

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ASENKRON MOTORLARIN DENETİMİ İÇİN İNTERNET
TABANLI SANAL VE UZAKTAN ERİŞİMLİ BİR LABORATUAR
GELİŞTİRME**

Ahmet TEKİN

Tez Yöneticisi
Yrd. Doç. Dr. Fikret ATA
Prof. Dr. Muammer GÖKBULUT

Doktora Tezi
Elektrik-Elektronik Mühendisliği Anabilim Dalı

ELAZIĞ, 2009

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ASENKRON MOTORLARIN DENETİMİ İÇİN İNTERNET
TABANLI SANAL VE UZAKTAN ERİŞİMLİ BİR LABORATUAR
GELİŞTİRME**

Ahmet TEKİN

Doktora Tezi

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Bu tez, tarihinde aşağıda belirtilen jüri tarafından oybirliği/oyçokluğu ile başarılı/başarısız olarak değerlendirilmiştir.

Danışman: Yrd. Doç. Dr. Fikret ATA

Üye: Prof. Dr. Ömer Faruk BAY

Üye: Prof. Dr. Mustafa POYRAZ

Üye: Prof. Dr. Z. Hakan AKPOLAT

Üye: Yrd. Doç. Dr. Ahmet ÇINAR

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu' nun/...../..... tarih ve sayılı kararıyla onaylanmıştır.

TEŐEKKÖR

Bu tez alıőması süresince yardımlarını esirgemeyen ve deęerli fikirleriyle bana yol gösteren danışman hocalarım, Sayın Prof. Dr. Muammer GÖKBULUT ve Sayın Yrd. Do. Dr. Fikret ATA'ya ok teőekkür eder, őükranlarımı sunarım.

Her konuda desteklerini esirgemeyen, bilgi ve tecrübelerinden yararlandığım Öğr. Gör. Cafer BAL, Öğr. Gör. Erkan TANYILDIZI, Uzm. Musa ÇIBUK ve bölümümüz öğretim üye ve yardımcılarına teőekkürü bor bilirim.

Ayrıca tez alıőması boyunca gösterdikleri sabır, anlayış ve desteklerinden, sağladıkları huzurlu alıőma ortamından dolayı sevgili eşime ve moral kaynađım kızım Elif'e őükranlarımı sunarım.

İÇİNDEKİLER

İÇİNDEKİLER	I
ŞEKİLLER LİSTESİ.....	III
TABLolar LİSTESİ.....	V
EKLER LİSTESİ	VI
SİMGELER LİSTESİ.....	VII
KISALTMALAR LİSTESİ	IX
ÖZET	X
ABSTRACT	XII
1. GİRİŞ	01
1.1. Literatür Taraması ve Değerlendirilmesi	02
1.2. Tezin Amacı.....	02
1.3. Tezin Bölümleri	13
2. İNTERNET TABANLI LABORATUAR ORTAMLARI.....	14
2.1. Sanal Laboratuarlar ve Uzaktan Erişimli Laboratuarlar	14
2.2. Sanal ve Uzaktan Erişimli Laboratuarların Üstünlükleri ve Sakıncaları	16
2.3. Sanal veya Uzaktan Erişimli Laboratuar Yapıları.....	19
2.3.1. Paylaşım Temelli Programlar Kullanan Laboratuar Yapıları.....	19
2.3.2. İstemci - Sunucu İlişkisi Olan Laboratuar Yapıları	21
2.3.3. Yerel Kullanıcılara Hitap Eden Laboratuar Yapıları	24
3.DSP DENETİMLİ ASENKRON MOTORLAR İÇİN UZAKTAN ERİŞİMLİ LABORATUAR	25
3.1. Donanım Yapısı	26
3.1.1 Asenkron Motorların Modellenmesi	27
3.1.2 Asenkron Motorların Vektör Denetiminde Alan Yönlendirme Yöntemleri	29
3.1.3 Asenkron Motorların Vektör Denetimi	33
3.1.4 Asenkron Motorların Hız Denetimi	35
3.1.5 Asenkron motorun SBD ile Hız Denetimi	37
3.2. Yazılım Yapısı	44
3.2.1. HTML ve ASP	44
3.2.2. Matlab Web Sunucu.....	45
3.2.3. ControlDesk ve Python	54
3.2.4. Nesne Tabanlı Programlama ve Delphi	59

4. İNTERNET TABANLI LABORATUARIN GERÇEKLEŞTİRİLMESİ.....	60
4.1. Sanal Ortamın Gerçekleştirilmesi	62
4.1.1 Sanal Laboratuvarı Gerçekleştirmede Kullanılan Yapı	63
4.1.2. İnternet Üzerinden Simulink Dosyalarını Çalıştırmak.....	65
4.1.3. Denetim Algoritmasının Değiştirilmesi	67
4.1.4. Benzetim Bağlantılarının Kurulması.....	69
4.2. Uzaktan Erişimli Ortamın Gerçekleştirilmesi	72
4.2.1 Laboratuvarı Gerçekleştirme Aşamaları	72
4.2.2 Uygulama Dosyaları ve ControlDesk Layout Arayüzü	73
4.2.3 İstemci-Sunucu İletişimi İçin Geliştirilen Yazılım	75
4.2.4 ControlDesk Otomasyonu İçin Geliştirilen Python Programları	77
4.2.5 Denetim Algoritmasının Değiştirilmesi	80
4.2.6 Sunucu Arayüzü	81
5 KULLANICI ARAYÜZÜ VE SONUÇLAR.....	84
5.1. Benzetim Sonuçları ve Kullanıcı Arayüzü.....	84
5.2. Deneysel Sonuçlar ve Kullanıcı Arayüzü	89
6 SONUÇLAR	98
KAYNAKLAR	100
ÖZGEÇMİŞ.....	112
EKLER	113

ŞEKİLLER LİSTESİ

Şekil 2.1	Microsoft NetMeeting penceresinin görünümü	20
Şekil 2.2	Uzaktan erişimli laboratuvarın yapısı [46]	21
Şekil 2.3	AIM-LAB sistem konfigürasyonu [39]	22
Şekil 2.4	Uzaktan izleme ve denetim uygulamalarının blok diyagramı [40].....	22
Şekil 2.5	Uzaktan erişimin blok şeması [58]	23
Şekil 2.6	Uzaktan erişimli laboratuvarın donanım konfigürasyonu [61]	24
Şekil 2.7	Neuro-Lab ortamında üç katmanlı yapay sinir ağı yapısı [69]	25
Şekil 2.8	Benzetim ortamında B-H eğrisi [74]	25
Şekil 3.1	İnternet tabanlı deney düzeneğinin genel görünümü	27
Şekil 3.2	Üç fazlı motorun eşdeğer iki faz vektörleri	28
Şekil 3.3	DA ve AA motorların vektör diyagramları.....	30
Şekil 3.4	Alan yönlendirme vektör diyagramları	31
Şekil 3.5	Asenkron motorun dolaylı alan yönlendirmeli denetimi için blok yapısı.....	34
Şekil 3.6	Vektör denetimli asenkron motor için hız denetim bloğu	36
Şekil 3.7	Dolaylı vektör denetimli asenkron motorun SBD ile hız denetimi için blok yapısı	38
Şekil 3.8	İki girişli tek çıkışlı Sugeno tip SBD yapısı	39
Şekil 3.9	MWS yapılandırması	45
Şekil 3.10	Karmaşık ağ yapılarında MWS yapılandırması	46
Şekil 3.11	MWS modeli.....	48
Şekil 3.12	Bilgisayar yönetimi penceresi.....	52
Şekil 3.13	Klasör özellikleri penceresi	52
Şekil 3.14	Uygulama hata penceresi	53
Şekil 3.15	Kod üretme	55
Şekil 3.16	idfconverter ve cdaautomationlib modül yapısı	56
Şekil 4.1	İnternet tabanlı laboratuvarın ana sayfası	62
Şekil 4.2	İnternet tabanlı sanal laboratuvarın yapısı	64
Şekil 4.3	matweb.conf isimli dosyanın içeriği.....	65
Şekil 4.4	İnternet üzerinden benzetimi gerçekleştirilen asenkron motorun PI kontrol yapısının simulink benzetimi	65
Şekil 4.5	İnternet tabanlı sanal laboratuvarın çalışma akış şeması	66
Şekil 4.6	Upload-Download işlemleri akış şeması	68
Şekil 4.7	Hız denetleyici Simulink bloğu	69

Şekil 4.8. Benzetim bağlantılarının yapılabildiği web sayfası görünümü	70
Şekil 4.9 Bağlantı kural tablosu	70
Şekil 4.10 Bağlantı durumu ile ilgili hata mesajı	71
Şekil 4.11 Asenkron motorun hız kontrolü için benzetim düzeneği	71
Şekil 4.12 Asenkron motorun hız denetimi deneyi için Matlab/Simulink dosyası	74
Şekil 4.13 ControlDesk ile hazırlanan Layout arayüzü	75
Şekil 4.14 İstemci ile sunucu arasında oluşan olayları gösteren akış şeması	77
Şekil 4.15 Delphi-Python-ControlDesk arasındaki ilişkiyi gösteren akış şeması	79
Şekil 4.16 Denetim algoritmasının değiştirilmesi	80
Şekil 4.17 Sunucu aktiviteleri arayüzü	81
Şekil 4.18 Sunucu – Sonuç Grafikler arayüzü	82
Şekil 4.19 Sunucu – Program Seçenekleri arayüzü	83
Şekil 5.1 Sanal laboratuvarın parametre değiştirme sayfası	84
Şekil 5.2 Sanal laboratuvarın parametre değiştirme sayfası	85
Şekil 5.3 PI denetleyici ile referans hız ve motor hızının değişimi	86
Şekil 5.4 PI denetleyici ile d ve q eksenli gerçek ve referans akımları	86
Şekil 5.5 SBD için sanal laboratuvarın parametre değiştirme sayfası	87
Şekil 5.6 SBD denetleyici ile referans hız ve motor hızının değişimi	87
Şekil 5.7 SBD denetleyici ile d ve q eksenli gerçek ve referans akımları	88
Şekil 5.8 Download (a) ve Upload (b) sayfalarının kesit görünümü	89
Şekil 5.9 Online Deney web sayfası	89
Şekil 5.10 İstemci aktiviteleri arayüzü	90
Şekil 5.11 İstemci – Motor Kontrol Paneli arayüzü	91
Şekil 5.12 Denetleyicilerin parametre değiştirme panelleri	92
Şekil 5.13 İstemci –Grafikler arayüzü	93
Şekil 5.14 IP kamera görüntüleri	94
Şekil 5.15 Motor yüksüz durumda iken PI denetleyici ile 1000 d/d ve 2000 d/d’lık basamak hızı	95
Şekil 5.16 Motor yüksüz durumda iken SBD denetleyici ile 1000 d/d ve 2000 d/d’lık basamak hızı izleme başarımı	97

TABLÖLAR LİSTESİ

Tablo 3.1 Matlabserver temel seçenekleri.....	48
Tablo 3.2 Matlabserver temel seçenekleri.....	50
Tablo 3.3 MWS’da kullanılan fonksiyonlar	51
Tablo 3.4 Cdautomationlib modülünün sınıf ve koleksiyonları	57
Tablo 4.1 Örnek kodlar	67
Tablo 4.2 Bağlantı bloklarının görevleri	72

EKLER LİSTESİ

- Ek – 1 : Sunucu bilgisayar, DS1104 denetleyici kartı, asenkron motor ve IP kameranin teknik özellikleri
- Ek -2 : Controldesk yazılımda kullanılan fonksiyonlar için Python dilinin bazı sınıf, metod ve özellikleri

SİMGELER LİSTESİ

B	: Sürtünme katsayısı
d^s-q^s	: Duran çatı eksenleri
d^r-q^r	: Rotor eksenleri
d^e-q^e	: Senkron çatı eksenleri
$e(t)$	: Hata
$E(t)$	: Karesel hata
i_a	: DA motorunun endüvi akımı, Amper
$i_{ds}^s - i_{qs}^s$	: Duran çatı eksen stator akımları, Amper
$i_{dr}^s - i_{qr}^s$	: Duran çatı eksen rotor akımları, Amper
$i_{ds}^e - i_{qs}^e$	: Senkron çatı eksen stator akımları, Amper
$i_{dr}^e - i_{qr}^e$	: Senkron çatı eksen rotor akımları, Amper
i_f	: DA motorunun alan akımı, Amper
J	: Eylemsizlik sabiti
k_t	: Moment sabiti
L_{lr}	: İndirgenmiş rotor kaçak endüktansı
L_m	: Ortak endüktans
L_s	: Stator özendüktansı
L_r	: İndirgenmiş rotor endüktansı
p	: Kutup sayısı
R_s	: Stator bir faz direnci
R_r	: İndirgenmiş rotor bir faz direnci
S	: Laplace operatörü
T_e	: Elektromekanik moment
T_L	: Yük momenti
x	: YSA/SBD giriş vektörü
w	: Ağırlık vektörü
v_{as}, v_{bs}, v_{cs}	: Stator faz gerilimleri, volt
$v_{ds}^s - v_{qs}^s$	: Duran çatı eksen stator gerilimleri, volt
$v_{dr}^s - v_{qr}^s$	: Duran çatı eksen rotor gerilimleri, volt
$v_{dr}^e - v_{qr}^e$	: Senkron çatı eksen rotor gerilimleri, volt

Δw	: Ağırlıklara uygulanacak düzeltme miktarı
δ	: Yöresel hata
θ	: Mekanik açısal konum, rad
θ_e	: Elektriksel açısal konum, rad
σ	: Kaçak faktörü
τ_r	: Rotor zaman sabiti
φ_f	: Uyartım akısı
$\Psi_{ds}^s - \Psi_{qs}^s$	: Duran çatıdaki stator akıları
$\psi_{dr}^s - \psi_{qr}^s$	: Duran çatıdaki rotor akıları
$\Psi_{ds}^e - \Psi_{qs}^e$	: Senkron çatıdaki stator akıları
$\Psi_{dr}^e - \Psi_{qr}^e$	: Senkron çatıdaki rotor akıları
ω	: Mekanik açısal hızı , rad/s
ω_e	: Senkron açısal hızı, rad/s
ω_r	: Rotor açısal hızı, rad/s
ω_{sl}	: Kayma açısal hızı, rad/s

KISALTMALAR LİSTESİ

AA	: Alternatif (Değişken) akım
ASP	: Aktif Sunucu Sayfası (Active Server Page)
BD	: Bulanık denetleyici
BM	: Bulanık mantık
BSA	: Bulanık sinirsel ağ
CNC	: Computer Numerical Control
DA	: Doğru akım
DAQ	: Veri Toplama (Data Acquisition)
DSP	: Sayısal işaret işleyici (Digital Signal Processing)
FPGA	: Programlanabilir mantık blokları (Field Programmable Gate Array)
FTP	: Dosya tranfer protokolü (File Transfer Protocol)
GPİB	: Veri toplama ve kontrol kartı (General-Purpose Interface Bus)
HTML	: Hareketli Metin İşleme Dili (Hyper Text Markup Language)
MWS	: Matlab Web Sunucu (Matlab Web Server)
PDE	: Matlab - Kısmı diferansiyel denklem aracı
PHP	: Web programlama dili (Hypertext Preprocessor)
PI	: Oransal+integral denetleyici
PWM	: Darbe genişlik modülasyonu
RTI	: Real Time Interface
RTW	: Matlab - Real Time Workshop
SBA	: Sinirsel bulanık ağ
SBD	: Sinirsel bulanık denetleyici
TCP/IP	: Transmission Control Protocol / Internet Protocol
XML	: Genişletilebilir işaretleme dili (eXtensible Markup Language)
YSA	: Yapay sinir ağları

ÖZET

Doktora Tezi

ASENKRON MOTORLARIN DENETİMİ İÇİN İNTERNET TABANLI SANAL VE UZAKTAN ERİŞİMLİ BİR LABORATUAR GELİŞTİRME

Ahmet TEKİN

Fırat Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

2009, Sayfa: 112

Elektrik makinaları ve denetim sistemleri gibi birçok elektrik mühendisliği ve teknolojisi derslerinde, teorinin uygulamalarla desteklenmesi gereklidir. Bu kapsamda, eğitim kurumlarının mali kısıtlamaları ve maliyeti yüksek laboratuvar donanımları nedeniyle öğrencilerin artan eğitim amaçlı taleplerini karşılamak zorlaşmaktadır. Bu nedenle, mühendislik bilimleri eğitiminde internetin kullanımı, öğrencinin uzaktan deneyler gerçekleştirmesini sağlayan sanal ve uzaktan erişimli laboratuvarlar geliştirmeye doğru yönelmiştir. Uzaktan erişimli laboratuvarlar genellikle LabView, Netmeeting ve Matlab/Simulink gibi yazılımlar kullanılarak gerçekleştirilmiştir. Matlab Web Sunucu (MWS) yazılımı, bilgisayar ve giriş/çıkış portlarını kullanarak gerçek zamanlı uygulamaların yapılabilmesi için uygun bir ortamdır. Ancak MWS, sayısal işaret işleyiciler (DSP) gibi bazı donanımları desteklemez. Bu nedenle, uzaktan erişimli bir laboratuvar geliştirmek için ek konfigürasyonların yapılmasına ve sunucu ile kullanıcı arasında iletişimi sağlayan bir arayüzün geliştirilmesine ihtiyaç duyulur.

Bu tezde, DSP denetimli asenkron motorlar için sanal ve uzaktan erişimli bir laboratuvar geliştirilmiştir. İlk aşamada, asenkron motorun oransal-integral (PI) ve sinirsel bulanık ağlar (SBA) ile denetimi için sanal bir laboratuvar ortamı hazırlanmıştır. Etkileşimli web sayfaları aracılığı ile denetim parametreleri değiştirilerek ya da kullanıcı tarafından geliştirilen denetim

algoritması sisteme yerleştirilerek sistemin benzetimi yapılabilir. İkinci aşamada ise uzaktan erişimli bir laboratuvar ortamı geliştirilmiştir. Uzaktan erişimli deney setinde, PWM üretimi ile akım ve hız denetim algoritmalarını gerçekleştirmek için Matlab/Simulink uyumlu çalışan DS1104 denetleyici kartı kullanılmıştır. Asenkron motorun PI ve SBA ile denetimi için gerekli uygulama dosyaları, Matlab Real Time Workshop (RTW) ve dSpace Real Time Interface (RTI) blokları kullanılarak hazırlanmış ve ControlDesk Developer yazılımı ile bir grafiksel kullanıcı arayüzü tasarlanmıştır. Controldesk'e uzaktan otomatik olarak erişebilmek için Python dili kullanılarak bir arayüz yazılımı hazırlanmıştır. Ayrıca, deneysel çalışmaların uzaktan gerçekleştirilebilmesi amacıyla istemci-sunucu iletişimini sağlayabilmek için Delphi programlama dili kullanılarak bir iletişim yazılımı geliştirilmiştir. Geliştirilen uzaktan erişimli laboratuvar ortamı, kullanıcıların denetim sisteminin parametrelerini değiştirerek ya da kendi denetim algoritmasını yerleştirerek deneylerini yapmasını ve deneysel sonuçlarını analiz etmesini sağlayacak şekilde esnek bir yapıda hazırlanmıştır.

Anahtar Kelimeler: Uzaktan erişimli laboratuvar, web tabanlı laboratuvar, sayısal işaret işleyiciler, asenkron motor denetimi.

ABSTRACT

PhD Thesis

INTERNET BASED VIRTUAL AND REMOTE ACCESS LABORATORY FOR INDUCTION MOTOR CONTROL

Ahmet TEKİN

Firat University
Graduate School of Natural And Applied Sciences
Department of Electrical and Electronics Engineering

2009, Page: 112

In many courses in the electrical engineering and science such as electrical machines and control, the feedback link between theory and practice should be provided. Furthermore, it is often difficult for schools with their limited and expensive equipment facilities to meet the increasing educational needs of students. Therefore, during the recent years, the use of the internet in the education of engineering sciences has been enhanced towards virtual and the remote access laboratory, allowing the students to conduct the experiments over the internet. Remote laboratories are mostly based on existing software platforms, such as LabView, and Netmeeting. Matlab Web Server (MWS) environment provides the real time operation using a PC and input/output boards. However, MWS does not allow online access to some hardware such as Digital Signal Processor (DSP). Additional configurations are needed and interface between local application and server must be developed for implementation of remote real laboratory.

In this thesis, a web-based virtual and remote access laboratory for digital signal processor (DSP) controlled induction motor (IM) is presented. In the first stage, a virtual laboratory environment is prepared for the simulation of proportional-integral (PI) and neuro-fuzzy controlled IM. Using the interactive web pages, the control system can be simulated by changing the control parameters or uploading of the control algorithms designed by the user. In

the second stage, remote laboratory environment is developed. The remote experimental rig uses MATLAB/Simulink compatible dSpace DS1104 signal processor to realize the control algorithm, current control and PWM modulation. Application files for PI and neuro-fuzzy control of IM are prepared with the MATLAB Real-Time Workshop (RTW) and dSpace Real Time Interface (RTI) and, a graphical user interface is designed using the dSpace ControlDesk Developer (CDD). An interface using the Python code is developed for remote automation of the control desk. Furthermore, server-client communication software with Delphi programming language is developed for remote implementation of the experiment. The web-based remote laboratory allows the students to conduct the experiment by changing the variables of control system and then to analyze the results on the remote computer.

Keywords: Remote laboratory, web-based laboratory, digital signal processor, induction motor control.

1. GİRİŞ

Günümüzde bilişim teknolojileri hızla gelişmekte ve internet kullanımı giderek artmaktadır. Yoğun kullanımıyla günlük yaşamımızı bir ağ gibi örmekle birlikte kişisel kullanım ve pek çok ticari şirket için her alanda bir başvuru aracı olmuştur. İnternet artık bankacılık, iletişim, sağlık, ticaret, şirket yönetimi, bilgi paylaşımı gibi birçok alanda olduğu gibi eğitim alanında da vazgeçilmez bir araç olarak karşımıza çıkmaktadır.

Bilgisayar Destekli Eğitim ile eğitim alanına giren bilgisayarlar, internet kullanımının artması ile uzaktan eğitime de yeni bir yön vermiştir. Ülkemizde yürütülen uzaktan eğitimin ders materyalleri kitap, televizyon veya kaset iken, şimdi bunların yerlerini web sayfaları almıştır. İnternet öğretmen-öğrenci ve okul-araştırma merkezleri arasında uygun çoklu ortam iletişim kanalı oluşturmaktadır. Yine internet özellikle uzaktan ve grup öğrenmelerini içeren birçok yeni öğrenme araçlarını sunmaktadır. Bu öğrenme araçları kullanılarak verilen eğitimin daha esnek, daha etkileşimli ve daha gerçekçi hale getirilmesine çalışılmaktadır. Çağdaş eğitim anlayışında, öğretim sürecinde kavramları yaparak öğrenme ve geri besleme sağlayabilmek giderek önem kazanmaktadır.

Mühendislik ve teknik eğitimde teorik bilgilerin, laboratuvar ortamında uygulanarak veya benzetimlerle desteklenerek pekiştirilmesi neredeyse zorunluluk haline gelmiştir. Günümüzde internet üzerinden laboratuvar ortamında deneylerin gerçekleştirilmesi veya yine internet üzerinden deneylerin benzetimlerle desteklenmesi mümkün olmuştur.

Teknolojik araçların ve buna bağlı olarak internetin gelişmesine paralel olarak, gerçek yaşama ait uygulamaların benzetim yoluyla sanal olarak oluşturulması, son günlerde seçilen yöntemlerden biridir. Laboratuvar konusunda maddi sıkıntılar çeken eğitim kurumları genellikle teorik bilgileri benzetimlerle desteklemeyi tercih etmektedir. Çeşitli alanlara yönelik eğitim amaçlı yazılımlar geliştirilmeye başlanmıştır. Özellikle uygulamalı bilimler alanında deneysel çalışma öncesi benzetim programları yardımı ile bilgisayar ortamında deneyin benzetiminin yapılması değer kazanmıştır.

Bugünlerde uzaktan erişimli, gerçek zamanlı denetim laboratuvarları da giderek önem kazanmaya başlamıştır. Uzaktan erişimli denetim laboratuvarları, web üzerinden kaynakları paylaşmakta, çeşitlilik ve zaman kavramlarında esneklik sağlamaktadır. Bunun yanında aynı donanımları birçok kişi kullandığı için maliyeti de düşürmektedir. Gerek endüstri gerekse eğitim alanındaki donanımlara uzaktan erişim aynı zamanda kullanıcıların ulaşım gibi sorunlarını da ortadan kaldırmaktadır. Özellikle sürücü sistemlerin denetiminde maliyeti yüksek donanım gerektiğinden her eğitim kurumunun bu donanımı temin etmesinde de mali zorluklarla

karşılaşmakta ve bu sistemlerin endüstriye uygulanabilirliği gecikmektedir. Bu da ancak eğitim kurumlarında sınırlı sayıda bulunabilen maliyeti yüksek donanımlara internet üzerinden erişerek zaman, ulaşım, sermaye gibi kavramları ortadan kaldırılarak sağlanabilmektedir. Bu nedenlerden dolayı birçok üniversitede çeşitli alanlara yönelik eğitim amaçlı yazılımlar geliştirilmeye başlanmış ve özellikle uygulamalı bilimler alanında deneysel çalışmaların yapılması önem kazanmıştır.

1.1. Literatür Taraması ve Değerlendirilmesi

Literatürde internet üzerinden, gerek gerçek zamanlı deneysel çalışmalar gerekse benzetim üzerine birçok çalışmanın yapıldığı gözlenmektedir. Bu çalışmalardan bazıları aşağıda verilmiştir:

Tzeng ve Tien [1] tarafından yapılan çalışmada, National Taiwan Normal Üniversitesinin Elektrik Mühendisliği bölümü öğrencilerine yönelik, elektrik makineleri deneyleri için etkileşimli, gerçekçi ve esnek bir yapıya sahip web tabanlı bir laboratuvar ortamı sunulmuştur. Hazırladıkları laboratuvar ortamında öğretmen ve öğrenciler için insan-insan ve insan-bilgisayar arasında etkileşim bulunmaktadır. Sadece bireysel öğrenme değil aynı zamanda grup öğrenmesini de destekleyecek şekilde hazırlanmıştır. Laboratuvarın hazırlanmasında Sanal Gerçeklik (Virtual Reality) teknolojisi, Java ve HTML dilleri kullanılmıştır.

Cui ve diğ. [2] elektrik ve elektronik eğitiminde kullanılmak üzere, veri toplama ve denetiminde, veri analizi ve veri sunumunda kullanılan, G programlama dili tabanlı bir grafiksel program geliştirme ortamı olan Labview yazılımını kullanarak sanal bir laboratuvar gerçekleştirmişlerdir. Elektroteknik derslerine yönelik hazırladıkları sanal laboratuvar ortamında veri toplama kartı (DAQ) kullanarak, ağ üzerinden geleneksel laboratuvar ortamı gibi çalışma imkanlarını öğrencilere sunmuşlardır. Animasyonlarla öğrencilerin ilgisini artırmak için Macromedia Flash ve HTML ortamları kullanılmıştır. Ayrıca yapılan çalışmaları izlemek için web kamera düzeneği kurulmuştur.

Jou [3] tarafından yapılan çalışmada, üniversite ve meslek yüksek okulu öğrencilerine yönelik Makine Üretimi Teknolojisi dersi için web tabanlı bir eğitim sistemi geliştirilmiştir. CNC benzetimleri ve CNC kodlama metodlarını kapsayan eş zamanlı öğrenme materyalleri ile gerçek uygulamalardan önce teorik bilgiler öğrencilere verilerek benzetimler yapılabilir. CNC'deki dönme ve delme gibi gösterilerle öğrencilerin öğrenmesi amaçlanmıştır. Öğrencilerin veri girmesi, ders materyallerine erişmesi gibi işlemlere müsaade eden web tabanlı yapı, HTML, XML, 3D Web Player, Java ve Virtual Reality teknolojileri kullanılarak hazırlanmıştır.

Cheng ve diğ. [4], Hong Kong Polytechnic Üniversitesinin Elektrik Mühendisliği bölümü öğrencileri için Güç Elektroniği derslerinde kullanılmak üzere web tabanlı bir sanal laboratuvar hazırlamışlardır. Buradaki sanal laboratuardan kasıt web tabanlı benzetim çalışması anlamında değil, gerçek güç elektroniği deneylerinin yapıldığı anlamındadır. Laboratuvarın oluşturulmasında JavaScript, HTML, Flash ve Labview gibi web programlama ortamları kullanılmıştır. Laboratuardaki tüm ölçüm, sinyal üretim ve güç kaynağı cihazları ile Labview yazılımı arasındaki iletişim, cihaz denetimi çözümlerinde kullanılan GPIB kartı ve DAQ kartı ile kurulmaktadır. DC/DC konverter ve akım anahtarlama konverter deneyleri yapılmaktadır.

Li ve Khan [5] tarafından yapılan çalışmada, Kingsville Texas A&M Üniversitesi tipik elektrik ve elektronik devrelerin benzetim, tasarım, analiz ve ölçümleri için laboratuvar derslerine yönelik bir bilişim teknolojisi uygulaması sunulmuştur. Osilaskop kullanımı, güç amplifikatörleri, RL, RC ve RLC devrelerinin geçici durumları gibi konuların bulunduğu 14 deney hazırlanmıştır. Gerçek devrelerin ölçümünde, bilgisayarlı veri elde etmek için DAQ kartları kullanılmıştır. Sanal laboratuvarın geliştirilmesinde devre benzetimleri için bilişim teknolojilerinin benzetim yazılımlarından Labview aracı kullanılmıştır. Analiz ve tasarımda ise yine bilişim teknolojilerinin matematiksel araçlarından biri olan MathCad kullanılmıştır. Ölçüm, benzetim ve teorik hesaplama arasındaki sonuçların karşılaştırılması ve veri kaydı için bilişim teknolojilerinin tablo oluşturma araçlarından Microsoft Excel kullanılmıştır.

Magistris [6], İtalya Naples Federico II Üniversitesindeki farklı seviyelerdeki Elektrik Mühendisliği müfredatlarına uygun, Temel Elektrik ve Elektromanyetik derslerinde kullanılmak üzere Matlab Web Sunucu (MWS) tabanlı bir sanal laboratuvar ortamı hazırlamıştır. Daha karmaşık çözümler için Matlab'ın PDE aracı kullanılmıştır. Elektrik alanları ve manyetik alanlar ile ilgili benzetimler web üzerinden yapılabilmektedir.

Ko ve diğ. [7] tarafından yapılan çalışmada, Singapore National Üniversitesindeki öğrencilere Haberleşme dersleri ile ilgili frekans modülasyon deneylerinin yapılabildiği web tabanlı sanal laboratuvar uygulaması sunulmuştur. Sistem sunucu-istemci yapısı ile çalışmaktadır. Java, JavaScript, Labview ve G ortamları kullanılarak arayüzler hazırlanmıştır. Spektrum analiz cihazı, frekans sayıcı ve sinyal generatörü gibi programlanabilir cihazlar, GPIB ve DAQ kartı üzerinden bir sunucu bilgisayara bağlanmıştır. Hazırladıkları bu sanal laboratuvar ortamı, öğrencilere çok pahalı olan spektrum analiz cihazını kullanarak deneyim kazanmalarına yardımcı olmakta ve uzaktan mühendislik eğitimi için bir çözüm sağlamaktadır. Yine Ko ve diğ.[8] tarafından yapılan başka bir çalışmada, çok girişli-çok çıkışlı eş iki tank düzeneği ile ilgili gerçek zamanlı bir laboratuvar deneyinin web tabanlı uygulaması sunulmuştur. Hazırlanan uzaktan erişimli deney ortamı Singapur National Üniversitesi Elektrik Mühendisliği bölümü öğrenci ve personeline hizmet vermek üzere gerçekleştirilmiştir. Labview, Java ve HTML

ortamları kullanılarak arayüzler tasarlanmıştır. Ses ve görüntü geri beslemesi için Microsoft NetMeeting programından yararlanılmıştır.

Ferrero ve Piuri [9], veri toplama kartı ile ağ üzerinden uzaktan ölçümlere ve benzetimlere izin veren modüler bir sanal laboratuvar yapısı hazırlamışlardır. Sinyal generatörleri ile ilgili uygulamalar bulunmaktadır. Sunucu-istemci mimarisi ile çalışan bir yapı kullanılmıştır. Sanal araçları oluşturmada Labview ortamı kullanılırken, arayüzler için istemci yazılım ve donanımından bağımsız Java, CGI ve HTML dillerini içeren web teknolojisi kullanılmıştır. Yine Ferrero ve Salicone [10] yaptıkları çalışmada, elektriksel ölçümlerde düşük maliyetli bir çözüm sunmak amacıyla, bir web sayfası ile uzaktan erişimli bir laboratuvarı entegre etmek ve üç-faz üç-üçlü yük ile ilgili deneylerin uzaktan nasıl gerçekleştirileceğini göstermişlerdir. Web sayfaları tasarımında Macromedia Flash ortamı ve uzaktan erişimli laboratuvar ortamı için de Labview ortamı kullanılmıştır. Öğrencilerin uzaktan erişimli laboratuvara erişimleri için herhangi bir özel yazılım bulunmamaktadır. Öğrencilerin ihtiyaç duyduğu teorik bilgiler için bir web sayfası hazırlanmış, tartışma ortamları oluşturulmuş ve tartışılan çözümler için farklı ölçüm araçları ve teknolojilerin bulunduğu gerçek test ortamlarına erişimler gerçekleştirilmiştir. Öğrencilerin web ortamında teorik ve laboratuvar derslerine tam zamanlı olarak erişimleri ile derslerine önceden hazırlanmaları amaçlanmıştır.

Guimaraes ve diğ. [11] tarafından yapılan çalışmada, XR4000 hareketli robotuna uzaktan erişimli bir sanal laboratuvar sunulmuştur. Bu sanal laboratuvar sıradan bir internet uygulaması değil, yeni bir nesil haberleşme hizmeti olarak gerçekleştirilmiştir. Komponent tabanlı yazılım yapısı ve çoklu ortam akışına destek verebilen karmaşık bir erişim altyapısı kullanmaktadır. Uzaktaki kullanıcı internet üzerinden Routera ulaşmakta ve buradan Wireless ile XR4000 hareketli robota ulaşmaktadır. Mimaride kullanılan iki sunucudan biri ile robot denetimi yapılırken diğeri ile de robotun video görüntüleri kaydedilmektedir. Sanal laboratuvarın oluşturulmasında HTML, XML, Java, C++ vb. yazılım ortamları kullanılmıştır.

Albu ve diğ. [12] tarafından yapılan çalışmada, internet tabanlı deney ortamı ile yeni bir sanal laboratuvar yaklaşımı sunulmuştur. Bu sanal laboratuvar birkaç üniversite arasında dağıtılmış gerçek donanımları kullanmaktadır. Labview ortamında istemci-sunucu mimarisi oluşturularak, çok kullanıcı bir yazılım çözümü gerçekleştirilmiştir. Karşılıklı sohbet, video ve arşivleme gibi detaylar düşünülmüştür. Örnek uygulama olarak düşük gerilim kontaktörleri ve röle denetimi seçilmiştir. Bu uygulama Bucharest Politehnica Üniversitesi Elektrik Mühendisliği öğrencileri ile Arizona State Üniversitesinin Güç Sistemleri Mühendisliği öğrencileri arasında uygulanmıştır.

Demirbaş [13], internet üzerinden uzaktan erişimli bir doğru akım (DA) motoru deney düzeneği geliştirmiştir. Deney düzeneği laboratuvar ortamında bir sunucu bilgisayara

bağlanmıştır. Kullanıcılar internet bağlantısı olan herhangi bir bilgisayardan deney düzeneğine bağlanabilmekte ve gerçek zamanlı olarak deneyleri gerçekleştirebilmektedir. Yapılan çalışma Gazi Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi Bölümü'nde Elektrik Makineleri derslerinde uygulanmıştır. Deneyde internet üzerinden motorun referans hız değerleri ve oransal-integral (PI) katsayıları değiştirilerek motor tepkisi ölçülmüştür. Kullanıcı ölçülen değerleri bilgisayar ekranında görebildiği gibi deney düzeneğine bağlı bir kamera yardımıyla osiloskop ekranından gerçek sonuçları da görebilmektedir.

Uran ve Jezernik [14] tarafından, Maribor Üniversitesi Elektrik Elektronik Fakültesi'nde Matematik, Güç Elektroniği ve Denetim Sistemleri Tasarımı dersleri için MWS tabanlı bir sanal laboratuvar ortamı sunulmuştur. 2005 yılından beri kullanılan sanal laboratuvar ortamı, öğrencilerin kendi yazdıkları Matlab m-dosyalarını internet üzerinden çalıştırmalarına izin vermektedir. Sadece bir yazı editörü ve internet tarayıcısı ile öğrenciler benzetimlerini hazırlayabilmektedir. Uran ve diğ. [15] tarafından yapılan diğer bir çalışmada ise Denetim Sistemleri Tasarımı dersleri için MWS tabanlı sanal bir laboratuvar ile yaparak öğrenmeyi desteklemek amacıyla uzaktan erişimli gerçek bir laboratuvar sunulmuştur. DA motor kaskat denetimi, iki eksenli mekatronik aygıt denetimi ve RC osilatör deneyleri internet üzerinden gerçekleştirilmiştir. MWS'ın eğitimdeki etkileşimli uygulamaları için web tabanlı benzetimlerini geliştirme aşamalarını anlatan bir başka çalışma Pester ve Ismailov tarafından sunulmuştur [16]. Irmak [17] ise teknik eğitimde yardımcı ders aracı olarak kullanılabilmek amacıyla DA motorunun web tabanlı benzetim çalışmasını sunmuştur.

Shiakolas ve diğ. [18] tarafından, internet üzerinden mühendislik alanındaki analiz, tasarım ve benzetimleri için araç geliştirme yöntemleri sunulmuştur. Araç geliştirirken güçlü ve çok yönlü bir yazılım paketi olan Matlab ve Matlab'ın web araçlarından yararlanılmıştır. Gerçek zamanlı etkileşimli benzetimlerle mühendisliğin temellerinin anlaşılmasını kolaylaştırmak amaçlanmıştır. Yine Shiakolas ve diğ. [19], [18]' de anlattıkları yöntemleri kullanarak internet üzerinden kullanılabilen, doğrusal zamanla değişmeyen dinamik sistemler, geri beslemeli denetim sistemleri için benzetim ve analiz araçları hazırlamışlardır. Bu çalışmaya benzer bir çalışma da Yan ve diğ. [20] tarafından gerçekleştirilmiş ve MWS tabanlı sanal bir sinyal işleme laboratuvarı oluşturulmuştur.

Albino Mendez ve diğ. [21] tarafından yapılan çalışmada, tek giriş tek çıkışlı, hem sürekli hem de ayrık zamanlı lineer denetim sistemlerinin benzetimi ve analizi için geliştirdikleri web tabanlı araç sunulmuştur. Web tabanlı araç Java Appletleri kullanılarak hazırlanmış ve sonuçlar grafiksel olarak elde edilmiştir. Hazırlanan araç La Laguna Üniversitesi Kontrol Mühendisliği bölümü öğrencileri üzerinde denenmiş ve öğrenciler üzerinde pozitif sonuçlar alınmıştır.

Gokbulut ve diğ. [22–23], fırçasız doğru akım motoru ile asenkron motor için, hız denetleyici olarak yapay sinir ağları ile sinirsel bulanık ağ ve oransal-integral (PI) denetleyicilerin kullanıldığı eğitim araçları geliştirmişlerdir. Hazırlanan eğitim araçları sırasıyla Delphi ve C++ ortamlarında kodlanmıştır. Sadece yerel kullanıcılara hitap etmektedir.

Akçayol ve diğ. [24], darbe genişlik modülasyonu (DGM- PWM-Pulse Width Modulation) işlevini gerçekleştiren bilgisayar destekli bir yazılım geliştirmişlerdir. Hazırlanan yazılım için Delphi programlama dili kullanılmıştır. Bu yazılım, öğrencilerin DGM işlevini daha iyi anlamalarını sağlayan bir eğitim aracıdır. Bu araç sayesinde öğrenciler DGM tekniğini laboratuvar ortamı gereksiz ev ortamında bile eğitime devam edebilmektedirler. Yine Akçayol ve diğ. [25], doğru akım motorunun hız denetiminde, bulanık mantık ve klasik denetleyicilerinin karşılaştırılması ve daha iyi öğrenilmesi için öğrencilere yardımcı olacak şekilde hazırlanan bir sanal uygulama ortamı sunulmuştur. Bu çalışma, hem lisans hem de lisansüstü öğrencilerine yönelik eğitsel bir araç olarak görülmektedir. Öğrencilerin, sanal uygulamada farklı parametreler ile farklı çalışma ortamları hazırlayarak bu iki denetleyicinin davranışlarını incelemeleri amaçlanmıştır. Akçayol, Elmas ve diğ. [26–27], yukarıda belirtilen yapıda, bulanık mantık denetleyicilerini kullanarak sırasıyla, fırçasız doğru akım motoru ve asenkron motor için bir paket program hazırlamışlardır. Akçayol, Elmas ve diğ. [24–27]’nin yaptıkları çalışmalar sadece yerel kullanıcılara hizmet vermektedir.

Hercog ve diğ. [28], Maribor Üniversitesi Bilgisayar Bilimi ve Elektrik Mühendisliği Fakültesi Kontrol Mühendisliği eğitimi ve araştırmaları için uygun olan, DSP (Digital Signal Processing) tabanlı hızlı denetim sistemleri sunulmuştur. Bu sistemler Matlab/Simulink’deki model tabanlı denetim sistemi tasarımından, DSP tabanlı hedef uygulamaya kolay bir geçiş sağlamaktadır. Matlab/Simulink/Real Time WorkShop (RTW) ile Labview gibi iki ticari yazılım ve özel geliştirilmiş donanımdan oluşmuştur. Denetim algoritmalarını geliştirmek, benzetim, çevrimdışı analizler ve hızlı çalıştırılabilir kod üretimi için Matlab/Simulink ve RTW yazılımları kullanılmıştır. Labview VI ile de çevrimiçi parametre değişikliği, sinyal izleme ve çevrimiçi analiz uygulamaları gerçekleştirilmiştir. DSP–2 RCP olarak adlandırılan sistem güçlü, esnek ve kullanımı kolay bir yapıya sahip olması nedeniyle denetim sistemleri araştırmaları gibi eğitim süreçlerinde kullanılabilir. DA motorunun kaskat denetiminin bir örneği verilmiştir.

Cheneval ve diğ. [29] tarafından Java programlama dili kullanılarak web üzerinden paylaşılabilen ve dağıtılabilen etkileşimli öğrenme materyalleri hazırlanmıştır. Sinyal işleme laboratuvarı için blok diyagram tabanlı bir yaklaşım geliştirilmiştir. Küçük dalga teorisiyle ilgili 12 bölümden oluşan bir laboratuvar ortamı hazırlanmıştır. Blok diyagram modeli, her bir blok

sunucu bilgisayardan direkt olarak alıp getirebilmesi ve tüm ortak modelleri önceden hazırlamaya izin vermesi gibi özelliklerinden dolayı esnek bir yapıdadır.

Changhong ve diğ. [30], web tabanlı uzaktan denetim servisi (WBRCS) diye adlandırılan yeni bir tasarım önermişlerdir. Sistem eşzamanlı olarak birçok uzaktan denetim hizmetlerini sağlamaktadır. Kişinin sadece bir internet tarayıcısına ihtiyacı bulunmaktadır. Bir Java uygulaması olan uzaktan denetim hizmetlerinin konsoluna, internet tarayıcısı ile ulaşılabilmektedir. Her bir denetim hizmeti dinamik olarak oluşturulmuştur. Hizmet bileşenleri arasındaki iletişim için XML kodları kullanılmıştır. Kamera denetim hizmeti ve motor denetim hizmeti olmak üzere iki tane uzaktan denetim uygulaması geliştirilmiştir.

Hurley ve Lee [31], Elektrik Elektronik Mühendisliği öğrencilerine yönelik, iPES diye adlandırılan ve güç elektroniğinin temelleri ile ilgili hem pratik hem de teorik bilgiler bulunan uzaktan erişimli web tabanlı uygulama laboratuvarı geliştirmişlerdir. Laboratuvar uygulamalarının sayısı genişletilebilen esnek bir yapıya sahiptir. Öğrencilerin ön laboratuvar araştırmalarını desteklemektedir. Pspice programı kullanılarak bazı uygulamaların benzetimleri hazırlanmıştır. Matlab programı ile de öğrencilerin kontrol ve geri beslemelerini analiz etmeleri sağlanmıştır. Ayrıca bir DA/DA konverter deneyi, GPIB kartı ve Labview VI yazılımı kullanılarak gerçek zamanlı olarak internet üzerinden gerçekleştirilmiştir.

Palma ve diğ. [32] tarafından yapılan çalışmada, Texas A&M Üniversitesi Elektrik Elektronik Mühendisliği 2. sınıf öğrencilerinin Elektrik Devre Teorisi dersi için WebCT diye adlandırılan etkileşimli ev ödevi sistemi web tabanlı hazırlanmıştır. Web sayfasında ders ile ilgili müfredat programı, ev ödevi, ev ödevi çözümleri, laboratuvar, tasarım, yardım, mail ve sohbet gibi bölümler bulunmaktadır. Özellikle WebCT içerisinde bulunan ara sınav modülü, soru bankası ve web üzerinden yayınlanması üzerinde durulmuştur. Elektriksel devrelerin matematiksel modellerini türetmeye ve ev ödevlerinin çözümünde öğrencilere kolaylıklar sağlamaktadır.

Huang ve Lu [33], CAI ismini verdikleri bilgisayar destekli bir yapı sunmuşlardır. CAI sistemi, bilgi modülü, araç modülü, sınav modülü ve sıkça sorulan sorular modülü olmak üzere dört modülden oluşmaktadır. Bu sistem osilaskop, fonksiyon generatörü ve lojik analizör gibi elektronik araçların nasıl kullanılacağını öğrencilere öğretmek için kullanılmaktadır. Ayrıca CAI sistemi uzaktan eğitimde de kullanılabilecek bir yapıdadır. CAI sistemi Java ve HTML dilleri kullanılarak hazırlanmıştır.

Marin ve diğ. [34], Adobe Shockwave ve Macromedia Director yazılımları kullanılarak, modern mikroişlemciler, yapıları ve uygulamaları ile ilgili derslerde kullanılmak üzere, İspanya Seville Üniversitesi Haberleşme Mühendisliği öğrencileri için TAM ismi verilen, web tabanlı bir eğitim aracı hazırlanmıştır. Web sayfalarının hazırlanmasında PHP dili kullanılmıştır.

Hazırlanan web sayfasında laboratuvar uygulamaları, teorik dersler, kendi kendine öğrenme senaryoları, sohbet ve tartışma gibi ortamlar mevcuttur.

Shiakolasan ve Piyabongkarn [35], sistem modelleme ve denetim eğitimi için HIL ismini verdikleri gerçek zamanlı sayısal denetim sistemlerinin geliştirilmesi üzerine bir çalışma sunmuşlardır. Donanım olarak standart kişisel bilgisayarlara (host-target) ve DAQ kartına, yazılım olarak da MathWorks'un xPC Target, Simulink, RTW araçlarına ve Microsoft Visual C/C++ derleyicisine ihtiyaç bulunmaktadır. Host-target bilgisayar ortamı denetim algoritmalarının Simulink blokları kullanılarak gerçekleştirilmesine ve etkileşimli bir biçimde sistemin gerçek zamanlı olarak izlenmesine izin vermektedir. Deneysel sonuçlar daha sonraki kullanımlar için Matlab formatında kaydedilmektedir.

Kozick ve Crane [36] tarafından, Bucknell Üniversitesi Elektrik Mühendisliği öğrencileri için sistem modelleme, gerçek zamanlı sayısal sinyal işleme, benzetim ve denetim amaçlı bir laboratuvar geliştirilmiştir. Geliştirilen laboratuvar ortamı öğrencilerin için benzetim ve gerçek sistemler arasındaki uçurumlara bir köprü görevi görmektedir. Öğrenciler deneysel ölçümler ile bilgisayar benzetimi sonuçları farklılıkları yorumlayabilmektedir. Bir sıvı seviyesinin denetimi ile ilgili deneysel örnek bir çalışma yapılmıştır.

Keyhani ve diğ. [37] tarafından yapılan çalışmada, motor sürme uygulamaları için DSP tabanlı denetim yöntemlerini içeren, internet tabanlı görsel bir eğitim aracı geliştirilmiştir. Sistem gerçek donanım birimlerinin bulunduğu bir set ve sayısal denetim algoritmalarının gerçekleştirilebilmesi için grafiksel bir arayüzden oluşmaktadır. Grafik arayüzü için Borland C++ Builder dilinde arayüzler hazırlanmıştır. Sistemde bulunan 3 fazlı fırçasız DA motorunun denetimi için PWM ve gerilim kaynaklı inverter tasarlanmıştır.

Swamy ve diğ. [38], ücretsiz olarak bulunabilen Microsoft NetMeeting yazılımını kullanarak, eğitim amaçlı laboratuvarlardaki donanımların uzaktan kontrol edilebileceğini göstermişlerdir. Çalışmada Microsoft NetMeeting programının kullanımı anlatılmıştır. İnternet tabanlı denetim amacına uygun olarak, Microsoft NetMeeting programının kullanımı açıklanmıştır. Denetim deneyleri, Matlab'ın gerçek zamanlı araçları ve NetMeeting sayesinde uzaktan erişilebilir duruma getirilmiştir. Öğrencilerin Java gibi yüksek seviyeli ortamlarda kod yazması yerine denetim algoritmalarına odaklanması amaçlanmıştır. Elde edilen veriler FTP ortamına kaydedilerek, diğer kullanıcıların paylaşımına açılmıştır.

Shen ve diğ. [39], otomatik internet ölçüm laboratuvarı (AIM-LAB) olarak adlandırdıkları internet tabanlı bir laboratuvar ortamı gerçekleştirmişlerdir. Uzaktan eğitim amacıyla gerçekleştirilen AIM-LAB ile laboratuvar ölçüm cihazlarının etkili bir şekilde kullanılabilmesi sağlamıştır. Bir yarı-iletken cihazın karakteristik değerlerinin incelenmesi ile ilgili bir uzaktan erişimli deney gerçekleştirilmiştir. İstemci yazılımı Java tabanlı olup, AIM-

LAB’da kullanılan HPIB sürücü kartı C++ dilini desteklemesi sebebiyle, sunucu yazılımı Visual C++ tabanlı yazılmıştır. AIM-LAB ile mühendislik eğitimi, özellikle kaynak sıkıntısı olan internet üzerindeki uygun insan grupları ile daha cazibeli bir hale getirilebilir.

