

T.C

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

ANİ YÜK DEĞİŞİKLİKLERİ İÇİN KARARLI, ZEKİ
KONTROLÖR TASARIMI

Ertuğrul AKBAŞ
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

2005

T.C

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

ANİ YÜK DEĞİŞİKLİKLERİ İÇİN KARARLI, ZEKİ
KONTROLÖR TASARIMI

Ertuğrul AKBAŞ

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

Doç. Dr. E. Murat ESİN

2005

Ertuğrul AKBAŞ ‘un tez çalışması , G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu’nun .../.../2004 tarih ve 2004/... sayılı kararıyla oluşturulan jüri tarafından Bilgisayar Mühendisliği Anabilim Dalında doktora tezi olarak kabul edilmiştir.

JÜRİ

ÜYE : Doç. Dr. E. Murat ESİN
(Tez Danışmanı)

ÜYE :

ÜYE :

ÜYE :

ÜYE :

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu’nun tarih ve/..... kararı.

İmza/Mühür

ÖZET

Bu çalışmada endüstriyel ortamlarda ani yük değişikliğine dayalı bir kontrol algoritması tasarlanmıştır. Bu tasarıma ulaşmak için literatürde daha önce yapılmış çalışmalar analiz edilmiş ve istenilen çalışma şartlarında performans kriterleri ve bunlara göre yapılan çalışmaların avantajlı tarafları tespit edilmiştir.

Bu çalışmada analiz edilen yöntemler:

- 1-Ters Denetim Modeli (Direct Inverse Modelling Architecture) (DIMA)
- 2-Ters Denetim Modeli (Direct Inverse Modelling Architecture) (DIMA) Tip II
- 3-Geri Beslemeli –Hata Öğrenmeli Denetim (Feedback Error Learning Architecture) (FELA)
- 4- Kayan Kipli Kontrol (Neuro- Sliding Mode Architecture) (NESMA)

Bu yöntemlerin analizi ve performans karşılaştırması için öncelikle bu parametreler tespit edilip daha sonra bu parametreler üzerinde bir karşılaştırma sonucuna gidilmiştir.

Bu karşılaştırmalar ışığında ani yük değişikliklerine karşı kararlı bir zeki kontrol sistemi tasarlanmıştır.

SUMMARY

In this project, we designed a robust control algorithm for payload variations.

In order to evaluate and formulate our work we worked on and analyzed previous works on the literature. Then we came a conclude about the parameters that we need for evaluation. These parameters are used among those algorithms in order to evaluate them.

Algorithms considered:

- 1-Direct Inverse Modelling Architecture (DIMA)
- 2- Direct Inverse Modelling Architecture (DIMA) Type II
- 3-Feedbak Error Learning Architecture (FELA)
- 4- Neuro- Sliding Mode Architecture (NESMA)

In order to compare those parameters we determined important comparison criteria then we used those parameters in order to compare algorithms

Comparison results obtained are used in order to design new robust intelligent control algorithm for payload variations.

TEŞEKKÜR

Bu tezin hazırlanması sırasında yapılan araştırmalara yaptığı yapıcı öneri, katkı ve yardımlarından dolayı tez danışmanı değerli hocam sayın Doç. Dr. E. Murat Esin'e teşekkürü bir borç bilirim.

Doktora araştırmalarım süresince manevi desteğini hiçbir zaman esirgemeyen eşim Selda Akbaş ve oğlumuz Talha'ya, kızımız Zeynep Betül'e ve akademik çalışma yapmam için manevi desteğini esirgemeyen aileme de ayrıca teşekkür ederim.

Tezin hazırlanışı sırasında simülasyon ve deneysel çalışmalara katkılarından dolayı, Dok. Melikşah Ertuğrul, Dok. Hasan Palaz'a da teşekkürlerimi sunarım.

Doktora tezimin, robotik ve yapay zeka araştırmalarında bulunacak araştırmacılara katkıda bulunması dileğiyle,

GEBZE., Ocak 2005

Yük. Müh. Ertuğrul Akbaş

İÇİNDEKİLER DİZİNİ

ÖZET	iv
SUMMARY	v
TEŞEKKÜR	vi
İÇİNDEKİLER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	ix
ŞEKİLLER DİZİNİ	xi
ÇİZELGELER DİZİNİ	xiii

1. GİRİŞ	HATA! YER İŞARETİ TANIMLANMAMIŞ.
1.1. ÇALIŞMANIN HEDEFİ	HATA! YER İŞARETİ TANIMLANMAMIŞ.
2. ROBOT SİSTEMLERİ.....	5.
2.1. LİTERATÜRDE TEMEL ROBOT TASARIMININ TEMEL İLKELERİ.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
2.2. İKİ SERBESTLİK DERECELİ TEST ROBOTU VE TASARIMI.	HATA! YER İŞARETİ TANIMLANMAMIŞ.
2.3. ROBOT KONTROLÜ KONUSUNDA YAPILAN ÇALIŞMALAR.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
2.4. ROBOT KONTROLÜNDEKİ PROBLEMLER	HATA! YER İŞARETİ TANIMLANMAMIŞ.
2.4.1. Gelişim Süreci	<i>Hata! Yer işareti tanımlanmamış.</i>
2.5 LİTERATÜRDE YAPAY ZEKA ÇALIŞMALARI	HATA! YER İŞARETİ TANIMLANMAMIŞ.
2.5.1 Biyolojik Sinir Ağları	<i>Hata! Yer işareti tanımlanmamış.</i>
2.6 TEMEL YAPAY SINIR AĞLARI ALGORİTMALARI	HATA! YER İŞARETİ TANIMLANMAMIŞ.
2.6.1. Geri Yayınım Algoritması (GERİ YAYINIM).....	<i>Hata! Yer işareti tanımlanmamış.</i>
2.6.2. Radial Basis Functions	<i>Hata! Yer işareti tanımlanmamış.</i>
3. ZEKİ KONTROL SİSTEMLERİ.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
3.1. GİRİŞ	HATA! YER İŞARETİ TANIMLANMAMIŞ.
3.2 EĞİM DÜŞÜMÜ YÖNTEMİ.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
3.3 YAPAY SINIR AĞLARI İLE SİSTEM TANIMA	HATA! YER İŞARETİ TANIMLANMAMIŞ.
3.3.1 Ters Denetim Yöntemi (TDY)	<i>Hata! Yer işareti tanımlanmamış.</i>
3.3.2 Model Dayanaklı Uyarlamalı Denetim (MDUD)	<i>Hata! Yer işareti tanımlanmamış.</i>
3.3.3 Kendi Kendine Öğrenme İle Denetim (KKÖD)	<i>Hata! Yer işareti tanımlanmamış.</i>
3.4 DİNAMİK SINIR AĞLARI İLE DOĞRUSAL OLMAYAN DİNAMİK SİSTEMLERİN DENETİMİ.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
3.5 YAPAY SINIR AĞLARININ ROBOT DENETİMİNDE KULLANIMI.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
3.5.1 Ters Denetim Yöntemi (Direct Inverse Modelling Architecture) (DIMA) ...	<i>Hata! Yer işareti tanımlanmamış.</i>
3.5.2 Geri Beslemeli-Hata Öğrenmeli	<i>Hata! Yer işareti tanımlanmamış.</i>
3.5.3 Kayan Kipli Kontrol.....	<i>Hata! Yer işareti tanımlanmamış.</i>
3.6 BULANIK MANTIK VE BULANIK KONTROLÖRLERİN TEMEL ÖZELLİKLERİ	HATA! YER İŞARETİ TANIMLANMAMIŞ.
3.6.1 Bulanık Kümeler ve Bulanık Mantık	<i>Hata! Yer işareti tanımlanmamış.</i>
3.6.2. Bulanık Mantık ve Mantıksal Yaklaşıklık	<i>Hata! Yer işareti tanımlanmamış.</i>
3.6.3. Bulanık Mantık Kontrolörlerinin Ana Fikri	<i>Hata! Yer işareti tanımlanmamış.</i>
3.6.4 Bulanık Koşullar ve Bulanık Kontrol Kuralları.....	<i>Hata! Yer işareti tanımlanmamış.</i>
3.6.5 Bulandırma ve Durulama Operatörü.....	<i>Hata! Yer işareti tanımlanmamış.</i>
3.6.6. Bulanık Mantık Kontrolörünün Tasarım Parametreleri..	<i>Hata! Yer işareti tanımlanmamış.</i>
3.6.7 Üyelik Fonksiyonları.....	<i>Hata! Yer işareti tanımlanmamış.</i>
3.6.8 Kural Tabanı	<i>Hata! Yer işareti tanımlanmamış.</i>
4.TASARLANAN YÖNTEMLER VE TEST ORTAMININ OLUŞTURULMASI.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
5. ZEKİ KONTROLÖR TASARIMI VE PERFORMANS ANALİZLERİ	HATA! YER İŞARETİ TANIMLANMAMIŞ.

5.1. GİRİŞ	HATA! YER İŞARETİ TANIMLANMAMIŞ.
5.2 ZEKİ SİSTEMLER	HATA! YER İŞARETİ TANIMLANMAMIŞ.
5.3 YAPAY KAYAN KIPLI KONTROL (NEURO SLIDING MODE CONTROL)	HATA! YER İŞARETİ TANIMLANMAMIŞ.
5.3.1. Süreksiz Dinamik Sistemlerde Kayma Kipi	Hata! Yer işareti tanımlanmamış.
5.3.2. Robotun DYS Tabanlı Kayan Kipli Kontrolü.....	Hata! Yer işareti tanımlanmamış.
5.3.2.1. Eşdeğer Kontrol Yöntemi.....	Hata! Yer işareti tanımlanmamış.
5.3.3. Üçüncü Yöntem : Kestirilmiş Eşdeğer Kontrol Yöntemi	Hata! Yer işareti tanımlanmamış.
5.4 (GNFVSC).....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
5.5. ANFVSC.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
5.6. TEST SONUÇLARI	HATA! YER İŞARETİ TANIMLANMAMIŞ.
5.7 TEST YAPILARI	HATA! YER İŞARETİ TANIMLANMAMIŞ.
5.8 SONUÇ	HATA! YER İŞARETİ TANIMLANMAMIŞ.
6. DİNAMİK BİR YÜK DEĞİŞİMİ ALTINDA LİNEER OLMAYAN BİR SİSTEMİN KONTROLÖR DİZAYNI	HATA! YER İŞARETİ TANIMLANMAMIŞ.
6.1 TERS DİNAMİKLER KULLANILARAK KLASİK KONTROLLER DİZAYNI	HATA! YER İŞARETİ TANIMLANMAMIŞ.
6.1.1 İkincil Kontroller Dizaynı	Hata! Yer işareti tanımlanmamış.
6.2 ZEKİ PD KONTROLLER	HATA! YER İŞARETİ TANIMLANMAMIŞ.
6.2.1 Direkt Ters Modelleme Mimarisi (DIMA).....	Hata! Yer işareti tanımlanmamış.
6.2.2 Yapay Kayan Modda Kontrol.....	Hata! Yer işareti tanımlanmamış.
6.3 PD ve YAPAY-PD KONTROLÖRLERİN KARŞILAŞTIRILMASI	HATA! YER İŞARETİ TANIMLANMAMIŞ.
7.SONUÇ VE ÖNERİLER	HATA! YER İŞARETİ TANIMLANMAMIŞ.
8. KAYNAKLAR.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ÖZGEÇMİŞ.....	HATA! YER İŞARETİ TANIMLANMAMIŞ.

SİMGELER VE KISALTMALAR DİZİNİ

DYS	Değişken Yapılı Sistem
SMC	Sliding Mode Control (Kayan Kipli Kontrol)
PID	Oransal-İntegral-Türevsel Kontrol
ADC	Analogdan Dijitale Çevirici
SCARA	Selective Compliance Assembly Robot Arm
DSP	Sayısal İşaret İşlemcileri
YSAS	Yapay Sinir Ağı Sistemleri
ADALINE	ADaptive LINEar combiner
MADALINE	Many ADALINES
LMS	Enküçük Toplamlar
RBF	Radial Bases Functions
TDY	Ters Denetim Yöntemi
MDUD	Model Dayanaklı Uyarlamalı Denetim
KKÖD	Kendi Kendine Öğrenme İle Denetim
CMAC	Cerebeller Model Articulation Controller
DIMA	Direct Inverse Modelling Architecture
FELA	Feedbak Error Learning Architecture
NESMA	Neuro- Sliding Mode Architecture
DIMA	Ters Denetim Yöntemi (Direct Inverse Modelling Architecture)
FELA	Geri Beslemeli-Hata Öğrenmeli Denetim
DYK	Değişken Yapılı Kontrol
BK	Bulanık Kontrolör
FLC	Fuzzy Logic Controller
BMK	Bulanık Mantık Kontrol
YSA	Yapay Sinir Ağları
BYSA	Bulanık Yapay Sinir Ağları
MIMO	Multi Input Multi Output
GNFVSC	Gaussian Neuro Fuzzy Variable Structure Control

RBFNN	Radial Basis Functions Network
VSC	Variable Structure Control
ANFVSC	Adaptive Neuro Fuzzy Variable Structure Control
ANFIS	Adaptive Neuro Fuzzy Interference System
DA	Doğru Akım
AA	Alternatif Akım

ŞEKİLLER DİZİNİ

ŞEKİL 1.1. BİR ENDÜSTRİYEL ROBOTTAKİ ÜÇ TEMEL KISIM TANIMLANMAMIŞ.	2HATA! YER İŞARETİ
ŞEKİL 2.1. SCARA ROBOT KOLUNUN ŞEMATİK GÖRÜNÜMÜ TANIMLANMAMIŞ.	7HATA! YER İŞARETİ
ŞEKİL. 2.2. SCARA ROBOT KOLUNUN ÇALIŞMA ALANI (WORK SPACE). TANIMLANMAMIŞ.	8HATA! YER İŞARETİ
ŞEKİL 2.3. ÇOK KATMANLI BİR YSA.	17HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 2.4. GERİ YAYINIM ALGORİTMASI	18HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 2.5. RBF YAPAY SİNİR AĞI	19HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 3.1. EĞİM DÜŞÜMÜ YÖNTEMİNDE TÜREV YÖNLERİ TANIMLANMAMIŞ.	23HATA! YER İŞARETİ
ŞEKİL 3.2. SİSTEM TANIMA MİMARİSİ	24HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 3.3. DOĞRUDAN UYARLAMALI DENETİM MİMARİSİ TANIMLANMAMIŞ.	25HATA! YER İŞARETİ
ŞEKİL 3.4. KENDİ KENDİNE ÖĞRENME MİMARİSİ İLE DENETLEYİCİNİN EĞİTİLMESİ YER İŞARETİ TANIMLANMAMIŞ.	26HATA!
ŞEKİL 3.5. DENETİM MİMARİSİ	27HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 3.6. DİNAMİK NÖRON YAPISI	28HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 3.7. DİNAMİK NÖRONLAR İLE KURULAN DENETİM MİMARİSİ TANIMLANMAMIŞ.	28HATA! YER İŞARETİ
ŞEKİL 3.8. BULANIK HIZLARIN DİYAGRAM OLARAK GÖSTERİLİŞİ. TANIMLANMAMIŞ.	37HATA! YER İŞARETİ
ŞEKİL 3.9. BULANIK MANTIK KONTROLÖRÜNÜN GENEL YAPISI TANIMLANMAMIŞ.	38HATA! YER İŞARETİ
ŞEKİL 5.1 DİMA	48HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.2. FELA	49HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.3. SÜREKSİZLİK YÜZEYİ İLE DURUM UZAYININ BÖLÜMLENMESİ İŞARETİ TANIMLANMAMIŞ.	51HATA! YER
ŞEKİL 5.4. İKİ KARARSIZ YAPI	51HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.5. DYS	52HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.6. DİNAMİK YÜZEYLERİN ARAKESİTİNDE KAYMA KİPİNİN OLUŞUMU İŞARETİ TANIMLANMAMIŞ.	53HATA! YER
ŞEKİL 5.7 EŞDEĞER KONTROLÜN BLOK DİYAGRAMI TANIMLANMAMIŞ.	55HATA! YER İŞARETİ
ŞEKİL 5.8. İKİNCİ YÖNTEME AİT BLOK DİYAGRAM TANIMLANMAMIŞ.	58HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.9. KESTİRİLMİŞ EŞDEĞER KONTROLLÜ SİSTEMİN BLOK DİYAGRAMI İŞARETİ TANIMLANMAMIŞ.	59HATA! YER
ŞEKİL. 5.10. NEURO SLİDİNG MODE CONTROL	61HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.11. BULANIK KÜME ÜYELİK FONKSİYONLARI TANIMLANMAMIŞ.	63HATA! YER İŞARETİ
ŞEKİL 5.12. GNFVSC YAPISI	64HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.13. RBFNN YAPISI	65HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL.5.14. ANFİS YAPISI	70HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.15. SİSTEM TANIMA MODELİ	71HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.16. JOINT SPACE TRAJECTORY.	72HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.17. TASK SPACE TRAJECTORY.	72HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.18. POZİSYON HATALARI (NEURO-SMC)	73HATA! YER İŞARETİ TANIMLANMAMIŞ.

ŞEKİL 5.19. ANI YÜK DEĞİŞİMİ POZİSYON HATALARI (NEURO-SMC)	73	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.20. POZİSYON HATALARI (DIMA)	73	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.21. ANI YÜK DEĞİŞİMİ POZİSYON HATALARI (DIMA)	73	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.22. POSITION ERRORS OF FELA	73	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.23. POSITION ERRORS OF FELA WITH PAYLOAD	73	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.24. GNFVSC- UC	74	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.25. GNFVSC-UC- NOT BOUNDY	74	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.26. GNFVSC-UC -BOUNDY	74	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.27. GNFVSC-UC- BOUNDY-PAYLOAD	74	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.28. GNFVSC	74	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.29. GNFVSC NOT BOUNDY	74	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.30. GNFVSC NOT BOUNDY-PALOAD	74	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.31. GNFVSC PAYLOAD	74	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.32. ANFIS	75	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.33. ANFIS NIF	75	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.34. ANFIS WITH PAYLOAD	75	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 5.35. ANFIS NIF WITH PAYLOAD	75	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.1. BİRİNCİ VE İKİNCİL KONTROLLER İLE SİSTEM	75	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.2 RBFNN YAPISI	80	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.3. YAPAY KAYAN MODDA KONTROL	85	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.4. İSTENEN VE ÖÇLÜLEN AÇISAL HIZ DEĞERLERİ	88	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.5. X-Y KOORDİNATLARINDAKİ HAREKET	90	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.6 X-Y KOORDİNAT HATALARI.	90	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.7. ÖLÇÜLEN VE İSTENEN AÇILAR VE AÇISAL HIZLAR.	91	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.8. AÇI VE AÇISAL HATAR.	91	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.9. ROBOTA UYGULANACAK OLAN KONTROL KUVVETİ	92	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.10. PD VE NEURO-PD'İN KARŞILAŞTIRILMASI	93	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.11. PD VE NEURO-PD'NİN KARŞILAŞTIRILMASI	94	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.12. PD VE NEURO-PD'NİN KARŞILAŞTIRILMASI	95	HATA! YER İŞARETİ TANIMLANMAMIŞ.
ŞEKİL 6.13. NEURO SMC VE DIMA WITH RBF'NİN KARŞILAŞTIRILMASI	96	HATA! YER İŞARETİ TANIMLANMAMIŞ.

ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1. Robot Kolu Parametreleri	7
2.2. RBF Algoritması	20
5.1. GNFVSC ve ANFVSC Metotlarının Karşılaştırması Tablosu	77
5.2. DIMA, FELA, NEURO SMC Karşılaştırma Tablosu	78
6.1. DIMA, PID, NEURO SMC Karşılaştırma Tablosu	90

1. GİRİŞ

Bu çalışmada, bir endüstriyel robotun kontrolunda gerekli olan kinematik, ters kinematik, gerçek zaman yörünge planlaması, dinamik modelleme, kontrol yöntemlerinin tanıtılması, kontrolör ve kontrol sistemi için öneriler ve yenilikler ortaya konulmuştur.

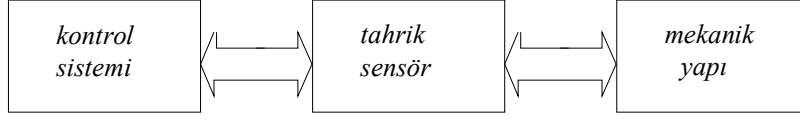
Endüstride üretim kalitesi ve miktarının artırılması ihtiyacı, son yıllarda değişik otomasyon sistemlerine olan talebi artırmıştır. Öyleki ürün kalite ve standardından başka üretim ortamlarının da standartlaştırma çalışmaları hem dünyada hemde ülkemizde sürmektedir.

