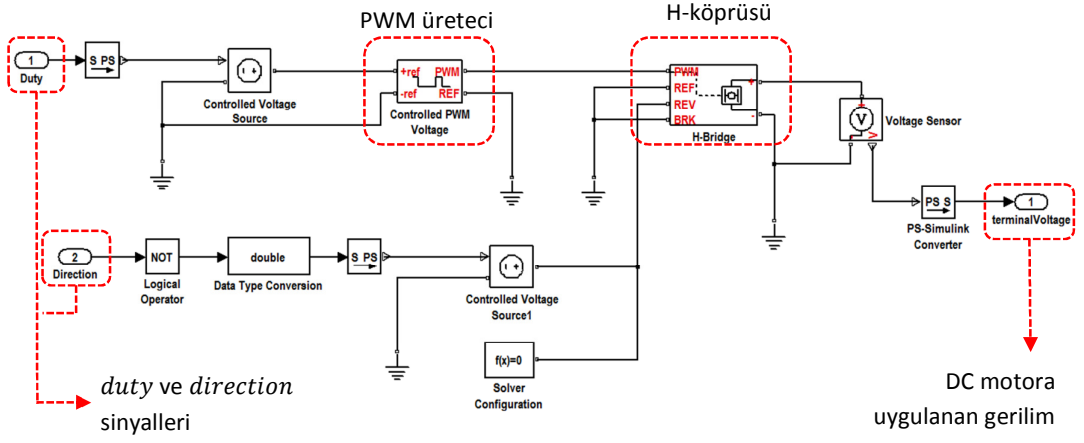
$$v_{a,ort} = duty \cdot direction \cdot K_{supply} \quad (3.6)$$

elde edilir.  $c(t) = duty \cdot direction$  olarak (3.6)'da tekrar yerine konursa ve Laplace dönüşümü gerçekleştirilirse,

$$\frac{V_a(s)}{C(s)} = K_{supply} \quad (3.7)$$

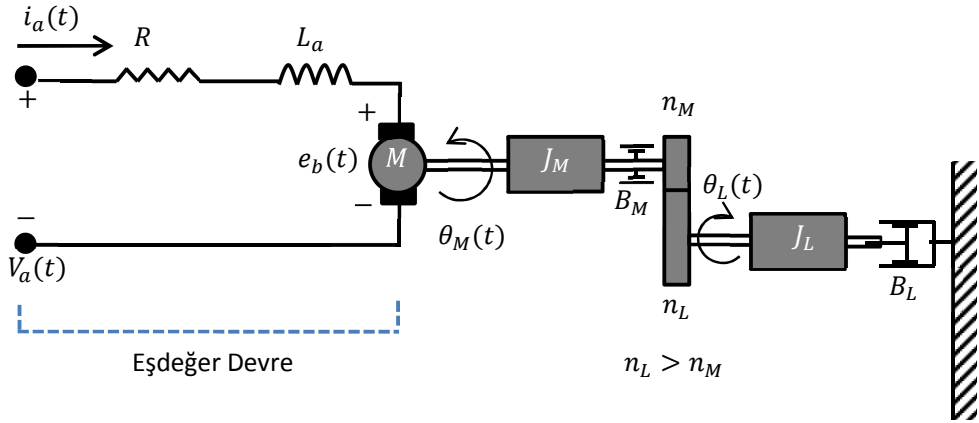
olarak sürücünün giriş/çıkış ilişkisini veren kazanç türünden transfer fonksiyonu elde edilebilir. Burada  $c(t)$  aslında, kullanılan denetim yasaasının (PI algoritması) çıkışıdır ve 3.2.2 numaralı başlıkta ayrıntılı bir şekilde verilmiştir.

SimScape kütüphanesi ile gerçekleştirilen sürücü modeli de Şekil 3.42’de verilmiştir.



Şekil 3.42. DC motor sürücüsünün Simulink/SimScape modeli

Elektromekanik altsistemin diğer bileşenleri de DC motor-dişli grubudur. Şekil 3.43’de DC motor ve dişli grubunun basitleştirilmiş modelini temsil eden şema verilmiştir.



Şekil 3.43. DC motor ve dişli grubunun şeması

Şekil 3.43’de görüldüğü gibi, elektriksel kısım direnç ( $R_a$ ) ve endüktans ( $L_a$ ) elemanlarından; mekaniksel sistem de  $J_M$  eylemsizlik kütlesi ve  $B_M$  dönel sönümleme elemanından oluşmaktadır. Bu dönel elektromekanik sistemin matematiksel modeli çıkarılırken, armatür uçlarına uygulanan  $V_a$  gerilim değişkenine karşılık mekanik sistemin açısal hareketi (konum ve hız) göz önünde bulundurulmuştur.

Bu sistemde rotor, deęişken mıknatıs alanı içinde dönmeye zorlandıęından bobinin üzerinde akımın akışını zorlaştıracı yönde bir zıt elektromotor kuvvet ( $e_b$ ) meydana gelir. Buna göre sargı uçlarına uygulanan  $v_a$  gerilim deęişimine karşılık olulan akım deęişiminin oluşturduęu mıknatıs akısı,  $v_a$  gerilimine ters yönde bir  $e_b$  gerilim farkı oluşturur. Bu durumda, elektriksel devredeki akım deęişimini oluşturan gerilim  $v_a - e_b$  fark gerilimine eşittir.

Elektriksel kısma Kirchoff yasası uygulanırsa:

$$v_a - e_b = L_a \frac{di_a}{dt} + R_a i_a(t) \quad (3.8)$$

elde edilir.

Burada  $e_b$  zıt elektromotor kuvvet, rotorun hızı ile orantılı olup doğrusallaştırılmış baęıntısı

$$e_b = K_b \omega_M \quad (3.9)$$

olarak verilir. Meydana gelen akım deęişimi sonucu bobin etrafında bir mıknatıs alanı oluşur ve bu mıknatıs alanı rotor üzerinde bir döndürme momenti oluşturur. Gerçekte, akım ile meydana gelen döndürme momenti arasındaki baęıntı doğrusal deęildir. Sistemin dinamik davranışına açıklık getirmek için akım ile moment deęişimi arasındaki baęıntının doğrusal olduęu varsayılırsa, oluşturulan moment ( $M_M$ ) aşıęıdaki şekilde ifade edilebilir:

$$M_M = K_t i_a(t) \quad (3.10)$$

(3.10), elektriksel sistem ile mekaniksel sistem arasında elektromekanik bir tork dönüşüm denklemi olup burada  $K_t$  tork dönüşüm sabitidir. (3.10) ile motorun yük tarafına (dişli grubu) aktaracaęı momentin denklemi elde edilmiştir.

Şekil 3.43'de yine yük tarafının (dişli kutusu) basitleştirilmiş bir modeli verilmiştir. Dönel servomekanizmalarda dişli takımları hız düşürmek ve buna karşılık üretilen torku yükseltmek amacı ile yaygın olarak kullanılırlar. Şekil 3.43'deki basitleştirilmiş dişli grubu modeli ele alınırsa yük, bir dişli takımı boyunca motor tarafından üretilen tork ile sürülmektedir. Bu çalışmada kullanılan DC motoruna ait dişli kutusunda 8 adet dişli

bulunsa da, dişli oranlarının bileşkesi alınmış ve dişli kutusunun modeli büyük ve küçük dişliden oluşacak şekilde basitleştirilmiştir.

Motorun sağladığı  $M_M$  döndürme momentinin bir kısmı motor eylemsizlik momentini ( $J_M$ ), bir kısmı da motor yapışkanlık sürtünmesini ( $B_M$ ) yenmek için harcanacaktır ve geri kalan kısmı da küçük dişliye iletilecektir. Net moment,

$$M_M = J_M \ddot{\theta}_M + B_M \dot{\theta}_M + M_1 \quad (3.11)$$

olarak elde edilir. Burada  $M_1$ , küçük dişliye iletilen momenttir.  $\dot{\theta}_M$  ve  $\ddot{\theta}_M$  sırası ile motor milinin açısal hızı ve ivmesidir.

Büyük dişliye ulaşan döndürme momentinin ( $M_2$ ), yük milinin eylemsizlik momentine ( $J_L$ ) ve viskoz sürtünmeye ( $B_L$ ) harcanır:

$$M_2 = J_L \ddot{\theta}_L + B_L \dot{\theta}_L \quad (3.12)$$

$n = N_2/N_1$  dişli kutusunun girişi ile çıkışı arasındaki bileşke çevrim oranı olmak üzere, dişli takımının özelliklerinden,

$$\begin{aligned} M_1 \theta_M &= M_2 \theta_L \rightarrow M_2 = M_1 \frac{N_2}{N_1} = n M_1 \\ N_1 \theta_M &= N_2 \theta_L \rightarrow \theta_L = \frac{N_1}{N_2} \theta_M = \frac{1}{n} \theta_M \end{aligned} \quad (3.13)$$

elde edilir. Burada  $N_1$  ve  $N_2$  sırası ile küçük ve büyük dişlinin diş sayısıdır. (3.10)'da elde edilen  $M_2 = n M_1$  ve  $\theta_L = \frac{1}{n} \theta_M$  (3.11) ve (3.12)'de yerine koyup, tek eşitliğe indirgenirse;

$$M_M = J_M \ddot{\theta}_M + B_M \dot{\theta}_M + \frac{J_L}{n^2} \ddot{\theta}_M + \frac{B_L}{n^2} \dot{\theta}_M \quad (3.14)$$

elde edilir. (3.14) yeniden düzenlenirse:

$$M_M = \left( J_M + \frac{J_L}{n^2} \right) \ddot{\theta}_M + \left( B_M + \frac{B_L}{n^2} \right) \dot{\theta}_M \quad (3.15)$$

Motor ve dişli kutusu sistemi, tekil bir sisteme indirgenirse, eşdeğer eylemsizlik momenti ( $J_{eq}$ ) ve viskoz sürtünme katsayısı ( $B_{eq}$ ):

$$\begin{aligned} J_{eq} &= J_M + \frac{J_L}{n^2} \\ B_{eq} &= B_M + \frac{B_L}{n^2} \end{aligned} \quad (3.16)$$

(3.15) denklemi yeniden düzenlenirse:

$$M_M = J_{eq}\ddot{\theta}_M + B_{eq}\dot{\theta}_M \quad (3.17)$$

Elde edilmek istenen, motor tarafından üretilen torka karşılık ( $M_M$ ), yük milinin konumu olduğuna göre ( $\theta_L$ ), aşağıdaki dönüşümler gerçekleştirilebilir:

$$\begin{aligned} \theta_M &= n\theta_L \\ \dot{\theta}_M &= n\dot{\theta}_L \\ \ddot{\theta}_M &= n\ddot{\theta}_L \end{aligned} \quad (3.18)$$

$$M_M = n(J_{eq}\ddot{\theta}_L + B_{eq}\dot{\theta}_L) \quad (3.19)$$

Sonuç olarak (3.19) ile motorun ürettiği momente karşılık, yük miline ilişkin dinamik davranışlarını temsil eden ifadeye erişilebilir. Burada dikkat edilmesi gereken, (3.19)'da mekanik parçalardan kaynaklanan sürtünme momenti ( $T_f$ ) ihmal edilmiştir.  $T_f$ 'nin yönü yük milinin dönüş yönüne bağlıdır (Loh ve ark. 2007) ve sistemi doğrusal olmayan yapar. Moment eşitliği aşağıdaki gibi yeniden ifade edilebilir:

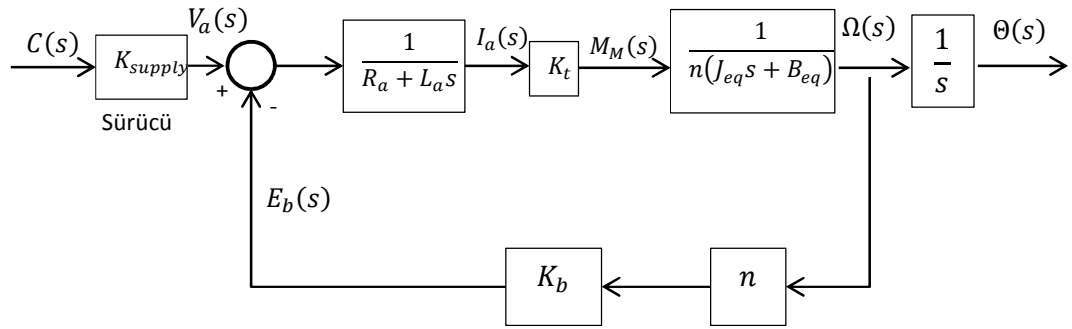
$$M_M = n(J_{eq}\ddot{\theta}_L + B_{eq}\dot{\theta}_L) + T_f \text{sgn}(\dot{\theta}) \quad (3.20)$$

(3.20) doğrusal olmadığı için, denetleyici tasarımlarında (3.19), ileriki başlıklarda bahsedilecek karşılaştırmalı analizlerde (3.20) tercih edilmiştir.

Burada amaçlanan sistemin girişi (armatür gerilimi  $v_a$ ) ve sistemin çıkışı (yük milinin konumu  $\theta_L$  ve hızı  $\omega_L$ ) arasında bir ilişki elde etmektir. Sırası ile Laplace dönüşümleri alınmış (3.8), (3.9), (3.10), (3.19) denklemleri (Çizelge 3.7) ile (3.7), giriş ve çıkış ilişkileri göz önünde bulundurularak blok diyagramı (Şekil 3.44) ile temsil edilirse, tüm denetlenen sisteme (elektromekanik altsistem) ait  $\Theta(s)/C(s)$  transfer fonksiyonu elde edilir.

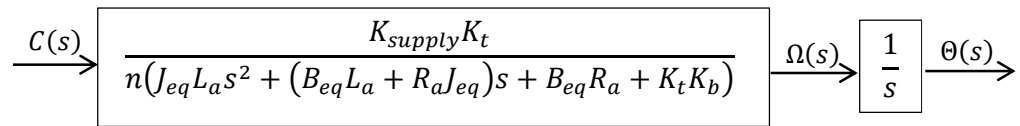
**Çizelge 3.7.** DC motorun denklemleri ve transfer fonksiyonu temsilleri

Denklem	Transfer Fonksiyonu Temsili
$v_a - e_b = L_a \frac{di_a}{dt} + R_a i_a(t)$	$\frac{I_a}{V_a(s) - E_b(s)} = \frac{1}{R_a + L_a s}$
$e_b = K_b \omega_M$	$\frac{E_b(s)}{\Omega(s)} = K_b$
$M_M = K_t i_a(t)$	$\frac{M_M(s)}{I_a(s)} = K_t$
$M_M = n(J_{eq} \ddot{\theta}_L + B_{eq} \dot{\theta}_L) = n(J_{eq} \dot{\omega}_L + B_{eq} \omega)$	$\frac{\Theta_L(s)}{M_M(s)} = \frac{1}{n(J_{eq} s + B_{eq})}$



**Şekil 3.44.** DC motor ve dişli grubunun blok diyagramı

Şekil 3.44'deki blok diyagramı indirgenirse (Şekil 3.45) denetçi girişine karşılık hızın transfer fonksiyonu  $\Omega(s)/C(s)$  ve konumun transfer fonksiyonu  $\Theta(s)/C(s)$  elde edilir.



**Şekil 3.45.** Blok diyagramının indirgenmiş hali

$$\frac{\Omega(s)}{C(s)} = \frac{K_{supply}K_t}{n(J_{eq}L_a s^2 + (B_{eq}L_a + R_a J_{eq})s + B_{eq}R_a + K_t K_b)}$$

$$\left(\frac{\Omega(s)}{C(s)}\right)^* = \frac{K_{supply}K_t}{n(B_{eq}L_a + R_a J_{eq})s + B_{eq}R_a + K_t K_b} \quad (3.21)$$

$$\frac{\Theta(s)}{C(s)} = \frac{K_{supply}K_t}{n(J_{eq}L_a s^2 + (B_{eq}L_a + R_a J_{eq})s + B_{eq}R_a + K_t K_b)}$$

Çizelge 3.3'de (sayfa 35) verilen motor parametrelerine göre  $L_a$  sıfıra çok yakın bir değerdir ve (3.21)'de  $\Omega^*(s)/C(s)$ ,  $\Omega(s)/C(s)$ 'nin  $L_a \cong 0$  kabulü ile sadeleştirilmiş halidir. Bu sadeleştirme Şekil 3.54'de (sayfa 75) görüldüğü üzere, sistemin dinamiğini neredeyse hiç etkilememektedir. Sadeleştirilmiş  $\Omega^*(s)/C(s)$ , hız denetleyicisi tasarımını kolaylaştırdığı için ileriki başlıklarda kullanılmıştır.

(3.21)'deki  $\Omega^*(s)/C(s)$  zaman tipinde bir transfer fonksiyonudur ve

$$\left(\frac{\Omega(s)}{C(s)}\right)^* = \frac{K_{sys}}{\tau s + 1} \quad (3.22)$$

genel biçiminde verilir. Burada  $K_{sys}$  sistem kazancıdır.  $\tau$ , zaman sabiti olup birinci dereceden sistemlerin dinamik davranışı ile ilgili temel bir parametredir. Bir sistemin zaman sabiti ne kadar az ise, sistem o kadar hızlıdır (Yüksel 2011).

(3.21), (3.22)'deki genel biçiminde yeniden düzenlenirse

$$K_{sys} = \frac{K_{supply}K_t}{B_{eq}R_a + K_t K_b}$$

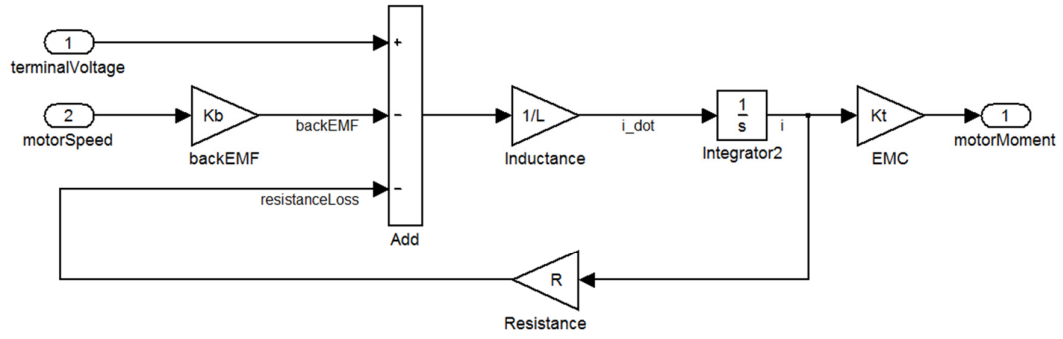
$$\tau = \frac{n(B_{eq}L_a + R_a J_{eq})}{B_{eq}R_a + K_t K_b} \quad (3.23)$$

elde edilir.  $K_{sys}$  ve  $\tau$ 'nin değerleri,  $B_{eq}$  ve  $J_{eq}$  parametrelerinin bu aşamadaki belirsizliğinden dolayı sayfa 75'de verilmiştir.

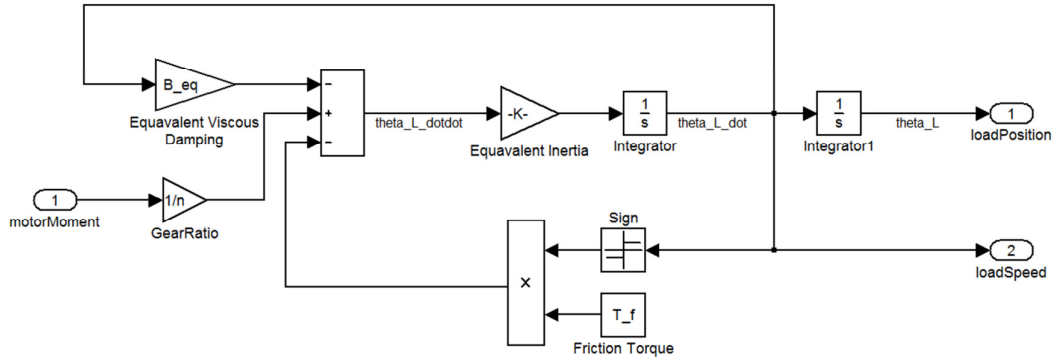
(3.21) ile dinamiği tanımlanan elektromekanik altsistemin parametreleri Çizelge 3.3'de verilmişti. DC motora ait ve Çizelge 3.3'de (sayfa 35) yer almayan eşdeğer viskoz sürtünme sabiti ve eşdeğer eylemsizlik momentinin değeri tam olarak bilinmediğinden *parametre kestirimi* yöntemleri ile ölçüm verilerinden yaklaşık olarak hesaplanması gerekli olmuştur. Parametre kestirimi 3.1.5.1 numaralı başlıkta (sayfa 43) tarif edildiği gibi gerçekleştirilmiştir ve Çizelge 3.8'de verilen sonuçlar elde edilmiştir.