Oltean ve diğ. [40] tarafından, uzaktan fırın sıcaklığının elektriksel ısı denetimi ve izlenmesi ile ilgili bir çalışma sunulmuştur. Bu dijital sistem yerel bir dijital denetimci ve sunucu-istemci uygulaması olmak üzere iki kısımdan oluşmaktadır. Fırın sıcaklığının elektriksel ısı denetimi için PIC16F628 mikroişlemcisi kullanılmıştır. Sunucu-istemci uygulaması Labview/CVI dilinde gerçekleştirilmiş ve yerel dijital denetimcinin parametre değişimi ve izleme için Transmission Control Protocol / Internet Protocol (TCP/IP) protokolü kullanılmıştır. Hazırlanan bu ortam ile sıcaklık ölçümlerinin on-line görünmesi, düşük enerji talebi, düşük maliyet, yüksek güvenilirlik, denetim parametrelerinde çok az osilasyon ve herhangi bir zamanda on-line parametre değiştirme imkanı gibi üstünlükler sağlanmıştır.

Watson ve diğ. [41] tarafından yapılan çalışmada, North Dakota State Üniversitesi mühendislik bölümlerindeki uzaktan eğitim öğrencileri için internet üzerinden erişimli bir laboratuvar ortamı geliştirmişlerdir. Bu ortamda, elektrik devrelerinin analizi, denetim teorisi, materyal seçimi ve PLC cihazlarının kullanımı ile ilgili çeşitli benzetimler ile temel denetim deneylerini içeren gerçek zamanlı uygulamalar bulunmaktadır. Daha çok elektrik ve kimya mühendisliği öğrencilerine yönelik olan internet tabanlı laboratuvar 2002 yılından beri kullanılmaktadır.

Spanias ve Atti [42], internet tabanlı sanal bir sinyal işleme laboratuvarı sunmuşlardır. Java Digital Signal Processing (J-DSP) diye adlandırılan laboratuvar nesneye yönelik Java appletleri üzerine kuruludur. Arizona State Üniversitesi Elektik Mühendisliği Bölümünde geliştirilen J-DSP halen kullanılmaktadır. J-DSP, sinyal işleme fonksiyonlarının birçoğuna sahip zengin bir yapısı bulunmaktadır. Ayrıca filtre tasarımı ve modern sinyal işleme algoritmaları gibi konularda benzetim modülleri oluşturulmuştur. J-DSP laboratuvar yazılımı ve uygulamaları, Elektrik Mühendisliği öğrencileri için online laboratuvar uygulamaları sağlayan, gömülmüş animasyonlar ile programlanabilir internet benzetimlerine müsaade eden ve öğrenme oranını yükselten bir ortamdır.

Chirico ve diğ. [43] tarafından yapılan çalışmada, elektronik ölçümleri ile ilgili online sanal bir laboratuvar oluşturmak ve internet üzerinden gerçek laboratuvarları paylaşmak için bir ortam hazırlanmıştır. Teorik bilgilerin uygulanması bilimsel öğrenmede önemli bir yaklaşımdır. Ancak bu faaliyetleri desteklenmek için web tabanlı eğitimlerde uygun bir çözüm bulunamamıştır. Online laboratuvarlar bu konuya hitap etmektedirler. Online laboratuvarlar gerçek deneylerin web üzerinden uygulanmasına izin vermekte ve uzaktaki deneysel araçlar ile kullanıcılar arasındaki uyumluluğu sağlamaktadırlar. Bu amaçla sanal laboratuvar sunucusu

(VLS) ve gerçek laboratuvar sunucusu (RLS) olmak üzere iki tane sunucu oluşturulmuştur. Deneyler farklı gerçek laboratuvarlarda dağıtılarak daha geniş bir ağa yayılmıştır ve yerel bir bilgisayar ile sunucular kontrol edilmiştir.

Familia [44], Robotik ve Mekatronik derslerinde kullanmak amacıyla, VIRLAB ismini verdiği uzaktan erişimli bir laboratuvar hazırlamıştır. Üç üniversite arasında paylaştırılmış, gerçek robot ve CNC makinelerinin bulunduğu üç ayrı laboratuvar ortamı bulunmaktadır. Senkron öğrenme, asenkron öğrenme, eş zamanlı dersler gibi özelliklere sahiptir. Sanal Gerçeklik (Virtual Reality) ve Java ortamları kullanılarak internet tabanlı laboratuvar gerçekleştirilmiştir. İstemci tarafı kullanımı kolay, herhangi bir platformdan bağımsız ve ek yazılıma ihtiyaç duymamaktadır.

Ciubotariu ve Hancock [45], Calgary Üniversitesi Elektrik ve Bilgisayar Mühendisliği Bölümünde uzaktan eğitim amaçlı bir sanal laboratuvar hazırlamışlardır. Uzaktan erişimli cihaz denetimi ve temel dijital devre deney tasarımı olmak üzere iki ortam hazırlanmıştır. Laboratuvarın ana amacı osilaskop, dalga üretici ve güç kaynağı gibi temel cihazların kullanımı için uzaktan erişimli bir eğitim sitesi oluşturmaktır. Ayrıca trafik ışıklarının denetimi ile ilgili deneyler uzaktan erişimli olarak gerçekleştirilmiştir.

Hashemian ve Riddley [46] tarafından yapılan çalışmada, sınıftan veya internet erişimi olan herhangi bir yerden, öğrencilerin laboratuvar deneylerini gerçekleştirmeleri için uzaktan erişimli bir laboratuvar sunulmuştur. Laboratuvar ortamı dijital devre tasarımlarında kullanılan FPGA platformunu kullanarak, dijital devre tasarım derslerine yardımcı olmaktadır. İnternet erişimli laboratuvarın arayüzü Labview yazılımı ile hazırlanmış ve DAQ kartı, FPGA kartı, osilaskop, dalga generatörü cihazları da donanım yapısında kullanılmıştır. Microsoft Windows Xp'nin uzak masa üstü bağlantısı kullanılarak laboratuvara uzaktan tam erişim sağlanmaktadır. Benzer bir çalışma da Persiano ve diğ. [47] tarafından yapılmıştır. FPGA uygulamalarının uzaktan denetimine izin veren laboratuvar ortamında iki tekerlekli araç ve robot kolu deneyleri yapılabilmektedir. FPGA kartına ulaşmak için Microsoft Windows Xp'nin uzak masa üstü bağlantısı kullanılmış ve Quartus II yazılımı ile de deneyler gerçekleştirilmiştir.

Sanchez ve diğ. [48] tarafından yapılan çalışmada, Yazılım Mühendisliği, Bilgisayar Mühendisliği gibi bölümlerde gelişmiş dersleri öğretmek amacıyla, uzaktan erişimli deneysel çalışma yapmaya yönelik bir laboratuvar ortamı oluşturulmuştur. Geliştirilen laboratuvar ile öğrencilerin yaparak öğrenmesi desteklenmiştir. Yapılan çalışmada internet üzerinden kontrol edilebilen bir demiryolu örneği sunulmuştur. Laboratuvar ortamında 5 adet lokomotiften oluşan bir tren, 14 istasyon, 6 semafor ve bilgi toplama amaçlı kullanılan 21 adet hassas sensör bulunmaktadır. Java ortamında hazırlanan arayüz sayesinde, öğrenciler treni hareket ettirebilmekte, hızını ve yönünü değiştirebilmektedirler. Ayrıca, sabit bir kamera ile düzenek

izlenmekte ve çok sayıda kablosuz mikro-kameralarla ayrı ayrı noktalardan görüntüler elde edilmektedir.

Bellmunt ve diğ. [49], İspanya Catalonia Teknik Üniversitesi Elektrik Mühendisliği Bölümü öğrencileri, yüksek lisans ve doktora öğrencileri için PLC cihazlarının uzaktan denetimi ve deneysel amaçlarla internet üzerinden kullanılabilmesini sağlayan çalışmayı sunmuşlardır. PLC cihazlarındaki Ethernet kartı ile TCP/IP protokolü kullanılarak iletişim kurulmuştur. Çalışmalar IP kamera ile izlenmiştir.

Yukarıdaki çalışmalara benzer şekilde yapılan diğer sanal laboratuvar veya uzaktan erişimli gerçek laboratuvar çalışmaları; güç sistemleri [50–51], elektrik devreleri [52–53], doğru akım motorları [54–58], RC osilatör [59], asenkron motorlar [60], filtre tasarımı [61], denetim sistemleri [62–66], modülasyon [67], elektriksel güç istasyonları [68], yapay sinir ağları [69], sayısal işaret işleme [70–72], elektromanyetik [73–74], yarı iletkenler [75], kimya [76], ölçüm ve analiz cihazları [77], dijital devreler [78], robot denetimi [79–81], elektrik makinaları [82–83], adım motoru [84] gibi çeşitli alanlara yönelik hazırlanmıştır.

Taranan literatür özetinde, özellikle uzaktan eğitimde etkileşimi ve eğitim kalitesini artırmak veya yüksek maliyetleri ortadan kaldırmak amacıyla internet ve bilgisayar teknolojilerinin kullanıldığı görülmektedir. Uzaktan eğitimin amacı, maliyetlerin azaltılması, öğrencinin sınıf ortamından kurtarılması, zaman ve mekan kısıtlaması olmaksızın, öğrencinin istediği zamanda ve yerde eğitim almasıdır. Bu amaçla mühendislik ve teknik eğitimde uzaktan eğitim uygulamaları için sanal laboratuvar (Virtual Laboratory) kavramı ortaya çıkmıştır. Geliştirilen sanal laboratuvar ortamları kullanılarak benzetimler yapmak ve sonuçlarını görmek mümkün olmuştur. Mühendislik ve teknik eğitimde birçok mesleki dersler uygulamalıdır. Öğrencinin bilgi ve becerilerini geliştirmesi için laboratuvar çalışmalarında bulunması gerekmektedir. Uzaktan eğitim uygulamalarında öğrencinin laboratuvar çalışmalarını yapabilmesi için uzaktan erişimli laboratuvar (Remote Access Laboratory) kavramı gündeme gelmiştir. Diğer bir deyişle öğrenciler internet üzerinden herhangi bir yerde bulunan gerçek laboratuvara erişerek deneylerini gerçekleştirebilmekte, deney sonuçlarını grafiksel veya sayısal olarak alabilmektedir. Ayrıca bir kamera vasıtası ile de laboratuvar ortamını ve deney sonuçlarını eş zamanlı olarak görebilmektedirler. Bu gibi çalışmalar mühendislik ve teknik eğitimde uzaktan eğitim uygulamaları için bir yöntem olmuştur. Literatürde Elektrik Elektronik Mühendisliğinin çeşitli alanlarına uygulanan sanal ve uzaktan erişimli laboratuvar çalışmaları verilmiştir.

1.2. Tezin Amacı

Sanal ve uzaktan erişimli laboratuvarlar genellikle LabView, Netmeeting ve Java Appletleri gibi yazılımlar kullanılarak gerçekleştirilmiştir [8,38,40]. Ancak, bu ortamlar istemci bilgisayara çoğu zaman bazı programların yüklenmesini gerektirir. Ayrıca, paylaşım temelli olarak geliştirilen uzaktan erişimli laboratuvar ortamları güvenilir olmadığından istemci-sunucu mimarisi ile çalışan ortamlara ihtiyaç duyulur. MWS yazılımı, m-dosyaları aracılığı ile PC ve giriş/çıkış portlarını kullanarak web tabanlı ve gerçek zamanlı uygulamaların yapılabilmesi için uygun bir ortamdır. Literatürde, MWS kullanılarak veri toplama kartları aracılığı ile doğru akım motorlarının denetimine yönelik uzaktan erişimli laboratuvarlar hazırlanmıştır [13,15]. Ancak MWS, uzaktan erişimli olarak RTW' yi ve dolayısıyla bazı sayısal işaret işleyiciler (DSP) gibi donanımları desteklememektedir. Diğer taraftan, yüksek performanslı sürücü sistemlerin denetiminde, denetim algoritmalarının DSP ile gerçekleştirilmesi gereklidir. Bu nedenle, DSP ile elektrik motorlarının denetimi amacıyla uzaktan erişimli bir laboratuvar geliştirmek için ek konfigürasyonların yapılmasına, sunucu ile kullanıcı arasında iletişimi ve uzaktan DSP yazılımına otomatik erişimi sağlayan arayüzlerin geliştirilmesine ihtiyaç duyulur. Erişilebilen kaynaklarda DSP ile sürülen yüksek performanslı sürücü sistemleri için uzaktan erişimli bir laboratuvar ortamı ile karşılaşılmamıştır.

Bu tezde, DSP denetimli asenkron motorlar için sanal ve uzaktan erişimli bir laboratuvar geliştirilmiştir. İlk aşamada, asenkron motorun PI ve Sinirsel-Bulanık Ağlarla denetimi için sanal bir laboratuvar ortamı hazırlanmıştır. Benzetim çalışmalarının yapıldığı sanal laboratuvar ortamına web tabanlı erişim için MWS' dan yararlanılmış ve Matlab/Simulink dosyalarına uzaktan erişim için ek konfigürasyonlar yapılmıştır. Ayrıca, kullanıcının yaparak öğrenmesine katkı sağlamak için Delphi programlama dili kullanılarak, kullanıcının benzetim bağlantılarını kurabilmesini sağlayan bir arayüz hazırlanmış ve bu arayüz ActivexForm'lara dönüştürülerek web sayfalarına taşınmıştır. İkinci aşamada ise uzaktan erişimli bir laboratuvar ortamı geliştirilmiştir. Uzaktan erişimli deney setinde, PWM üretimi ile akım ve hız denetim algoritmalarını gerçekleştirmek için Matlab/Simulink uyumlu DS1104 denetleyici kartı kullanılmıştır. Motorun PI ve Sinirsel-Bulanık Ağlarla denetimi için gerekli uygulama dosyaları, Matlab RTW ve dSpace RTI blokları kullanılarak hazırlanmış ve ControlDesk Developer yazılımı ile bir grafiksel kullanıcı arayüzü tasarlanmıştır. Controldesk yazılımına uzaktan otomatik olarak erişebilmek için Python dili kullanılarak bir arayüz yazılımı hazırlanmıştır. Ayrıca, deneysel çalışmaların uzaktan gerçekleştirilebilmesi amacıyla istemci-sunucu iletişimini sağlayabilmek için Delphi programlama dili kullanılarak bir iletişim yazılımı geliştirilmiştir.

Kullanıcı etkileşimli arayüzler aracılığı ile gerek benzetim gerekse uzaktan erişimli deneysel ortamda, denetim parametrelerini değiştirebilmekte ve geliştirdiği denetim algoritmasını sisteme yükleyerek sistemin çalışma sonuçlarını grafiksel olarak yorumlayabilmektedir.

1.3. Tezin Bölümleri

Bu tez çalışması 6 bölümden oluşmaktadır. Tezin 1. bölümünde tez konusuyla ilgili genel bilgiler verilmiş ve literatür taraması sunulmuştur. Bölüm 2’de literatürdeki internet tabanlı sanal veya gerçek laboratuvar ortamları sunulmuş, sanal veya gerçek laboratuvarlar ve bu ortamlarının hazırlanmasında kullanılan yapılar açıklanmıştır. Ayrıca mühendislik ve teknik eğitimde kullanılan bu laboratuvar ortamlarının yararları ve sakıncaları üzerinde durulmuştur. Bölüm 3’de geliştirilen uzaktan erişimli DSP denetimli asenkron motor laboratuvarının sanal ve deneysel çalışma ortamlarında kullanılan donanım ve yazılım yapısından bahsedilmiştir. Bölüm 4’de internet tabanlı laboratuvarın benzetim ortamı ve uzaktan erişim ortamının gerçekleştirme aşamaları sunulmuş ve laboratuvarın sahip olduğu özellikler anlatılmıştır. Bölüm 5’de kullanıcı arayüzü, benzetim ve deneysel çalışmalardan elde edilen sonuçlar tartışılmıştır. Bölüm 6’da yapılan çalışmanın değerlendirilmesi sunulmuş ve bu tez çalışmasından sonra ileriye yönelik çalışmalara ilişkin önerilerde bulunulmuştur.

2. İNTERNET TABANLI LABORATUAR ORTAMLARI

Günümüzde teknolojinin hemen hemen her alanında ciddi bir şekilde köklü değişimler gerçekleşmektedir. Özellikle bilim ve teknolojiadaki anlık değişikliklerden eğitim sistemi de etkilenmektedir. Gerek eğitimde gerekse eğitimde karşılaşılan sorunların çözülmesinde bilgisayar hızla yaygınlaşan bir araç haline gelmiştir. Eğitimin daha etkili ve daha verimli hale getirilmesi için multimedya araçlarının kullanılması ve bilgisayarların eğitime dahil edilmesiyle Bilgisayar Destekli Eğitim önem kazanmaya başlamıştır. Bilgisayar Destekli Eğitim, eğitim alanında halen güncel olarak kullanılmaktadır.

Bilgisayar Destekli Eğitim ile eğitim alanında yerini alan bilgisayarlar, internetin gelişmesiyle uzaktan eğitimde de kullanılmaya başlanmıştır. İnternet teknolojisinin hızı, iş gücü, zaman ve mekandan tasarruf sağlaması, ölçme ve değerlendirmeyi kolaylaştırması gibi özelliklerinden dolayı internet uzaktan eğitimin vazgeçilmez bir unsuru olmuştur. Hem mühendislik hem de teknik eğitim kurumları teknoloji üretebilen mühendislerin ve teknik elemanların yetişmesine imkân sağlayacak laboratuarlara sahip olmalıdır. Mühendislik ve teknik eğitim programları, teorik derslerin yanında pratik eğitimin yapıldığı laboratuvarlarla istenilen seviyelere taşınabilir. Ancak üniversitelerimizde bulunan laboratuvar alt yapıları ve öğrenci sayısı göz önüne alındığında önemli sorunlarla karşılaşmaktadır. Bu sorunların giderilmesi için internet tabanlı laboratuvar ortamları geliştirilmiştir. Bu laboratuvar ortamlarına günün her saati uzaktan erişilerek çalışmalar yapılmasına olanak sağlanmaktadır. Literatürde, iki farklı internet tabanlı laboratuvar ortamı göze çarpmaktadır:

- Benzetim veya gösterilerle desteklenmiş sanal laboratuvarlar,
- Uzaktan erişimli deney veya ölçüm düzenekli gerçek laboratuvarlar.

Bu internet tabanlı laboratuvarlar öğrencilerin öğrenmesine katkıda bulunan, eldeki tüm kaynakları en iyi bir şekilde kullanmayı sağlayan, etkin ve etkileşimli ortamlardır. İnternet tabanlı laboratuvar ortamları, zaman ve yer esnekliği, grup ile öğrenme kolaylığı ve kaynaklara anında ulaşma gibi üstünlüklerden dolayı oldukça çekici hale gelmiştir.

2.1. Sanal ve Uzaktan Erişimli Laboratuvarlar

İnternetin eğitim alanındaki kullanım alanlarından biri, gösteri veya benzetimlerle desteklenmiş sanal laboratuvarlar hazırlamaktır. Sanal laboratuvarlar öğrencilere, anlaşılması zor konuların daha iyi anlaşılmasına yardım etmek ve soyut kavramların gösterilerle veya benzetimlerle desteklenmesi imkanını sağlamaktadır. Aynı zamanda teorik ve pratik uygulamaların birlikte yapıldığı etkili ve etkin ortamlar sunmaktadır. Teorik bilgilerin gerçek

problemlerin çözümünde nasıl uygulandığını değerlendirmek veya anlamak amacıyla benzetimler veya deneysel çalışmalar gerekmektedir.

Laboratuvar ortamında yapılan deneysel çalışmalar sınırlı kaynaklarla gerçekleştirildiğinden birtakım sorunlarla karşılaşmaktadır. İlk olarak iletken ortamlarda öğrencilerin deneylerini gerçekleştirmeleri amacıyla uygun fiziksel ortamlar geliştirilmelidir. Diğer güvenliğini sağlamak için teknik insan gücü kullanılarak donanımlar uygun yerlere yerleştirilmelidir. Son olarak da laboratuvarın maksimum kullanımını sağlamak için uygun çalışma programları oluşturulmalıdır. Bu sorunların çözülmesi son derece önemlidir.

Eğitim içeriklerindeki bazı konular belirli bir sırada anlatılmaktadır. Yani bazı konular için sınıf ortamları yeterli iken, belirli bazı konuların mutlaka laboratuvar ortamlarında gerçekleştirilmesi gerekmektedir. Bu durum arzu edilen ve bir gereklilik olmasına rağmen sınırlı mekan, zaman ve cihazlardan dolayı sıkıntılar yaşanmaktadır. Ayrıca laboratuvardaki personelin yetersizliği ve laboratuvar programlarının çok yoğun olması öğrencilerin eksikliklerini tamamlamasına izin vermemektedir. Deneyler esnasında öğrenciler laboratuvarı ya hiç ya da çok az kullanmış olmaktadır.

Bu sıkıntıların üstesinden gelmek için internet tabanlı sanal laboratuvarlar geliştirilmeye başlanmıştır. Sanal laboratuvarlar öğrencilerin internet üzerinden zaman ve mekandan bağımsız laboratuvardaki deneylerin benzetimlerine erişmesine izin vermektedir. Özellikle uygun bir şekilde tasarlanmış sanal laboratuvarlarda öğrenci gerçek bir laboratuvarda çalışıyormuş gibi çalışmalarını yapabilmekte ve diğer öğrencilere çalışmalarını ulaştırabilmektedir. Hatta daha da geliştirilmiş laboratuvarlarda internet üzerinden DAQ kartı gibi bir takım donanımsal araçlar kullanılarak deney veya ölçümler yapılabilir. Ayrıca sanal laboratuvarlara öğrencilerin dışında farklı yerlerdeki araştırmacılar da erişerek, aynı anda birçok araştırma gerçekleştirebilmekte ve tasarımlar işbirliği içerisinde tamamlanabilmektedir. Bu da deneylerin çeşitli aşamalarındaki hayati derecede önemli değişimlerinin incelenmesi veya gözlemlenmesini bile sağlamaktadır [85].

Uzaktan erişimli laboratuvarlar ise fiziksel dünya ile etkileşime müsaade eden ortamlardır. Uygun elektronik kontrol, bilgisayar, izleme sistemleri ve harici sistemler gibi donanımlar kullanılabilir. Bu sistemlerin birleşimi ve internetin iletişim şartları, uzaktaki öğrenciye, sanki fiziksel laboratuvardaymış gibi, fiziksel laboratuvarda bulunan araç ve gereçleri izlemek ve kontrol etmek için izin vermektedir.

Uzaktan erişimli laboratuvarların çalışma şekli şöyledir: internet bağlantısı olan bir kullanıcı, kendi (istemci) bilgisayarı üzerinde çalışan uygun bir şekilde tasarlanmış olan grafiksel arayüz kullanarak, uzak laboratuvardaki bilgisayarı ve çeşitli donanımları kullanabilmektir. Arayüz programı, incelenen sistemin dinamiklerindeki değişiklikleri veya bazı

cihazların ayarlarını yapabilmeli ve gerçek zamanlı olarak laboratuvar sonuçlarının resim ve video görüntülerini alabilmelidir. Deneysel çalışma sonucunda elde edilen veriler, bazı araç veya sensörlerle algılanıp daha sonraki çalışmalar için istemci bilgisayar üzerinde gösterebilmeli veya istemci bilgisayara indirilebilmelidir.

2.2. Sanal ve Uzaktan Erişimli Laboratuvarların Üstünlükleri ve Sakıncaları

Mühendislik ve teknik eğitimde internet tabanlı sanal laboratuvarların önemli bir yeri vardır. Eğitimde internet tabanlı laboratuvar kullanmanın öne çıkan önemli yararları şunlardır [17, 86–87]:

- Aynı dersi alan başka üniversitede veya başka fakülte okuyan öğrenciler arasındaki fark ortadan kalmış olur. Böylelikle eğitim kurumlarında bir öğrenme eşitliği ortaya çıkar. Sanal laboratuvar ortamlarında bütün öğrenciler tek bir yapıdan aynı şartlar altında öğrenme imkânına sahip olurlar.
- Öğrencinin öğrenmesi tamamen kendi kontrolü altındadır. Öğrenci kendi algılama hızı veya zekâ seviyesine paralel olarak benzetimlerini gerçekleştirir. Böylece, kendi kendine öğrenebilme bilinci kazanır.
- Öğrenci ihtiyaç duyduğu zaman ders notları ya da konuyla ilgili teorik bilgilere rahatlıkla ulaşabilir.
- Öğrenciye verilen eğitimin değerlendirilmesi ve gözlenmesi sanal laboratuvar ortamında ve merkezi olarak daha kolay ve daha kısa sürede yapılabilir.
- Öğretim elemanı dersi anlatırken, ilgili konunun benzetimini sanal laboratuvar ortamında göstererek, anlatılan teoriyi uygulamalı olarak görebilir. Bu sayede konunun pekişmesi daha kolay olur.
- Sanal laboratuvar ortamında verilen eğitimin içeriği, teknolojik gelişime paralel olarak her an değiştirilebilir ve kolaylıkla güncelleştirilebilir.
- Sanal laboratuvar ortamlarında deney benzetimlerini pekiştiren öğrenci, gerçek laboratuvar ortamında zorluk çekmeyecektir. Gerçek laboratuvar ortamında yapılacak uygulamanın aşamaları çabuk kavranır ve uygulamalar daha kısa sürede tamamlanır.
- Sanal ortamlarda yapılan uygulamalar eğitim süresi tamamlandıktan sonra da yardımcı bir araç olarak her zaman kullanılabilir. Bu da alınan eğitimin kalıcı olmasına, bilgilerin taze tutulmasına ve eğitimdeki devamlılığa yardımcı olur.
- Sanal ortamlarda yapılan uygulamalar ile daha emniyetli ortamlar oluşturulur ve muhtemel kazalar önlenmiş olur.

- Gerçek laboratuvar ortamında, öğrenci deneye katılamadığı zaman, genelde o deneyi bir daha yapma imkânı bulamamaktadır. Oysa sanal laboratuvar ortamlarında öğrenciler bir deneyin benzetimini gerçekleştirmek için zaman ve mekan kavramlarından tamamen bağımsızdır.
- Gerçek laboratuvar şartlarında yapılan deneylerde elde edilen sonuçlar ile teorik hesaplamalar arasında genelde çok az fark olur. Sanal laboratuvar ortamındaki benzetimlerden elde edilen sonuçlar, deney öncesi yapılan teorik hesaplamalara çok yakın ya da aynıdır.
- Gerçek laboratuvar ortamında deneyler kısıtlı bir zamanda yapılmak zorundadır. Deneyden sonra hala anlaşılmayan kısımlar, sanal laboratuvar ortamında deneyin tekrarı yapılarak kalıcı bir şekilde öğrenilebilir.
- Gerçek laboratuvar ortamında yapılan deneylerin grafiklerini çizmek için verileri yine bir bilgisayar ortamına aktararak, tablolar oluşturmak gerekmektedir. Sanal laboratuvar ortamında ise tüm veriler kayıt edilerek değişimleri gözlenebilir, zamana bağlı değişen grafikleri çizilebilir ve diğer bilgisayar ortamlarına gönderilebilir.
- Gerçek laboratuvar ortamları çoğunlukla kararlı durumda ölçüm yapabilecek donanımlarla donatılmıştır. Dolayısıyla sanal laboratuvar ortamlarında dinamik değişimlerin izlenebilmesi ve etkileşimli benzetim modellerinin oluşturulması öğrenmeye önemli katkı sağlayacaktır.
- Gerçek laboratuvar ortamında yapılması emniyet açısından sakıncalı olan uygulamalar sanal laboratuvar ortamlarında rahatlıkla yapılabilir ve sonuçları gözlemlenebilir. Bu sayede deneyi yapan öğrenciye ve cihazlara zarar verecek olan uygulamalar, sanal laboratuvar ortamları sayesinde engellenmiş olur.
- Gerçek laboratuvar uygulamalarında deney şartları ideal olmadığından deney sonucu olumsuz etkilenebilir. Sanal laboratuvar ortamından tüm şartlar ideal olarak kabul edilir ve istenmeyen herhangi bir olumsuzluk ile karşılaşmaz.
- Gerçek laboratuvar ortamında, maddi ve yer sıkıntısından dolayı bulunmayan araçların sanal laboratuvar ortamlarında uygulamaları yapılabilir. Bu şekilde öğrenciler gerçek laboratuvar ortamında göremedikleri araçlar hakkında bilgi sahibi olurlar ve dünyadaki eğitimi yakından takip edebilirler.

Sanal laboratuvar ortamların tek başına kullanılmasının bazı sakıncaları bulunmaktadır. Bu sakıncalar şunlardır [87],

- Mühendislik eğitiminde mesleki derslerin içerikleri belirli bir sıra ile verilmektedir. Sanal ortam hazırlanırken bu sıraya uyulsa da kullanıcı ders konularını çok çabuk tamamlamak isteyebilir ve bu konuların işleyiş sıralarının bozulmasına neden olabilir.

Konuların yeterince kavranılmadan geçilmesi, yanlış öğrenmeye veya uygulamalar üzerinde yanlış yorumlar yapılmasına neden olabilir.

- Mühendislik eğitiminde derslerde verilen teorik bilgi ve laboratuvar uygulamaları eş zamanlı sürdürülmektedir. Öğrenciler aldıkları teorik bilgileri laboratuvar uygulamalarında destekleyerek hayata geçirilir. Öğrenciler karşılaştığı problemleri, öğretim elamanına sorarak anında öğrenebilirler. Her ne kadar sanal laboratuvar ortamında bu soruların cevaplarını bulmak mümkün olsa da kullanıcı tarafından yeterince anlaşılmayabilir.
- Sanal laboratuvar ortamında öğrenciler ne kadar etkileşim içerisinde olursa olsunlar, yüz yüze eğitimde olduğu gibi bire bir kaynaşma olmayacaktır. Bu durum konu hakkında daha tecrübeli olan öğretim elemanlarının bilgilerinden yararlanma imkânını ortadan kaldırabilir.
- Mühendislik eğitiminin hemen hemen her kademesinde matematik bilgisi gerekmektedir. Klasik eğitimde ders sırasında verilen matematiksel kavramlar uzman öğretici tarafından anlaşılabilir yapıda verilmektedir. Sanal laboratuvar ortamında bu kavramlar çok iyi verilse bile öğrenci tarafından anlaşılması zor olabilir.
- Mühendislik eğitiminde yaparak öğrenme esastır. Her şeyden önce öğrencilerin yaparak öğrenmeleri kısıtlanabilir ve gerçek laboratuvar ortamında öğrencinin kazanacağı beceri, deneyim ve tecrübelerden yoksun kalabilir.

Yukarıda sıralanan sakıncalardan dolayı mühendislik ve teknik eğitim öğrencilerinin derslerinin mutlaka gerçek laboratuvarlarla desteklenmesi gerekmektedir. Bunun için öğrencileri ya fiziki laboratuvar mekanlarına götürerek ya da fiziki laboratuvar mekanlarına internet üzerinden erişerek mümkün olmaktadır.

Uzaktan erişimli laboratuvarların sağladığı yararlar ise şunlardır [88]:

- Öğrenciler daha ileri düzeyde deneyler gerçekleştirebilir.
- Öğrencinin kendi kendine öğrenmesine olanak sağlar ve kendisini değerlendirmesine yardımcı olur.
- Öğrenci deneylerini gerçekleştirmek için zaman ve mekandan bağımsızdır. Dolayısıyla öğrenci daha kısa sürede, daha çok deney yapabilir.
- Öğretim elemanı deney ile ilgili teorik bilgiyi anlatırken, anında deneyi gerçekleştirerek, öğrenmeyi pekiştirmeyi sağlayabilir,
- Öğretim elemanlarının yeni deneylerini oluşturabilmeleri için fırsatlar ve ortamlar sunar.
- Uzaktan deney sayesinde diğer üniversitelerde bulunan öğrencilerin de deney yapmaları sağlanmış olur.

- Uzaktan deney imkanları ile eğitim kurumunun diğer eğitim kurumları arasındaki eğitim hizmeti ve kalitesi ortaya çıkar.
- Eğitim kurumunun, laboratuvarlar için çok sayıda yetişmiş elemana ihtiyacı kalmaz.
- Uzaktan yapılan deneyler ile daha emniyetli ortamlar oluşturulur ve muhtemel kazalar önlenmiş olur.
- Mühendislik eğitiminde yaparak öğrenme esastır. Gerçek laboratuvar ortamları yaparak öğrenmeyi desteklemektedir.
- Gerçek laboratuvar ortamlarının hazırlanması için yüksek yatırımlar gerektirmektedir. Bu yatırımların, sadece belirli bir zaman aralığında değil sürekli olarak öğrencilerin kullanımına sunulması sağlanır.

2.3. Sanal veya Uzaktan Erişimli Laboratuvar Yapıları

Yapılan çalışmalar incelendiğinde, oluşturulan sanal veya uzaktan erişimli laboratuvar yapıları aşağıdaki şekilde özetlenmiştir:

- Paylaşım temelli programlar kullanan sanal veya uzaktan erişimli laboratuvar yapıları [38, 46],
- İstemci-sunucu ilişkisi olan sanal veya uzaktan erişimli laboratuvar yapıları [39, 40, 58, 61],
- Yerel kullanıcılara hitap eden sanal laboratuvar yapıları [69, 74].

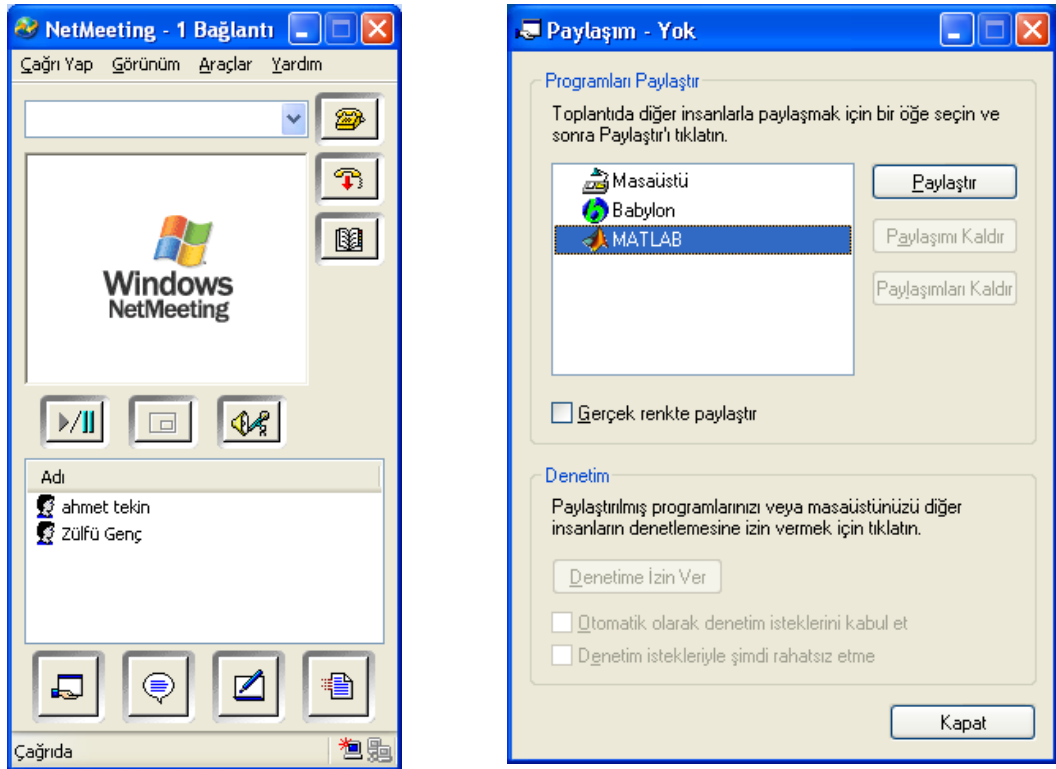
2.3.1. Paylaşım Temelli Programlar Kullanan Laboratuvar Yapıları

Microsoft NetMeeting gibi paylaşım temelli programlar ile internet üzerinden sesli ve görüntülü iletişim kurulabilmekte, birden fazla bilgisayar ile toplantı yapılabilmekte ve karşıdaki bilgisayarda bulunan programların kullanılmasına izin verilebilmektedir. Microsoft XP işletim sistemi yazılımı ile birlikte bilgisayara kurulan Microsoft NetMeeting programını çalıştırmak için Başlat/Çalıştır seçeneklerini seçerek Conf komutunu yazmak yeterlidir. Şekil 2.1’de görüldüğü gibi kullanıcılar arasında bağlantı kurulduktan sonra, kullanıcıların bilgisayarında bulunan programlar paylaştırılabilmektedir.

Arlington Texas Üniversitesinde denetim deneylerini gerçekleştirmek amacıyla, Microsoft NetMeeting programını kullanarak Matlab’ın gerçek zamanlı araçları internet üzerinden paylaştırılmış, basit ve ucuz bir yöntemle uzaktan erişilebilir duruma getirilmiştir. Öğrencilerin Java gibi yüksek seviyeli ortamlarda kod yazması yerine denetim algoritmalarına odaklanması amaçlanmıştır. Ücretsiz olarak temin edilebilen yazılımları kullanarak, eğitim

amaçlı laboratuarlardaki donanımların uzaktan denetlenebileceği gösterilmiştir [38]. Burada bir deneysel çalışma mevcuttur. Ancak Matlab programının paylaştırılması ile aynı zamanda sanal laboratuvar yapılarında kullanılan benzetim ve animasyonlara da fırsat verilmiştir. Diğer bir ifadeyle kullanıcı karşıdaki bilgisayara bağlanarak hem deneysel hem de benzetim çalışmaları yapabilmektedir.

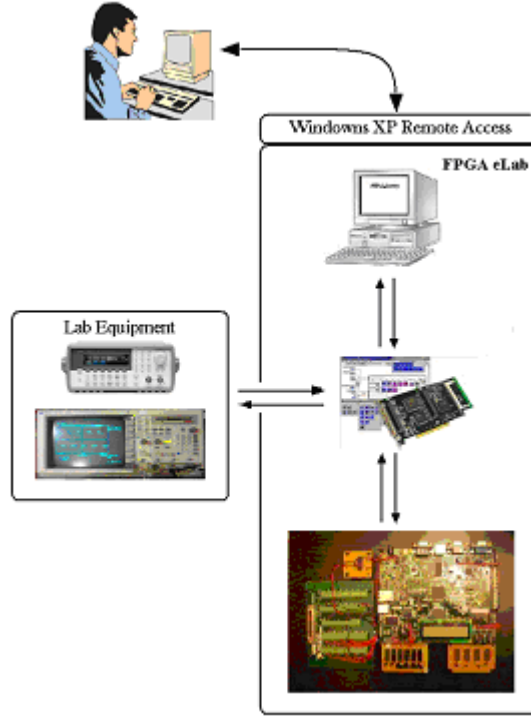
Bu gibi paylaşım temelli programları kullanmanın birinci sakıncası, her iki bilgisayarda da paylaşım programlarının yüklenmiş olması gerekmektedir. Sanal laboratuvar ortamları hazırlanırken istemci tarafında, minimum yazılım veya minimum donanım ihtiyacının olması arzu edilmektedir. İkinci sakıncası ise paylaştırılan programlar hiçbir kısıtlama olmaksızın tam yetki ile kullanılmaktadır. Bu durumda karşıdaki bilgisayara istemeden de olsa zarar verilebilmektedir.



Şekil 2.1 Microsoft NetMeeting penceresinin görünümü

Paylaşım temelli program kullanan başka bir çalışma Hashemian ve Riddley [46] tarafından yapılmıştır. Şekil 2.2’de uzaktan erişimli laboratuvarın yapısı görülmektedir. İnternet erişimi olan herhangi bir yerden, öğrencilerin laboratuvar deneylerini gerçekleştirmeleri için uzaktan erişimli bir laboratuvar sunulmuştur. Laboratuvar ortamı dijital devre tasarımlarında kullanılan FPGA platformunu kullanarak, dijital devre tasarım derslerine yardımcı olmaktadır. İnternet erişimli laboratuvarın arayüzü Labview yazılımı ile hazırlanmış ve DAQ kartı, FPGA

kartı, osilaskop, dalga generatörü cihazları da donanım yapısında kullanılmıştır. Microsoft Windows Xp'nin uzak masaüstü bağlantısı kullanılarak, laboratuara uzaktan tam erişim sağlanmaktadır. Bu yapıdaki en büyük sorun, uzak masaüstü bağlantısı ile bağlanılan bilgisayar üzerinde sınırsız yetki verilmesidir. Bunun anlamı bağlanılan bilgisayarda deneyin yanı sıra istenilen her işlemin yapılabilmesidir. Bu durum uzaktaki kullanıcının hem bilgisayara hem de donanımlara zarar verebileceği riski taşımaktadır.



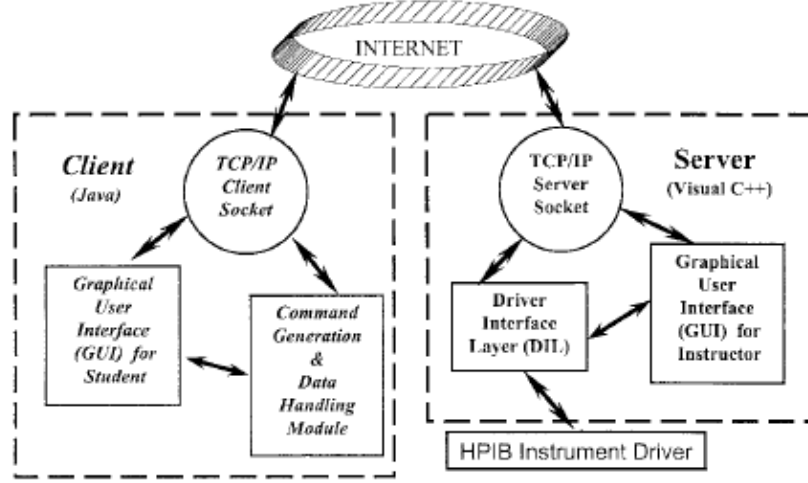
Şekil 2.2 Uzaktan erişimli laboratuvarın yapısı [46]

2.3.2. İstemci - Sunucu İlişkisi Laboratuar Yapıları

İstemci-sunucu ilişkisi olan laboratuar yapılarında bilgisayarlar arasındaki haberleşme TCP/IP protokolü üzerinden gerçekleştirilmektedir. Bir programlama dili kullanılarak istemci ve sunucu arayüzleri hazırlanmaktadır. Sunucu tarafındaki bilgisayara bağlı donanımlara erişilmekte ve deneyler yapılmaktadır. Deney sonuçları istemci bilgisayar tarafından grafiksel veya sayısal olarak görülebilmektedir.

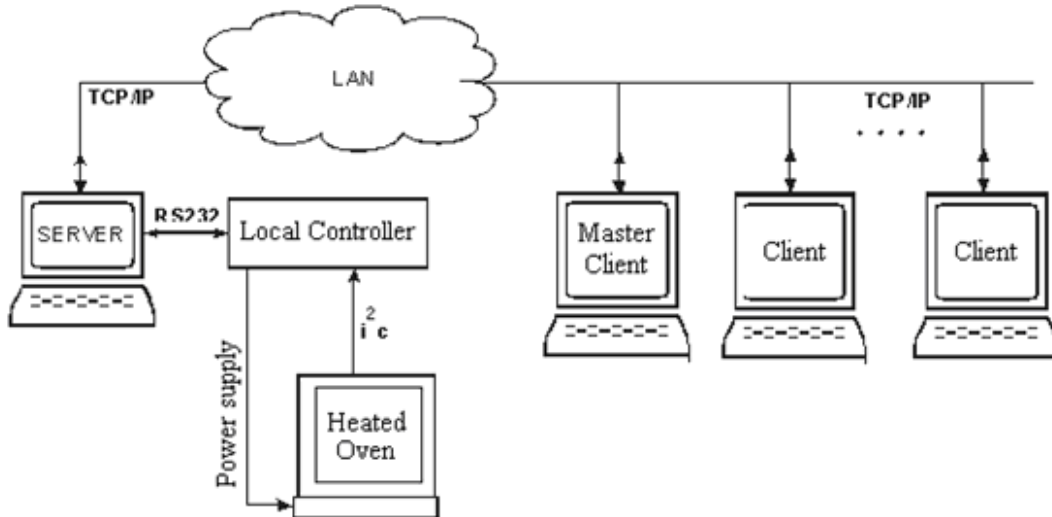
Shen ve diğ. [39], AIM-LAB olarak adlandırılan, internet tabanlı ve etkileşimli bir laboratuar ortamı gerçekleştirmişlerdir. AIM-LAB ile laboratuar ölçüm cihazlarının özellikle uzaktan eğitimde etkili bir şekilde kullanılabilmesi sağlanmıştır. Yarı-iletken bir cihazın karakteristik değerlerinin gözlemlenmesine ilişkin bir uygulama gerçekleştirilmiştir. AIM-LAB'da kullanılan GPIB sürücü kartı C++ dilini desteklemesinden dolayı sunucu yazılımı

Visual C++ tabanlı, istemci yazılımı ise Java tabanlı yazılmıştır. AIM-LAB sistem konfigürasyonu Şekil 2.3'te verilmiştir.



Şekil 2.3 AIM-LAB sistem konfigürasyonu [39]

Oltean ve diğ. [40] tarafından, uzaktan fırın sıcaklığının elektriksel ısı denetimi ve izlenmesi ile ilgili bir çalışma sunulmuştur. PIC16F628 mikroişlemcisi kullanılarak fırın sıcaklığının elektriksel ısı denetimi gerçekleştirilmiştir. İstemci ve sunucu arayüzleri Labview/CVI ortamında gerçekleştirilmiş ve yerel dijital denetimcinin parametre değişimine izin verilmiştir. Şekil 2.4'te uzaktan izleme ve denetim uygulamalarının blok diyagramı verilmiştir.

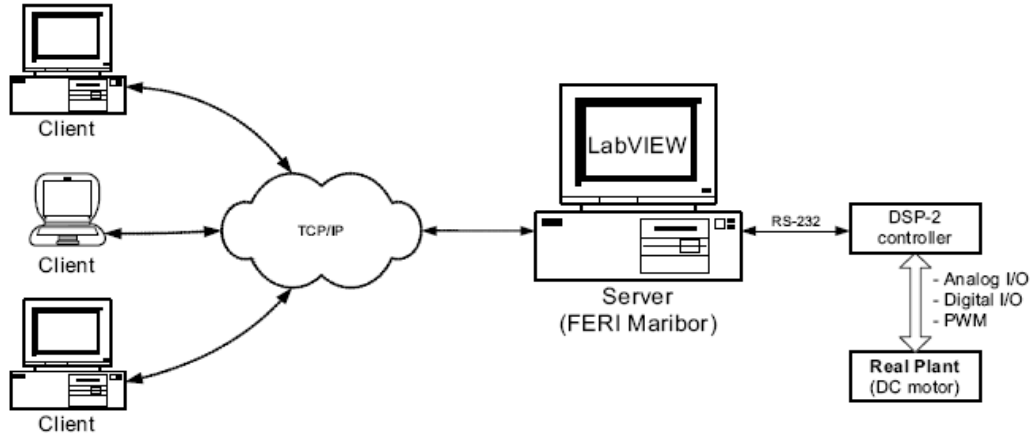


Şekil 2.4 Uzaktan izleme ve denetim uygulamalarının blok diyagramı [40]

Sistem modelleme, tasarım ve analizlerinin daha hızlı geliştirilebildiği Matlab, Labview gibi yazılımları kullanan sanal veya uzaktan erişimli laboratuvar yapıları da yine istemci –

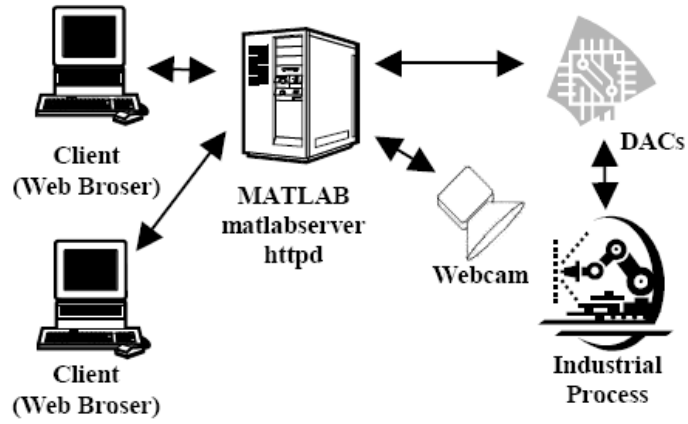
sunucu mimarisi üzerinde çalışmaktadır. MATLAB yüksek seviyeli programlama dili olmasının yanında algoritma geliştirme, verilerin görselleştirilmesi, veri analizi ve sayısal hesaplamalar için etkileşimli bir yazılım paketidir. Yine Labview programı da veri toplama ve denetiminde, veri analizi ve veri sunumunda kullanılan, G programlama dili tabanlı bir grafiksel program geliştirme ortamıdır. Literatürde bu programlama ortamları kullanılarak birçok internet tabanlı sanal veya uzaktan erişimli gerçek laboratuvar ortamları hazırlanmıştır.

Hercog ve diğ. [58] tarafından yapılan çalışmada Maribor Üniversitesi Elektrik Mühendisliği ve Bilgisayar Bilimleri Fakültesinde, elektrik sürücüler için Şekil 2.5'te gösterilen uzaktan erişimli bir laboratuvar tasarımı geliştirilmiştir. Bir DA motorunun denetimi ile ilgili deneyler bulunmaktadır. Tanımlanan uzaktan erişimli laboratuvar, motor denetimi için özel donanımlardan, Matlab ve Labview gibi iki ticari yazılım paketinden oluşmaktadır. Denetim algoritmalarının geliştirilmesi amacıyla Matlab/Simulink yazılımı, grafiksel arayüz ve uzaktan yapılacak işlemler için Labview yazılımı kullanılmıştır. İstemci bilgisayar tarafında Labview Run Time Engine yazılımının yüklenmiş olması gerekliliği bu yapının bir dezavantajıdır.



Şekil 2.5 Uzaktan erişimin blok şeması [58]

Diez ve diğ. [61] ise daha çok Sistem Mühendisliği ve Kontrol Mühendisliği eğitiminde kullanılmak üzere, MWS ortamını kullanarak internet tabanlı sanal ve uzaktan erişimli gerçek denetim laboratuvarı hazırlamışlardır. Sanal laboratuvar ortamında IIR filtre tasarımı ve denetim tasarımı ile ilgili benzetimler gerçekleştirilmektedir. Uzaktan erişimli laboratuvar ortamında ise Şekil 2.6'dan da görüldüğü gibi veri toplama kartı (DAC) kullanılarak bir DA motorunun denetimi gerçekleştirilmiştir. Ayrıca laboratuvar ortamı bir kamera vasıtası ile izlenmiştir.