Üretimde otomasyonun bir parçası olan endüstriyel robotlar yada programlanabilen çok fonksiyonlu manipülatörler, bu bakımdan oldukça önemli olmaya başlamıştır.

Endüstriyel robotların ve robotik sistemlerin gelişimi, artan ihtiyaçlara paralel olarak hızla sürmektedir. Bu gelişmeler, mekanik donanımları bakımından olduğu gibi kontrolör yapıları ve kapasiteleri ile kullanıcı arabirimlerinde de oldukça tatminkar boyutlara ulaşılmıştır. Buna rağmen günümüz robotlarının halen en büyük problemleri hızdır. Hareketi sırasında robotun hızını sınırlayan iki önemli neden, kolun kendi ağırlığı dolayısı ile ortaya çıkan mekanik yapısının dinamiği ile birlikte kontrolörün hesap gücüdür.

Bir robot sisteminde birbiri ile etkileşimli üç temel kısım vardır. Bunlar;

1. Mekanik yapı; bağlaçlar (link), eklemler (joint), dönüştürme ve/veya aktarma organları (transmission),
2. Tahrik düzenleri (motor) ve algılama düzenleri (sensörler),
3. Kontrol sistemi.



Şekil 1.1. Bir Endüstriyel Robottaki Üç Temel Kısım

Robotun performansını artırmak, bu üç kısımda farklı düzenleme ve yenilikler ile yapılacak çalışmalarla sağlanabilir. Mekanik yapıda; yeni malzeme ve mekaniklerin kullanılması ile performans artırılabilen, daha hafif ve katı robot kollarının yapılması mümkün olmaktadır. Düşük ağırlık/mukavemet oranları ile hafif bağlaçların imal edilebilmesi, ivme ve hızın artırılabilmesine olanak sağlar.

Günümüz sürücü elemanlarından fırçasız DA motorları ile kalıcı mıknatıslı AA senkron servo motorlar da performans artışına pozitif yönde etkiyen faktörlerdir. Bu motorların rotor eylemsizliklerinin düşük olması ve ayrıca bakım gerektirmeyen düşük maliyetli olmaları en önemli özellikleridir.

Dişli ve aktarma sistemleri, daima sürtünme ve boşluk gibi doğrusal olmayan (nonlinear) dinamikleri içerir. Bunlardan kurtulmak şimdilik zor görülmektedir. Biraz pahalı bile olsa, yüksek hassasiyetli dişli takımları ve aktarma organları kullanmak zorunluluğu vardır. Yada son yıllarda üzerinde çokça durulan doğrudan sürücülü (direct drive) robotlara yönelik söz konusudur. [Akbaş E. and Esin M., 2001]

Sensörler, robot kontrolünde performansı artırmak bakımından önemli unsurlardan biridir. Özellikle konum sensörleri olan artımsal kodlayıcılar (encoder), konum ölçme doğruluğuna etki eder.

Kontrolörlerin de sisteminin performansına etkisi kaçınılmazdır. Kontrol sisteminde gerekli tüm hesaplamalar, bir kontrol çevrimi içinde yapılabilir olmalıdır. Özellikle model tabanlı modern kontrol yöntemleri, robot dinamik modellerini kullandıklarından sistem performansı artırılabilir. Bu sorunun bir kısmı ancak işlemci teknolojisindeki gelişmeler ile aşılabılır. Bu günkü gelişmeler bu

bakımdan sevindirici boyutlara ulaşmıştır [Fu et al][Koren, Y.][Craig, I.J.][Coiffet, P.][Nof, S.Y.][Asada, H. and Sotino, J.J.E.][Koivo, A.J.][Critchlow, A.J.][Ranky, et al][Spong, J.J. and Vidyasagar, M.][Pillay, P.and Krishnan, R.][Liu, T.and Liu, C.][Sen, P.C.] [Sebestian, T.et al.][Paul, R.P.][Lee, C.S.G.].

1.1. Çalışmanın Hedefi

Bu çalışmanın esas amacı; modern kontrol yöntemleri ile endüstriyel robotların kontrolünün gerçekleştirilmesi ve yeni kontrol yöntemlerinin önerilmesidir. Önerilecek olan yeni yöntemin özellikle endüstriyel ortamlar için önemli bir çalışma kriteri olan ani yük değişikliklerine dayanıklı olması amaçlanacaktır.

Yeni kontrol algoritmalarının tasarlanması ve literatürdeki algoritmaların analizi, karşılaştırılması ve varsa üstün yönlerinin tespiti için iki serbeslik dereceli bir robot simülasyon ortamında tasarlanmıştır. Bu robotun literatürde kullanılan ve daha sonra karşılaştırılacak pek çok kontrol algoritmasının test edildiği robot mekaniği ve kenamatiği ile aynı olması sağlanmıştır. Elde edilen dinamik modelin; adaptif, dayanıklı (robust) adaptif, değişken yapılı (variable structure), kayan kipli (sliding mode) kontrol teknikleriyle kontrolünün sağlanması ve kontrolör yapısı içinde bulanık mantıklı adaptasyon (fuzzy logic adaptation) mekanizmasının geliştirilmesi ile robot kontrolüne yeni bir yaklaşım sunulması hedeflenmiştir. Ayrıca bu yöntemlerin birbirlerine göre karşılaştırmalarının da yapılmasına olanak sağlamaktır. Bu karşılaştırma ile gerçek zamanda çalışma için hangi yöntemin daha uygun olduğu da araştırılacaktır. Ayrıca bu çalışmada robot kontrolörlerinin performans karşılaştırmaları ile ilgili yeni yaklaşımlarda bulunulması öngörülmüş ve özellikle endüstriyel robotlar için çok önemli olan çalışma zamanında ani yük değişiklikleri için kararlı kontrolörler tasarlanması hedeflenmiştir.

Dinamik modelin eldesinde Lagrange-Euler ve Newton-Euler yöntemleri ele alınmış, ancak robotunun simülasyon çalışmaları için L-E yöntemi kullanılmıştır. Ayrıca parametre doğrusallaştırma ile kontrol yöntemlerinin uygulanabilmesi için robotunun regresör matrisi elde edilerek parametre doğrusallaştırma (parameter linearization) yöntemi ile robot modeli elde edilmiştir.

Simülasyon çalışmaları için iki serbestlik derecesinin yapılacak çalışmalar için yeterli olduğuna karar verilmiştir. Çünkü literatürde genelde iki SD (serbestlik dereceli) robotlar için çözümler ve simülasyon çalışmaları verilmiştir. Az sayıda da olsa üç veya altı SD' li çözümler vardır. Burada, çalışmanın daha kolay anlaşılması gereksiz hesap yükünden kurtulmak için, iki SD yeterli görülmektedir.

Bu tezde önce temel yapay zeka kontrol yaklaşımlarına değinilmiştir. Yapay zeka destekli adaptif kontrol yaklaşımları ve dayanıklılık (robustluk) incelemesinden başka Yapay Zeka Destekli Değişken Yapılı Sistem DYS (Variable Structure System, VSS) ve Yapay Zeka Destekli Kayan Kipli Kontrol (Sliding Mode Control, SMC) ve bulanık adaptif kayan kipli robot kontrolü üzerinde durulmuş ayrıca yeni bir konu olarak ani yük değişikliğine karşı kararlı ve endüstriyel uygulamaları yapılabilecek kadar da basit bir sistem olan zeki kararlaştırmalı PID kontrolör tasarımı yapılmıştır.

Elde edilen kontrol yöntemleri iki serbestlik dereceli robot üzerinde uygulanmış ve bazı uygulama sonuçları sunulmuştur.

Tezin organizasyonu şöyle yapılmıştır: İkinci bölümde robot sistemlerinin tasarımında karşılaşılan problemler ve bu problemlerin çözümü için önerilen teknikleri tartışmaktadır.

Üçüncü bölümde yapay zeka teknikleri tartışılmakta ve bu tekniklerin matematiksel modelleri ve bunların uygulamaları ile ilgili yöntemler tartışılmıştır.

Dördüncü bölüm test koşulları (test cases) ve test ve karşılaştırma için parametreleri ve çözülmek istenene temel problemleri açıklamaktadır.

Beşinci bölümde literatürde daha önce yapılmış olan çalışmalar yeni test kriterleri belirlenerek test edilecek, buradan stabilite ile ilgili parametreler, değerler elde edilecek ve kendi aralarında bir sıralama yapılacaktır.

Altıncı bölümde ani yük değişikliği için kararlı yapay zeka teknikleri kullanarak yeni bir kontrol metodu tasarlanacak ve bu metodun klasik yöntemler ile karşılaştırması yapılacaktır.

Yedinci bölüm sonuçlar bölümünü oluşturmaktadır.

2. ROBOT SİSTEMLERİ

2.1. Literatürde Temel Robot Tasarımının Temel İlkeleri

Bu çalışmada kullanılacak ve geliştirilecek kontrol algoritmalarının testi ve karşılaştırılması için gerekli olan en temel eleman algoritmaların koşacağı bir robottur. İster endüstriyel, ister laboratuvar tipi olsun, robot tasarımında makine, kontrol ve elektronik mühendislik dallarının birlikte ekip çalışması yapması zorunluluğu vardır. Bu nedenle bu disipline “mekatronik” (mechatronics) denilmektedir. Bir robotun mekanik yapısı ve tipi uygulama ihtiyacına göre tespit edilir.

Robotun uzuvları çalışma sırasında maruz kalacakları dış etkenlere karşı en az şekil değiştirecek şekilde tasarlanıp imal edilir [Koren, Y.][Craig, I.J.][Pillay, P. and Krishnan, R.][Liu, T. and Liu, C.][Nagy, F.N. and Siegler, A.]. Robotun çalışma bölgesi (work space) içinde yüksek hassasiyetlik ve tekrarlayabilirlik özelliğini sağlayabilmesi için, katı uzuvlarla birlikte, güç iletim elemanları ve yataklamalar yüksek mukavemete sahip olmak zorundadır.

Sürücü elemanları (motorlar) ise ortalama yerçekimi ve ivmelenme momentleri ve robotun kendi ağırlığını taşıyacak şekilde seçilmelidir. Bir robotun serbestlik derecesi, robotun görevini belirleyici en temel unsurdur. Tasarımı yapılacak bir endüstriyel robot, bilgisayar destekli üç boyutlu katı modelleme paket yazılımları ile tüm işlevleri yerine getirecek şekilde tasarlanmak ve araştırılarak geliştirilmek zorundadır. Bu aşamada büyük miktarda en uygun şekle sokma problemleriyle karşılaşilmektedir.

Robot eklemleri lineer ve dönel olmak üzere iki ana gruba ayrılır. Robot eklemleri, güç (moment) ilettikleri için, burada kullanılacak yataklar için yükün yanı sıra eğilme ve burulma momentleri de göz önüne alınarak seçilirler. Özellikle dönel eklemler antropomorfik (insan kolu benzetişimi) robotlarda yerçekiminden doğan momentler yüksektir.

Aktarma ya da iletim organları, motor momentlerini artırıp açısal hızları azaltacak çevrim oranlarına sahip olmalıdır. Bu amaç için çok değişik dişli

mekanizmaları vardır. Genelde üretici firmasının adı ile anılan ve “Harmonik Dişli” denilen iletim organları kullanılmaktadır. Bu dişliler dönel eklemlerde kullanılan son derece uygun boyutlu, yüksek çevirme oranlı, yok denecek kadar az boşluklu, özel amaçlı iletim organlarıdır [Fu, K.S. et al][Pillay, P. and Krishnan, R.].

Bir robotun ayırım yeteneği ise, robotun gerçekleyebileceği en küçük hareketin ifadesidir. Ayırım yeteneği hesabında, genellikle kodlayıcı (encoder) işaret sayısı, tako-generatör ve ADC’ nin kullanılması ve sistemin sürtünmesi, boşluk ve güç iletim millerindeki burulmalar bu hesaba ele alınmalıdır.

Robot sürücü sistemi için kullanılacak motorların seçilirken genellikle kalıcı mıknatıslı DA servo motorlar ile kalıcı mıknatıslı AA senkron servo motorlar tercih edilir.

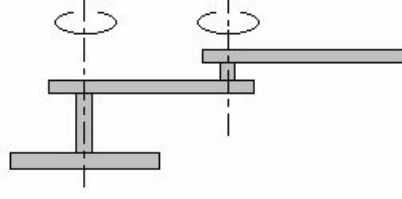
Kalıcı mıknatıslı senkron servo motorlarda rotorda kafes olmadığı için rotor eylemsizlikleri küçüktür. Bu nedenle daha hızlı cevap verirler ve rotordaki kalıcı mıknatıslar sebebiyle rotor kayıpları yok denecek kadar azdır [Pillay, P. and Krishnan, R.][Sen, P.C.].

Bütün bu kıstas göz önüne alınarak tasarlanan robot aslında bir laboratuvar tipi iki serbestlik dereceli bir robottur. Bu robotun literatürdeki pek çok kontrol algoritmasının testi için kullanılmış olan iki serbestlik dereceli robotlarla benzerlik göstermesi sağlanmıştır.

2.2. İki serbestlik Dereceli Test Robotu ve Tasarımı

Bu çalışmada, Şekil 2.1’ de gösterilen iki serbestlik dereceli test robotu (simülasyonlarda test ortamı olarak kullanıldı. Robotun çalışma bölgesi Şekil 2.2 de gösterilmiştir.

Robot dinamiklerinin genel formu (2.1) de gösterilmiştir. Robot model parametreleri standart birimler halinde Çizelge 2.1’ de gösterilmiştir.



ŞEKİL 2.1. SCARA Robot Kolumun şematik Görünümü

$$M(\theta)\ddot{\theta} + V(\theta, \dot{\theta}) = \tau - f_c \quad (2.1)$$

Eğer açısal konumlar (angular positions) ve açısal hızlar (angular velocities) durum değişkenleri (state variables of the system) olarak tanımlanırsa, sistem modeli birinci dereceden diferansiyel denklem olarak tanımlanabilir. (2.1) de verilen denklemin açık şekli (2.2) ve (2.3) de verilmiştir.

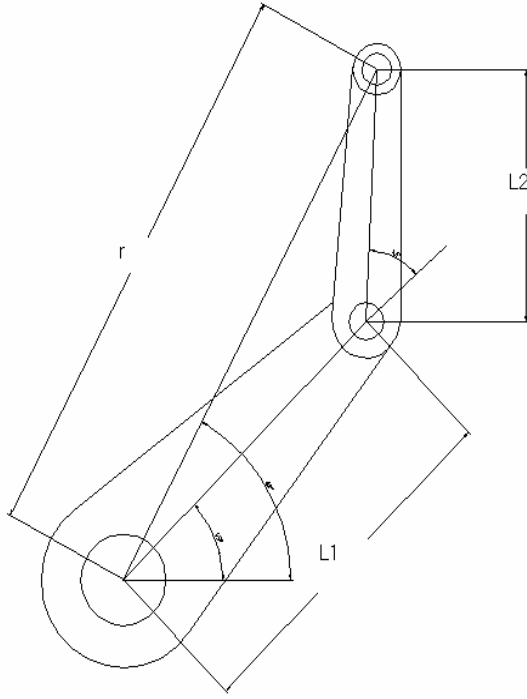
$$M(\theta) = \begin{bmatrix} p_1 + 2p_3 \cos(\theta_2) & p_2 + p_3 \cos(\theta_2) \\ p_2 + p_3 \cos(\theta_2) & p_2 \end{bmatrix} \quad (2.2)$$

$$V(\theta, \theta') = \begin{bmatrix} -\theta_2'(2\theta_1' + \theta_2')p_3 \sin(\theta_2) \\ \theta_1'^2 p_3 \sin(\theta_2) \end{bmatrix} \quad (2.3)$$

$$p_1 = 2.0857, p_2 = 0.1168 \text{ ve } p_3 = 0.1630.$$

Çizelge 2.1. Robot Kolu Parametreleri

Motor 1 Rotor Inertia	0.2670	I1	Payload Mass	0.000	Mp
Arm 1 Inertia	0.3340	I2	Arm 1 length	0.359	L1
Motor 2 Rotor Inertia	0.0075	I3	Arm 2 length	0.240	L2
Motor 2 Stator Inertia	0.0400	I3C	Arm 1 CG	0.136	L3
Arm 2 inertia	0.0630	I4	Arm 2 CG	0.102	L4
Payload Inertia	0.0000	IP	Axis 1 Friction	5.300	F1
Motor 1 Mass	73.000	M1	Axis 2 Friction	1.100	F2
Arm 1 Mass	9.7800	M2	Torque Limit 1	245.0	
Motor 2 Mass	14.000	M3	Torque Limit 2	39.20	



ŞEKİL. 2.2. SCARA Robot Kolunun Çalışma Alanı (work space).

2.3. Robot Kontrolü Konusunda Yapılan Çalışmalar

Bu çalışmada yapay zeka algoritmaları ile iki serbestlik dereceli bir robotun kontrolü üzerinde araştırmalar yapıldı ve yapılan çalışmaya temel teşkil edecek literatürdeki temel çalışmalar belirlendi.

Günümüzde endüstriyel robotların kontrolörleri genellikle basit *PID* tipi kontrolörlerdir. Arimoto ve Miyazaki 1988’de *PID* ler için asimptotik kararlılığı ele almışlar ve Lyapunov’un ikinci yöntemini kullanarak kararlılık ispatı yapmışlardır [Craig, J.J.][Lind, H.].

Mikrokontrolörlerin ve bilgisayarların hız ve performansları arttıkça model tabanlı modern kontrol yöntemlerinin uygulama bulma imkânı da artmaya başlamıştır. Tümüyle robot dinamiğini kullanarak robot kontrolü yapmak gerçekten çok güçlü bilgisayarları gerektirmektedir. Bu uygulama problemleri aşıldıkça önceden belirlenmiş yörünge izlenebilecek veya yörünge gerçekte dış algılayıcı ya da duyargalardan (sensör) gelen bilgilerle değişmesi halinde bile değişen yörünge izlenebilecektir. Örneğin, hareketli bir bant üzerinde cisimlerin de

hareketli olması halinde, bant izlenirken aynı zamanda bant üzerindeki hareketli cisimlerin görüntüleri kamera vasıtası ile alınarak, cismin yer değiştirme bilgileri, yani cismin bant üzerindeki koordinatları bilgisayara aktarılarak, kontrolördeki ters kinematik algoritması üzerinden eklem değişkenlerine çevrilen konum ve hız bilgileri ile cisimlerin robot eli tarafından uygun şekilde kavranıp alınabilmesi veya gerekli bir başka işlemin yapılabilmesi mümkündür. Bu örnekte de anlatıldığı gibi çalışma sırasında hemen (*on-line*) yörünge değişimine karşı adaptasyon ve izleme problemi bugün çözülmüş durumdadır. Ancak hız problemi hala devam etmektedir [Craig, J.J.].

Robot kolu bu değişimi gerçek zamanda izlerken aynı zamanda gerçek zamanda (real time) yörünge planlaması (path-planning) yapıyor demektir. Bu da gerçekten çok güçlü işlemcileri ve donanımları gerektirmektedir. Bu problemler uygulamada sayısal işaret işlemcileri (DSP) veya Transputer'lerle paralel işlemci donanımı kurularak aşılmaya çalışılmaktadır. Bunların fiyat/performans oranları ise hayli düşerek uygun denilebilecek seviyelere gelmiştir [Lind, H.].

Karmaşık modern kontrol algoritmalarının laboratuvar ortamlarının dışına çıkamayışının nedenlerinden birisi ve en önemlisi, hesap gücü problemidir. Gelişmiş bir robotik sistem elde edilince daha gelişmiş görevler beklentisi nedeniyle hesap gücü problemi devam edecektir. Örneğin robot görme işlevlerini de içeren bir uygulama sistemi için gerçek zaman işletim için oldukça güçlü kontrolörler ve kontrol sistemi gerekir. Bu da maliyet artışına neden olur.

İkincisi ve daha da önemli olan modelleme problemidir. Yüksek mertebeden, örneğin 6 SD' li doğru ve kesin bir dinamik modeli geliştirmek oldukça zordur. Endüstriyel tip robotlar, genelde katı (rigid) uzuvlardan oluşur. Katı uzuv kabulü yapılarak yapılacak olan dinamik modelde, hala belirsizlikler vardır. Bu belirsizlikler hakkında kesin bir şey söylemek oldukça zordur. Literatürde bu belirsizlikler yapısal ve yapısal olmayan belirsizlikler olarak ikiye ayrılmıştır [Craig, J.J.].

Yapısal belirsizlikleri uygun şekilde modellemek mümkündür. Ancak yapısal olmayan belirsizlikleri, yani modellenmemiş belirsizlikler ya da bir başka deyişle

modellenememiş belirsizlikler, ya durumlara ya da dış bozuculara bağlıdır. Dinamik modelleme analitik olarak elde edilebilir. Modelde tüm dinamik etkiler ve yapısal parametrelerin değerleri doğru olarak bulunabiliyorsa model tabanlı yöntemler, performans için dış bozucular yokken sıfır hata gibi ideal bir sonuç verebilirler. Hem hesaplanmış moment tekniği hem de ileri besleme kontrolörleri bu tip model tabanlı yöntemlerdir [Koivo, A.J].