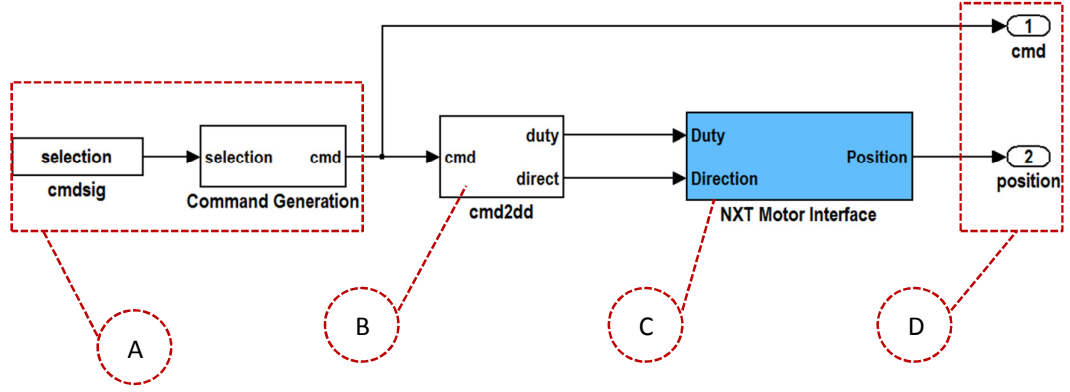


Şekil 3.47. DC motor elektriksel kısmın Simulink modeli



Şekil 3.48. DC motor mekanik kısmın Simulink modeli

**Elektromekanik sistem üzerinden verilerin toplanması:** Parametre kestirimi için gerekli olan deney verileri, xPC Target hedef bilgisayarı ile elektromekanik altsisteme gönderilen giriş sinyaline karşılık yük mili konum verileri toplanarak elde edilmiştir. Buna yönelik gerçek-zamanlı bir test uygulaması geliştirilmiştir ve bu uygulama (Şekil 3.49) xPC Target ortamında gerçekleştirilmiştir.

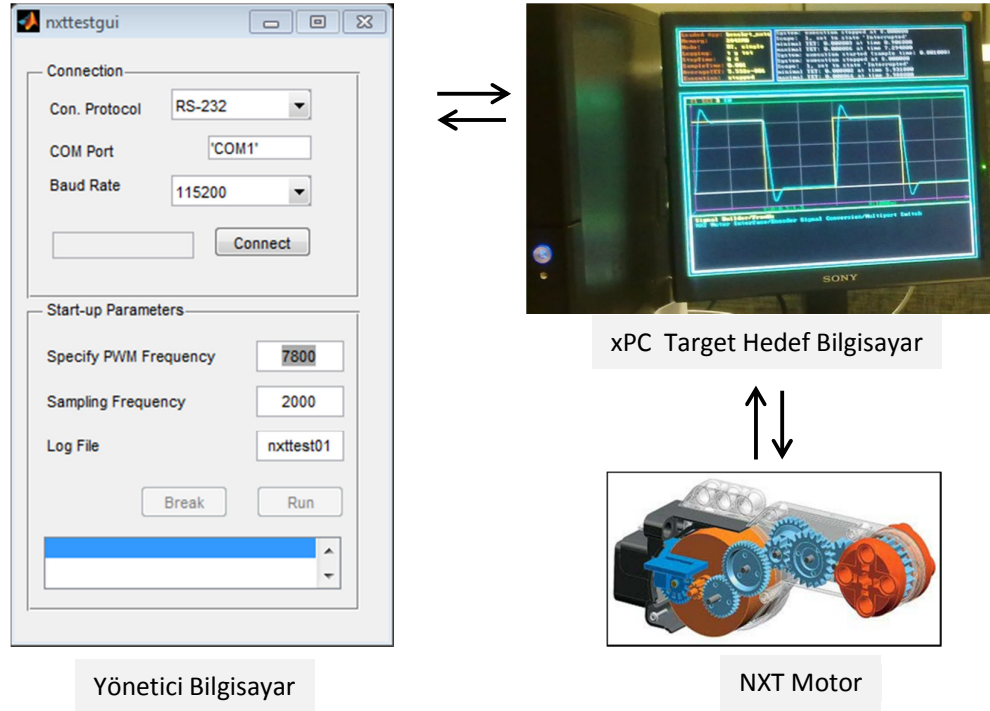


**Şekil 3.49.** Motorun gerçek zamanlı sınanmasına yönelik Simulink ortamında geliştirilen uygulama

Şekil 3.49’da verilen Simulink modelinde,

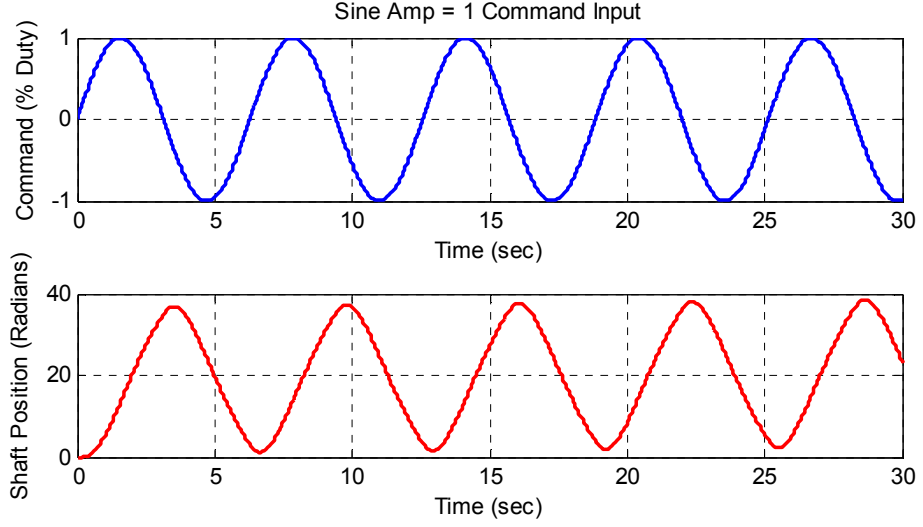
- (A) ile gösterilen kısımda, gerçek sisteme giriş olarak gönderilen  $c(t)$  giriş sinyali üretilmektedir.
- Üretilen  $c(t)$  giriş sinyali, (B) ile gösterilen altsistem ile görev çevrimi (*duty*) ve yön (*direction*) sinyallerine dönüştürülür.
- (C) ile gösterilen altsistem, elektromekanik sistem ile I/O kartları arasındaki arayüzü sağlamaktadır. Bu altsistemde, Humusoft MF624 I/O kartına ilişkin kanallar tanımlıdır ve bu kartın sürücülerini bulunur.
- (D) ile gösterilen çıkış kapıları (*outports*) ile gerçek zamanlı çalışma süresince, hangi sinyallerin hafızada tutulacağını belirlenir. Bu çalışmada, giriş komut sinyaline ( $c(t)$ ) karşılık, yük mili konum verileri saklanmıştır.

Şekil 3.49’da verilen gerçek-zamanlı uygulama, başlatılmak, sonlandırılmak, çalışma sonunda depolanmış verileri kullanıcı tarafına (xPC Target yönetici PC) geri taşımak gibi görevleri yerine getiren bir programa gereksinim duyar. Bu kapsamda, xPC Target hedef PC’yi yönetebilmek için, yönetici PC üzerinde çalışan bir grafiksel kullanıcı arayüzü geliştirilmiştir (Şekil 3.50).



**Şekil 3.50.** Parametre kestirimi deneyleri için geliştirilen GUI ve prensip şeması

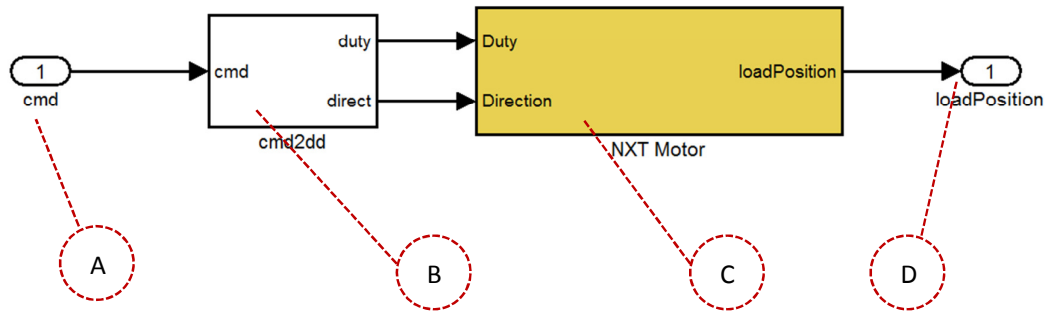
Şekil 3.50'deki program, önceden tanımlanmış test giriş sinyallerini sırası ile gerçek sisteme giriş olarak gönderir ve test sonunda depolanmış verileri kullanıcı tarafına geri aktarır. Bu *parametre kestirimi* çalışmasında, 30 s süreli, genliği 1 ve frekansı 1 rad/s olan giriş sinüs fonksiyonu test sinyali uygulanmıştır ve düzenek üzerinden alınan veriler Şekil 3.51'de verilmiştir.



**Şekil 3.51.** Giriş test sinyaline karşılık elektromekanik sistemin yanıtı

Şekil 3.51'deki ilk grafik giriş test sinyalini, ikinci grafik ise motor mili konumunu göstermektedir.

**Parametre kestirimi işlemine ön-hazırlık:** Parametre kestirimi için gerekli olan giriş/çıkış ölçüm verileri (Şekil 3.51) elde edildikten sonra, optimizasyon hesabına girecek bir Simulink modelinin ayrıca oluşturulması gerekir ve bu model Şekil 3.52'de verilmiştir.



**Şekil 3.52.** Parametre kestirimi için oluşturulan Simulink modeli

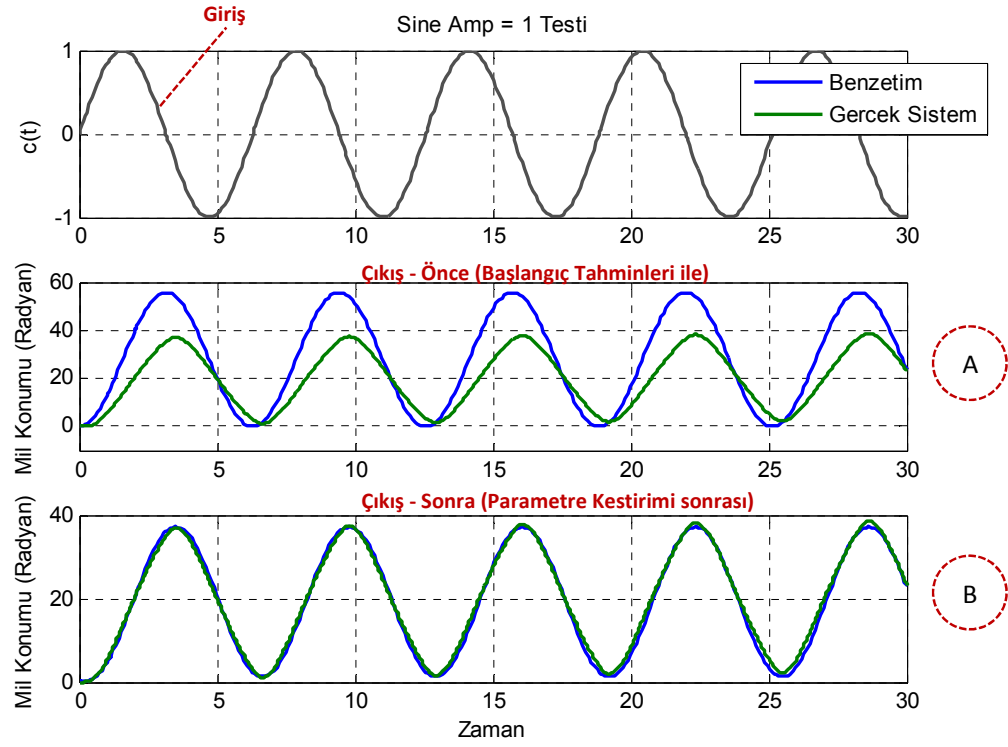
Şekil 3.52’de verilen Simulink modelinde,

- (A) ve (D) ile modelin girişi ( $c(t)$ ) ve çıkışı (konum) belirlenir;
- (B) ile  $c(t)$  sinyali (3.5)’de verildiği şekilde *duty* ve *direction* sinyallerine ayrıştırılır;
- (C) ise elektromekanik altsistemdir ve açık modeli Şekil 3.46’da (sayfa 67) verilmiştir.

Optimizasyon yöntemi olarak doğrusal olmayan en küçük kareler yöntemi kullanılmıştır ve  $B_{eq}$  ve  $J_{eq}$  parametrelerinin başlangıç tahmini olarak Çizelge 3.8’deki değerler kullanılmıştır.

***Parametre kestimi işlemi ve sonuçların alınması:*** Şekil 3.52’de verilen Simulink modeli Simulink Design Optimization aracı ile optimize edilir. Benzetim modelinin yanıtı ile gerçek sistemin yanıtı belirli bir tolerans içerisinde eşleşene kadar iterasyonlar tekrarlanır.

Şekil 3.53’de optimizasyon işlemi öncesi ve sonrası sistem yanıtlarının karşılaştırılması verilmiştir.



**Şekil 3.53.** Optimizasyon öncesi ve sonrası benzetim/gerçek sistem yanıtlarının karşılaştırılması

Şekil 3.53’de birinci grafik, Şekil 3.51’de verilen test giriş sinyalinin aynısıdır. İkinci ve üçüncü grafiklerde, deney ve benzetim sonuçlarının karşılaştırıldığı motor konum-zaman (s) grafiğidir.

- (A)’da  $B_{eq}$  ve  $J_{eq}$  parametrelerinin başlangıç tahmin değerlerine göre, sistem yanıtlarının karşılaştırılması verilmiştir. Görüldüğü üzere benzetim modelinin yanıtı ile gerçek sistemin yanıtı oldukça farklıdır.
- (B)’de ise aynı karşılaştırma, kestirilen son değerlere göre verilmiştir. Görüldüğü üzere, benzetim modeli ile gerçek sistemin yanıtı oldukça benzerdir.

Hesaplanan  $B_{eq}$  ve  $J_{eq}$  parametreleri Çizelge 3.8'de verilmiştir.

**Çizelge 3.8.** DC motorun bilinmeyen parametreleri ve yaklaşık hesaplanan değerler

<b>Parametre</b>	<b>Değer</b>	
Eşdeğer viskoz sürtünme sabiti	$B_{eq} = 1,1278 \cdot 10^{-3} \frac{Nm}{rad/s}$ (Başlangıç tahmini)	$B_{eq} = 4,031 \cdot 10^{-8} \frac{Nm}{rad/s}$ (Kestirilen değer)
Eşdeğer eylemsizlik Momenti	$J_{eq} = 1 \cdot 10^{-5} kgm^2$ (Başlangıç tahmini)	$J_{eq} = 0,0084 kgm^2$ (Kestirilen değer)

Bilinmeyen parametreler de hesaplandığına göre, denetlenen sistemin (*plant*) eksiksiz bir modeli elde edilmiş olunur. Bu denetlenen sistem modeli, denetleyici tasarımı için esas teşkil eder. Denetleyici tasarımından önce, denetlenen sistemin dinamik davranışlarını (zaman ve frekans alanında) incelemekte fayda vardır.



## Dinamik Davranışların Analizleri

Açık döngü sistemin kazancını ve zaman sabitini hesaplamak için Denklem 3.22’de sabitler yerine konulduğunda

$$\frac{\Omega^*(s)}{C(s)} = \frac{19.23}{0.3788s + 1}$$

transfer fonksiyonu elde edilir. Buradan

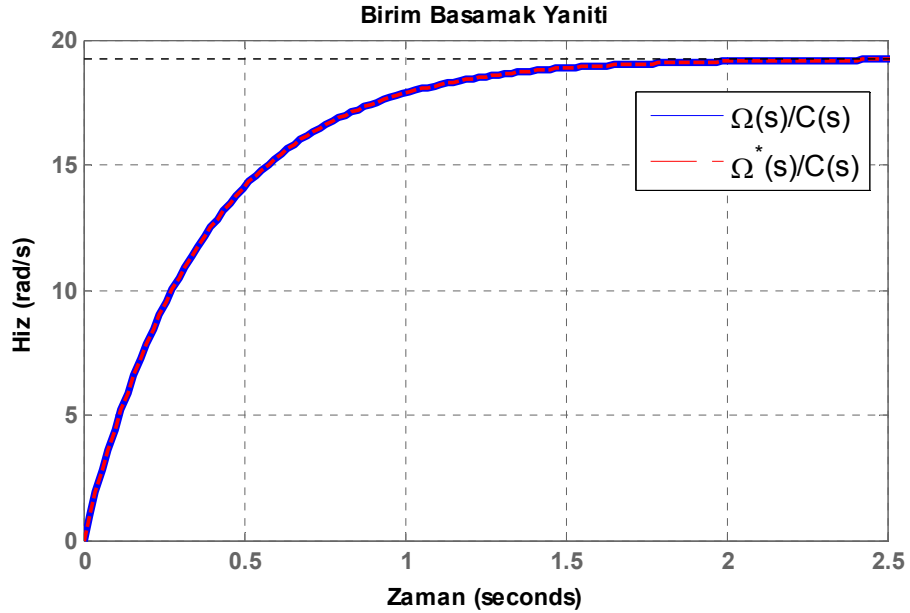
$$K_{sys} = 19.23$$

$$\tau = 0.3788 \text{ s}$$

olarak hesaplanır.  $\Omega^*(s)/C(s)$  sisteminin sadece bir adet kutubu vardır:

$$p_1 = -2.64$$

Şekil 3.54’de görüldüğü gibi sistemin  $\Omega(s)/C(s)$  birim basamak yanıtı sönümlü, titreşimsiz ve üstel gecikmeli bir biçimdedir. Zaman gecikmesi ve yukarıda belirtildiği gibi  $\tau = 0.3788 \text{ s}$ ’dir. Sistemin oturma zamanı yaklaşık olarak  $4\tau \cong 1.4819 \text{ s}$ ’dir.



Şekil 3.54. Denetlenen sistemin birim basamak yanıtı

Bir sistemin kutuplarının karmaşık düzlemdeki yeri, o sistemin kararlılığına işaret eder. Sistemin kutubu karmaşık düzlemin sol tarafında kaldığı için sistem kararlıdır. Kararlı sistemler, sonu bir girişe sonlu bir çıkışla yanıt veren sistemlerdir. Bu sonuç, Şekil 3.54'de de görülmektedir.

Ayrıca Denklem (3.21) ile temsil edilen sistemi kalıcı durum hatasına göre sınıflandırarak, denetlenebilirliğini ortaya konulabilir. Denklem (3.21)'de parametrelerin değerleri yerine konulup sıfır-kutup-kazanç gösterimi ile yeniden düzenlenirse

$$\begin{aligned}\frac{\Omega(s)}{C(s)} &= \frac{2426020.4082}{s^0(s + 47780)(s + 2.64)} \\ \frac{\Omega^*(s)}{C(s)} &= \frac{50.7687}{s^0(s + 2.64)} \\ \frac{\Theta(s)}{C(s)} &= \frac{2426020.4082}{s^1(s + 47780)(s + 2.64)}\end{aligned}\tag{3.24}$$

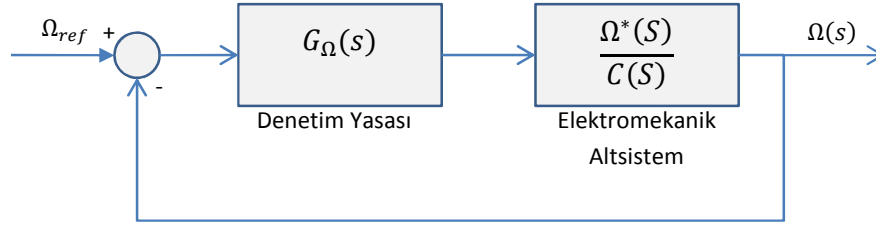
elde edilir. (3.25) ile  $\Omega(s)/C(s)$  ve  $\Omega^*(s)/C(s)$  sisteminin Tip0,  $\Theta(s)/C(s)$  sisteminin ise Tip1 olduğu anlaşılır. O halde,  $\Omega(s)/C(s)$  ve  $\Omega^*(s)/C(s)$  basamak girişini,  $\Theta(s)/C(s)$  ise rampa girişini belirli bir kalıcı durum hatası ile takip edebilecektir. Denetleyici tasarımında bu sistemlerin tip numarası göz önünde bulundurulmuştur.

### 3.2.2.2. DC Motorun Hız Denetleyicisinin Tasarımı

Şekil 3.13’de (sayfa 29) nesne takip sisteminin, hız veya konum durumlarından birine sahip olabileceği verilmişti. Bundan dolayı DC motor için hız denetim sisteminin tasarımı gerekli olmuştur. Başlıkta bu çalışmaya yönelik farklı hız denetleyici yapılanmaları verilmiş ve bu yapılanmalar karşılaştırmalı olarak tartışılmıştır.

Başlangıç olarak P denetleyicisi tasarlanmış, fakat kalıcı durum hatalarının giderilemediği için PI denetleyicisi tercih edilmiştir. Birden fazla P ve PI yapılanmaları bu başlıkta karşılaştırmalı olarak verilmiş ve aralarından en uygunu çalışma için seçilmiştir.

Şekil 3.40’da (sayfa 57) verilen denetim sistemi yeniden düzenlenerek Şekil 3.55’deki blok diyagramına ile temsil edilebilir.



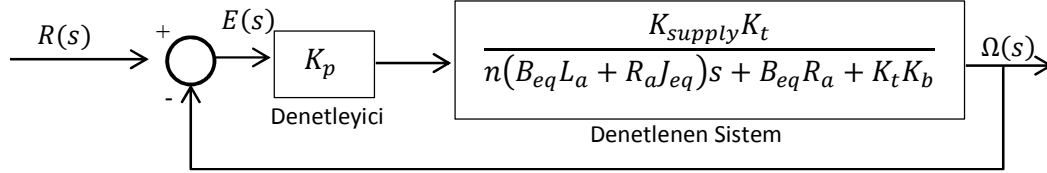
Şekil 3.55. Hız denetim sistemi blok diyagramı

Hız denetleyicisi tasarlanırken aşağıdaki birim basamak yanıtı gereksinimleri göz önünde bulundurulmuştur:

- Kalıcı durum hatası sıfır olmalıdır,
- %5 ve altında bir aşım olmalıdır,
- sistem yanıtı titreşimsiz sönümlü olmalıdır,
- 1 saniyeden daha az oturma süresi istenmektedir.