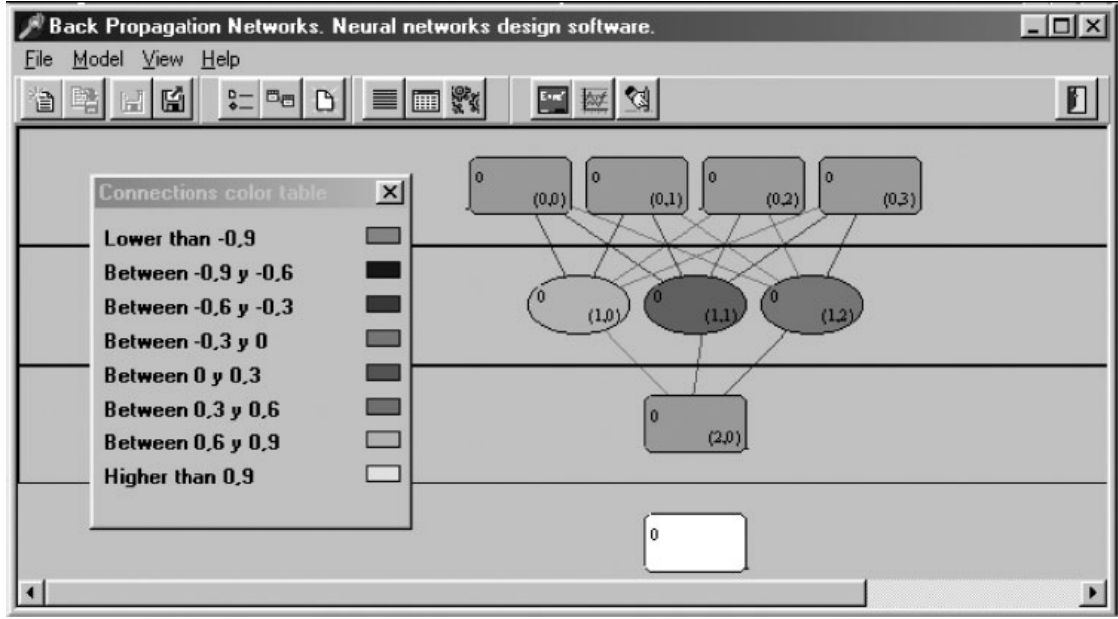


Şekil 2.6 Uzaktan erişimli laboratuvarın donanım konfigürasyonu [61]

2.3.3. Yerel Kullanıcılara Hitap Eden Laboratuvar Yapıları

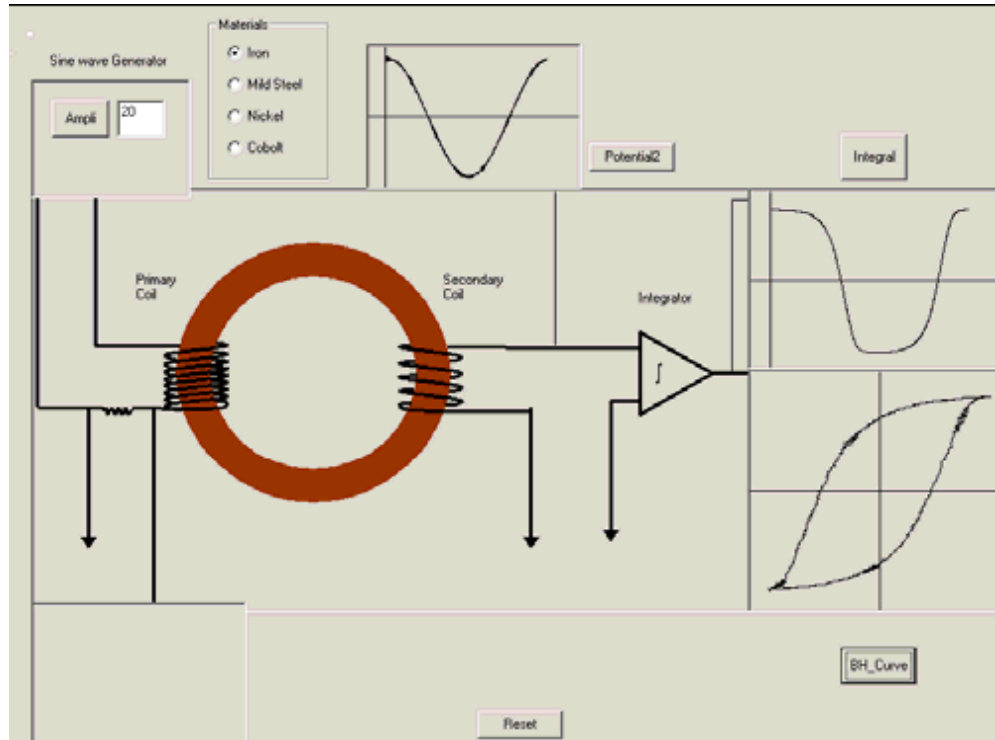
Bu tür laboratuvar ortamları Delphi, Visual Basic veya C++ Builder gibi görsel programlama dilleri kullanılarak, benzetim amaçlı eğitim araçları hazırlanmaktadır. Bu eğitim araçları internet tabanlı olmayıp çalıştırılabilir (exe uzantılı) dosyalardır. Kullanıcı bu çalıştırılabilir dosyaları bilgisayarına kurarak benzetimlerini gerçekleştirebilmektedir. İnternet tabanlı olmamaları bu yapıların önemli bir dezavantajıdır.

Rosello ve diğ [69] İspanya Vigo Üniversitesinde Bilgisayar Mühendisliği öğrencileri için temel yapay sinir ağları kavramlarını öğretmek üzere Neuro-Lab isminde bir benzetim yazılımı gerçekleştirmişlerdir. Kullanımı kolay olan, öğrenme ve araştırma aktivitelerini desteklemek hazırlanan yazılım bu ortamı nesneye dayalı programlama teknikleri ile hazırlanmıştır. Şekil 2.7’de Neuro-Lab ortamında üç katmanlı yapay sinir ağı yapısı arayüzü verilmiştir.



Şekil 2.7 Neuro-Lab ortamında üç katmanlı yapay sinir ağı yapısı [69]

Mukhopadhyay ve diğ. [74] tarafından yapılan çalışmada ise Yeni Zelanda Massey Üniversitesi Bilgi ve Telekomünikasyon Mühendisliği Bölümünde anlatılan Elektromanyetik dersleri için hazırlanmış bir benzetim ortamı sunulmuştur. Öğrencilerin gerçek deneyleri yapmadan önce kullandıkları benzetim ortamı Visual Basic programlama dilinde tasarlanmıştır. Şekil 2.8’de benzetim ortamındaki B-H eğrisi örneği verilmiştir.



Şekil 2.8 Benzetim ortamında B-H eğrisi [74]

3.DSP DENETİMLİ ASENKRON MOTORLAR İÇİN UZAKTAN ERİŞİMLİ LABORATUAR

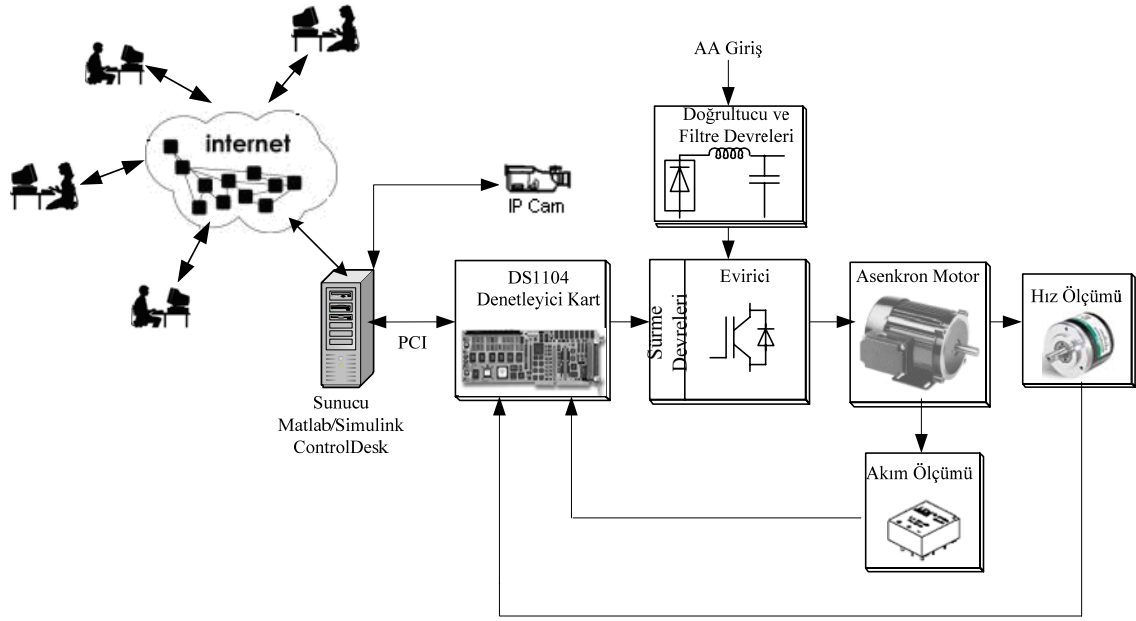
Geliştirilen internet tabanlı denetim laboratuvarında, benzetimlerle desteklenmiş sanal laboratuvar ortamı ve deneysel çalışma ortamı bulunmaktadır. Bu bölümde benzetim ve deneysel laboratuvar ortamını geliştirmede kullanılan donanım ve yazılım altyapısı hakkında bilgiler verilmiştir.

Benzetim ortamı hazırlanırken Active Server Page (ASP), Hyper Text Markup Language (HTML) dilleri kullanılarak dinamik web sayfaları hazırlanmıştır. Sanal laboratuvar ortamında benzetim çalışmalarının yapılabilmesi için MWS ortamı kullanılmıştır. Sunucu bilgisayara yüklenmiş olan Matlab programını çalıştıran Windows Xp gibi bir sunucu yazılımının da olması gerekmektedir. Ayrıca veritabanı işlemleri için Microsoft Office Access kullanılmıştır.

Deneysel çalışma ortamında bir sunucu bilgisayar, sunucu bilgisayarın ana kartına takılmış DS1104 sayısal işaret işlemci kartı ve IP kamera bulunmaktadır. Python programlama dili kullanılarak DS1104 kartına erişilmektedir. Deneysel çalışma ortamının hazırlanmasında MWS yetersiz kaldığı için Delphi programlama dili kullanılarak sunucu ve istemci arayüzleri hazırlanmıştır. Hazırlanan arayüzler sayesinde istemci bilgisayar ile sunucu bilgisayar arasında iletişim kurulmakta ve deney düzeneğinde deneyler gerçekleştirilmektedir.

3.1. Donanım Yapısı

Hazırlanan uzaktan erişimli kontrol laboratuvarının yapısı Şekil 3.1’de verilmiştir. Şekil 3.1’de gösterildiği gibi bir istemci bilgisayar, DS1104 denetleyici kartı takılı ve bir asenkron motorun bağlı bulunduğu sunucu bilgisayara internet üzerinden erişebilmektedir. İstemci bilgisayar isteklerini sunucu bilgisayara internet üzerinden göndermekte ve sunucu bilgisayar da bu istekler doğrultusunda kullanıcıların deneylerini gerçekleştirmeleri için hizmet vermektedir. Ayrıca laboratuvar ortamı bir IP kamera vasıtası ile izlenebilmektedir. Sunucu bilgisayar, DS1104 denetleyici kartı, Asenkron motor ve IP kameranın teknik özellikleri EK-1 de verilmiştir.

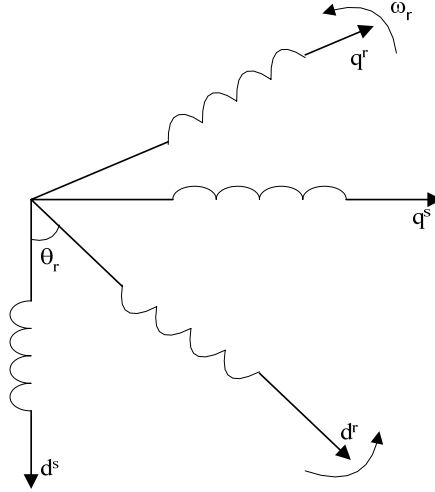


Şekil 3.1 İnternet tabanlı deney düzeneğinin genel görünümü

3.1.1 Asenkron Motorların Modellenmesi

Asenkron motorun hız denetimi uygulamalarında motorun dinamik modeli kullanılmaktadır. Dinamik model, eksen dönüşümleri kullanılarak vektör denklemleri ile elde edilir ve bu denklemler denetim yapılarında kullanılır. Bu bölümde çalışmada kullanılan eksen dönüşümü, dinamik denklemlerin elde edilmesi ve vektör denetim yöntemi verilmiştir.

Asenkron motorların dinamik devre modelleri, doğrusal olmayan ve zamanla değişen diferansiyel denklemler ile tanımlanabilir. Ancak bu denklemler devre modelinin çözümünü karmaşık ve zor hale getirir. Bu nedenle, üç fazlı asenkron motorların dinamik analizinde ve denetiminde iki eksen teorisi ile elde edilen iki faza dönüştürülmüş durum uzay modeli kullanılır. Üç fazlı bir asenkron motorun iki faz eşdeğeri Şekil 3.2'deki gibi gösterilebilir. Burada d^s , q^s stator d ve q eksenlerine, d^r - q^r ise rotor d ve q eksenlerine karşılık gelir. Eksen dönüşümleri sonucunda AA motorların dinamik davranışının modellendiği vektör dönüşümleri ortaya çıkmış ve yüksek başarılı hız sürücülerinin denetiminde kullanılan vektör denetim yöntemlerinin temelini oluşturmuştur [89].



Şekil 3.2 Üç fazlı motorun eşdeğer iki faz vektörleri

Asenkron motorların modellenmesi için dönen (rotor) referans çatı, duran referans çatı ve senkron dönen referans çatı dinamik modelleri kullanılmaktadır. Bu çalışmada asenkron motorun duran referans çatı dinamik modeli kullanılmıştır.

Asenkron Motorun Duran Referans Çatı Dinamik Modeli

Üç fazlı asenkron motor için duran referans çatıdaki gerilim denklemleri,

$$v_{qs}^s = R_s i_{qs}^s + \frac{d}{dt} \psi_{qs}^s \quad (3.1)$$

$$v_{ds}^s = R_s i_{ds}^s + \frac{d}{dt} \psi_{ds}^s \quad (3.2)$$

Senkron referans çatıda ω_r açısal hızıyla dönen rotora ilişkin rotor gerilim denklemleri ise benzer şekilde,

$$v_{qr}^e = R_r i_{qr}^e + \frac{d}{dt} \psi_{qr}^e + (\omega_e - \omega_r) \psi_{dr}^e \quad (3.3)$$

$$v_{dr}^e = R_r i_{dr}^e + \frac{d}{dt} \psi_{dr}^e - (\omega_e - \omega_r) \psi_{qr}^e \quad (3.4)$$

olarak elde edilebilir. Duran çatıda $\omega_e=0$ ve sincap kafesli bir asenkron motor için $v_{qr}^e = 0$ ve $v_{dr}^e = 0$ olduğundan yukarıda verilen senkron çatı denklemlerinde yerine yazılırsa,

$$v_{qr}^s = 0 = R_r i_{qr}^s + \frac{d}{dt} \psi_{qr}^s - \omega \psi_{dr}^s \quad (3.5)$$

$$v_{dr}^s = 0 = R_r i_{dr}^s + \frac{d}{dt} \psi_{dr}^s + \omega \psi_{qr}^s \quad (3.6)$$

olur. Motorun ürettiği moment,

$$\begin{aligned}
T_e &= \frac{3}{2} \frac{p}{2} \left(\psi_{qr}^s i_{dr}^s - \psi_{dr}^s i_{qr}^s \right) \\
&= \frac{3}{2} \frac{p}{2} \left(\psi_{ds}^s i_{qs}^s - \psi_{qs}^s i_{ds}^s \right) \\
&= \frac{3}{2} \frac{p}{2} L_m \left(i_{dr}^s i_{qs}^s - i_{qr}^s i_{ds}^s \right)
\end{aligned} \tag{3.7}$$

şeklinde elde edilir. Motor momentinin motorun mekanik devresine uygulanmasıyla rotoru ω_r açısal hızıyla dönen motorun üreteceği elektriksel moment,

$$T_e = T_L + J \frac{d}{dt} \omega + B\omega \tag{3.8}$$

bağıntısıyla bulunur. Burada T_L : yük momenti, J : eylemsizlik sabiti, B : sürtünme katsayısı ve ω : mekanik hızdır.

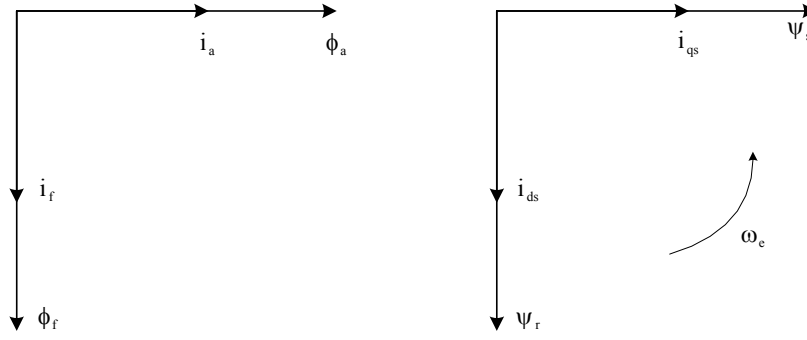
3.1.2 Asenkron Motorların Vektör Denetiminde Alan Yönlendirme Yöntemleri

Asenkron motorların, stator ve rotor akımları eksen dönüşümü ile birbirine dik duruma getirildikten sonra bu bileşenlerin yönlendirilmesi ile vektör denetimi gerçekleştirilir. Eksen dönüşümü sayesinde zamanla değişen parametre sorunu, alan yönlendirmesi ile de motorun akım ve akı vektörleri arasındaki kenetleme etkisi kaldırılabilir. Böylece bir asenkron motorun hem geçici hem de sürekli durumda yüksek başarılı hız denetimi yapılabilir [90-91].

AA motorlar eksen dönüşümü ile serbest uyarımlı DA motoru gibi denetlenmeye çalışılmaktadır. Serbest uyarımlı DA motorlarında endüvi ve alan sargıları birbirine doğal olarak diktir ve bu sargılarda üretilen manyetomotor kuvvetleri de birbirinden bağımsızdır. Serbest uyarımlı bir DA motorda alan doyumu ve endüvi reaksiyonu ihmal edilirse motorun üreteceği moment aşağıdaki gibi ifade edilir.

$$T_e = k_t \phi_f \phi_a = k_t' i_f i_a \tag{3.9}$$

Burada k_t' : moment sabiti, i_f : alan akımı ve i_a : endüvi akımıdır. Bu denklemden de görüleceği gibi uyarım akısı/akımı sabit tutularak DA motorun üreteceği moment, sadece endüvi akımıyla denetlenebilir. Eksen dönüşümü ile bir birine dik olarak elde edilen d ve q bileşenleri, serbest uyarımlı DA motorunun sırasıyla uyarım ve endüvi akımlarına karşılık gelir. Şekil 3.3 (a) ve (b)' de serbest uyarımlı DA motorun akım-akı arasındaki ve asenkron motorun q-d akımları arasındaki vektörel ilişki görülmektedir. Şekilden görüldüğü gibi, serbest uyarımlı DA motorunun moment açısı 90° dir ve motor momentini i_a akımının denetimi, akı ise i_f uyarım akımının denetimiyle birbirinden bağımsız olarak yapılabilir.



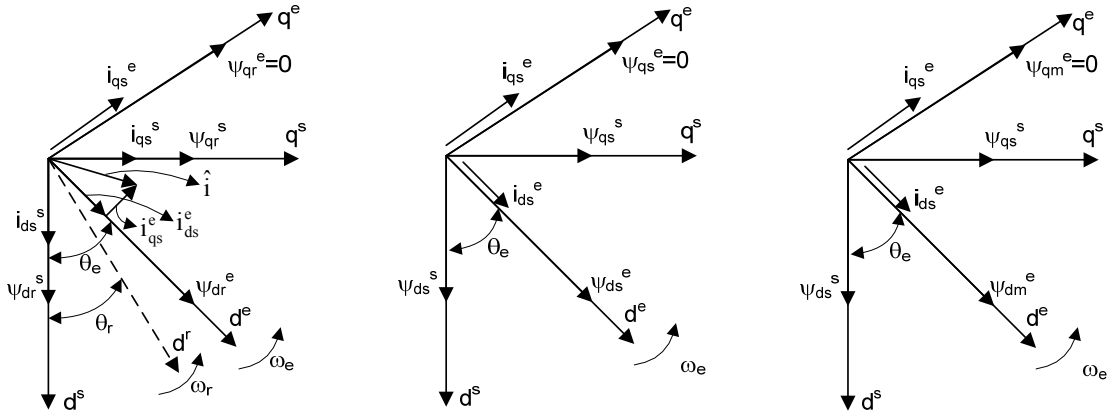
Şekil 3.3 DA ve AA motorların vektör diyagramları, (a) Serbest uyarımlı DA motoru, (b) Vektör denetimli asenkron motor

Asenkron motorlarda moment ve akının bağımsız denetlenebilmesi için, DA büyüklüklerine sahip senkron çatıdaki vektör denklemleri kullanılır. Bu amaçla asenkron motorun senkron çatı dinamik denklemlerindeki motorun i_{ds}^e akımının, alan akısıyla çakıştırılması gereklidir. Senkron çatıdaki motorun stator d-eksen akımı i_{ds}^e ; rotor alanı, stator alanı veya hava aralığı alanı ile çakıştırılabilir ve bu da vektör denetiminin alan yönlendirme şeklini belirler [92]. Şekil 3.3 (b)' de görüldüğü gibi asenkron motorun i_{ds}^e akımı, rotor alanıyla çakıştırılırsa asenkron motorun moment denklemi,

$$T_e = k_t \psi_r i_{qs} = k_t' i_{ds}^e i_{qs}^e \quad (3.10)$$

olur. Bu durumda asenkron motorun alan yönlendirmesi ile elde edilen moment denklemi, denklem (3.9) ile verilen DA motorunun moment denklemiyle eşdeğerdir. Bu denkleme göre bir asenkron motorda i_{ds}^e akımı sabit tutularak, i_{qs}^e akımının denetimiyle motor momenti denetlenebilir ve i_{qs}^e akımı sabit tutularak i_{ds}^e akımının denetimiyle motor akısının denetimi mümkündür.

Yukarıda da belirtildiği gibi, asenkron motorun denetiminde ortaya çıkan kenetleme sorununun giderilmesi alan yönlendirmesi ile mümkündür [90]. Alan yönlendirme işlemi; senkron hızda dönen motorun rotor, stator veya hava aralığı alanlarından birinin, d-eksenine yönlendirilmesiyle gerçekleştirilir. Şekil 3.4'de rotor, stator ve hava aralığı alan yönlendirmelerine ilişkin vektör diyagramları verilmiştir [89]. Bu çalışmada alan yönlendirme yöntemi olarak rotor alan yönlendirmesi kullanılmıştır.



Şekil 3.4 Alan yönlendirme vektör diyagramları, (a) Rotor alan yönlendirmesi, (b) Stator alan yönlendirmesi, (c) Hava aralığı alan yönlendirmesi

Rotor Alan Yönlendirmesi

Rotor alan yönlendirmesi, sinüsoidal olarak uyarılmış ve senkron hızda dönen bir asenkron motorun, senkron çatı d-ekseniyle rotor alanı d- eksen akısının çakıştırılmasını ifade eder. Şekil 3.4(a)' da rotor alan yönlendirmesinin vektör diyagramı verilmiştir. Bu şekilden de görüleceği gibi rotor alanının q-bileşeni $\psi_{qr}^e = 0$ olacaktır. Senkron referans çatıdaki rotor halkalanma denklemleri,

$$\psi_{qr}^e = L_m i_{qs}^e + (L_{lr} + L_m) i_{qr}^e \quad (3.11)$$

$$\psi_{dr}^e = L_m i_{ds}^e + (L_{lr} + L_m) i_{dr}^e \quad (3.12)$$

olarak oluşturulabilir. Bu denklemlerden i_{qr}^e ve i_{dr}^e akımları aşağıdaki gibi elde edilir [89-90, 93].

$$i_{qr}^e = \frac{1}{L_r} \psi_{qr}^e - \frac{L_m}{L_r} i_{qs}^e \quad (3.13)$$

$$i_{dr}^e = \frac{1}{L_r} \psi_{dr}^e - \frac{L_m}{L_r} i_{ds}^e \quad (3.14)$$

Rotor akısının yönlendirilmesiyle $\psi_{qr}^e = 0$ olduğundan denklem (3.13) ve (3.8) yeniden düzenlenirse,

$$i_{qr}^e = -\frac{L_m}{L_r} i_{qs}^e \quad (3.15)$$

$$T_e = -\frac{3}{2} \frac{p}{2} \psi_{dr}^e i_{qr}^e \quad (3.16)$$

olarak elde edilir. Denklem (3.15), denklem (3.16)' da yerine yazılarak yeni moment denklemi,

$$T_e = \frac{3}{2} \frac{p}{2} \frac{L_m}{L_r} \psi_{dr}^e i_{qs}^e \quad (3.17)$$

olur. Bu denklemden, ψ_{dr}^e rotor akı halkalanması sabit tutulursa, motor momentinin stator akımının q bileşeni olan i_{qs}^e ile değiştirilebileceği görülmektedir. $\psi_{qr}^e = 0$ olarak tutulursa,

$$\frac{d\psi_{qr}^e}{dt} = 0 \quad (3.18)$$

olacağından denklem (3.3) ile verilen q-ekseni rotor gerilim denkleminde kayma frekansı

$$v_{qr}^e = 0 = R_r i_{qr}^e + \frac{d}{dt} \psi_{qr}^e + (\omega_e - \omega_r) \psi_{dr}^e \quad (3.19)$$

$$\omega_e - \omega_r = \omega_{sl} = -\frac{R_r}{\psi_{dr}^e} i_{qr}^e \quad (3.20)$$

olarak bulunur. Eğer ψ_{dr}^e rotor akı halkalanması sabit tutulursa sürekli durumda $d\psi_{dr}^e / dt = 0$ olacaktır. Bu durumda denklem (3.4)'den,

$$v_{dr}^e = 0 = R_r i_{dr}^e + \frac{d}{dt} \psi_{dr}^e - (\omega_e - \omega_r) \psi_{qr}^e \quad (3.21)$$

$$i_{dr}^e = 0 \quad (3.22)$$

olarak elde edilir. $i_{dr}^e = 0$ olması durumunda denklem (3.13) ve denklem (3.14) 'den aşağıdaki gibi bulunur.

$$\psi_{dr}^e = L_m i_{ds}^e \quad (3.23)$$

$$i_{qr}^e = -\frac{L_m}{L_r} i_{qs}^e \quad (3.24)$$

Bu denklemler denklem (3.20)'de yerine yazılarak kayma frekansı, aşağıdaki gibi yazılabilir.

$$\omega_{sl} = \frac{R_r}{L_r} \frac{i_{qs}^e}{i_{ds}^e} \quad (3.25)$$

Uygulamada rotor alanının genliği ($\psi_{qr}^e = 0$ olduğundan $\hat{\psi} = \psi_{dr}^e$) i_{ds}^e akımının değiştirilmesi ile ayarlanabilir. Rotor alanının d°-eksenine yönlendirilmesi denklem (3.25) gereği ya kayma hızı ya da i_{qs}^e akımıyla sürdürülebilir. ψ_{dr}^e rotor akı halkalanması sabit değilse denklem (3.11), denklem (3.12) ve denklem (3.21)' den,

$$i_{dr}^e = \frac{\psi_{dr}^e}{L_r} - \frac{L_m}{L_r} i_{ds}^e \quad (3.26)$$

$$\psi_{dr}^e = \frac{L_m}{1 + S\tau_r} i_{ds}^e \quad (3.27)$$

olarak bulunur. Denklem (3.27)'den de görüleceği gibi ψ_{dr}^e akısının dinamiği rotor zaman sabitine bağlıdır.

3.1.3 Asenkron Motorların Vektör Denetimi

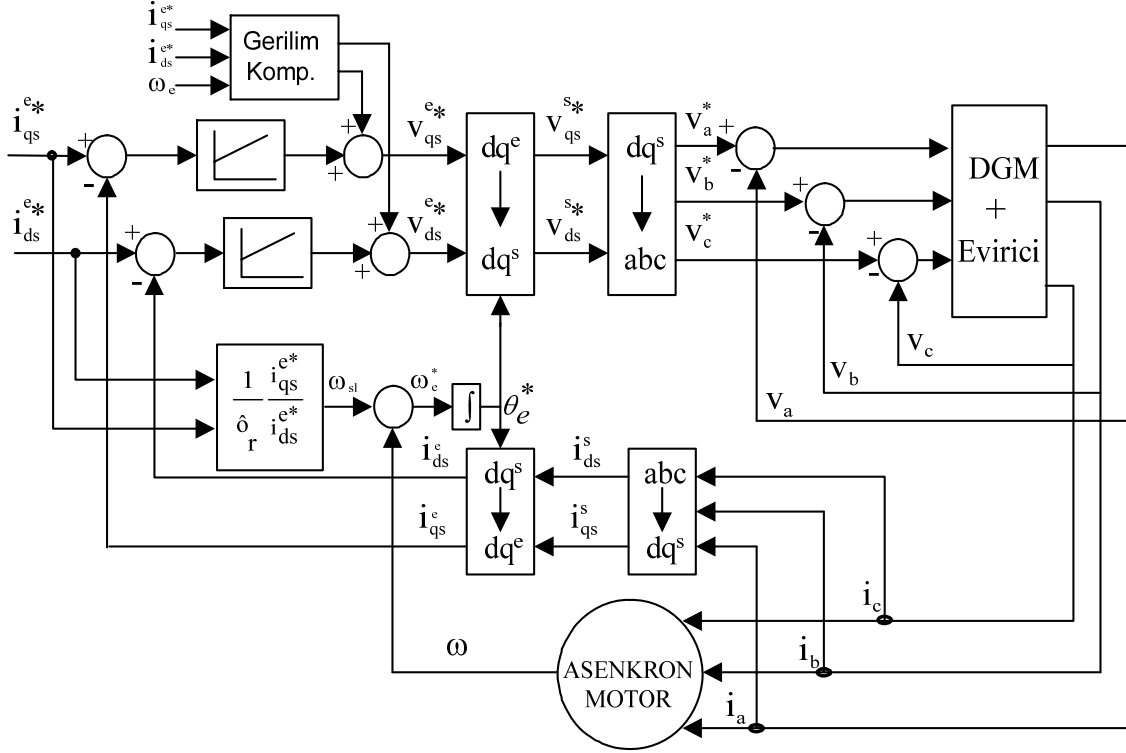
Asenkron motorların vektör denetimi, akı vektörünün genlik ve konum bilgisinin elde edilme şekline göre iki farklı yöntemle elde edilir ve bu yöntemler aynı zamanda vektör denetim yöntemlerini belirler [92, 94]. Bu yöntemler, doğrudan vektör denetim yapısı ve dolaylı vektör denetim yapısı olarak adlandırılır.

Bu iki yöntem arasındaki temel fark akı kestirimi için seçilen motor modelinin farklı olmasından kaynaklanmaktadır. Doğrudan vektör denetim yönteminde, ihtiyaç duyulan rotor akı bilgisi, motorun fiziksel yapısında değişikliklerle veya doğrudan akı tahmini ile elde edilir. Özel imal edilmiş fiziksel değişiklikli motorların endüstriyel uygulamaları yoktur. Doğrudan akı tahmini için, stator akım ve gerilim büyüklükleri ölçülerek motor denklemlerinden akı tahmini yapılır [93]. Fakat bu yöntemde düşük hızlarda meydana gelen açık integrasyon sorunu vardır. Bu sorunun giderilmesi için literatürde çeşitli algılayıcısız vektör denetim yöntemleri geliştirilmiştir [92, 95-96]. Bu çalışmada asenkron motorun vektör denetimi için dolaylı vektör denetim yöntemi kullanılmıştır.

Asenkron Motorların Dolaylı Vektör Denetimi

Dolaylı vektör denetim yönteminde, rotor akısının genlik ve faz bilgisinin elde edilmesinde doğrudan denetim yönteminde olduğu gibi akı algılayıcısı/gözlemcisine gerek duyulmaz [97]. Dolaylı vektör denetiminde, akı vektörünün konumunun belirlenmesi için konum/hız algılayıcıları kullanılır. Rotor akısının konumu, motor miline yerleştirilen konum veya hız algılayıcılarından elde edilen konum bilgisi ile referans işaretten elde edilen referans

kayma değerinden yararlanılarak bulunur. Böylece rotor akısının konum bilgisi, doğrudan akı algılayıcıları veya gözlemleyicileri ile değil de dolaylı yoldan hesaplamalar ile elde edilir.



Şekil 3.5 Asenkron motorun dolaylı alan yönlendirmeli denetimi için blok yapısı

Dolaylı vektör denetim yapısı, birim vektör bileşenlerinin ($\sin\theta_e$ ve $\cos\theta_e$) belirlenmesi dışında doğrudan denetim yapısıyla temel olarak aynıdır [89, 93]. Şekil 3.5' de blok diyagramı verilen dolaylı vektör denetim yapısında, rotor akısının konumu motor modelinden yararlanılarak bulunur. Rotor alan yönlendirmesine göre elde edilen ve denklem (3.25)'de verilen kayma frekansı,

$$\omega_{sl}^* = \omega_e - \omega_r = \omega_e - \frac{p}{2} \omega_e = \frac{R_r}{L_r} \frac{i_{qs}^e}{i_{ds}^e} \quad (3.28)$$

olduğundan,

$$\omega_e = \omega_{sl} + \omega_r \quad (3.29)$$

ve akının konumu

$$\theta_e = \int (\omega_{sl} + \omega_r) dt \quad (3.30)$$

olarak hesaplanabilir. Ancak bu hesaplamada, rotor hızı/konumuna gereksinim vardır. Bu nedenle dolaylı vektör denetiminde motor hızı/konumunun algılayıcı veya gözlemleyiciler ile belirlenmesi gereklidir.

Uygulamada rotor akısının genliği i_{ds}^{e*} akımıyla, rotor akı yönlendirmesi ise i_{qs}^{e*} akımının denetimiyle yapılır.

$$\psi_{dr}^{e*} = \frac{L_m}{1 + S\tau_r} i_{ds}^{e*} \quad (3.31)$$

$$T_e^* = \frac{3}{2} \frac{P}{2} \psi_{dr}^{e*} i_{qs}^{e*} \quad (3.32)$$

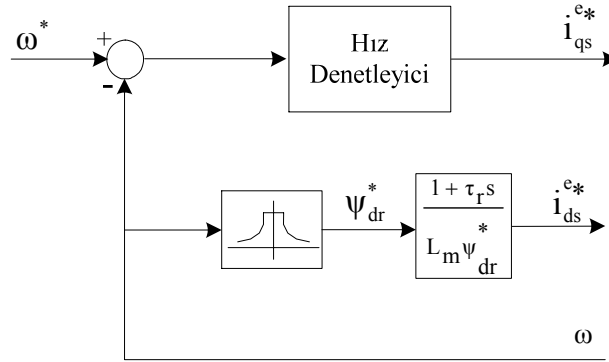
Denetim yapısında i_{qs}^{e*} akımı hız/moment denetleyicinin çıkışından elde edilebilir. Eğer i_{ds}^{e*} akımı sabit tutulursa, L_m manyetik indüktansında oluşacak her hangi bir değişim akıda kaymaya neden olacaktır. Bu durumda rotor alan yönlendirilmesinin sürdürülebilmesi için ($\psi_{qr}^e = 0$) gerekli ω_{sl}^* kayma hızı, i_{qs}^{e*} akımıyla sağlanır.

3.1.4 Asenkron Motorların Hız Denetimi

Asenkron motorların hız denetimi temel olarak skaler (V/f) ve vektörel (Alan yönlendirmeli) olarak gerçekleştirilmektedir [92, 93]. Skaler hız denetim yönteminde, motorun sürekli durumdaki eşdeğer devre modelinden yararlanılarak stator geriliminin frekansına oranı sabit tutulur. Bu denetim yönteminde motorun sürekli durum denklemleri kullanıldığından motorun geçici durumdaki davranışı kötüleşmektedir. Bununla birlikte sürekli durumda, düşük hızlar dışında geniş bir aralıkta hız denetimi gerçekleştirilmektedir. Düşük hızlarda stator direnci üzerine düşen gerilimin artmasından dolayı bu yöntemin başarımı oldukça kötüdür. Ayrıca bu yöntemde, motorun kenetleme etkisi tamamen ortadan kaldırılamadığından moment ve akı, frekans ve gerilimin (veya akımın) bir fonksiyonu olarak kalır ve bu nedenle de cevap süresi artar.

Asenkron motorların skaler hız denetimi, yöntemin basitliği nedeniyle yüksek başarımlı gerektirmeyen birçok sürücü sisteminde yaygın olarak kullanılmaktadır. Ancak yüksek başarımlı sürücü sistemlerinde hassas bir hız/konum ve moment denetimi gerekli olduğundan bu yöntem yetersiz kalabilmektedir. Asenkron motorun yüksek başarımlı denetimi motorun akı ve akımının bağımsız denetlenebildiği vektör denetim yöntemleri kullanılır. Vektör denetiminin başarımı hızlı ve parametre değişimlerinden etkilenmeyen bir denetleyiciyi gerektirir [98].

Asenkron motorların doğrudan ve dolaylı vektör denetimi yöntemlerinin her ikisinde de gerekli olan referans akım, hız denetleyicinin çıkışından elde edilmektedir. Vektör denetimli asenkron motorun hız denetim yapısı için genel bir blok diyagramı Şekil 3.6’ da verilmiştir. Şekilden görüldüğü gibi referans rotor akısı, alan zayıflatma bölgesinde motorun çalışabilmesi için hıza bağlı özel bir tablodan elde edilirken, motorun momentini denetleyen i_{qs}^{e*} referans akımı hız denetleyiciden elde edilmektedir. Eğer anma hızının üzerinde bir hız denetimi yapılmayacaksa sürekli durumda $i_{ds}^{e*} = \psi_{dr}^* / L_m$ olduğundan, referans rotor akısı yerine i_{ds}^{e*} referans akımı sabit olarak kullanılabilir.



Şekil 3.6 Vektör denetimli asenkron motor için hız denetim bloğu

Asenkron motor, doğru bir vektör denetimi ile DA sürücülerde olduğu gibi doğrusal sistemlerin denetiminde kullanılan geleneksel yöntemlerle denetlenebilir.

Asenkron motorların hız denetimlerinde geleneksel sabit parametrelili denetleyiciler, tasarımlarının kolay ve maliyetlerinin düşük olması nedeniyle yaygın olarak kullanılmıştır. Asenkron motor sürücü sistemlerinde, sabit parametrelili denetleyiciler motorun doğrusal olmayan yapısı, parametrelerindeki değişim, yük momenti ve eylemsizlik katsayılarındaki değişim gibi nedenlerle etkin bir hız ya da konum denetimi sağlayamazlar [89].

İnsan beyninin çalışma ilkesinin benzetimini amaçlayan YSA, katmanlar halinde düzenlenmiş çok sayıda doğrusal olmayan hücrelerin ağırlıklandırılmış bağlantılarla birbirine bağlandığı paralel çalışma yeteneğine sahip matematiksel bir model olarak tanımlanır [99]. YSA belirlenen bir amacı sağlamak için hücreler arasındaki bağlantı ağırlıklarını çeşitli öğrenme kuralları ile değiştirebilen, bilgi toplayabilen ve ağırlıkları yardımı ile bu bilgiyi saklayabilen paralel bir işlemcidir [100]. YSA bu özellikleriyle nesne tanılama, görüntü ve işaret işleme, parametre tanılama ve sistem denetimi gibi birçok farklı alanda uygulanmıştır.

Denetim alanında YSA, doğrusal olmayan ve uyarlanırlı yapıları, öğrenme ve genelleme yetenekleri, sistem parametrelerinden bağımsız olarak tasarlanabilmeleri gibi üstün özelliklerinden dolayı doğrusal olmayan bir çok sistemim denetiminde ve tanınmasında kullanılmıştır [101–102]. YSA'nın eğitim sorunu, saklı katman sayısı ve bu katmanda kullanılacak hücre sayısının belirlenmesinde kesin bir kuralın olmaması YSA'nın başlıca sorunlarıdır.

Bulanık Mantık (BM), insanın düşünce sisteminde olduğu gibi, var-yok mantığının dışında ara değişkenleri de kullanarak, insanın karar verme yeteneğini matematiksel olarak modellemeye çalışır [103]. Sistemin matematiksel modeline ihtiyaç duymadan uzman bilgi ve tecrübelerini bünyesinde bulundurabilen BM sistemler, doğrusal olmayan ve zamanla değişen parametrelere sahip sistemlerin denetiminde yaygın olarak kullanılmıştır [104–108]. BM sisteminin oluşturulmasında kural tablosu ve üyelik fonksiyonları şekli ve sayısının belirlenmesinde kesin bir yöntem olmaması karşılaşılan en büyük zorluklardır.

Doğrusal olmayan ve zamanla değişen parametrelere sahip sistemlerin sabit parametrelili denetim yöntemleri ile denetiminde karşılaşılan sorunlardan dolayı modern denetim yöntemlerinde; doğrusal olmayan denetim, optimum denetim, dayanıklı denetim ve uyarlamalı denetim yöntemleri kullanılmaktadır [107, 108]. Asenkron motor sürücülerinde karşılaşılan doğrusal olmayan motor yapısı, modellenemeyen dinamikler ve zamanla değişen parametre sorunları asenkron motorların sürme sistemlerinde de bu yöntemlerin kullanılmasını zorunlu hale getirmiştir. Son yıllarda yapılan çalışmalarda bu yönde olup; asenkron motorların kayan kipli denetleyiciler, BD, YSA denetleyiciler ve SBD ile dayanıklı veya uyarlamalı denetimleri yapılmaya başlanmıştır. Bu yöntemlerden özellikle SBD hem YSA hem de BD'nin üstün özelliklerini yapısında taşıdığından son yıllarda asenkron motor sürücü sistemlerinde de kullanılmaya başlanmıştır. Bu tezde hazırlanan uzaktan erişimli laboratuarda, örnek çalışma olarak, SBA kullanılarak asenkron motorun hız denetimi gerçekleştirilmiştir. Ancak hazırlanan arayüz kullanıcının istediği denetim algoritmasını yazılıma yerleştirmesine izin vermektedir.

3.1.5 Asenkron motorun SBD ile Hız Denetimi

Dolaylı vektör denetim yapısında, ihtiyaç duyulan referans kayma hızı (ω_{sl}^*) denklem (3.25) kullanılarak,

$$\omega_{sl}^* = \frac{R_r}{L_r} \frac{i_{qs}^{e*}}{i_{ds}^{e*}} \quad (3.33)$$

38

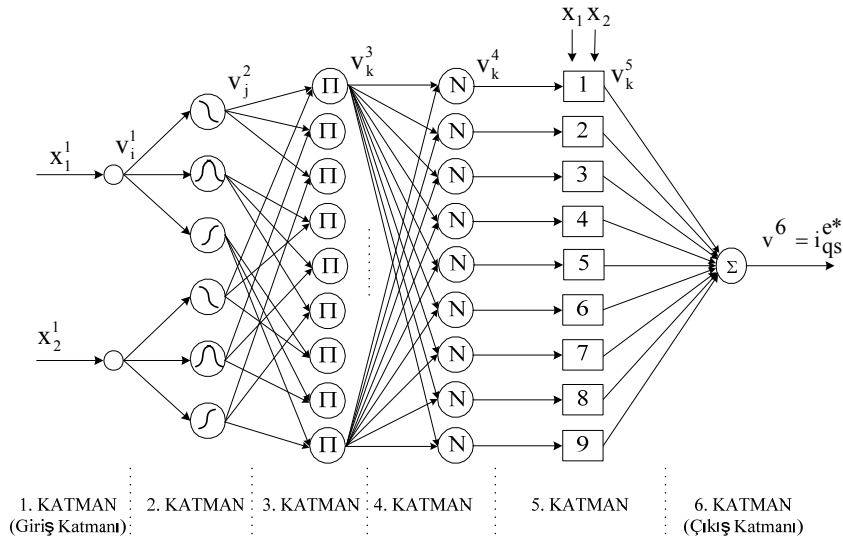
Burada, x_i giriş değişkeni, y çıkış değişkeni, $A_i^j \mu_{A_i^j}(x_i)$ üyelik fonksiyonlu önkoşullarının

dilsel değişkenleri ve $a_i^j \in \mathbb{R}$ ise doğrusal $f_i(x_1, x_2, \dots, x_n)$ fonksiyonunun katsayılarıdır.

x_1 ve x_2 giriş değişkenleri ve y çıkışı için Sugeno bulanık kuralları aşağıdaki gibi tanımlanır.

$$\begin{aligned} R^1 : & \text{IF } x_1 \text{ is } A_1^1 \text{ AND } x_2 \text{ is } A_2^1, \text{ THEN } y = f_1 = p_1 x_1 + q_1 x_2 + r_1 \\ R^2 : & \text{IF } x_1 \text{ is } A_1^2 \text{ AND } x_2 \text{ is } A_2^2, \text{ THEN } y = f_2 = p_2 x_1 + q_2 x_2 + r_2 \end{aligned} \quad (3.37)$$

Çalışmada kullanılan iki giriş tek çıkışlı ve her bir giriş için üç üyelik fonksiyonunun seçildiği Sugeno tipi SBD yapısı Şekil 3.8’de verilmiştir.



Şekil 3.8 İki girişli tek çıkışlı Sugeno tip SBD yapısı

Ağın girişten çıkışa doğru denklemleri katmanlar halinde ele alınmıştır.

1. Katman:

Bu katmandaki her hücre doğrusal aktivasyon fonksiyonlarına sahiptir ve giriş değişkenlerini çıkışlarına aktarırlar. Giriş değişkenleri olarak PD-SBD için aşağıdaki gibi hata ve hatanın değişimi olarak alınmıştır.

$$x_1 = e(t) = \omega^*(t) - \omega(t) \quad (3.38)$$

$$x_2 = \Delta e(t) = e(t) - e(t-1) \quad (3.39)$$

Tanımlanan x_i girişlerine göre 1. katmanın çıkışları,

$$u_i^1 = x_i^1 \quad (3.40)$$

$$v_i^1 = f_i^1(u_i^1) = x_i^1 \quad (3.41)$$

olarak yazılabilir.

2.Katman:

Bu katman üyelik fonksiyonlarının gerçekleştirildiği katmandır ve bu katmandaki her bir sinir hücresinin aktivasyon fonksiyonu üyelik fonksiyonun yerini tutar. Her bir giriş için üçer adet üyelik fonksiyonu belirlenmiştir. Bu üyelik fonksiyonları aşağıda denklemleri verildiği gibi ikisi sigmoidal diğeri ise genelleştirilmiş bell fonksiyonudur.

$$u_j^2 = -\sigma_{ij} \left(x_i^2 - m_{ij} \right), \quad v_j^2 = f_j^2(u_j^2) = \frac{1}{1 + \exp(u_j^2)} \quad j=1,3. \quad (3.42)$$

$$u_j^2 = \left| \frac{x_i^2 - m_{ij}}{\sigma_{ij}} \right|^{2b_i}, \quad v_j^2 = f_j^2(u_j^2) = \frac{1}{1 + u_j^2} \quad j=2. \quad (3.43)$$

Burada σ_{ij} , m_{ij} ve b_i üyelik fonksiyonlarının uyarlanacak parametreleridir ve giriş parametreleri olarak da adlandırılır. x_i^2 , 2. katmanın i. hücresinin girişidir.

3.Katman:

Kural tabanının oluşturulduğu bu katman Π sembolüyle gösterilen düğümlerden oluşmuştur ve her düğüm kendine gelen işaretlerin çarpımını çıkışına verir. Burada And bulanık işlemi gerçekleştirilmekle birlikte diğer T-norm işlemleri de kullanılabilir. And bulanık işleminin gerçekleştirilmesiyle k. hücrenin çıkışı,

$$u_k^3 = \prod_j x_j^3 \quad (3.44)$$

$$v_k^3 = f_k^3(u_k^3) = u_k^3 \quad (3.45)$$

olarak yazılır. Burada x_j^3 , 3. katmanın j. hücresinin girişidir.

4.Katman:

Bu katmandaki her bir hücre N ile gösterilmiştir. Her bir hücre, bir kuralın normalleştirilmiş ateşleme kuvvetini hesaplar. m. düğüm çıkışı, m. kuralın ateşleme derecesinin bütün kuralların ateşleme derecelerinin toplamına oranından aşağıdaki gibi hesaplanır.

$$u_k^4 = \sum_k x_k^4 \quad (3.46)$$

$$v_k^4 = f_k^4(u_k^4) = \frac{x_k^4}{u_k^4} \quad (3.47)$$

x_k^4 , 4. katmanın k. hücresinin girişidir.

5.Katman:

Bu katmandaki her bir n düğümünde, ağırlıklandırılmış $v_k^4(px_1 + qx_2 + r)$ sonuç değeri hesaplanır. Burada v_k^4 4. katmanın çıkışı, $\{p,q,r\}$ ise uyarlanacak parametre kümesidir. Bu katmandaki parametreler sonuç parametreleri olarak adlandırılırlar.

$$u_k^5 = x_k^5 \quad (3.48)$$

$$v_k^5 = f_o^6(u_o^6) = u_k^5 g_k = u_k^5 (p_k x_1^1 + q_k x_2^1 + r_k) \quad (3.49)$$

x_k^5 , 5. katmanın k. hücresinin girişidir.

6.Katman:

Bu katmanda Σ sembolüyle gösterilen tek bir düğüm vardır ve girişine gelen işaretlerin toplamını çıkışına verir. SBD' nin y çıkışı,

$$u_o^6 = \sum_k x_k^6 \quad (3.50)$$

$$v_o^6 = i_{qs}^{e*} = f_o^6(u_o^6) = u_o^6 \quad (3.51)$$

olarak bulunur.