Modellenmemiş dinamikler ise daima vardır. Bunlardan kurtulmak da mümkün değildir. Bunlar, genel olarak zamanla değişen sürtünmeler, uzuv ve eklemlerdeki esneklikler ile motorlardaki moment titreşimleri ve doyma gibi karakteristik değişimlerdir. İşte bütün bu belirsizlikler, model tabanlı kontrolörlerin performansını azaltıcı etkenlerdir. Bu ise dayanıklılık (robustluk) analizini zorunlu kılar. Buradaki dayanıklılığın anlamı, belirli bir kararlılık derecesi ve bilinmeyen yük ve ölçme gürültüleri gibi belirsizlikler altında sistem performansının ne kadar etkileneceğidir. Adaptif ve öğrenen (learning) kontrol yaklaşımları ise bilinmeyen bir sistemin kontrolüne ilişkin yaklaşımlardır. Adaptif yöntemlerden kasıt, sistem parametrelerinin görev sırasında tespiti ve sistemin yeni durumlarına dayanarak istenen performansı verecek şekilde yeniden güncellenmesi ile yapılan kontroldür. Çoğu adaptif yöntemlerde sistemin doğrusal olmayan (nonlinear) modeli kullanılmayıp sistemin doğrusal (linear) zamanla değişmeyen modeli kullanılır. Öğrenen kontrol tekniğinde ise sistemin herhangi bir parametrik modeli kullanılmaz. Bunun yerine kontrolör kazançları sistem hareket ederken, uygun şekilde ayarlanır [Coiffet, P.][Sebastian, T. et al][Paul, R.P.].

Üzerinde çalışılan bu konuların yapılan çalışmaya temel teşkil edecek özellikler içermesi özellikle; modellenmemiş dinamikler, kontrol tekniğinin online yada off-line olması gibi hususlar ve bunların kendi aralarında çalışma ortamına göre performansları analiz edilip bunların ışığında bu çalışma ortamları için literatürdeki bu çalışmaların eksik bıraktığı yada kapatamadığı noktalar kapatılmaya çalışılmıştır.

2.4. Robot Kontrolündeki Problemler

Birçok robotun kontrolünde aşağıdaki şu genel problemler göz önüne alınabilir;

- Doğrusal olmayan dinamikler; merkezkaç, Coriolis, yerçekimi.
- Uzuvlar arasındaki dinamik etkileşim (coupling).
- İç parametre değişimleri; sürtünmeler, kütle ve eklemlerdeki değişimler.
- Dış bozucular; ölçme gürültüleri, çalışma sırasında maruz kalınan yüklerden gelen bozucular.
- Modellenmemiş yüksek frekanslı dinamikler; eklemlerin esnemesi.
- Cevap hızı, uygulamada çevrim süresinin azaltılması [Fu, K.S. et al][BRADY, M. et al.][ASADA, H. and SLOTINE, J.J.E.][Koivo, A.J.][Stoten, D.P.].

2.4.1. Gelişim Süreci

Son 15 yıldır robotik konusunda oldukça fazla sayıda araştırma yapılmış ve yine bu konuda oldukça fazla sayıda makale yayınlanmıştır.

Bu makalelerin çoğu robotların kontrol problemine yönelik olmuştur. Bu yayın ve araştırmalarda temelde şu konulara değinildiğini söyleyebiliriz;

- kinematik
- yörünge planlaması ve üretilmesi
- dinamikler
- kontrol yöntemleri
- klasik kontrol PID
- hesaplanmış moment tekniği
- adaptif kontrol
- optimal kontrol
- robust adaptif kontrol
- özuyarlamalı (self tuning) kontrol
- değişken yapı (vss) kontrol
- kayan kipli (sliding mode) kontrol

- h-infinity optimal/adaptif kontrol
- öğrenen (learning) kontrol
- kuvvet kontrolu
- empedans kontrolu
- yapay sinir ağları ve genetik algoritmalar ile kontrol
- compliant kontrol
- bulanık mantıkla kontrol
- engellerden sakınma
- bunların kombinasyonları ve uygulamaları.
- robot görme
- sensörler
- robot programlama dilleri
- nonholonomik robotlar
- yürüyen (mobil) robotlar
- fazla SD' li (redundant) robotlar
- esnek uzuvlu robotlar
- esnek eklemlili robotlar
- modüler robotlar
- mikrorobotik
- kavrama problemi

ve diğerleri sayılabilir [Fu, K.S. et al][Craig, I.J][Nof, S.Y][Critchlow, A.J.][Paul, R.P.][Lee, C.S.G.][Craig, J.J.][Stoten, D.P][Kushal, K.C. and Sneckenberger, J.E.].

2.5 Literatürde Yapay Zeka Çalışmaları

Bu çalışmada üzerimde durulan kontrol yöntemlerinin hepsi yapay zeka tekniklerini kontrol alanına uygulanması ile elde edilen tekniklerdir. Bu süreç uzun ve pek çok çalışmayı temel alarak buralara gelmiştir.

1943 yılı Yapay Sinir Ağı Sistemleri için bir doğuş yılı olarak görülür. McCulloch ve Pitts'in yaptıkları ilk YSAS (1943), vakum tüplerinden ve çeşitli aletlerden oluşuyordu. Bu sistem genel olarak basit mantıksal işlemleri çözmede

kullanıldı. Fakat sistem büyük bir alan kaplıyor, büyük parçalardan oluşuyor ve yavaş işliyordu. Yine de bu iki araştırmacı YSA araştırmalarında, ilk fonksiyonel nöronu yaparak, yeni bir devir açtılar.

Bu tarihten sonra ilk on yıl içerisinde 1949 da Donal Hebb, bilginin, sinirler arası bağlantılarda saklanabileceğini keşfetti. Daha sonra Rosenblatt, 1958 da nöron benzeri bir kavram ortaya attı. Bu kavram perceptron idi ve bu küçük sanal makine desenleri sınıflandırmayı, bağlantı ağırlıklarını değiştirerek öğrenebiliyordu. Bu gelişmeden sonra desen tanıma üzerine çalışmalar gittikçe hızlandı.

60 yılların başında, Bernard Widrow-Marcian Hoff 1960-62 tarafından ortaya konulan ve wirow-Hoff adı verilen güçlü öğrenme kuralları kümesine bağlı ADALINE (ADaptive LINEar combiner) adlı bir sistem icat edildi.

Bu kural kümesine göre desen tanıma sırasında oluşan hata payları karesel bir orantı ile hesaplanıp sisteme geri gönderiliyordu. Böylece sistem, tanıma işlemini daha doğru bir şekilde yeniden yapıyordu. Bu sistem daha sonraları geliştirilerek MADALINE (Many ADALINES) yapıldı. Bu alet, desen tanıma da, hava durumu tahmininde ve uyumluluk kontrol sistemlerinde kullanıldı.

Bundan sonra araştırmalar, daha çok olayın formülize edilmesi ve yeni matematiksel formüller üretme çerçevesinde yapıldı. Amari (1972-77), Fukushima (1980), Kohonen (1977-82, 84, 88), Anderson (1977) ve Hopfield (1982,1984) YSA üzerinde çalışmalar yaptılar. Bütün bu araştırmalara rağmen YSA 80'li yılların sonuna kadar pek ilgi odağı olamadı. John Hopfield (1982-84) hafıza ilişkilerini anlamak üzere bir ağ mimarisi geliştirdi. Bu gelişme YSA konusundaki çalışmalara hız verdi ve bu çalışmaların ilgi odağı olmasını sağladı. YSA araştırmalarını hızlandıran başka bir gelişme de 1986 yılında, McClelland ve Rumelhart tarafından yapıldı. Geliştirilen bu yeni kurallar 1960'dan 1987'ye kadar olan süreçteki, çözilemeyen problemleri ortadan kaldırdı. Böylece yeni YSA araştırma programlarının ortaya çıkışları hızlandı, araştırmalar kuvvetlenmeye başladı. Bunu, dünya çapında verilen konferanslar, yeni yayınlar izledi. YSA ile çözülecek problemlerin sayısı ve bu daldaki uygulamaların âdeti azlıktan çokluğa doğru hızla ilerledi. 1990'larda artık YSA' ı için özel makineler, sistemler, mikro-çipler,

yazılımlar geliştirildi, araştırma grupları arttı, kullanım alanı muazzam derecede genişledi aynı zamanda bu dala aktarılan maddi kaynak inanılmaz derecelere yükseldi.

Yapay sinir ağları aşağıdaki gibi tanımlanabilir: Pek çok paralel işlem yapan elemandan oluşmuş ve bu elemanların fonksiyonu da YSA'nın yapısını belirlemek olan bir sistemdir. Bu sistemde YSA'nın yapısı bu elemanlar arasındaki bağlantının şekli ve kuvveti ile belirlenir.

2.5.1 Biyolojik Sinir Ağları

Bir sinir hücresi diğer sinir hücrelerine Axon ve Dendritlerle bağlıdır. Bilgi akışı elektrik sinyalleri olarak bu kanallardan iletilir. Sinir sistemini oluşturan ve birbirlerine Axon ve Dendritlerle bağlı, aralarında elektriksel akış olan, birbiriyle etkileşim içinde olan, canlılarda bütün refleksel, hareketsetel, algısal, düşünsel olayları sağlayan canlı hücrelere sinir hücreleri denir. İnsan sinir sistemi yaklaşık olarak 1012 (1 trilyon) nöron (sinir hücresi) içerir. Sinir sisteminin temel yapıtaşı olan nöronlar, çeşitli uyarılara kasılarak yanıt verirler.

Bir sinir hücresinin bir axon çıkışı vardır. Daha sonra bu axondan yüzlerce bağlantı çıkabilir. Ve sinir hücrelerine giren yüzlerce dendritler vardır. Bu dendritlerin herbiri bir sinir hücresinden çıkan axona bağlı kanallardır. Kısaca sinir hücresine giren kanallara (bilgi alma) dendrit, çıkışı sağlayan kanala (bilgi verme) axon denir. Yandaki şekilde bir sinir hücresi görülmektedir. Görüldüğü üzeri kalın çıkış Axon, ince girişler ise Dendritlerdir. Sinir hücrelerinin bilgi göndermesi üç çeşittir :

- 1) Elektriksel
- 2) Kimyasal
- 3) Mekanik.

Her bir gönderme tepkisine sinaps adı verilir. Bir sinir hücresine giren sinapslar, o hücrenin özelliklerine göre işlenir ve tek bir çıktı (sinaps) axondan gönderilir. Şunu unutmamak gerekir ki her bir sinir hücresine sayısız dendritler girebilir.

Her bir sinir hücresi bir eşik değerine sahiptir. Buna eşik (threshold) değeri adı verilir ve Teta ile gösterilir. Eğer gelen sinapsların toplu bir işlemten geçirildikten sonraki sonuç değeri threshold değerini aşabilirse, ulaşılan değer axondan sinaps olarak gönderilir. Ayrıca her bir dentritin ağırlık değeri vardır. Gelen bilgi bu ağırlık değeri ile çarpılır (işlenir) ve önemine göre (ağırlık değerinin büyüklüğüne göre) yeni değeri oluşturulur ve hedef sinir hücresine yeni değeri ile girer. Burada dikkatimizi çekmesi gereken nokta, ilişkilerin derecelendirilmesidir. İki veya daha fazla bilgiler arasındaki bağlar ilişkilendirilirken, ağırlık değerleri (önemlilik) belirlenir. Eğer sonradan bu önem derecesi değiştirilmesi gerekirse, bu değerler üzerinde oynama yapılır. Bilgisayar uygulamalarında ağırlık değerlerinin önemi büyüktür. Her bir sinir hücresinin değişik eşik değerlerine sahip olduğunu belirtmiştik. Sinir hücrelerindeki bu değerler, kendi özelliklerine göre değişir.

Beyin ve omurilik birlikte Merkezi Sinir Sistemini oluştururlar. Beyinden ve omurilikten zuhur eden, vücudun her yerine ulaşan sık ve beyaz sinir hücreleri de Çevresel (Peripheral) Sinir Sistemini oluşturur. Merkezi sinir sistemi iki bölümden oluşuyordu. 1) Beyin; bütünleşmiş, kurumsal, düşünce, davranış kontrolü. 2) Omurilik; beyinden gelen ve beyine giden mesajları taşır, omurga refleks merkezidir.

Beyin canlıların en önemli organıdır. Bütün canlılar bu organ sayesinde vücutlarını kontrol eder ve karar verirler. Bu organizmalarda çok daha gelişmiştir. Bir insan beyni yaklaşık 1.4 kg ağırlığındadır. Buna rağmen içerisinde milyarlarca sinir hücresi barındırırlar. Peki, zeka neye bağlıdır? Beyindeki sinir hücrelerinin fazlalığına veya beynin büyüklüğüne mi? Bunun cevabı kesin olarak bilinmese de genel kanı, sinir hücreleri arasındaki bağların çokluğunun önemli rol oynadığıdır. Yani sinir hücreleri arasındaki bağlar (dentritler) ne kadar karışık ve çoksa, düşünce gücünün o denli fazla olduğu görüşü yaygındır. Ve bu bağlar beyin jimnastiği yaptıkça, düşündükçe gelişir. Aslında bu bağların her biri bilgiler arası bağlantıları ifade eder. Burası çok önemlidir çünkü bilgisayar ortamında da önemli bir noktayı teşkil eder. Beyinde bir diğer merak konusu da hafızadır. Bilgiler nerede depolanır? Bunun da cevabı henüz bilinmiyor. Bir görüş bilgilerin sinir hücrelerinde saklandığıdır, diğer bir görüş ise beynin özel bir hafıza merkezi olduğudur. Fakat bu konudaki araştırmalar pek hızlı yürüyememektedir ve genelde bulgular teorik

kalmaktadır. Bunun başlıca nedenlerinden bir canlı denek kullanılamamasıdır. Bilgisayarın araştırma alanına girişine kadar beyin hakkında bilinenler çok azdı. Daha sonra sinir hücreleri bilgisayarlar da simüle edilmeye ve duruma göre tepkileri kontrol edilmeye başlandı. Bir bakıma beynin simülasyonu yapıldı. Tabii ki milyarlarca hücreyi simüle etmek şu anki teknoloji itibari ile imkânsızdır.

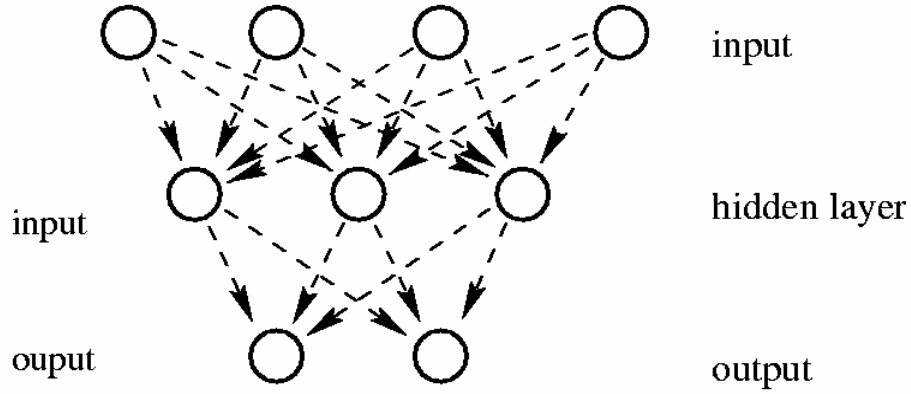
Peki, sinirler bilgisayar ortamında nasıl uygulanır. Aslında kullanılan sistem tamamen algoritmalarından oluşmuştur ve mantık olarak sinir hücreleri örnek alındığından isim olarak bu seçilmiştir.

Temel olarak bir yapay zeka sistemi genellikle 3 tabakadan oluşmuş şekilde gösterilir. İnsan beyninde ise bu katmanlar sayısız denecek kadar çoktur.

1. Tabaka giriş tabakasıdır ve girdi bilgileri buradan alınır ve sisteme verilir.
2. Tabaka ise gizli tabakadır ve kullanımı kullanıcıya kalmıştır. Bu tabaka, ilişkiler arasındaki bağları daha karmaşık hale getirebilir.
3. Tabaka ise çıktı tabakasıdır ki girdiler işlenerek sonuçlar buradan alınır. Her bir yuvarlağın (sinirin) bir fonksiyonu ve eşik değeri vardır. İçi dolu ufak daireler ise bağlantı ağırlıklarını temsil eder.

2.6 Temel Yapay Sinir Ağları Algoritmaları

Yapay sinir ağlarının ortaya çıkışı insan beyninin nasıl çalıştığı ve işlemleri nasıl gerçekleştirdiğine dayanmaktadır. İnsan beyni milyonlarca nörondan oluşmuştur ve bular birbirlerine çok sıkı bir şekilde bağlıdırlar. YSA da bu sisteme benzer bir yapı göstermekte olup farklı şiddete birbirine bağlanmış pek çok elemandan oluşmuştur.



ŞEKİL 2.3. Çok katmanlı bir YSA.

2.6.1. Geri Yayınım Algoritması (GERİ YAYINIM)

En temel YSA şekillerinden biri olan geri yayılım' da en önemli özelliği sistemin hatayı LMS algoritması kullanarak sıfırlamaya çalışması bunu yaparken de hatayı sistemde geriye doğru yayması olarak gösterebiliriz. Bu sistemde bir giriş katmanı bir çıkış katmanı ve arada saklı (hidden) katmanlar olmak üzere 3 çeşit katman vardır. Saklı katman sayısı uygulamadan uygulamaya değişmekte olup genelde 2 saklı katman kullanılır.

Bu katmanlardaki düğüm (node) sayısı ise giriş katmanında giriş bilgisinin boyutuna, ara katmanlarda seçilebilir, çıkış katmanında ise çıkış bilgisinin boyutuna göre karar verilir. Bu yapının en önemli özelliklerinden birisi ara katmanlardaki sayıdır. Bunlardaki nöron sayısı ile çok esnek bir öğrenme modelinin kurulabilmesi mümkündür.

Minsky ve Papert 1969 yılında iki katmanlı bir yapının pek çok kısıtlamayı çözebileceğini gösterdiler. Fakat onların en büyük eksikliği girişten ara katmanlara olan ağırlıkları nasıl hataya göre düzelteceklerini bilmemeleriydi. Bu soruya cevap Rumelhart, Hilton ve diğerleri tarafından geldi. Bu çözümün ardında yatan ise ara katmandaki birimlerdeki hata çıkış katmanındaki hataların geriye doğru yayılımı ile

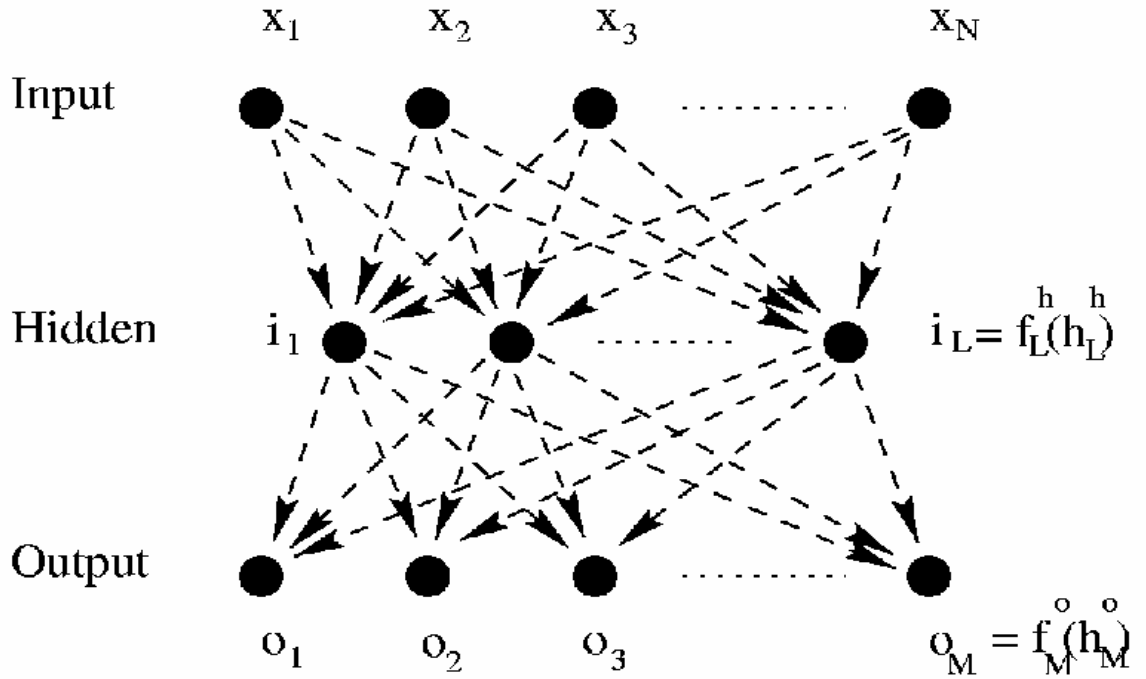
bulunabileceği idi. Buna göre birimler arasındaki ağırlık (şiddet) iyileştirmesi aşağıdaki hata formülüne göre olmaktadır.

$$E = \frac{1}{2} \sum_k \delta_k^2$$

Buna göre de

Ağırlık iyileştirme = Öğrenme katsayısı x Bölgesel gradyant x giriş sinyali

İşte bu yüzden bu ilişkiye geri yayılım denmiştir.