## P Denetimli Sistemin Tasarımı

Başlangıç olarak, denetleyici kazanç türünde (P) seçilmiştir ve denetim sistemi oluşturulmuştur. Denetim sisteminin blok diyagramı Şekil 3.56'da verilmiştir.



Şekil 3.56. P denetimli denetim sistemi (doğrusal model)

Şekil 3.56'daki geri besleme döngüsü indirgenirse, denetim sisteminin transfer fonksiyonu

$$G_{\Omega,P} = \frac{\Omega(s)}{R(s)} = \frac{K_p * K_{supply} * K_t}{R_a J_{eq} n s + (B_{eq} R_a + K_t K_b) n + K_p K_{supply} K_t}$$

olarak elde edilir.

$GH(s)$  sistemin açık döngü transfer fonksiyonu olmak üzere, başvuru sinyali  $R(s)$  ile hata sinyali  $E(s)$  ilişkisini veren transfer fonksiyonu

$$\frac{E(s)}{R(s)} = \frac{1}{1 + GH(s)}$$

olarak verilir ve sistemin kalıcı durum hatası  $e_{ss}$  son değer teoreminden

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \frac{s}{1 + GH(s)} R(s)$$

şeklinde hesaplanabilir (Yüksel 2011).

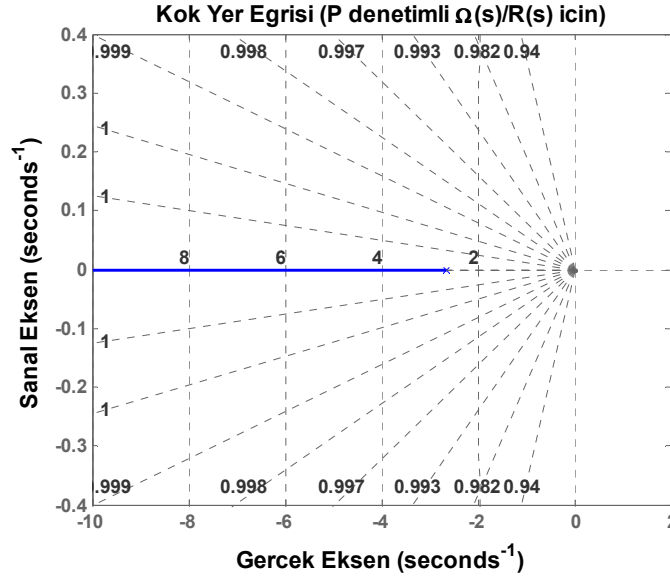
Kalıcı durum hatası  $e_{ss}$ , birim basamak başvuru sinyaline göre  $R(s) = 1/s$  göre yeniden düzenlenirse

$$e_{ss} = \lim_{s \rightarrow 0} s \frac{n(B_{eq} L_a + R_a J_{eq}) s + B_{eq} R_a + K_t K_b}{n(B_{eq} L_a + R_a J_{eq}) s + B_{eq} R_a + K_t K_b + K_p K_{supply} K_t} \frac{1}{s}$$

$$e_{ss} = \frac{B_{eq}R_a + K_t K_b}{B_{eq}R_a + K_t K_b + K_p K_{supply} K_t}$$

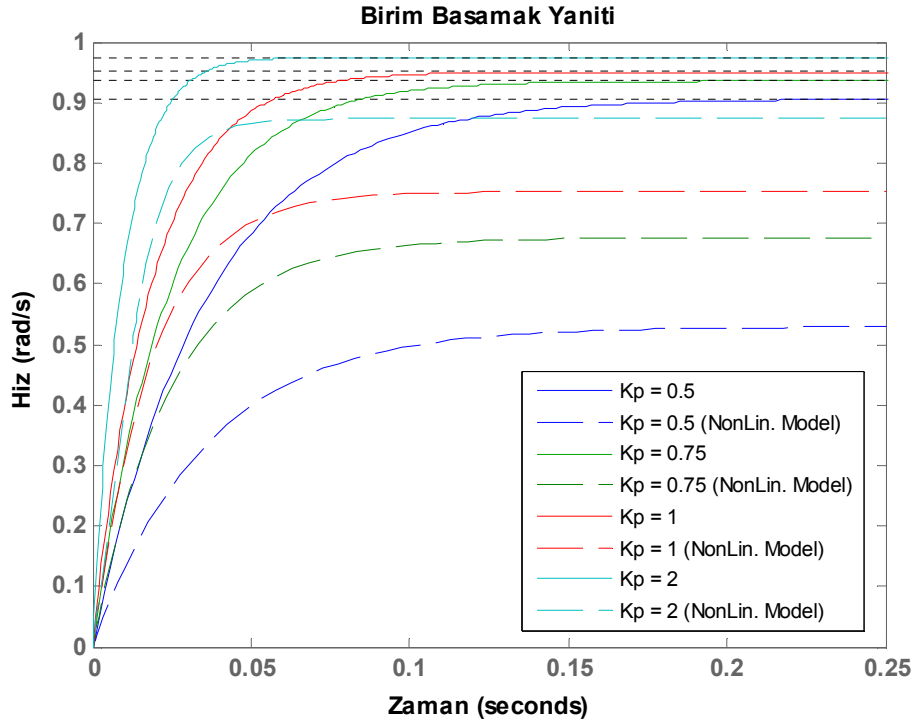
elde edilir. Görüldüğü üzere,  $e_{ss}$  kalıcı durum hatası sıfırdan farklıdır ve artan  $K_p$  ile azalacaktır.

$\Omega(s)/R(s)$  denetim sisteminin açık döngü kök yer eğrileri grafiği incelenirse (Şekil 3.57), sahip olabileceği köklerin negatif gerçekteksel ile üzerinde olduğu görülür. Bu, sistemin tüm  $K_p$  değerleri için kararlı olduğu ve titreşimsiz sönümlü olduğunu gösterir. Ayrıca, kökler dikey eksenden uzaklaştıkça (sola doğru) sistemin kazancının artacağı görülür. Artan sistem kazancı ile birlikte, sistemin zaman sabiti azalacak ve yanıt hızı artacaktır. O halde, artan  $K_p$  ile birlikte sistemin yanıt hızının artması beklenir.



Şekil 3.57. P denetimli sistemin açık döngü kök-yer eğrileri grafiği

$G_{\Omega}(s) = K_p$  olarak tanımlanan denetleyici, farklı ve artan orantı kazancı ( $K_p$ ) değerleri için denenmiştir ve Şekil 3.58’de ve Çizelge 3.9’da verilen birim basamak yanıtları elde edilmiştir.



Şekil 3.58. P denetleyici ile oluşturulan denetim sisteminin farklı orantı kazançlarına göre birim basamak yanıtı

Çizelge 3.9. P denetimli sistemin birim basamak yanıtı

Orantı Kazancı	Kalıcı Durum Yanıtı	Kalıcı Durum Hatası
0.5	0.9058 rad/s	0.0942 rad/s
0.75	0.9352 rad/s	0.0648 rad/s
1	0.9506 rad/s	0.0494 rad/s
2	0.9747 rad/s	0.0253 rad/s

Şekil 3.58'den ve Çizelge 3.9'dan görüldüğü üzere sistem, referans hız girişini belirli bir kalıcı durum hatası ile takip edebilmektedir. Bu kalıcı durum hatası, doğrusal olmayan denetim sisteminde daha da fazladır.

Doğrusal denetim sistemi modeli, sistemin doğrudan transfer fonksiyonudur. Oysa doğrusal olmayan denetim sistemi modelinde sürtünme torku ( $T_f$ ) etkisi dahildir ve denetleyicinin çıkışı sınırlandırılmıştır. Sürtünme torku, sayfa 93'deki anlatılan gerçek-zamanlı testlerde, ölçüm verilerinden yola çıkarak parametre kestirimi yöntemi ile

hesaplanmıştır. Çoğu deneyde yaklaşık  $\pm\%20$  belirsizlikle  $T_f = 0.14 Nm$  olarak hesaplanmıştır.

Bu kalıcı durum hatası, denklem (3.21) ile temsil edilen  $\Omega(s)/C(s)$  sistemin Tip-0 olmasının bir sonucudur. Basamak girişlerindeki kalıcı durum hatasını ortadan kaldırmak için bu sisteme bir integral eleman dahil edilmelidir. Bundan dolayı hız denetim sistemi, PI türünde denetleyici ile oluşturularak, sistemin Tip-1 olması sağlanabilir.

Tez konusu olan servo sistemin hız denetiminde kalıcı durum hatası istenmediğinden, P denetleyici bu çalışma için uygun değildir. Bundan dolayı PI denetleyici tasarımları test edilmiştir ve uygun olanı çalışma için seçilmiştir.

## PI Denetimli Sistemin Tasarımı

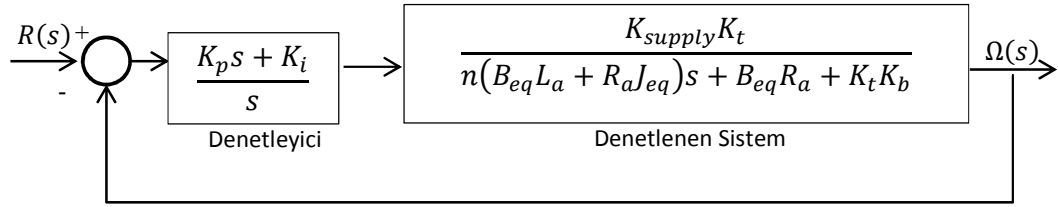
Bir PI denetim yasası

$$G_{\Omega} = K_p + K_i \frac{1}{s} \quad (3.25)$$

şeklinde verilir (Yüksel 2011) ve burada  $K_p$ ,  $K_i$  sırası ile orantı ve integral kazancı anlamına gelir. Şekil 3.59'da verilen kapalı döngü blok diyagramı indirgenirse, denetim sisteminin transfer fonksiyonu

$$\frac{\Omega(s)}{R(s)} = \frac{K_{supply}(K_p s + K_i K_t)}{R_a J_{eq} n s^2 + (B_{eq} R_a + K_t K_p + K_{supply} K_p) s + K_i K_{supply} K_t}$$

olarak elde edilir.



Şekil 3.59. PI denetimli denetim sistemi (doğrusal model)

$R(s) = 1/s$  basamak başvuru girişi,  $GH(s)$  Şekil 3.59'daki sistemin açık döngü transfer fonksiyonu

$$GH(s) = \frac{K_{supply} K_t (K_p s + K_i)}{n (n(B_{eq} L_a + R_a J_{eq}) s + B_{eq} R_a + K_t K_b) s}$$

olmak üzere, sistemin kalıcı durum hatası  $e_{ss}$  elde edilmek istenirse, son değer teoreminden

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} s E(s) = \lim_{s \rightarrow 0} \frac{s}{1 + GH(s)} R(s)$$

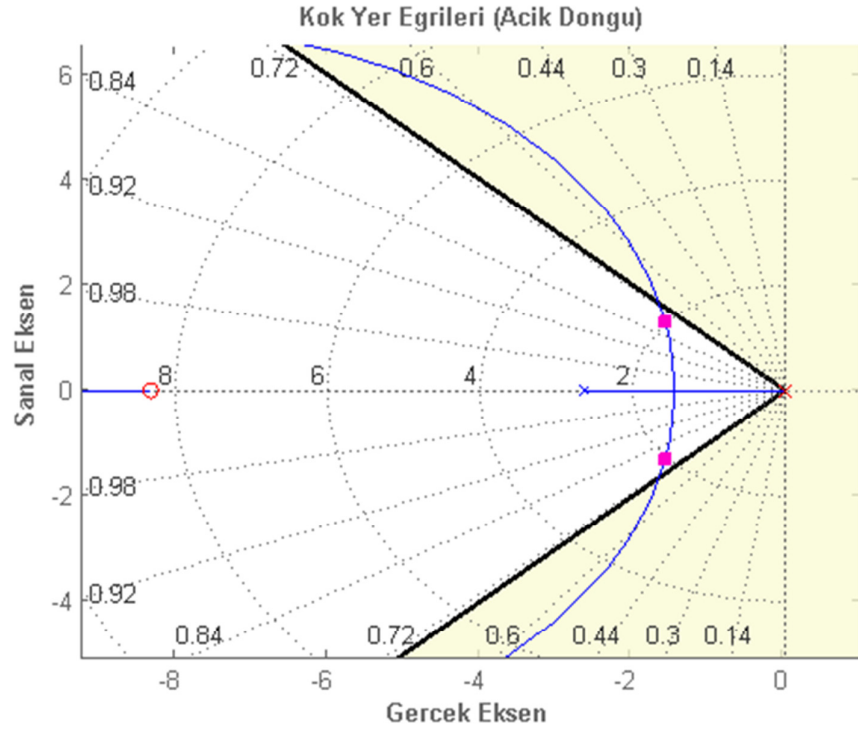
$$e_{ss} = \lim_{s \rightarrow 0} \frac{(n(B_{eq} L_a + R_a J_{eq}) s + B_{eq} R_a + K_t K_b) s}{(n(B_{eq} L_a + R_a J_{eq}) s + B_{eq} R_a + K_t K_b) s + K_{supply} K_t (K_p s + K_i)} = 0$$



elde edilir. Görüldüğü üzere,  $r(t) = u(t)$  birim basamak referans girişine karşılık denetim sisteminin kalıcı durum hatası  $e_{ss} = 0$  olacaktır. Bu da, PI denetimli sistemin kalıcı durum hatasını tamamen gidereceğini gösterir.

Uygun  $K_p$  ve  $K_i$  katsayılarının belirlenmesinde Simulink Control Design yazılımının Control and Estimation Tools Manager (denetim ve tahmin araçları yöneticisi) uygulamasından faydalanılmıştır. Bu araç, doğrusal ve zamanla değişmeyen (LTI) sistemlerin doğrusal analizlerinin gerçekleştirilmesine ve denetleyicilerin etkileşimli olarak hesaplanmasına olanak tanımaktadır.

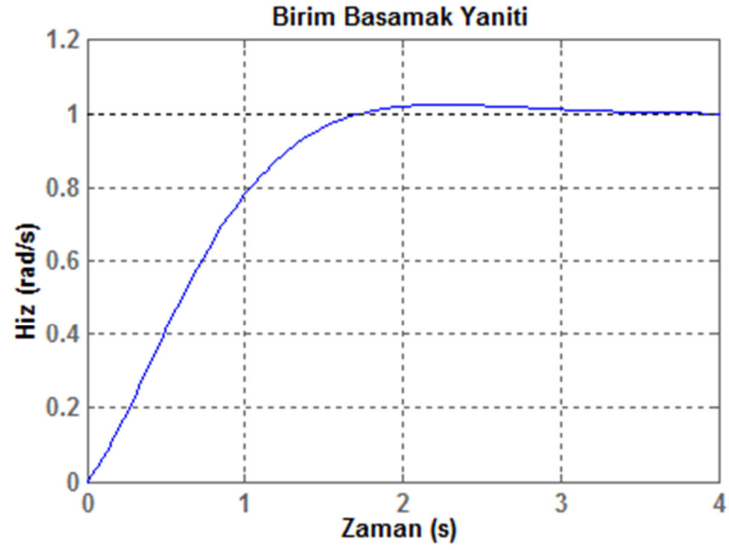
Denetleyici tasarlanırken kök-yer eğrilerinden, analiz edilirken de birim basamak yanıt grafiğinden faydalanılmıştır. Control and Estimation Tools Manager ortamında, kök-yer eğrisi grafiklerine gereksinim ölçütleri eklenebilmektedir. Şekil 3.57’de P denetimli sistemin kök-yer eğrisine  $\xi > 1/\sqrt{2}$  tasarım ölçütü (%4.32 ve altı aşım oranına denk gelen sönüm oranıdır) tanımlanmıştır. Kökler açık renkli bölgede kaldığı sürece (Şekil 3.60) bu ölçütün sağlanacağı anlamına gelmektedir. Orijinde bir adet kutup ve açık renkli bölge içerisinde bir adet gerçek sıfır ilave edilecek sistemin PI denetimli olması sağlanmıştır.



Şekil 3.60. PI denetimli sistemin (başlangıç ayarları ile) açık-döngü kök-yer eğrileri grafiği

Şekil 3.60'daki yapılanmaya göre  $K_p = 0.0098$  ve  $K_i = 0.08202$ 'dir ve sistemin birim basamak yanıtı Şekil 3.61'de verilmiştir. Mevcut denetim sisteminin açık döngü transfer fonksiyonu (kutup-sıfır-kazanç gösterimi) aşağıdaki gibidir:

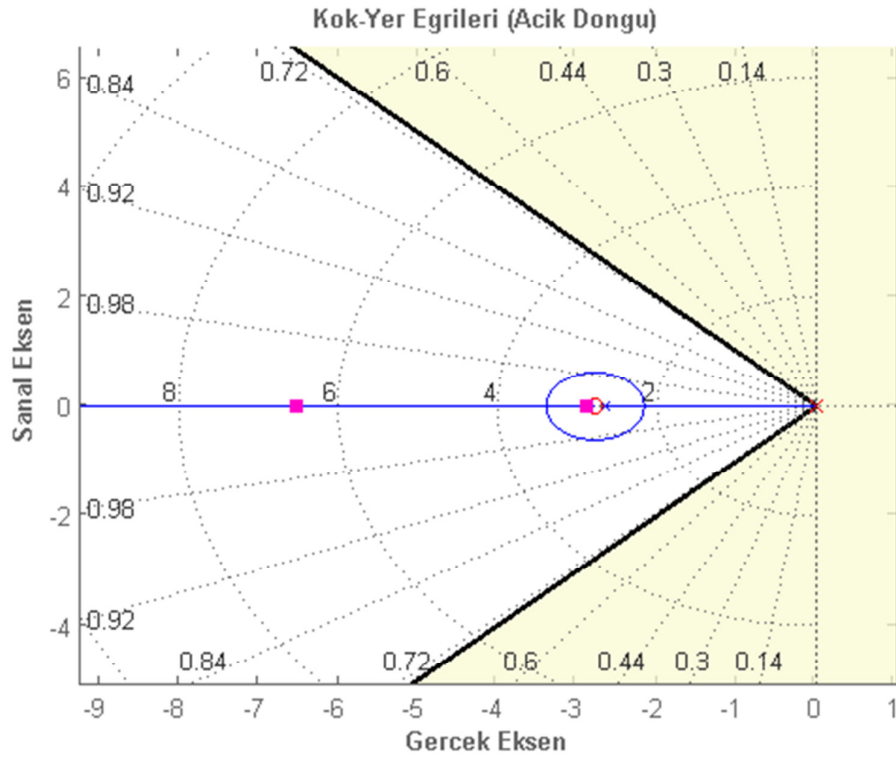
$$\frac{\Omega(s)}{E(s)} = \frac{0.49992(s + 8.329)}{s^2 + 3.14s + 4.164}$$



Şekil 3.61. Başlangıç ayarları ile PI denetimli sistemin birim basamak yaniti

Şekil 3.61'den görüldüğü üzere, aşım oranı %2.36, oturma zamanı 2.59 s, sönüm oranı 0.7694, doğal frekansı 2.0406 rad/s ve kalıcı durum hatası 0'dır.

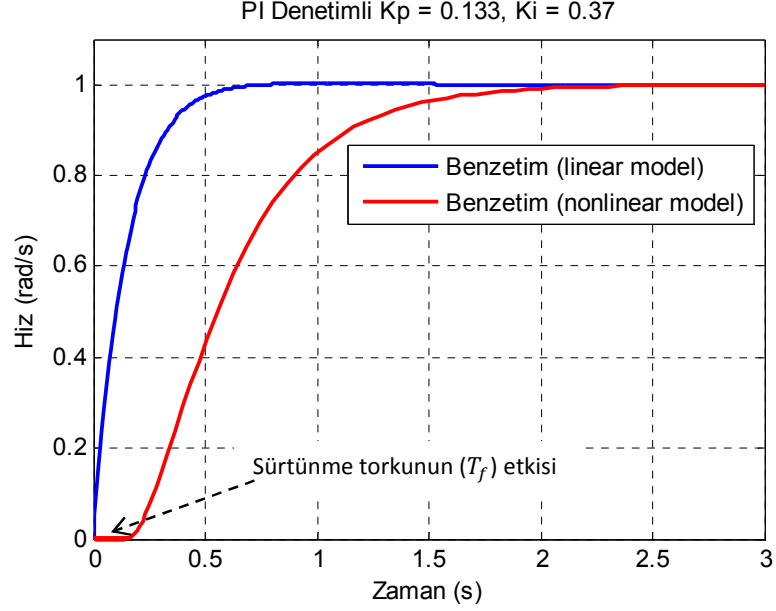
PI denetimle kalıcı durum hatası giderilmesine karşın, sistem bu sefer de yavaşlamıştır. Sistem, daha hızlı yanıt verecek şekilde yanıtının iyileştirilmesi gerekmektedir. Bunun için, denetleyicinin çıkışını yani açık döngü sistemin kazancını yükseltecek şekilde kök-yer eğrileri yeniden düzenlenmiştir. Açık döngü sistemin, Control and Estimation Tools ile etkileşimli olarak yeniden ayarlanmış kök-yer eğrileri Şekil 3.62'de verilmiştir.