Sinirsel Bulanık Denetleyicilerin Eğitimi

Bir SBD, BD işlevlerini YSA ile gerçekleştirerek YSA'nın öğrenme ve uyarlama yeteneğini kazanır. Böylece SBD'nin tasarımında, BD'nin tasarımında karşılaşılan üyelik fonksiyonlarının şeklinin belirlenmesi ve kural tabanının oluşturulması gibi zorluklar, YSA tarafından kazandırılan öğrenme ve uyarlama yeteneği ile aşılabılır. Ağın eğitimi geriye yayılım algoritması ile örneksel öğrenme yöntemi kullanılarak giriş ve çıkış parametrelerinin güncellenmesi için, izleme hatası (e) minimize edilecek karesel hata (E) aşağıdaki gibi belirlenir.

$$e(k) = \omega^*(t) - \omega(t)$$

$$E = \frac{1}{2} e^2(t) \quad (3.52)$$

SBD' nin uyarlanacak herhangi bir giriş ve çıkış parametresi ϕ olarak alınırsa, geriye yayılım algoritmasını kullanarak herhangi bir ϕ parametresinin güncellenmesi,

$$\phi(k) = \phi(k-1) + \Delta\phi(k) = \phi(k-1) + \left(-\alpha \frac{\partial E(k)}{\partial \phi(k)} \right) \quad (3.53)$$

olarak bulunur. Burada α -öğrenme oranıdır ve $\frac{\partial E}{\partial \phi}$ kısmi türevinin elde edilebilmesi için, E fonksiyonundan ilgili parametreye kadar kısmi türev zinciri kullanılır. E fonksiyonundan SBD' nin çıkışına kadar olan türev zinciri aşağıdaki gibi bulunabilir.

$$\delta^1 = \frac{\partial E}{\partial e} \frac{\partial e}{\partial \omega} \frac{\partial \omega}{\partial v^6} \quad (3.54)$$

Burada δ^1 -yöresel gradyandır. Denklem (3.38) ve denklem (3.51) kullanılarak SBD' nin çıkışından itibaren sonuç parametrelerinden p_k 'nin güncellenebilmesi için gerekli gradyanlar aşağıdaki gibi bulunur. Diğer sonuç parametreleri q_k ve r_k da benzer şekilde güncellenebilir.

$$\frac{\partial E}{\partial p_k} = \delta^1 \frac{\partial v^6}{\partial u^6} \frac{\partial u^6}{\partial v_k^5} \frac{\partial v_k^5}{\partial p_k} = \delta^1 u_k^5 x_1^1$$

$$\frac{\partial E}{\partial p_k} = \delta^1 \frac{x_k^4}{\sum_k x_k^4} x_1^1 \quad (3.55)$$

SBD' nin önkoşul parametrelerinin güncellenebilmesi için gerekli eğim hesabı ise aşağıdaki gibi elde edilebilir

Sigmoidal üyelik fonksiyonları için $\sigma_{i,1-3}$;

$$\frac{\partial E}{\partial \sigma_{i,1-3}} = \delta^1 g_k \frac{\sum_k x_k^4 - x_k^4}{\left(\sum_k x_k^4 \right)^2} \prod_j x_j^3 (x_i^1 - m_{i,1-3}) v_j^2 (1 - v_j^2) \quad (3.56)$$

$m_{i,1-3}$ parametreleri için,

$$\frac{\partial E}{\partial m_{i,1-3}} = \delta^1 g_k \frac{\sum_k x_k^4 - x_k^4}{\left(\sum_k x_k^4\right)^2} \prod_j x_j^3 (-\sigma_{i,1-3}) v_j^2 (1 - v_j^2) \quad (3.57)$$

Genelleştirilmiş bell fonksiyonu için,

$$\frac{\partial E}{\partial \sigma_{i,2}} = \delta^1 g_k \frac{\sum_k x_k^4 - x_k^4}{\left(\sum_k x_k^4\right)^2} \prod_j x_j^3 \left(\frac{2b_i}{x_i^1 - m_{i,2}}\right) v_j^2 (1 - v_j^2) \quad (3.58)$$

$$\frac{\partial E}{\partial m_{i,2}} = \delta^1 g_k \frac{\sum_k x_k^4 - x_k^4}{\left(\sum_k x_k^4\right)^2} \prod_j x_j^3 \frac{\partial v_j^2}{\partial m_{i,2}} \quad (3.59)$$

Burada $\frac{\partial v_j^2}{\partial m_{i,2}}$ türevi aşağıdaki gibi elde edilebilir.

$$\frac{\partial v_j^2}{\partial m_{i,2}} = \begin{cases} \left(\frac{2b_i}{x_i^1 - m_{i,2}}\right) v_j^2 (1 - v_j^2) & \text{if } x_i^1 \neq m_{i,2} \\ 0 & \text{if } x_i^1 = m_{i,2} \end{cases} \quad (3.60)$$

$$\frac{\partial E}{\partial b_i} = \delta^1 g_k \frac{\sum_k x_k^4 - x_k^4}{\left(\sum_k x_k^4\right)^2} \prod_j x_j^3 \frac{\partial v_j^2}{\partial b_i} \quad (3.61)$$

Burada $\frac{\partial v_j^2}{\partial b_i}$ türevi aşağıdaki gibi elde edilebilir.

$$\frac{\partial v_j^2}{\partial b_i} = \begin{cases} -2 \ln \left| \frac{x_i^1 - m_{i,2}}{\sigma_{i,2}} \right| v_j^2 (1 - v_j^2) & \text{if } x_i^1 \neq b_i \\ 0 & \text{if } x_i^1 = b_i \end{cases} \quad (3.62)$$

3.2. Yazılım Yapısı

Uzaktan erişimli laboratuvar ortamları hazırlanırken; istemci tarafa mümkün olduğunca az iş yükü bırakılması, istemci tarafında herhangi ek bir yazılım yada donanıma ihtiyaç duyulmaması veya istemci yazılımının kullanımı kolay ve görsel olması gibi hususlara dikkat edilmelidir. Örneğin Flash, Java Appletleri veya ActiveX denetimleri gibi yazılım ortamları kullanıldığında istemciye ek programlar yüklemek gerekecektir. Uzaktan erişimli laboratuvarı gerçekleştirmede kullanılacak yazılımları seçerken bu kısıtlar göz önüne alınarak benzetim ortamının hazırlanmasında HTML, ASP ve MWS yazılımları seçilmiş, deneysel çalışma ortamının hazırlanmasında ise Delphi ve Python yazılım ortamları kullanılmıştır.

3.2.1. HTML ve ASP

HTML (Hyper Text Markup Language-Yüksek Metin İşaret Dili) dili ile hazırlanan web sayfaları internet ortamına taşınabilmektedir. HTML standardında hazırlanan eğitim araçlarına, Internet Explorer gibi bir internet tarayıcısı yazılımı ile erişilebilmektedir. Web sayfasının daha etkileşimli tasarlanması ve kullanıcıdan alınan verilerin daha sonra kullanılmasına imkân veren ASP (Active Server Pages- Etkin Sunucu Sayfaları) gibi nesneye yönelik bir programlama dili kullanılması gerekmektedir. ASP kodu Microsoft tarafından geliştirilmiş bir teknolojidir ve web sayfası tasarımı yapmak için kullanılmaktadır. ASP kodun içerisinde HTML ve ASP scriptleri bulunabilmektedir. ASP programlama dili ile hazırlanan web sayfasına girildiğinde sunucu bilgisayar ASP kodunu çalıştırmakta ve sonuç web sayfasında görülmektedir. ASP sayesinde veri tabanı işlemleri ve elektronik ticaret siteleri gibi etkileşimli uygulamalar kolayca oluşturulabilmektedir. Bir HTML belgesi çift tıklandığında varsayılan tarayıcı açılacak ve bu sayfa yorumlanarak, görüntülenecektir. Fakat uzantısı asp olan bir dosya çift tıklandığında ya Windows bu dosyayı hangi programla açmak istenildiğini soracaktır ya da varsayılan editörde içerik olarak görüntüleyecektir. Asp dosyalarının yayınlanabilmesi için web sunucuya ihtiyacı vardır. Web sunucunun kurulabilmesi için bilgisayarda ya bir ethernet kartı ya da modemin bulunması ve TCP/IP protokolünün yüklü olması gerekmektedir. HTML sayfaları içerisine ASP kodlarını gömerek dinamik sayfalar oluşturabilmektedir. Bu HTML kodlarının içerisine gömdüğümüz ASP kodları sunucu tarafında işlenerek, istemcinin tarayıcısına sadece durağan HTML kodları gönderilmektedir. ASP programlama dili ile dinamiklik, güvenlik, modülerlik, hız, sistem yöneticisine kolaylık, veri madenciliği ve etkileşim sağlanabilmektedir.

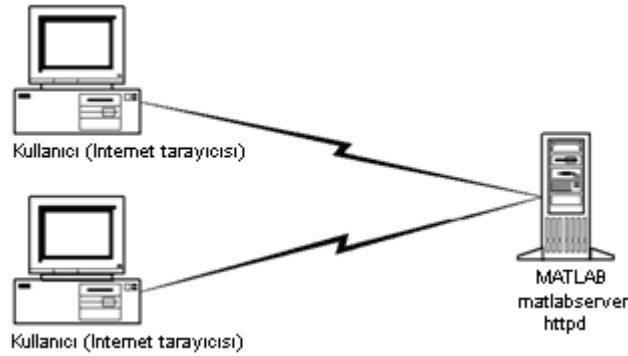
Dinamik web sayfaları yüksek performansta hız ve erişim kolaylığı sağlamaktadır. Statik sayfalara göre esnek bir yapıya sahiptirler. Sitede değişiklik yapmak isteyen kullanıcının

herhangi bir programcılık bilgisine ihtiyacı bulunmadan istediği değişiklikleri yapabilmektedir. Hızlı çalışmaları, kullanıcılar tarafından kolaylıkla güncellenebilmesi, arama motorlarında daha kolay indekslenmeleri gibi üstünlüklerinden dolayı bu tez çalışmasında sitenin yapısı dinamik olarak tasarlanmıştır.

HTML ve ASP sayfalarını oluşturmak için Microsoft FrontPage gibi editör programlarını kullanmak kolaylık sağlamaktadır. Microsoft FrontPage içerisinde bulunan web bileşenleri ile sayfa içerisinde istenen tasarım kolaylıkla sağlanabilmektedir. Bunun dışında animasyonlar, grafikler ve sayfa düzeni için Macromedia serilerinden yararlanılmıştır.

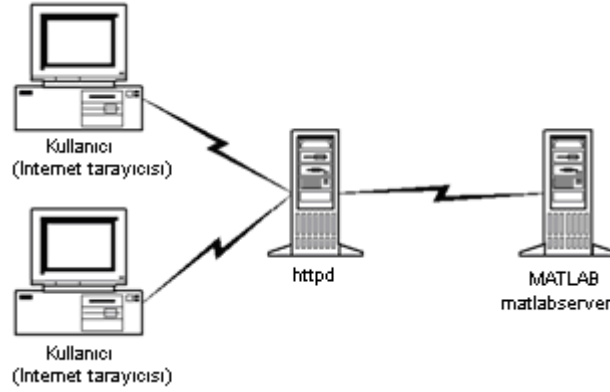
3.2.2. Matlab Web Sunucu

Matlab Web Sunucu (MWS) internetin yeteneklerini kullanarak, bir internet tarayıcısında (web browser) Matlab programına veri göndererek hesaplamalar yaptırmak ve sonuçlarını göstermek için web tabanlı Matlab uygulamalarını oluşturmaya izin vermektedir [109]. En basit yapılandırma biçimi Şekil 3.9’da gösterildiği gibi istemci bilgisayar üzerinde bir internet tarayıcısı, sunucu bilgisayar üzerinde ise Matlab, MWS ve sunucu hizmeti veren Windows Xp gibi yazılımların kurulmuş olması gerekmektedir.



Şekil 3.9 MWS yapılandırması

Şekil 3.10’da görüldüğü gibi daha karmaşık ağ yapılarında Web sunucu hizmeti başka bir bilgisayarda, uygulama hizmeti de bir başka bilgisayarda verilebilmektedir. Şekil 3.10’daki yapı ile sunucunun yükü azaltılarak uygulamaların daha hızlı çalışmaları sağlanabilmektedir.



Şekil 3.10 Karmaşık ağ yapılarında MWS yapılandırması

MWS, Matlab ile istemci bilgisayar arasındaki veri iletişimini TCP/IP protokolü ile sağlamaktadır. Ayrıca MWS, uygulamalar için gerekli olan ağ yazılımı ve donanımından başka fazladan bir yazılıma veya donanıma ihtiyaç duymamaktadır. MWS uygulamaları Hypertext Markup Language (HTML), m-dosyaları ve grafiklerin birleşiminden oluşmaktadır. MWS uygulamalarını aktif bir şekilde kullanabilmek için temel HTML ve Matlab programlama bilgisine ihtiyaç bulunmaktadır.

Uygulama geliştirmek için yapılması gereken işlemler şunlardır:

- Kullanıcılardan veri girişlerini almak ve uygulamanın çıkış sonuçlarını göstermek için HTML dosyaları oluşturulmalıdır. HTML dosyaları herhangi bir yazı editörü veya Microsoft Office Frontpage, Macromedia Dreamweaver, Adobe PageMill, SoftQuad HoTMetaL gibi ticari yazılımlar ile hazırlanabilmektedir.
- matweb.conf isimli konfigürasyon dosyasında ilgili konfigürasyon verileri ve uygulama adı yazılmalıdır.
- Bir Matlab m-dosyası aşağıdaki kurallara uygun bir şekilde hazırlanmalıdır:
 - o HTML giriş formundan girilen verileri almalıdır.
 - o Veri analizlerini yaparak istenen grafikleri üretebilmelidir.
 - o Matlab yapısı içine çıkış verilerini yerleştirmelidir.
 - o Çıkış verilerini HTML çıkış belgesi şablonuna yerleştirmek için Matlab programının htmlrep fonksiyonunu çağrılmalıdır. HTML belgesinden MATLAB programına gönderilebilecek maksimum veri 256 KB'ı geçmemelidir.

MWS'in Gereksinimleri

MWS'in hatasız bir şekilde çalışması için en az Matlab 5.0 sürümünün sunucu bilgisayara kurulmuş olması ve ek hafıza gerekmektedir. MWS çalışırken, yapılandırılmış

(konfigüre edilmiş) Matlab uygulama sayısına bağlı olarak hafıza gereksinimi değişmektedir. MWS altında çalışan her bir Matlab uygulaması 256 KB hafıza harcamaktadır. MWS'ın ihtiyacı olan TCP/IP ağ yazılımının bilgisayarda kurulu olması gerekmektedir [83, 109].

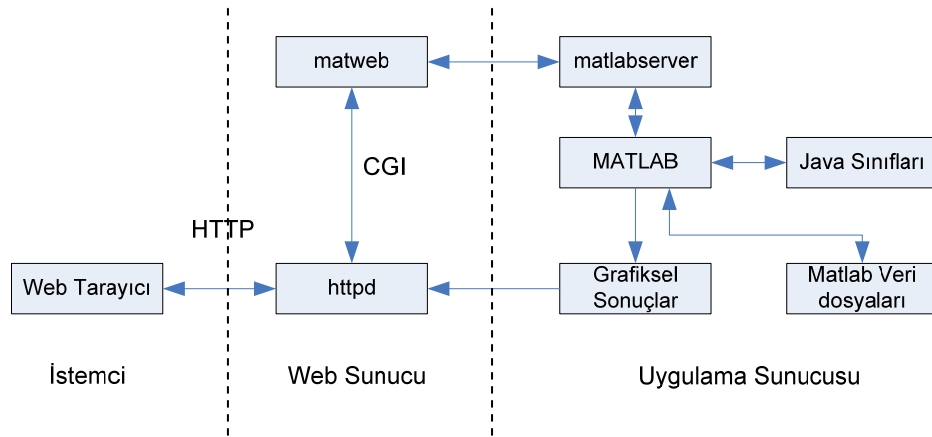
MWS'a girişleri göndermek ve MWS'dan çıkışları almak için bilgisayarda Netscape Navigator Release veya Microsoft Internet Explorer gibi internet tarayıcısı yazılımlarının kurulmuş olması gerekmektedir.

Üzerinde Matlab kurulu olan ağ erişimli bir bilgisayara bir web sunucu yazılımını yüklenmesi gerekmektedir. Web sunucu yazılımı Common Gateway Interface (CGI) programlarını çalıştırma yeteneğine sahip olmalıdır. MWS, UNIX veya Microsoft Windows işletim sistemlerinde uyumlu bir şekilde çalışmaktadır. Matlab programını bilgisayara kurarken seçim kutusundan MWS'ı da seçip bilgisayara kurmak gerekmektedir.

MWS Bileşenleri

MWS, kullanıcıların Matlab programında uygulamalar oluşturulmasını ve bu uygulamalara web üzerinden erişilmesini sağlayan bileşenlerden oluşur [83, 109]:

- **matlabserver** : Web uygulamaları ve MATLAB arasındaki iletişimi yönetir. Çok yönlü hizmet veren bir TCP/IP sunucusudur. matlabserver HTML belgesindeki mlmfile diye isimlendirilmiş gizli (hidden) alanda belirtilen MATLAB m-dosyasını çalıştırır. M-dosyasını çalıştırırken matweb.m dosyasının yardımını alır. matlabwserver.conf dosyasında gerekli konfigürasyonlar yapılarak MWS'in hangi TCP/IP portunu dinleyeceği ve aynı anda kaç adet uygulamanın eş zamanlı çalıştırılabileceği ayarlanabilir.
- **matweb** : matlabserver'in bir TCP/IP istemcisidir. HTML dosyasından verileri almak ve bunları matlabserver'e iletmek maksadıyla CGI kullanır.
- **matweb.m** : Web uygulamasının istediği m-dosyasını çalıştırır.
- **matweb.conf** : matlabserver'a bağlanmak amacıyla matweb'in ihtiyaç duyduğu bir konfigürasyon dosyasıdır. Uygulamalar matweb.conf içerisinde listelenmelidir.



Şekil 3.11 MWS modeli

Herhangi bir m-dosyası bir web uygulaması ile birlikte kullanılmaktadır. Web uygulaması tarafından belirtilen m-dosyasını çalıştırması için matweb.m dosyasının Matlab dizin yolunda olması gerekmektedir. matweb ve matweb.conf dosyaları da cgi-bin klasörünün altında olması gerekmektedir. Üretilen grafikler buraya yerleştirilmelidir. Web sunucu bu grafikleri bulabilmeli ve programlar yazılabilmelidir. Şekil 3.11 ile MWS modeli verilmiştir.

matlabserver Windows NT veya UNIX işletim sistemi arkasında sürekli çalışmaktadır. matlabserver başlatıldığında başlangıç veri ayarları için matlabserver.conf dosyasını görmektedir. Windows NT işletim sisteminde Matlab kurulurken matlabserver.conf dosyası, matlab\werbserver klasörü içinde oluşturulmuştur. matlabserver.conf dosyasının içine bakıldığında ilk satırda yapılandırma ayarları gözükmemektedir. matlabserver.conf dosyasındaki veri ile temel seçenekler ayarlanmaktadır. Dosya içerisinde port numarası (p) ve eş zamanlı MATLAB uygulama sayısı (m) belirtilmektedir. matlabserver.conf dosyasının varsayılan içeriği açık bir şekilde aşağıdaki gibidir:

-m 1

Yukarıdaki ifade de matlabserver varsayılan port sayısı (8888) ile MATLAB'ın bir kopyasının çalışacağı anlatılmaktadır. matlabserver'in temel seçenekleri Tablo 3.1'de verilmiştir.

Tablo 3.1 Matlabserver temel seçenekleri

Seçenek	Anlamı
-p [n]	Port numarası. 8888 varsayılandır.
-m [n]	MATLAB'ın n kopyası. 1 varsayılandır.

Aşağıdaki HTML kod satırı kullanılarak MATLAB ile iletişim kurulmaktadır. Matweb, HTTP sunucu üzerinde çalışan ve matlabserver ile iletişim kuran bir programdır. matweb farklı bir bilgisayarda çalışan matlabserver'i yerleştirmek amacıyla matweb.conf dosyasının içindeki bilgiye ihtiyaç duyar.

```
...  
<form action="/cgi-bin/matweb.exe" method="post">  
...
```

matweb, HTML formlarından veri göndermek için CGI kullanan matlabserver'in bir istemcisidir. matweb bilgiyi matlabserver'a transfer edip, m-dosyasında yazılan uygulamaları çalıştırarak işlem sonuçlarını üretmektedir. MWS kurulduğunda matweb programı matlab\webserver\bin klasörüne yerleşir. HTTP sunucuya erişmek için matweb programının bir kopyasını uygulamanın cgi-bin klasörüne kopyalanması gerekmektedir.

matlabserver ile bağlanmak için matweb, matweb.conf yapılandırma dosyasındaki depolanmış bilgiye ihtiyaç duyar. matweb.conf dosyası matweb programı ile birlikte uygulamanın bulunduğu klasör içindeki cgi-bin klasöründe oluşturulur. matweb.conf dosyasının bir örneği aşağıdaki gibidir:

```
[simu_pik]  
mlserver=ATEKIN  
mldir=c:\simu_pi_kon
```

Birden fazla uygulama yapılandırmaları aynı dosya içinde görünmelidir. mlserver=ATEKIN satırında olduğu gibi her bir değişkenin değeri tek bir eşittir işareti (=) ile atanır. Ana uygulama dosya (m-dosyası) ismi köşeli parantez işareti ([]) ile gösterilmektedir. Örneğin [simu_pik] ana uygulama dosyasının ismidir. matweb.conf dosyasının içerdiği tüm alanlar Tablo 3.2'de verilmiştir.

Tablo 3.2 Matlabserver temel seçenekleri

Değişken	Anlamı	Örnek
[uygulama] (zorunlu)	Matlab uygulamasını çalıştırır	simu_pik
mldir (seçmeli)	Dosyaları okuma ve yazma için çalışma dizini yolunu belirtir	c:\simu_pi_kon
mllog (seçmeli)	Uygulama ve Matlab arasındaki tüm değişiklikleri kaydeden bir log dosyası üretir	c:\simu_pi_kon\simu.log
mlserver (zorunlu)	Çalışan matlabserver'in host ismidir	ATEKIN
mlport (seçmeli)	matlabserver'in dinlediği porttur. Bu değer matlabserver.conf dosyasındaki port numarası ile uyuşmalıdır	8888 (varsayılan)
mltimeout (seçmeli)	Zaman aşımından önce matlabserver'in beklediği süredir (saniye)	180 (varsayılan)

Yeni bir MWS uygulaması oluşturup, uygulamanın matweb.conf konfigürasyon verilerini girdikten sonra, uygulamayı kullanmadan önce matlabserver'i yeniden başlatmak gerekecektir.

Hazırlanan uygulamanın html-dosyasında aşağıdaki satır mutlaka bulunmalıdır:

```
...
<input type="hidden" name="mlmfile" value="simu_pik">
...
```

Adı (name) mlmfile ve değeri (value) simu_pik olan argümanlar yukarıdaki satırda görülmektedir. mlmfile argümanı çalıştırılacak Matlab m-dosyasının ismini içermektedir. matlabserver, Matlab uygulamasını (simu_pik.m) çalıştırmak için matweb.m dosyasından elde edilen mlmfile'in değerini (value) kullanır. simu_pik.m dosyası index.html dosyasından verileri alarak, bu verileri işler ve elde edilen sonuçları sonuc.htm dosyasında gösterir.

İşlenen veriler sonuc.htm dosyası ile web üzerinden görüntülenir. Hazırlanan simu_pik.m dosyasında aşağıdaki kod satırı bulunmaktadır:

```
...
str = htmlrep(s, 'sonuc.htm')
...
```

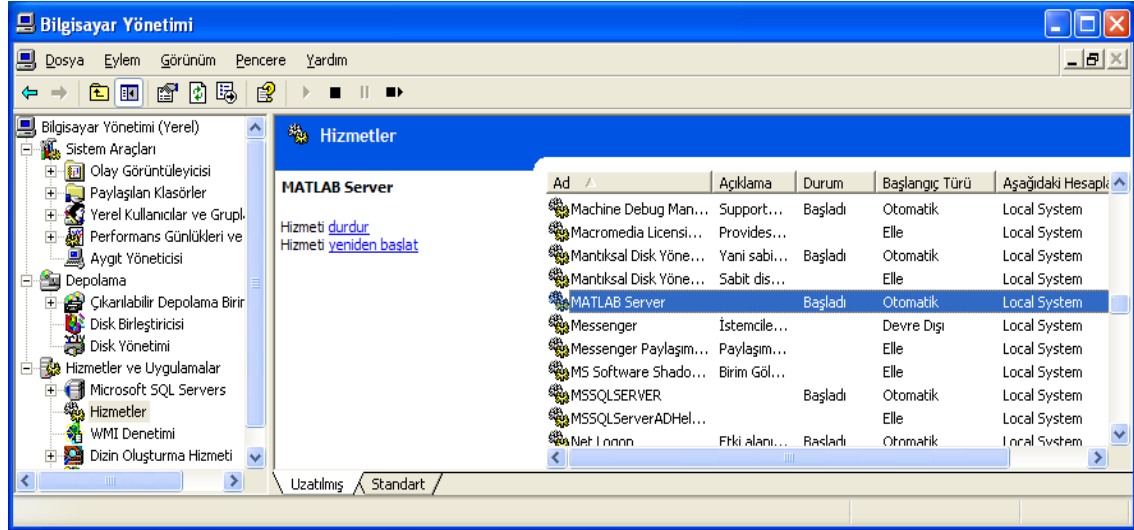
m-dosyasındaki hesaplama sonuçları s değişkeninde tutulmaktadır. Htmlrep fonksiyonu, s değişkeninden verileri alarak, sonuc.htm dosyasındaki değişken alanlarına atar. sonuc.htm dosyasında oluşturulan form kullanıcının internet tarayıcısına iletilir. MWS’da kullanılan fonksiyonlar Tablo 3.3’te verilmiştir.

Tablo 3.3 MWS’da kullanılan fonksiyonlar

	htmlrep
Yazım Şekli	outstring = htmlrep(instruct, infile)
Açıklama	outstring = htmlrep(instruct, infile) infile bir HTML dosyasıdır. m-dosyasındaki hesaplanan tüm değişkenler instruct ile tutulur ve infile ile yer değiştirir. Değişkenler sayısal veya string değerler içeren karakter dizisi, matris veya dizi olabilir. Sayısal ve string değişkenler yer değiştirmelidir. Çıkış outstring’e geri döndürülmektedir. infile’deki değişken isimleri dolar (\$) işareti arasına alınmalıdır. instruct değişken isimlerini ve aynı değerleri içeren bir MATLAB yapısıdır.
	matweb
Yazım Şekli	matweb (instruct)
Açıklama	matweb Matlab uygulamalarını çağıran bir m-dosyasıdır. matweb fonksiyonu (m-dosyası) matlabserver’a yardım etmektedir. instruct aşağıdaki alanlardan oluşmaktadır: • HTML giriş belgesindeki tüm veriler • m-dosyasının isminin depolandığı mfile • matweb.conf’da çalışma dizinlerinin belirlendiği mldir • Dosya ismi oluşturmak ve içeriğini korumak için mlid
	wscleanup
Yazım Şekli	deletecount = wscleanup(filespec, timewindow)
Açıklama	filespec ile belirtilen dosyaları, timewindow ile belirtilen süre sonunda siler
	wsprintjpg
Yazım Şekli	status = wsprintjpg(fig, jpegfilename)
Açıklama	Jpegfilename ismi ile bir jpeg dosyası oluşturur.

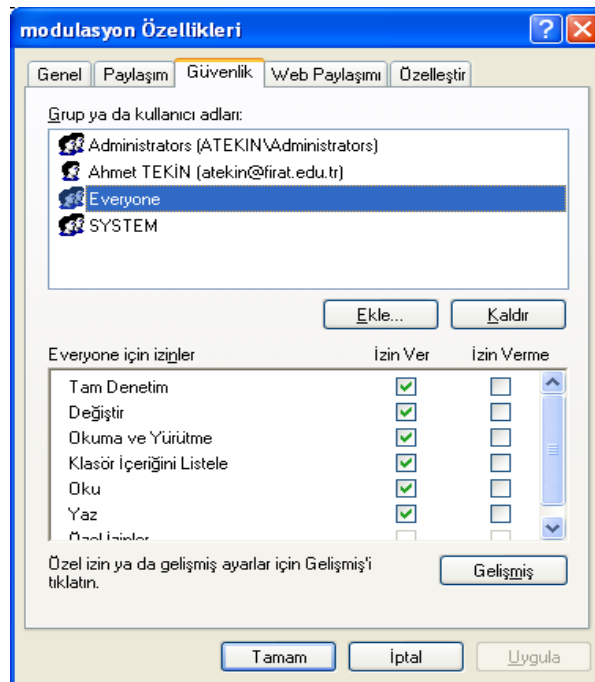
Yeni bir MWS uygulaması oluşturulduğunda, matweb.conf yol tanımlarının mutlaka yapılması gerekmektedir. Uygulamayı kullanmadan önce matlabserver’ı yeniden başlatmak gerekecektir. matlabserver’ı yeniden başlatmak için Bilgisayarım nesnesi üzerinde sağ tıklayıp

Yönet komutunu seçmek gerekmektedir. Şekil 3.12’de görüldüğü gibi Matlab Server seçilerek “Hizmeti Yeniden Başlat” tıklanmalıdır.

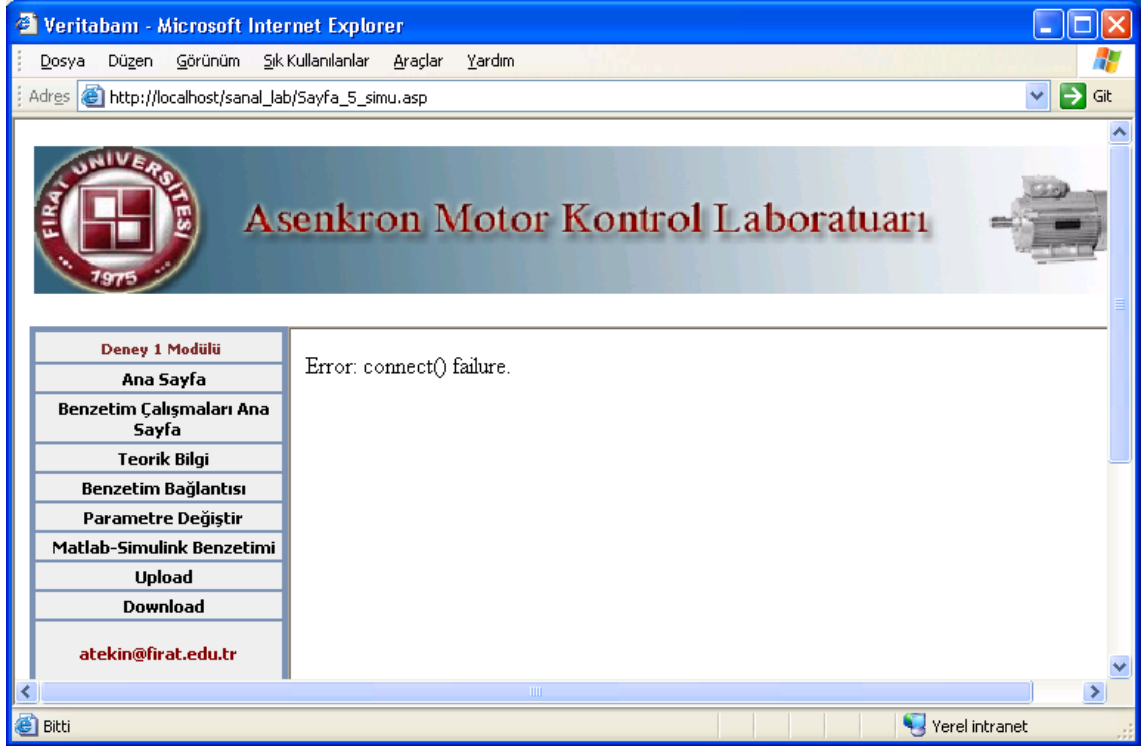


Şekil 3.12 Bilgisayar yönetimi penceresi

Ayrıca uygulamanın internet üzerindeki her bilgisayarda çalışması için uygulamanın bulunduğu klasör üzerinde sağ tıklayıp özellikler/güvenlik seçilerek Şekil 3.13’deki gibi Everyone isminde bir kullanıcı oluşturup yetkilerini vermek gerekmektedir. Şekil 3.14’deki gibi hata mesajı varsa matlabserver çalışmıyor demektir.



Şekil 3.13 Klasör özellikleri penceresi



Şekil 3.14 Uygulama hata penceresi

MWS' in Avantajları

MWS ile gerçek zamanlı benzetimler internet üzerinden gerçekleştirilebilmektedir. Bu özellik mühendislik ve teknik eğitim bölümlerinde sanal laboratuvar uygulamalarında son derece önemlidir. Benzetim arayüzündeki giriş değişkenleri sayısal veya metinsel olabilmekte ve çıkışlar da sayısal, metinsel veya grafiksel olabilmektedir. Özellikle mühendislik alanlarında çok sık kullanılan 2 ve 3 boyutlu grafiklerin, Matlab programı ile çok rahat çizdirilebilmesi önemli avantajlarından biridir.

MWS ile kullanıcılar Matlab programını kendi bilgisayarlarına kurmadan Matlab uygulamalarını çalıştırabilmektedirler. Kullanıcıların bilgisayarında Internet Explorer gibi bir internet tarayıcısının olması yeterlidir. Bununla birlikte kullanıcıların, benzetimleri çalıştırmak için özel bir Matlab programcılığı bilgisine de sahip olmalarına gerek bulunmamaktadır. Zaman ve mekandan tasarruf ve kullanıcının öğrenme motivasyonunu artırması da başka bir üstünlüğüdür. Kullanıcının istediği her parametre değerini uygulayarak daha geniş bir deneyime sahip olmasını sağlamaktadır.

MWS uygulamalarının sunucu üzerinde yüklü olan işletim sisteminden bağımsız olarak çalışması önemli bir özelliktir. Matlab uygulamasının yüklü olduğu sunucuya standart HTTP protokolü ile erişilmektedir. Bu özellik sayesinde Matlab programı ile kurulan bağlantı, sadece internet tarayıcıları ile değil, aynı zamanda bu görevi yerine getiren uygulamalarla da

sağlanabilmektedir. Böylece HTML dosyaları içerisinde gömülü olan Java appletlerini ve ActiveX kontrollerini de bağımsız olarak kullanmak mümkün olmaktadır [16, 83].

MWS'ın Dezavantajları

MPEG ya da AVI formatındaki uygulamaların cevaplama süresinin 1 dk. veya daha fazla olması dezavantajlarından biridir. MWS, Matlab programı içerisinde son derece önemli bir yeri olan grafiksel kullanıcı arayüzü bileşenlerini (GUI) desteklememektedir. Bu problem, MWS'a bilgileri sunan HTML giriş sayfasının daha estetik ve daha görsel olacak şekilde tasarlamakla aşılabılır. Giriş değişkenlerinin değerlerini verirken, “+”, “%” veya “&” gibi karakterler kullanıldığında sorunlar çıkmaktadır. Çünkü bu karakterler HTML protokolünün kontrol amaçlı kullandığı karakterlerdir. MWS kullanımının bir diğer dezavantajı ise, sunucu bilgisayarın güçlü bir donanıma sahip olması gerekliliğidir. MATLAB yorumlayıcı bir dildir, yüksek kapasiteli bir işlemciye ve yüksek RAM özelliklerine ihtiyaç duymaktadır [16, 83].

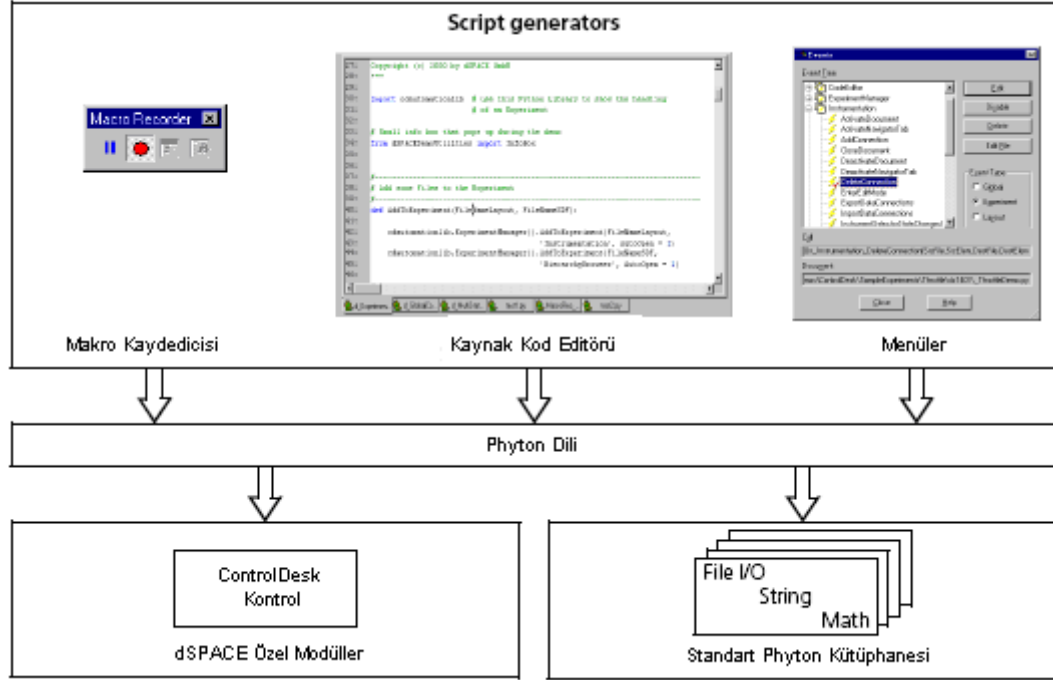
3.2.3. ControlDesk ve Python

MWS bazı DAQ veya GPIB gibi veri kartlarını desteklemekle birlikte gerçek zamanlı deney yapmaya yönelik DS1104 denetleyici kartı gibi bazı donanımları desteklememektedir. Bu çalışmada DS1104 denetleyici kartı ile asenkron motorun hız denetimi gerçekleştirileceğinden, DS1104 denetleyici kartına uzaktan erişimi sağlamak için yazılım geliştirmeye ihtiyaç duyulmuştur. DS1104 denetleyici kartının ControlDesk yazılımı Python programlama dili ile hazırlanmıştır. Bu nedenle ControlDesk yazılımına erişmek için yazılımlar geliştirmek zorunludur.

Bir Simulink model dosyası ile dSPACE çalışması oluşturmak için dSPACE tarafından geliştirilen ControlDesk Developer yazılımı kullanılmıştır. Bu yazılım, algoritmanın gerçek zamanlı olarak yürütülmesi için değişkenlerin yazılım denetleyicilerine atanmasını ve bir kullanıcı arayüzü oluşturulmasını sağlar. Böylece değişkenlerin gerçek zamanlı olarak denetimi ve gözlemlenmesi etkileşimli olarak sağlanır.

Python programlama dili kullanılarak hazırlanan ControlDesk yazılımı, Python derleyicisini kullanmaktadır. Python yüksek seviyeli, nesne tabanlı bir programlama dilidir. ControlDesk yazılımının temelini oluşturan Python derleyicisi, Python kodlarını çalıştırmaya veya Python komutlarını girmeye izin vermektedir. ControlDesk yazılımı bilgisayara kurulduğunda Python derleyicisi de bilgisayara yüklenir. Kullanıcı isterse harici bir Python derleyicisini bilgisayara ayrıca yükleyebilir.

Python dilinin fonksiyonel kullanımı için kütüphanelerin (modüllerin) eklenmesi (import edilmesi) gerekmektedir. dSPACE kurulurken standart Python kütüphanelerinin yanı sıra ControlDesk için özel Python modüllerini içeren kütüphaneler de bilgisayara yüklenmektedir [110].



Şekil 3.15 Kod üretme

Şekil 3.15’de görüldüğü gibi ControlDesk yazılımı kod üretmek için farklı araçlar kullanılmaktadır. ControlDesk kod üretimini veya kod değişikliklerini makrolar ile kaynak kod editörü ile veya menülerden olayları işleterek gerçekleştirmektedir.

ControlDesk yazılımında kullanılan hemen hemen bütün fonksiyonlar, özel bir Python sınıf kütüphanesi (modülü) olan `cdautomationlib` tarafından gerçekleştirilmektedir. `cdautomationlib` kütüphanesi ControlDesk yazılımına ulaşmak için ihtiyaç duyulan değişkenler, fonksiyonlar ve sınıfları sağlamaktadır. Her bir sınıf, özellik ve metotlar tanımlanmıştır. Özellikler, basit özellikler veya diğer sınıfların nesneleri olabilmektedir.

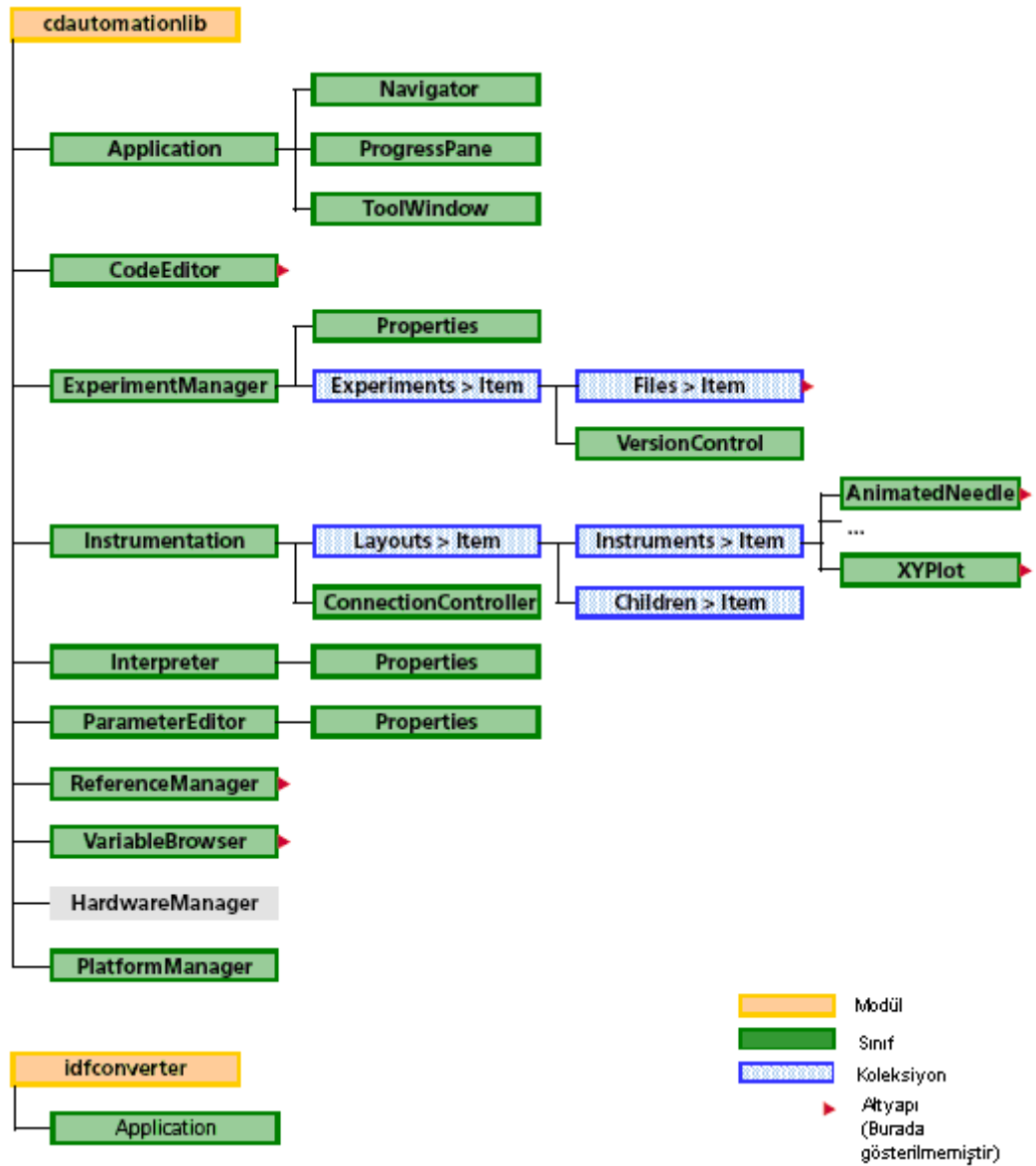
Bir sınıf tanımlı yapmak için sınıf-nesne durumunu bildirmek gerekmektedir. Nesne, sınıf metodları ve özelliklerden oluşmaktadır. Nesne ControlDesk yazılımının somut elemanlarını temsil etmektedir. Aynı tipteki birden fazla nesneyi yönetmeyi sağlayan sınıf koleksiyon olarak tanımlanmaktadır. Örneğin belirli bir layout’a erişmek için verilen layout’lardan biri seçilmesi gerekir. Koleksiyonlar sadece belirli bir nesneyi seçmek için metod sağlamazlar aynı zamanda koleksiyondan nesneleri atmak veya metodları toplamak için kullanılırlar.

ControlDesk yazılımının ana penceresinin büyüklüğünü 100 pixel olarak değiştirmek için bir örnek:

```
import cdautomationlib
OBJ = cdautomationlib.Application()
OBJ.Height = 100
```

cdautomationlib Modülünün Yapısı

cdautomationlib modülü ControlDesk yazılımına erişmek için kullanılan bir modüldür. idfconverter ve cdautomationlib modüllerinin yapısı Şekil 3.16’da gösterilmiştir. Şekil 3.16’da gösterilen modül, sınıf ve koleksiyonlar Tablo 3.4’te açıklanmıştır.



Şekil 3.16 idfconverter ve cdautomationlib modül yapısı

Tablo 3.4 Cdautomationlib modülünün sınıf ve koleksiyonları

Sınıf/Koleksiyon	Görevi
Application	Application sınıfı ControlDesk'in genel özelliklerini işlemek için metod ve özellikleri sağlar. Bu özellik ve metodlar kullanılarak çalışma bölgesi penceresini düzenlenebilir ve ControlDesk'in ana penceresinin görünümü değiştirilebilir.
CodeEditor	CodeEditor sınıfı text dosyaları, C / CPP / CXX / CC dosyaları ve Python kod dosyalarını (PY) işlemek için gerekli metod ve özellikleri sağlar.
ConnectionController	ConnectionController sınıf tüm bağlantı özelliklerini işlemek için kullanılmaktadır. Bu sınıf için tanımlanan metodlar kullanılarak araçlar (instrument) ve değişkenler arasında bağlantılar eklenebilir.
ExperimentManager	ExperimentManager sınıfı deneyleri işletmek için kullanılır. Bu sınıfta tanımlanan metod ve özellikleri kullanarak, deneyleri oluşturabilir, açabilir, kapatabilir veya kaydedebilir, açık bir deneyden dosya silebilir veya dosya eklenebilir, deney dosyalarının özellikleri ayarlayabilir.
HardwareManager	HardwareManager sınıfı donanımı gerçek zamanlı işletmek için kullanılır. Uygulamalar gerçek zamanlı olarak boarda (karta) yüklenebilir veya kurulu kart listesi görülebilir.
Instrumentation	Instrumentation sınıfı layout üzerindeki araçları işlemek için gerekli metodları sağlar. Mevcut açık bir layout'un listesini alınabilir ve layoutlar silinebilir, açabilir veya oluşturulabilir. Instrumentation sınıfı kullanarak ConnectionController ve Layouts nesnelere erişilebilir ve Layouts nesnesi kullanılarak Instruments nesnelere erişilebilir.
Instruments	Instruments koleksiyonu özel araçlara erişmek, bir layout üzerindeki araç gruplarını kullanmak ve seçmek için gerekli metodları sağlar. Her bir aracın tipi için bir sınıf tanımlanmıştır. Bu sınıflar özel görünüş ve davranış tanımı için metod ve özellikleri temin etmektedir.
Interpreter	Interpreter sınıfı kod üretimi ile ilgili olarak makro kaydedicisi (Macro Recorder) ve olay işletme (Event Handling) ayarlamak/elde etmek için kullanılabilir.

Tablo 3.4 Cdautomationlib modülünün sınıf ve koleksiyonları (devam)

Sınıf/Koleksiyon	Görevi
Layouts	Layouts koleksiyonu Instrumentation sınıfı içindeki layout fonksiyonlarına erişmek için kullanılmaktadır. Bu sınıf içinde tanımlanmış metodlar kullanılarak bir layouttan taşınabilir, geri alınabilir veya eklenebilir ve bir layout penceresini kapatabilir, etkinleştirebilir veya durdurabilir. Araç veya araç gruplarını seçebilir ve sunulan layout penceresini değiştirilebilir.
ParameterEditor	ParameterEditor sınıfı parametre ve parametre dosyalarına erişmek için kullanılmaktadır. Bu sınıfta tanımlanmış metodları kullanarak parametre dosyalarını kaydetmek veya yüklemek, parametre dosyalarında parametre okumak veya parametre dosyalarına parametre yazmak gibi işlemler yapılabilir.
PlatformManager	PlatformManager sınıfı gerçek zamanlı donanım veya Simulink ortamlarını işletmek ve ortama uygulama yüklemek, uygulamaları çalıştırmak veya durdurmak için gerekli özellik ve metodları sağlar.
ReferenceManager	ReferenceManager sınıfı referans grupları ve verileri tanımlamak için kullanılmaktadır.
VariableBrowser	VariableBrowser sınıfı tanımlanan sistem dosyalarını yüklemek için kullanılmaktadır.

ControlDesk yazılımında kullanılan fonksiyonlar için Python programlama dilinin bazı sınıf, metod ve özellikleri Ek-2’de açıklanmıştır.

3.2.4. Nesne Tabanlı Programlama ve Delphi

Delphi programlama dilinin temelinde Pascal dili bulunmaktadır. Özellikle nesne yönelimli programlama anlayışıyla yapılandırılmış Turbo Pascal dilinin görsel sürümüdür. Nesne, sınıf, kalıtım, fonksiyon, aşırı yükleme (overloading) gibi temel nesneye yönelik programlama tekniklerini ve daha fazlasını içeren güçlü ve esnek bir programlama dilidir. Bilgisayar programcılığında yaşanan en önemli gelişme, programlama dillerinin yaşanan zorlukları ortadan kaldıracak bir şekilde evrim geçirmesi olmuştur. Lineer bir şekilde kodlama gerektiren program dilleri (Procedural programming) yerine organize edilmesi kolay küçük parçalardan oluşan modüler yapıdaki programlar, daha karmaşık yapıların daha hızlı ve kolay üretilmesine olanak sağlamıştır. Böylece hazırlanan bir modül farklı bir uygulamada kolayca

kullanılabilir hale gelmiştir. Bu programlama dilleri nesne tabanlı programlama dilleri (Object Oriented Programming – OOP) olarak adlandırılmışlardır. Delphi programlama dili nesne yönelimli bir dil olduğu için eklenen bütün nesnelerin (Formlar da dahil) kodlarını oluşturmaktadır. Bu kodları "Unit" ler içerisinde barındırmaktadır. Borland tarafından geliştirilen Delphi programlama dili, Win32 ve .NET platformları üzerinde yazılım geliştirmeye imkan sağlamaktadır. GNU/Linux platformu üzerinde geliştirme imkanı sağlayan Kylix isimli bir sürümü de bulunmaktadır [112–113].

Bu üstünlüklerinden dolayı Delphi programlama dili kullanılarak deneysel çalışmalarda kullanılmak üzere istemci ve sunucu arayüz programları hazırlanmıştır. Arayüz programları vasıtası ile istemci bilgisayar ile sunucu bilgisayar arasında iletişim kurulmakta ve deney düzeneğinde deneyler gerçekleştirilmektedir. İnternet üzerinden yapılan işlemlerin güvenliğinin sağlanması birinci derecede önem taşımaktadır. Geliştirilen yazılımda, sunucu bilgisayarın performansını artırmak ve kullanıcının erişimini kolaylaştırmak amaçlanmıştır.