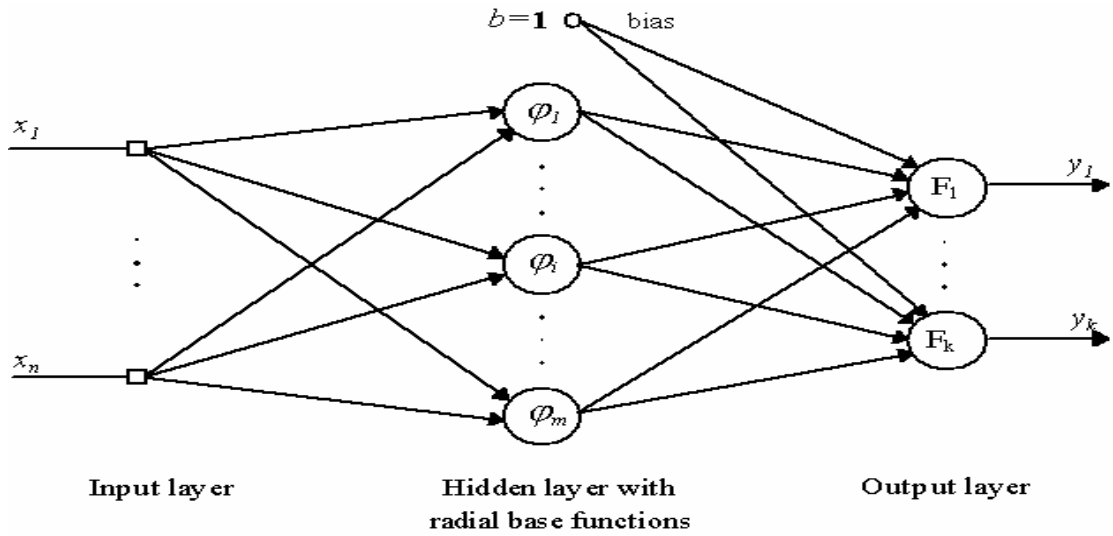
Şekil 2.4. Geri yayılım Algoritması

Bu algoritmanın pek çok değişik varyantı mevcuttur. Yapılan çalışmalar Levenberg-Marquart algoritmasının diğerleri arasında en hızlısı olduğunu göstermiştir.

2.6.2. Radial Basis Functions

RBF öteden beri fonksiyon atamada kullanıla gelen ve pek çok başarılı sonuçlar elde edilen bir metottur. Dolayısıyla bu metodun YSA ile kullanılması gayet iyi sonuçlar doğurmaktadır. RBF YSA diğer temel YSA çeşitlerinden farklılık arz

etmektedir bunun da en önemli sebebi genelde YSA sistemlerinde kullanılan aktivasyon fonksiyonu olarak kullanılan fonksiyonların aksine burada Gaus fonksiyonu kullanılmıştır. RBF YSA da yapı bellidir. Bir giriş katmanı bir ara katman ve birde çıkış katmanı vardır. Burada giriş ve çıkış katmanlarındaki nöron sayısı öğrenmek için kullanılacak veri ile ilgilidir ki bu yapı hemen hemen bütün YSA' larda aynıdır. Önemli olan Ara katmandaki nöron sayısı ve de aktivasyon fonksiyonu olarak kullanılan Gaus fonksiyonunu merkezinin neresi olacağıdır.



ŞEKİL 2.5. RBF Yapay sinir Ağı

Yukarı belirtildiği gibi bu yapıyı kullanabilmek için öncelikle merkezleri seçmek gerekmektedir ki bu da en önemli konulardan biridir.

RBF YSA' da merkezlerin seçimi:

1- Rasgele seçim

Burada kullanılacak olan veri içerisinde rasgele bazı noktaları merkez olarak atarız.

En büyük problem ise RBF lerin artmasıdır.

2- K means clustering algoritması:

Burada yapılan işlem yukarıdan farklı olarak veriler arasında bir gruplandırmaya gidilerek veriyi en iyi temsil edecek merkezleri seçmektir.

Bu algoritmada grup sayısının önceden bilinmesi gerektiği için ön bir bilgiye ihtiyaç duymaktadır.

Hata ise aşağıdaki formül ile ifade edilir.

$$E = \sum_{i=1}^N \sum_{j=1}^m M_{ij} \|x_i - c_j\|$$

Çizelge 2.2. RBF Algoritması

1. Adım. Fonksiyon merkezlerini ata
Başlangıçta merkezler rastgele atanır.
2. Adım Bütün verileri en yakın fonksiyon merkezi içinde topla
Her bir veri için x_i , x_i yi grup j^* ye ata , $\|x_i - c_{j^*}\| = \min_j \|x_i - c_j\|$
3. Adım c leri hesapla
Her grup için, $c_j = \frac{1}{m_j} \sum_{x_i \in \text{group } j} x_i$
 m_j grup j deki veri sayısı.
4. Hiçbir değişikliğe gerek kalmama kadar 2. adıma dön

3- Kohonen YSA kullanılarak seçim.

Burada Kohonen algoritması olarak bilinen YSA kullanılarak veriler gruplandırılırlar. Adımlar tablo 1.de verilenle aynıdır

4-MaxiMin Dictance Algoritması

Diğer bir metot olan MaxiMin algoritmasında ki adımlar aşağıdaki gibidir.

1. Adım İlk fonksiyon merkezlerini ata
Rasgele seçilen merkezler
- Adım 2. Her bir veri için minimum mesafeyi hesapla
Her bir veri x_i , $d_i = \min_j \|x_i - c_j\|$
- Adım 3. En büyük minimum mesafeli veriyi seç
 $D = d_{j^*} = \max_i \{d_i\}$
eğer $D > \frac{1}{2} \{D_{\text{önceki}}\}$ yada $D > D_{\text{min}}$ (D_{min} belirtilen minimum mesafe
veri x_j yeni bir merkez olarak ata $c_j = x_j$.)

Adım 4.Yeni bir merkez oluşturulmayıncaya kadar 2. Adıma dönerek devam et

3. ZEKİ KONTROL SİSTEMLERİ

Kontrol sistemlerindeki bu çalışmada çözülmek istenen problemlerin genel çözüm yaklaşımlarının başında yapay zeka teknikleri gelmektedir. Özellikle sistem dinamiğinin matematik modelinin tam bilinmediği ya da çalışma esnasında değiştiği durumlarda yapay zeka en çok kullanılan ve en başarılı sonuçlar alınan tekniktir.

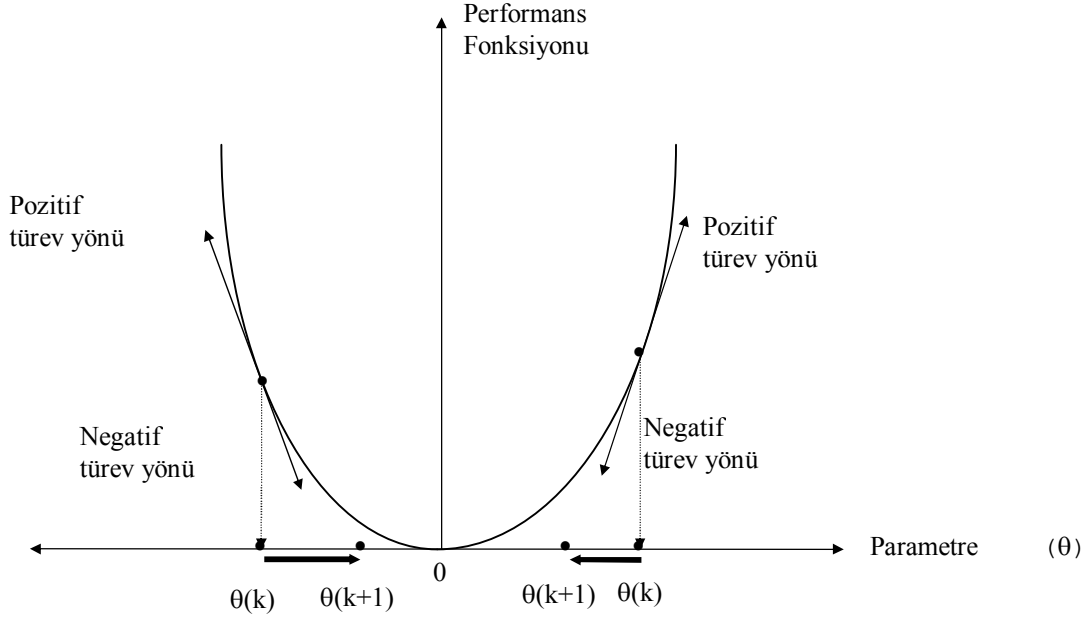
3.1. Giriş

Yapay sinir ağları, paralel bilgi işleme ve zaman içerisinde genelleme özelliklerine sahip olduklarından dolayı 1990'lı yılların popüler araştırma konularından biri oldu. Bugün birçok değişik mimari ve mevcut öğrenme kuralları ile sistem tanıma ve kontrolünde, işaret işlemede kullanılabilmektedirler.

Bu çalışmada yapay sinir ağlarının öğrenme metodolojisi kullanılarak, zamanla değişen özelliklere sahip bir dinamik sistemin (2 serbestlik dereceli robot) kontrolünü sağlayan bir yöntem tanıtılmaktadır. Bu çalışmada kullanılan yöntemlerde öğrenme algoritmasında sistem parametrelerinin güncellenmesi (weight update) için eğim düşümlü yöntem kullanılan yöntemdir. Bunun başlıca sebebi bu tür yaklaşımlar için ideal çözüm olması ve kolay uygulanabilir olmasıdır.

3.2 Eğim Düşümü Yöntemi

Birçok minimizasyon probleminde kullanılan *Eğim Düşümü Yöntemi* (*Gradient Descent Method*) yapay sinir ağlarının eğitiminde de oldukça iyi bir şekilde kullanılabilmektedir. Temel olarak ağ parametrelerinin tanımladığı uzay içerisinde performansın en iyi olduğu parametre kombinasyonunu bulmayı sağlayan bu yöntemin çalışma esası, parametre güncelleme yönünün performans fonksiyonunun belirlenen parametreye göre kısmi türevinin gösterdiği yönün tersi şeklinde olmasıdır. Bu, Şekil 3.1. de tek boyutta, karesel bir performans fonksiyonu için gösterilmiştir.



ŞEKİL 3.1. Eğim düşümü yönteminde türev yönleri ve parametre güncelleme

Her bir iterasyonda, performans fonksiyonunun θ parametresine göre türevi hesaplanır ve bir sonraki adımda bu türevin gösterdiği yönün tersi yönde parametre güncellenir. Bu, (3.1) ile verilebilir ve literatürde Eğim Düşümü Yöntemi olarak bilinir.

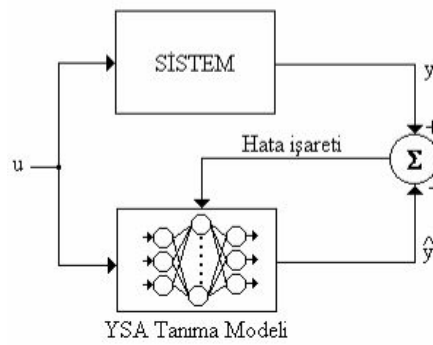
$$\Delta\theta = -\eta \frac{\partial J}{\partial \theta} \quad (3.1)$$

Bu mekanizma içerisinde görülen η parametresi öğrenme katsayısıdır ve tipik değeri (0,1) aralığındadır.

3.3 Yapay Sinir Ağları ile Sistem Tanıma

Denetleyici tasarımı, denetlenecek sisteme ilişkin temsili bilgi gerektirir. Birçok uygulamada matematiksel model ile gerçek sistem arasında, etkileri hassas bir şekilde model içerisine katılmayan uyumsuzluklar olup bunların da denetleyici başarısına olumsuz etkisi sıkça karşılaşılan bir durumdur. Bu nedenle denetleyici tasarımında kullanılan bilgi, sistemi tatmin edici ölçüde tanımlayıcı niteliğe sahip

olmalıdır. Tanımayı güçleştiren başlıca etkenler sistemin doğrusal olmayan davranışı, doyma, gecikme ve zamanla değişen parametrelerdir. Bu yüzden hem matematiksel olarak işlenebilirliği olan hem de sisteme ilişkin bu davranış özelliklerini içeren bir model bulmak çözümü oldukça güç bir sorundur. Tanınacak sistem T_p ile gösterilsin. T_p kapalı uzay $U \subseteq \mathbf{R}^n$ 'i $Y \subseteq \mathbf{R}^m$ 'e eşleştirir. Amaç T_i ile gösterilen ve T_p yi yeteri derecede iyi bir şekilde temsil eden bir sınıfı bulmaktır. Diğer bir deyişle T_i (3.2) denklemini sağlayacak biçimde oluşturulmalıdır. Şekil 3.2 sistem tanıma yordamını şematik olarak göstermektedir.



Şekil 3.2. Sistem Tanıma Mimarisi

$$\|T_i(u) - T_p(u)\| \leq \varepsilon, u \in U \quad (3.2)$$

Yapay sinir ağları ile sistem tanıma işleminin en önemli yanı uyarlamalı bir değişken güncelleme yordamının belirlenmesidir. Bu yordam, gerçek sistem ile onu taklit edecek modelin çıkışları arasındaki farkın enküçüklenmesini sağlamalı ve bu gerçek sistemden elde edilen gözlem sonuçları ile uyumlu olmalıdır.

3.3.1 Ters Denetim Yöntemi (TDY)

Bu yöntemde YSA tabanlı denetleyici, sistem davranışının tersini öğrenir. Bu işlemde denetlenecek sistemden ileri yönde elde edilen giriş/durum/çıkış bilgisi kullanılır. Sistem davranışının (3.3) denklemi ile verildiğini varsayalım;

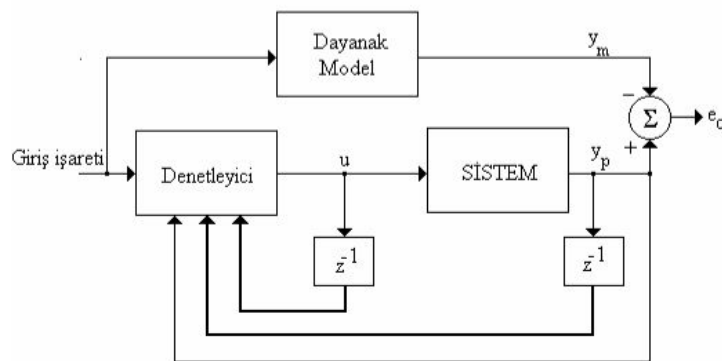
$$x(k+1) = F[x(k), u(k)] \quad (3.3)$$

$$u(k) = G[x(k+1), x(k)] \quad (3.4)$$

Burada $x \in R^m$ ve $u \in R^n$ vektörel büyüklüklerdir. Sistem R^{m+n} uzayından R^m uzayına doğrusal olmayan bir eşleştirme yapmaktadır. Amaç, bu eşleştirme bilgisinden sistem girişlerini çekmektir. Eğer bir denetleyici durum geçiş bilgisini kullanarak bu geçişe neden olan girdileri hesaplayabilirse, bu denetleyici sistemin tersini gerçekleştirebilir ve sistemi denetleyebilir. Bunun için gerek şart, sistem durumlarının gözlenebilir olmasıdır. Yordam yapay sinir ağları açısından değerlendirilirse, eğitim çiftleri kümesinin, ayrıklaştırılmış sistem denklemlerinden elde edilebileceği ve hata geriye yayma yöntemi ile denetleyicinin eğitilebileceği ortaya çıkmaktadır.

3.3.2 Model Dayanaklı Uyarlamalı Denetim (MDUD)

Bu çalışmada ele alınacak temel yöntemlerden biri de model dayanaklı uyarlamalı denetim yordamıdır. Yöntem, denetim girişlerini hesaplarken sistem çıkışlarının kararlı bir istenen model gibi davranmasını da sağlamaktadır. Birçok uygulamada sistem parametrelerinin tam olarak bilinmemesi, tasarlanan denetleyicinin yeterli ölçüde başarılı olmasını engeller. Buna karşılık, denetleyicinin değişen şartlar altında kendisini ortama uyarlaması gerekir. Bu noktada, yapay sinir ağları sistem parametrelerinin tahmin edilmesinde ve en uygun girişin hesaplanmasında kullanılabilir.



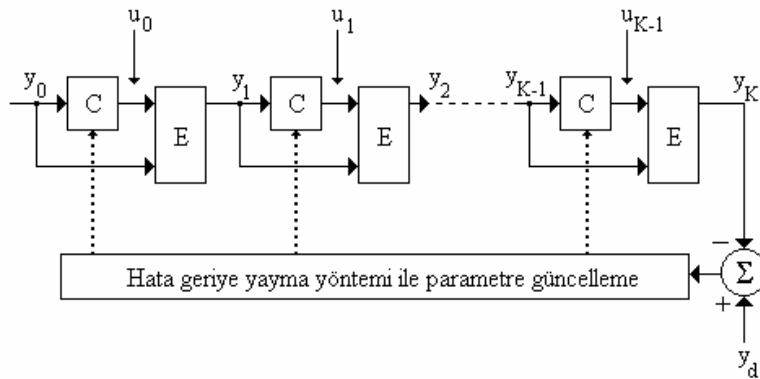
Şekil 3.3. Doğrudan Uyarlamalı Denetim Mimarisi

3.3.3 Kendi Kendine Öğrenme İle Denetim (KKÖD)

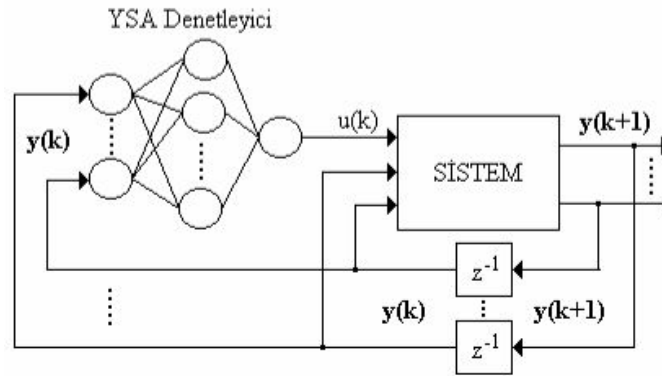
Bu yöntem dışarıdan referans işareti almaz. Denetleyici, önceden tanımlanan sürekli hal çıkışlarına ulaşmayı sağlayacak biçimde eğitilir. Denetleyici eğitimi hata geriye yayma yöntemi ile yapılır.