Şekil 3.62. Düzenlenmiş ayarlar ile PI denetimli sistemin açık-döngü kök-yer eğrileri grafiği

Şekil 3.62'deki yapılanmaya göre  $K_p = 0.1332$  ve  $K_i = 0.3699$ 'tür ve sistemin birim basamak yanıtı Şekil 3.63'de verilmiştir. Mevcut denetim sisteminin açık döngü transfer fonksiyonu (kutup-sıfır-kazanç gösterimi) aşağıdaki gibidir:

$$\frac{\Omega(s)}{E(s)} = \frac{6.764(s + 2.777)}{(s + 2.878)(s + 6.526)}$$



Şekil 3.63. Düzenlenmiş ayarlar ile PI denetimli sistemin birim basamak yanıtı

Tasarlanan denetleyicilerin hem doğrusal hem de doğrusal olmayan denetim sistemi ile test edilmiştir. Kök yer eğrileri, frekans yanıtı vs. gibi denetim sistemi tasarım yöntemleri sadece doğrusal modellere uygulanabildiğinden, denetim sistemi doğrusal modeller (transfer fonksiyonları vs.) ile tasarlanmıştır. Denetim sisteminin yanıtı analiz edilirken de doğrusal olmayan modeller kullanılmıştır. Bunun nedeni, doğrusal modellerin ile doğrusalsızlıkların dahil edildiği modellerin yanıtı farklılık göstermektedir.

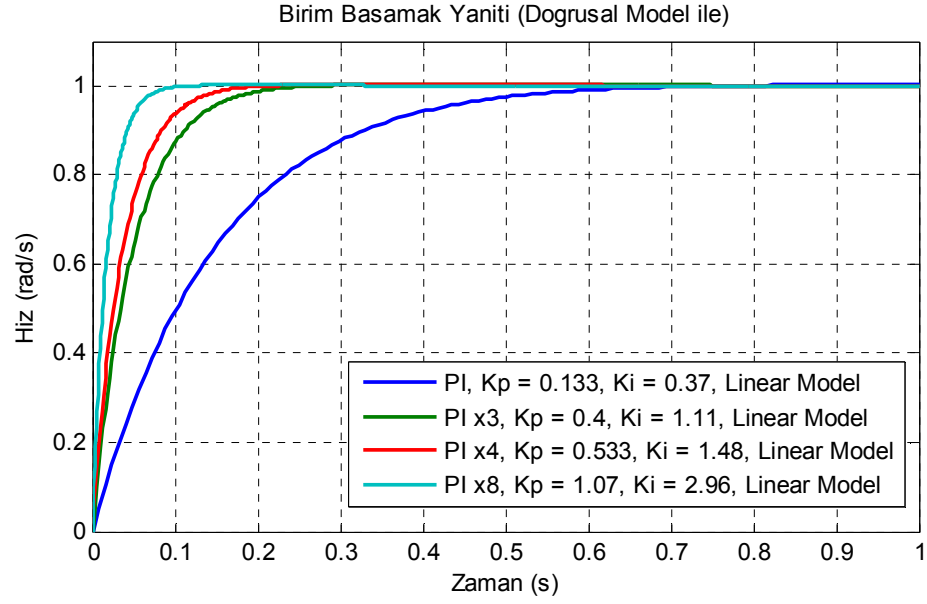
Şekil 3.63'deki doğrusal modelin birim basamak yanıtına göre, sistemin aşım oranı %0.211, oturma zamanı 0.53 s, sönüm oranı 1 ve kalıcı durum hatası 0'dır. Sistem, bir önceki yapılamaya göre daha hızlıdır. Doğrusal olmayan modelin oturma zamanı ise 1.71 s ve aşım oranı %0'dır. Doğrusal olmayan modelde sürtünme torku nedeni ile, denetleyicinin çıkışı belirli bir seviyeye ulaşana kadar sistem hareketsiz kalmıştır. Böylece başlangıçta 0.15 s kadar bir gecikme meydana gelmiştir.

Sistemdeki doğrusalsizliklar nedeni ile bu yapılanma yetersizdir. Başlangıçtaki gecikmeleri ortadan kaldırmak ve sistemin yanıtını hızlandırmak için denetim sisteminin kazancı sırası ile 3, 4 ve 8 katına çıkarılarak  $K_p$  ve  $K_i$  kazanç katsayıları yeniden düzenlenmiştir ve topluca özeti **Çizelge 3.10**'da verilmiştir.

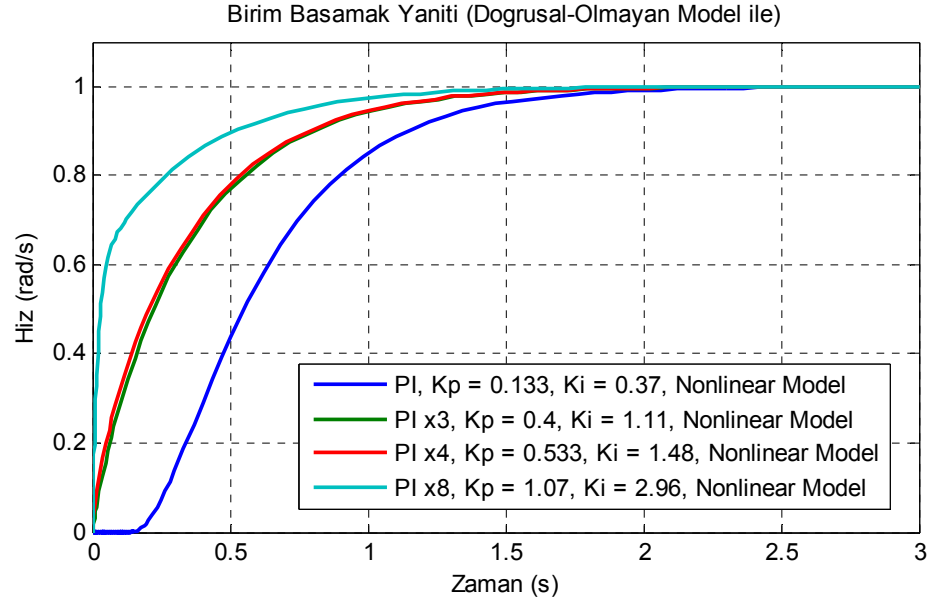
**Çizelge 3.10.** Aday Denetleyiciler ve Özellikleri

Tanım	Denetleyicinin Kazançları		Denetim Sisteminin Açık-Döngü T.F.
	$K_p$	$K_i$	
PI (özgün) (PI 1x)	0.133	0.37	$\frac{\Omega(s)}{E(s)} = \frac{6.7679(s + 2.777)}{s(s + 2.64)}$
PI x3	0.4	1.11	$\frac{\Omega(s)}{E(s)} = \frac{20.292(s + 2.777)}{s(s + 2.64)}$
PI x4	0.533	1.48	$\frac{\Omega(s)}{E(s)} = \frac{27.0559(s + 2.777)}{s(s + 2.64)}$
PI x8	1.07	2.96	$\frac{\Omega(s)}{E(s)} = \frac{54.1119(s + 2.777)}{s(s + 2.64)}$

Çizelge 3.10'da verilen denetleyiciler ile oluşturulan denetim sistemlerinin birim basamak yanıtları Şekil 3.64 ve Şekil 3.65'de verilmiştir. Şekil 3.64'deki yanıtlar doğrusallaştırılmış denetim sistemi ile, Şekil 3.65'de yanıtlar ise doğrusal olmayan denetim sistemi ile elde edilmiştir.



**Şekil 3.64.** Çeşitli PI yapılanmaları ile doğrusallaştırılmış denetim sisteminin birim basamak yanıtı



**Şekil 3.65.** Çeşitli PI yapılanmaları ile doğrusal-olmayan denetim sisteminin birim basamak yanıtı

Şekil 3.64 ve Şekil 3.65’de görüldüğü üzere artan sistem kazancı ile sistemin yanıtı hızlanmaktadır. Ayrıca, Şekil 3.65’de görüldüğü üzere, artan sistem kazancı ile sürtünme torkundan kaynaklanan başlangıç gecikmesi giderilmiştir.

Gerçek sistem üzerinde gerçekleştirilen testlerde (sayfa 93’de ayrıntılı verilmiştir), özgün-PI denetleyicisinin kazancı 4 katına çıkana kadar kararlı sonuçlar alındığı gözlenmiştir. Fakat daha yüksek kazançlarda, sistem kararsızlaşmaktadır ve referans hız değeri civarında titreşimler meydana gelmektedir. Bu nedenden dolayı, yanıt hızı ve kararlılık göz önünde bulundurulduğunda, bu tez çalışması için en uygun denetleyici Çizelge 3.10’da verilen PI-4x yapılanmasıdır.

Denetim sisteminin uygun bir davranış sergilediği benzetim ortamında sınıandıktan sonra, tasarlanan denetleyiciyi deney düzeneği üzerinde gerçek-zamanlı olarak test edilmesi gerekli görülmüştür. Şekil 3.40 (sayfa 57) göz önünde bulundurulduğunda, denetlenen sistem üzerinden açısal hız ölçümü yapılması gereklidir. DC motor üzerinde hız ölçümü yapan bir algılayıcı bulunmadığı için, anlık hız bilgisi sadece konum algılayıcısından (*encoder*) sağlanan veriler ile tahmin edilmek zorundadır.

Bir encoderden sağlanan konum verilerinden rotor hızını elde edebilmenin çeşitli yöntemleri vardır. Bunlardan biri frekans ölçümü yöntemi, diğeri ise periyod ölçümü yöntemidir. Frekans-ölçümü yöntemi, belirli bir zaman aralığındaki *encoder*’in ürettiği darbeleri saymaya dayalıdır. Periyod ölçümü yöntemi ise her bir darbenin genişliğini ölçmeye dayalıdır (Briz ve ark. 1994). Ayrıca Kalman filtresi yöntemi ve Luenberger gözlemcisi gibi gözlemci (*observer*) tabanlı hız ölçümü yöntemleri de mevcuttur (Petrella ve ark. 2007). Bu çalışmada, basitliği ve kolay uygulanabilirliği nedeni ile frekans-ölçümü yöntemi tercih edilmiştir.

Frekans-ölçümü yönteminde rotor hızı, doğrudan encoder darbelerinin frekansını hesaplayarak elde edilir. Sabit-aralıklı ve önceden belirlenmiş bir gözlem zaman-diliminde gözlemlenen darbeler sayılır. Hızın bu zaman-dilimleri içerisinde sabit olduğu kabul edilir ve (3.26) ile hız yaklaşık olarak hesaplanır:

$$\omega = \frac{d\theta}{dt} \cong \frac{\Delta\theta}{T_{sc}} \cong \frac{2\pi\Delta N}{N_p T_{sc}} [rad/s] \quad (3.26)$$

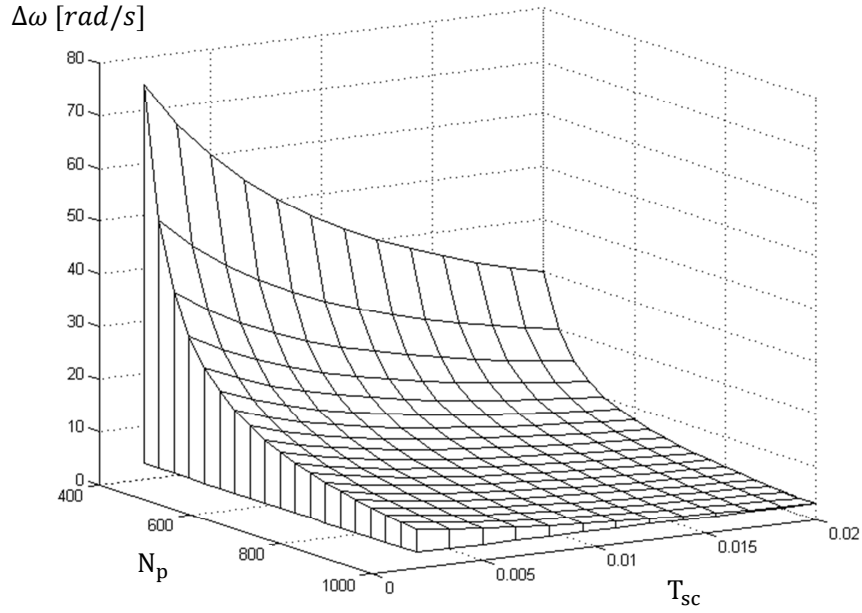


Burada  $N_p$ , encoderin bir turu başına ürettiği darbe sayıdır;  $T_{sc}$  zaman dilimidir ve  $\Delta N$  ise bu zaman dilimi içerisinde gözlemlenen darbe sayısıdır. Ancak, encoder darbeleri ve gözlem-dilimi arasındaki eş-zamanlılığın yetersizliğinden, ölçülen  $\Delta N$  darbe sayısı tam belirli değildir. Bu da o zaman-dilimi içerisindeki ortalama hızın üzerine niceleme (*quantisation*) hataları bindirir. Bu  $\Delta\omega$  niceleme hatası,

$$\Delta\omega = \frac{2\pi}{N_p T_{sc}} [\text{rad/s}] \quad (3.27)$$

olarak verilir.

Denklem (3.27)'nin bir sonucu olarak, niceleme hataları çalışma hızından bağımsızdır ve sadece *encoderin* darbe/devir sayısına ve gözlem-dilimine bağlıdır (Petrella ve ark. 2007) ve bu ilişki Şekil 3.66'da görülmektedir.



**Şekil 3.66.** Niceleme hatalarının darbe/devir sayısı ve zaman-dilimi ile ilişkisi (Petrella ve ark. 2007)

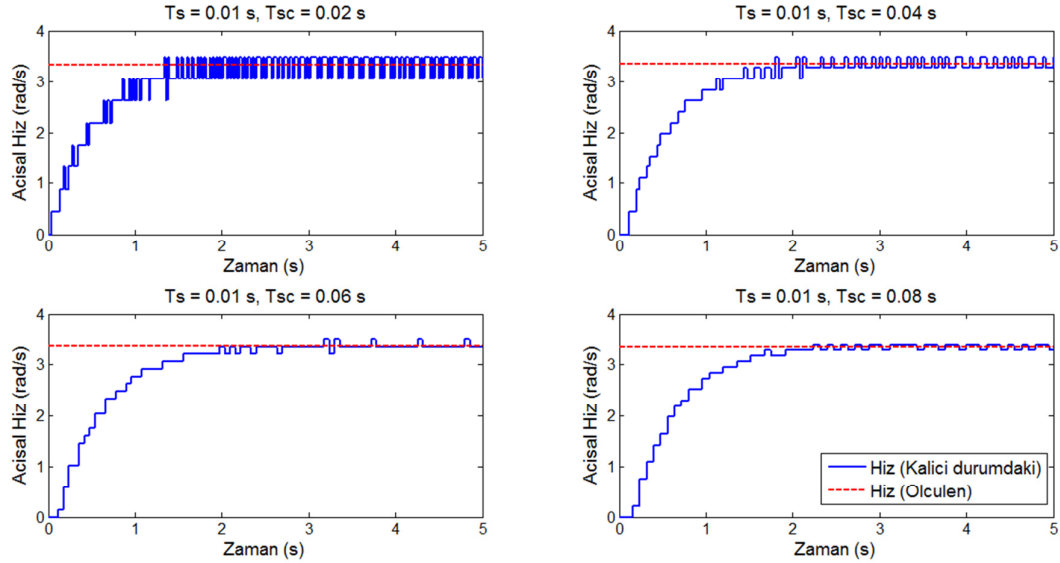
Burada amaç,  $\Delta\omega$  hız hatalarını en düşük yapmaktır. (3.27)'ün bir sonucu olarak, hız hatalarını en düşük yapmak için yüksek hassasiyetli bir encoder (yüksek  $N_p$ ) kullanmak ve  $T_{sc}$  zaman-dilimini yüksek tutmak gerekir. Fakat, hız hesaplanırken yine  $T_{sc}$  kadar bir geri-besleme gecikmesi söz konusudur ve artan  $T_{sc}$  ile birlikte gerçek hız ile ölçülen

hız arasında artan bir faz farkı oluşacaktır. Yüksek faz farkı, sistemin kararlılığını azaltacağından  $T_{sc}$  sınırlı bir değere kadar yüksek tutulabilir (Briz ve ark. 1994).

Bu çalışmada kullanılan DC motora ait encoderin darbe/devir sayısı  $N_p = 720$ 'dir ve uygun gözlem zaman-diliminin ( $T_{sc}$ ) belirlenmesi gerekli olmuştur. Uygun  $T_{sc}$ 'yi belirleyebilmek için bu çalışmada kullanılan DC motor deney düzeneğinde, sabit bir *duty* oranında boşa çalıştırılmıştır ve denklem (3.26)'daki algoritma ile hız ölçümleri alınmıştır. Her durumda sistemin kalıcı durumdaki hız deneysel olarak 3.3423 rad/s hesaplanmıştır. Elde edilen hız ölçümleri, farklı  $T_{sc}$  zaman-dilimleri için tekrarlanmıştır ve Şekil 3.67'de karşılaştırmalı olarak verilmiştir. Görüldüğü üzere, gözlem zaman-dilimi, örnekleme zamanının ( $T_s$ ) yaklaşık 6 veya 8 katı olarak seçildiğinde, niceleme hataları en düşük olmaktadır. Bu çalışmada

$$T_{sc} = 6T_s \quad (3.28)$$

olarak seçilmiştir.



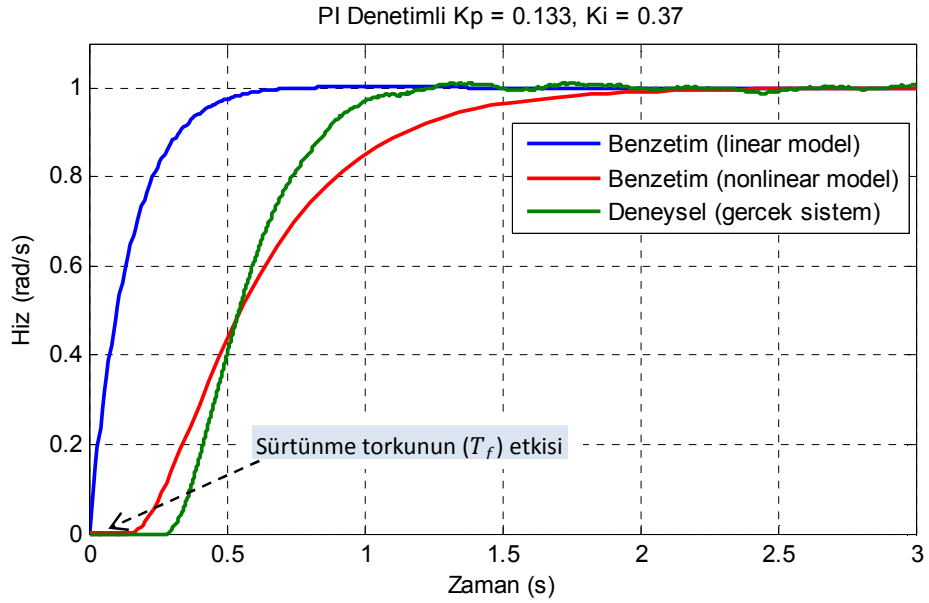
**Şekil 3.67.** Farklı gözlem zaman-dilimlerinde ölçülen hızlar

Denetleyici gerçek sistem üzerinde test edilirken hız geri beslemesi Denklem (3.26) ve (3.28)'deki esaslara göre elde edilmiştir.

### Hız Denetleyicisinin Gerçek Zamanlı Testleri

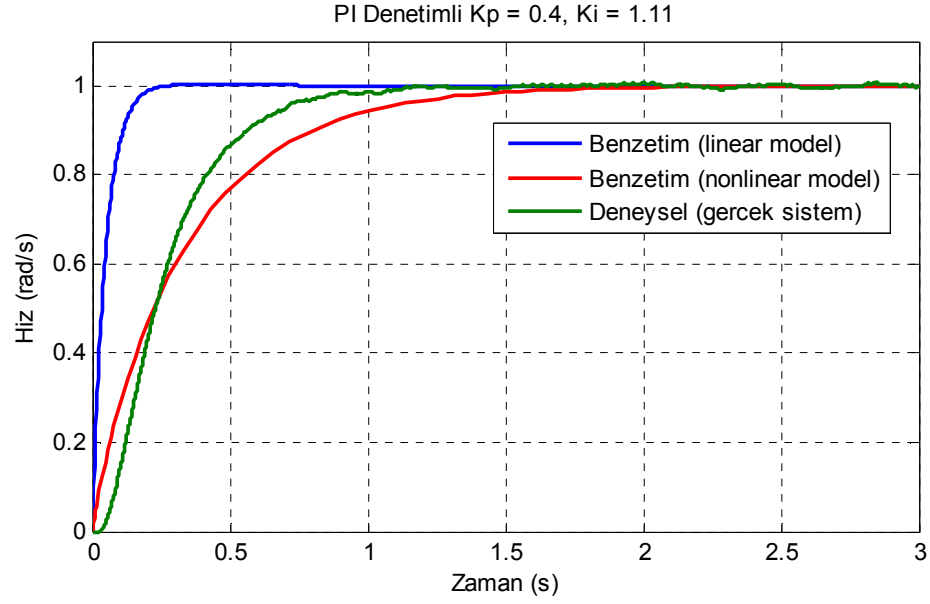
Çizelge 3.10’da verilen aday hız denetleyicileri, deney düzeneği üzerinde gerçek sisteme uygulanmış, test edilmiş ve sonuçları bu kısımda tartışılmıştır.

DC motora (gerçek sistem), “PI 1x” denetleyicisi uygulandığında, benzetim çalışmalarından da tahmin edildiği gibi, sistemin yanıtı sürtünme torku nedeniyle gecikmeli ve yavaştır (Şekil 3.68). Oturma zamanı 1.04 s olarak ölçülmüştür.