4. İNTERNET TABANLI LABORATUARIN GERÇEKLEŞTİRİLMESİ

Günümüzde internetin gelişimi ve kullanım oranının yüksek olması eğitim alanında da uygulanabilirliğini ortaya çıkarmıştır. Mühendislik eğitimi ve teknik eğitim alanlarında web tabanlı sanal veya gerçek laboratuvarların oluşturulmasına fırsatlar vermiştir. Özellikle sürücü sistemlerin denetiminde maliyeti yüksek donanım gerektiğinden her eğitim kurumunun bu donanımı temin etmesinde de mali zorluklarla karşılaşmakta ve bu sistemlerin endüstriye uygulanabilirliği gecikmektedir. Çağdaş eğitim anlayışında, öğretim sürecinde kavramları yaparak öğrenme ve geri besleme sağlayabilmek son derecede önem kazanmaktadır. Bu da ancak eğitim kurumlarında sınırlı sayıda bulunabilen maliyeti yüksek donanımlara internet üzerinden erişerek zaman, ulaşım, sermaye gibi kavramları ortadan kaldırılarak sağlanabilmektedir. Uzaktan erişimli deneysel çalışma ortamları ile web üzerinden kaynaklar paylaşmakta, çeşitlilik ve zaman kavramlarında esneklik sağlamaktadır. Bunun yanında aynı kaynakları birçok kişi kullandığı için maliyeti de düşürmektedir. Gerek endüstri gerekse eğitim alanındaki kaynaklara/cihazlara uzaktan erişim aynı zamanda kullanıcıların ulaşım gibi sorunlarını da ortadan kaldırmaktadır.

Zaman ve parasal ihtiyaçlar birçok eğitim kurumunun sıkıntısı olmuştur. Bu sebeple benzetim çalışmaları da çok önem kazanmıştır. Benzetim çalışmaları için web tabanlı sanal laboratuvarlar oluşturularak, kullanıcılara benzetimlerini gerçekleştirmek, benzetim sonuçlarını yorumlamak, hazırladığı benzetimleri başkalarıyla paylaşmak, yapacağı benzetimlerle ilgili teorik bilgilere ulaşmak için fırsatlar verilmiştir. Yine gerçek zamanlı sistem tasarımı ve uygulama geliştirme sürecinde ortaya çıkması mümkün olan problemlerin önceden tespit edilmesi amacıyla, uygulamaya geçmeden önce benzetim ile sistemi modellemek gerekmektedir. Ayrıca öğrencilere verilen eğitimin etkileşimli benzetim modelleriyle desteklenmesi ile öğrenmenin daha kalıcı olması sağlanmaktadır.

Gerçekleştirilen bu tez çalışması iki aşamalı olarak hazırlanmıştır. Birinci aşamada öncelikle sürücü sistemlerin matematiksel modelleri doğrultusunda Matlab/Simulink benzetimleri tasarlanmıştır. Matlab/Simulink'in son yıllarda akademik ve endüstri çevrelerinde yaygın bir biçimde tercih edilmesi ve özellikle denetim sistemleri tasarımında Matlab kullanımını geleneksel hale getirenler için, MWS ortamı kullanarak internet üzerinden Simulink (mdl) dosyaları çalıştırılmıştır. Daha sonra, hazırlanan benzetim modellerine web üzerinden erişim sağlanmıştır. Tasarlanan etkileşimli web sayfaları sayesinde, kullanıcılar benzetim modeli üzerinde bulunan parametre değerlerini değiştirebilmekte ve bu değişim sonucunda sistem tepkisini izleyebilmektedirler. Web sayfalarında konuyla ilgili teorik bilgilere de yer verilmiştir. Ayrıca öğrencilerin yaparak öğrenmelerine katkıda bulunmak için, öğrencinin

benzetim bağlantıları kurarak, benzetimlerini gerçekleştirmelerine imkan sunulmuştur. Daha önemlisi Matlab/Simulink programına sahip olan kullanıcıların kendi denetim algoritmalarını denemelerine de fırsat verilmiştir.

Tez çalışmasının ikinci aşamasında ise uzaktaki kullanıcıların sunucu bilgisayar üzerindeki Matlab/Simulink tabanlı deneyleri DS1104 denetleyici kartı üzerinde yapmasına izin veren bir laboratuvar ortamı, istemci/sunucu iletişimi ile gerçekleştirilmiştir. İstemci bilgisayar ile sunucu bilgisayar arasındaki iletişim TCP/IP protokolü ile sağlanmıştır. ControlDesk Developer yazılımı ile DS1104 denetleyici kartına erişilebilmektedir. ControlDesk Developer yazılımı Python programlama dilini desteklemektedir. Dolayısıyla DS1104 denetleyici kartına erişmek için Python programlama dilinde programlar yazılmıştır. İnternet üzerinden DS1104 denetleyici kartına erişerek deneysel çalışma yapmak için MWS ortamı yetersiz kalmıştır. MWS ortamı ile DAQ veya GPIB kartlarına erişerek, bu kartlara bağlı donanımlar üzerinden ölçüm veya deneyler yapılabilir. Ancak DSP gibi kartlara erişmek için yeni yazılımlar geliştirilmelidir. Bu sebeple Delphi programlama dili kullanılarak istemci ve sunucu arayüzleri hazırlanmıştır. İstemci bilgisayar, asenkron motorun DS1104 denetleyici kartı ile bağlı bulunduğu sunucu bilgisayara erişip, DS1104 denetleyici kartı ile birlikte gelen ControlDesk yazılımını da bir ara yazılım olarak kullanarak, hız denetim deneyini gerçekleştirmekte ve gerekli parametre değişikliklerini yapabilmektedir. Python dilinde hazırlanan programlar sunucu bilgisayar üzerinde bulunmaktadır. Asenkron motoru çalıştırıp durdurması, parametre değiştirmesi ve verileri elde etmesi için, uzaktaki kullanıcının sahip olması gereken tek şey, Delphi ortamında hazırlanmış arayüz programıdır. Yine benzetim çalışmasında olduğu gibi Matlab/Simulink programına sahip olan öğrencilerin kendi denetim algoritmalarını sisteme yükleyerek sonuçlarını yorumlayabilmektedirler.

Bu çalışmada maliyetinin düşüklüğü, çeşitli ortamlarda çalışabilmesi, bakım kolaylığı ve yüksek güçlerde üretilmesi gibi üstünlüklerinden dolayı asenkron motorların denetimi için bir sanal ve uzaktan erişimli deneysel çalışma ortamı geliştirilmiştir. Asenkron motorların denetiminde yüksek performans elde etmek için motorun akı ve momentleri arasındaki kenetleme ortadan kaldırılarak birbirinden bağımsız denetlenebilir hale getirildiği vektör denetim yöntemleri tercih edilmektedir [90, 114]. Şekil 4.1’de internet tabanlı benzetim ve uzaktan erişimli deneysel çalışma laboratuvarının ana sayfası görülmektedir.



Şekil 4.1 İnternet tabanlı laboratuvarın ana sayfası

Kullanıcı adı ve şifresiyle siteye giren kullanıcı asenkron motorların hız denetimi ile ilgili benzetim çalışmalarını ve deneysel çalışmalarını yapabilmektedir. Web sayfalarında deneylerle ilgili teorik bilgilere de yer verilmiş ve etkileşimli bir eğitim sağlamak için ziyaretçi defteri, tartışma forum ortamı ve anketlerle desteklenmiştir. Yine ana sayfa üzerinden siteye üye olmayan kullanıcılar üye olabilmekte ve kullanıcılar için önemli duyurular da burada yayınlanmaktadır.

4.1. Sanal Ortamın Gerçekleştirilmesi

İnternet tabanlı laboratuvar geliştirirken kullanım kolaylığı, hız, görsel bir arayüz ve istemci bilgisayar tarafına en az iş yükü gibi etkenlere dikkat edilmelidir. Benzetimin çalıştırılabilmesi için istemci bilgisayar tarafında herhangi bir özel yazılım ya da ek bir donanıma ihtiyaç duyulmamalıdır. Benzetim arayüzünün mümkün olduğunca hızlı yüklenebilmesi, görsel fakat yorucu olmayan bileşenlerden oluşması ve aktarılmak istenen bilgilerin açık bir dille ifade edilmesi gibi özellikler eklenerek kullanım kolaylığı sağlanmalıdır.

İnternet tabanlı olarak geliştirilen benzetimlerin cevaplama süresi, internet bağlantı hızı ve sunucu bilgisayarın donanım özelliklerine bağlıdır. Bu sebeple sunucu bilgisayarın işlemcisinin ve hafızasının yüksek olması, sadece benzetimin gerektirdiği işlemleri gerçekleştirmesi ve bunun dışında herhangi bir amaçla kullanılmaması hususuna dikkat edilmelidir. Ayrıca web sayfalarının oluşturulmasında kullanılacak yazılımlar, benzetimlerin gerçekleştirilmesinde kullanılacak yazılımlar ve sunucu yazılımları birbiriyle uyumlu çalışacak şekilde seçilmelidir [83].

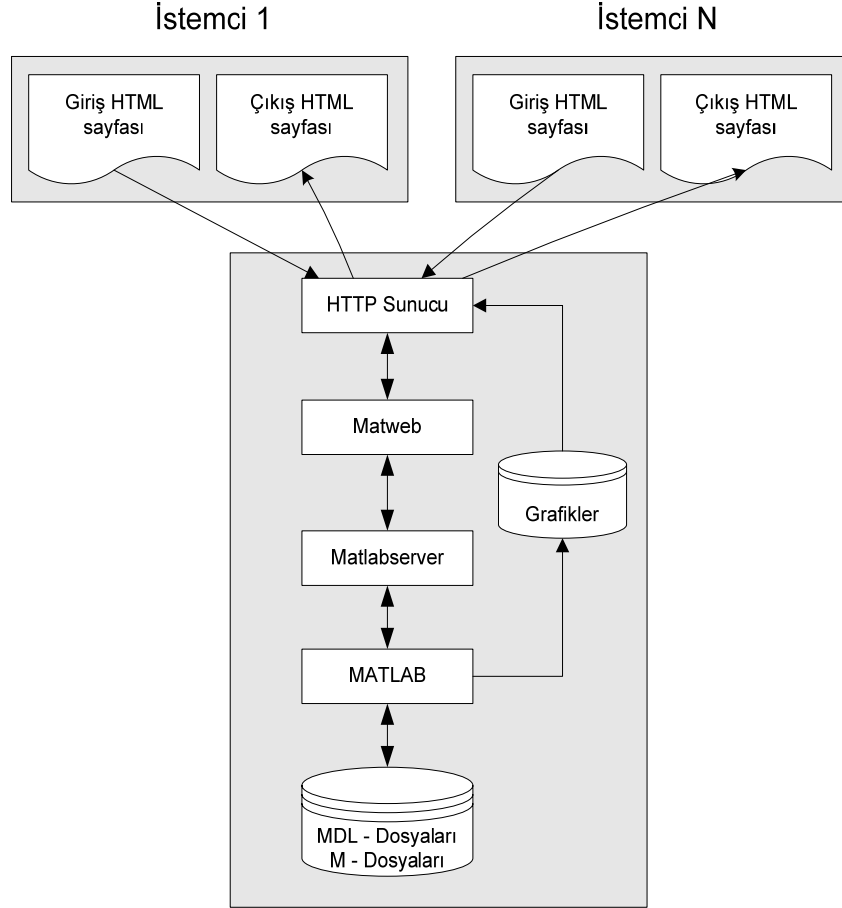
Bu tez çalışmasında geliştirilen internet tabanlı laboratuvar, yukarıda belirtilen temel özellikler göz önünde bulundurularak hazırlanmıştır. Benzetim çalışmalarında istemcinin sanal laboratuvara ulaşması için Internet Explorer gibi bir tarayıcıdan başka bir yazılım veya donanıma ihtiyacı yoktur. Bütün işlemler sunucu bilgisayarda gerçekleştirilmektedir. Kullanıcının yaparak kendi kendine öğrenmesini desteklemek, etkileşimi sağlamak ve kullanıcının daha ileri düzeyde deneyler yapmasını sağlamak amacıyla, bilgisayarında Matlab programı olan kullanıcılar hız denetleyici, akım denetleyici veya PWM üretici algoritmalarını hazırlayıp sunucu tarafına yükleyerek (upload) kendi denetleme algoritmalarını da gerçekleştirebilmektedir. İnternet üzerinden benzetim çalışmalarında kullanılan araçlardan biri olan MWS ile sanal bir laboratuvar ortamı hazırlanmıştır. MWS, Matlab uygulamalarını internet üzerinden çalıştırmaya imkan sağlamaktadır. İnternet üzerinden gönderilen veriler sunucu bilgisayarda işlenerek, sonuçlar istemci bilgisayarda görüntülenmektedir. İstemci bilgisayar ile Matlab arasındaki veri alışverişi TCP/IP protokolü ile gerçekleştirilmektedir. Sunucu bilgisayar yazılımı olarak Windows Xp işletim sistemi, sunucu bilgisayar tarafında veritabanı işlemleri için Microsoft Office Access ve web sayfalarının hazırlanmasında Active Server Page (ASP) dili kullanılmıştır. Yine sunucu bilgisayara Matlab yazılımı yüklenmiştir.

4.1.1 Sanal Laboratuvarı Gerçekleştirmede Kullanılan Yapı

Benzetimi gerçekleştirilen sistemlere ait matematiksel modellerin çözümü Matlab/Simulink ortamında gerçekleştirilmiştir. Matlab, özellikle mühendislik alanında sıkça kullanılan, üstün özellikleri olan yüksek seviyeli bir dildir. Benzetimlerdeki matematiksel modeller Matlab /Simulink ortamında çözdürülürken, her bir benzetim için ayrı bir m-dosyası hazırlanmıştır. Modele ait bağımsız değişkenler internet ortamından alınırken, bağımlı değişkenler bu m-dosyası içerisinde hesaplanmakta ve grafikler üretilmektedir.

Benzetim modellerinin internet üzerinden veri alışverişi yapabilmesi için Matlab programının bir bileşeni olan MWS kullanılmıştır. Yapısı Şekil 4.2’de verilen internet tabanlı sanal laboratuvarın çalışma şekli şöyledir; istemci bilgisayardan gelen verileri alıp, sunucu

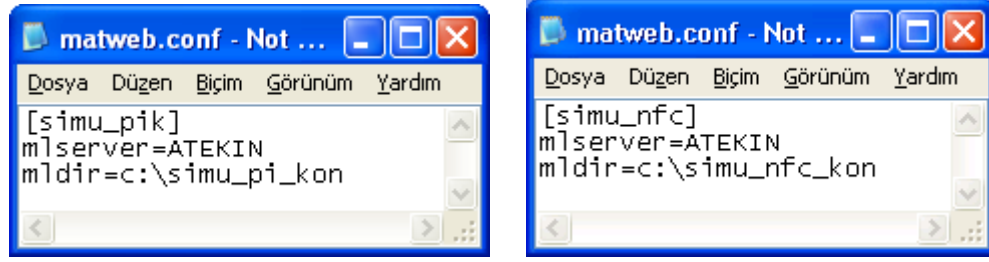
tarafındaki ilgili m-dosyası ile bu verileri işlemekte ve elde edilen sonuçları sayısal veya grafiksel olarak istemci bilgisayara geri göndermektedir [109]. MWS ile ilgili ayrıntılı bilgi Bölüm 3’te verilmiştir.



Şekil 4.2 İnternet tabanlı sanal laboratuarın yapısı

Sunucu bilgisayar tarafında bulunan “matweb” in görevi verileri almak ve “matlabserver” a iletmektir. “matlabserver” ise Matlab ile web uygulaması arasındaki iletişimi yönetmekte ve HTML belgesinde ismi belirtilen m-dosyasını çalıştırmaktadır.

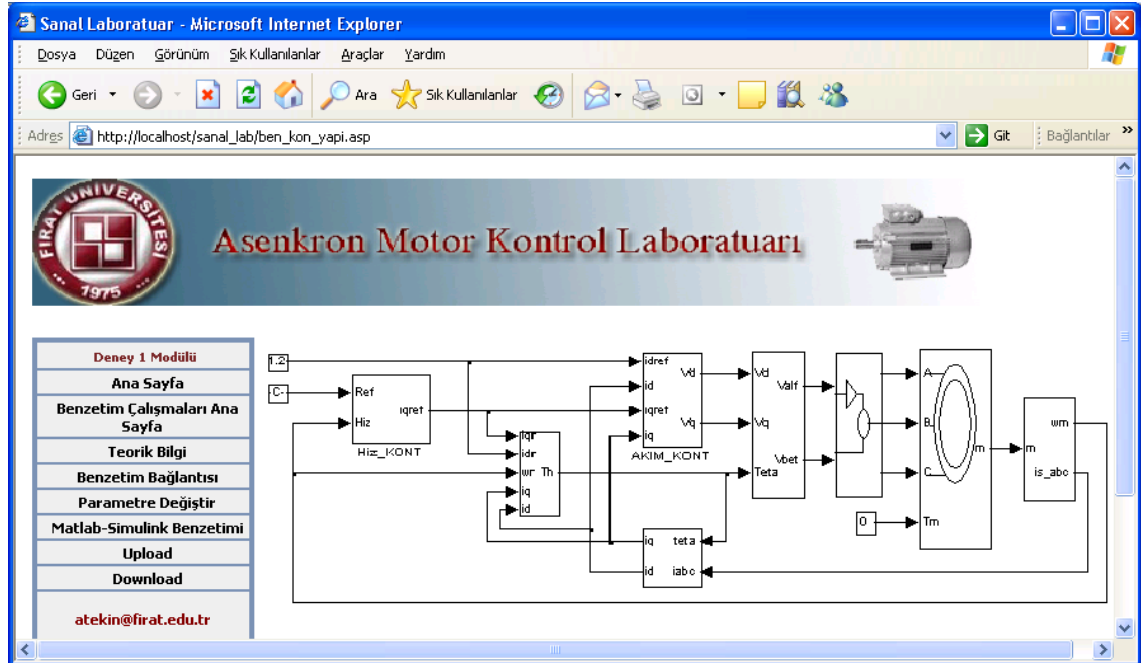
Şekil 4.2’deki yapının hata vermeden çalışması için sunucu bilgisayar tarafında CGI/BIN klasöründe bulunması gereken “matweb.conf” isimli konfigürasyon dosyasının ayarlanması gerekmektedir. Bu dosyanın içersine konfigürasyon verileri ve uygulamanın adının yazılı olması gerekmektedir. Bizim uygulamamızdaki konfigürasyon dosyasının içeriği Şekil 4.3’de verilmiştir.



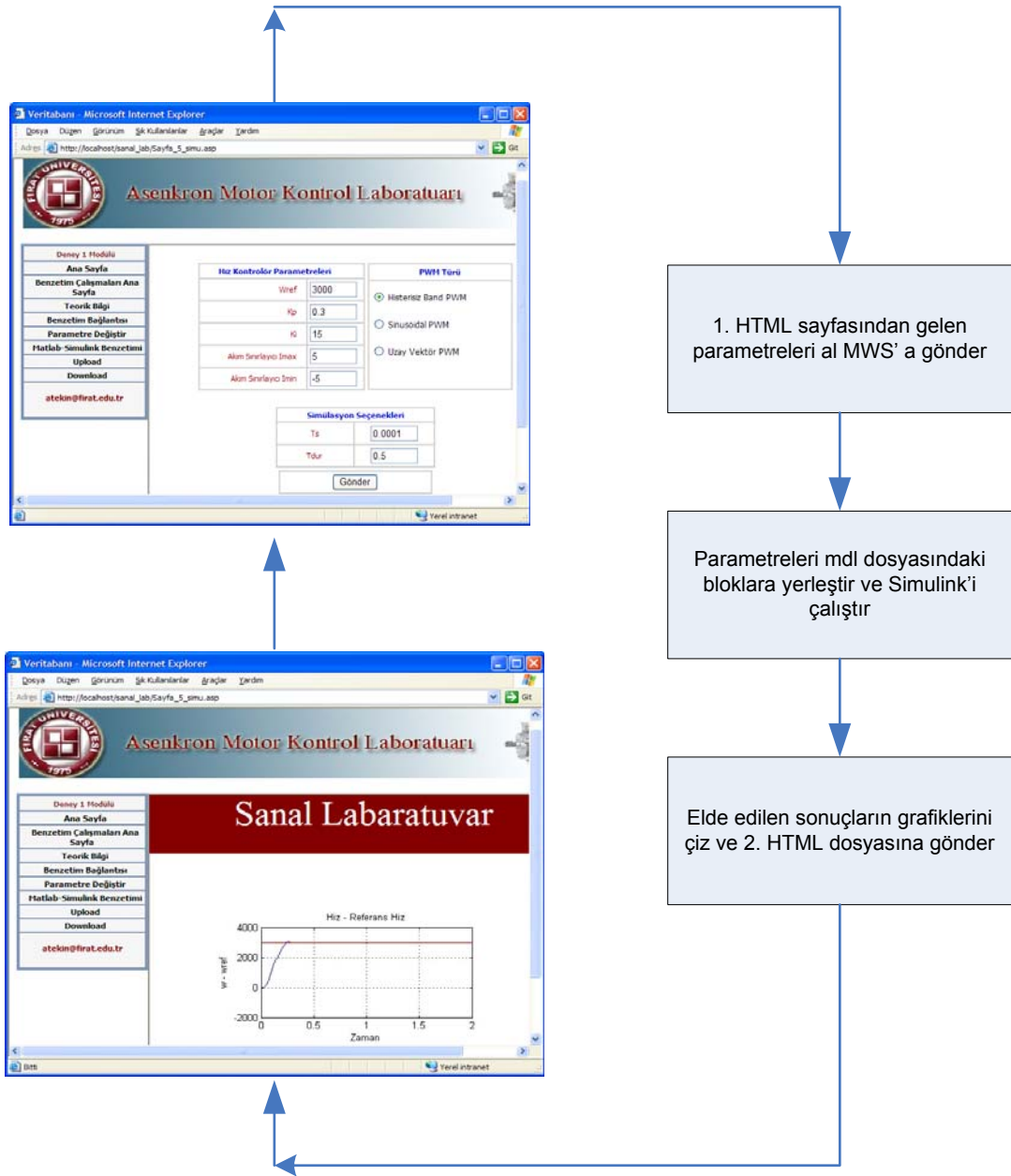
Şekil 4.3 matweb.conf isimli dosyanın içeriği

4.1.2. İnternet Üzerinden Simulink Dosyalarını Çalıştırmak

Matlab/Simulink'te çok alanlı benzetim ve model tabanlı dinamik sistem tasarımı yapılabilmektedir. Gelişmiş blok kütüphaneleri ve grafiksel ortamı sayesinde, işaret işleme ve haberleşme sistemleri gibi pek çok sistemi modelleyebilmekte, benzetimini gerçekleştirebilmekte ve gerçek zamanlı uygulamalarda kullanılabilmektedir. Son yıllarda akademik ve endüstri çevrelerinde yaygın bir biçimde kullanılan Simulink, DSP ile birlikte kullanımıyla daha yaygın hale gelmiştir. Bu gibi sebeplerden dolayı, sanal laboratuvar oluşturulurken, asenkron motorun hız denetimi için Şekil 4.4.'deki gibi bir Simulink (mdl) dosyası oluşturulmuş ve internet üzerinden bu Simulink dosyasına erişilmiştir. Şekil 4.4'de istemci web sayfasında, hazırlanan Simulink dosyası görülmektedir.



Şekil 4.4 İnternet üzerinden benzetimi gerçekleştirilen asenkron motorun PI denetim yapısının simulink benzetimi



Şekil 4.5 İnternet tabanlı sanal laboratuvarın çalışma akış şeması

Şekil 4.5'te hazırlanan web tabanlı sanal laboratuvarın çalışması ile ilgili akış şeması verilmiştir. MWS uygulamaları m-dosyaları, HTML ve grafik dosyalarının birleşiminden oluşmaktadır. İki tane HTML dosyasına ihtiyaç vardır. Birinci HTML dosyası kullanıcıdan giriş parametrelerini alarak MWS'a göndermekte ve ilgili m-dosyasını çalıştırmaktadır. m-dosyası birinci HTML dosyasından gelen parametreleri okuyarak program kodlarını çalıştırmakta ve sonuç grafiklerini oluşturmaktadır. Hazırlanan m-dosyasına, simulink benzetiminin model parametrelerini değiştirmek ve simulink benzetimini çalıştırmak için bazı kodlar eklenmiştir.

Model parametrelerini deęiřtirmek iin “set param”, simulink modelini alıřtırmak iin “sim” komut kodları kullanılmıřtır. rnek kodlar Tablo 4.1’de verilmiřtir. İkinci HTML dosyası da Matlab’den ıkıř sonularını alarak grafiksel veya sayısal olarak istemci bilgisayar internet tarayıcısında gstermektedir [109].

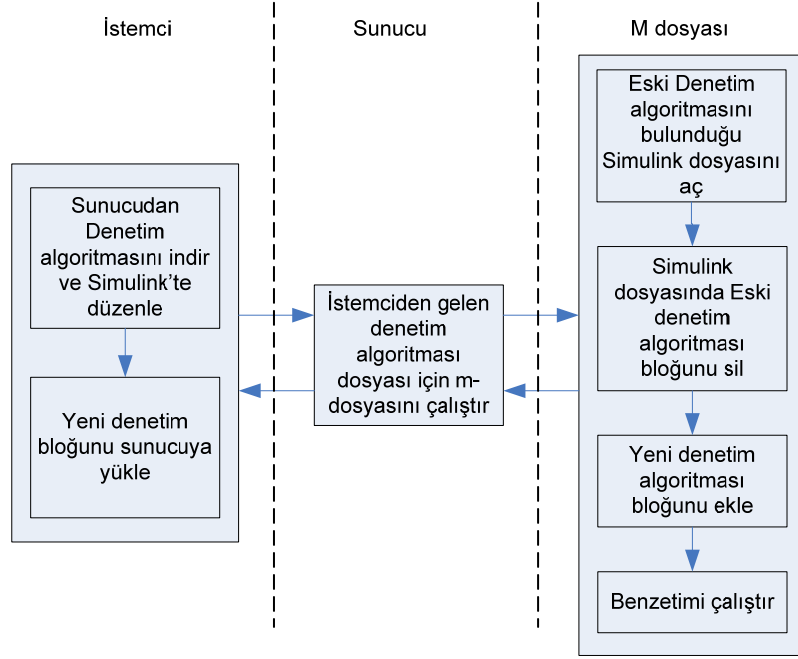
Tablo 4.1 rnek kodlar

rnek Kod	Anlamı
sim(‘model adı’,seenek);	Model adı verilen simulink dosyasını alıřtırır.
set_param(parametre,deęeri);	Parametreye deęiřkeninin deęerini atar.

řekil 4.4’deki istemci web arayznde kullanıcıya iki seenek sunulmuřtur. Birincisi, kullanıcı web arayznde verilen hız ve akım denetleyici ile PWM yntemlerindeki parametreleri deęiřtirerek benzetim alıřmasını yapabilmektedir. Parametre deęiřiklięi ve kullanıcı arayz ile ilgili aıklamalar Blm 5’te verilmiřtir. İkincisi ise, kullanıcı hız ve akım denetleyici bloklarını bilgisayarına indirerek (Download), kendi tasarladıęı denetleyici bloklarını sunucu bilgisayara ykleyip (Upload) benzetim alıřmalarını yapabilmektedir.

4.1.3. Denetim Algoritmasının Deęiřtirilmesi

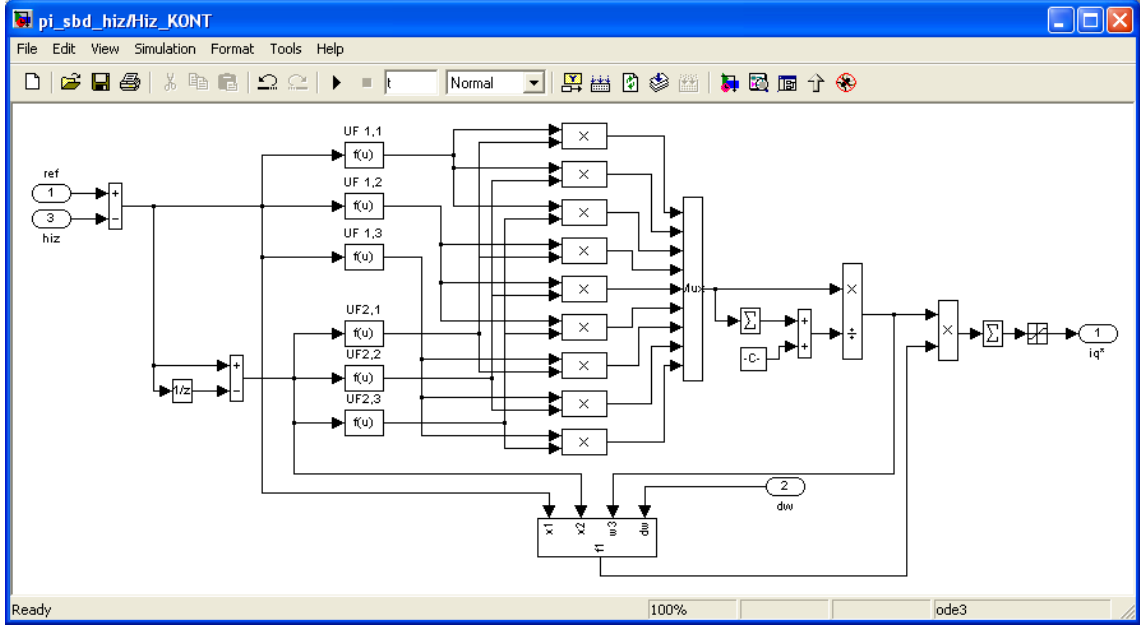
Kullanıcının gerek web sayfası ile etkileřimini artırmak gerekse yaparak ęrenmesine katkıda bulunmak amacıyla, kullanıcının kendi hazırladıęı denetim algoritmalarını da deneme fırsatı verilmiřtir. Kullanıcı iin hazırlanan web arayzndeki Upload ve Download seenekleri kullanarak, bilgisayarında Matlab programı olan kullanıcı Hız Denetleyici ve Akım Denetleyici algoritmalarını deęiřtirebilecek ve bu deęiřik algoritmalar altında alıřan sistemin sonularını web arayznden grebilecektir.



Şekil 4.6 Upload-Download işlemleri akış şeması

Şekil 4.6’da Upload – Download işlemlerinin gerçekleştirilmesini gösteren akış şeması verilmiştir. İstemci tarafındaki kullanıcı için hazırlanan web sayfasından Download seçeneğini kullanarak, mevcut sistemin denetim algoritması Simulink bloğunu bilgisayarına indirip, Matlab/Simulink programı ile denetim algoritmasını bloğunu yeniden tasarlayabilecektir. Tasarladığı yeni denetim bloğunu Upload seçeneği ile sunucu bilgisayara kaydedebilecektir. İstemciden gelen yeni denetim bloğunun işletilmesi amacıyla bir m-dosyası hazırlanmıştır. Bu m-dosyası eski denetim algoritmasının bulunduğu simulink dosyasını açarak, eski denetim bloğunu silip, yeni denetim bloğunu yerleştirecek ve benzetimi çalıştıracaktır. Elde edilen benzetim sonuçları web sayfasında görülecektir.

Şekil 4.4’deki hız ve akım denetleyici Simulink blokları kapalı bir şekildedir. Kapalı bir halde verilen SBA hız denetleyicisinin açık hali Şekil 4.7’de verilmiştir. Kullanıcı bu kapalı blokları bilgisayarına indirerek Matlab/Simulink programı ile yeniden düzenleyebilmekte veya tamamen değiştirebilmektedir.

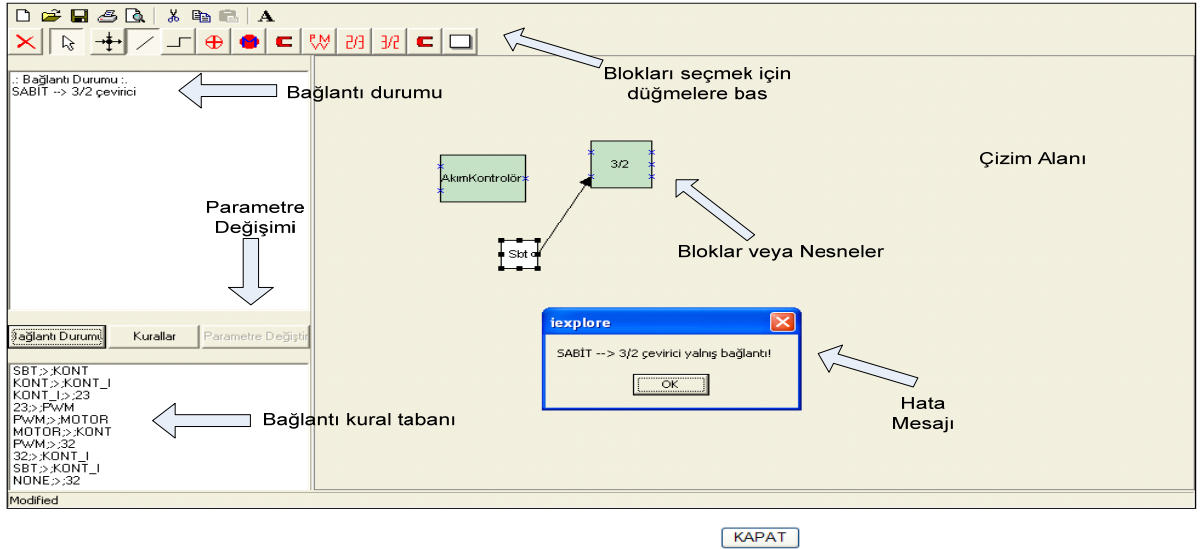


Şekil 4.7 Hız denetleyici Simulink bloğu

4.1.4. Benzetim Bağlantılarının Kurulması

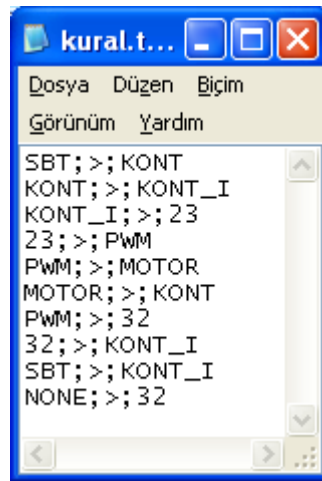
Literatürdeki birçok internet tabanlı laboratuvarlarda genellikle öğrenci kendi benzetimlerini oluşturamamakta sadece var olan benzetimlerin parametrelerini değiştirebilmektedir. Bu da öğrenciye minimum yazılım gereksinimleri sunmakta ancak öğrencinin yaparak öğrenmeyi gerçekleştirmesini engellemektedir. Çalışmamızda öğrenci, görsel bir kullanıcı arayüzüne sahip, hali hazırda barındırdığı çeşitli blok setleri ile “Sürükle ve Bırak” yöntemine dayanan, kullanımı kolay olan bir web arayüzü ile benzetim bağlantılarını oluşturabilmektedir.

Matlab programına sahip olmayan öğrencilerin benzetim bağlantılarını kurarak benzetimlerini gerçekleştirmeleri için, nesne tabanlı programlama (Object-Oriented Programming-OOP) tekniklerini kullanan, hızlı uygulamaları gerçekleştirebilen Delphi programlama dilinde arayüzler hazırlanmıştır. OOP, organize edilmesi kolay küçük parçalardan oluşan modüler yapıdaki programlar oluşturularak daha karmaşık algoritmaların daha hızlı ve kolay çözülmesine imkan vermektedir. Böylece hazırlanan bir modül farklı bir uygulamada kolaylıkla kullanılabilir. İnternet üzerinde çok sayıda açık kaynak kodları olan OOP uygulamaları bulunmaktadır [115–116]. Yazılım ekranında bulunan akım denetleyici, hız denetleyici, asenkron motor, 2/3 dönüştürücü, 3/2 dönüştürücü gibi nesneler Delphi’de tanımlanmıştır. Delphi’de hazırlanan kodlar ActivexForm’lara dönüştürülerek Web sayfalarına taşınmıştır. Şekil 4.8’de benzetim bağlantılarının yapılabildiği web sayfası görünümü verilmiştir.

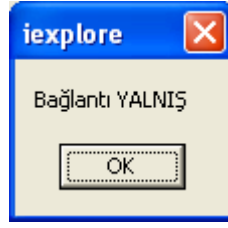


Şekil 4.8. Benzetim bağlantılarının yapılabildiği web sayfası görünümü

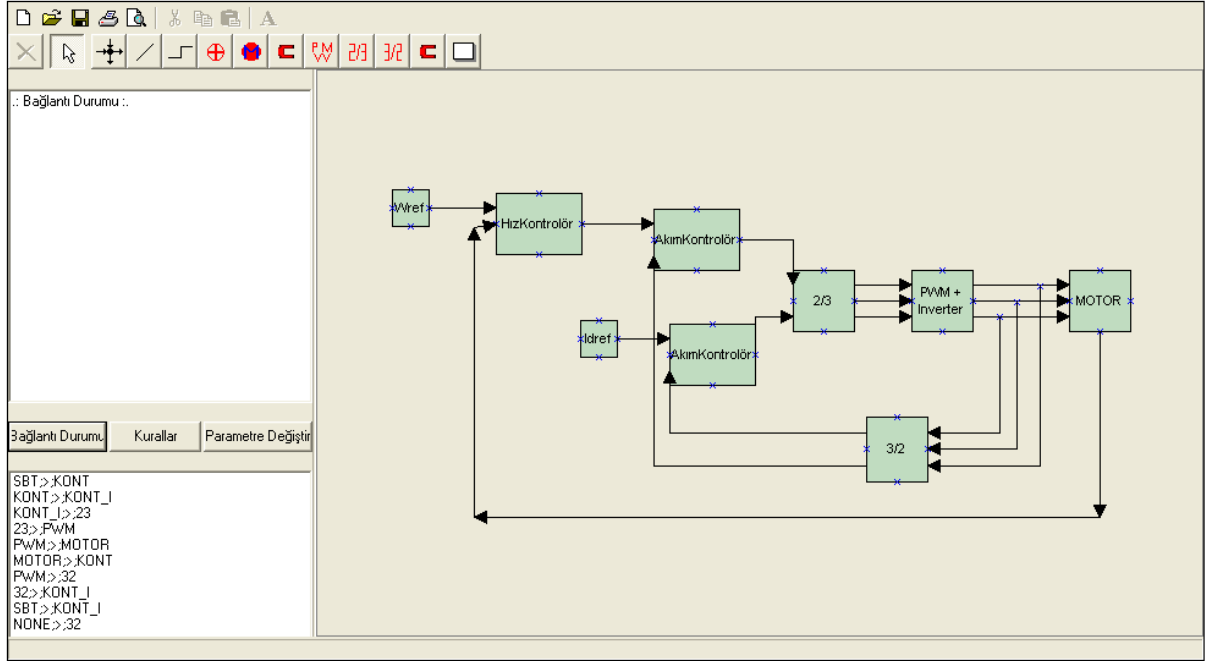
Kullanıcılar Şekil 4.8’deki grafiksel arayüzü kullanarak çok rahat bir şekilde sürükleyip bırak tekniği ile benzetim bağlantılarını kurabilmekte ve benzetimlerini internet üzerinden gerçekleştirebilmektedir. Benzetim düzeneğini kuran kullanıcı, “Kurallar” düğmesine tıklayarak bağlantı kural tabanını görmektedir. Şekil 4.9’da görüldüğü gibi kural tablosunda hangi nesnelerin, hangi nesnelerle bağlanabileceği tanımlanmıştır. Kullanıcı daha sonra “Bağlantı Durumu” düğmesine tıklayarak bağlantısının doğru olup olmadığını kontrol edebilmektedir. Kural tablosu ile kullanıcının oluşturduğu bağlantı karşılaştırılmaktadır. Yanlış bağlantı mevcutsa Şekil 4.10’daki gibi hata mesajı verilmektedir. Pasif durumda olan “Parametre Değiştir” düğmesi, benzetim düzeneği bağlantıları doğru ise aktif duruma geçmektedir. Şekil 4.11’de asenkron motorun hız denetimi için benzetim düzeneği kurulmuştur.



Şekil 4.9 Bağlantı kural tablosu



Şekil 4.10 Bağlantı durumu ile ilgili hata mesajı



Şekil 4.11 Asenkron motorun hız denetimi için benzetim düzeneği

Benzetim bağlantılarını doğru bir şekilde kuramayan kullanıcı, “Teorik Bilgi” sayfasına girerek, benzetim çalışması ile ilgili ön bilgiler alabilmektedir. Teorik bilgi sayfaları, benzetim modelleri üzerinde yapılan çalışmaların daha bilinçli bir şekilde yürütülmesine, elde edilen benzetim sonuçlarının daha sağlıklı yorumlanabilmesine ve konunun daha öğretici bir hale getirilmesine yardım etmektedir. Motor denetimi alanına yönelik hazırlanan bloklar ve görevleri Tablo 4.2’de verilmiştir.

Tablo 4.2 Bağlantı bloklarının görevleri

Blok	Görevi	Blok	Görevi
	Düz bağlantı bloğu		Denetleyici bloğu
	Köşeli bağlantı bloğu		PWM bloğu
	Yay bağlantı bloğu		2/3 dönüştürücü bloğu
	Toplama bloğu		3/2 dönüştürücü bloğu
	Motor bloğu		Sabit değer bloğu

4.2. Uzaktan Erişimli Ortamın Gerçekleştirilmesi

Deneysel çalışma ortamının gerçekleştirme aşamasında uzaktaki kullanıcıların sunucu bilgisayar üzerindeki Matlab/Simulink tabanlı deneyleri DS1104 denetleyici kartı üzerinden yapmasına izin veren bir laboratuvar ortamı hazırlanmıştır. İnternet üzerinden DS1104 denetleyici kartına erişerek deneysel çalışma yapmak için Delphi programlama dili kullanılarak istemci ve sunucu arayüzleri hazırlanmıştır. İstemci bilgisayar, asenkron motorun DS1104 denetleyici kartı ile bağlı bulunduğu sunucu bilgisayara erişip, DS1104 denetleyici kartı ile birlikte gelen ControlDesk yazılımını da bir ara yazılım olarak kullanarak, hız denetim deneyini gerçekleştirmekte ve istenilen parametre değişikliklerini yapabilmektedir. Ara yazılım olarak kullanılan ControlDesk yazılımına erişmek için, ControlDesk yazılımı ile uyumlu çalışan, Python programlama dili kullanılmıştır. Python dilinde hazırlanan programlar sunucu bilgisayar üzerinde bulunmaktadır. Uzaktaki kullanıcının asenkron motoru çalıştırıp durdurması, parametre değiştirmesi ve verileri elde etmesi için, Delphi ortamında hazırlanmış arayüz programına ihtiyacı vardır. Ayrıca görüntü beslemesi için bir IP kamera laboratuvar ortamına yerleştirilmiştir.

4.2.1 Laboratuvarı Gerçekleştirme Aşamaları

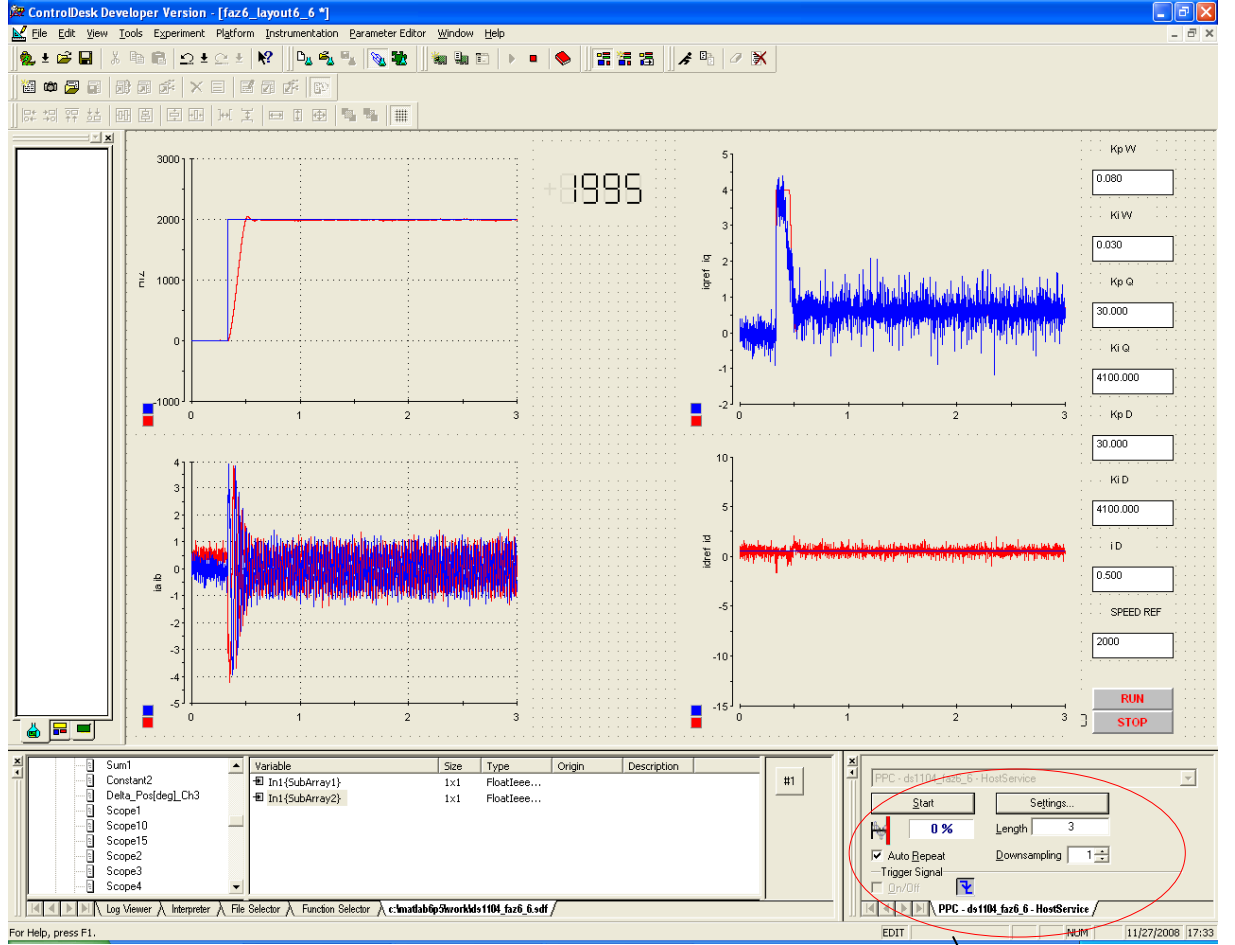
Bu çalışmada örnek olarak asenkron motorun SBD ve PI denetleyici ile hız denetimi deneyi hazırlanmıştır. Ancak Matlab/Simulink programı olan kullanıcıların geliştirdikleri denetim algoritmalarını sunucuya yükleyerek (upload) deneylerini gerçekleştirebilecekleri kullanıcı esnekliği sağlanmıştır. Uzaktan erişimli laboratuvarı oluşturma aşamaları aşağıda sıralanmıştır. Aşağıda sıralanan tüm işlemler sunucu bilgisayar tarafında gerçekleştirilmektedir.

- Matlab/Simulink programı kullanılarak asenkron motorun DSP ile hız denetimini gerçekleştirmek için Şekil 4.12'deki gibi bir simulink (mdl) dosyası oluşturulmuştur.

- Matlab/RTW programı ile simulink modeli derlenerek C/C++ kodları üretilmekte ve bu kodlar doğrudan sayısal işaret işlemciye (DSP) aktarılmaktadır.
- Bir Simulink model dosyası ile uygulamalar oluşturmak için dSPACE tarafından geliştirilen ControlDesk Developer yazılımı kullanılmıştır. Bu yazılım, algoritmanın gerçek zamanlı olarak yürütülmesi için değişkenlerin yazılım denetleyicilerine atanmasını ve bir kullanıcı arayüzü oluşturulmasını sağlamaktadır. Değişkenlerin gerçek zamanlı olarak denetimi ve gözlemlenmesi etkileşimli olarak gerçekleştirilmek üzere Şekil 4.13'teki gibi bir ControlDesk Layout arayüzü hazırlanmıştır.
- ControlDesk yazılımı, Python derleyicisini kullanmaktadır. ControlDesk yazılımının temelini oluşturan Python derleyicisi, Python kodlarını çalıştırmaya veya Python komutlarını girmeye izin vermektedir. Motoru çalıştırmak için Python dilinde programlar yazılmıştır. Bu programlar sayesinde ilgili layout arayüzü üzerinde işlemler gerçekleştirilebilmektedir. Delphi-Python-ControlDesk arasındaki ilişkiyi gerçekleştiren yazılımın akış şeması Şekil 4.15'te verilmiştir.
- Motoru durdurmak için Python dilinde programlar yazılmıştır.
- Motorun denetim parametrelerini değiştirmek ve veri elde etmek için Python dilinde programlar yazılmıştır.
- İstemci ve sunucu bilgisayar için Delphi programlama dilinde arayüz hazırlanmıştır. İstemci bilgisayardan gelen isteklere göre, sunucu bilgisayar ilgili Python programını çalıştırmaktadır.

4.2.2 Uygulama Dosyaları ve ControlDesk Layout Arayüzü

Uzaktan erişimli benzetim ve deneysel çalışma laboratuvarı geliştirilirken benzetim ortamının hazırlanmasında MWS yazılımı kullanılmıştır. MWS ortamı m-dosyaları aracılığı ile bir PC ve giriş/çıkış kartı kullanarak gerçek zamanlı uygulamalar sağlamaktadır. Ancak MWS, uzaktan erişimli olarak RTW'yi ve dolayısı ile DS1104 denetleyici kartı gibi bazı donanımların denetiminde yetersiz kalmaktadır. Gerçek zamanlı deneysel çalışmaları gerçekleştirmek ve veri alış verişini sağlamak için ek yazılım ve farklı yapılandırmalara ihtiyaç duyulmaktadır. DS1104 denetleyici kartı için, gerçek zamanlı kod üretmek ve DSP kartına yüklemek için dSpace RTI blokları kullanılmaktadır. RTI blokları, Matlab/Simulink ve DS1104 gibi fiziksel donanımlar arasında bağlantı kurmayı sağlayan bazı alt blokları içeren dört alt kütüphaneden oluşmaktadır. Bu kütüphaneler dijital-analog dönüştürücü, analog-dijital dönüştürücü, artımlı encoder arayüzü ve darbe modülasyon üniteleridir. Asenkron motorun DSP ile denetimini gerçekleştirmek için Simulink blok diyagramı hazırlanmış ve RTI bloklarının da bulunduğu yapı Şekil 4.12'de



Şekil 4.13 ControlDesk ile hazırlanan Layout arayüzü

Veri Yakalama
Araç

Şekil 4.13'teki ControDesk Layout arayüzüne yerleştirilen veri yakalama aracı ile sistemin çıkış değişkenleri elde edilebilmektedir. Çıkış değişkenlerinin ne kadar süre ile nasıl elde edileceği bu araç sayesinde belirlenmektedir. Yakalanması istenen verinin zaman uzunluğunu belirtmek için "Length" edit kutusu içerisine zaman değeri yazılabilmektedir. Yakalanacak MAT veya CSV dosya tipi gibi verinin formatı ve kayıt seçenekleri "Settings" butonu ile açılan pencerede ayarlanabilmektedir. Veri yakalamayı başlatmak için "Start" butonu kullanılmaktadır.