Benzetimin ilk aşaması sistemin bir yapay sinir ağı tarafından tanınmasıdır. Şekil 5 de E kutuları sisteme ait yapay sinir ağı modelleridir. Denetim mimarisi K adımda sistem çıkışlarının başlangıç değerinden (y_0) istenen son değerine (y_d) getirilebilmesini sağlamaktadır. Adım sayısı tasarımcı tarafından belirlenmektedir. Eğitim için başarı ölçütü (5) ile verilmiştir.

$$J = E\left(\|y_d - y_K\|^2\right) \quad (3.5)$$



Şekil 3.4. Kendi kendine öğrenme mimarisi ile denetleyicinin eğitilmesi



Şekil 3.5. Denetim mimarisi

Enküçükleme işlemi K. adımda çıkış hatasının hesaplanıp hata geriye yayma yöntemi ile ilk adıma kadar getirilmesi ve bu işlem sırasında ağırlıkların güncellenmesine dayanmaktadır. Şekil 3.4. de gösterildiği biçimde K. aşamada yapılan parametre güncellemesi (K-1). aşamada yapılacak parametre güncellemesini etkilemektedir. Bu, değişimlerin K adım boyunca hesaplanarak biriktirilmesini ve elde edilen değişim vektörünün kullanılmasını gerektirmektedir. Uygulamada her adımda parametrelerin güncellenmesinin de başarı ölçütünü etkilemediği görülmüştür. Eğitilen YSA denetleyici Şekil 3.5’de görüldüğü gibi sisteme yerleştirilmektedir

3.4 Dinamik Sinir Ağları İle Doğrusal Olmayan Dinamik Sistemlerin Denetimi

Bu bölümde yeni bir nöron modeli ele alınacak ve bu model ile kurulan ağların denetim uygulamaları araştırılacaktır. Modelin yapısı Şekil 3.6 de görülmektedir. Nöron tek girişlidir, giriş katında ikinci dereceden bir sistem yer almakta ve bu sistemin cevabı ölçeklenerek doğrusal olmayan bir fonksiyondan geçirilmek suretiyle çıkış elde edilmektedir.

güncellenmesi (3.10) denkleminde ifade edildiği gibi eğim düşümü yordamının uygulanması şekline dönüşmektedir.

$$w_{k+1} = w_k - \mu \frac{\partial J}{\partial w} \quad (3.10)$$

$$J = \frac{1}{2} E(e^2(k)) \quad (3.11)$$

$$e(k) = y_d(k) - u(k) \quad (3.12)$$

Daha açık biçimiyle parametre güncelleme kuralı (3.13),(3.14) ve (3.15) denklemleri ile verilebilir ($i=0,1,2$ ve $j=1,2$)

$$\Delta a_i(k+1) = \mu_{a_i} E(e(k) g_s^2(k) \text{sech}^2[v_1(k)] x(k-i)) \quad (3.13)$$

$$\Delta b_j(k+1) = -\mu_{b_j} E(e(k) g_s^2(k) \text{sech}^2[v_1(k)] v_1(k-j)) \quad (3.14)$$

$$\Delta g_s(k+1) = g_s(k) \mu_{g_s} E(e(k) \text{sech}^2[v_1(k)] v_1(k)) \quad (3.15)$$

3.5 Yapay Sinir Ağlarının Robot Denetiminde Kullanımı

Yapay sinir ağları birçok araştırma alanlarında yeni bir ufuk açtı. Bu anlayış, kontrol kavramına da yeni bir düşünüş tarzı getirdi. Artan teknolojik talepler kontrol mühendislerini daha yetenekli denetleyiciler tasarlamaya yöneltti. Bu noktada yeni bir yaklaşım olarak paralel veri işleyebilme özellikleri, uyarlanabilirlik ve güçlü dönüştürme yetenekleri nedeniyle yapay sinir ağları önerildi. Özellikle öğrenebilme özelliği bu ağları daha çekici kıldı.

Yapay sinir ağlarının eğitiminde birçok yöntem kullanılmaktadır. Bu çalışmada iki yöntem ele alınmıştır: hata geriye yayma yöntemi ve Levenberg-Marquardt en uygun şekle sokma tekniği. Hata geriye yayma yönteminde performans fonksiyonunun birinci türevi kullanıldığından öğrenme zamanı özellikle minimum etrafında çok uzundur, buna karşılık Levenberg-Marquardt yönteminde öğrenme zamanı daha kısadır ve bu, ikinci türevlerden gelen ek bilginin de kullanılmasına atfedilir. Levenberg-Marquardt yönteminde işlem karmaşıklığı ve donanım ihtiyaçları daha fazladır.

Doğrusal olmayan dinamik sistemlerin tanımlanması denetleyici eğitiminde önemli bir bölümdür. Bu yüzden tanıma, bu çalışmanın kapsamına alınmış ve simülasyon sonuçları ile birlikte tartışılmıştır. Prosedürün altında yatan ana düşünce ilgilenilen sisteme ilişkin düzenli ve matematiksel olarak kolay işlenebilir bir modelin elde edilmesidir.

Bu iki eğitim yöntemine ve tanıma kavramına dayanarak çeşitli kontrol stratejileri bu çalışma içerisinde tartışılmıştır. Hata geriye yayma ile kontrol, sistem tersini elde etme ile kontrol, kendini ayarlayan kontrol, model referanslı uyarlamalı kontrol, kendi kendine öğrenme ile kontrol ve dinamik sinir ağı modeli ile kontrol yaklaşımları ile denetleyici tasarımı metodolojisi birçok simülasyon sonuçları ile ele alınmıştır.

Günümüzde pek çok kontrol tekniği mevcuttur. Bu kontrol teknikleri klasik kontrol ve zeki kontrol olarak iki gruba ayrılabilirler.

Klasik kontrol tekniklerine örnek olarak:

- 1-PID kontrol
 - 2-Model Referans Uyarlamalı Kontrol(Model Reference Adaptive Control)
 - 3-Öz Ayarlamalı Regülatörler (Self tuning regulators)
 - 4-Hesaplanmış Uygulacak Kuvvet (Computed Torq)
 - 5- Kayan Kipli Kontrol (Sliding Mode Kontrol)
- verilebilir.

Diğer bir grup olan zeki kontrole ise

- 1-Yapay sinir ağları ile kontrol
 - 2-Bulanık mantık ile kontrol
 - 3-Genetik algoritmalar ile kontrol
- örnek olarak verilebilir.

Kontrol teorisinin devamlı olarak ilerlemesine insanların artan ihtiyaçları ve buna paralel olarak zorlaşan kontrol problemlerini gösterebiliriz.

Günümüzde kontrol edilmek istenen platformlar o kadar karmaşık ve lineer olmayan ki insanların senelerdir üzerinde çalıştığı ve durum denklemleri ile analiz ile de zirveye ulaştırdığı lineer kontrol teknikleri yetersiz kalmaktadır. İnsanlar önceleri bu lineer kontrol tekniklerini kullanarak lineer olmayan sistemleri kontrol etmeyi

denemişler ve bunda da bir nebze olsun başarılı olmuşlardır. Bunu ise lineer olmayan sistemleri çalışma noktası etrafında lineer hale getirerek başarmışlardır. Ama bu yöntem pek çok hassas kontrol işlemi için kullanılabilir değildir. Bundan dolayıdır ki insanoğlu daha esnek ve her koşula uygulanabilir bir kontrol stratejisi peşine düşmüş ve zeki kontrol tekniklerinin doğuşu böylece başlamış olmuştur.

Bu çalışmada yukarıda da bahsedilen zeki kontrol tekniklerinden yapay sinir ağları ile kontrol konusuna değineceğiz.

Yapay sinir ağları kendi doğasından gelen özellikleri sebebiyle bir çeşit adaptif kontrol elemanı olarak görülebilirler. Bu özelliklerinin yanı sıra YSA fonksiyon tanıma ve fonksiyon uyarda da kullanılan bir metottur. YSA nin bu yukarıda bahsedilen bu özelliklerini de kullanarak YSA' nin kontrol de nasıl kullanıldıklarına bir bakalım.

YSA' lar kontrolde özellikle de Robotik uygulamalarda

1-Kinematik

2-Dinamik

3-Güzergah Tanımlama

4-Hareket Kontrolü

gibi uygulamalarda kullanılırlar.

Robot kontrolü ve dolayısıyla YSA ile kontrol ile ilgili ilk çalışma CMAC (Cerebeller Model Articulation Controller) ile [J. Albus] tarafından yapılmıştır. Burada önerilen yapı ise standart YSA yapısından farklı olarak bir "look-up table" tablo şeklinde olup YSA' nin çıkışı bu tablodaki çeşitli hücrelerin değerlerinin toplanmasıyla ortaya çıkar "

Diğer bir çalışma da ters dinamik ve ters kinematik öğretmeyle ilgili olarak yapılan çalışmadır. Bu çalışmayı Psaltis[D.Psaltis and A. Sideris, A], Guez ve Ahmed [A. Guez and Z. Ahmad] yapmışlardır. Bu konulardaki diğer önemli çalışmaları Elsy [R. Elsey,1988], Grossberg ve Kuperstein [S. Grossberg and M.Kuperstein,1989], Fukuda [T. Fukuda et al, 1990] yapmışlardır.

Mitsua Kawato [Kawato M. et al, 1988] "Hierarchical neural ağ model for voluntary movement " adlı makalesinde yeni bir metod uygulamış ve bu metotta sistemde iki adet kontroler kullanıp bunlardan bir tanesini YSA diğerini de PD kontroler olarak seçmiştir.

Buradaki en önemli yenilik de modeldeki ters dinamiğin on-line olarak bir YSA' ya öğretilmesi olup gayet başarılı sonuçlar vermiştir.

Daha önce de bahsedildiği gibi pek çok YSA ile kontrol metodu mevcuttur. Bunları sıralarsak:

- 1-Direkt Ters Denetim (Direct Inverse Modelling Architecture (DIMA))
- 2- Direkt Ters Denetim (Direct Inverse Modelling Architecture (DIMA) Tip II)
- 3-Geri Beslemeli-Hata Öğrenmeli Denetim(Feedbak Error Learning Architecture (FELA))
- 4- Kayan Kipli Denetim (Neuro- Sliding Mode Architecture (NESMA))

3.5.1 Ters Denetim Yöntemi (Direct Inverse Modelling Architecture) (DIMA)

Bu metotlardan ilki olan DIMA da amaç sistemin ters dinamiğinin off-line olarak öğrenilmesi ve bunun sonucunda kontrol için birinci kontrol elemanı olarak kullanılmasıdır. Buradaki YSA' nın birincil kontrol elemanı olarak kullanılmasının sebebi lineer olmayan yapının ters dinamik sayesinde lineer hale getirilmesi ve dolayısıyla PD kontrol elemanı ile kullanılabilmesidir.

İkinci tip DIMA da olaylar aynen cereyan etmekte fakat öğrenme on-line olmaktadır.

Bu metodun en önemli dezavantajları:

- 1-Öğrenmenin off-line olması
- 2-Sistemin hangi bölgelerde öğretilmesi gerektiğinin bilinmemesi
- 3-Öğrenmenin çok fazla zaman alması
- 4- Ortamdaki değişime cevap verememesi

3.5.2 Geri Beslemeli-Hata Öğrenmeli

Diğer bir metot olan FELA on-line bir kontrol elemanı olup olaya değişik bir açıdan yaklaşmıştır. Burada amaç sisteme lineer olmayan etkenleri kazandıran parametreleri bulmaktır.

Bu amaçla tasarlanan FELA gayet iyi sonuçlar vermekte ve YSA ile kontrolde kullanılmaktadır.

3.5.3 Kayan Kipli Kontrol

Yapay sinir ağlarının (YSA) ana özellikleri, örnekleri tanıyabilmeleri, girişleri sınıflandırabilmeleri ve öğrenerek dinamik çevre karşısında kendilerini adapte edebilmeleridir. Fakat akıllı kontrolördeki YSA' larının eşleme özellikleri siyah kutu yaklaşımıdır. Siyah kutunun ne yaptığını anlamak mümkündür, fakat nasıl yaptığını anlamak mümkün değildir. Bulanık lojik (BL) insan bilgi ve tecrübesini kullanmakta ve çıkarım yapabilmektedir. Fakat BL teorisinde öğrenme mekanizması bulunmamaktadır. Bu dezavantajları aşmak için Bulanık YSA (BYSA) geliştirilmiştir. Bu yapıda YSA parçası öğrenme için, BL bilgilerin temsili ve sisteme aktarımında kullanılmaktadır. Adaptif BYSA sistemi gerçekleştirilen sistem performanslarında artışa sebep olduğundan, sistem tanımlama ve kontrolünde geniş uygulama alanı bulmuştur.

Değişken yapıli kontrolün (DYK), YSA ve BL' in kontrol uygulamalarında birbirlerini tamamlayıcı özellikleri vardır. Bu yüzden bu tekniklerin tümleştirilerek lineer olmayan sistemlerin kontrolünde kullanılmasında kazanımlar olacağı açıktır. Böylece ortaya çıkan kontrolör lineer olmayan bir kontrolör olacaktır ve klasik DYK'de karşılaşılan problemlerin çözümünü sağlayabilecektir. BYSA kullanılan kontrol sistemlerinin kullanım alanları basit sistemlerden (sadece birkaç değişkenin olduğu ve deneme yanılmaya izin verilen çamaşır makinesi, klima ve fırın gibi benzeri cihazlar) karmaşık sistemlere genişledikçe (çok fazla değişkenin bulunduğu ve başarısız deneme yanılmaya müsaade edilemeyen kimyasal süreçler, güç santralleri, robotik sistemler ve uçaklar gibi benzeri sistemler), bu kontrol sistemlerin sistematik tasarımlarına acil ihtiyaç vardır. Ayrıca bu kontrol sistemlerinin aşağıdaki özelliklere sahip olması istenmektedir,

1. İyi bir izleme performansı sağlarken çatırtıyı azaltabilmelidir ve hatta yok etmelidir,
2. Kurallar tablosunu kendisi çalışma esnasında oluşturabilmelidir, böylece sistem boş kural tablosu ile çalışmaya başlayabilmelidir,
3. Kontrol ve öğrenmenin aynı anda birlikte çalışma esnasında yapıldığı on-line kontrolör olmalıdır,
4. Kontrol sistemi gürbüz ve hata giderebilir yapıda olmalıdır,

5. Kontrol sistemi kararlı olmalıdır,
6. Kontrol sistemi yapısal bulanık kuramını kullanmalı ve böylece sistemin anlaşılması ve yorumu kolay olmalıdır,
7. Kontrol sistemin içerdiği üyelik fonksiyonları ve kural tablolarının çalışma esnasında değişimleri kolayca gözlenebilmelidir,
8. Kullanıcıya kontrol stratejilerini kontrolöre eklemeye izin vermelidir,
9. Çalışma esnasında kontrol sistemi bütün parametrelerini kendisi ayarlamalı ve optimum performansı vermelidir.

3.6 Bulanık Mantık ve Bulanık Kontrolörlerin Temel Özellikleri

Bulanık kontrol, bulanık küme teorisinin kontrol sistemlerine uygulanmasında son yıllarda araştırmacılar için oldukça etkin ve yeni bir alan olmuştur[J. Albus][Hung J. Y]. Bulanık kontrolün endüstriyel proseslerin kontrolünden başka, artık el tipi küçük kameralardan, çamaşır makinelerine kadar güncel uygulama alanları bularak evlerimize kadar girmesi de bu önemi göstermek bakımından önemlidir. Gerçek zamandaki endüstriyel proseslerde klasik kontrol yöntemleri ile kontrol her zaman verimli olmayabilir. Çünkü mevcut duyargalardan yeterince uygun ölçme bilgileri alınamayabilir. Bu da kontrolör verimini etkileyen faktörlerden birisidir [Lee, C.C][Zhu, H.A. and Teo, C.L.][Su, C.Y. and Stepanenko][Çela, A. and Hamam, Y.].

Bulanık mantık esasına dayalı bulanık kontrol, insanın düşünce yapısına uygun ve halen kullanılan mantıksal sistemlerden daha çok, doğal dile yakın olan sistemlerdir. Bulanık mantık esasına dayalı kontrolörler (Fuzzy Logic Controller, (FLC)) otomatik kontrol stratejisi içinde uzman bilgisine dayalı olayları, dilsel kontrol stratejisine dönüştürmeyi sağlar. Burada, **BK (Bulanık Kontrolör)**' lerin kurulmasındaki temel yöntemleri, **bulandırıcı** (fuzzifier) ve **durulayıcı** (defuzzifier) yapılarını, bulanık kontrol kurallarının ve veri tabanının oluşturulması ve bulanık olayların mekanizması kısaca verilecektir.

Literatürde bulanık kontrolle ilgili çalışmalardan bazıları, suyun kalite kontrolü, tren ve metro işletmesi , tren’le yük doldurma/boşaltma, asansör kontrolü, nükleer reaktör kontrolü, enerji iletimi, bulanık bellekler, bulanık bilgisayarlar gibi alanlar sayılabilir [Lee, C.C.][Lee, C.C.]. Bu ve benzeri birçok örnekten de görüldüğü gibi bulanık kontrol, hızlı bir gelişim sürecindedir. Bu gelişmeler arz / talep ilişkisini de etkileyerek, bulanık mikrokontrolörlerin geliştirilmesini gündeme getirmiş ve yirmiye yakın firma tarafından bulanık mikrokontrolörler ve geliştirme sistemleri pazarda yer almaya başlamıştır. Fiyatlarının da klasik mikrokontrolörlerden çok fazla olmayışı bunlara olan talebi artırmaktadır. Bu mikrokontrolörlerin mimarilerinde bulanık kurallar yonga üzerinde bulunduğu için, dilsel kontrol kurallarının geliştirilmesi ve işlenmesi diğerlerine göre çok daha basit, hızlı ve kolay olmaktadır. Bu özellikleri, gerçek zamanda veri işlemeyi daha uygun yapmaktadır.

Ancak ne yazık ki literatürde henüz, bulanık kontrolör tasarımı için sistematik bir prosedür yoktur. İki ayrı araştırma aynı konuyu içerse bile, tasarımcının bakış açısına bağlı olarak farklılıklar gösterebilmektedir. Bu da konu ile ilgili uzman bilgisi ve uzman sisteminin etkinliğine bağlı olarak değişmektedir.

3.6.1 Bulanık Kümeler ve Bulanık Mantık

Bu ayrıtta bulanık kümeler ve bulanık mantığın temel yapısı konuyu anlamak bakımından kısaca verilecektir.

Genellikle $\{u\}$ ile gösterilen ayrık veya sürekli olabilen nesnelerin oluşturduğu bir küme U olsun.

$$U=\{u\} \quad (3.16)$$

u ’lar U evrensel kümesinin jenerik elemanı olarak adlandırılır.

Tanım: Bulanık Küme (fuzzy set)

Bahsedilen U evrenindeki bir bulanık küme F , $[0,1]$ aralığında değer alabilen μ_F üyelik fonksiyonu ile karakterize edilir. Yani;

$$\mu_F : U \rightarrow [0,1] \quad (3.17)$$

Bir bulanık küme, yalnızca $\{0,1\}$ gibi iki değer alabilen bütün üyelik fonksiyonunun sıradan bir kümesinin genelleştirilmiş olarak görülebilir.

Böylece U daki bir F bulanık kümesi, bir u eleman çiftinin sıradan bir kümesi olarak gösterilebilir ve bunun üyelik fonksiyonunun derecesi;

$$F = \{ (u, \mu_F(u) \mid u \in U \} \quad (3.18)$$

dir.

U sürekli olduğu zaman, bulanık küme F şöyle yazılabilir.

$$F = \int_u (\mu_F(u) / u) du \quad (3.19)$$

U ayrık olduğu zaman, bulanık küme şöyle olur;

$$F = \sum_{i=1}^n \mu_F(u_i) / u_i \quad (3.20)$$

Tanım: Dilsel Değişken

Dilsel değişken, değişkenin ismi x olan ve x deki $(x, T(x), U, G, M)$ gibi bir fonksiyon ile karakterize edilir. Burada,

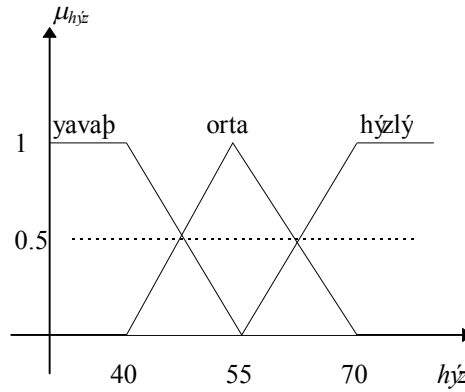
$T(x)$: x kümesinin bir terimidir. Bu U 'da tanımlı olan her bir bulanık sayı değeri ile x 'in dilsel değerlerinin isimlerinin bir kümesidir.

G : x 'in değerlerinin isimlerinin üretilmesi için bir kuraldır.

M : x 'in her değer ve anlamı ile uygun olması için bir kuraldır.

Örnek: “hız” dilsel bir değişken olarak tanımlanabilir. Bunun küme terimi $T(hız)$; $T(hız) = \{yavaş, orta, hızlı, çok yavaş, biraz hızlı, \dots\}$

Burada her $T(hız)$ terimi, bahsedilen $U=[0,100]$ de bir bulanık küme ile karakterize edilebilir. Örneğin, “yavaş” olarak hızın 40 km/saat dan daha az olduğunu yorumlayabiliriz. “orta” olarak hızın 55 km/saat a daha yakın olduğunu, “hızlı” olarak da 70 km/saat olduğunu düşünebiliriz. Bu terimler bulanık kümeler olarak karakterize edilebilir ve tüm üyelik fonksiyonları Şekil 3.8. 'de gösterilmiştir.



Şekil 3.8. Bulanık hızların diyagram olarak gösterilişi.

Burada “hız” üç terimden oluşan dilsel değişkendir.

3.6.2. Bulanık Mantık ve Mantıksal Yaklaşıklık

Bulanık mantık ile matematikteki mantıktaki önermelerden çıkarımların farkını daha iyi anlamak için şu önermeleri ve çıkarımları verebiliriz;

önerme_1: kuşlar uçar

önerme_2: serçe bir kuştur

sonuç : serçe uçar

Bulanık mantıkta ise;

önerme: hava çok soğuk ise sıkı giyin

çıkarım: hava çok soğuk değil ise biraz sıkı giyinebilirsin

Bulanık çıkarım 1973’de L. Zadeh tarafından önerilen mantıksal yaklaşıma dayanılarak elde edilen çıkarım kurallarına dayanır.