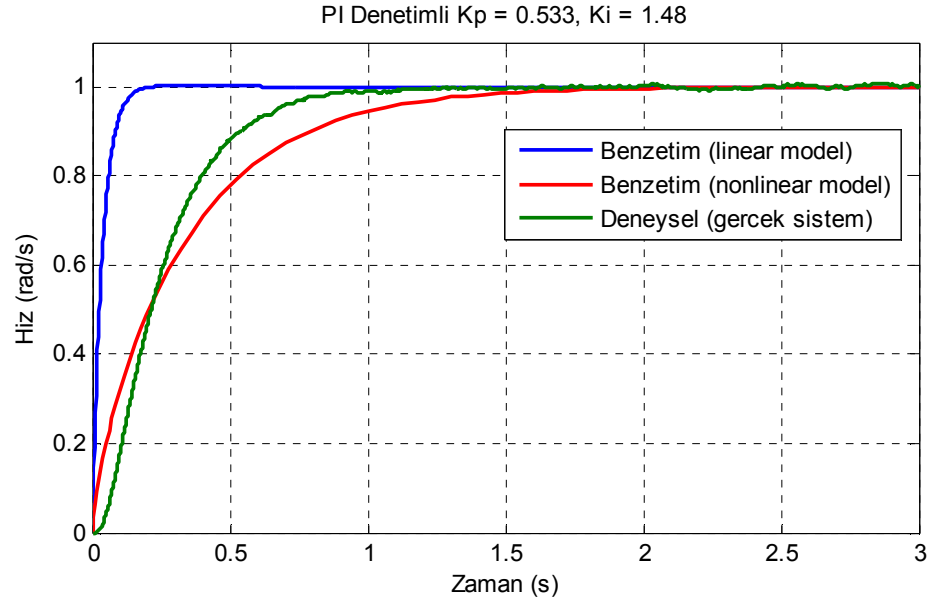


Şekil 3.68. ”PI 1x” Denetimli Sistemin Birim Basamak Yanıtı (Gerçek Zamanlı Test)

“PI 3x” ve “PI 4x” ile denetlenen gerçek sistemin birim basamak yanıtı Şekil 3.69 ve Şekil 3.70’de verilmiştir. “PI 1x” yapılanmasına göre motorun yanıtı hızlanmıştır (oturma zamanı 0.86 ve 0.82 s) ve sürtünme torkundan kaynaklanan başlangıç gecikmesi ortadan kalkmıştır.

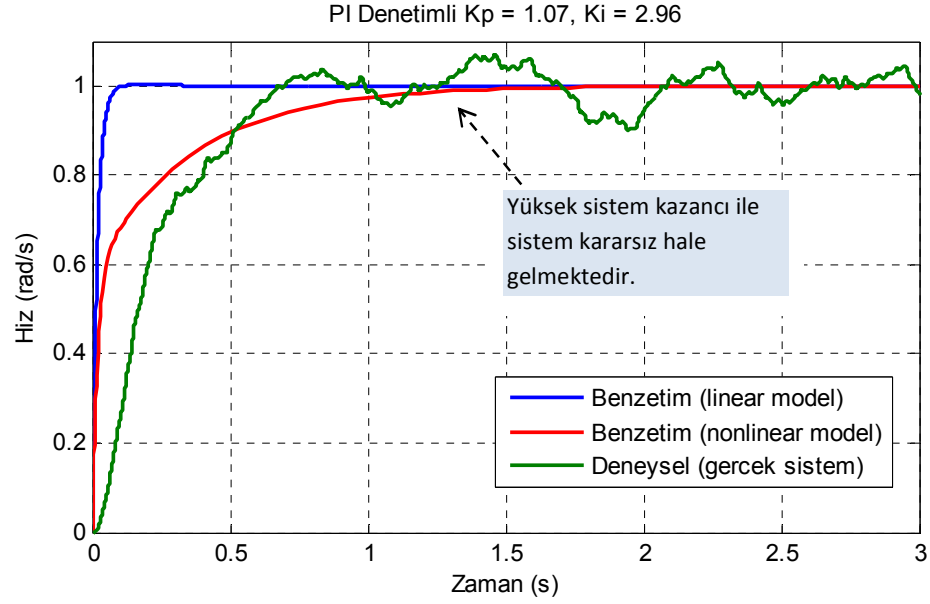


**Şekil 3.69.** "PI 3x" Denetimli Sistemin Birim Basamak Yanıtı (Gerçek Zamanlı Test)



**Şekil 3.70.** "PI 4x" Denetimli Sistemin Birim Basamak Yanıtı (Gerçek Zamanlı Test)

"PI 8x" ile denetlenen sistem benzetimlerde iyi bir sonuç vermesine karşın, gerçek sisteme uygulandığında yüksek sistem kazancından dolayı kararsız hale gelmiştir. Bu etki Şekil 3.71'de açıkça görülmektedir.



**Şekil 3.71.** "PI 4x" Denetimli Sistemin Birim Basamak Yanıtı (Gerçek Zamanlı Test)

Gerçek zamanlı testlerin toplu bir özeti Çizelge 3.11’de verilmiştir. Görüldüğü üzere, denetleyicinin artan kazancı ile birlikte sistemin yanıtı hızlanmaktadır, fakat belirli bir seviyeden sonra sistem kararsızlaşmaktadır.

**Çizelge 3.11.** Farklı denetim yapıları ile sistemin gerçek-zamanlı test sonuçları

Kullanılan Denetleyici	Denetleyicinin Kazancı	Sistemin Oturma Zamanı (Deneysel)
PI 1x	$K_G = 0.13323$	1.04 s
PI 3x	$3K_G = 0.39969$	0.86 s
PI 4x (en uygun)	$4K_G = 0.53293$	0.82 s
PI 8x	$8K_G = 1.0659$	$\infty$ (Kararsız)

Çizelge 3.11’den yola çıkarak, nesne takip sisteminde "PI 4x" ile isimlendirilen hız denetleyicisinin kullanılmasına karar verilmiştir. Seçilen denetleyiciye ilişkin özellikler Çizelge 3.10’da (sayfa 88) verilmiştir.

### 3.2.2.3. DC Motorun Konum Denetleyicisinin Tasarımı

Şekil 3.13’de (sayfa 29) nesne takip sisteminin, hız veya konum durumlarından birine sahip olabileceği verilmişti. DC motorun hız denetleyicisinin tasarımı ve doğrulanması bir önceki başlıkta verilmişti, bu başlıkta ise konum denetimi tasarımı incelenmiştir.

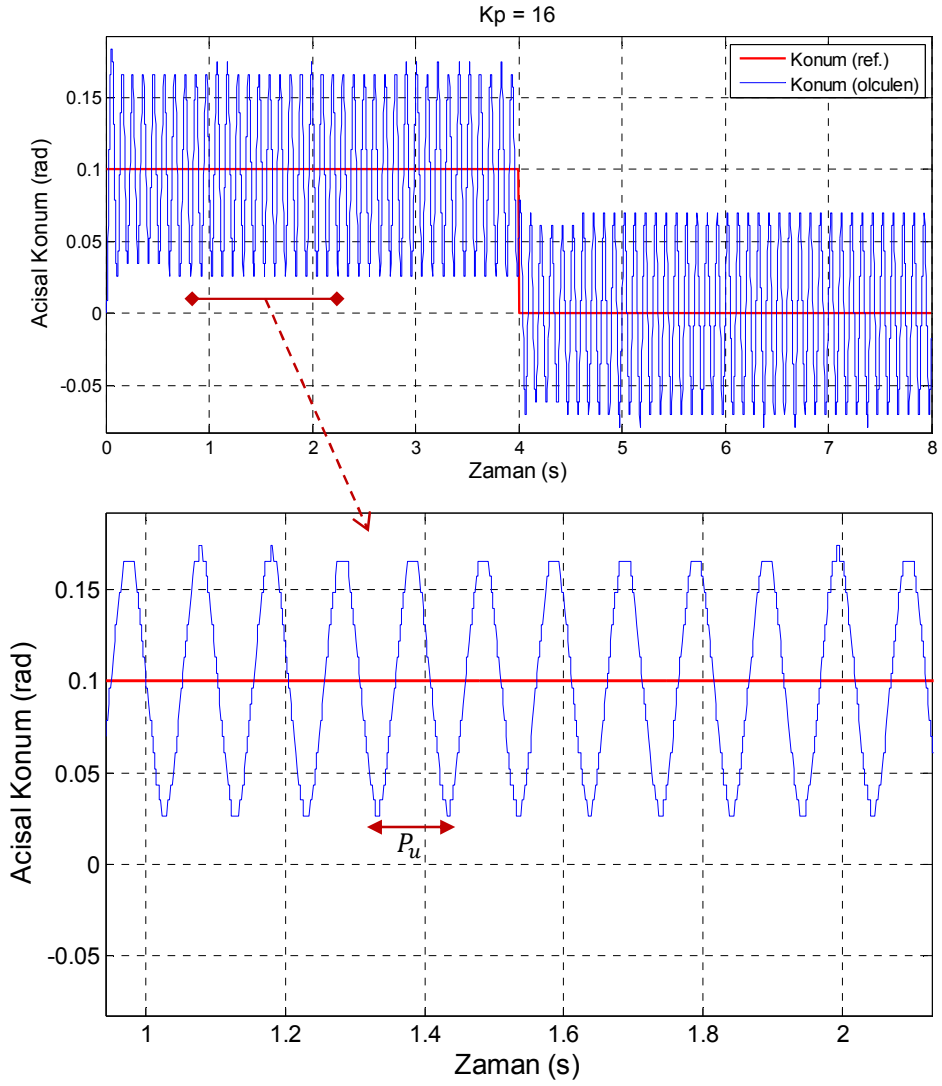
Sistemin mekanik kısımlarından (özellikle dişli kutusu) kaynaklanan statik sürtünme torku nedeniyle, DC motorun düşük gerilim seviyelerinde doğrusalsızlıklar mevcuttur. Bu doğrusalsızlıklar, önceki başlıklarda verilen sistem modelinin içerisine kısmen dahil edildiğinden, konum denetimi matematiksel modeller ile değil, gerçek sistem ile deneysel olarak tasarlanmıştır.

Konum denetleyicisi olarak PID yasası, PID denetleyicinin parametrelerinin ayarlanmasına yönelik ise Ziegler-Nichols yöntemi tercih edilmiştir. Denklem (3.21) ile temsil edilen  $\Theta(s)/C(s)$  sistemi Tip-1 bir sistemdir ve sisteme PID ile bir integral eleman sağlanması, sistemin rampa sinyalini izlerken oluşan kalıcı durum hatasını da ortadan kaldıracaktır. Ayrıca  $\Theta(s)/C(s)$  transfer fonksiyonunun payda ve pay mertebelerinin farkı 3 olduğu için Ziegler-Nichols yöntemini uygulamak mümkün olmaktadır.

Ziegler-Nichols yöntemi, geribeslemeli ve kapalı-döngü bir sistemin basamak giriş ile sürekli salınımda test edilmesine dayanır (Yüksel 2011). İlk aşamada denetim yasası olarak

$$G_{\Theta}(s) = K_p \quad (3.29)$$

oransal denetim uygulanmıştır.  $K_p$  kazancı, sistem sınır kararlı hale, başka deyişle sürekli salınım haline gelene kadar artırılmıştır.  $K_{pmax} = 16$  için sistemin sınır kararlı olduğu gözlemlenmiştir (Şekil 3.72.a).



Şekil 3.72. (a) ve (b). Konum denetim sisteminin sınır kararlıdaki yanıtı ve yakından görünümü

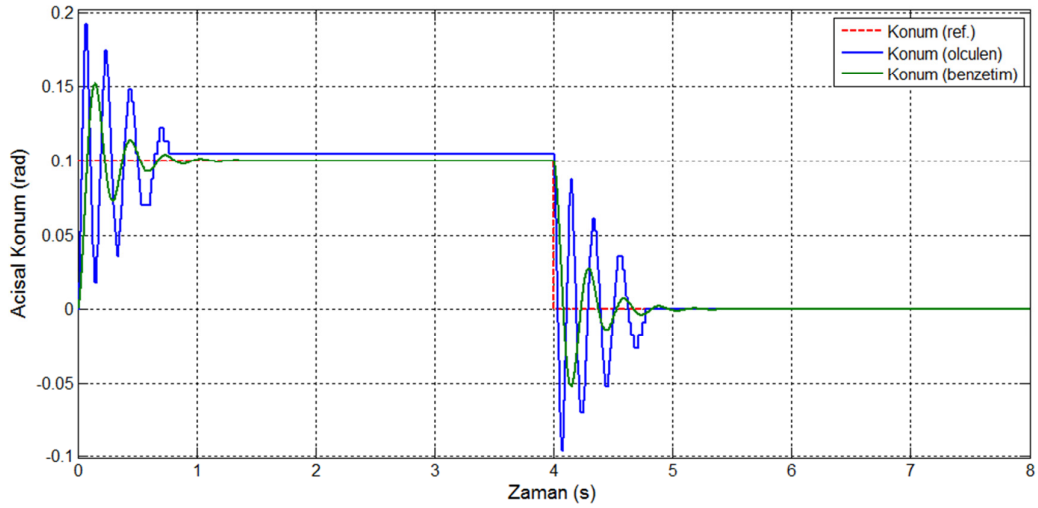
Denklem (3.30)'da verilen ardışık-PID denetim yasaından yola çıkarak

$$G_{\Theta}(s) = K_p \left( 1 + \frac{T_i}{s} + T_d s \right) \quad (3.30)$$

integral zaman sabiti ( $T_i$ ) ve türev sabiti ( $T_d$ ), Ziegler-Nichols yöntemine göre

$$\begin{aligned}
K_p &= 0.6K_{pmax} \\
T_i &= 0.825P_u \\
T_d &= 0.125P_u
\end{aligned}
\tag{3.31}$$

ile hesaplanır (Yüksel 2011). Burada  $P_u$  sürekli salınım periyotudur ve Şekil 3.72 (b) üzerinden  $P_u = 1.106$  s olarak hesaplanmıştır. Denetim sistemine ilişkin katsayılar denklem (3.32) ile hesaplanmış ve denetim sistemi hem benzetim ortamında hem de deney düzeneği üzerinde test edilmiştir ve basamak girişine karşılık yanıtı Şekil 3.73’de verilmiştir.



**Şekil 3.73.** Ayarlanmış katsayılar ile konum denetim sisteminin yanıtı

Şekil 3.73’de görüldüğü gibi, benzetim sonuçları ile gerçek sistem yanıtlarının kısmen eşleştiği görülmektedir. Bunun başlıca nedeni, motor milinin hareket edebilmesi için aşılması gereken sürtünme torkudur ( $T_f$ ). Denetim sinyali ilk defa oluşturulduğunda, sürtünme torku nedeni ile yük milinde açısız bir yer değiştirme o an meydana gelmeyeceğinden, denetleyicinin çıkışı hatayı hızlıca telafi edebilmek için yükselmektedir. Böylece aşım artmaktadır. Ayrıca referans konum  $0.1$  rad iken göreceli olarak düşük bir kalıcı durum hatası oluşmakta iken, referans konum  $0$  rad olduğunda kalıcı durum hatası olmamaktadır. Bunun nedeni, sürtünme torkunun ( $T_f$ ) motorun ileri yada geri yönde hareket ederken farklı değerde olmasıdır. Buna benzer doğrusalsızlıklar, Şekil 3.73’de benzetim modeline dahil edilmemiştir.



### 3.2.3. Durum Geçişleri Denetimi Altsistemi

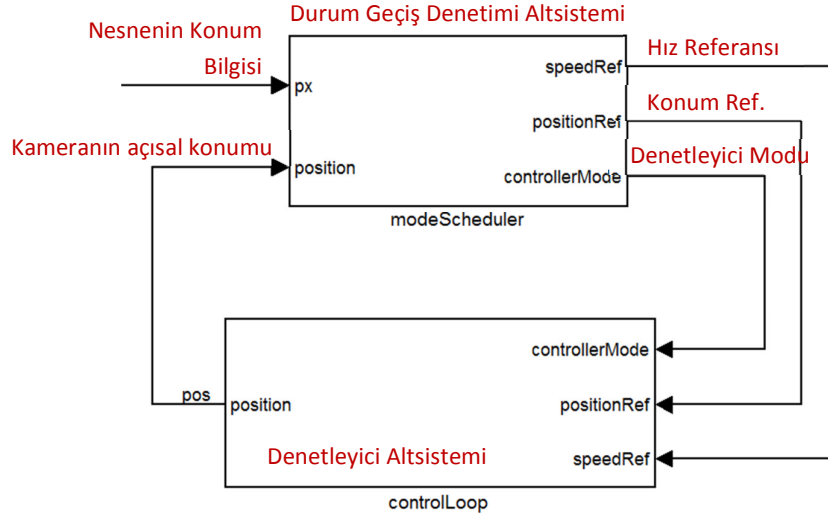
Nesne takip sisteminin çalışma ilkelerine göre, makinenin farklı çalışma modlarına sahip olduğu 3.1.2 numaralı başlıkta (sayfa 26) ayrıntılı bir şekilde anlatılmıştı. Nesnenin görüntü merkezi içerisinde veya dışında olmasına ve nesnenin görüş alanı içerisinde olup olmamasına göre makine farklı durumlara sahip olabilirdi. Makinenin alabileceği bu çalışma durumları ve geçiş koşulları Çizelge 3.12’de özetlenmiştir.

**Çizelge 3.12.** Makinenin Alabileceği Durumlar ve Geçiş Koşulları

<b>Çalışma Modu</b>	<b>Koşul</b>
Hız Denetimi Modu <i>Denetleyici Modu = 1</i>	Nesne merkez bölgesi dışında veya görüş alanında nesne algılanmamış ise $-10 > e_{px} \wedge e_{px} > +10$
Konum Denetimi Modu <i>Denetleyici Modu = 2</i>	Nesne merkez bölgesinde ise $-10 < e_{px} < +10$
Durma Modu <i>Denetleyici Modu = 1</i>	Nesne görüş alanı içerisinde değilse yada algılanmamışsa.
İlk Başlatma Bekleme Modu <i>Denetleyici Modu = 1</i>	Makine ilk defa başlatıldığında, 2 sn süresince.

Durum geçiş denetimi altsistemi farklı çalışma durumlarına karşılık, o an için denetleyicinin hız yada konum denetiminin hangisini gerçekleştireceğini belirler. Şekil 3.13’de (sayfa 29) verilen algoritmaya göre tasarlanan durum geçiş denetimleri, Stateflow ortamında tasarlanmıştır (Şekil 3.75).

Durum geçişleri denetimi altsistemi, eğer konum denetimi gerçekleştirilecekse konum referans değerini, hız denetimi gerçekleştirilecekse hız referans değerini denetleyiciye sağlar. Sözü edilen bu çalışma prensibi ile oluşturulan Simulink modelinin ilgili kısmı Şekil 3.74’de verilmiştir. Şekil 3.74’deki durum geçişleri denetimi ve denetleyici altsistemleri sırası ile Şekil 3.75 ve Şekil 3.76’da gösterilmiştir.

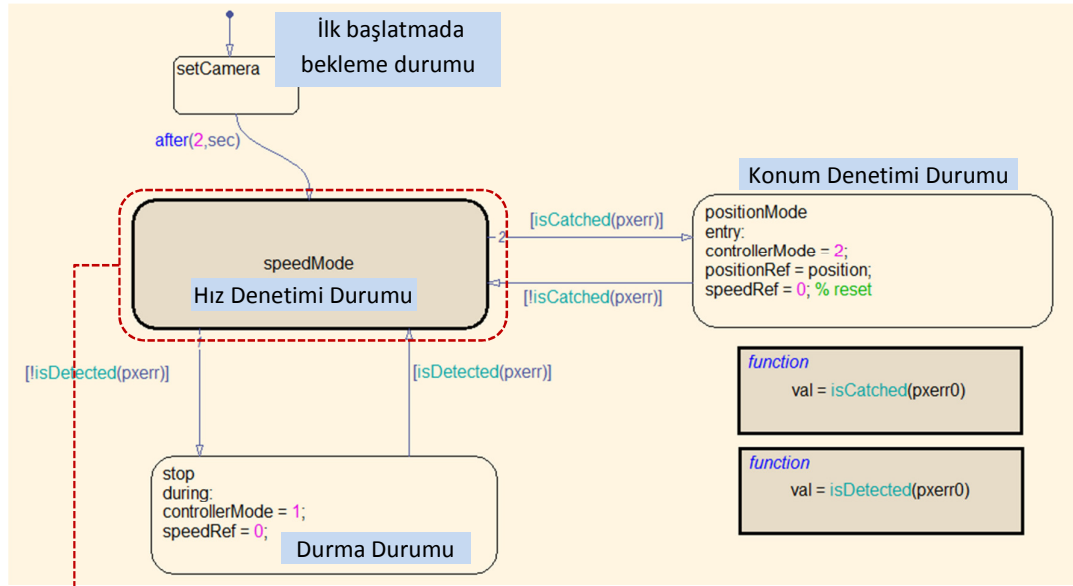


**Şekil 3.74.** Durum geçiş denetimi ve Denetleyici Altsistemleri

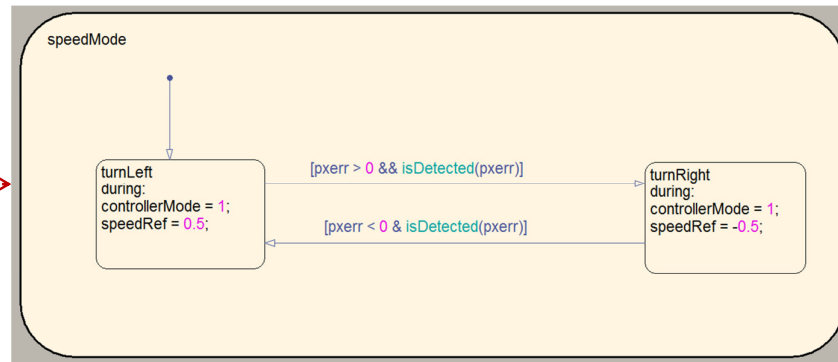
Şekil 3.74’de verilen bu Stateflow şemasında, nesnenin görüntü merkezinden sapma miktarı ( $e_{px}$ ) ve kameranın anlık açısai konumu ( $\theta$ ) giriş olarak sağlanmaktadır. Sapma miktarı ve kameranın açısai konum verilerinden yola çıkılarak

- hız denetleyicisinin referansı,
- konum denetleyicisinin referansı
- ve o an için hangi denetleyicinin etkin olacağını belirleyen denetleyici modu

sinyalleri hesaplanmakta ve denetleyici altsistemine giriş olarak gönderilmektedir. Durum geçiş denetimi altsisteminde yer alan Stateflow şeması Şekil 3.75’de verilmiştir.