4.2.3 İstemci-Sunucu İletişimi İçin Geliştirilen Yazılım

Sunucu bilgisayar tarafında, sunucu hizmeti verebilmesi için Windows Xp gibi bir yazılıma ihtiyaç vardır. Bunun yanında yine sunucu bilgisayarda Matlab/Simulink ve bir Simulink model dosyası ile bir çalışma oluşturmak için dSPACE tarafından geliştirilen ControlDesk Developer yazılımının yüklenmiş olması gerekmektedir. ControlDesk yazılımı,

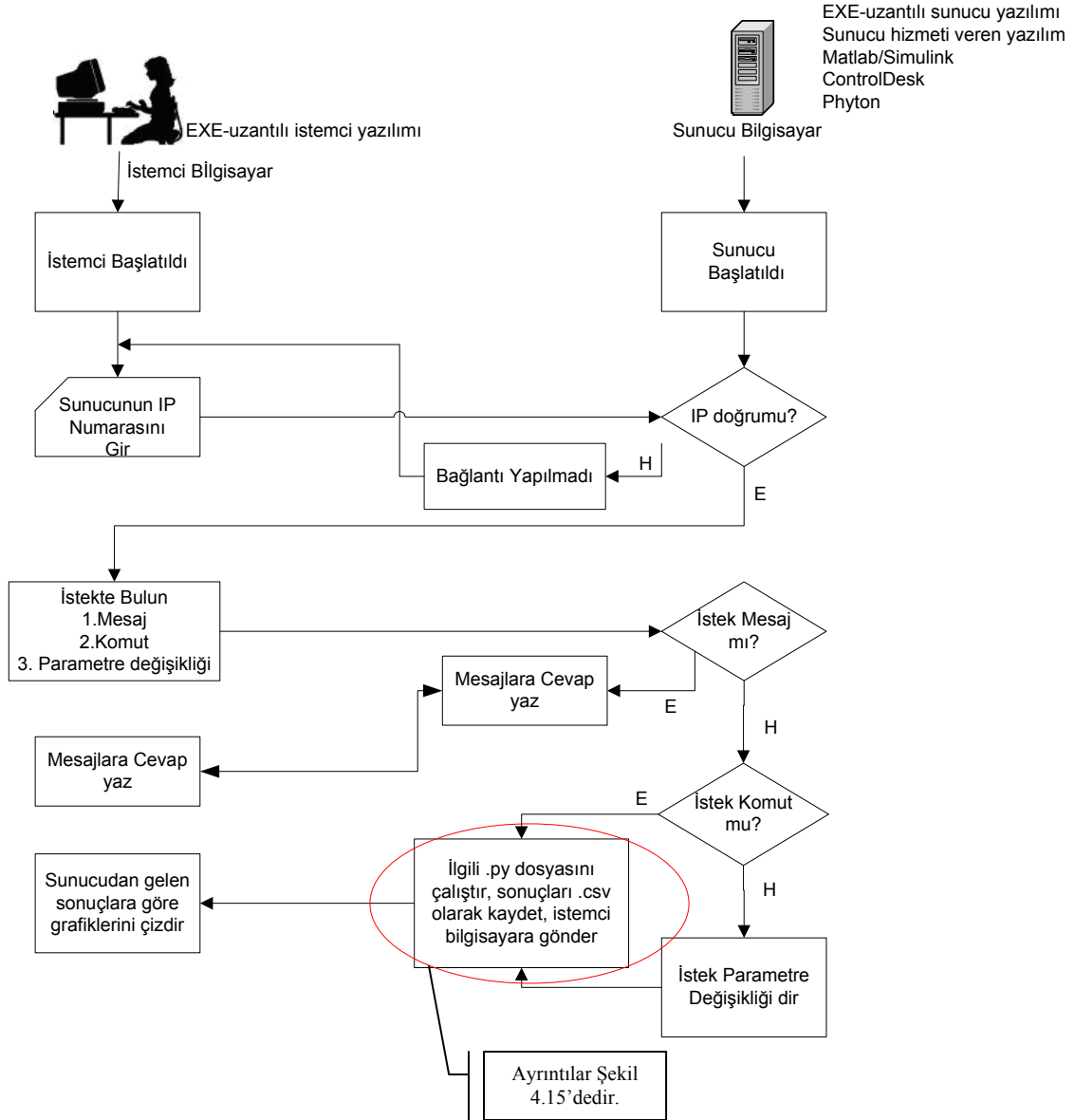
algoritmanın gerçek zamanlı olarak yürütülmesi için değişkenlerin yazılım denetleyicilerine atanmasını ve bir kullanıcı arayüzü oluşturulmasını sağlamaktadır. Böylece değişkenlerin gerçek zamanlı olarak denetimi ve gözlemlenmesi etkileşimli olarak sağlanmaktadır. Deneysel çalışmalarda denetim yapısının gerçekleştirilmesi için Matlab/Simulink ortamında modeller oluşturulmuştur. Oluşturulan bu modeller, Matlab RTW ve dSPACE' in RTI blokları sayesinde gerçek zamanlı kodlara dönüştürülerek DS1104 denetleyici kartın program hafızasına yüklenmektedir. Yapılan bu çalışmada ControlDesk yazılımına internet üzerinden Python programları ile ulaşılarak asenkron motorun hız denetim deneyi gerçekleştirilmiştir.

Sunucu bilgisayar ile istemci bilgisayar arasındaki iletişimi kurmak için arayüz programları Delphi programlama dilinde yazılmıştır. İletişim için sunucu bilgisayar üzerinde çalışan ve istemci bilgisayar üzerinde çalışan arayüz programları olmak üzere iki tane yazılım gerçekleştirilmiştir. İnternet üzerinden sunucu-istemci arasındaki haberleşmeyi, bilgi iletimi ve paylaşımını sağlamak için ağ protokolü olarak TCP/IP protokolü kullanılmıştır. Uzaktan erişimli denetim laboratuvarının erişim adresi: http://10.1.5.8/sanal_lab dır. Fırat Üniversitesi yerel alan ağı üzerinden, internet erişimi olan her bilgisayardan siteye erişilerek, çalıştırılabilir (exe uzantılı) dosya olan istemci yazılımı indirilebilmektedir. Sunucu hizmeti veren bilgisayara dış IP adresi tahsis edilmesi durumunda siteye Fırat Üniversitesi yerel ağı dışından da erişilebilmektedir. Dolayısıyla Fırat Üniversitesi dışından da kullanıcılar deneylerini yapabilmektedirler.

Şekil 4.14'te istemci bilgisayar ile sunucu bilgisayar arasında gerçekleşen olayları gösteren akış şeması verilmiştir. İstemci yazılımını bilgisayarına indiren kullanıcı, sunucu bilgisayarın IP adresini yazıp, sunucu bilgisayara bağlanabilmektedir. Bir istemci bilgisayar, sunucu bilgisayara bağlandığında, sunucu ve istemci arayüz programlarında bağlantı bilgisi görülmektedir. Sunucu bilgisayara bağlantı olana kadar, sunucu bilgisayar sürekli dinleme durumundadır. Sunucu bilgisayara aynı anda sadece bir istemci bağlanabilir.

Uzaktaki istemci bilgisayar ile sunucu bilgisayar arasındaki bağlantı gerçekleştirildikten sonra, istemci bilgisayar paketlerini sunucu bilgisayara gönderebilmektedir. Geliştirilen yazılım içerisinde parametre paketi, mesaj paketi ve komut paketi olmak üzere 3 tip paket tanımlanmıştır. Mesaj paketleri sunucu ile istemci bilgisayar arasında rahatlıkla gönderilebilmektedir. Ancak istemciden gelen paket parametre paketi veya komut paketi ise, sunucu bu sefer ilgili Python programlarını çalıştırarak ControlDesk yazılımına iletilmektedir. Kullanıcının yaptığı asenkron motorun hız denetimi deneyi esnasında yapılan çeşitli ölçüm ve hesaplamalar CSV-dosya tipinde sunucu bilgisayara kaydedilerek istemci bilgisayara gönderilmektedir. İstemci tarafına gelen sayısal sonuçlar grafiklere dönüştürülebilmektedir.

Sunucu ile istemci bilgisayar arasındaki veri alışverişlerindeki kopmalar, hatalar hem istemci hem de sunucu bilgisayar ekranından görülebilmektedir.



Şekil 4.14 İstemci ile sunucu arasında oluşan olayları gösteren akış şeması

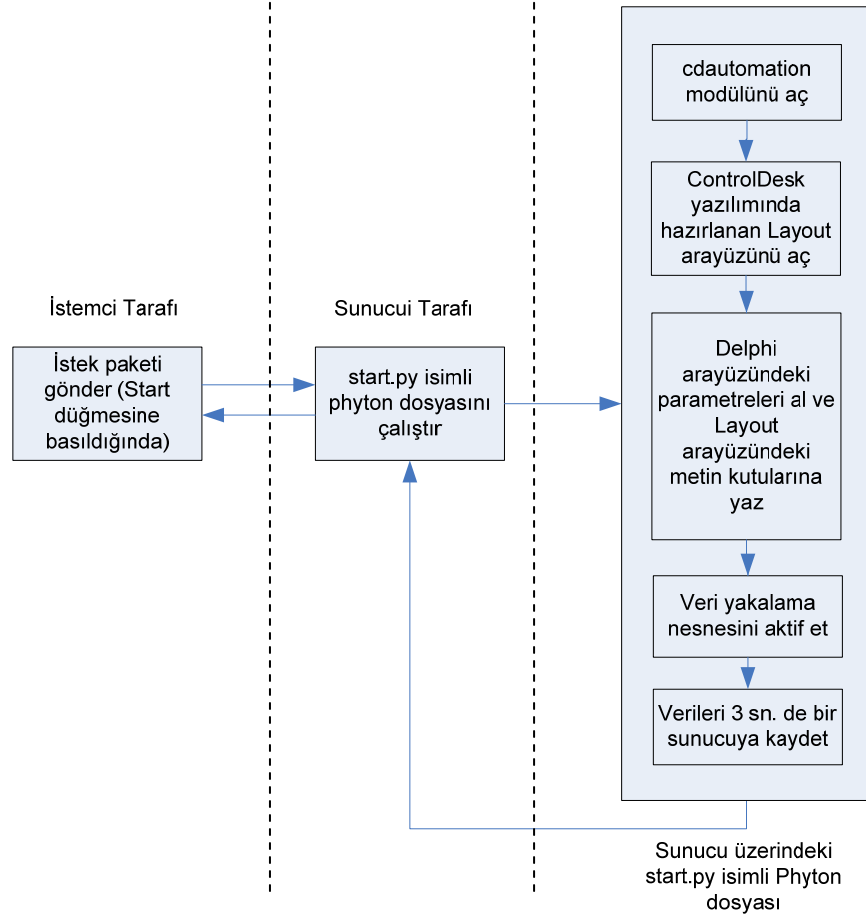
4.2.4 ControlDesk Otomasyonu İçin Geliştirilen Python Programları

Gerçek zamanlı deney esnasında ölçülen ve hesaplanan veriler sunucu bilgisayarda kaydedilmekte ve uzaktaki istemci bilgisayara gönderilmektedir. Deneysel verileri elde etmek ve deney sonuçlarını görmek amacıyla ControlDesk yazılımı, ara bir yazılım olarak

kullanılmaktadır. ControlDesk programı Python derleyicisini kullanmaktadır. Uzaktan asenkron motoru çalıştırmak, durdurmak, denetim parametrelerini değiştirmek ve veri elde etmek için Python dilinde programlar yazılmıştır. Asenkron motorun hız denetimini yapmak amacıyla uzaktaki kullanıcı, sunucu bilgisayara bir istek gönderdiğinde ControlDesk programı otomatik olarak çalışmakta ve istemcinin isteklerini gerçekleştirmektedir.

İstemciden gelen istekleri sunucunun gerçekleştirebilmesi için Delphi ile hazırlanan sunucu arayüz programında bir dizin yolu tanımlanmalıdır. Sunucu bilgisayarın kullandığı Python, Simulink, ControlDesk ve deneysel çalışma sonuçlarının bulunduğu dosyalar bu dizin yolundaki klasör içine atılmalıdır. Her bir dosya farklı klasörlere atıldığında, her bir dosya için sunucu tarafında ayrı ayrı dizin yolu tanımlamak gerekecektir. Bu da iş yükünü artıracaktır.

ControlDesk layout dosyasında hazırlanan arayüze veri yakalama aracı yerleştirilerek deneysel sonuçların kaydedilmesi sağlanmıştır. Python programlama dili ile bu veri yakalama aracını aktif etmek, animasyon ve veri yakalama işlemini eş zamanlı olarak çalıştırmak için programlar yazılmıştır. Uzaktaki kullanıcının gönderdiği komutlar ile deneyde yapılan çeşitli ölçüm ve hesaplamalara göre ControlDesk Layout arayüzünde grafikler çizilmekte ve veriler kaydedilmektedir. Veri yakalama aracı MAT ve CSV olmak üzere 2 farklı dosya tipinde verileri sunucu bilgisayara kaydedebilmektedir. Delphinin CSV-dosya tipindeki dosyaları işleyebilme özelliğinden dolayı verilerin sunucu bilgisayara CSV dosya tipinde kaydedilmesi sağlanmıştır. Sunucu bilgisayarda bu veriler işlenerek istemci bilgisayar ekranına sayısal ve grafiksel olarak gönderilmektedir. Layout arayüzüne veri yakalama aracından başka referans hız girmek ve denetleyici parametrelerini değiştirmek için metin kutuları yerleştirilmiştir.

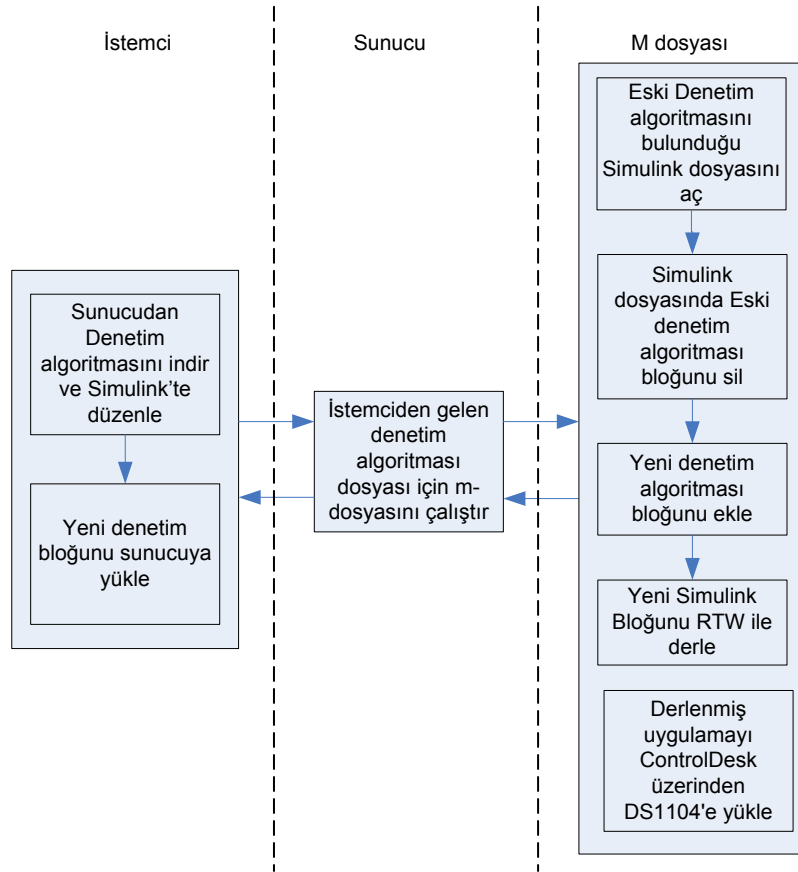


Şekil 4.15 Delphi-Python-ControlDesk arasındaki ilişkiyi gösteren akış şeması

Şekil 4.15'te ise Delphi programında hazırlanmış istemci ve sunucu arayüzleri ile Python programları arasında gerçekleşen olayları gösteren akış şeması verilmiştir. Akış şemasındaki adı geçen kütüphane, sınıf ve metodlar Ek-2'de detaylı olarak açıklanmıştır. Örneğin istemci bilgisayardan start düğmesine basıldığında, start.py isimli Python programı sunucu bilgisayarda çalıştırılacaktır. Şekil 4.15'deki akış şemasında görüldüğü gibi ControlDesk yazılımına ulaşmak için cdautomation.lib kütüphanesi açılmıştır. Donanımı gerçek zamanlı işletmek amacıyla HardwareManager sınıfı aktif edilerek, Matlab RTW ile derlenmiş simulink uygulama dosyasını DS1104 denetleyici kartına yüklemek amacıyla LoadApplication metodu kullanılmıştır. İstemci arayüzündeki metin kutularındaki parametre değerleri, ControlDesk yazılımı ile hazırlanan Layout dosyasındaki metin kutularına yerleştirilmiştir. ActiveCapture metodu ile veri yakalama aracı aktif edilmiştir. Veri yakalama aracının aktif edilmesi ile deneysel sonuçların sunucu tarafına 3 saniyede bir CSV dosya tipinde kaydedilmesi sağlanmıştır.

4.2.5 Denetim Algoritmasının Değiştirilmesi

Deneysel çalışma ortamında kullanıcının değişik denetim algoritmaları hazırlayıp, sistem üzerinde denemesine fırsat veren yazılımın akış şeması Şekil 4.16'da verilmiştir. Delphi ile hazırlanan istemci arayüzünde Download, Upload ve Derle diye üç tane düğme bulunmaktadır. Download düğmesini kullanarak sunucu bilgisayarda bulunan denetim algoritmasını bilgisayarına indiren kullanıcı Matlab/Simulink programını kullanarak denetim algoritmasını yeniden tasarlayabilecektir. Yeni denetim bloğunu sunucuya Upload düğmesini kullanarak yükleyen kullanıcı Derle düğmesi ile RTW ortamında derleyerek ControlDesk yazılımı üzerinden DS1104 kartına yükleyecektir.



Şekil 4.16 Denetim algoritmasının değiştirilmesi

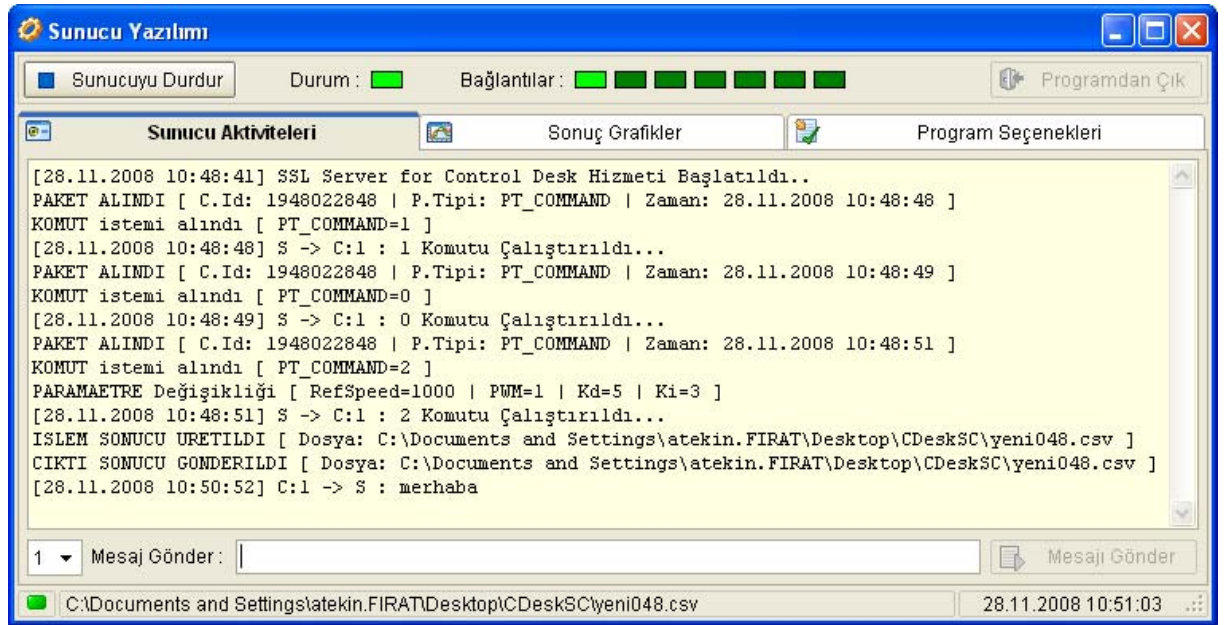
İstemciden gelen denetim algoritmasının işletilmesi için bir m-dosyası hazırlanmıştır. M-dosyası eski denetim algoritmasının bulunduğu Simulink dosyasını açarak, eski denetim algoritmasını silip, yeni denetim algoritmasını ekleyecektir. Artık yeni denetim algoritması RTW ortamında derlenerek, derlenmiş uygulama ControlDesk üzerinden DS1104'e yüklenecektir.

4.2.6 Sunucu Arayüzü

Yazılımın sunucu ve istemci taraflarında asenkron motorun hız denetimi deneyleri yapılabilmekte ve sonuçlar grafiksel olarak yorumlanabilmektedir.

Şekil 4.17’de gösterilen sunucu taraflı yazılımda;

- istemci uygulamaları için bağlantılar kabul edilebilir,
- sunucu aktiviteleri izlenebilir,
- istemci bağlantı sayısı görülebilir ve istemci bağlantı sayısı sınırlandırılabilir,
- istemciden gelen paketler yorumlanarak işletilir,
- veri alışverişi yapılabilir,
- asenkron motordan alınan veriler sayısal olarak görülebilir ve grafikleri çizdirilebilir.



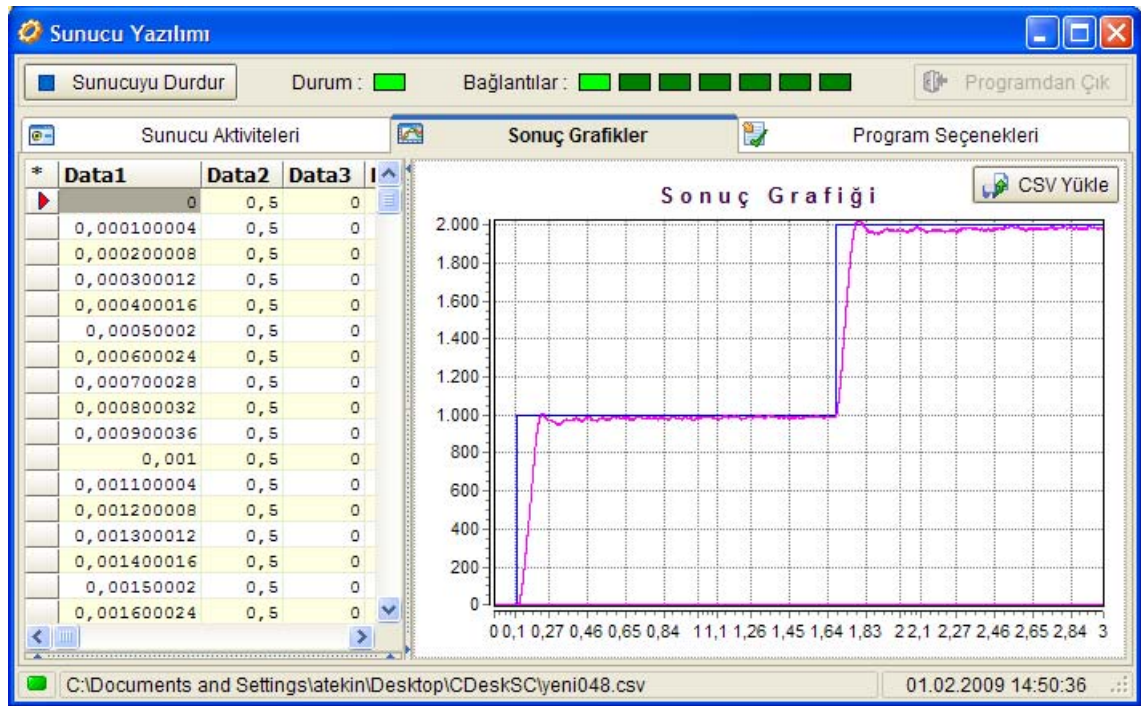
Şekil 4.17 Sunucu aktiviteleri arayüzü

Şekil 4.17’deki sunucu yazılımı arayüzünün üst tarafına bakıldığında sunucuyu başlatmak/durdurmak için bir düğme bulunmaktadır. Durum ışığının yeşil olması sunucunun hazır olduğunu kırmızı olması ise hazır olmadığını göstermektedir. Aynı anda yedi kullanıcı sunucuya bağlanabilmektedir. Bağlantı sayısı arttıkça yeşil ışık sayısı da artmaktadır. Yazılımdan çıkmak için Sunucuyu Durdur düğmesi ile önce sunucu durdurulmalı sonra Programdan Çık düğmesi ile programdan çıkılmalıdır. Sunucu yazılımı arayüzü Sunucu Aktiviteleri, Sonuç Grafikler ve Program Seçenekleri diye üç sekmeden oluşmaktadır.

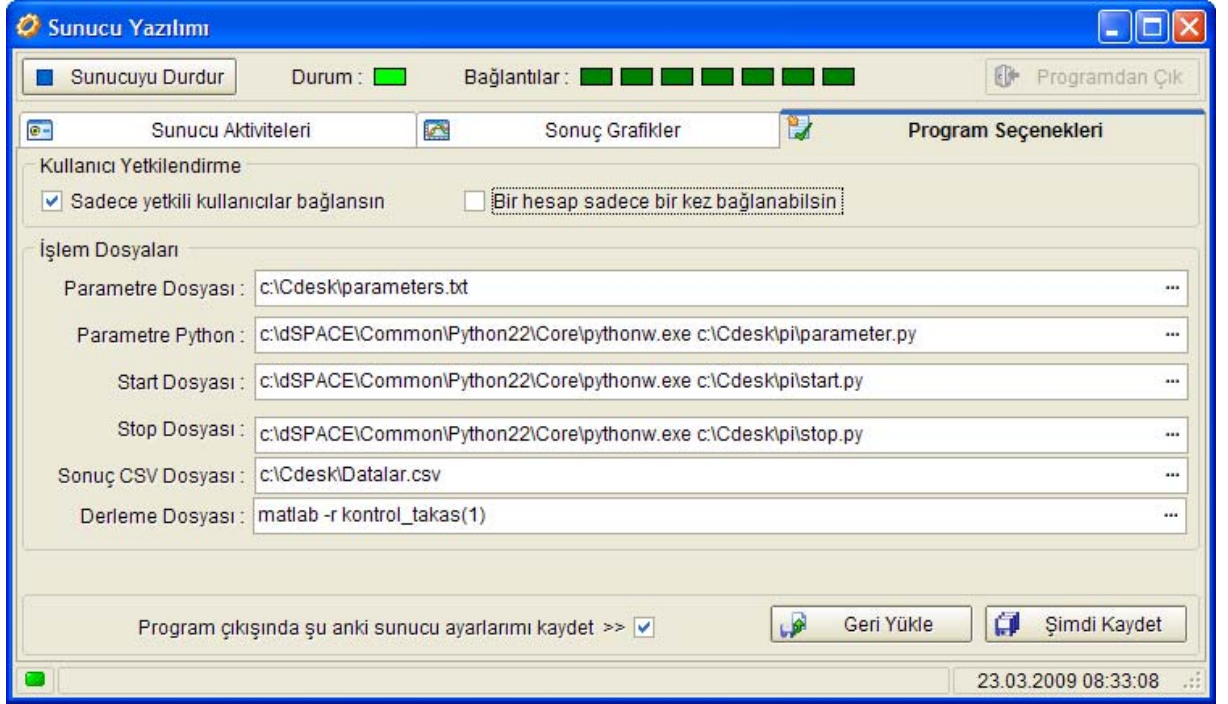
Sunucu yazılımı arayüzünün orta kısmında, Sunucu Aktiviteleri sekmesinde istemci ile sunucu arasında gerçekleşen olaylar gözükmemektedir. Örneğin;

- sunucunun hizmete başladığı tarih/saat bilgisi,
- istemcinin 1 (start/motoru çalıştır) komutunu çalıştırdığı,
- istemcinin 0 (stop/motoru durdur) komutunu çalıştırdığı,
- istemcinin 2 komutunu çalıştırarak Referans hız, Kd, Ki gibi parametreleri değiştirdiği,
- elde edilen deneysel çalışma sonuçlarının istemciye gönderildiği,
- istemciden “merhaba” mesajının geldiği Şekil 4.17’de görülmektedir.

Şekil 4.18’deki sunucu yazılımı arayüzünün orta kısmında, “Sonuç Grafikler” sekmesinde deneysel sonuçlardan elde edilen sonuçlar sayısal ve grafiksel olarak gözükmemektedir. Şekil 4.19’da ise “Program Seçenekleri” sekmesinde sunucu bilgisayarın kullandığı Python, Simulink, ControlDesk gibi dosyaların bulunduğu dizin yolları tanımlanmaktadır. Sunucu yazılımının alt kısmında ise mesaj gönderme satırı bulunmaktadır.



Şekil 4.18 Sunucu – Sonuç Grafikler arayüzü



Şekil 4.19 Sunucu – Program Seçenekleri arayüzü

5 KULLANICI ARAYÜZÜ VE SONUÇLAR

Bu bölümde, hazırlanan kullanıcı arayüzü kullanılarak asenkron motorun PI ve SBA ile hız denetiminden elde edilen benzetim ve deney sonuçları verilmiştir. Gerek benzetim, gerekse deneysel çalışmalarda, kullanıcıların sonuçları alıp değerlendirebilecekleri iki seçenek sunulmuştur. İlk seçenekte kullanıcı, hazırlanmış bağlantıları ve default olarak atanmış denetim parametrelerini değiştirerek benzetim ve deney sonuçlarını elde edebilir. İkinci seçenekte ise Matlab/Simulink programına sahip olan kullanıcılar, gerek benzetim gerekse deneysel çalışma ortamında kendi denetim algoritmalarını yerleştirerek sonuç alabilirler.

5.1. Benzetim Sonuçları ve Kullanıcı Arayüzü

Bir kullanıcı, uzaktan erişimli laboratuara bağlandığında Şekil 4.1’de verilen ana web sayfası görüntülenecek ve ana sayfanın sol tarafındaki menüden Benzetim Çalışmaları seçildiğinde benzetim çalışmaları ana sayfasına ulaşılabilecektir. Şekil 5.1’de benzetim web sayfası seçenekleri verilmiştir.

Şekil 5.1 Sanal laboratuvarın parametre değiştirme sayfası

Şekil 5.1'deki web sayfasında görüldüğü gibi Benzetim Hakkında Teorik Bilgi, Benzetim Bağlantısı, Benzetim Parametreleri, Matlab-Simulink Benzetimi, Upload ve Download gibi seçenekler bulunmaktadır. Kullanıcı, benzetim çalışması hakkında istediği bilgiye Teorik Bilgilerden erişebilecektir. Genellikle mevcut sanal laboratuarlarda, sistem düzeneği bir resim dosyası olarak görüntülenmekte öğrenciye deney düzeneğini kurabilme becerisi kazandırılmamaktadır. Benzetim Bağlantısı seçeneği ile kullanıcı kendi benzetim düzeneğini web üzerinden kurabilecektir. Benzetim çalışmaları, motorun PI ve SBA ile denetimi için hazırlanmıştır. Parametre değiştir seçeneği ile kullanıcı asenkron motor, akım denetleyici, hız denetleyici gibi parametreleri değiştirebilecek ve parametre değişimlerindeki sonuçları grafiksel olarak görebilecektir. Motorun PI denetimi ile ilgili Parametre Değiştir seçeneğinden görüntülenecek sayfalar Şekil 5.1 ve Şekil 5.2'de verilmiştir.

Deney 1 Modülü

- Ana Sayfa
- Benzetim Çalışmaları Ana Sayfa
- Teorik Bilgi
- Benzetim Bağlantısı
- Parametre Değiştir
- Matlab-Simulink Benzetimi
- Upload
- Download
- atekin@firat.edu.tr

Motor Parametreleri

P (W)	1294	Vdc (Volt)	310
V (Vrms)	380	Rs (Ohm)	8.231
F (Hz)	50	LIs (H)	0.02129
J (kgm ²)	0.0019	Rr' (Ohm)	4.49
B (Nms)	0.000263	Llr' (H)	0.02129
P	2	Lm (H)	0.5787

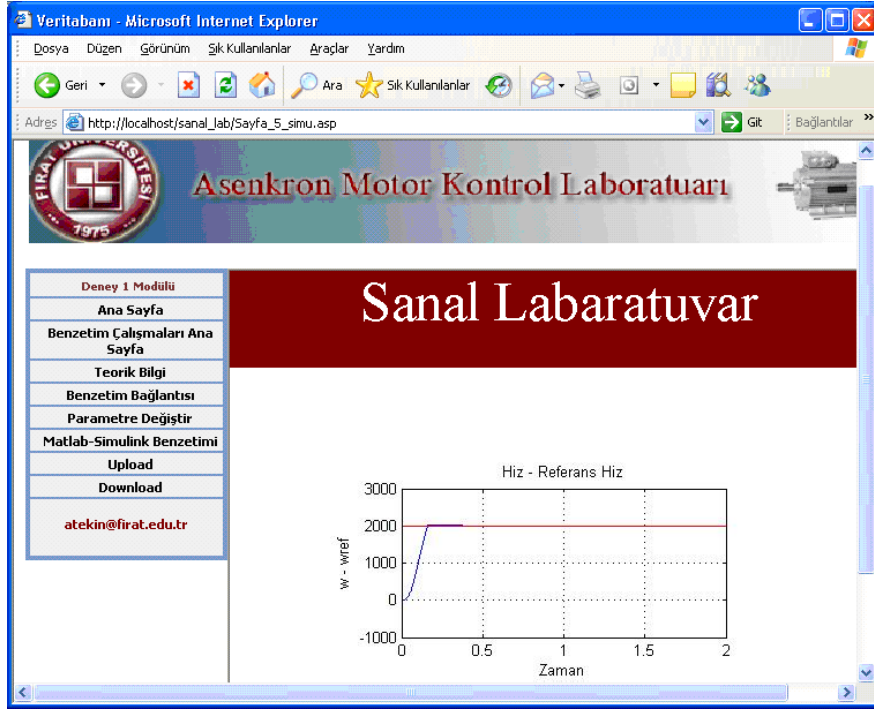
Akım Kontrolör Parametreleri

Idref	1.2		
Kp	21	Iqref_Kp	21
Ki	4200	Iqref_Ki	4200
Gerilim Sınırlayıcı Umax	220	Gerilim Sınırlayıcı Umin	-220

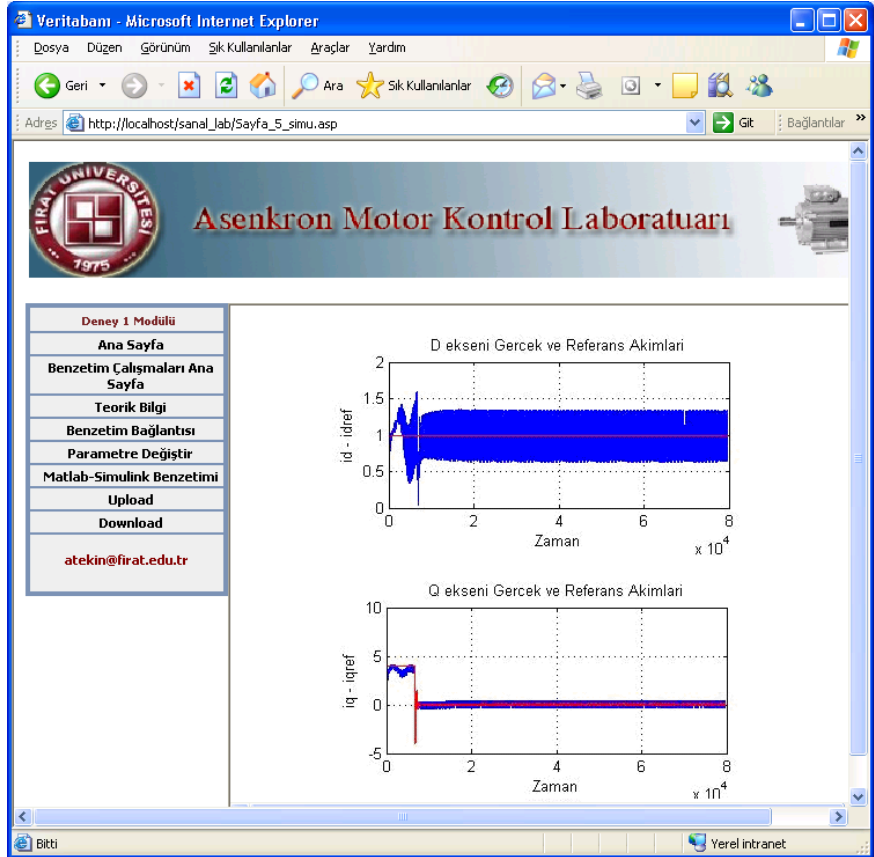
Şekil 5.2 Sanal laboratuvarın parametre değiştirme sayfası

Kullanıcı, başlangıçta sabit olarak gelen hız ve akım denetleyici ile motor parametrelerini değiştirerek ve PWM türünü seçerek benzetim sonuçlarını alabilecektir. Pratik uygulamalara uyumluluğu açısından hız ve akım denetleyicilerin çıkışları sınırlandırılabilir. Motorun aşırı gerilimleri önlemek için akım denetleme çıkışları sınırlandırılabilir, örnekleme periyodu ve benzetim süresi değiştirilebilir. Parametre değişikliklerini tamamlayan kullanıcı, “Gönder” düğmesine tıklayarak grafiksel sonuçları Şekil 5.3'te verilen web sayfası üzerinden görebilmektedir.

Şekil 5.3'te, motorun tasarlanmış bir PI hız ve akım denetleyicileri ile elde edilen sabit bir referans hızı izleme performansı, Şekil 5.4'te ise dq akımları izleme performansı verilmiştir.



Şekil 5.3 PI denetleyici ile referans hız ve motor hızının değişimi



Şekil 5.4 PI denetleyici ile d ve q eksen gerçekte ve referans akımları

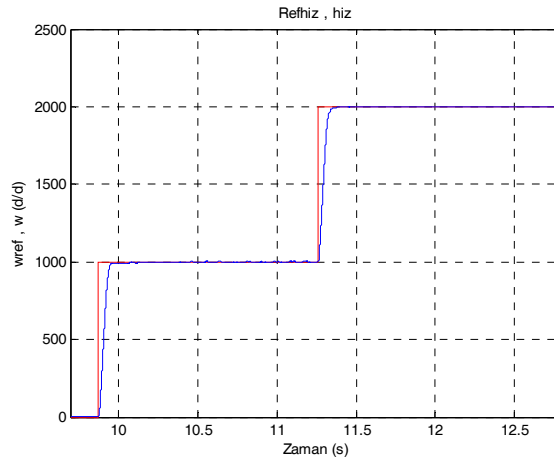
Motorun SBD ile ilgili Parametre Değiştir seçeneğinden elde edilen web sayfası Şekil 5.5'te verilmiştir. Web sayfasında görüldüğü gibi motor parametreleri, akım denetleyici parametreleri, hız denetleyici parametreleri ve SBD ile ilgili parametreler değiştirebilmektedir.

Motor Parametreleri				Hız Kontrolör		Akım Kontrolörleri	
P (W)	1294	Vdc (Volt)	310	Wref	2000	Idref	1
V (Vrms)	380	Rs (Ohm)	8.231	Akım Sınır. Imax		Idref_Kp	21
F (Hz)	50	Ls (H)	0.02129	Akım Sınır. Imin		Idref_Ki	4200
J (kgm^2)	0.0019	Rr' (Ohm)	4.49	Gerilim Sınır. Umax	286	Iqref_Kp	21
B (Nms)	0.000263	Llr' (H)	0.02129	Gerilim Sınır. Umin	-286	Iqref_Ki	4200
P	2	Lm (H)	0.5787	Ts	0.00005	Tdur	2

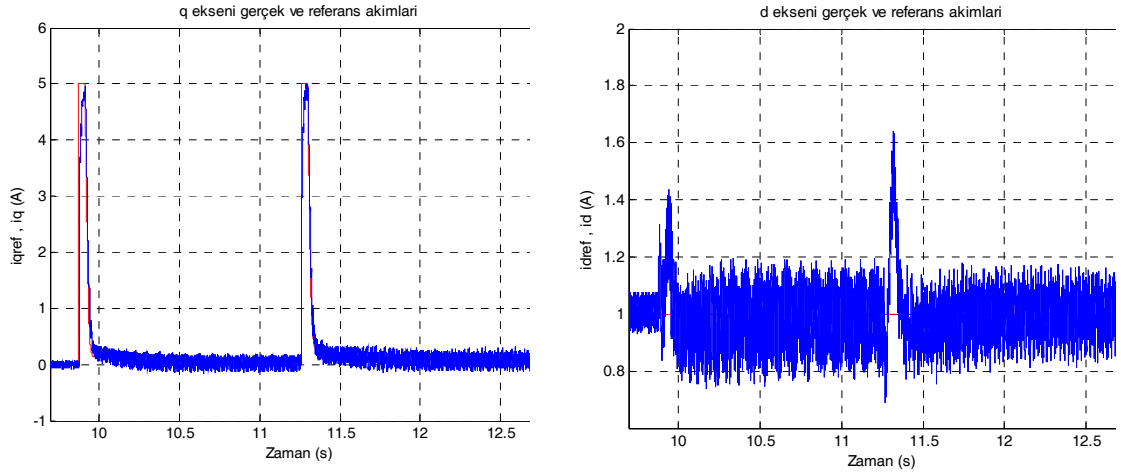
SBD Ayarları	
Eğitim Seçenekleri	
Öğrenme Oranı	0.01
Üyelik Fonk. Sayısı	3
SBD Kontrolör	
☐ PINFC ☒ PD NFC ☐ PD-I NFC	
Gönder	

Şekil 5.5 SBD için sanal laboratuvarın parametre değiştirme sayfası

Şekil 5.6'da, motorun tasarlanmış bir SBA hız denetleyici ile elde edilen sabit bir referans hızı izleme performansı, Şekil 5.7'de ise dq akımları izleme performansı verilmiştir.

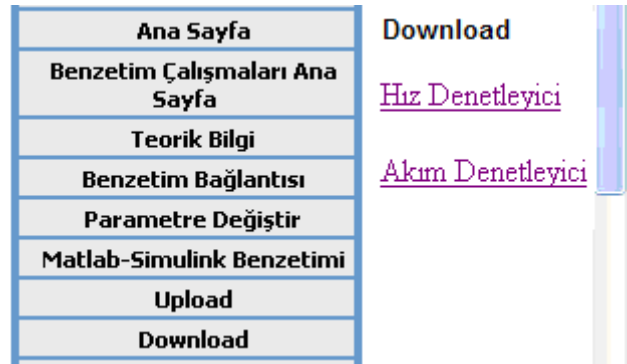


Şekil 5.6 SBD denetleyici ile referans hız ve motor hızının değişimi



Şekil 5.7 SBD denetleyici ile d ve q eksenini gerçek ve referans akımları

Mevcut sanal laboratuvarlarda, yukarıda verildiği gibi kullanıcılar web üzerinden sistemlerin parametrelerini değiştirerek sonuçlarını izleyebilmektedir. Sanal laboratuvarlardaki önemli eksikliklerden biri kullanıcı kendi denetim algoritmalarını deneyememektedir. Şekil 5.1’deki download ve upload seçeneklerinden görüldüğü gibi bu tez çalışmasında, bilgisayarında Matlab programı olan kullanıcı, hazırlanmış hız denetleyici ve akım denetleyici algoritmalarını kendi bilgisayarına indirerek (download) bu algoritmaları değiştirebilecek ve sunucu bilgisayara yükleyerek (Upload) kendi çalışmalarını web arayüzünden görebilecektir. Bu esneklik, kullanıcıya yaparak öğrenmede katkıda bulunacak ve öğrencinin etkileşimini artıracaktır. Şekil 5.8’de download ve upload seçeneğinin kesit sayfaları verilmiştir.



(a)

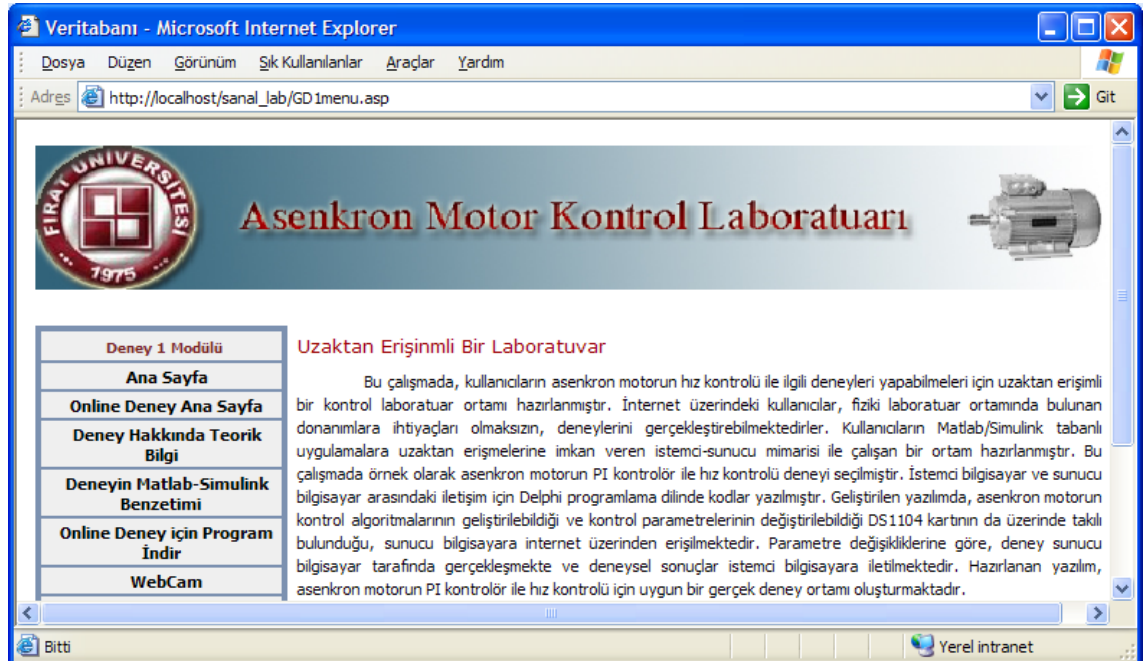
Ana Sayfa
Benzetim Çalışmaları Ana Sayfa
Teorik Bilgi
Benzetim Bağlantısı
Parametre Değiştir
Matlab-Simulink Benzetimi
Upload
Download

Upload

(b)
Şekil 5.8 Download (a) ve Upload (b) sayfalarının kesit görünümü

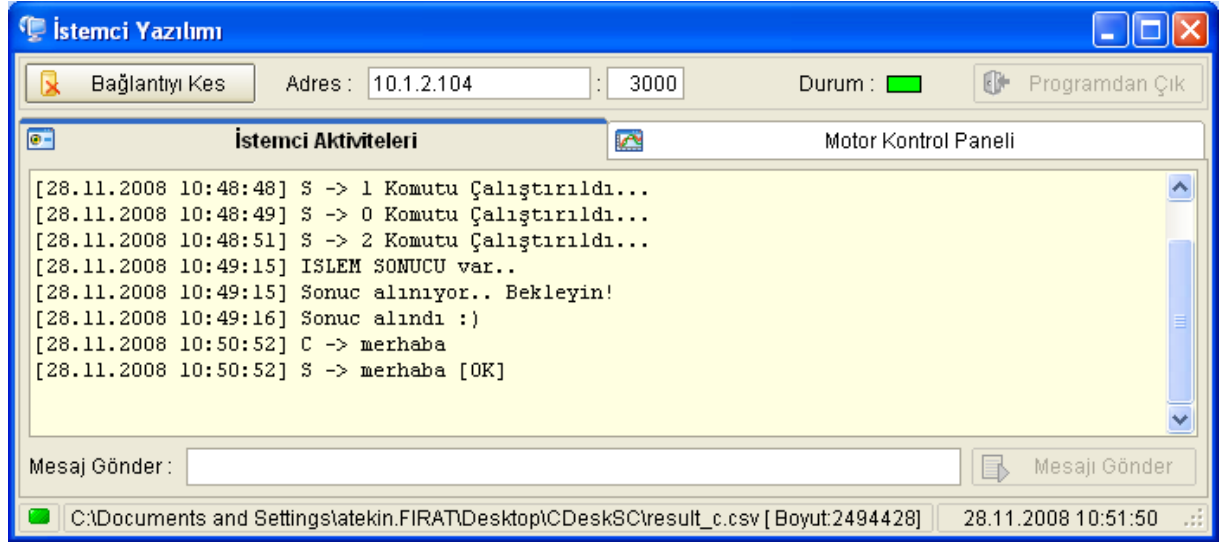
5.2. Deneyel Sonuçlar ve Kullanıcı Arayüzü

Bu tez çalışmasında asenkron motorun PI ve SBA ile hız denetimi Bölüm 3’te belirtilen donanım gerçekleştirimi ile DS1104 denetleyici kartı kullanılarak yapılmıştır. Yapılan deneysel çalışmalar sonucunda elde edilen veriler grafiksel olarak geliştirilen Delphi arayüzünden alınmaktadır. Benzetim çalışmalarını tamamlayan kullanıcı uzaktan erişimli laboratuvarın ana sayfasının sol tarafındaki menüden Online Deney seçeneğini seçtiğinde Şekil 5.9’daki web sayfası görüntüsü ile karşılaşacaktır.



Şekil 5.9 Online Deney web sayfası

Kullanıcı bu web arayüzü ile deney hakkında teorik bilgi alabilecek, Matlab/Simulink ile hazırlanmış deney şemasını inceleyebilecek ve deney düzeneğini bir IP kamera ile izleyebilecektir. Kullanıcının internet üzerinden deneylerini yapabilmesi için bu web sayfasından Delphi programlama dili ile hazırlanmış arayüz programını indirmelidir. Şekil 5.10'da istemci tarafı arayüzü verilmiştir.



Şekil 5.10 İstemci aktiviteleri arayüzü

Şekil 5.10'da gösterilen istemci tarafı yazılımda;

- uygulamalar için sunucuya bağlanılabilir,
- veri alışverişi yapılabilir,
- sunucudan gelen verilerin grafikleri çizdirilebilir,
- istemci aktiviteleri izlenebilir,
- parametre paketleri, komut paketleri veya mesaj paketleri gönderilebilir.