Burada A , A' , B , B' bulanık kümeleri, klasik mantıktaki kesin kümeler yerine dilsel değişkenlere karşı gelir. $A'=A$ ve $B'=B$ olduğunda modus ponens’e indirgenir. Bu da özellikle Bulanık Mantık Kontrolünde kullanılır. Halbuki $B'=değilB$ ve $A'=değilA$ olduğu zaman modus tolens’e indirgenir. Bu da geriye amaç süren çıkarımlar için oldukça uygun bir yapıdır [Lee, C.C][Lee, C.C].

3.6.3. Bulanık Mantık Kontrolörlerinin Ana Fikri

Şekil 3.9 'da bu yapıya ait temel bloklar, *bulandırıcı* ve *durulayıcı* bloklarıdır.

1. Bulandırıcı Arabirimi

Bu arabirim bloğu şu fonksiyonları sağlar;

- Giriş değişkenlerinin ölçüsünü belirlemek,
- Sözü edilen uzaya karşı gelen giriş değişkenlerinin oranını dönüştüren performans ölçeklemesi yapmak,
- Bulandırıcının fonksiyonları bulanık kümelerin etiketleri olarak görülebilecek uygun dilsel değişkenlere, giriş verilerini dönüştürmeyi sağlamak.

2. Bilgi Tabanı Bloğu

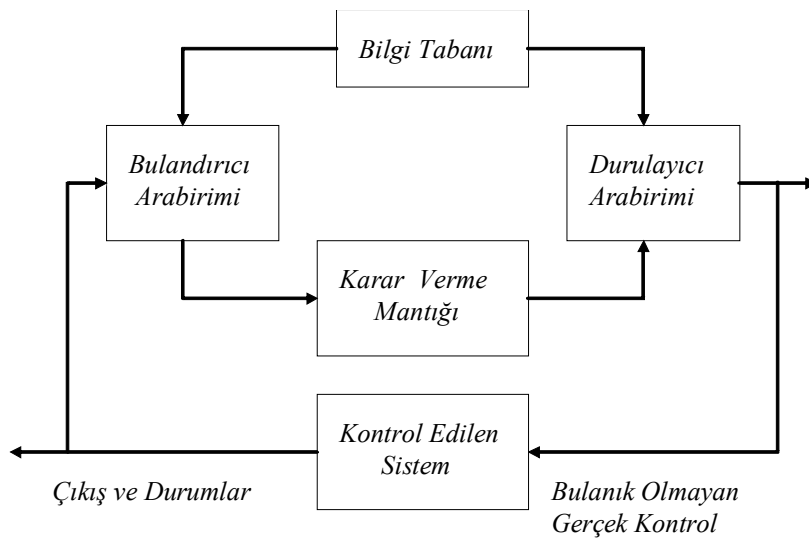
Burada kontrol amacına uygun olarak dilsel(bulanık) kontrol kural tabanı ve bir veri tabanı vardır.

a. Veri tabanı

BMK'da bulanık veri manipülasyonlarını ve tanımlanmış dilsel kontrol kurallarını kullanarak gerekli tanımlamaları sağlar.

b. Kural tabanı

Kontrol amaçlarını karakterize eder. Kontrol topluluğu, dilsel kontrol kurallarının bir kümesi aracılığı ile bilinir.



Şekil 3.9. Bulanık Mantık Kontrolörünün Genel Yapısı

3. Karar Verme Mantık Bloğu

Bu blok BMK'nın çekirdeğidir. Bulanık mantıktaki çıkarım kurallarını ve bulanık gerçeklemeyi kullanarak kontrol etkisini elde etmek ve ayrıca bulanık mantıktaki insanın karar vermesine benzer simülasyonları yapma özelliğine sahip olan bir bloktur.

4. Durulayıcı Arabirimi

Bu arabirim şu görevleri yerine getirir;

a. Ölçekleme

Söz konusu uzaya karşı gelen çıkış değişkenlerinin değer aralıklarını dönüştürerek ölçekleme yapmak,

b. Durulama

Bulanık kontrol etkisinden çıkarılmış olan ve bulanık olmayan kontrol etkisini üretmek.

3.6.4 Bulanık Koşullar ve Bulanık Kontrol Kuralları

BMK'da bulanık sistemin dinamik davranışı uzman bilgisi tabanına dayalı, dilsel tanımlama kurallarının bir kümesi ile karakterize edilir.

Uzman bilgi sistemine dayalı olarak şu form gerçekleşir;

if (sağlanan koşullar kümesi) **then** (çıkarım yapılabilecek sonuçların bir kümesi)

Bulanık mantıkla desteklenen **if-then** koşul sonuçlarına genelde “*bulanık koşul deyimleri*” denir. Burada bu bir bulanık koşul deyimini olarak değerlendirilir. Temelde *bulanık kontrol kuralları*, *kontrol politikasına* ve *domen bilgisine* sahip olmak için uygun bir yoldur. Burada domen bilgisinden kasıt, ölçülen durum bilgileri veya çıkış bilgileri olabilir.

Örnek:

2 girişli-1çıkışlı bir (ÇGTÇ) sisteminin kontrol kuralları şu formda olabilir;

kural_1: Eğer x, A_1 ise ve y, B_1 ise z, C_1 dir.

kural_2: Eğer x, A_2 ise ve y, B_2 ise z, C_2 dir.

kural_n: Eğer x, A_n ise ve y, B_n ise z, C_n dir.

Burada x, y, z dilsel değişkenlerdir. İlk ikisi sistemin *durumlarını* temsil ederken sonuncusu *kontrol* değişkenini gösterir.

A_i, B_i, C_i 'ler sözkonusu U, V, W uzayında x, y, z değişkenlerinin dilsel değerleridir. Buradaki her cümle kural kümesinde kuralları birbirine bağlar ve buna kural tabanı denir.

Bir bulanık kontrol kuralı örneğin “ **if** (x, A_i ise **AND** y, B_i ise) **then** (z, C_i) **dir.**” gibi bir çıkarımla gerçekleştirilebilir.

3.6.5 Bulandırma ve Durulama Operatörü

Bir bulandırma operatörü, bulanık kümelerdeki bulanık olmayan veri dönüşümünü sağlar,

$$x = \text{bulandırıcı} (x_o) \quad (3.21)$$

Burada x_o bir prosesden alınan kesin (crisp) giriş değeridir. x bulanık bir kümedir. *bulandırıcı* (fuzzifier) bulandırma operatörünü gösterir.

Çıkarım işleminin sonucu, bir bulanık kümedir ve kontrol etkisinin mümkün olan bir dağılımını gösterir. Gerçek zaman kontrolde bulanık olmayan kontrol etkisi genellikle gereklidir. Bunun için, bulanık kontrol algoritmasından çıkarılmış olan kontrol etkisi durulanmalıdır. Yani;

$$z_o = \text{durulayıcı} (z) \quad (3.22)$$

Burada z_o bulanık olmayan, duru kontrol çıkışıdır. Durulayıcı (defuzzifier) durulama operatörüdür.

3.6.6. Bulanık Mantık Kontrolörünün Tasarım Parametreleri

Bir Bulanık Mantık Kontrolörü için tasarım parametrelerinin genel prensibi şöyle verilebilir;

1. Bulandırma yöntemi ve bulandırma operatörünün yorumlanması
2. Veri tabanı
 - a. Söz konusu uzayın ayrıklaştırılması/normalleştirilmesi,
 - b. Giriş-Çıkış uzaylarının bulanık parçalara ayrıştırılması,

- c. Seçimlerin tamamlılığı yada tamamlılık olması,
- d. Bulanık bir ön kümeden üyelik fonksiyonlarının seçimi

3. Kural tabanı

- a. Bulanık kontrol kurallarının giriş (durumlar) ve çıkış (kontrol) değişkenlerinin seçimi
- b. Bulanık kontrol kurallarının kaynakları ve üretilmesi, elde edilmesi
- c. Bulanık kontrol kural tiplerinin belirlenmesi
- d. Bulanık kontrol kurallarının bütünlülük, uyumluluk, ve etkileşimliliği.

4. Karar Verme Mantığı

- a. Bulanık çıkarımın tasarımı,
- b. “VE” cümle bağıllığının yorumlanması,
- c. “AYRICA” cümle bağıllığının yorumlanması,
- d. Bileşim operatörünün tanımı,
- e. Çıkarım mekanizması,

5. Durulama stratejisi ve durulama operatörünün yorumlanması.

3.6.7 Üyelik Fonksiyonları

Üyelik fonksiyonlarının belirlenmesinde iki yöntem söz konusudur.

a. Sayısal tanımlama

Bu durumda bir bulanık kümenin üyelik fonksiyonunun değeri, boyutu ayrıklaştırma derecesine bağlı sayılardan oluşan bir vektör olarak temsil edilir.

$$\mu_f(u) = \sum_{i=1}^n \frac{a_i}{u_i} \quad (3.23)$$

Örneğin burada, $a=(0.34, 0.71, 1.00, 0.71, 0.34)$ olabilir.

b. Fonksiyonel tanımlama

Burada üyelik fonksiyonları bir fonksiyonla tanımlanır. Çan eğrisi, üçgen, trapezoidal gibi fonksiyonlar olabilir. Bu fonksiyonlar BMK’ da

kullanılabilir. Çünkü bu fonksiyonların normalizasyonu ve değişimlere uyarlanabilme aritmetiği daha kolaydır. Örneğin;

$$\mu_f(x) = \exp\left[\frac{-(x - u_f)^2}{2\sigma_f^2}\right] \quad (3.24)$$

Sonuç olarak her iki tanımla da üyelik fonksiyonları hesaplanabilmekte ve karar verme mantığına uygun olarak üyelik dereceleri seçilebilmektedir.

3.6.8 Kural Tabanı

Bulanık bir sistem, uzman bilgisine dayanan dilsel deyimlerin oluşturduğu bir küme ile karakterize edilebilir. Uzman bilgisi (expert knowledge), bulanık mantıkta bulanık koşul deyimleri ile kolayca gerçekleştirilebilen genellikle “**if-then**” kuralları formundadır. Bulanık kontrol kuralları, BMK’ nın kural kümesi veya kural tabanlı bulanık koşul deyimleri olarak tanımlanabilir.

Yapıları gereği iki tip bulanık kontrol kuralı vardır.

1. Durum değerlendiren bulanık kontrol kuralları
2. Amaç değerlendiren bulanık kontrol kuralları

4.TASARLANAN YÖNTEMLER VE TEST ORTAMININ OLUŞTURULMASI

Bu bölümde literatürdeki yapay zeka tekniklerinde bazıları analiz edilecektir.. Bunları sıralarsak:

- 1-Direct Ters Kontrol (Direct inverse modeling architecture with radial basis functions (DIMA)),
- 2-Geri Beslemeli-Hata Öğrenmeli Kontrol (Feedback error learning architecture(FELA)),
- 3-Kayan Kipli Kontrol (Neuro-sliding mode control),
- 4-Gaussian Yapay Bulanık Kayan Kipli Kontrol (Gaussian neuro fuzzy variable structure control)
- 5-Adaptive Yapay Bulanık Çıkarım Sistemi (Adaptive neuro fuzzy inference system(ANFIS)).
- 6- Kayan Kipli Denetim (Neuro- Sliding Mode Architecture (NESMA))

Bu metotlardan ilki olan DIMA da amaç sistemin ters dinamiğinin off-line olarak öğrenilmesi ve bunun sonucunda kontrol için birinci kontrol elemanı olarak kullanılmasıdır. Buradaki YSA' nın birincil kontrol elemanı olarak kullanılmasının sebebi lineer olmayan yapının ters dinamik sayesinde lineer hale getirilmesi ve dolayısıyla PD kontrol elemanı ile kullanılabilmesidir.

İkinci tip DIMA da olaylar aynen cereyan etmekte fakat öğrenme on-line olmaktadır.

Bu metodun en önemli dezavantajları:

- 1-Öğrenmenin off-line olması
- 2-Sistemin hangi bölgelerde öğretilmesi gerektiğinin bilinmemesi
- 3-Öğrenmenin çok fazla zaman alması
- 4- Ortamdaki değişime cevap verememesi

Diğer bir metot olan FELA on-line bir kontrol elemanı olup olaya değişik bir açıdan yaklaşmıştır. Burada amaç sisteme lineer olmayan etkenleri kazandıran parametreleri bulmaktır.

Bu amaçla tasarlanan FELA gayet iyi sonuçlar vermekte ve YSA ile kontrolde kullanılmaktadır.

Yapay sinir ağlarının (YSA) ana özellikleri, örnekleri tanıyabilmeleri, girişleri sınıflandırabilmeleri ve öğrenerek dinamik çevre karşısında kendilerini adapte edebilmeleridir. Fakat akıllı kontrolördeki YSA' larının eşleme özellikleri siyah kutu yaklaşımıdır. Siyah kutunun ne yaptığını anlamak mümkündür, fakat nasıl yaptığını anlamak mümkün değildir. Bulanık lojik (BL) insan bilgi ve tecrübesini kullanmakta ve çıkarım yapabilmektedir. Fakat BL teorisinde öğrenme mekanizması bulunmamaktadır. Bu dezavantajları aşmak için Bulanık YSA (BYSA) geliştirilmiştir. Bu yapıda YSA parçası öğrenme için, BL bilgilerin temsili ve sisteme aktarımında kullanılmaktadır. Adaptif BYSA sistemi gerçekleştirilen sistem performanslarında artışa sebep olduğundan, sistem tanımlama ve kontrolünde geniş uygulama alanı bulmuştur.

Değişken yapılı kontrolün (DYK), YSA ve BL' in kontrol uygulamalarında birbirlerini tamamlayıcı özellikleri vardır. Bu yüzden bu tekniklerin tümleştirilerek lineer olmayan sistemlerin kontrolünde kullanılmasında kazanımlar olacağı açıktır. Böylece ortaya çıkan kontrolör lineer olmayan bir kontrolör olacaktır ve klasik DYK' de karşılaşılan problemlerin çözümünü sağlayabilecektir. BYSA kullanılan kontrol sistemlerinin kullanım alanları basit sistemlerden (sadece birkaç değişkenin olduğu ve deneme yanılmaya izin verilen çamaşır makinesi, klima ve fırın gibi benzeri cihazlar) karmaşık sistemlere genişledikçe (çok fazla değişkenin bulunduğu ve başarısız deneme yanılmaya müade edilemeyen kimyasal süreçler, güç santralleri, robotik sistemler ve uçaklar gibi benzeri sistemler), bu kontrol sistemlerin sistematik tasarımlarına acil ihtiyaç vardır. Ayrıca bu kontrol sistemlerinin aşağıdaki özelliklere sahip olması istenmektedir,

1. İyi bir izleme performansı sağlarken çatırtıyı azaltabilmelidir ve hatta yok etmelidir,
2. Kurallar tablosunu kendisi çalışma esnasında oluşturabilmelidir, böylece sistem boş kural tablosu ile çalışmaya başlayabilmelidir,
3. Kontrol ve öğrenmenin aynı anda birlikte çalışma esnasında yapıldığı on-line kontrolör olmalıdır,
4. Kontrol sistemi gürbüz ve hata giderebilir yapıda olmalıdır,

5. Kontrol sistemi kararlı olmalıdır,
6. Kontrol sistemi yapısal bulanık kuramını kullanmalı ve böylece sistemin anlaşılması ve yorumu kolay olmalıdır,
7. Kontrol sistemin içerdiği üyelik fonksiyonları ve kural tablolarının çalışma esnasında değişimleri kolayca gözlenebilmelidir,
8. Kullanıcıya kontrol stratejilerini kontrolöre eklemeye izin vermelidir,
9. Çalışma esnasında kontrol sistemi bütün parametrelerini kendisi ayarlamalı ve optimum performansı vermelidir.

Literatürde ilk beş ihtiyacı karşılamak ve klasik DYK' ün dezavantajlarını gidermek için Gauss fonksiyonu tabanlı BYSA' lı DYK kontrol sistemi geliştirilmesi üzerine çalışmalar yapılmıştır. Bu kontrol sistemi BL ile YSA arasında yumuşak geçiş olarak tanımlanan radyal tabanlı YSA içermektedir.

Bu çalışmada test ortamı için ikinci bölümde tasarlanan SCARA tipi robot kullanılacaktır. Çalışmada özellikle denklem 4.1, denklem 4.2 ve denklem 4.3

Hata hesaplama denklemi

$$\left| error_i = \sqrt{\sum_{j=1} (q_i(j) - q_i^d(j))^2} \right| \quad (4.1)$$

Max hata tanımı denklem 5.70 de gösterilmiştir

$$\max error_i = \max_{1 \leq j \leq k} |q_i(j) - q_i^d(j)| \quad (4.2)$$

Ve kartezyen koordinatlardaki hata tanımı ki bizim elde etmek istediğimiz bu denklemi minimize etmek, denklem 5.71 de gösterilmiştir.

$$\left| error_{taskspace} \right| = \sqrt{\sum_j \left(\left\| \begin{pmatrix} x(j) - x^d(j) \\ y(j) - y^d(j) \end{pmatrix} \right\|^2 \right)} \quad (4.3)$$

Temel karşılaştırma parametresi olarak kullanılacaktır. Ayrıca bu parametreler ile birlikte öğrenme zamanı, sistemle ilgili önceden bilgi gerektrip gerektirmemesi, online veya offline öğrenme metodlarını kullanması gibi kıstaslar da göz önüne alınacaktır.

Sistem performansı (iki serbestlik dereceli robot) X,Y kartezyen koordinatlarda bir dikdörtgen oluşturacak olan yörüngeyi takip performansı ile ölçülecektir. Sistem performansı analiz edilirken sadece takip performansı değil aynı zamanda kontrol sistemlerinin tasarlanması için gerekli ön bilgi, öğrenme zamanı gibi parametreleri de ölçülecektir.

Karşılaştırmalar yapılırken sadece yukarıdaki parametreler değil aynı zamanda

1-(Hesap Yönetim)Computation

2-(Tasarımdan önceki gerekli bilgi) Necessary Knowledge

3-(Yakınsama zamanı) Convergence Time

4-Öğrenme Algoritması

gibi parametreler de göz önüne alınıp bunlar da karşılaştırma tablolarında gösterilecektir.

5. ZEKİ KONTROLÖR TASARIMI ve PERFORMANS ANALİZLERİ

5.1. Giriş

Bu bölümde ani yük değişiklikleri ne kararlı olan zeki kontrol sistemlerinden en çok kullanılanları ve ev en son geliştirilenleri incelenecektir. Bu incelemeler zeki sistemlerin benzetim ortamında geliştirilecek olan performans kıstasları karşısındaki davranış analizleri ve ani yük değişimi analizde kullanılabilecek en iyi karşılaştırma kriterlerinin tespiti gibi tezin esas amacına hizmet eden önemli çalışmalardır. Bu bölümde kullanılacak olan zeki kontrol sistemleri literatürde defalarca karşılaştırılabilir [E. Akbaş and H. Palaz][E. Akbaş and E. Murat Esin] Bu tezde istenilen amaca ulaştırabilecek kriterler tespit edilip kullanılacaktır.

Bu bölümde literatürde en çok kullanılan zeki kontrol algoritmaları yeniden tasarlanacak ve bu algoritmaların testi ve analizi için gerekli benzetim düzenekleri oluşturulacaktır.

Tasarlanacak olan algoritmalar:

- 1-Direct Ters Kontrol (Direct inverse modeling architecture with radial basis functions (DIMA)),
- 2-Geri Beslemeli-Hata Öğrenmeli Kontrol (Feedback error learning architecture(FELA)),
- 3-Kayan Kipli Kontrol (Neuro-sliding mode control),
- 4-Gaussian Yapay Bulanık Kayan Kipli Kontrol (Gaussian neuro fuzzy variable structure control)
- 5-Adaptive Yapay Bulanık Çıkarım Sistemi (Adaptive neuro fuzzy inference system(ANFIS)).

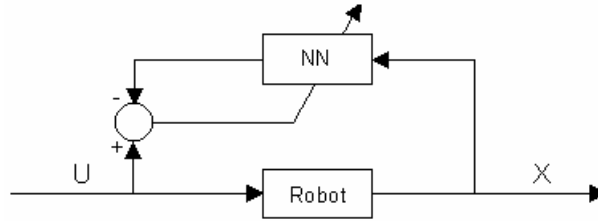
Bu metodlardan 1,2 ve 3 2. bölümde açıklanmıştır. Hatırlatmak için tekrar edersek;

5.2 Zeki Sistemler

Direct Ters Kontrol

DIMA da amaç sistemin ters dinamiğinin daha önceden (off-line) olarak öğrenilmesi ve bunun sonucunda kontrol için birinci kontrol elemanı olarak kullanılmasıdır. Buradaki YSA' nın birincil kontrol elemanı olarak kullanılmasının

sebebi lineer olmayan yapının ters dinamik sayesinde lineer hale getirilmesi ve dolayısıyla PD kontrol elemanı ile kullanılabilmesidir.