(a)



(b)

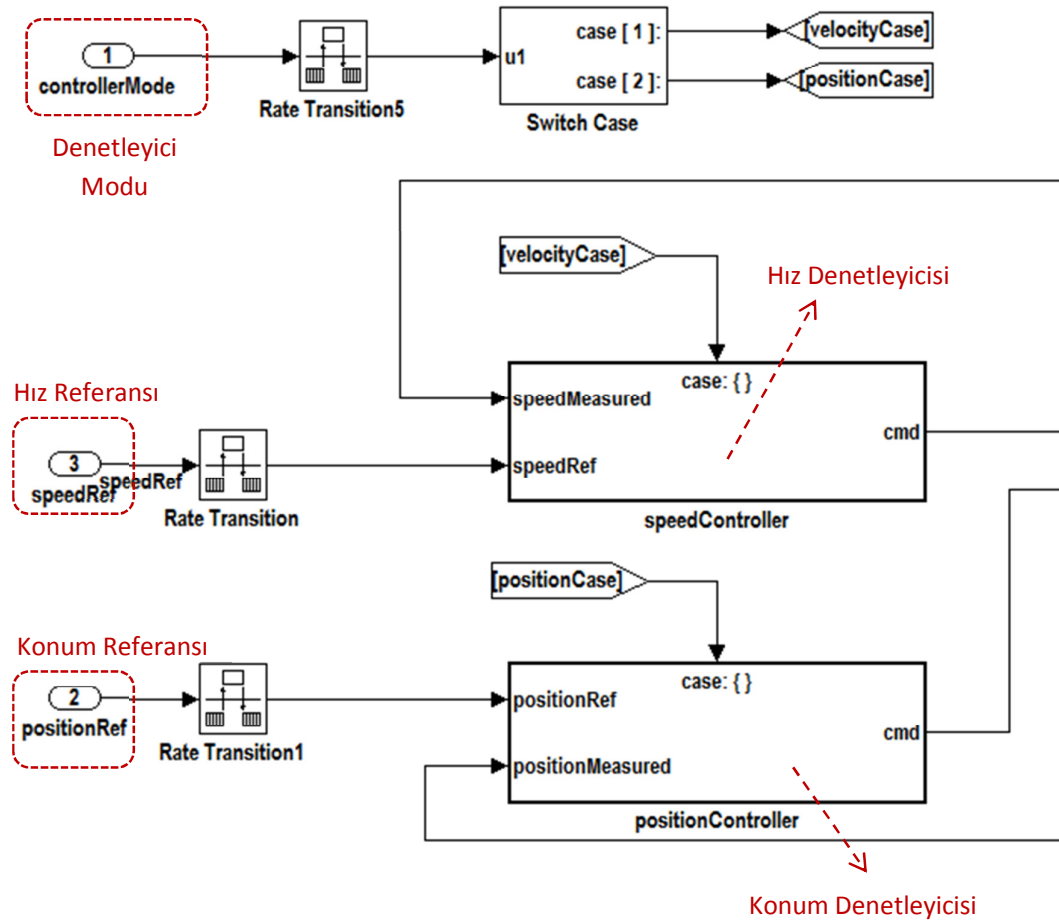
**Şekil 3.75.** Durum geçiş denetimine ilişkin Stateflow şeması

Şekil 3.75’de Stateflow şemasına göre,

- makine ilk başlatıldığında 2 sn kadar bekleme durumunda kalması istenmiştir. Bunun nedeni, kameranın başlatılabilmesi için donanımına süre tanımadır.
- eğer sapma miktarı  $-10 < e_{px} < +10$  aralığında ise makine “konum denetimi” durumuna geçecektir ve denetleyicinin konum referansı kameranın o anki açısıl konumu olacaktır ( $\theta_{ref} = \theta$ ).

- Eğer sapma miktarı, bu aralığın dışında ise makine “hız denetimi” moduna geçecektir ve  $e_{px} > 0$  yada  $e_{px} < 0$  olmasına göre denetleyicinin hız referansı  $\omega_{ref} = \pm 0.5 \text{ rad/s}$  olacaktır (Şekil 3.75b).
- Eğer nesne görüş alanı içerisinde algılanmamışsa makine “durma” durumuna geçecektir hız referansını  $\omega_{ref} = 0 \text{ rad/s}$  olarak belirleyecektir.

Şekil 3.74’de görülen denetleyici altsisteminin içeriği Şekil 3.76’da verilmiştir. Denetleyici altsisteminde hız ve konum denetleyicisi beraber bulunmaktadır. “Denetleyici modu” bilgisine göre hız ve konum denetleyicilerinden sadece biri etkin olacaktır. Bu geçiş, *switch-case* akış denetimi ile sağlanmıştır.



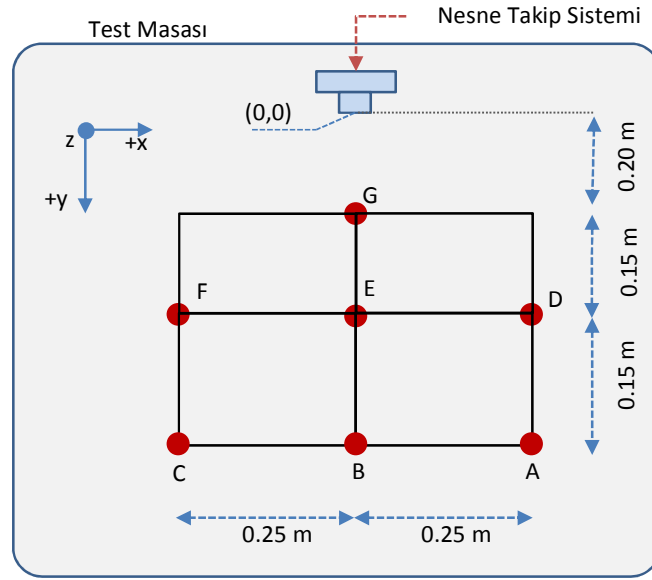
Şekil 3.76. Denetleyici Altsistemi

#### 4. BULGULAR VE TARTIŞMA

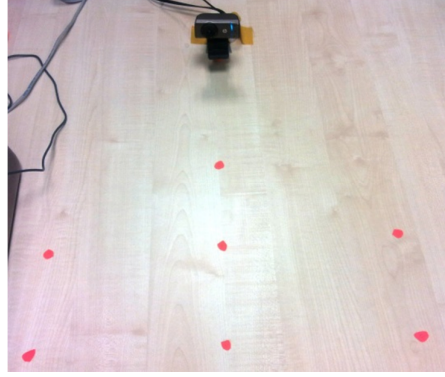
Bu tez kapsamında önceki başlıklarda, mekatronik bir sistem tasarımının bir çok unsurunu içeren “nesne takip sistemi”nin alt sistemleri ayrı ayrı tasarlanmış ve alt sistemlerin başarımları test edilmiştir. Bu başlık altında sistem bir bütün olarak ele alınmış ve bazı performans testleri uygulanmıştır.

##### 4.1. Takip Performansı Testleri

Nesne takip sisteminin takip performansını sınamak için test masası üzerinde bir deney ızgarası oluşturulmuştur (Şekil 4.1 ve Şekil 4.2). Bu ızgara üzerinde 7 adet farklı nokta belirlenmiştir. Takip performansları için nesne, bu ızgara üzerinde önceden tanımlanmış bir iz üzerinde, göreceli olarak düşük ve sabit bir hızda hareket ettirilmiştir.



Şekil 4.1. Deney Izgarası



**Şekil 4.2.** Deney masası ve işaretlenmiş noktalar

Takip performansını test etmek için iki farklı test türü uygulanmıştır. “Hareketsiz nesne testleri” ile sistemin tepki süreleri ölçülmüş, “hareketli nesne testleri” ile de takip performansı test edilmiştir.

Deneyle gerçekleştirilirken, sistem üzerinden

- nesnenin görüntü içerisindeki piksel cinsinden zamana göre konumu,
- kameranın zamana göre açısal konumu,
- denetleyicinin modu,
- konum denetim sisteminin referansı

bilgileri toplanmış, kaydedilmiş ve görselleştirilmiştir. Deney sonuçları bu bilgiler ile yorumlanmıştır.

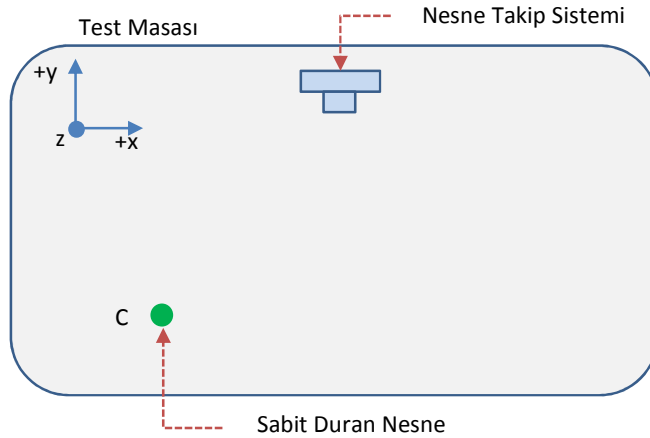
Deney düzeneği üzerinden alınan zamana bağlı ölçümler görselleştirilirken, çok eksenli grafikler kullanılmıştır. Şekil 4.5, Şekil 4.6, Şekil 4.12 ve Şekil 4.13’de verilen *zamana bağlı ölçüm verileri* grafikleri şu şekilde yorumlanmalıdır:

- Birinci grafikteki sol-y eksenini, nesnenin görüntü içerisindeki piksel cinsinden konumunu temsil etmektedir. Görüntü merkezi sınırları  $\Delta \bar{c} = \pm 10$  piksel olarak verildiğinden,  $y = 10$  ve  $y = -10$  yatay doğruları sırası ile görüntü merkezinin üst ve alt sınırlarını temsil etmektedir.  $y = 0$  doğrusu ise tam görüntü merkezini temsil etmektedir.

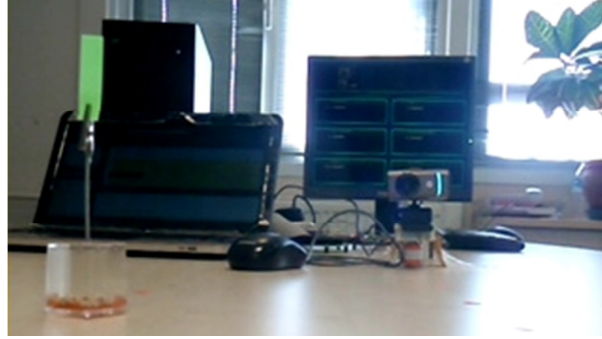
- İkinci grafikteki sol-y eksenini ise kameranın açısal konumunu derece cinsinden temsil etmektedir. Sistem “konum denetimi” modunda iken, ayrıca referans konum eğrisi de yer almaktadır. Bu eğri, konum denetimi sisteminin o andaki referans girişini temsil eder.
- Birinci ve ikinci grafiklerdeki sağ-y eksenini ise sistemin anlık denetim modunu temsil etmektedir. Alabileceği iki değer vardır: “Hız denetimi” ve “Konum denetimi”.

#### 4.1.1. Hareketsiz Nesne Testleri

Bu deney ile sistemin tepki süresi test edilmek istenmiştir. Deney başlatılmadan önce, kameranın konumu  $0^\circ$ 'ye ayarlanır ve nesne C noktasına bırakılır (Şekil 4.3). Makine başlatıldıktan sonra, kamera otomatik olarak nesneye yönelecektir (Şekil 4.4).



Şekil 4.3. Test sehvası ve hareketsiz nesnenin konumu



Deney Bařlamadan  
Önce



Deney  
Bitimi

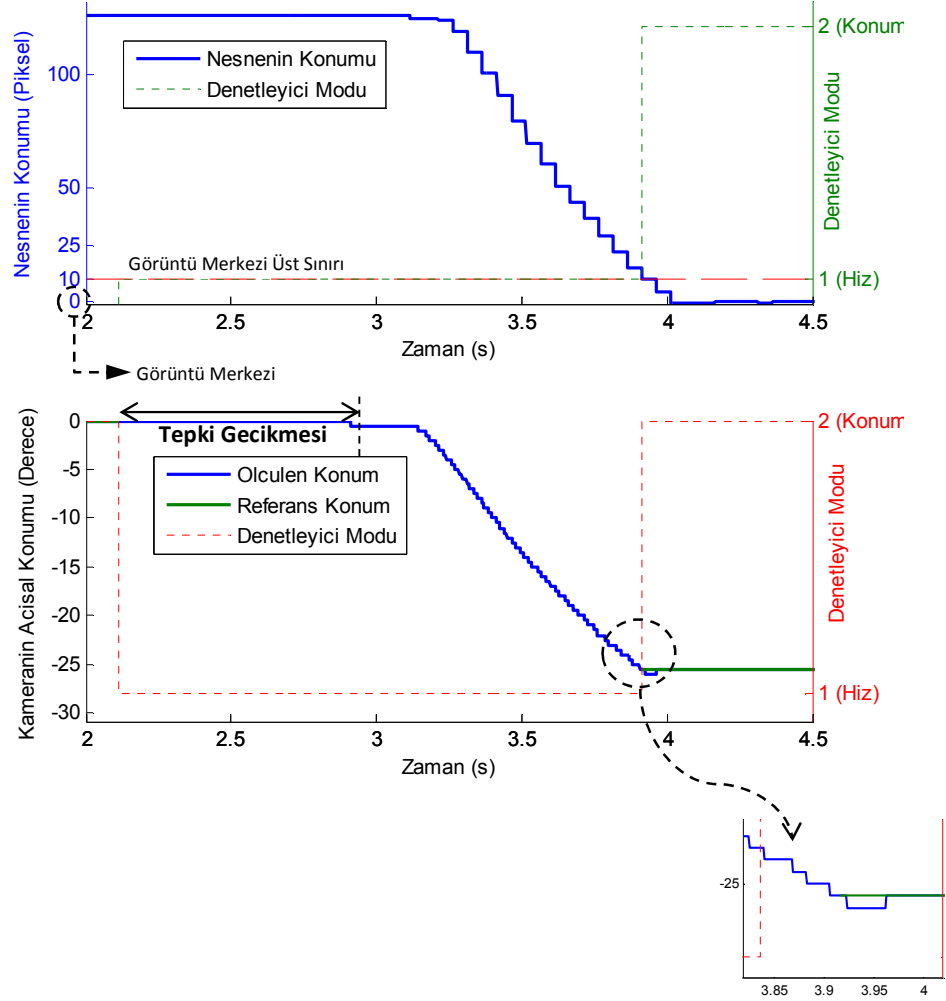
**řekil 4.4.** Test süresince deney düzeneęinin görünümü

Bu deney, Çizelge 3.10'da (sayfa 88) verilen "PI 1x" ve "PI 4x" hız denetimi yapılanması ile birer kez tekrarlanmıştır. Böylece, göreceli olarak daha yavaş (PI 1x) ve daha hızlı bir hız denetiminin (PI 4x), genel sistem performansı üzerindeki etkisi ortaya konmuştur. Bu iki deneyin sonuçları ise sayfa 110'da karşılaştırmalı olarak tartışılmıştır.



#### 4.1.1.1. “PI-1x” Hız Denetimi ile Hareketsiz Nesne Takibi Testi

Bu deneyde, hız denetleyicisi “PI-1x” olarak ayarlanmıştır ve deneyler bir önceki başlıkta tarif edildiği gibi gerçekleştirilmiştir. Deney bitiminde, sistem üzerinden toplanan veriler Şekil 4.5’de görselleştirilmiştir.



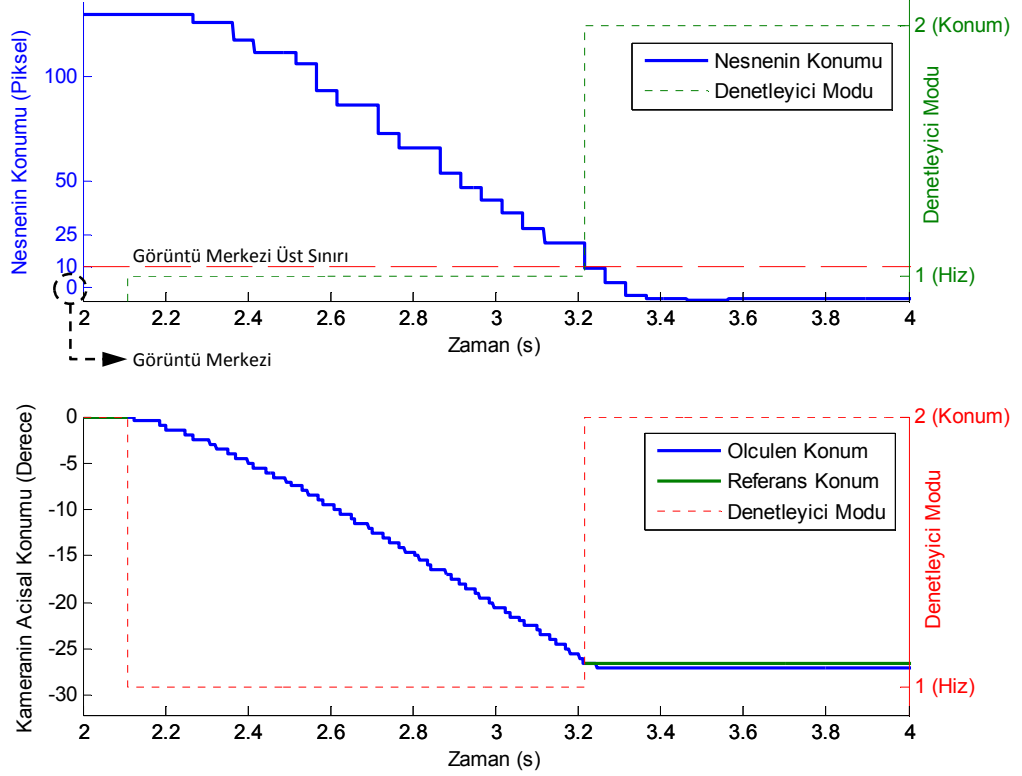
Şekil 4.5. “PI-1x” hız denetimi ile hareketsiz nesne takibi deneyi ve deneyin zamana bağlı ölçüm verileri grafiği

Şekil 4.5 incelendiğinde aşağıdaki sonuçlara varılabilir ve yorumlar getirilebilir:

- Zaman eksenini 2. saniyeden itibaren gösterilmiştir. Bunun nedeni, Çizelge 3.12’de (sayfa 99) verildiği üzere, sistem ilk başlatılırken 2 saniye boyunca bekleme durumunda kalmaktadır. Beklenildiği üzere sistem 2. saniyeden itibaren etkin olacaktır.
- Makine bekleme süresini tamamladıktan 0.065 saniye sonra nesneyi algılamıştır ve o zaman adımında ( $t = 2.065 s$ ) hız moduna geçmiştir. Nesne yatay görüntü düzleminde pozitif tarafta kaldığı için, sistem hız referansını  $\omega_{ref} = -1 rad/s$  olarak ayarlar. Kamera, nesnenin merkezi görüntü merkezi sınırları içerisinde olana kadar dönme hareketini gerçekleştirir.
- Nesnenin görüntü içerisindeki merkezi, görüntü merkezi sınırına ulaştığı anda ( $t = 3.916 s$ ), makine “konum denetimi” moduna geçmiştir. Kameranın o andaki konumu ( $\theta_{ref} = -25.5^\circ$ ), aynı zamanda konum denetiminin referans konumu olarak ayarlanmıştır ve kamera bu konumda sabit kalmıştır. Şekil 4.5’in büyütülmüş kısmında bu durum görülmektedir.
- Nesne 2.065’inci saniyede algılanmasına rağmen, kameranın harekete geçmesi yaklaşık 1 saniye kadar gecikmiştir. Bunun nedeni, “PI-1x” yapılanmasının başlangıç torkunu hemen telafi edemeyişidir.
- Sonuçta ise nesnenin algılanması ile nesne merkezinin görüntü merkezi bölgesine ulaşması ( $3.916 - 2.065$ )  $s = 1.851 s$  süre almıştır.
- $t = 3.916 s$ ’den itibaren nesnenin görüntü içerisindeki merkezi, görüntü merkezi bölgesindedir. Durgun nesnenin takip edilmesi başarılıdır. Fakat takip, bir sonraki başlıkta anlatılan deneye göre daha yavaştır.

#### 4.1.1.2. “PI-4x” Hız Denetimi ile Hareketsiz Nesne Takibi Testi

Bu deneyde, hız denetleyici “PI-1x” olarak ayarlanmıştır ve deneyler bir önceki başlıkta tarif edildiği gibi gerçekleştirilmiştir. Deney bitiminde, sistem üzerinden toplanan veriler Şekil 4.6’da görselleştirilmiştir.



Şekil 4.6. “PI-4x” hız denetimi ile hareketsiz nesne takibi deneyi ve deneyin zamana bağlı ölçüm verileri grafiği

Şekil 4.6 incelendiğinde “PI-1x” Hız Denetimi ile Hareketsiz Nesne Takibi Deneyi”nden farklı olarak aşağıdaki sonuçlara varılabilir:

- Nesne, makine etkin duruma geçtikten 0.111 saniye sonra algılanmıştır ( $t = 2.111$  s) ve kamera tepki gecikmesi olmaksızın harekete başlayabilmiştir. Bunun nedeni, “PI-4x” yapılanmasının sürtünme torkunu telafi etmesidir. Böylece tepki gecikmeleri ortadan kaldırılmıştır.