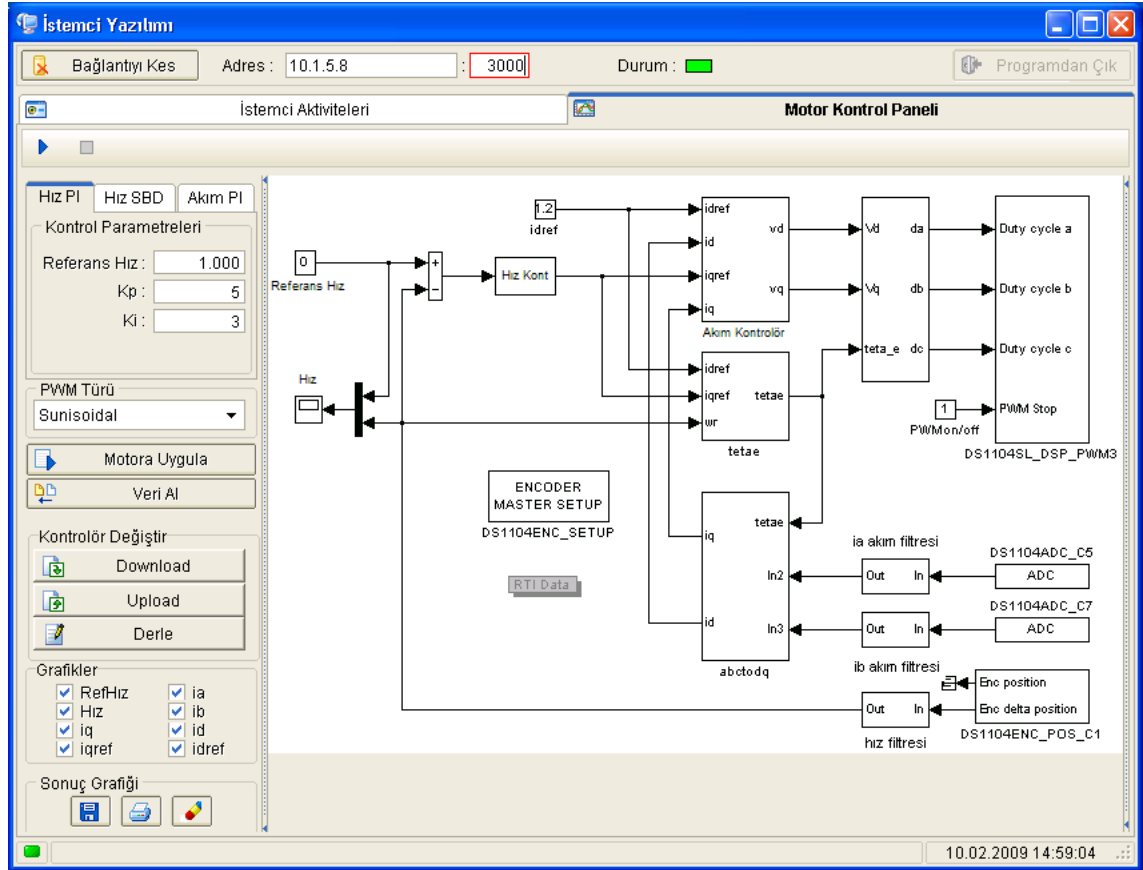
Şekil 5.10'daki istemci yazılımı arayüzünün üst tarafına bakıldığında sunucuya bağlanmak/bağlantıyı kesmek için bir düğme bulunmaktadır. Sunucu bilgisayarın IP adresini ve port numarasını yazmak için iki tane metin kutusu bulunmaktadır. Durum ışığının yeşil olması sunucuya bağlanıldığını göstermektedir. Yazılımdan çıkmak için önce bağlantı kesilmesi sonra Programdan Çık düğmesi ile yazılımdan çıkılmalıdır. İstemci yazılımı arayüzü İstemci Aktiviteleri ve Motor Kontrol Paneli diye iki sekmeden oluşmaktadır.

İstemci yazılımı arayüzünün orta kısmında, İstemci Aktiviteleri sekmesinde istemci ile sunucu arasında gerçekleşen olaylar görülmektedir. Örneğin;

- istemcinin 1 (start/motoru çalıştır) komutunu çalıştırdığı,
- istemcinin 0 (stop/motoru durdur) komutunu çalıştırdığı,

- istemcinin 2 komutunu çalıştırarak parametre değiştirdiği,
- elde edilen deneysel çalışma sonuçlarının sunucudan alındığı,
- sunucuya “merhaba” mesajının gönderildiği Şekil 5.10’da görülmektedir.

İstemci yazılımı arayüzünün orta kısmında, Motor Kontrol Paneli sekmesi Şekil 5.11’de verilmiştir.



Şekil 5.11 İstemci – Motor Kontrol Paneli arayüzü

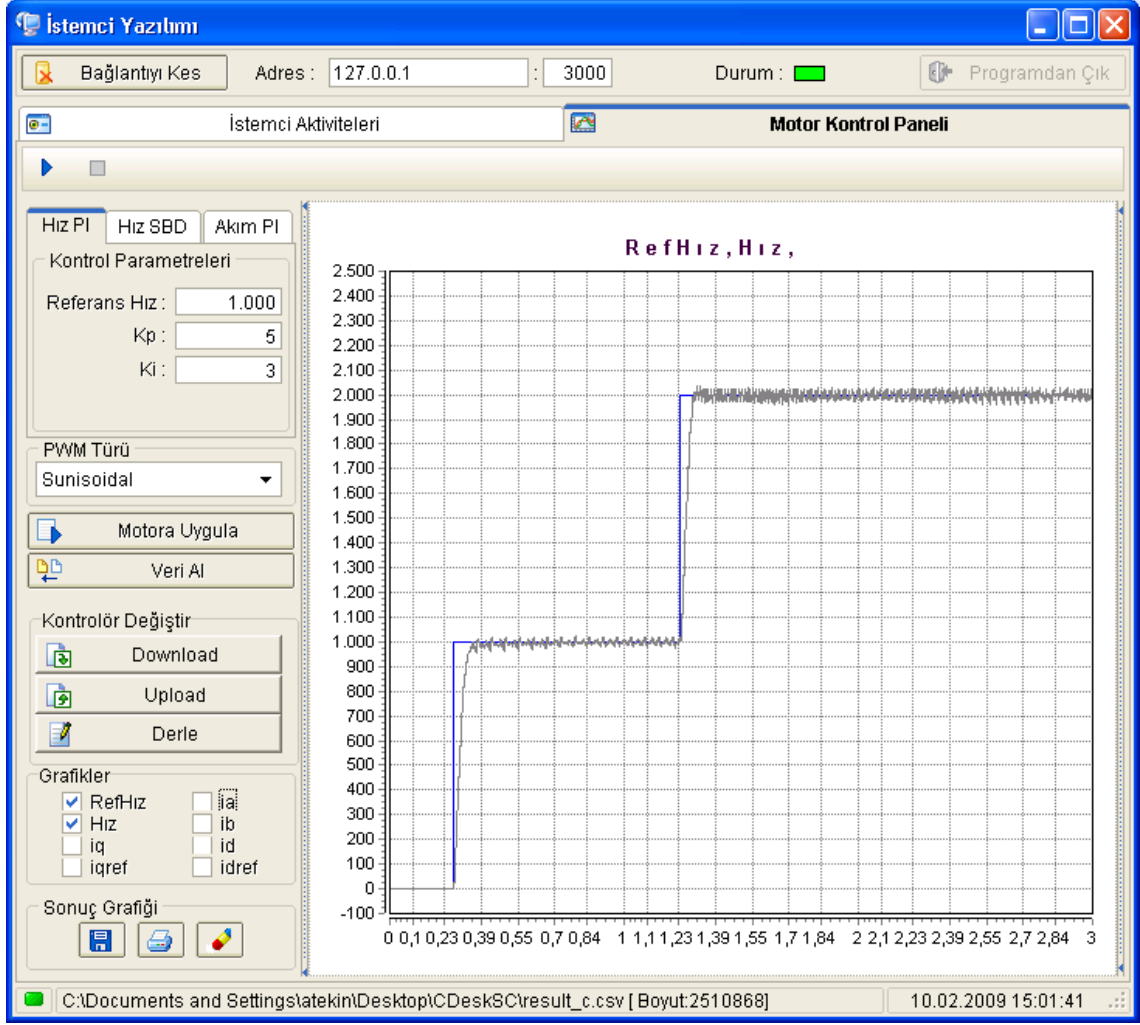
Sekmenin üst tarafında start/stop düğmeleri ile motor çalıştırılıp/durdurulmaktadır. PWM türü seçeneğinde sinüsoidal modülasyon ile akım denetimi yapılabilmektedir. Sekmenin sol tarafında hız denetleyici, PI ya da SBD olarak, akım denetleyici ise PI olarak seçilebilmekte ve ilgili metin kutuları yardımıyla parametre değişiklikleri yapılabilmektedir. Motora Uygula düğmesine tıklandığında parametre bilgileri sunucuya iletilmekte ve kullanıcının belirlediği parametrelerle deney gerçekleştirilmektedir. Şekil 5.12’de denetleyicilerin parametre değiştirme panelleri verilmiştir.

The image displays three side-by-side panels for configuring control parameters. Each panel has tabs for 'Hız PI', 'Hız SBD', and 'Akım PI'. The first panel (Hız PI) shows 'Referans Hız' as 1.000, 'Kp' as 5, and 'Ki' as 3. The second panel (Hız SBD) shows 'Çıkarım Yönt.' as Sugenc, 'Üyelik Fonk.' as Gauss, 'Ü.Fonk.Sayısı' as 3, and 'Öğrenme Oranı' as 0,001. The third panel (Akım PI) shows 'IdRef' as 1, 'Idref_Kp' as 21, 'Idref_Ki' as 4.200, 'Iqref_Kp' as 21, and 'Iqref_Ki' as 4.200.

Panel	Tab	Parametre	Değer
1	Hız PI	Referans Hız	1.000
		Kp	5
		Ki	3
2	Hız SBD	Çıkarım Yönt.	Sugenc
		Üyelik Fonk.	Gauss
		Ü.Fonk.Sayısı	3
		Öğrenme Oranı	0,001
3	Akım PI	IdRef	1
		Idref_Kp	21
		Idref_Ki	4.200
		Iqref_Kp	21
		Iqref_Ki	4.200

Şekil 5.12 Denetleyicilerin parametre değiştirme panelleri

Kullanıcının değişik denetim algoritmalarını denemesi için sol tarafta Denetim Değiştir diye bir kısım tanımlanmıştır. Burada Upload, Download ve Derle isimlerinde düğmeler bulunmaktadır. Matlab/Simulink yazılımına sahip olan kullanıcı bu düğmeleri kullanarak hız ve akım denetleyicileri bilgisayarına yükleyerek kendi denetim algoritmalarını geliştirebilecek ve kendi algoritmasını kullanarak deneyini yapabilecektir. Upload ve Download ile ilgili işlemler Bölüm 5.1’de verilmiştir. Derle seçeneği ile kullanıcının sisteme yerleştiği algoritması ile uygulama dosyası derlenerek DSP’ye yüklenmektedir. Veri Al düğmesi ile deneysel çalışma sonuçları grafiksel olarak alınmakta ve istenilen grafikler Şekil 5.13’teki gibi çizdirilebilmektedir. Uzak bir yerden sunucuya bağlanıp paket göndermek yaklaşık 1-5 sn. gibi çok küçük bir zaman almaktadır. Deneyin gerçekleştirilmesi ve verilerin istemciye ulaştırılması yaklaşık 5-10 sn. arasındadır.



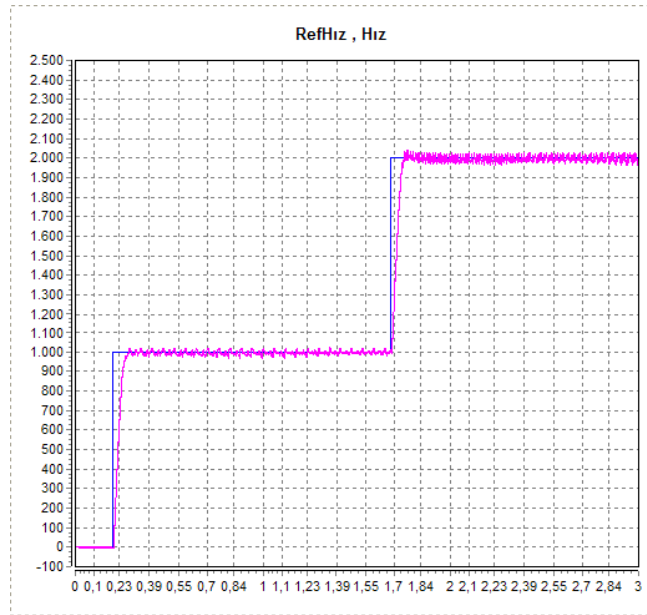
Şekil 5.13 İstemci –Grafikler arayüzü

İnternet üzerinden deneysel çalışmalar gerçekleştirilirken, deney seti ve bu setin kurulu olduğu laboratuvar ortamı bir IP kamera ile izlenebilmektedir. Çalışmada, yüksek hız ve çözünürlüğe sahip bir ağ kamerası kullanılmıştır. Bu sayede deney seti izlenebilmektedir. IP kamera sistemden bağımsız çalışmakta ve elde ettikleri görüntüleri doğrudan internet ortamına aktarmaktadırlar. Bu sayede, hem görüntü aktarım hızının yüksek olması hem de sunucu üzerindeki iş yükünün azaltılması sağlanmıştır. IP kamera asenkron motoru yakından görüntülemektedir. Böylelikle uzaktan deneyi yapan bir kullanıcı, asenkron motorun hareketlerini ve dönüp dönmediklerini izleyebilmektedirler. Deneysel çalışmalar sırasında IP kamera vasıtası ile elde edilen bazı görüntüler Şekil 5.14’te verilmiştir.

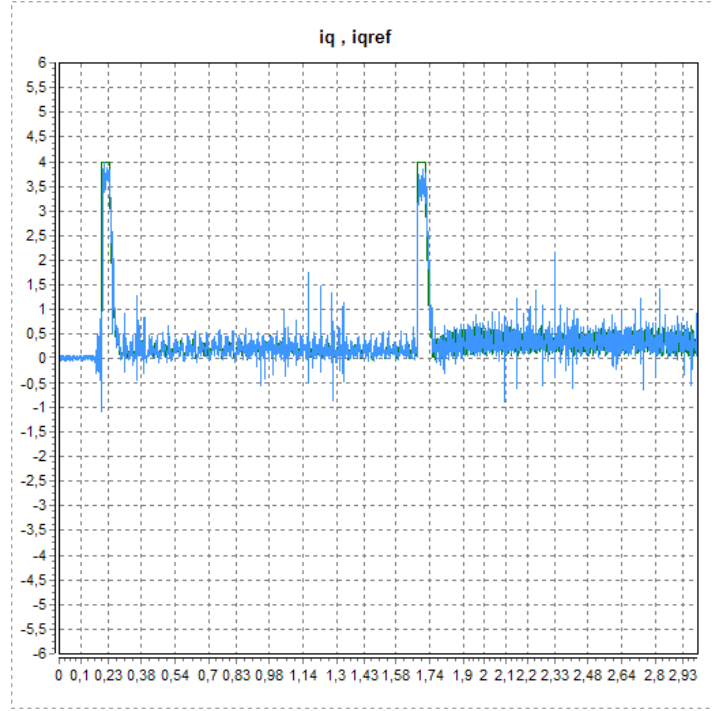


Şekil 5.14 IP kamera görüntüleri

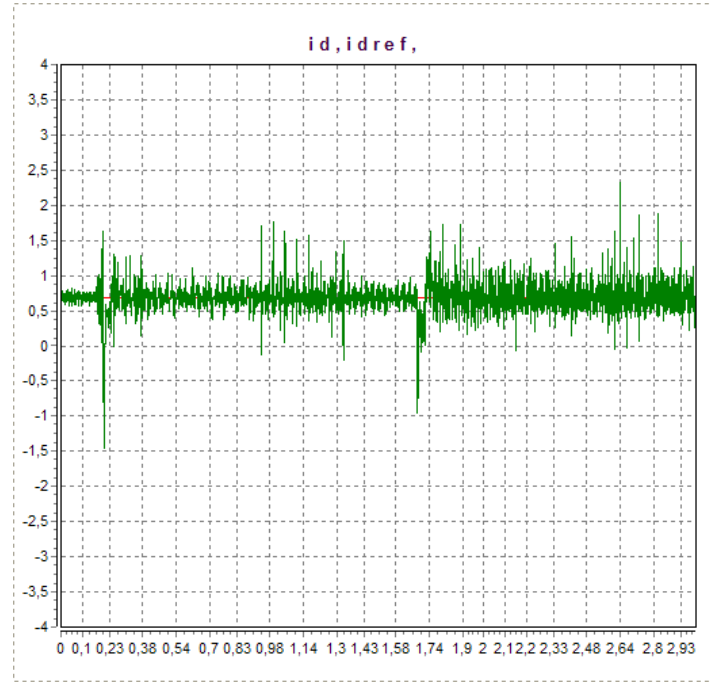
Şekil 5.15'te, motorun tasarlanmış bir PI hız denetleyici ile elde edilen kademeli sabit bir referans hızı izleme performansı, referans ve gerçek dq akımları verilmiştir.



(a)



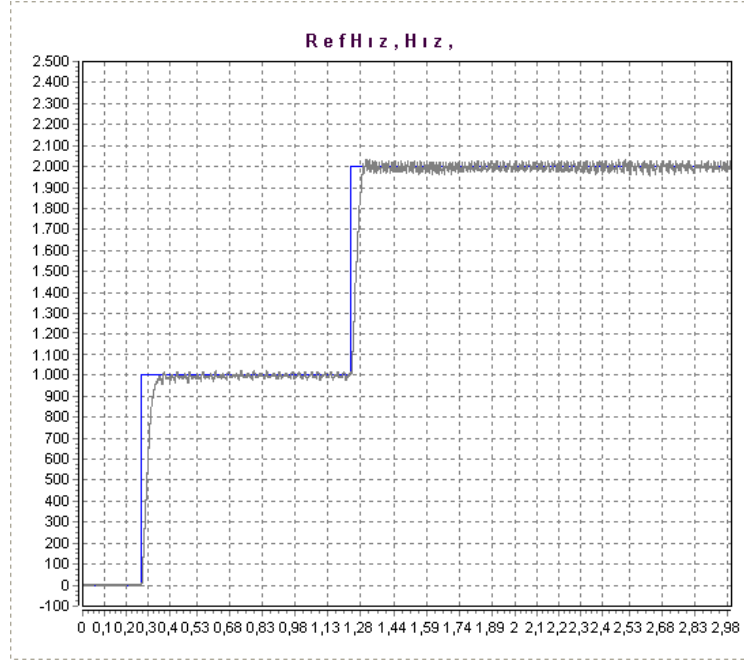
(b)



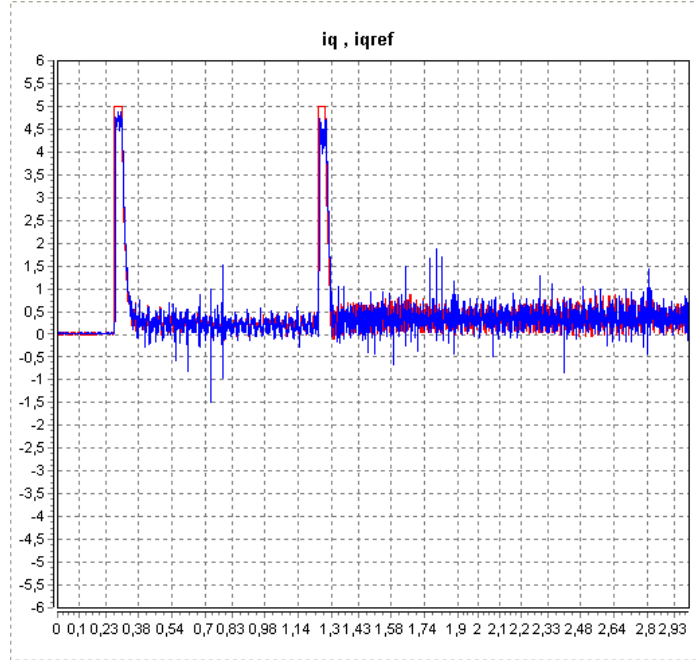
(c)

Şekil 5.15 Motor yüksüz durumda iken PI denetleyici ile 1000 d/d ve 2000 d/d'lık basamak hızı (a) PI denetleyicili referans hız ve motor hızının değişimi (b) Iq-Iqref akımlarının değişimi (c) Id-Idref akımlarının değişimi

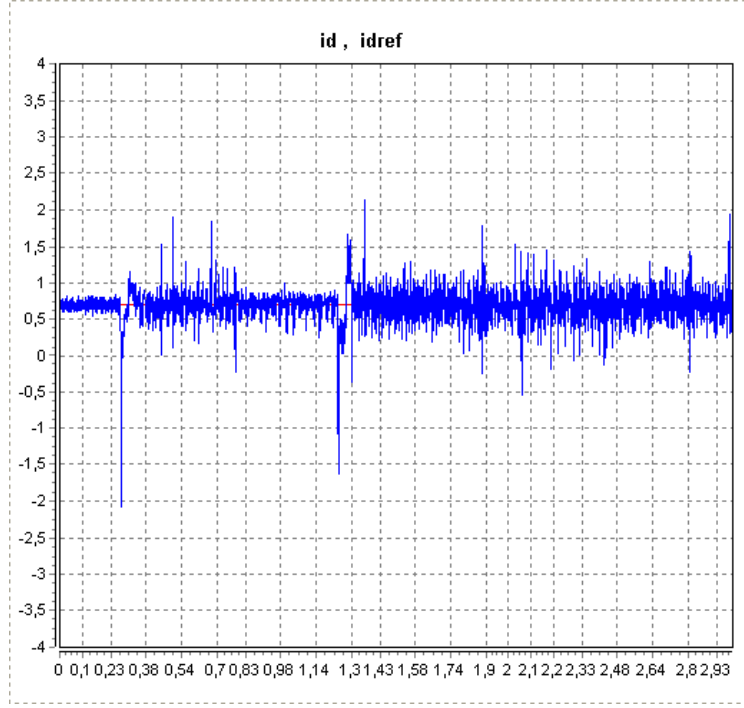
Şekil 5.16’da, motorun tasarlanmış bir SBD ile elde edilen kademeli sabit bir referans hızı izleme performansı, referans ve gerçek dq akımları verilmiştir.



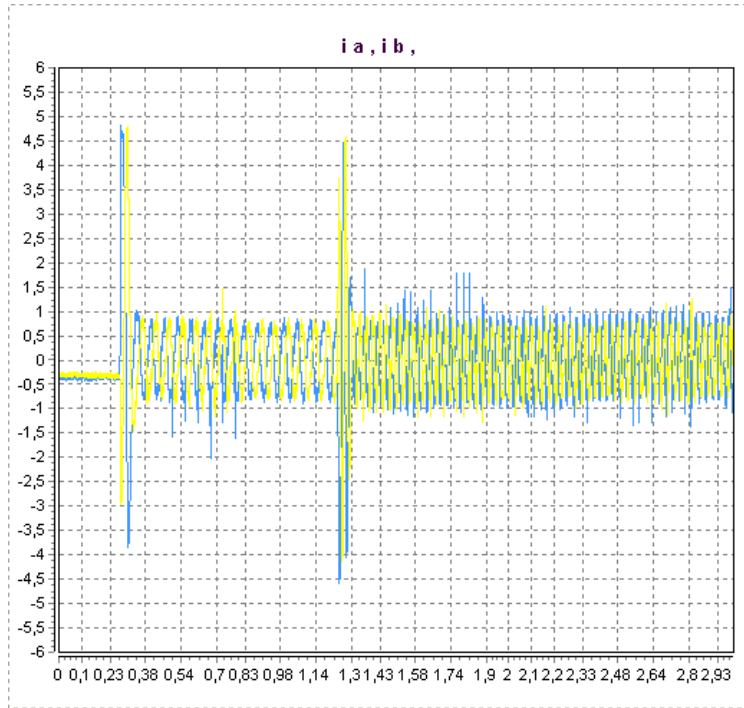
(a)



(b)



(c)



(d)

Şekil 5.16 Motor yüksüz durumda iken SBD denetleyici ile 1000 d/d ve 2000 d/d'lık basamak hızı izleme başarımı (a) PI denetleyicili referans hız ve motor hızının değişimi (b) I_q - I_{qref} akımlarının değişimi (c) I_d - I_{dref} akımlarının değişimi (d) I_a - I_b akımlarının değişimi

6 SONUÇLAR

Elektrik makinaları ve denetim sistemleri gibi birçok elektrik mühendisliği ve teknolojisi derslerinde, teorelin uygulamalarla desteklenmesi gereklidir. Bu kapsamda, eğitim kurumlarının mali kısıtlamaları ve maliyeti yüksek laboratuvar donanımları nedeniyle öğrencilerin artan eğitim amaçlı taleplerini karşılamak zorlaşmaktadır. Bu nedenle, mühendislik bilimleri eğitiminde internet kullanımı, öğrencinin uzaktan benzetim çalışmalarını yapmasını ve deneylerin gerçekleştirmesini sağlayan internet tabanlı sanal ve uzaktan erişimli laboratuvar ortamları geliştirmeye doğru yönelmiştir. Genel olarak bu tür ortamlar, kullanıcı tarafında minimum yazılıma ihtiyaç duyan etkileşimli ve esnek bir yapıda hazırlanmalıdır.

Sanal ve uzaktan erişimli laboratuvarlar genellikle LabView, MWS, Netmeeting, Activex denetimleri, Java Appletleri, Flash ve Matlab/Simulink gibi yazılımlar kullanılarak gerçekleştirilir. Ancak MWS dışındaki diğer yazılımlar kullanıldığında istemci bilgisayara çoğu zaman bazı programların yüklenmesi gerekmektedir. Ayrıca, paylaşım temelli olarak geliştirilen uzaktan erişimli laboratuvar ortamları güvenilir olmadığından istemci-sunucu mimarisi ile çalışan ortamlara ihtiyaç duyulur. Bu kapsamda, MWS yazılımı, m-dosyaları aracılığı ile PC ve giriş/çıkış portlarını kullanarak web tabanlı ve gerçek zamanlı uygulamaların yapılabilmesi için uygun bir ortamdır. Ancak MWS, uzaktan erişimli olarak RTW' yi ve dolayısıyla bazı DSP gibi donanımları desteklememektedir. Diğer taraftan, yüksek performanslı sürücü sistemlerin denetiminde, denetim algoritmalarının DSP ile gerçekleştirilmesi gereklidir. Bu nedenle, DSP ile elektrik motorlarının denetimi amacıyla uzaktan erişimli bir laboratuvar geliştirmek için ek konfigürasyonların yapılmasına, sunucu ile kullanıcı arasında iletişimi ve uzaktan DSP yazılımına otomatik erişimi sağlayan arayüzlerin geliştirilmesine ihtiyaç duyulur. Literatürde, MWS kullanılarak veri toplama kartları aracılığı ile doğru akım motorlarının denetimine yönelik uzaktan erişimli laboratuvarlar hazırlanmıştır. Ancak, erişilebilen kaynaklarda DSP ile sürülen yüksek performanslı sürücü sistemleri için uzaktan erişimli bir laboratuvar ortamı ile karşılaşılmanmıştır.

Bu tezde, DSP denetimli asenkron motorlar için sanal ve uzaktan erişimli bir laboratuvar geliştirilmiştir. İlk aşamada, asenkron motorun PI ve Sinirsel-Bulanık Ağlarla denetimi için sanal bir laboratuvar ortamı hazırlanmıştır. Etkileşimli web sayfaları aracılığı ile kullanıcı, denetim parametrelerini değiştirilerek benzetimini yapabilmekte ya da kullanıcı tarafından geliştirilen denetim algoritması sisteme yerleştirilerek sistemin benzetimi yapılabilmektedir. Benzetim çalışmalarının yapıldığı sanal laboratuvar ortamına web tabanlı erişim için MWS' dan yararlanılmış ve Matlab/Simulink dosyalarına uzaktan erişim için ek konfigürasyonlar yapılmıştır. Ayrıca, kullanıcının yaparak öğrenmesine katkı sağlamak için Delphi programlama

dili kullanılarak, kullanıcının benzetim bağlantılarını kurabilmesini sağlayan bir arayüz hazırlanmış ve bu arayüz ActivexForm'lara dönüştürülerek web sayfalarına taşınmıştır. İkinci aşamada ise uzaktan erişimli bir laboratuvar ortamı geliştirilmiştir. Uzaktan erişimli deney setinde, PWM üretimi ile akım ve hız denetim algoritmalarını gerçekleştirmek için Matlab/Simulink uyumlu DS1104 denetleyici kartı kullanılmıştır. Motorun PI ve Sinirsel-Bulanık Ağlarla denetimi için gerekli uygulama dosyaları, Matlab RTW ve dSpace RTI blokları kullanılarak hazırlanmış ve ControlDesk Developer yazılımı ile bir grafiksel kullanıcı arayüzü tasarlanmıştır. Controldesk'e uzaktan otomatik olarak erişebilmek için Python dili kullanılarak bir arayüz yazılımı hazırlanmıştır. Ayrıca, deneysel çalışmaların uzaktan gerçekleştirilebilmesi amacıyla istemci-sunucu iletişimini sağlayabilmek için Delphi programlama dili kullanılarak bir iletişim yazılımı geliştirilmiştir. Geliştirilen uzaktan erişimli laboratuvar ortamı, kullanıcıların denetim sisteminin parametrelerini değiştirerek ya da kendi denetim algoritmasını yerleştirerek deneylerini yapmasını ve deneysel sonuçlarını analiz etmesini sağlayacak şekilde esnek bir yapıda hazırlanmıştır.

Hazırlanan sanal ve uzaktan erişimli laboratuvar ortamında, asenkron motorun PI ve Sinirsel-Bulanık Ağlarla hız denetimi örnek olarak gerçekleştirilmiş ve denetim parametreleri değiştirilerek elde edilen denetim sonuçları değerlendirilmiştir. Daha sonra yapılabilecek çalışmalar için;

i-) Kullanıcının Matlab/Simulink yazılımına ihtiyaç duymaksızın kendi denetim algoritmasını sunucuya yüklemesini sağlayacak yazılım geliştirilmesi,

ii-) Uzaktan erişimle motorun yük şartlarının değiştirilerek farklı yükler altında denetim algoritmalarının test edilebilmesi için gerekli donanım ve yazılım alt yapısının oluşturulması, önerilebilir.

KAYNAKLAR

1. Tzeng, H.-W., Tien, C.-M., 2000, Design of a virtual laboratory for teaching electric machinery, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, (2), 971-976.
2. Cui, X., Zhang, X., Chen, X., 2005, A virtual laboratory for electrical and electronics teaching, International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications Proceeding, (1), 491-494.
3. Jou, M., 2005, Development of an e-learning system for teaching machining technology, Proceedings of the 2005 International Conference on Active Media Technology, AMT 2005, 347–352.
4. Cheng, K.W.E., Chan, C.L., Cheung, N.C., Sutanto, D., 2002, Virtual laboratory development for teaching power electronics, IEEE Annual Power Electronics Specialists Conference, (2), 461–466.
5. Li, S., Khan, A.A., 2005, Applying IT tools to a laboratory course for measurement, analysis, and design of electric and electronic circuits, IEEE Transactions on Education, 48 (3), 520–530.
6. de Magistris, M., 2005, A MATLAB-based virtual laboratory for teaching introductory quasi-stationary electromagnetics, IEEE Transactions on Education, 48 (1), 81–88.
7. Ko, C.C., Chen, B.M., Hu, S., Ramakrishnan, V., Cheng, C.D., Zhuang, Y., Chen, J., 2001, A web-based virtual laboratory on a frequency modulation experiment, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 31 (3), 295–303.
8. Ko, C. C., Chen, B. M., Chen, J., Zhuang, Y., Tan, K. C., 2001, Development of a web-based laboratory for control experiments on acoupled tank apparatus, IEEE-Transactions on Education, (44), 76-86.

9. Ferrero, A., Piuri, V., 1999, A simulation tool for virtual laboratory experiments in a WWW environment, *IEEE Transactions on Instrumentation and Measurement*, 48 (3), 741–746.
10. Ferrero, A., Salicone, S., 2007, Towards an hypertext of electric measurement: different approaches for an on-line, remote, didactic lab, *IEEE-Transactions on Instrumentation and Measurement*, (56), 89-94.
11. Guimaraes, E., Maffei, A., Pereira, J., Russo, B., Cardozo, E., Bergerman, M., Magalhaes, M.F., 2003, REAL: A virtual laboratory for mobile robot experiments, *IEEE Transactions on Education* 46 (1), 37–42.
12. Albu, M.M., Holbert, K.E., Heydt, G.T., Grigorescu, S.D., Truşca, V., 2004, Embedding remote experimentation in power engineering education, *IEEE Transactions on Power Systems* 19 (1), 139–143.
13. Demirbaş, Ş., 2007, İnternet tabanlı PI kontrollü bir doğru akım motoru deney seti, *Gazi Üniversitesi Müh. Mim. Fak. Der.*, 22 (2), 401-410.
14. Uran, S., Jezernik, K., 2006, Matlab Web Server and M-file application, *EPE-PEMC, Protoroz, Slovenia*, 2088-2092.
15. Uran, S., Hercog, D., Jezernik, K., 2006, Matlab Web Server and web based control design learning, *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, 5420-5425.
16. Pester A, Ismailov R., 2001, Interactive applications in teaching with the MATLAB Web Server, *Vestnik National'nogo Techniceskogo Universiteta*, 14–19.
17. Irmak, E., 2006, Doğru akım motorunun temel giriş sinyallerine tepkisinin internet üzerinden benzetimi, *Politeknik Dergisi*, 9 (2), 71-77.
18. Shiakolas, P.S., Chandra, V., Kebrle, J., 2002, Environment for engineering design, analysis, and simulation for education using MATLAB via the World Wide Web. I.

Environment description and development, *Computer Applications in Engineering Education*, 10 (3), 99-108.

19. Shiakolas, P.S., Chandra, V., Kebrle, J., Wilhite, D., 2002, Environment for engineering design, analysis, and simulation for education using MATLAB via the World Wide Web. II. Representative examples - system simulation and planar mechanism synthesis and analysis, *Computer Applications in Engineering Education* 10 (3), 109-120.
20. Yan, P., Valkama, M., Renfors, M., 2004, Distance learning in communications signal processing using MATLAB web server, Report - Helsinki University of Technology, Signal Processing Laboratory, 46, 244-247.
21. Albino Mendez, J., Lorenzo, C., Acosta, L., Torres, S., Gonzalez, E., 2006, A Web-based tool for control engineering teaching, *Computer Applications in Engineering Education* 14 (3), 178–187.
22. Gokbulut, M. and Tekin A., 2006, An educational tool for neural network control of brushless DC motors, *International Journal of Engineering Education*, 22(1), 197-204.
23. Gokbulut M., Bal, C., Dandil, B., 2006, A virtual electrical drive control laboratory: neuro-fuzzy control of induction motors, *Computer Applications in Engineering Education*, (14), 211-221.
24. Akcayol, M. A., Yigit, T., 2004, A Computer-based educational tool for pulse width modulator for static converters, *Comput Appl Eng Educ*, (12) , 215 – 223.
25. Akcayol, M. A., Elmas, C., Erdem, O. A. and Kurt, M., 2004, An educational tool for fuzzy logic controller and classical controllers, *Comput Appl Eng Educ*, (12), 126-135.
26. Akcayol, M. A., Cetin ,A., Elmas, C., 2002, An educational tool for fuzzy logic-controlled BDCM, *IEEE Trans Educ*, 45(1), 33-42.
27. Elmas, C., Akcayol, M. A., 2004, Virtual electrical machinery laboratory: A fuzzy logic controller for induction motor drive, *Int J Eng Educ*, (20), 226-233.

28. Hercog, D. Curkovic, M. Jezernik, K., 2006, DSP Based rapid control prototyping systems for engineering education and research, Computer-Aided Control Systems Design, 2006 IEEE International Symposium On, 2292–2297.
29. Cheneval, Y., Balmelli, L., Prandoni, P., Kovacevic, J. Vetterli, M., 1998, Interactive DSP education using Java, IEEE Conference on Acoustics, Speech and Signal Processing, (3), 1905–1908.
30. Changhong, W., Fei, T., Yufeng, W., Guangcheng, M., 2003, Web-Based remote control service system, Industrial Electronics, ISIE '03. IEEE International Symposium on, (1), 337-341.
31. Hurley, W.G., Lee, C. K., 2005, Development, implementation, and assessment of web-based power electronics laboratory, IEEE Transactions On Education, 48(4), 567-573.
32. Palma, L., Morrison, R.F., Enjeti, P.N., Howze, J.W., 2005, Use of web-based materials to teach electric circuit theory, IEEE Transactions On Education, 48(4), 729-734.
33. Huang, H. P., Lu, C. H., 2004, Java-Based distance learning environment for electronic instruments, IEEE Transactions On Education, 47(1), 88–94.
34. Marin, S. L. T., Garcia, F. J. B., Torres, R. M., Vazquez, S. G., and Moreno, A. J. L., 2005, Implementation of a web-based educational tool for digital signal processing teaching using the technological acceptance model, IEEE Transactions On Education, 48(4), 632-641.
35. Shiakolasan, P. S., Piyabongkarn, D., 2003, Development of a real-time digital control system with a hardware-in-the-loop magnetic levitation device for reinforcement of controls education, IEEE Transactions On Education, 46(1), 79-87.
36. Kozick R.J., Crane C.C., 1996, An integrated environment for modeling, simulation, digital signal processing, and control, IEEE Trans. On Education, 39(2), 114–119.

37. Keyhani, A., Marwali, M. N., Higuera, L. E., Athalye, G., and Baumgartner, G., 2002, An integrated virtual learning system for the development of motor drive systems, *IEEE-Transactions on Power Systems*, (17), 1-6.
38. Swamy, N., Kuljaca, O., Lewis, F. L., 2002, Internet-based educational control systems lab using NetMeeting, *IEEE-Transactions on Education*, (45), 145–151.
39. Shen, H., Xu, Z., Dalager, B., Kristiansen, V., Strom, O., Shur, M. S., Fjeldly, T. A., Lu, J. Q., Ytterdal, T., 1999, Conducting laboratory experiments over the Internet, *IEEE-Transactions on Education*, (42), 180-185.
40. Oltean, S. E., Abrudean, M., Dulau, M., 2006, Remote monitor and control application for thermal processes using TCP/IP, *Automation, Quality and Testing, Robotics*, 2006 IEEE International Conference On, (1), 209-213.
41. Watson, J. L., Bibel, G., Ebeling, K., Erjavec, J., Salehfar, H., and Zahui, M., 2004, On-line Laboratories for undergraduate distance engineering students, 34th ASEE/IEEE Frontiers in Education Conference, Savannah, GA.
42. Spanias, A., Atti, V., 2005, Interactive online undergraduate laboratories using J-DSP, *Education, IEEE Transactions On*, 48 (4), 735- 749.
43. Chirico, M., Scapolla, A. M., And Bagnasco, A., 2005, A New and open model to share laboratories on the internet, *IEEE Transactions On Instrumentation And Measurement*, 54(3), 1111-1117.
44. Familia, R., 2005, A virtual laboratory for cooperative learning of robotics and mechatronics, ITHET 6th Annual International Conference, , Juan Dolio, Dominican Republic.
45. Ciubotariu, C., Hancock, G., 2004, Work in progress - virtual laboratory with a remote control instrumentation component”, 34th ASEE/IEEE Frontiers in Education Conference, Savannah, GA.

46. Hashemian, R., Riddley, J., 2007, FPGA e-lab, a technique to remote access a laboratory to design and test, IEEE International Conference on Microelectronic Systems Education (MSE'07).
47. Persiano, G. V., Rapuano, S., Zoino, F., Morganella, A., Chiusolo, G., 2007, Distance learning in digital electronics: laboratory practice on FPGA, Instrumentation and Measurement Technology Conference - IMTC 2007 Warsaw, Poland.
48. Sanchez, P., Alvarez, B., Iborra, A., Fernandez-Merono J. M., And Pastor, J. A., 2003, Web-based activities around a digital model railroad platform, IEEE Transactions On Education, 46(2), 302-306.
49. Bellmunt, O. G., Miracle, D. M., Arellano, S. G., Sumper, A., Andreu, A. S., 2006, A distance PLC programming course employing a remote laboratory based on a flexible manufacturing cell, IEEE Transactions On Education, 49(2), 278-284.
50. Karady, G.G., Reta-Hernandez, M., Bose, A., 2000, Role of laboratory education in power engineering: Is the virtual laboratory feasible? Part II, Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference, (3), 1478–1483.
51. Meliopoulos, A.P.S., Cokkinides, G.J., 2000, Role of laboratory education in power engineering: Is the virtual laboratory feasible? Part III. Virtual power system laboratories: Is the technology ready?, Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference, (3), 1484-1489.
52. Çolak, İ., Irmak, E., Sefa, İ., Demirbaş, Ş., Bayındır, R., 2006, Temel elektrik devrelerinin analizi için internet tabanlı bir eğitim aracı, 6. Uluslararası Eğitim Teknolojileri Konferansı, Kıbrıs, 446-451.
53. Rong, G., Miaoliang, Z., Yabo, D., Dandan, S., Yonggu, W., 2005, A case study of virtual circuit laboratory for undergraduate student courses, Information Technology Based Higher Education and Training, 2005. ITHET 2005. 6th International Conference On.

54. Çolak, İ., Irmak, E., Bayındır, R., Demirbaş, Ş., Sefa, İ., 2006, Açık veya kapalı döngü denetim sistemlerinin web tabanlı benzetimi, 6. Uluslararası Eğitim Teknolojileri Konferansı, Kıbrıs, 452-461.
55. Demirbaş, Ş., Irmak, E., Sefa, İ., Bayındır, R., Çolak, İ., 2005, DA motorunun web tabanlı modellenmesi ve analizi, Bilimde Modern Yöntemler Sempozyumu, Kocaeli, 444-451.
56. Çolak İ., Demirbaş Ş., Irmak E., Sefa İ., Bayındır R., 2006, Remote control of dc motor over the internet, 3rd International Conference on Technical and Physical Problems in Power Engineering, Ankara, 216-218.
57. Valera, A., Diez, J. L., Valles, M., Albertos, P., 2005, Virtual and remote control laboratory development, IEEE Control Systems Magazine, 35-39.
58. Hercog, D., Gergi, B., Matko, V., 2005, Remote Lab for Electric Drives, IEEE ISIE 2005, Dubrovnik, Croatia, 1685-1690.
59. Uran, S., Hercog, D., Jezernik, K., 2006, Web-based matlab and controller design learning, Industrial Electronics Society Newsletter, 52(4), 14-16.
60. Demirbaş, Ş., Irmak, E., Çolak, İ., 2006, Asenkron motor benzetimi için web tabanlı bir eğitim aracı, Politeknik Dergisi, 9 (4), 147-152.
61. Diez, J.L., Valles, M., Valera, A., 2002, A Global approach for the remote process simulation and control, 15 th Triennial World Congress, Spain.
62. Ondera, M., Huba, M., 2006, Web-based tools for exact linearization control design, 14th Mediterranean Conference on Control and Automation: Ancona, Italy.
63. Sysel, M., Pomykacz, I., 2005, Delta adaptive toolbox on Matlab Web Server, CIMCA, International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-1 (CIMCA-IAWTIC'05), 134-137.

64. Garcia, R. C., Heck B. S., 1999, Enhancing classical controls education via interactive GUI design, *IEEE Control Systems Magazine*, 77-82.
65. Steidley, C. and Bachnak, R., 2005, Developing a prototype virtual laboratory for distance science and engineering education, *ITHET 6th Annual International Conference*.
66. Fan, R., Cheded, L., Toker, O., 2005, Internet-based SCADA: a new approach using Java and XML, *Computing & Control Engineering Journal*, (16), 22–26.
67. Sengur, A., Guldemir, H., 2005, An educational interface for automatic recognition of analog modulated signals, *Journal of Applied Sciences*, 5(3), 513-517.
68. Çolak, I. and Yilmaz, E. N., 1999, Simulation of the parallel connection of electrical power stations, *IJEEE International Journal of Electrical Engineering Education*, 36 (4), 332-341, 1999.
69. Rosello, E.G., Perez-Schofield, J.B.G., Dacosta, J.G. and Cota, M.P., 2003, Neuro-lab: A Highly-Reusable software-based environment to teach artificial neural networks, *Comput Appl Eng Educ*, (11), 93-102.
70. Spanias, A., Atti, V., Papandreou-Suppappola, A., Ahmed, K., Zaman, M. And Thrasyvoulou, T., 2004, On-Line signal processing using J-Dsp, *IEEE Signal Processing Letters*, 11(10), 821-825.
71. Spanias, A., Urban, S., Constantinou, A., Tainpi, M., Clausen, A., Zhang, X., Foutz, J. And Stylianou, G., 2000, Development and evaluation of a web-based signal and speech processing laboratory for distance learning, *Acoustics, Speech, And Signal Processing*, *IEEE International Conference On*, (6), 3534-3537.
72. Spanias, A. Ahmed, K. I. U., Papandreou-Suppappola, A., Zaman, M., 2003, Assessment of the Java-DSP (J-DSP) on-line laboratory software, *Frontiers in Education*,. *FIE 2003, 33rd Annual*, (1).

73. Iskander, M. F., 2002, Technology-based electromagnetic education, IEEE Transactions On Microwave Theory And Techniques, 50(3), 1015-1020.
74. Mukhopadhyay, S. C., Pinder, D. N., Gupta, G. S., and Demidenko, S., 2005, Virtual laboratory for measurements in electromagnetics, Instrumentation and Measurement Technology Conference, Ottawa, Canada, 1702-1705.
75. Borisov, A. A., Popov, N. V., Shauerman, A. A., 2006, Foundations of making virtual laboratories in engineering education, 7th International Sierin Workshop And Tutoril Edm'2006, Sessin V, Erlagol, 180-181.
76. Moros, R., Luft, F., Papp, H., 2002, Virtual laboratory course in chemical engineering and unit operations (VIPRATECH), Proceedings of the International Conference on Computers in Education (ICCE'02), IEEE.
77. Rak, R. J., Godziemba-Maliszewski, M., Majkowski, A., 2006, A proposal of virtual laboratory structure, IMTC 2006 – Instrumentation and Measurement Technology Conference Sorrento, Italy, 847-850.
78. Efstathiou, K., Karadimas, D. and Zafeiropoulos, K., 2007, A remote electrical engineering laboratory based on re-configurable hardware, Instrumentation and Measurement Technology Conference - IMTC 2007 Warsaw, Poland, 1-6.
79. Singaraju, T., Turan, A., Gokasan, M. and Bogosyan, S., 2006, Hardware-in-the-loop simulation of PUMA 560 via internet, The 32nd Annual Conference of the IEEE Industrial Electron-ics Society, IECON 2006, Conservatoire National des Arts & Metiers, Paris, France, 5426-5432.
80. Turan, A., Bogosyan, S., Gokasan, M., 2006, Development of a client-server communication method for MATLAB/Simulink based remote robotics experiments, Proceedings of the IEEE International Symposium on Industrial Electronics, Montreal, Quebec, Canada, 3201-3206.

81. Temeltas, H., Gokasan, M., Bogosyan, S. Kilic, A., 2002, Hardware in the loop simulation of robot manipulators through Internet in mechatronics education, IECON 02, IEEE 2002 28th Annual Conf., (4), 2617-2622.
82. Tanyıldızı, E., 2007, Sanal Elektrik Makinaları Laboratuvarının Oluşturulması, Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
83. Irmak, E., 2007, Uzaktan Eğitim Amaçlı İnternet Tabanlı Laboratuvar Uygulaması, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
84. Çıbuk, M., 2002, Bilgisayar Ağlarını Kullanarak Bir Adım Motorunun Uzaktan Kontrolü, Yüksek Lisans Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
85. Ko, C., Chen, B. & Chen, J., 2004, Creating web-based laboratories, Advanced Information and Knowledge Processing, Springer.
86. Çolak, İ., Irmak, E., Demirbaş, Ş., Bayındır R., 2005, Teknik eğitimde internet teknolojisinin kullanımı, 1.Uluslararası Mesleki ve Teknik Eğitim Teknolojileri Kongresi, İstanbul.
87. Akben, S. B., Subaşı, A., Kıymık, M.K., 2005, Mühendislikte E-egitim uygulamasının getireceği sonuçlar, EEB 2005 2. Ulusal Sempozyumu, Samsun, 94-98.
88. Esche, S. K., 2002, Remote experimentation - one building block in online engineering education, Proceedings of the 2002 ASEE/SEFI/TUB International Colloquium on Global Changes in Engineering Education, Berlin, Germany.
89. Bose, K. B., 2002, Modern Power Electronics and AC Drivers, Prentice Hall, New Jersey
90. Harnefors, L., 2001, Design and Analysis of General Rotor Flux Oriented Vector Control Systems, IEEE Trans. Id. Electron., 48, 383-390.
91. Verghese, G.C. and Sonders, S.R., 1998, Observer for Flux Estimation in Induction Machines, IEEE Trans. Ind. Electr. 35, 85-94.

92. Sarioğlu, M.K., Gökaşan, M. ve Boğosyan, S., 2003, Asenkron Makinalar ve Kontrolü, Birsen Yayınevi, İstanbul.
93. Sen, P.C., 1990, Electric Motor Drives and Control Past Present, and Future, IEEE Trans. on Ind. Elect., 37, 6, 477-482.
94. Lai, Y.S., 2003, Machine Modeling and Universal Controller for Vector Controlled Induction Motor Driver, IEEE Trans. on Energy Conversion, 18, 1, 23-32.
95. Can, H., 2003, Asenkron Motorların Düşük Hızlarda Duyargasız Vektör Kontrolü Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
96. Bal, C., 2007, Asenkron Motorun Kayma Kipli ve Sinirsel Bulanık Gözlemleyicilerle Algılayıcısız Hız Denetimi, Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
97. Peresada, S., Tilli, A. and Tonielli, A., 2003, Theoretical and Experimental Comparison of Indirect Field-Oriented Controllers for Induction Motors, IEEE Trans. on Power Elect., 18,1,151-163.
98. Telford, D., Dunnigen, W. and Williams, W., 2003, Online Identification of Induction Machine Electrical Parameter for Vector Control Loop Tuning, IEEE Trans. on Ind. Electronics, 50, 2, 253-261.
99. Dandıl, B., 2004, Sinirsel Bulanık Denetleyiciler Ile Asenkron Motorun Dayanıklı Hız/Konum Denetimi, Doktora Tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elazığ.
100. Arias, A., Romeral, L. and Joyne, M.G., 2000, Fuzzy Logic Direct Torque Control, Proc. IEEE Int. Symp. Ind. Elect., 253-258.
101. Haykin, S., 1994, Neural Networks, A Comprehensive Foundation, Macmillan Coll. Publ., New York.
102. Narendra, K.S. and Parthasarathy, K., 1990, Identification and Control of Dynamical Systems Using Neural Networks, IEEE Trans. on Neural Networks, 1,1, 4-27.