Şekil 5.1 DIMA

$$x(k+1) = F[x(k), u(k)] \quad (5.1)$$

$$u(k) = G[x(k+1), x(k)] \quad (5.2)$$

Bu metotta robot çıkışı denklem (5.1) deki gibi hesaplanır ayrıca uygulanacak olan kuvvet de (5.2) denklemini ile hesaplanır.

Bu metodun en önemli dezavantajları:

- 1-Öğrenmenin off-line olması
- 2-Sistemin hangi bölgelerde öğretilmesi gerektiğinin bilinmemesi
- 3-Öğrenmenin çok fazla zaman alması
- 4- Ortamdaki değişime cevap verememesi

Geri Beslemeli – Hata Öğrenmeli Kontrol

Diğer bir metodu olan FELA on-line bir kontrol elemanı olup olaya değişik bir açıdan yaklaşmıştır. Burada amaç sisteme lineer olmayan etkenleri kazandıran parametreleri bulmaktır.

Hesap kolaylığı için $y(t)$ nöron çıkışı ortalama membrane potansiyeline eşitlenir. Ve vektör notasyonunda $y^d=x$ için ,

$$x(t) = [x_1(t), x_2(t), x_3(t), \dots, x_n(t)] \quad (5.3)$$

$$W = [w_1, w_2, w_3, \dots, w_n]^T \quad (5.4)$$

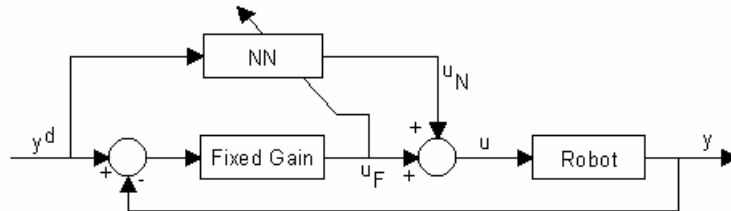
$$y = W^T X(t) \quad (5.5)$$

$$s(t) = z(t) - y(t) \quad (5.6)$$

Burada $z(t)$ bilinmeyen lineer olmayan sistem çıkışı ve $y(t)$ de yapay zeka sisteminin çıkışıdır. Burada önemli olan öğrenme metodudur ki o da (5.7) denklemi ile ifade edilir.

$$\tau dW(t)/dt = X(t)s(t) \quad (5.7)$$

Burada τ öğrenme katsayısıdır.



ŞEKİL 5.2. FELA

Değişken yapıli kontrolün (DYK), YSA ve Bulanık Mantık (BL)’in kontrol uygulamalarında birbirlerini tamamlayıcı özellikleri vardır. Bu yüzden bu tekniklerin tümleştirilerek lineer olmayan sistemlerin kontrolünde kullanılmasında kazanımlar olacağı açıktır. Böylece ortaya çıkan kontrolör lineer olmayan bir kontrolör olacaktır ve klasik DYK’de karşılaşılan problemlerin çözümünü sağlayabilecektir. BYSA kullanılan kontrol sistemlerinin kullanım alanları basit sistemlerden (sadece birkaç değişkenin olduğu ve deneme yanılmaya izin verilen çamaşır makinesi, klima ve fırın gibi benzeri cihazlar) karmaşık sistemlere genişledikçe (çok fazla değişkenin bulunduğu ve başarısız deneme yanılmaya müsaade edilemeyen kimyasal süreçler, güç santralleri, robotik sistemler ve uçaklar gibi benzeri sistemler), bu kontrol sistemlerin sistematik tasarımlarına acil ihtiyaç vardır. Ayrıca bu kontrol sistemlerinin aşağıdaki özelliklere sahip olması istenmektedir,

İyi bir izleme performansı sağlarken çatırtıyı azaltabilmelidir ve hatta yok etmelidir, Kurallar tablosunu kendisi çalışma esnasında oluşturabilmelidir, böylece sistem boş kural tablosu ile çalışmaya başlayabilmelidir,

Kontrol ve öğrenmenin aynı anda birlikte çalışma esnasında yapıldığı on-line kontrolör olmalıdır,

Kontrol sistemi gürbüz ve hata giderebilir yapıda olmalıdır,

Kontrol sistemi kararlı olmalıdır,

Kontrol sistemi yapısal bulanık kuramını kullanmalı ve böylece sistemin anlaşılması ve yorumu kolay olmalıdır,

Kontrol sistemin içerdği üyelik fonksiyonları ve kural tablolarının çalışma esnasında değişimleri kolayca gözlenebilmelidir,

Kullanıcıya kontrol stratejilerini kontrolöre eklemeye izin vermelidir,

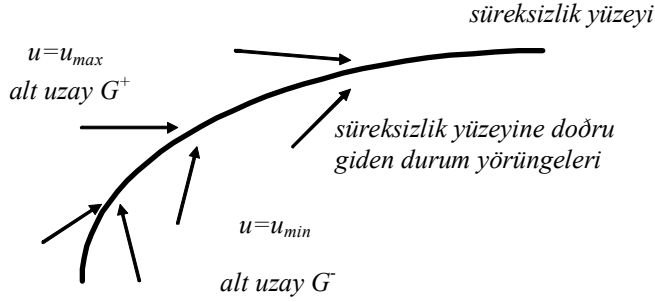
Çalışma esnasında kontrol sistemi bütün parametrelerini kendisi ayarlamalı ve optimum performansı vermelidir.

5.3 Yapay Kayan Kipli Kontrol (Neuro Sliding Mode Control)

Değişken yapıli kontrolün (DYK) ile birlikte kayan kipli kontrol (KPK) ilk olarak 1950lerin başında önerildi ve yetmişlerden sonra da çok rağbet görüp pek çok alanda uygulandı [11]. Bu metodun kullanılmasının ana sebebi KPK özellikleridir. Bu özellikler lineer olmayan sistemlerin kontrolündeki başarısı, MIMO sistemler uygulanabilir olması ve discrete time sistemler için uygulanabilir olmasıdır [2]. Bunun yanı sıra KPK sistemlerin en iyi özelliği gürbüz olmasıdır. Daha açık bir ifade ile KPK sistemleri parametre veya dışsal karıştırıcılara karşı duyarsızdır [BRADY, M. et al.].

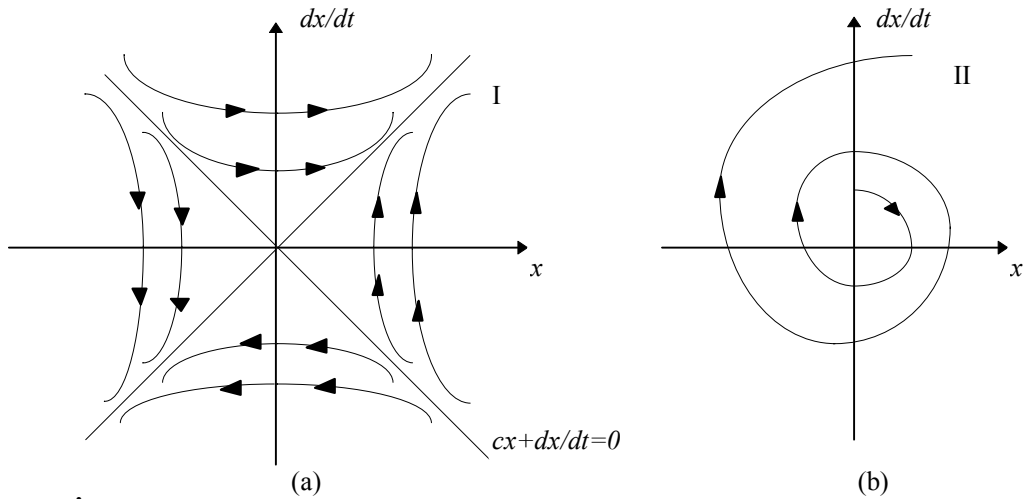
KPK sistemlerin temel dezavantajları chattering ve equivalent control hesabının çok zor olmasıdır. Bunun hesabı için kontrol edilecek olan sistemin gerçek dinamik model denklemlerine ihtiyaç vardır. Literatürde bunu bertaraf etmek için çeşitli yöntemler önerilmiştir. Bunlardan biri sign fonksiyonu yerine saturation fonksiyonu kullanılması [Liu, T. and Liu, C]. Ayrıca ortalama filtre (averaging filter) de önerilen yöntemlerden birisidir [Sen, P.C]

Kayan kip, Değişken Yapılı Sistemler (DYS) 'in özel bir biçimidir. DYS tabanlı sistemlerde sistemin davranışı, süreksiz yüzeylerde, durum yörüngelerinde gösterilir. Hiper düzleme karşı gelen sistemin durumları olarak, kontrol girişi, bir u_{max} değerinden, bir u_{min} değerine kadar anahtarlanır. Hiper düzlemler, anahtarlama veya kayma yüzeyleri (sliding surface) olarak bilinmektedir [Utkin, V].



ŞEKİL 5.3. Süreksizlik yüzeyi ile durum uzayının bölünmesi

Kayma kipinde, durumların hareketi, tüm durum uzayının belli bir alt uzayında kalacak şekilde sınırlandırılır. Bu alt uzay içindeki sistem davranışı, Utkin tarafından eşdeğer bir sistem olarak adlandırılmıştır [Utkin, V.I][Habibi, S.R. and Richards, R.J]. Değişken yapıli sistemlerin teorisinin temel mantığını göstermek için, iki kararsız yapı göz önüne alalım.



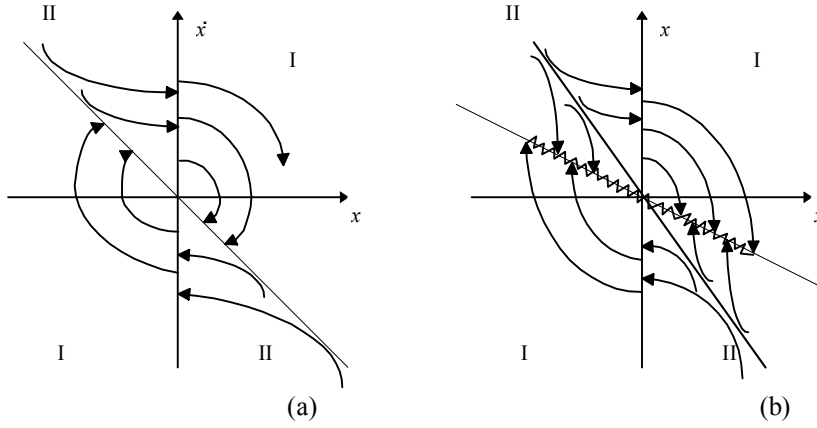
ŞEKİL 5.4. İki kararsız yapı

Bir sisteminin durumları, uygun bir anahtarlama kuralı ile, süreksizlik yüzeylerinde sistemin kararsız bir yapısından, öteki kararsız yapıya geçirilebiliyorsa, kararsız iki yapıdan oluşan kararsız bir sistemden, kararlı bir sistem elde edilebilir.

DYS 'de bu yeni oluşan sistemlerin özellikleri, farklı yapıların yörünge parçalarından, istenen yörüngeler birleştirilerek elde edilmiş olur. Bu yörüngeler yeni bir tip hareketi tanımlar ki buna kayma kipi (sliding mode) denir [Utkin, V].

Şimdi yukarıdaki Şekil 5.3. de verilen kararsız iki sistem, eksenlerinden üst üste koyarak çakıştırılırsa ve eksenlerin ayırdığı bölgenin birinde 1. sistemin, diğer bölgede 2. sistemin yörüngeleri izlenecek şekilde anahtarlama yapılırsa Şekil 5.4. de görülen sonuçur (asimtotik) kararlı yapılar oluşur.

Değişken yapılu bir sistem, sistemi oluşturan yapıların her birinde gözükmeyen yeni özelliklere de sahip olabilir.



Şekil 5.5. DYS

a. İki kararsız yapıdan oluşan kararlı DYS

b. 2. dereceden DYS de kayan kipi oluşumu

5.3.1. Süreksiz Dinamik Sistemlerde Kayma Kipi

Genel bir DYS göz önüne alalım;

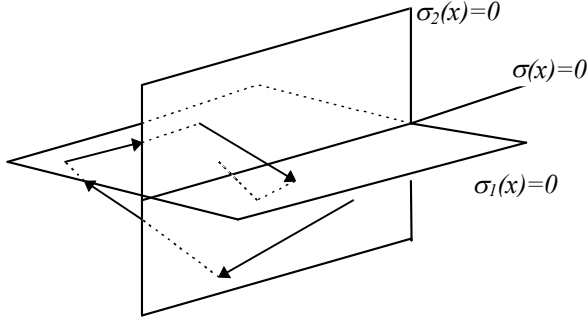
$$\dot{x} = f(x, t, u) \quad (5.8)$$

Burada $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$, $f \in \mathbb{R}^n$ dir. Kontrol vektörünün her elemanı x ve t 'nin sürekli fonksiyonlarının bir kümesinden birisine eşit olabilir. Durum uzayında,

$$u_i = \begin{cases} u_i^+(x, t), & \sigma_i(x) > 0 \\ u_i^-(x, t), & \sigma_i(x) < 0 \end{cases} \quad (5.9)$$

$i=1, \dots, m$ dir. Sürekli u_i^+ ve u_i^- fonksiyonlarının ve $\sigma \in \mathbb{R}^m$ olan $\sigma_i(x)$, vektörünün seçimi DYS sisteminin tasarım problemini oluşturur. (5.8) ve (5.9) sisteminin

durumlarının, süreksizlik yüzeylerinin arakesitine eriştiğini kabul edelim. Ancak belli hallerde, bir durum, bu arakesite (manifolda) kaçınılmaz olarak küçük bir sapma ile yerleşebilir. Bu durumu şu şekilde göstermek uygundur;



ŞEKİL 5.6. Dinamik yüzeylerin arakesitinde kayma kipinin oluşumu

(5.8) ve (5.9) denklemlerinin sağ tarafı süreksiz olduğu için, normalde klasik teoremlerle varlık ve tekliği sağlanamaz. Ayrıca sağ tarafları genelde süreksiz yüzeylerde tanımlanamazlar. Bu matematiksel güçlükler, gerçek sistemlerdeki ideal olmayan durumların ortaya çıkmasına neden olur. Kayma kipini iyi anlayabilmek için, küçük zaman gecikmesi, histeresiz ve diğer tipten ideal olmayan durumlar göz önüne alınmalıdır. Bu ideal olmayan haller, gerçek kayma kipini belirler. Bu da sistemin durumlarının süreksizlik yüzeyinin üzerinde değil de, çok yakınında hareket etmesi demektir. İdeal kayma kiplerinin denklemlerini bulmak için genelde *eşdeğer kontrol* yöntemi olarak adlandırılan yöntem kullanılır. Bu yöntemde (5.8) sisteminin yörüngeleri boyunca $\sigma(x)$ vektörünün zamana göre türevi, sıfıra eşittir ve kontrol için bu sistem çözülür. Eğer varsa bu eşdeğer kontrol orijinal sistemde yerine yazılır. Sonuçta elde edilen denklemler ideal kayma kipinin denklemler olur [Utkin, V].

5.3.2. Robotun DYS Tabanlı Kayan Kipli Kontrolü

5.3.2.1. Eşdeğer Kontrol Yöntemi

Bu bölümde, dinamik modeli daha önceki bölümlerde elde edilen robotunun kayan kipli kontrolü yapılacaktır. Robotun dinamiğindeki ivme terimi çekilerek yazılırsa;

$$\ddot{q} = M(q)^{-1} \left[\tau - C(q, \dot{q})\dot{q} - g(q) \right] \quad (5.10)$$

olur.

Buradan şu dönüşümler yapılırsa,

$$x_1 = q \quad x_2 = \dot{q} \quad (5.11)$$

robotun dinamiği $M\dot{x} + Cx + g = \tau$ biçimine gelir. Robotun durum uzayı matematiksel modeli;

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -M(x_1)^{-1}[C(x_1, x_2)x_2 + g(x_1)] \end{bmatrix} + \begin{bmatrix} 0 \\ M(x_1)^{-1} \end{bmatrix} u \quad (5.12)$$

şeklinde elde edilir. Buradan genel robot dinamiğinin kapalı biçimini;

$$\dot{x} = f(q, \dot{q}, t) + Bu \quad (5.13)$$

şeklinde yazılabilir. Eşdeğer kontrolü bulabilmek için, önce kayma yüzeyini seçmek gerekir. Kayma yüzeyi;

$$S = \{x: \sigma(x, t) = 0\} \quad (5.14)$$

seçilsin. Burada,

$$\sigma = G(x_{ref} - x) = Ge \quad (5.15)$$

ve x_{ref} referans veya istenen durum vektörüdür. Bu denklemi şu biçimde de yazılabilir;

$$\sigma = \phi(t) - \varphi(x) \quad (5.16)$$

Burada; $\phi(t) = G x_{ref}$ ve $\varphi(x) = G x$ dir. Ayrıca $G = \partial\phi(x)/\partial x$ türevine eşittir. (5.16) denkleminin türevi alınır ve sıfıra eşitlenirse bu çözüme eşdeğer kontrol denir. Diğer bir deyişle kayma fonksiyonunun türevini sıfır yapan kontrole eşdeğer kontrol denir (5.16) denkleminin zamana göre türevi;

$$\frac{d\sigma}{dt} = \frac{d\phi}{dt} - \frac{\partial\varphi}{\partial x} \frac{dx}{dt} \quad (5.17)$$

olur. Bu denklemde dx/dt yerine eşitini yazılırsa;

$$\frac{d\sigma}{dt} = \frac{d\phi}{dt} - G(f(x,t) + Bu) \quad (5.18)$$

$$\left. \frac{d\sigma}{dt} \right|_{u=u_{eq}} = \frac{d\phi}{dt} - G(f(x,t) + Bu_{eq}) = 0 \quad (5.19)$$

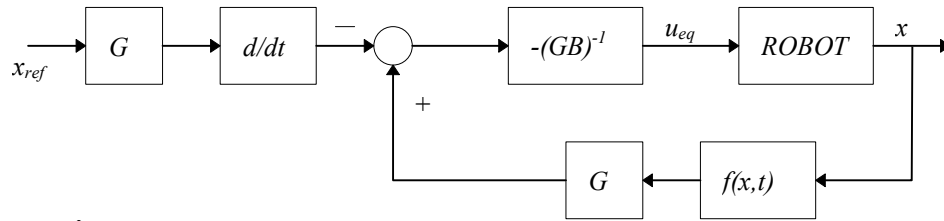
Sonuç olarak eşdeğer kontrol u_{eq} ;

$$u_{eq} = -(GB)^{-1} \left[Gf(x,t) - \frac{d\phi}{dt} \right] \quad (5.20)$$

şeklinde elde edilir. (5.19) denkleminde robotun durum uzayı modelindeki karşılıklarını yerine yazılırsa **Eşdeğer Kontrol Kuralı**;

$$u_{eq} = \tau_{eq} = - \left(G \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} \right)^{-1} \left(G \begin{bmatrix} \dot{q} \\ -M^{-1}(C\dot{q} + g) \end{bmatrix} - \frac{d\phi}{dt} \right) \quad (5.21)$$

şeklinde elde edilir. Eşdeğer kontrollü sistemin blok diyagramı;