- Nesnenin algılanması ile nesne merkezinin görüntü merkezi bölgesine ulaşması (3.217 – 2.111) s = 1.106 s süre almıştır. PI-1x yapılanmasına göre daha hızlı bir sonuçtur. Böylece, sistemin tepki hızı artmıştır.
- $t = 3.217$  s'den itibaren nesnenin görüntü içerisindeki merkezi, görüntü merkezi bölgesindedir. Durgun nesnenin takip edilmesi başarılıdır. Takip, bir önce başlıkta anlatılan deneye göre daha hızlıdır.

#### 4.1.1.3. Hareketsiz Nesne Takibi Deneyi Sonuçları

Her iki deneyde de, nesnenin takibi başarılı ile gerçekleştirilmiştir. Fakat, göreceli olarak daha yüksek kazançta sahip PI-4x hız denetimi kullanımı ile aşağıdaki iyileşmeler gözlemlenmiştir:

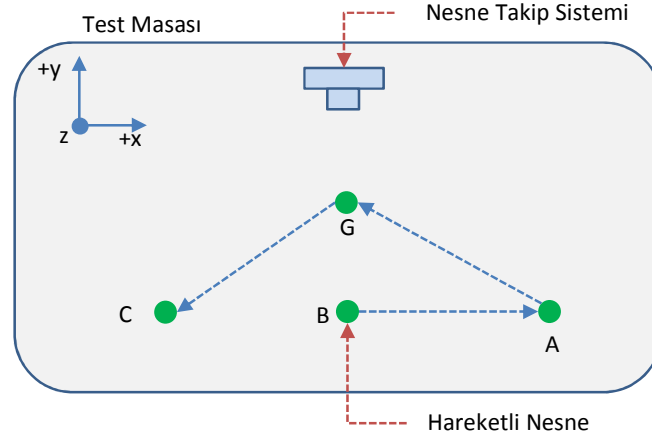
- PI-1x yapılanması ile gerçekleştirilen deneylerde, kameranın harekete geçene kadar kaybettiği süreler (teпки gecikmeleri) ortadan kaldırılmıştır,
- PI-4x ile sistemin genel tepki hızı artmıştır, kararlılığından ödün vermemiştir. Bu tepki sürelerinin karşılaştırması **Çizelge 3.13**'de verilmiştir.

**Çizelge 3.13.** Farklı denetleyici yapılanmasının genel performans üzerine etkisi

Hız Denetleyicisi	Nesnenin Görüntü Merkez Bölgesine Ulaşma Süresi
PI-1x	1.851 s
PI-4x	1.106 s

#### 4.1.2. Hareketli Nesne Testleri

Bu deney ile sistemin, hareketli nesnelerin takip performansı test edilmek istenmiştir. Nesne, Şekil 4.7’de görüldüğü gibi, sırası ile B, A, G ve C noktalarına doğru sabit (ve keyfi) hızla hareket ettirilmiş ve her bir noktada bir süre (1 yada 2 s) bekletilmiştir.

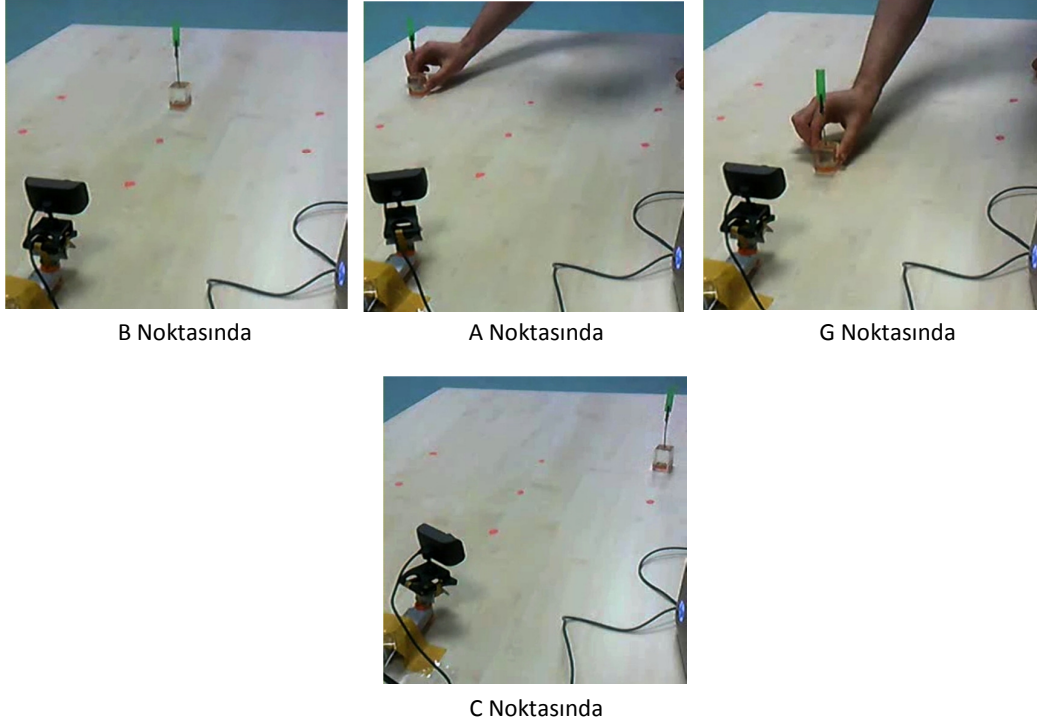


Şekil 4.7. Test sehvası ve nesnenin hareket yörüngesi

Bu deney, Çizelge 3.10’da (sayfa 88) verilen “PI 1x” ve “PI 4x” hız denetimi yapılanması ile birer kez tekrarlanmıştır. Böylece, göreceli olarak daha yavaş (PI 1x) ve daha hızlı bir hız denetiminin (PI 4x), genel sistem performansı üzerindeki etkisi ortaya konmuştur.

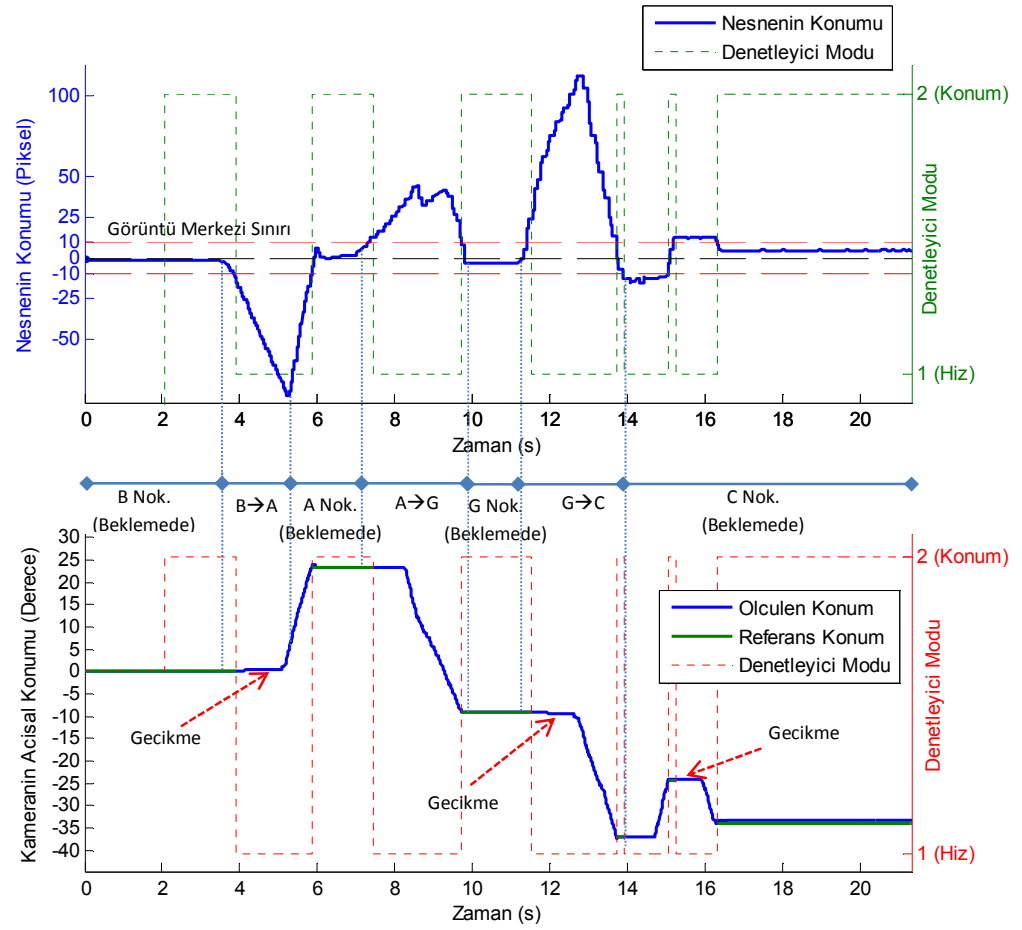
##### 4.1.2.1. “PI-1x” Hız Denetimi ile Hareketli Nesne Takibi Deneyi

Bu deneyde, hız denetleyici “PI-1x” olarak ayarlanmıştır ve deneyler Şekil 4.7’de tarif edildiği gibi gerçekleştirilmiştir. Deneyin aşamalarını özetleyen fotoğraflardan da (Şekil 4.8) anlaşılacağı üzere, sistem hareketli nesnenin takibini başarabilmiştir.



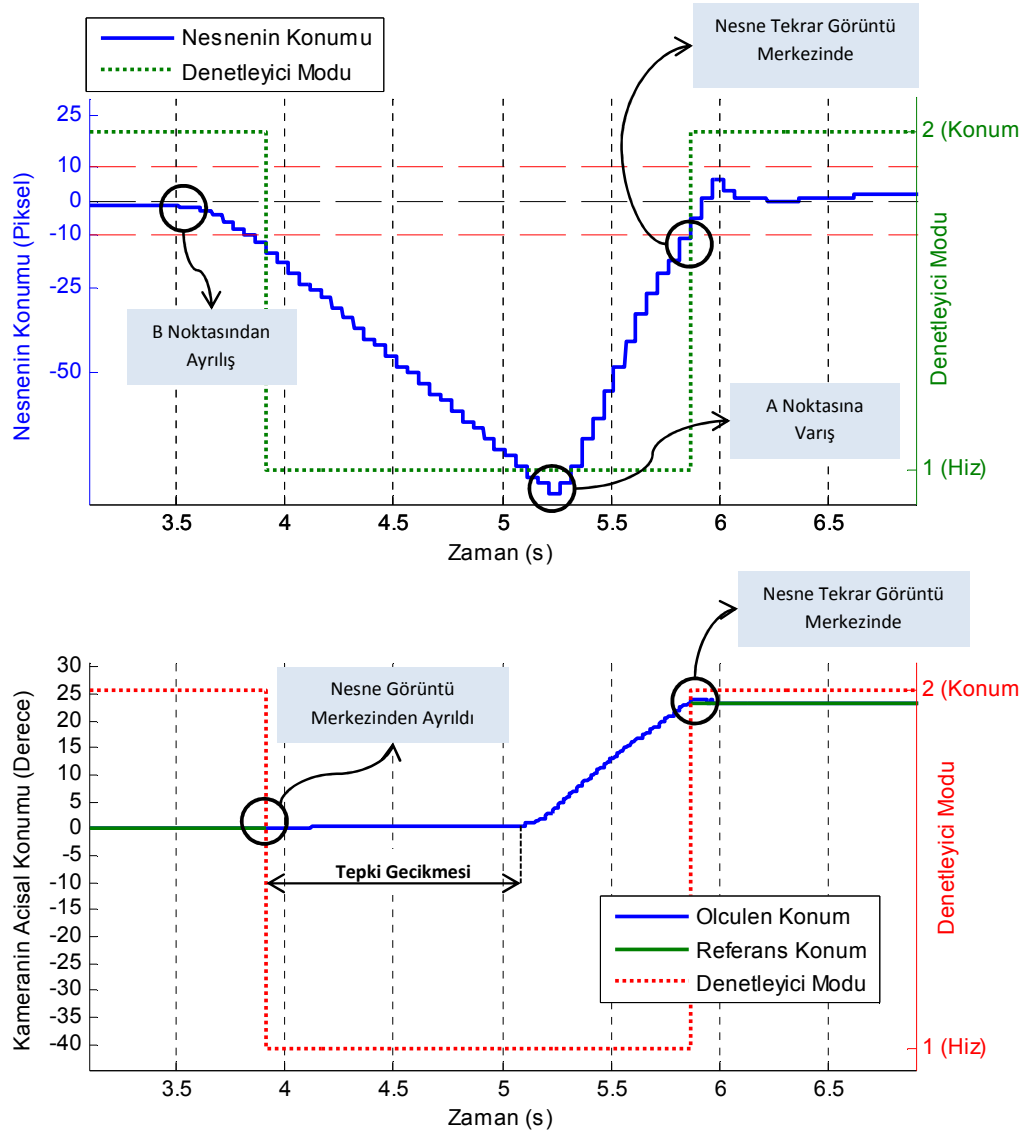
**Şekil 4.8.** “PI-1x” Hız Denetimi ile Hareketli Nesne Takibi Deneyi

Deney sonrası elde edilen zaman bağı ölçüm verileri grafiği Şekil 4.9’da verilmiştir.



**Şekil 4.9.** "PI-1x" hız denetimi ile hareketli nesne takibi deneyi ve deneyin zamana bağlı ölçüm verileri grafiği

"PI-1x" Hız Denetimi ile Hareketli Nesne Takibi Deneyi'nde, sistemin genel performansını değerlendirmek gerekirse, Şekil 4.9'da verilen grafiğin belirli bir bölgesinin incelenmesi yeterli olacaktır. Nesnenin B noktasından ayrılıp A noktasında bekleme yaptığı zamana kadar olan kısmı Şekil 4.10'da mercek altına alınmıştır.



**Şekil 4.10.** Nesne B ve A noktasında iken yakınlaştırılmış grafik

Yakınlaştırılmış grafik (Şekil 4.10) incelenirse aşağıdaki yorumlar yapılabilir:

Nesne  $t = 3.51s$ 'de B noktasından ayrılmış ve  $t = 5.27s$ 'de A noktasına varmıştır. Bu sırada nesne görüntü merkezi sınırından ayrıldığı anda sistem "hız denetimi" moduna geçmiştir ve  $\omega_{ref} = +1 rad/s$  ile nesnenin merkezi tekrar görüntü merkezi sınırında olana kadar açısız dönme hareketi gerçekleştirmiştir ve nesneyi tekrar görüntü merkezine almıştır. Nesne takip sistemine uygulanan hareket denetimi stratejisine göre, nesne görüntü merkezi sınırlarına girdiği andaki ( $t = 5.27s$ ) konumda sabit kalması



istenmişti. Kameranın o andaki konumu ( $\theta_{ref} = -23.5^\circ$ ), aynı zamanda konum denetiminin referans konumu olarak ayarlanmıştır ve kamera bu konumda sıfır kalıcı durum hatası ile sabit kalmıştır.

Şekil 4.10'dan görüldüğü üzere, nesnenin görüntü merkezinden ayrılması ile kameranın harekete geçmesi arasında yaklaşık 1 saniyelik tepki gecikmesi vardır. Kamera neredeyse, nesne A noktasına vardıktan sonra harekete geçebilmiştir. Şekil 3.65'deki (sayfa 89) benzetim çalışmalarına göre bu beklenen bir durumdur. “PI-1x” hız denetleyicisi başlangıçtaki sürtünme torkunu hemen telafi edememektedir ve bu da tepki gecikmelerine neden olmaktadır. Ayrıca bu tepki gecikmeleri, sistem her hız denetimi moduna geçtiğinde gözlemlenmektedir ve Şekil 3.49'da açıkça gösterilmiştir.

Benzer sonuçlar, nesnenin

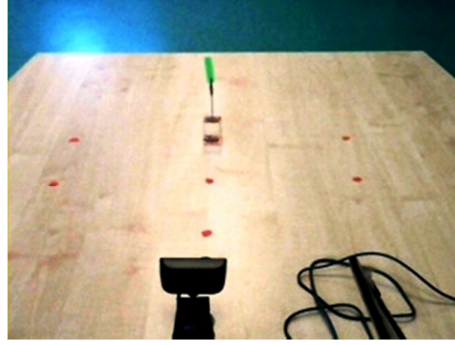
- A noktasından ayrılıp G noktasına vardığı ve beklediği,
- G noktasından ayrılıp C noktasına vardığı ve beklediği,

zaman dilimlerinde de gözlemlenebilir.

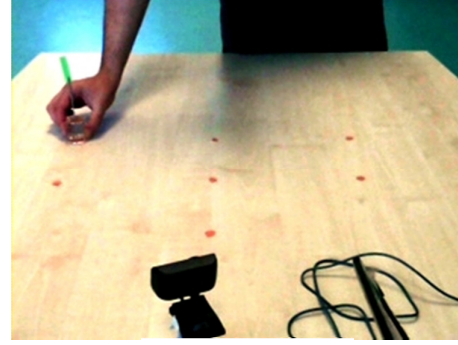
Sonuç olarak “PI-1x” hız denetimli sistem, hareketli nesneyi tekrar görüntü merkezi sınırına, fazladan gecikmeli olarak almayı başarmıştır.

#### 4.1.2.2. “PI-4x” Hız Denetimi ile Hareketli Nesne Takibi Deneyi

PI-1x hız denetimi yapılanması ile meydana gelen gecikmeli takipler, daha yüksek kazançlı ve daha hızlı “PI-4x” hız denetleyici yapılanması ile ortadan kaldırılmak istenmiştir. Bu deneyde, hız denetleyici “PI-4x” olarak ayarlanmıştır ve deneyler Şekil 4.7'de tarif edildiği gibi gerçekleştirilmiştir. Deneyin aşamalarını özetleyen fotoğraflardan da (Şekil 4.11) anlaşılacağı üzere, sistem hareketli nesnenin takibini başarabilmiştir.