103. Lin, C.T. and Lee, C.S., 1996, Neural Fuzzy Systems, Prentice Hall PTR, New Jersey.
104. Takagi, T. and Sugeno, M., 1985, Identical of System and Its Application to Modeling and Control, IEEE Trans. on Man and Cybernetics, 15, 1, 116-132.
105. Lee, C.C., 1990, Fuzzy Logic in Control Systems: Fuzzy Logic Controller Part1, IEEE Trans. on Man and Cybernetics, 20, 404-418.
106. Lee, C.C., 1990, Fuzzy Logic in Control Systems: Fuzzy Logic Controller Part II, IEEE Trans. on Man and Cybernetics, 20, 419-435.
107. Jang, J.S.R., 1993, ANFIS: Adaptive-Network-Based Fuzzy Inference System IEEE, Trans. on Man and Cybernetics, 23, 3, 665-684.
108. Jang, J.S.R, Sun, C.T. and Mizutani, E., Neuro-Fuzzy and Soft Computing, Simon&Schuster, Prentice Hall Inc., Upper Saddle River.
109. Matlab Web Server User's Guide, The Mathworks Inc., 2001.
110. ControlDesk Automation Guide, 2003.
111. ControlDesk Automation Reference, 2003.
112. İnternet: [http://tr.wikipedia.org/wiki/Delphi_\(programlama_dili\)](http://tr.wikipedia.org/wiki/Delphi_(programlama_dili)), Erişim Tarihi: Şubat 2009.
113. İnternet:
<http://www.enocta.com/web2/ContentShowOne.asp?CType=2&ContentID=233&T=5>,
Erişim Tarihi: Şubat 2009.
114. Garcia, G. O., Stephan, R. M. and Watanabe, E. H., 1994, Comparing the indirect field-oriented control with a scalar method, IEEE Trans. Ind. Appl., 41(2), 201-207.
115. İnternet: <http://www.tmssoftware.com>, Erişim Tarihi: Nisan 2007.
116. İnternet: http://www.walterzorn.com/grapher/grapher_app.htm, Erişim Tarihi: Mart 2007.

ÖZGEÇMİŞ

Ahmet TEKİN

Fırat Üniversitesi Rektörlüğü

Enformatik Bölümü

23119, Elazığ

atekin@firat.edu.tr

- 1974** Elazığ'da doğdu.
- 1992–1996** Fırat Üniversitesi Teknik Eğitim Fakültesi Elektronik Bilgisayar Eğitimi Bölümünden mezun oldu.
- 1996–1999** Milli Eğitim Bakanlığı Kuyucak Çok Programlı Lisesi ve Malatya Yunus Emre Endüstri Meslek Lisesi'nde Bilgisayar Öğretmeni olarak görev yaptı.
- 2000-** Fırat Üniversitesi Rektörlüğü Enformatik Bölümünde Okutman olarak çalışmaktadır.
- 2001–2003** Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Bilgisayar Eğitimi Anabilim Dalında “YSA ile Asenkron Motorların Hız Denetimi İçin Paket Program Geliştirme” konusunda Yüksek Lisans yaptı.
- 2003-** Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalında doktora öğrenimine başladı.

EKLER

EK – 1

EK – 1 .1 Sunucu Bilgisayar

İnternet tabanlı deneysel çalışmalar ve sanal çalışma ortamları aşağıdaki teknik özellikleri verilen sunucu bilgisayarın desteği ile hazırlanmıştır:

- Intel P4 3.0 GHz işlemci
- Asus P4-P800 anakart
- 80 GB harddisk
- 4x256 MB DDR ram
- 128 MB ekran kartı

EK – 1 .2 DS1104 Denetleyici Kartı ve Asenkron Motor

Asenkron motorun hız denetimi için, Matlab/Simulink programı ile uyumlu çalışan, DS1104 denetleyici kartı donanım elemanlarında biridir. Şekil 1’de gösterilen, denetim algoritmasının gerçekleştirildiği DS1104 kartı, üzerinde PowerPC 603e/250MHz ana işlemci ve Texas Instruments TMS320F240/20MHz uydu işlemci bulundurmaktadır.

Ana işlemcin özellikleri aşağıdaki gibidir:

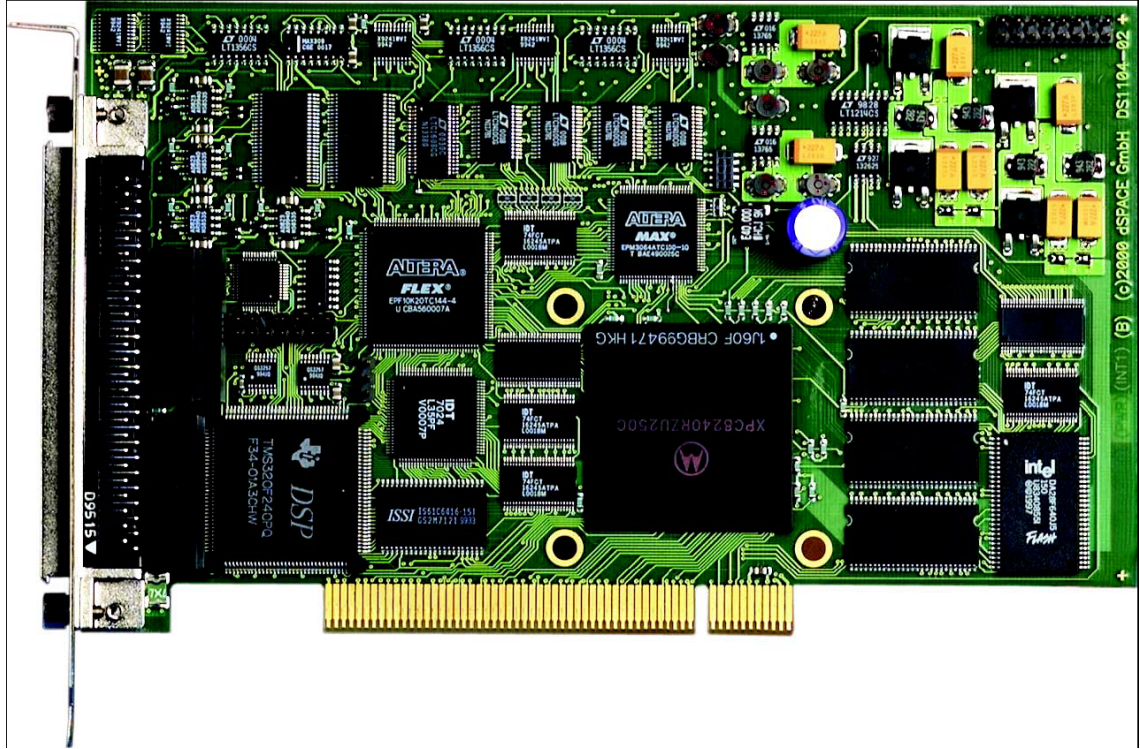
- 250 MHz işlemci hızı,
- 16 Kbyte L1 veri önbelleği,
- 16 Kbyte L1 komut önbelleği,
- Bir kesici kontrolör,
- Bir senkron DRAM kontrolör,
- Altı adet zamanlayıcı,
- Bir PCI arabirimi

DS1104 içindeki ana işlemcinin I/O, ADC ve DAC birimleri, I/O bitleri, artımlı enkoder arabirimi ve seri arabirimi mevcuttur.

DS1104’ ün bağımlı Sİİ ise TI’ ın üretmiş olduğu TMS320F240 Sİİ’ sidir. Bu Sİİ’ nin özellikleri aşağıda verilmiştir.

- Komut işleme hızı 50 ns,

- Üç adet hafıza birimi (program hafızası, bilgi hafızası ve giriş/çıkış hafızası),
- Üç adet genel amaçlı zamanlayıcı,
- Oniki adet karşılaştırma veya DGM çıkışı,
- Üç adet enkoder girişi,
- İki adet kesme birimi,
- Dört adet analog çıkış.



Şekil 1 DS1104 Denetleyici Kartı

Deneyisel çalışmada kullanılan asenkron motorun etiket ve parametre bilgileri ise aşağıdaki gibidir.

3 faz 220/380V	$L_s=0.5999$ H
$I_n=4.5/2.6$ A	$L_r=0.5999$ H
$n=2820$ d/d	$R_s=8.231$ Ω
$\cos\phi=0.85$	$T_n=3.72$ Nm
$L_{ls}=0.02129$ H	$J=0.0019$ Kg.m ²
$L_{lr}=0.02129$ H	$B=0.000263$
$L_m=0.5787$ H	$p=2$

EK – 1.3 IP Kamera

IP kamera gerekli noktaları farklı amaçlarla uzaktan izlemek ve kayıt altına almak için internet veya ağ bağlantısından faydalanarak gerek kablolu gerek kablosuz olarak kullanılabilme imkanı sunan kamera teknolojisidir. IP kamera diğer sistemler gibi ekstra aparatlar, kartlar, kayıt cihazları gerektirmeyen en ileri teknolojiye sahip olan kamera sistemidir. IP kameralar görüntü aktarılan mekanın canlı ve sesli olarak bulunulan her yerden, bilgisayar veya cep telefonu ile rahatlıkla izleyebilme olanağı sunmaktadır. VOIP, IP kamera, IP Tv gibi teknolojiler ileride görüntü ve sesin internet üzerinden çok daha rahatlıkla transfer edilebileceği şimdiden görülmektedir. Bunun en önemli sebebi önceden daha fazla para ödediğimiz ADSL, SHDSL, Ethernet gibi yüksek hızlı geniş band bağlantı teknolojilerinin hızla yaygınlaşması sayesinde ucuzlaması ve aynı süreçte de buna orantılı olarak hızlarının artmasıdır.



Şekil 2. Deneysel Çalışmalarda Kullanılan Ağ Kamerası

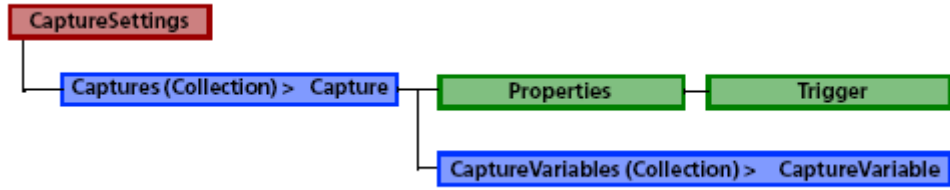
Bu tez çalışmasında gerçek zamanlı deneysel uygulamalarda ses ve görüntü geri beslemesi amacıyla Axis 205 ağ kamerası kullanılmıştır. Şekil 2’de görüntüsü verilen ağ kamerasının teknik özellikleri aşağıdaki gibi sıralanmıştır:

- 1/4” Mikron görüntü sensörü,
- 4.0 mm lens,
- 4–10000 lüks ışık hassasiyeti,
- JPEG formatında görüntü sıkıştırma,
- 640x480 piksel görüntü çözünürlüğü,
- Tüm çözünürlüklerde, saniyedeki görüntü hızı 30,
- Çok seviyeli şifre güvenliği

CONTROLDESK YAZILIMDA KULLANILAN FONKSİYONLAR İÇİN PYTHON DİLİNİN BAZI SINIF, METOD VE ÖZELLİKLERİ

CaptureSettings

ControlDesk yazılımı CaptureSettings penceresi veya CaptureSettings aracını kullanarak veri elde etmektedir. Layout üzerine yerleştirilen CaptureSettings nesnesi ile veri otomatik olarak elde edilmektedir. Elde edilen veriler mat-dosyası veya csv-dosyası olarak kaydedilebilmektedir. Veri elde etmek için kullanılan CaptureSettings aracının yapısı Şekil 1’ de verilmiştir [111].



Şekil 1 CaptureSettings sınıfının yapısı

<i>Yazım Şekli</i>	<code>OBJ.Instruments.Add(Type, NameInScript, Left, Top, Width, Height)</code> Var olan bir layout nesnesine sadece bir araç eklenebilir.
<i>Örnek Kullanım</i>	Yeni bir layout oluşturmak ve CaptureSettings_Control ismi ile CaptureSettings aracı eklemek

```

import cdautomationlib
layout =
cdautomationlib.Instrumentation().Layouts.Add()
InstrumentName = layout.Instruments.Add(
"Data Acquisition:CaptureSettings",
"CaptureSettings_Control",10,10,300,400)
Instr = layout.Instruments.Item(InstrumentName)

```

<i>Amaç</i>	CaptureSettings aracına bir örnek
<i>Özellikler</i>	Sınıfın bazı özellikleri Tablo 1’de verilmiştir.

Tablo 1 CaptureSettings sınıfının özellikleri

<i>Özellik</i>	<i>Tip</i>	<i>Amaç</i>
ActiveCapture	Capture	Eklenmiş capture nesnesini almak
AutoRedrawEnabled	Boolean	AutoRedraw durumunu almak/ayarlamak
Captures	Captures	Captures koleksiyon nesnesini almak
ClientHeight	Int	Ayarlanmış istemci yüksekliğini almak (çerçevesiz aracın görünen kısmının yüksekliği)
ClientWidth	Int	Ayarlanmış istemci genişliğini almak (çerçevesiz aracın görünen kısmının genişliği)
DocumentName	String	Layoutu içeren ismi almak
HasActiveCapture	Boolean	CaptureSetting aracının capture nesnesine eklenip eklenmeyeceğini kontrol etmek
Height	Int	Aracın yüksekliğini almak/ayarlamak
ID	String	ID özelliğini almak
IsInCapture	Boolean	CaptureSetting aracının şuanda veri yakalayıp yakalamadığını kontrol etmek
Left	Int	Layoutun sol kenarının uzaklığını almak/ayarlamak
Name	String	İsim özelliğini almak/ayarlamak
PathName	String	Bu aracın dizin yolunu almak
Top	Int	Layoutun üst kenarının uzaklığını almak/ayarlamak
Width	Int	Aracın genişliğini almak/ayarlamak

Metodlar

Sınıfın bazı metodları Tablo 2’de verilmiştir.

Tablo 2 CaptureSettings sınıfının metodları

<i>Metod</i>	<i>Amaç</i>
Activate	Aracı etkinleştirmek
ActivateCapture	İndex ile capture’yi etkinleştirmek
Deactivate	Aracın etkinliğini kaldırmak
Redraw	Aracın çalışmasını yenilemek
Refresh	Aracın çalışmasını yenilemek
SaveRefCapture	Capture’yi bir dosyaya kaydetmek
StartCapture	Capture işlemini başlatmak
StopCapture	Capture işlemini durdurmak

Tablo 2 CaptureSettings sınıfının metodları (devam)

TakeRefCapture	Capture'yi almak
UIActivate	Aracı UIActivate yapmak
UIDeactivate	Aracı UIDeactivate yapmak

ActivateCapture

<i>Sınıf</i>	CaptureSettings
<i>Yazım Şekli</i>	RetVal = OBJ.ActivateCapture(Index)
<i>Amaç</i>	Index ile capture'yi etkinleştirmek
<i>Parametreler</i>	Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
Index	Int	Capture'nin index'i

<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:
-------------------------	---

Tip	Açıklama
Boolean	True değeri capture'nin etkinleşmesinin başarılı olduğu anlamındadır

SaveRefCapture

<i>Sınıf</i>	CaptureSettings
<i>Yazım Şekli</i>	RetVal = OBJ.SaveRefCapture(CaptureFileName)
<i>Amaç</i>	Capture'yi bir dosyaya kaydetmek
<i>Parametreler</i>	Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
CaptureFileName	String	Tam dizin yolu ile birlikte capture için kaynak dosya adı

<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:
-------------------------	---

Tip	Açıklama
Boolean	True değeri capture'nin başarılı bir şekilde kaydedildiği anlamındadır

StartCapture

<i>Sınıf</i>	CaptureSettings
<i>Yazım Şekli</i>	RetVal = OBJ.StartCapture()
<i>Amaç</i>	Capture işlemini başlatmak
<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Boolean	True değeri capture işleminin başarılı bir şekilde başladığı anlamındadır

StopCapture

Sınıf

CaptureSettings

Yazım Şekli

RetVal = OBJ.StopCapture()

Amaç

Capture işlemini durdurmak

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Boolean	True değeri capture işleminin başarılı bir şekilde durdurulduğu anlamındadır

TakeRefCapture

Sınıf

CaptureSettings

Yazım Şekli

RetVal = OBJ.TakeRefCapture()

Amaç

Capture'yi almak

Açıklama

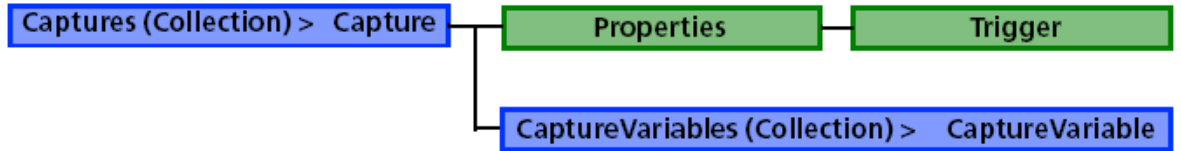
Mevcut veri yoksa başarılı olamaz

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Boolean	True değeri capture'nin başarılı bir şekilde kaydedildiği anlamındadır

Captures (Koleksiyon)



Şekil 2 Captures sınıfının yapısı

Yazım Şekli

Doğrudan bir yazım şekli yok

Örnek Kullanım

Koleksiyon içinde var olan Capture'ları saymak

```
import cdautomationlib
layout =
cdautomationlib.Instrumentation().Layouts.Add()
InstrumentName = layout.Instruments.Add(
"Data Acquisition:CaptureSettings",
"CaptureSettings_Control",10,10,300,400)
```

```
Instr = layout.Instruments.Item(InstrumentName)
Cap = Instr.Captures.Count
```

<i>Amaç</i>	Capture koleksiyonunun özelliklerini belirlemek
<i>Açıklama</i>	Yüklenmiş SDF dosyasındaki var olan Capture'ların sayısını sayısını belirlemek
<i>Özellikler</i>	Sınıfın bazı özellikleri Tablo 3'de verilmiştir.

Tablo 3 Captures sınıfının özellikleri

<i>Özellik</i>	<i>Tip</i>	<i>Amaç</i>
Count	Int	Captures koleksiyonundaki capture'lerin sayısını almak

Metodlar Sınıfın bazı metodları Tablo 4'de verilmiştir.

Tablo 4 Captures sınıfının metodları

<i>Metod</i>	<i>Amaç</i>
Item	Verilen index ile belirlenmiş capture nesnesini almak

Item

<i>Sınıf</i>	Captures(Collection)
<i>Yazım Şekli</i>	RetVal = OBJ.Item(Index)
<i>Amaç</i>	Verilen index ile belirlenmiş capture nesnesini almak
<i>Açıklama</i>	Koleksiyon nesnesine doğrudan erişilebilir: Capture = Instr.Captures[0]
<i>Parametreler</i>	Metod aşağıdaki parametreleri kullanmaktadır:

<i>Parametre</i>	<i>Tip</i>	<i>Açıklama</i>
Index	Variant	Capture nesnesinin adı veya index'i

Geri Dönen Değer Metod aşağıdaki tipteki bir değeri geri döndürür:

<i>Tip</i>	<i>Açıklama</i>
Capture	Capture nesnesinin koleksiyonu. Hata durumunda sıfır döner

Capture

Yazım Şekli

Doğrudan bir yazım şekli yok

Örnek Kullanım

Bağlantı tipini almak ve index'i 1 olan Capture nesnesini almak

```
import cdautomationlib
layout =
cdautomationlib.Instrumentation().Layouts.Add()
InstrumentName = layout.Instruments.Add(
"Data Acquisition:CaptureSettings",
"CaptureSettings_Control",10,10,300,300)
Instr = layout.Instruments.Item(InstrumentName)
Cap = Instr.Captures.Item(1)
Cappy = Cap.Connection
```

Amaç

Capture nesnesinin özelliklerini belirlemek

Özellikler

Sınıfın bazı özellikleri Tablo 5’de verilmiştir.

Tablo 5 Capture sınıfının özellikleri

Özellik	Tip	Amaç
CaptureVariables	CaptureVariables	CaptureVariables nesnesini almak
Connection	ConnectionTypeConstants	Capture’nin bağlantı tipini almak
PlatformName	String	Eklenmiş capture ortamının ismini almak
Name	String	Capture’nin ismini almak
Properties	Properties	Nesne özelliklerini almak
SimulationMode	SimulationModeConstants	Uygulamanın simülasyon modunu almak
Task	String	Capture’nin tanımlanmış görevlerini almak
VariableFileName	String	Tam olarak belirlenmiş dizin yolunu ile capture’nin var olan dosya adını almak

CaptureVariables (Collection)

CaptureVariables (Collection) > CaptureVariable

Şekil 3 CaptureVariable sınıfının yapısı

Yazım Şekli Doğrudan bir yazım şekli yok

Örnek Kullanım CaptureVariable nesnesini CaptureVariable koleksiyonuna eklemek

```
import cdautomationlib
layout =
cdautomationlib.Instrumentation().Layouts.Add()
InstrumentName = layout.Instruments.Add(
"Data Acquisition:CaptureSettings",
"CaptureSettings_Control",10,10,300,300)
Instr = layout.Instruments.Item(InstrumentName)
# The next line will only execute if a capture is
available.
Cap = Instr.Captures.Item(1)
Cappy = Cap.CaptureVariables.Add("Out_1")
```

Amaç CaptureVariable nesnesinin koleksiyon özelliklerini belirlemek

Özellikler Sınıfın bazı özellikleri Tablo 6’da verilmiştir.

Tablo 6 CaptureVariable sınıfının özellikleri

Özellik	Tip	Amaç
Count	Int	Koleksiyondaki capture değişkenlerinin sayısını almak

Metodlar Sınıfın bazı metodları Tablo 7’de verilmiştir.

Tablo 7 CaptureVariable sınıfının metodları

Metod	Amaç
Add	Bir CaptureVariable’ı mevcut listeye eklemek ve index’e göndermek
Item	İndex ile belirlenmiş CaptureVariable nesnesini almak
Remove	Mevcut listeden CaptureVariable nesnesini silmek

Add

Sınıf CaptureVariables(Collection)

*Yazım Şekli*RetVal = OBJ.Add(Name)

Amaç Bir CaptureVariable’ı mevcut listeye eklemek ve index’e göndermek

Parametreler Metod aşağıdaki parametreleri kullanmaktadır:

Geri Dönen Değer

Parametre	Tip	Açıklama
Name	String	Değişken adını eklemek

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Int	Listedeki değişken indexi

Item

Sınıf

CaptureVariables(Collection)

Yazım Şekli

RetVal = OBJ.Item(Index)

Amaç

Index veya isim ile belirlenmiş CaptureVariable nesnesini almak

Açıklama

Koleksiyonun bir nesnesine doğrudan ulaşılabilir

RetVal = CaptureVariables[Index]

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
Index	Variant	Değişken adı veya indexi

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
CaptureVariable	Capture nesnesi. Hata durumunda sıfır döner

Remove

Sınıf

CaptureVariables(Collection)

Yazım Şekli

RetVal = OBJ.Remove(Index)

Amaç

Mevcut listeden CaptureVariable nesnesini silmek

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
Index	Variant	Silinecek değişken adı veya indexi

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Boolean	Ttrue değeri capture nesnesinin listeden başarılı bir şekilde silindiğini gösterir

CaptureVariable

Yazım Şekli

Doğrudan bir yazım şekli yok

Örnek Kullanım

CaptureVariable nesnesini eklemek ve veri yakalamak için izin vermek

```
import cdatautomationlib
layout =
cdatautomationlib.Instrumentation().Layouts.Add(
InstrumentName = layout.Instruments.Add(
```

```

"Data Acquisition:CaptureSettings",
"CaptureSettings_Control",10,10,300,300)
Instr = layout.Instruments.Item(InstrumentName)
Cap = Instr.Captures.Item(1)
Cappy = Cap.CaptureVariables.Add("Out_1")
Var = Cap.CaptureVariables.Item(0)
Var_0 = Var.Enable(cdacon.true)

```

Amaç CaptureVariable nesnesinin özelliklerini belirlemek

Özellikler Sınıfın bazı özellikleri Tablo 8’de verilmiştir.

Tablo 8 CaptureVariable sınıfının özellikleri

Özellik	Tip	Amaç
Element	Object	Nesne elemanları ile ilişkili olanları almak
Flags	Int	Bayrak parçalarını depolanmış değişken bayrak sabitleri olarak almak
Name	String	Tüm değişken isimlerini almak

Metodlar Sınıfın bazı metodları Tablo 9’da verilmiştir.

Tablo 9 CaptureVariable sınıfının metodları

Metod	Amaç
Enable	İşaret değişkenlerinin yakalanmasına izin vermek/vermemek

Enable

Sınıf CaptureVariable

*Yazım Şekli*RetVal = OBJ.Enable(Enable)

Amaç İşaret değişkenlerinin yakalanmasına izin vermek/vermemek

Parametreler Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
Enable	Boolean	True değeri işaret değişkenlerinin yakalanmasına izin verildiği anlamındadır

Geri Dönen Değer Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Boolean	True değeri işaret değişkenlerinin başarılı bir şekilde yakalandığı anlamındadır

Properties



Şekil 4 Properties sınıfının yapısı

Yazım Şekli	Doğrudan bir yazım şekli yok
Örnek Kullanım	Bir Capture için Timestamping' e izin vermek

```

import cdautomationlib
layout =
cdautomationlib.Instrumentation().Layouts.Add()
InstrumentName = layout.Instruments.Add(
"Data Acquisition:CaptureSettings",
"CaptureSettings_Control",10,10,300,300)
Instr = layout.Instruments.Item(InstrumentName)
Cap = Instr.Captures.Item(1)
Prop = Cap.Properties.TimestampingEnabled
  
```

Amaç	Capture'ın Properties nesnelerinin özelliklerini belirlemek
Özellikler	Sınıfın bazı özellikleri Tablo 10'da verilmiştir.

Tablo 10 Properties sınıfının özellikleri

Özellik	Tip	Amaç
AcquisitionFilename	String	Elde edilen kaynak dosyayı almak
AcquisitionMode	AcquisitionModeConstants	Elde edilen modu almak
AutoRepeat	Boolean	Veri yakalamayı otomatik olarak yeniden başlatmaya ayarlamak/almak
AutoStart	Boolean	Animasyonu otomatik olarak başlatmak
DownsamplingFactor	Int	Downsampling faktörünü ayarlamak/almak
IntervalLen	Double	Süreyi almak/ayarlamak
ShowGraphicOutput	Boolean	Ekran görüntüsünde grafiksel çıktı almak

Tablo 10 Properties sınıfının özellikleri (devam)

TimestampingEnabled	Boolean	Timestamping'i almak/ayarlamak
TimestampingPossible	Boolean	Timestamping'in mümkün olup olmayacağını kontrol eder
Trigger	Trigger	Trigger nesnesini alır

Metodlar

Sınıfın bazı metodları Tablo 11'de verilmiştir.

Tablo 11 Properties sınıfının metodları

Metod	Amaç
SetAcquisitionMode	Gerekli ise kaynak dosyayı ve elde edilen modu ayarlamak

SetAcquisitionMode*Sınıf*

Properties

Yazım Şekli

RetVal = OBJ.SetAcquisitionMode(AcquisitionMode, Filename)

Amaç

Gerekli ise kaynak dosyayı ve elde edilen modu ayarlamak

Açıklama

Dosya tipi IDF, MAT veya CSV olabilir

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
AcquisitionMode	AcquisitionModeConstants	Elde edilen modun tipi
FileName	String	Elde edilen veri için kaynak dosya adı

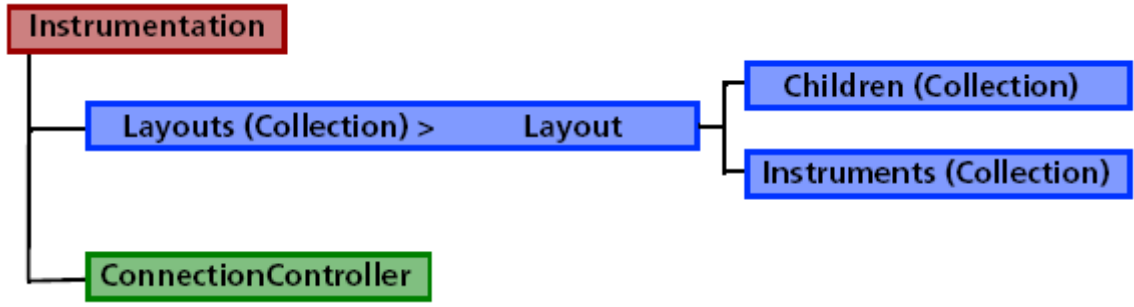
Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
ErrorCode	S_OK, işlemin başarılı olduğunu göstermektedir

Instrumentation

Instrumentation sınıfı layout araçları üzerinde işlemler yapmak için kullanılır. Bir layout oluşturulabilir, silinebilir, açılabilir ve hatta açık olan layoutların listesini görülebilir. Instrumentation sınıfını kullanarak Layout ve ConnectionController nesnelere erişilebilir. [111].



Şekil 5 Instrumentation sınıfının yapısı

Yazım Şekli OBJ = cdautomationlib.Instrumentation()

Örnek Kullanım Bir Instrumentation nesnesini animasyon moduna ayarlamak

```

import cdautomationlib
import cdacon
Instr = cdautomationlib.Instrumentation()
Instr.AnimationMode = cdacon.amcModeAnimation
  
```

Amaç Instrumentation nesnesinin özelliklerini belirlemek ve nesneyi kullanmak

Açıklama Tüm Instrumentation özelliklerine ve metodlarına ulaşmak için bu sınıf kullanılmaktadır

Geri Dönen Metod aşağıdaki tipteki bir değeri geri döndürür:

<i>Geri Dönen Değer</i>	<i>Tip</i>	<i>Açıklama</i>
OBJ	Instrumentation	Oluşturulmuş instrumentation nesnesi

Özellikler Sınıfın bazı özellikleri Tablo 12’de verilmiştir.

Tablo 12 Instrumentation sınıfının özellikleri

<i>Özellik</i>	<i>Tip</i>	<i>Amaç</i>
ActiveLayout	Layout	Son aktif layout nesnesini almak
AnimationMode	AnimationModeConstants	Animasyon moduna ayarlamak/almak
ConnectionController	ConnectionController	ConnectionController nesnesini almak
Layouts	Layouts	Açık halde bulunan layout nesnelerini almak

Metodlar

Sınıfın bazı metodları Tablo 13’de verilmiştir.

Tablo 13 Instrumentation sınıfının metodları

Metod	Amaç
GetLayout	İsmi verilen layoutu almak
GetLayoutList	Tüm açık layoutların dizin yolu listesini almak
NewLayout	Çalışma bölgesinde yeni bir layout oluşturmak
OpenLayout	Layout dosyasını açmak ve dosya adını geçerli kılmak
RemoveLayout	İsmi verilen layoutu kapatmak
RemoveLayouts	ControlDesk içindeki tüm layoutları kapatmak

GetLayout

Sınıf

Instrumentation

Yazım Şekli

Layout = OBJ.GetLayout(LayoutName)

Amaç

İsmi verilen layoutu almak

Açıklama

Metodu aşağıdaki gibi kullanmak daha uygun olacaktır

Instrumentation().Layouts.Item(LayoutName)

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
LayoutName	String	Layoutun tüm dizin yolu geri dönmeli

İstisna

Metod aşağıdaki istisnai duruma sebep olabilir

Tip	Açıklama
TypeError	Parametre doğru bir tipe sahip değilse

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Layout	İstenen Layout

GetLayoutList

Sınıf

Instrumentation

Yazım Şekli

LayoutList = OBJ.GetLayoutList()

Amaç

Tüm açık layoutların dizin yolu listesi ile dosya isimlerinin listesini almak

Açıklama

Metodu aşağıdaki gibi kullanmak daha uygun olacaktır

```
from cdautomationlib import Instrumentation
LayoutList = []
```

```
for Layout in Instrumentation().Layouts:  
LayoutList.append(Layout.PathName)
```

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
(string, string, ...)	Tüm Layout isim listesi

NewLayout

Sınıf

Instrumentation

Yazım Şekli

```
Layout = OBJ.NewLayout()
```

Amaç

Çalışma bölgesinde yeni bir layout oluşturmak

Açıklama

Metodu aşağıdaki gibi kullanmak daha uygun olacaktır

```
Instrumentation().Layouts.Add()
```

İstisna

Metod aşağıdaki istisnai duruma sebep olabilir

Tip	Açıklama
cdliberror	Layout oluşturulamamışsa

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Layout	Oluşturulmuş Layout

OpenLayout

Sınıf

Instrumentation

Yazım Şekli

```
Layout = OBJ.OpenLayout(LayoutName)
```

Amaç

Layout dosyasını açmak ve dosya adını geçerli kılmak

Açıklama

Metodu aşağıdaki gibi kullanmak daha uygun olacaktır

```
Instrumentation().Layouts.Add(LayoutName)
```

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
LayoutName	String	Layoutun tüm dizin yolu ile dosya adı açılmalı

İstisna

Metod aşağıdaki istisnai duruma sebep olabilir

Tip	Açıklama
TypeError	Parametre doğru tipe sahip değilse
TypeError	Parametre dosyada yoksa
cdliberror	Layout oluşturulamamışsa

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Layout	Layout sınıflarının durumu

RemoveLayout

Sınıf

Instrumentation

Yazım Şekli

`OBJ.RemoveLayout(LayoutName)`

Amaç

İsmi verilen layoutu kapatmak

Açıklama

Metodu aşağıdaki gibi kullanmak daha uygun olacaktır

`Instrumentation().Layouts.Remove(LayoutName)`

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
LayoutName	String	Layout başlığı veya tüm dizin yolu ile dosya adı kaldırılmalı

İstisna

Metod aşağıdaki istisnai duruma sebep olabilir

Tip	Açıklama
TypeError	Layout ismi string değilse
cdliberror	Layout bulunamadıysa

RemoveLayouts

Sınıf

Instrumentation

Yazım Şekli

`OBJ.RemoveLayouts()`

Amaç

ControlDesk içindeki tüm layoutları kapatmak

Açıklama

Metodu aşağıdaki gibi kullanmak daha uygun olacaktır

`Instrumentation().Layouts.RemoveAll()`

Instruments (Collection)

Yazım Şekli

Doğrudan bir yazım şekli yok

Örnek Kullanım

İki Bar aracını gruplandırmak

```
import cdautomationlib
lay =
cdautomationlib.Instrumentation().Layouts.Add()
InstrumentName_1 = lay.Instruments.Add(
"Virtual Instruments:Bar", "Bar_1", 10, 10, 200, 100)
InstrumentName_2 = lay.Instruments.Add(
"Virtual Instruments:Bar", "Bar_2", 220, 10, 200, 100)
lay.Instruments.Select(InstrumentName_1)
lay.Instruments.AddToSelection(InstrumentName_2)
lay.Instruments.Group()
```

Açıklama Bu koleksiyon bir layouttaki tüm araçları içermektedir.
Özellikler Sınıfın bazı özellikleri Tablo 14’de verilmiştir.

Tablo 14 Instruments sınıfının özellikleri

<i>Özellik</i>	<i>Tip</i>	<i>Amaç</i>
Count	Int	Bir layouttaki araçların sayısını almak

Metodlar Sınıfın bazı metodları Tablo 15’de verilmiştir.

Tablo 15 Instruments sınıfının özellikleri

<i>Metod</i>	<i>Amaç</i>
Add	Layouta yeni bir araç eklemek. Bir aracın dizin yolu ismi destekliyorsa, var olan bir aracın bir parçası olarak veya yeni araç grubu olarak oluşturulacaktır.
AddToSelection	Seçilen layouta dizin yolu ismi ile birlikte verilen aracı eklemek. Dizin yolu ismi layout ismini içermemelidir.
ClearSelection	Mevcut seçilmiş araçları silmek.
DeleteSelection	Tüm seçilmiş araçları silmek.
Find	Dizin yolu ismi ile birlikte verilen aracı bulmak. Dizine yolu ismi layout ismini içermemelidir.
Group	Seçilmiş tüm araçları bir gruba dönüştürmek.
Item	Bir araca indeksi ile erişmek. İndeks olarak ya Find metodunun değeri yada aracın dizin yolu ismi geri dönmektedir.
Remove	Dizin yolu ismi ile birlikte verilen aracı silmek. Dizine yolu ismi layout ismini içermemelidir.
RemoveAll	Layouttan tüm araçları silmek.
RemoveFromSelection	Seçilen layouttan dizin yolu ismi ile birlikte verilen aracı silmek. Dizine yolu ismi layout ismini içermemelidir.
Select	Dizin yolu ismi ile birlikte verilen aracı seçmek. Dizine yolu ismi layout ismini içermemelidir.
SelectAll	Layouttaki tüm araçları seçmek.
Ungroup	Seçilen grubu bireysel araçlara dönüştürmek.

Add

Sınıf

Instruments (Collection)

Yazım Şekli

RetVal = OBJ.Add(Type, NameinScript, Left, Top, Width, Height)

Amaç

Layouta yeni bir araç eklemek. Bir aracın dizin yolu ismi destekliyorsa, var olan bir aracın bir parçası olarak veya yeni araç grubu olarak oluşturulacaktır.

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
Type	String	Tanımlanan araç ismi. Tanımlanan isim, araç kütüphane ismi ve araç tipini içermektedir.
NameinScript	String	Araç ismi.
Left	Int	Layout üzerindeki aracın sol konumu
Top	Int	Layout üzerindeki aracın üst konumu
Width	Int	Layout üzerindeki aracın genişliği
Height	Int	Layout üzerindeki aracın yüksekliği

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
String	Aracın görünen ismi

AddToSelection

Sınıf

Instruments (Collection)

Yazım Şekli

RetVal = OBJ.AddToSelection(Name)

Amaç

Seçilen layouta dizin yolu ismi ile birlikte verilen aracı eklemek. Dizin yolu ismi layout ismini içermemelidir.

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
Name	String	Aracın dizin yolu ismi

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Boolean	True değeri aracın eklendiği anlamındadır

ClearSelection

Sınıf

Instruments (Collection)

Yazım Şekli

OBJ.ClearSelection()

Amaç

Mevcut seçilmiş araçları silmek.

DeleteSelection

<i>Sınıf</i>	Instruments (Collection)
<i>Yazım Şekli</i>	OBJ.DeleteSelection()
<i>Amaç</i>	Tüm seçilmiş araçları silmek.

Find

<i>Sınıf</i>	Instruments (Collection)
<i>Yazım Şekli</i>	RetVal = OBJ.Find(Value)
<i>Amaç</i>	Dizin yolu ismi ile birlikte verilen aracı bulmak. Dizin yolu ismi layout ismini içermemelidir.
<i>Parametreler</i>	Metod aşağıdaki parametreleri kullanmaktadır:

<i>Parametre</i>	<i>Tip</i>	<i>Açıklama</i>
Value	String	Aracın dizin yolu

<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:
-------------------------	---

<i>Tip</i>	<i>Açıklama</i>
Int	Aracın indeksi. 0 değeri dönmüşse araç bulunamadı demektir.

Group

<i>Sınıf</i>	Instruments (Collection)
<i>Yazım Şekli</i>	RetVal = OBJ.Group()
<i>Amaç</i>	Seçilmiş tüm araçları bir gruba dönüştürmek.
<i>Parametreler</i>	-
<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:

<i>Tip</i>	<i>Açıklama</i>
Object	Oluşturulmuş araç grubu

Item

<i>Sınıf</i>	Instruments (Collection)
<i>Yazım Şekli</i>	RetVal = OBJ.Item(Index)
<i>Amaç</i>	Bir araca indeksi ile erişmek. İndeks olarak ya Find metodunun değeri yada aracın dizin yolu ismi geri dönmektedir.
<i>Açıklama</i>	Koleksiyon nesnesine aşağıdaki gibi direkt olarak ulaşılabilir: RetVal = Instruments[Index]
<i>Parametreler</i>	Metod aşağıdaki parametreleri kullanmaktadır:

<i>Parametre</i>	<i>Tip</i>	<i>Açıklama</i>
Index	Variant	Aracın indeksi

<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:
-------------------------	---

Tip	Açıklama
Object	İstenen araç

Remove

Sınıf

Yazım Şekli

Amaç

Parametreler

Geri Dönen Değer

Instruments (Collection)

RetVal = OBJ.Remove(Value)

Dizin yolu ismi ile birlikte verilen aracı silmek. Dizin yolu ismi layout ismini içermemelidir.

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
Value	String	Aracın dizin yolu ismi

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Int	Koleksiyondaki aracın yeni numarası

RemoveAll

<i>Sınıf</i>	Instruments (Collection)
<i>Yazım Şekli</i>	<code>OBJ.RemoveAll()</code>
<i>Amaç</i>	Layouttan tüm araçları silmek.

RemoveFromSelection

<i>Sınıf</i>	Instruments (Collection)						
<i>Yazım Şekli</i>	<code>RetVal = OBJ.RemoveFromSelection(Name)</code>						
<i>Amaç</i>	Seçilen layouttan dizin yolu ismi ile birlikte verilen aracı silmek. Dizin yolu ismi layout ismini içermemelidir.						
<i>Parametreler</i>	Metod aşağıdaki parametreleri kullanmaktadır: <table><tr><th><i>Parametre</i></th><th><i>Tip</i></th><th><i>Açıklama</i></th></tr><tr><td>Name</td><td>String</td><td>Aracın dizin yolu ismi</td></tr></table>	<i>Parametre</i>	<i>Tip</i>	<i>Açıklama</i>	Name	String	Aracın dizin yolu ismi
<i>Parametre</i>	<i>Tip</i>	<i>Açıklama</i>					
Name	String	Aracın dizin yolu ismi					
<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:						

Tip	Açıklama
Boolean	True değeri aracın silindiği anlamındadır

Select

<i>Sınıf</i>	Instruments (Collection)
<i>Yazım Şekli</i>	<code>RetVal = OBJ.Select(Name)</code>
<i>Amaç</i>	Dizin yolu ismi ile birlikte verilen aracı seçmek. Dizin yolu ismi layout ismini içermemelidir.
<i>Parametreler</i>	Metod aşağıdaki parametreleri kullanmaktadır:

Geri Dönen Değer

Parametre	Tip	Açıklama
Name	String	Aracın dizin yolu ismi

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Boolean	True değeri aracın seçildiği anlamındadır

SelectAll

Sınıf

Instruments (Collection)

Yazım Şekli

OBJ.SelectAll()

Amaç

Layouttaki tüm araçları seçmek.

Ungroup

Sınıf

Instruments (Collection)

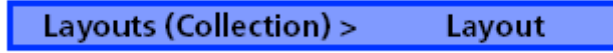
Yazım Şekli

OBJ.Ungroup()

Amaç

Seçilen grubu bireysel araçlara dönüştürmek.

Layouts (Collection)



Şekil 6 Layouts sınıfının yapısı

Yazım Şekli

Doğrudan bir yazım şekli yok

Örnek Kullanım

Açık layoutları saymak

```
import cdautomationlib
Inst = cdautomationlib.Instrumentation()
Inst.Layouts.Count
```

Amaç

Koleksiyondaki layoutlara bir layout eklemek, silmek ve yazmak

Açıklama

Sunulan sınıf layout nesnelerinin bir sınıfıdır.

Özellikler

Sınıfın bazı özellikleri Tablo 16’da verilmiştir.

Tablo 16 Layouts sınıfının özellikleri

Özellik	Tip	Amaç
ActiveLayout	Layout	Aktif layoutu almak
Count	Int	Açık layoutların sayısını almak

Metodlar

Sınıfın bazı metodları Tablo 17’de verilmiştir.

Tablo 17 Layouts sınıfının özellikleri

Metod	Amaç
Add	Dosya ismi ile birlikte verilen layoutu açmak. Dosya adı boşsa yeni bir layout oluşturmak.
Item	Dosya ismi ile birlikte verilen layouta erişmek.
Open	Verilen dosya ismi ile layoutu açmak.
Remove	Başlık veya dosya ismi verilen layoutu kapatmak.
RemoveAll	Bütün layoutları kapatmak.
Save	Bütün layoutları kaydetmek.

Add**Sınıf**

Layouts (Collection)

Yazım Şekli

```
RetVal = OBJ.Add([Filename=""])
```

Amaç

Dosya ismi ile birlikte verilen layoutu açmak. Dosya adı boşsa yeni bir layout oluşturmak.

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
FileName	String	Layoutun dosya adı

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
Layout	Açık layout

Item**Sınıf**

Layouts (Collection)

Yazım Şekli

```
RetVal = OBJ.Item(FileName)
```

Amaç

Dosya ismi ile birlikte verilen layouta erişmek.

Parametreler

Metod aşağıdaki parametreleri kullanmaktadır:

Parametre	Tip	Açıklama
FileName	String	Layoutun dosya adı

Geri Dönen Değer

Metod aşağıdaki tipteki bir değeri geri döndürür:

<i>Tip</i>	<i>Açıklama</i>
Layout	İstenen layout

Open

Sınıf Layouts (Collection)
*Yazım Şekli*RetVal = OBJ.Open(Filename)
Amaç Verilen dosya ismi ile layoutu açmak.
Parametreler Metod aşağıdaki parametreleri kullanmaktadır:

<i>Parametre</i>	<i>Tip</i>	<i>Açıklama</i>
FileName	String	Layoutun dosya adı

Geri Dönen Değer Metod aşağıdaki tipteki bir değeri geri döndürür:

<i>Tip</i>	<i>Açıklama</i>
Layout	Açılan layout

Remove

Sınıf Layouts (Collection)
*Yazım Şekli*RetVal = OBJ.Remove(Filename)
Amaç Başlık veya dosya ismi verilen layoutu kapatmak.
Parametreler Metod aşağıdaki parametreleri kullanmaktadır:

<i>Parametre</i>	<i>Tip</i>	<i>Açıklama</i>
FileName	String	Layoutun dosya adı veya başlığı

Geri Dönen Değer Metod aşağıdaki tipteki bir değeri geri döndürür:

<i>Tip</i>	<i>Açıklama</i>
Int	Açık layout sayısı

RemoveAll

Sınıf Layouts (Collection)
*Yazım Şekli*OBJ.RemoveAll()
Amaç Bütün layoutları kapatmak.

Save

Sınıf Layouts (Collection)
*Yazım Şekli*RetVal = OBJ.Save()
Amaç Bütün layoutları kaydetmek.
Geri Dönen Değer Metod aşağıdaki tipteki bir değeri geri döndürür:

<i>Tip</i>	<i>Açıklama</i>
Boolean	True değeri tüm layuotların başarılı bir şekilde kaydedildiğini göstermektedir.

HardwareManager

HardwareManager sınıfı donanımı gerçek zamanlı olarak çalıştırmaya izin vermektedir. Gerçek zamanlı uygulamalar karta yüklenebilir veya kurulu olan kartın listesi alınabilir [111].

<i>Modül</i>	cdautomationlib				
<i>Yazım Şekli</i>	OBJ = cdautoamtionlib.HardwareManager()				
<i>Amaç</i>	HardwareManager nesnesini oluşturmak.				
<i>Açıklama</i>	Hardware sınıfı donanım özelliklerine erişmek için ControlDesk yazılımı tarafından kullanılmaktadır.				
<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:				
	<table><tr><th>Tip</th><th>Açıklama</th></tr><tr><td>HardwareManager</td><td>Oluşturulan HardwareManager nesnesi</td></tr></table>	Tip	Açıklama	HardwareManager	Oluşturulan HardwareManager nesnesi
Tip	Açıklama				
HardwareManager	Oluşturulan HardwareManager nesnesi				
<i>Metodlar</i>	Sınıfın bazı metodları Tablo 18’de verilmiştir.				

Tablo 18 HardwareManager sınıfının metodları

Metod	Amaç
BatchMode	HardwareManager’ı batch moda ayarlamak
GetBoardNames	Kartın isim listesine erişmek.
GuiMode	HardwareManager’ı etkileşim moduna ayarlamak.
LoadApplication	Tek bir DSP’ye bir uygulama yüklemek.
LoadMPApplication	Çok işlemcili sisteme bir uygulama yüklemek.
StartApplication	RTP üzerine yüklenmiş uygulamayı çalıştırmak.
StopApplication	RTP üzerine yüklenmiş uygulamayı durdurmak.

BatchMode

<i>Sınıf</i>	HardwareManager
<i>Yazım Şekli</i>	OBJ.BatchMode()
<i>Amaç</i>	HardwareManager’ı batch moda ayarlamak.
<i>Açıklama</i>	Batch mod ayarlanmamışsa, HardwareManager herhangi bir istenen etkileşimi gerçekleştirmez.

GetBoardNames

<i>Sınıf</i>	HardwareManager
<i>Yazım Şekli</i>	BoardList = OBJ.GetBoardNames()
<i>Amaç</i>	Kartın isim listesine erişmek.
<i>Geri Dönen Değer</i>	Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
BoardList (string, string)	Kaydedilmiş kart isimleri

GuiMode

Sınıf	HardwareManager
Yazım Şekli	OBJ.GuiMode ()
Amaç	HardwareManager'ı etkileşim moduna ayarlamak.
Açıklama	Gui moda ayarlanmamışsa, HardwareManager herhangi bir istenen etkileşimi gerçekleştirmez.
Geri Dönen Değer	Metod aşağıdaki tipteki bir değeri geri döndürür:

Tip	Açıklama
BoardList (string, string)	Kaydedilmiş kart isimleri

LoadApplication

Sınıf	HardwareManager
Yazım Şekli	OBJ.LoadApplication(Board, AppPath [, Reserved = 0])

Amaç Tek bir DSP'ye bir uygulama yüklemek.

Parametreler

Parametre	Tip	Açıklama
Board	String	Kaydedilmiş kart ismi
AppPath	String	Nesne dosyaları için dizin yolu
Reserved	Int	Sonraki kullanım için parametre

İstisna Metod aşağıdaki istisnai duruma sebep olabilir

Tip	Açıklama
TypeError	Yanlış tipe sahip parametre
cdaliberror	Uygulama yüklenmedi

LoadMPApplication

Sınıf	HardwareManager
Yazım Şekli	OBJ.LoadMPApplications(MPSystem, PathName [, Reserved = 0])

Amaç Çok işlemcili sisteme bir uygulama yüklemek.

Parametreler

Parametre	Tip	Açıklama
MPSystem	String	MP sistemin adı
PathName	String	SDF veya APL dosyalarının dizin yolu
Reserved	Int	Sonraki kullanım için parametre

İstisna Metod aşağıdaki istisnai duruma sebep olabilir

Tip	Açıklama
TypeError	Yanlış tipe sahip parametre
cdaliberror	Uygulama yüklenmedi

StartApplication

Sınıf

HardwareManager

Yazım Şekli

OBJ.StartApplication(Board)

Amaç

RTP üzerine yüklenmiş uygulamayı çalıştırmak.

Parametreler

Parametre	Tip	Açıklama
Board	String	Kaydedilmiş kart ismi

İstisna

Metod aşağıdaki istisnai duruma sebep olabilir

Tip	Açıklama
TypeError	Yanlış tipe sahip parametre
cdaliberror	Uygulama yüklenmedi

StopApplication

Sınıf

HardwareManager

Yazım Şekli

OBJ.StopApplication(Board)

Amaç

RTP üzerine yüklenmiş uygulamayı durdurmak.

Parametreler

Parametre	Tip	Açıklama
Board	String	Kaydedilmiş kart ismi

İstisna

Metod aşağıdaki istisnai duruma sebep olabilir

Tip	Açıklama
TypeError	Yanlış tipe sahip parametre
cdaliberror	Uygulama yüklenmedi