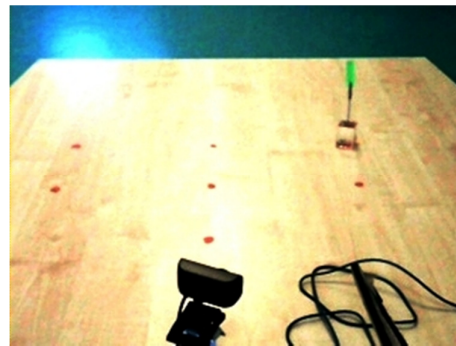
B Noktasında



A Noktasında



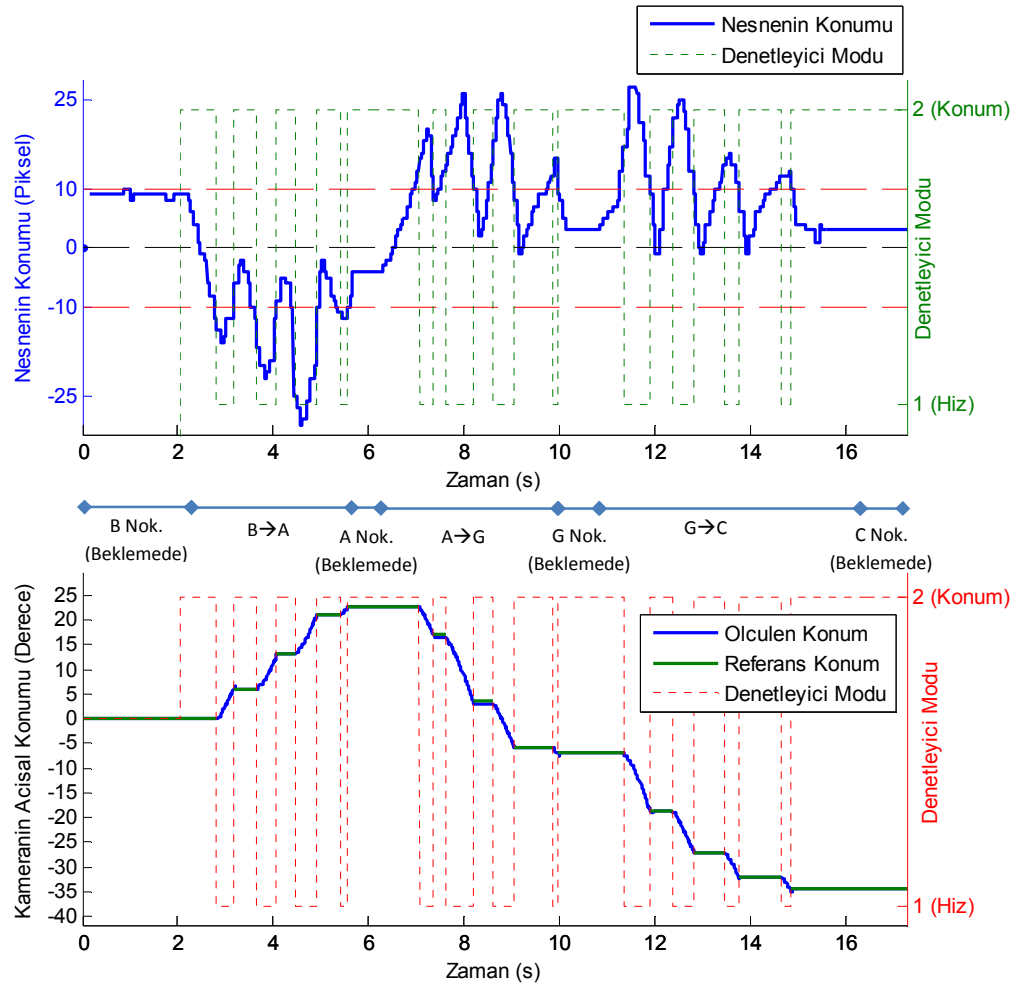
G Noktasında



C Noktasında

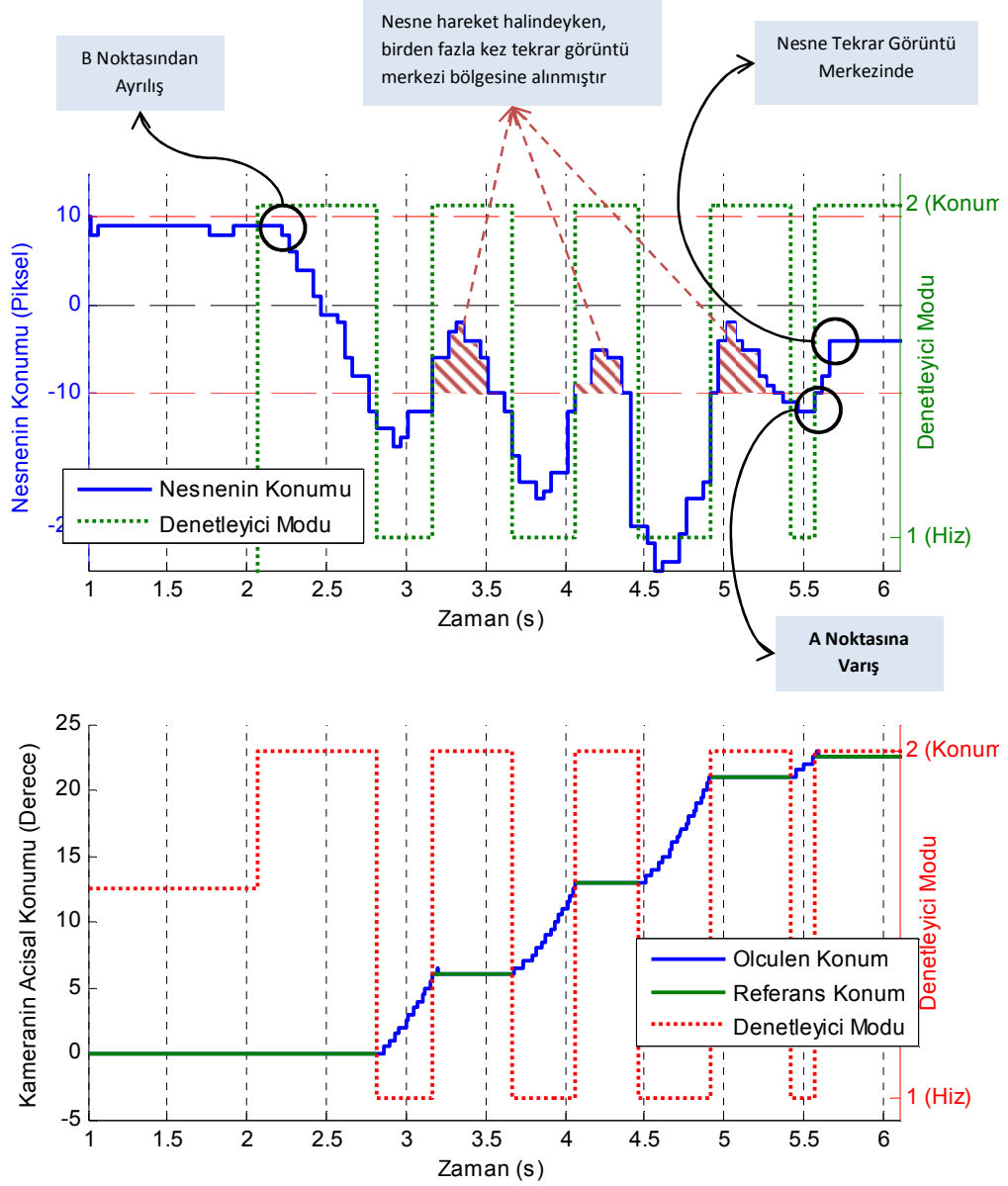
**Şekil 4.11.** “PI-4x” Hız Denetimi ile Hareketli Nesne Takibi Deneyi

Deney sonrası elde edilen zaman bağı ölçüm verileri grafiği Şekil 4.12’de verilmiştir.



**Şekil 4.12.** “PI-4x” hız denetimi ile hareketli nesne takibi deneyi ve deneyin zamana bağlı ölçüm verileri grafiği

“PI-4x Hız Denetimi ile Hareketli Nesne Takibi Deneyi”nde, sistemin genel performansını değerlendirmek gerekirse, Şekil 4.12’de verilen grafiğin belirli bir bölgesinin incelenmesi yeterli olacaktır. Nesnenin B noktasından ayrılıp A noktasında bekleme yaptığı zamana kadar olan kısmı Şekil 4.13’de mercek altına alınmıştır.



**Şekil 4.13.** Nesne B ve A noktasında iken yakınlaştırılmış grafik

Yakınlaştırılmış grafik (Şekil 4.13) incelenirse aşağıdaki yorumlar yapılabilir:

Nesne  $t = 2.21 \text{ s}$ 'de B noktasından ayrılmış ve  $t = 5.56 \text{ s}$ 'de A noktasına varmıştır. Bu sırada nesne görüntü merkezi sınırından ayrıldığı anda sistem “hız denetimi” moduna geçmiştir ve  $\omega_{ref} = +1 \text{ rad/s}$  ile nesnenin merkezi tekrar görüntü merkezi

sınırında olana kadar açısal dönme hareket gerçekleştirmiştir ve nesneyi tekrarlı olarak görüntü merkezine almıştır.

Bir önceki deneyden farklı olarak,

- sistemdeki doğrusalsızlıklardan kaynaklanan başlangıç gecikmeleri tamamen ortadan kalkmıştır (Şekil 3.65’deki (sayfa 89) benzetim çalışmalarında, “PI-4x” denetleyici yapılanmasının bu tepki gecikmelerini ortadan kaldırdığı görülmüştü),
- kamera nesne A noktasına vardıkdan sonra değil, B noktasından ayrılır ayrılmaz harekete geçmiştir ve nesneyi takibe başlamıştır. Bu çaba, Şekil 4.13’de açıkca gösterilmiştir.

Nesne takip sistemine uygulanan hareket denetimi stratejisine göre, nesne görüntü merkezi sınırlarına girdiği andaki konumda sabit kalması istenmişti. Şekil 4.13’de görüldüğü üzere, nesne hareket halindeyken birden fazla kez nesne görüntü merkezine alınmıştır ve kameranın o anlardaki açısal konumu referans konum olarak ayarlanmış ve bu referans değer sıfır kalıcı durum hatası ile takip edilmiştir.

Benzer sonuçlar, nesnenin

- A noktasından ayrılıp G noktasına vardığı ve beklediği,
- G noktasından ayrılıp C noktasına vardığı ve beklediği,

zaman dilimlerinde de gözlemlenebilir.

Sonuç olarak “PI-4x” hız denetimli sistem, hareketli nesneyi tekrar görüntü merkezi sınırına, tepki gecikmeleri olmaksızın almayı başarmıştır.

#### 4.2. Model-Tabanlı Tasarımın Mekatronik Tasarım Süreçlerine Katkısı

Tezin konusu olan mekatronik sistemin tasarımları model-tabanlı tasarım yaklaşımı ile gerçekleştirilmiştir. Bu başlıkta ise, model-tabanlı tasarımın geleneksel tasarım süreçlerine katkıları tartışılmıştır.

Yapılan anket araştırmalarında (Anonim 2008), mekatronik ürün geliştiren kurumların karşılaştığı önemli zorluklar genel hatları ile iki ana yetersizliğe işaret etmektedir:

- Farklı mühendislik disiplinlerinin, tek bir mekatronik ürün geliştirme sürecine dahil edilmesindeki zorluk,
- erken aşamalarda testlerin gerçekleştirilememesi.

Ayrıca, ileri aşamalarda farkedilen hatalar, erken aşamalarda farkedilenlere göre çok daha fazla maliyet kayıplarına neden olmaktadır. İlk prototipin elde edildiği aşama, hataların telafisi için en yüksek maliyetli aşamadır.

Bütün bunlar göz önünde bulundurularak, bu tez çalışması sonunda, model-tabanlı tasarım aşağıdaki artıları sağladığı sonucuna varılmıştır:

- Gerçek fiziksel prototiplerle çalışmadan önce, benzetim çalışmalarının tasarımları doğruladığı,
- tasarımın erken aşamalarında modellenen sistemin davranışlarının, gereksinimlere uygunluğunun benzetimlerle sınanabildiğinin,
- otomatik kod dönüşümünün, elle kodlamanın yerini aldığı ve kodlama hatalarının ortadan kaldırıldığı,
- hızlı denetim prototipleme ile farklı tasarım düşüncelerinin hızlıca gerçek donanım üzerinde test edilebildiği,

sonucuna varılmıştır ve farklı hedef donanımlara uygulanabilen doğrulanmış bir model elde edilmiştir.

### 4.3. İleriye Dönük Çalışmalar

Bu tez kapsamında geliştirilen mekatronik sistem üzerinde ileride aşağıdaki çalışmalar ilave edilebilir.

- Nesne takip sistemi sadece tek eksenli takip (yatay ekseninde dönme hareketi) yapabilmektedir. Dikey yönde dönme serbestliği ilave edilerek iki eksenli takip gerçekleştirilebilir.
- Sadece belirli bir renge ve alana sahip nesnelere değil; daha gelişmiş video işleme algoritmaları ile araç, insan vs. gibi daha karmaşık nesnelere takibi sağlanabilir.
- Sistem, bir hızlı prototipleme ortamında gerçekleştirmek yerine, bir son ürün işlemcisi (mikrodenetleyici birimleri vs. gibi) üzerinde gerçekleştirilerek tek başına sistem haline getirilebilir.

## 5. SONUÇ

Bu tez çalışması kapsamında, bir mekatronik sistem tasarımının tüm süreçlerini içeren bir örnek uygulama tasarlanmış ve gerçekleştirilmiştir. Bu örnek uygulama “nesne takip sistemi” olarak isimlendirilmiştir. Nesne takip sistemi, yatay düzlemde dönme hareketi yapabilen ve hareketli/durgun bir nesnenin takibini gerçekleştirebilen bir hareketli-kameradır. Belirli bir renge ve alana sahip bir nesneyi görüntü içerisinde algılayabilen, nesnenin merkezi ile görüntü merkezini eşleştirecek şekilde yatay kamera konumunu güncelleyebilen servo sistemdir.

Çalışma süresince aşağıdaki bulgulara ve sonuçlara varılmıştır:

- Nesne algılama ve nesne merkezinin hesaplanması işlemleri için alınan görüntüye sırası ile *eşik değerlendirme*, *bağlı bileşenlerin etiketlenilmesi* ve *blob analizi* yöntemleri uygulanmıştır. Görüntü içerisinde belirli bir renge ve sınırlı bir alana sahip nesnenin algılanması ve bu nesneye ilişkin unsurların (ağırlık merkezi, alan vs.) hesaplanmasında başarılı olduğu sonucuna varılmıştır.
- Denetlenen sistemin (elektromekanik altsistem) sahip olduğu belirsiz parametreler parametre kestirim yöntemleri ile hesaplanmış ve doğrulanmıştır. Elektromekanik altsisteme ilişkin eşdeğer eylemsizlik ( $J_{eq}$ ) ve viskoz sürtünme ( $B_{eq}$ ) sabitleri ve sürtünme torku ( $T_f$ ) parametre kestirimi yöntemleri ile hesaplanmıştır. Benzetim modelleri ve deneysel sonuçlar karşılaştırılarak hesaplanan bu parametrelerin güvenilir olduğu sonucuna varılmıştır.
- Elektromekanik altsistemin modeli  $\Omega(s)/C(s)$  ikinci mertebeden bir transfer fonksiyonu iken,  $L_a \cong 0$  kabulü ile birinci mertebeden bir fonksiyona indirgenmiştir. Sadeleştirilmiş model ile yapılan karşılaştırmalarda,  $L_a$  sıfıra çok yakın olduğu için, sistemin dinamiğine neredeyse hiç bir etkisi olmadığı sonucuna varılmıştır. Bu sadeleştirme, denetim sistemi tasarımını oldukça kolaylaştırmıştır.



- Denetim tasarımı aşamasında fiziksel prototipler kullanmak yerine sistemin benzetim modelleri kullanılmıştır. Bu sayede,
  - o farklı denetleyici algoritmaları hızlıca sınanabilmiş ve denetim sistemi tasarımı hızlıca tamamlanabilmiştir,
  - o uygun olmayan denetim stratejileri ve hatalar erken aşamalarda farkedilmiş ve düzeltilmiştir.
- Tasarlanan farklı denetleyici algoritmaları hızlı prototipleme ile gerçek zamanlı olarak test edilmiş, model ve deney sonuçları karşılaştırılmış ve modellerin tutarlılığı doğrulanmıştır. xPC Target gibi gerçek zamanlı test ortamlarının, kavram aşamasındaki fikirlerin hızlıca test edilmesine olanak sağladığı sonucuna varılmıştır.
- Hareket altsistemi, sürtünme torkunun ( $T_f$ ) dönüş yönüne bağlı olması ve denetleyici çıkışının  $-1 < c(t) < +1$  arasında sınırlı olması nedeniyle doğrusal değildir. Doğrusal olmayan ve doğrusallaştırılmış denetim sistemi modellerine ilişkin sistem yanıtları, göz ardı edilemeyecek kadar birbirinden farklı olduğu sonucu elde edilmiştir.
- Kök-yer eğrileri, frekans yanıtı vs. gibi klasik denetim kuramına ilişkin yöntemler tamamen doğrusal ve zamanla değişmeyen (LTI) sistemlere uygulanabildiğinden, denetim sisteminin tasarımında doğrusal modellerden, analizinde ise doğrusal olmayan modellerden faydalanılmıştır. Denetleyici tasarımında sadece doğrusal modellerin kullanımının yetersiz kaldığı sonucuna varılmıştır.
- Hız denetiminde P algoritmasının kalıcı durum hatasını gideremediği benzetim aşamasında farkedilmiştir. Benzetim çalışmaları PI algoritmasının kalıcı durum hatasını tamamen giderdiğini göstermiştir, fakat düşük kazançlı PI yapılanmasında, sürtünme torkundan kaynaklanan tepki gecikmeleri meydana gelmiştir. PI denetleyicinin kazancı yükseltilerek bu tepki gecikmeleri tamamen giderilebileceği sonucu elde edilmiştir.
- Hız denetiminde deneysel çalışmalar sonucunda, PI denetimli denetim sisteminin, artan sistem kazancı ile birlikte yanıt hızının da arttığını, fakat belirli bir kazançtan sonra sistemin kararsızlaştığı sonucuna varılmıştır.

- Çalışmada kullanılan DC motor bir hız algılayıcısına sahip olmadığı için, motorun hızı konum verilerinden hesaplanılmıştır. Hızın hesaplanması için “frekans ölçümü yöntemi” tercih edilmiştir. Yapılan deneysel çalışmalarda, hız ölçümünde karşılaşılan niceleme (*quantisation*) hatalarının ( $\Delta\omega$ ), gözlem-zamanı dilimine ( $T_{sc}$ ) bağlı olduğu ( $\Delta\omega = f(N_p, T_{sc})$ ) görülmüştür. Farklı  $T_{sc}$  değerleri için hız ölçümü deneylerinin sonuçları karşılaştırıldığında, çalışmada kullanılan motor için en uygun gözlem-zaman diliminin  $T_{sc} = 6T_s$  kadar olması sonucuna varılmıştır.
- Nesne takip sisteminin nesne takibi testlerinde, göreceli olarak düşük kazançlı hız denetimi yapılanmasında (“PI-1x”), sistemde tepki gecikmelerinin meydana geldiği sonucuna varılmıştır. Daha yüksek kazançlı “PI-4x” hız denetleyici ile tepki gecikmeleri olmaksızın takip edilebildiği sonucuna varılmıştır.
- Farklı mühendislik disiplinlerinin (makine, elektronik, kontrol ve bilgi teknolojileri) model-tabanlı tasarım ile mekatronik ürün geliştirme süreçlerine kolayca dahil edilebileceği sonucuna varılmıştır.

Tez çalışması için tasarlanan bu mekatronik sistemin tasarımında, model-tabanlı tasarımın unsurlarından olabildiğince faydalanılmıştır ve model-tabanlı tasarım mekatronik sistem tasarımına olan katkıları ortaya konmuştur.

## KAYNAKLAR

- Anonim, 2010.** The Autonomous Vehicle. [http://www.visiononline.org/vision-resources-details.cfm?content\\_id=2364](http://www.visiononline.org/vision-resources-details.cfm?content_id=2364) – (Erişim tarihi: 20 Şubat 2012)
- Anonim 2006.** Humusoft MF 624 User's Manual. Çek Cumhuriyeti
- Anonim 2007.** Humusoft Data Acquisition Library Reference Manual. Çek Cumhuriyeti
- Anonim 2008.** System Design: New Product Development for Mechatronics. <http://www.aberdeen.com/Aberdeen-Library/4576/RA-product-development-mechatronics.aspx> – (Erişim tarihi: 04 Temmuz 2012)
- Abourida, S. Belanger J. 2009.** Real-Time Platform for the Control Prototyping and Simulation of Power Electronics and Motor Drives. Third International Conference on Modeling, Simulation and Applied Optimization, 20-22 Ocak 2009, Sharjah
- Bishop, R. H. 2007.** The Mechatronics Handbook. USA
- Briz, F. Cancelas, J. A. Diez A. 1994.** Speed Measurement Using Rotary Encoders for High Performance AC Drives. 20th International Conference on Industrial Electronics, Control and Instrumentation, 1994, İspanya
- Chaumette, F., Hutchinson S. 2006.** Visual Servo Control Part 1: Basic Approaches. *IEEE Robotics & Automation Magazine*, 1070: 82-90
- Ding, W. Gong Z., Xie, S., Zou, H. 2006.** Real-time Vision-based Object Tracking from a Moving Platform in the Air. International Conference on Intelligent Robots and Systems, 9-15 Ekim 2006, Beijing Çin Halk Cumhuriyeti
- Gasperi, M., Hurbain, P., Hurbain I. 2007.** Extreme NXT. USA, 295 s.
- Hardin, W. 2008.** Machine Vision Brings Advanced Medical Technology to Life. [http://www.visiononline.org/vision-resources-details.cfm?content\\_id=1131](http://www.visiononline.org/vision-resources-details.cfm?content_id=1131) – (Erişim tarihi: 20 Şubat 2012)
- Kellett, P. 2006.** Machine Vision in the Automotive Industry. [http://www.visiononline.org/vision-resources-details.cfm?content\\_id=1175](http://www.visiononline.org/vision-resources-details.cfm?content_id=1175) – (Erişim tarihi: 20 Şubat 2012)
- Kim, S. Lee, D. Kim, H. J. 2011.** Image Based Visual Servoing for an Autonomous Quadrotor with Adaptive Backstepping Control. International Conference on Control, Automation and Systems. 26-29 Ekim 2011, Kore.

**Lang, H., Wang, Y. Silva, C. W. 2010.** Vision Based Object Identification and Tracking for Mobile Robot Visual Servo Control. 8th International Conference on Control and Automation, 9-11 Haziran 2010, Xiamen Çin Halk Cumhuriyeti

**Loh, R. N. K., Pornthanomwong, T., Pyko J. S., Lee, A., Karsiti M. N. 2007.** Modeling, Parameters Identification, and Control of an Electronic Throttle Control (ETC) System. International Conference on Intelligent and Advanced Systems. 25-28 Kasım 2007, Kuala Lumpur Malezya

**Mosterman, P. J. 2005.** Embedded Real-Time Control via MATLAB, Simulink, and xPC Target: Networked and Embedded Control Systems, Hristu-Varsakelis, D., Levine, W. S.. Boston, 419-446.

**Oherek, M., Robl, C. 2001.** Model-Based Design of an ECU with Data- and Event-Driven Parts Using Auto Code Generation. International Conference on Robotics & Automation, 21-26 Mayıs 2001, Seoul Kore.

**Oherek, M., Robl, C. 2002.** Model Based Design in the Development of Thermodynamic Systems and their Electronic Control Units. IEEE International Conference on Control Applications, 18-20 Eylül 2001, Glasgow

**Petrella, R. Tursini, M. Peretti, L. Zigliotto, M. 2007.** Speed Measurement Algorithms for Low-Resolution Incremental Encoder Equipped Drives: A Comparative Analysis. International Aegean Conference on Electrical Machines and Power Electronics, 2006

**Shapiro, L. 2000.** Computer Vision, USA, 617 s.

**Yamamoto, Y. 2009.** NXTway-GS Model-Based Design. Applied Systems First Division, Cybernet Systems Co., Japonya

**Yüksel, İ. 2011.** Otomatik Kontrol. Nobel Yayın Dağıtım, Ankara

## ÖZGEÇMİŞ

Adı Soyadı : Ahmet Demirkesen  
Doğum Yeri ve Tarihi : İskenderun – 09.11.1983  
Yabancı Dili : İngilizce

Eğitim Durumu

Lise : Süleyman Çelebi Lisesi (Y.D.A.) - 1998-2002  
Lisans : Sakarya Üniversitesi Mühendislik Fakültesi  
Makine Mühendisliği Bölümü – 2003-2008  
Yüksek Lisans : Uludağ Üniversitesi Fen Bilimleri Enstitüsü  
Makine Mühendisliği Anabilim Dalı – 2009-2012

İletişim (e-posta) : ahmetdksn@live.com