ŞEKİL 5.7 Eşdeğer kontrolün blok diyagramı

İkinci Yöntem $\{ u = u_{eq} + K \text{sign}(\sigma) \}$

Bu yöntemin seçilmesindeki neden, birinci yöntemdeki eşdeğer kontrol, sistem ancak kayma yüzeyinde iken geçerlidir. Bu nedenle sistemi kayma yüzeyine sürecektir ve sistemi kayma yüzeyinde tutarak sıfıra götürmesi için ilave bir terim gerekmektedir. Bu tasarım yöntemi bir Lyapunov fonksiyonu seçimi esasına

dayanılarak yapılır. Böylece sistemin kararlılığı garanti edilmiş olur. Bu amaç için önce bir aday Lyapunov fonksiyonu seçelim;

$$v = \frac{1}{2} \sigma^T W \sigma \quad W = W^T > 0 \quad (5.22)$$

Burada W pozitif tanımlı bir matristir. Dolayısı ile v aday Lyapunov fonksiyonu daima pozitif tanımlı olur. Bütünsel sonuçur kararlılık için, Lyapunov kararlılık kriterine göre, aday Lyapunov fonksiyonunun türevi daima negatif tanımlı olmalıdır. Bu amaçla *türevi negatif olan bir Lyapunov fonksiyonu olduğunu ve bunun da türevinin aşağıdaki gibi olduğunu kabul edelim* [Sabanovic, A. and Ohnishi, K][Ertuğrul, M.];

$$\dot{v} = -\sigma^T \Gamma(x, t) \text{sign}(\sigma) < 0 \quad (5.23)$$

Burada $\Gamma(x, t) > 0$ olan pozitif tanımlı bir matristir. (5.22) 'in türevi alınacak olursa

$$\dot{v} = \sigma^T W \dot{\sigma} \quad (5.24)$$

şeklinde bulunur. (5.23) ve (5.24) denklemleri birbirine eşit olmalıdır. Bu denklemler eşitlenirse,

$$-\sigma^T \Gamma \text{sign}(\sigma) = \sigma^T W \dot{\sigma} \quad (5.25)$$

$$-\Gamma \text{sign}(\sigma) = W \dot{\sigma} \quad (5.26)$$

olur. Şimdi, (5.16) denkleminin türevi alınıp (5.26) de yerine yazılırsa;

$$-\Gamma \text{sign}(\sigma) = W \left[\frac{d\phi}{dt} - G(f(x, t) + Bu) \right] \quad (5.27)$$

olur. Buradan kontrol vektörü çekilirse,

$$u = (WGB)^{-1} \left[-WGf(x, t) + W \frac{d\phi}{dt} + \Gamma \text{sign}(\sigma) \right] \quad (5.28)$$

bulunur. Burada sadelik olması için $W=I$ birim matris seçilir ve bu denklem yeniden düzenlenirse,

$$u = (GB)^{-1} \left[-Gf(x, t) + \frac{d\phi}{dt} \right] + (GB)^{-1} \Gamma \text{sign}(\sigma) \quad (5.29)$$

olur. Bu denklemin sağ tarafındaki birinci terim (5.21) de verilen eşdeğer kontrole eşit olduğundan, bunu şöyle yazmak mümkündür;

$$u = u_{eq} + (GB)^{-1} \Gamma \text{sign}(\sigma) \quad (5.30)$$

Denklemin sağ tarafındaki ikinci terimdeki *sign* fonksiyonunun önündeki çarpana K denirse, $K = (GB)^{-1} \Gamma$ olur ve sonuç olarak kontrol işareti;

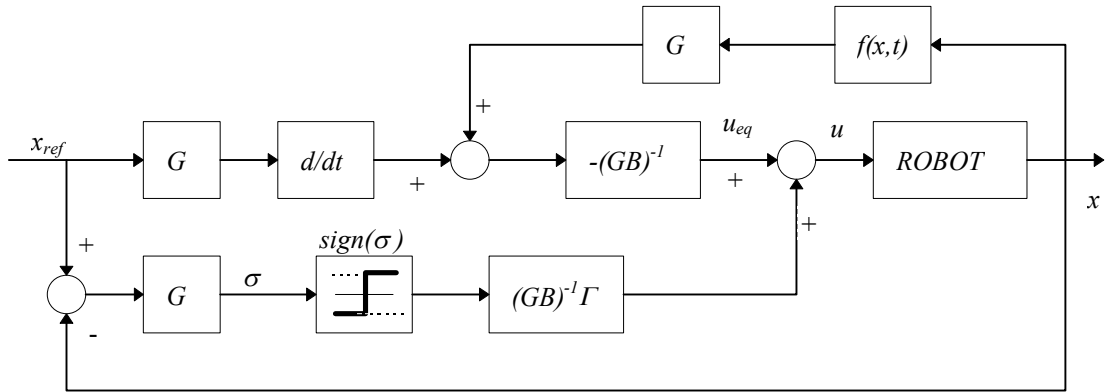
$$u = u_{eq} + K \text{sign}(\sigma) \quad (5.31)$$

biçiminde elde edilir[Sabanovic, A. and Ohnishi, K][Ertuğrul, M][Sabanovic, A., et al].

Buradaki K 'nın içinde bulunan B matrisi, robota ait bir parametredir. K 'nın hesaplanabilmesi için B 'nin bilinmesi gerekir. Diğer G ve Γ matrisleri tasarım parametreleri olduğundan tasarımcı tarafından uygun şekilde seçilir ve dolayısı ile bilinirler. Burada esas önemli olan, eşdeğer kontrol terimine ilave olarak gelen bu ikinci terimdir ve bazı modelleme ve yanlış ilk koşullardan gelecek olan hataları yoketmek için kullanılır. Ayrıca robot parametrelerinin zamanla değişmesine karşılık (5.29) den de görüleceği gibi bu çözüm diğerine göre daha iyi sonuç verir.

Eğer B bilinmiyor ve fakat nominal değerleri kullanılıyorsa, bu durumda K 'nın nasıl seçileceği oldukça önemli olmaktadır. Eğer K çok küçük seçilirse sistem referans yörüngeyi izleyemez. Şayet büyük seçilirse sistem referans yörüngeyi iyi izler fakat bu seferde kontrolör çıkışında osilasyona yani çatırtıya(chattering) neden olur [Sabanovic, A. and Ohnishi, K][Ertuğrul, M][Ostojic, M. et al][Sabanovic, A., et al]. Burada K 'nın seçiminde dikkat edilecek husus, sistemin referans yörüngeyi izlemesi sağlanırken aynı zamanda da kontrolör çıkışında çatırtıdamaya neden olmamaktır. Bu yöntemin kullanılmasının, yalnızca eşdeğer kontrol

yönteminin kullanılmasına göre daha yararlı olduğu açıktır. Bu yöntemin blok diyagramı aşağıda verilmiştir.



ŞEKİL 5.8. İkinci yönteme ait blok diyagram

5.3.3. Üçüncü Yöntem : Kestirilmiş Eşdeğer Kontrol Yöntemi

Burada verilecek olan yöntem ikinci yöntem olarak anılan yöntemde elde edilen kontrol kuralındaki eşdeğer kontrolün hesaplama güçlüğü nedeni ile eşdeğer kontrolün kestirimi (estimation) esasına dayanır. Daha önce de yapıldığı gibi, bir aday Liyapunov fonksiyonu seçerek, kontrol kuralını elde edebiliriz. Eşdeğer kontrol denklemindeki $f(x,t)$ ve B matrisleri tam tamına bilinmiyorsa veya bunlar hakkındaki bilgiler çok az ise, eşdeğer kontrolün hesaplanması ya mümkün değildir yada gerçek eşdeğer kontrolden çok farklı olacaktır. Bu nedenle literatürde eşdeğer kontrolün kestirimine yönelik çalışmalarda, çeşitli kestirim yöntemleri geliştirilmiş ve teklif edilmiştir [Sabanovic, A. and Ohnishi, K]. Bu tezde u_{eq} 'in kestirimi için eşdeğer kontrolün etkisinin fiziksel anlamından hareketle, eşdeğer kontrolün, toplam kontrolün ortalama değeri olduğu esastan hareket edilecektir. Bu nedenle eşdeğer kontrolü, toplam kontrol işaretindeki hızlı değişen yüksek frekanslı bileşenler yerine işaretin tümünün ortalamasını belirleyen alçak geçiren bir filtre ile tasarlamak uygun olur [Sabanovic, A. and Ohnishi, K][Ertuğrul, M].

Kontrola,

$$u = u_{eq} + \Delta u \quad (5.32)$$

denirse u_{eq} ortalama değere ait bileşeni, Δu yüksek frekanslı bileşeni temsil etmektedir Kestirilmiş u_{eq} için şu filtre tasarlanabilir;

$$\tau \hat{u}_{eq} + \hat{u}_{eq} = u \quad (5.33)$$

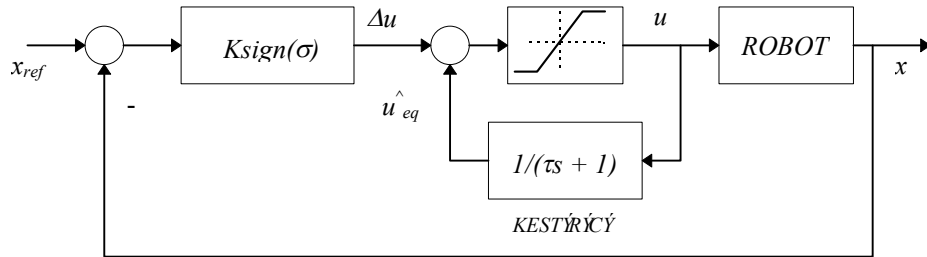
Buradan;

$$\hat{u}_{eq} = \frac{1}{\tau s + 1} u \quad (5.34)$$

Bu bir alçak geçiren filtredir. Burada $1/\tau$ değeri filtrenin kesim frekansının değerini gösterir. Burada bir alçak geçiren filtre tasarlamaktaki temel mantık, bir kontrol sisteminin karakteristiğini alçak frekanslı bileşenlerin belirlemiş olmasıdır. Yüksek frekanslı bileşenler ise genel olarak modellenmemiş bileşenlerden gelir. Bu tanımlara dayanarak kontrol işaretini şu şekilde seçebiliriz;

$$u = \hat{u}_{eq} + K \text{sign}(\sigma) \quad (5.35)$$

Bu yöntemde, kontrol edilecek sistemin $f(x,t)$ ve B matrislerinin bilinmemesi veya az bilinmesi gibi kötü duruma karşılık, bunların kestirilmiş değerleri kullanılarak u_{eq} hesaplamaya karşı gelen, kestirilmiş yada filtrelenmiş eşdeğer kontrol kullanılmaktadır. Böylece (5.30) 'de elde edilen kontrol kuralı yerine hesaplanabilmesi daha kolay olan (5.35) kontrol kuralı elde edilmiş olur [Sabanovic, A. and Ohnishi, K][Ertuğrul, M].



ŞEKİL 5.9. Kestirilmiş eşdeğer kontrollü sistemin blok diyagramı

Eşdeğer kontrolün yapay zeka kullanılarak hesaplanması

Ters kinematik ile Eşdeğer kontrol arasındaki ilişki

Daha önce de belirtildiği gibi kayan kipli kontrol n^{th} dereceden bir sistemi 1^{st} dereceden bir sistem denklemi olarak ifade etmemizi sağlar. Yeni dinamik denklem ,

$$\frac{dS}{dt} = 0 \quad (5.36)$$

İstenen kontrol algoritması için (5.36) un çözümü

$$\frac{dS(X^d, U^d, t)}{dt} = 0 \quad (5.37)$$

$$\frac{dS}{dt} = \frac{d\phi}{dt} - \frac{\partial S_a}{\partial X} \frac{dX}{dt} = \frac{d\phi}{dt} - G(F(X) + BU) \quad (5.38)$$

(5.37) kullanarak , (5.38) yeniden yazılırsa,

$$\frac{dS}{dt} = \frac{d\phi}{dt} - G(F(X^d) + BU^d) = 0 \quad (5.39)$$

5.39 çözümü,

$$U^d = -(GB)^{-1} \left(GF(X^d) - \frac{d\phi(t)}{dt} \right) \quad (5.40)$$

olarak elde edilir.

Eğer sistem kayan modda ise yani referansı tam olarak takip ediyorsa $F(X^d)$ yerine $F(X)$ yazılabilir (gerçek robot çıktısı) Böylece istenen kontrol sinyali.

$$U^d \equiv U_{eq} = -(GB)^{-1} \left(GF(X) - \frac{d\phi(t)}{dt} \right) \quad (5.41)$$

Kontrol Algoritması:

Basamak 1. Başlama koşullarını ata

Yapay zeka ağırlıkları (weights) 0.05 ufak rasgele sayılar seçilir.

adaptasyon hızı $0 < \mu < 1$. olacak şekilde seçilir

Basamak 2. Saklı yüzeyler ve yapay zeka çıktısının hesaplanması

equivalent kontrol kullanarak yapay zeka çıktısı hesaplanır

Step 3. ΔU nun hesaplanması

additional control term (ΔU) hesaplanır

Step 4. Robota uygulanacak kontrol sinyalinin bulunması

equivalent kontrol ile additional kontrol toplanarak robota uygulanacak kontrol kuvvetleri bulunur

Step 5. Robot çıktısı ölçülür

Measure the angular positions and velocities of the robot.

Step 6. Yapay zeka parametreleri güncellenir

Aşağıdaki formül il eyapay zeka parametreli güncellenir,

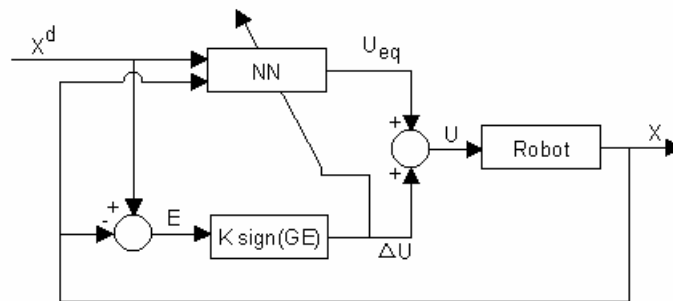
$$Wy_{j,i}(t+1) = Wy_{j,i}(t) + \mu \delta y_j Yout_i \quad j=1..2, \quad i=1..M$$

$$W_{z_{j,i}}(t+1) = W_{z_{j,i}}(t) + \mu \delta z_j Z_i \quad j=1..M, \quad i=1..8$$

Burada,

$$\delta y_j = (\Delta U_j)(1 - Ueq_j^2) \text{ and } \delta z_j = \left(\sum_{k=1}^2 \delta y_k W_{y_{k,j}} \right) (1 - Yout_j^2)$$

Step 7. İkiye dönülür



ŞEKİL. 5.10. Neuro sliding mode control

5.4 Gauss Fonksiyonlu Yapay Zeka Destekli Kayan Kipli Kontrol (GNFVSC)

GNFVSC tasarımı 2 ana bölüm içermektedir; birinci bölüm hesaplanmış(tahmin edilmiş) equivalent controller ve ikinci bölüm ise fuzzy corrective controllerlerdir. Fuzzy corrective controller motivasyonu VSC'nin geleneksel kurallarını temel alır. This is to force the system states into the thin boundary layer neighboring the sliding surface in the reaching phase. Kayan evrede tahmin edilen (beklenen) equivalent kontrol değeri için RBFNN ve VSC 'nin sinerjisi birleşir. Fuzzy equivalent control kanunu sistem dinamiklerini, sınır katmanında kalmaları için zorlar. RBFNN kanunu , sınır katmanındaki hata bilgisi gibi , fuzzy corrective sinyalden elde edilir. Bunun anlamı şudur; RBFNN'nin adaptasyonu sadece sınır katmanında yapılmalıdır. Bu adaptasyon , RBFNN 'nin öğrenilme ve plant süresi için uygulanabilir olmalıdır bu yüzden toplam kontrol çabasını en aza indirir.

Bulanık Düzeltici Kontrol (Fuzzy Corrective Controller)

Bu bölümde , fuzzy corrective controller , FL teorisi kullanılarak yapılandırılır. Amaç $\mathbf{s}^T \dot{\mathbf{s}} = -\mathbf{s}^T \mathbf{D} \operatorname{sgn}(\mathbf{s})$ dayanarak $S_i^T \dot{S}_i < 0$ şeklinde bir kontrol kuralı elde etmektir. Bu anlatım $\dot{\mathbf{s}} = \dot{\boldsymbol{\phi}} - \frac{\partial \mathbf{s}_a}{\partial \mathbf{x}} \dot{\mathbf{x}} = \dot{\boldsymbol{\phi}} - \mathbf{G}(\mathbf{f}(\mathbf{x}, t) + \mathbf{B}(\mathbf{x}, t)\mathbf{u}(t))$ kullanılarak aşağıdaki gibi yazılabilir,

$$S_i^T \dot{S}_i = S_i^T \dot{\boldsymbol{\phi}} - S_i^T (GF(X, t))_i - S_i^T B(X, t)U_i \quad (5.42)$$

$B(X, t) > 0$ şeklindeki bir varsayım alınıp daha sonra $\mathbf{s}^T \dot{\mathbf{s}} = -\mathbf{s}^T \mathbf{D} \operatorname{sgn}(\mathbf{s})$,a göre düzenlendiğinde , aşağıdaki yargıya varılabilir: Eğer S_i pozitif ise o zaman control sinyali U_i yükseltilmiş olmalıdır ve eğer kayan değer S_i negatif ise o zaman control sinyali U_i , $S_i^T \dot{S}_i$ 'e düşürmek için azaltılmış olmalıdır. Böyle bir sonuç için , fuzzy corrective control $U_{c_i}^f$ ve $S(X, t)_i$ 'nin aralarındaki karakteristik ilişki niteliksel olarak belirtilebilir: “ $U_{c_i}^f$ belirli limitler içinde S_i kadar geniştir” örneğinde olduğu gibi. fuzzy inference mekanizmasında ayarlanmış data , fuzzy set teorisinde temeldir., the associated fuzzy set involved in the fuzzy rules are defined aşağıdaki gibidir : AZ: daima sıfır , NS : negatif küçük , PM pozitif orta , PB : pozitif büyük , PL : pozitif geniş , PH : pozitif çok büyük . üyelik fonksiyonları Şekil 5.11. 'de belirtilen kayan yüzey değeri S_i ve fuzzy corrective control $U_{c_i}^f$ için uygun olanı hazırlar ve onların $[-1, 1]$ aralığında alacağı tüm değerleri saptar. S_i için sadece 11 fuzzy altkümesine kadar tanımlanabilir, fuzzy çıkarım mekanizması sadece 11 IF-THEN kuralı içerir. Fuzzy çıkarım mekanizmasının sonuçları aşağıdaki gibidir.

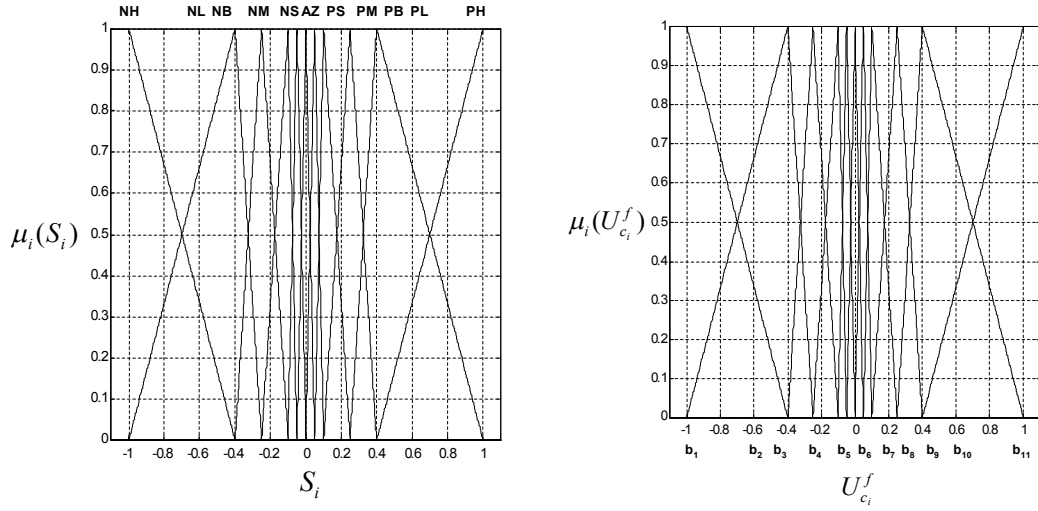
- | | |
|--|--|
| Rule 1 : if S_i is PH then $U_{c_i}^f$ is PH, | Rule 2 : if S_i is PL then $U_{c_i}^f$ is PL, |
| Rule 3 : if S_i is PB then $U_{c_i}^f$ is PB, | Rule 4 : if S_i is PM then $U_{c_i}^f$ is PM, |
| Rule 5 : if S_i is PS then $U_{c_i}^f$ is PS | Rule 6 : if S_i is AZ then $U_{c_i}^f$ is AZ, |
| Rule 7 : if S_i is NS then $U_{c_i}^f$ is NS, | Rule 8 : if S_i is NM then $U_{c_i}^f$ is NM, |
| Rule 9 : if S_i is NB then $U_{c_i}^f$ is NB, | Rule 10 : if S_i is NL then $U_{c_i}^f$ is NL, |
| Rule 11 : if S_i is NH then $U_{c_i}^f$ is NH. | |

Kural 2 şudur ; eğer kayan yüzey değeri S_i sınır katmanından çok uzakta ise , geniş bir $U_{c_i}^f$ kayan devinim koşulunun gerçekleşmesi için gereklidir. Daha fazlası, kural 6 şunu belirtir; eğer kayan yüzey değeri S_i sınır katmanının içinde ise, kayan

devinim koşulunun gerçekleşmesi için sadece çok küçük $U_{c_i}^f$ değeri gereklidir. Aynı analiz diğer fuzzy kuralları için de kullanılabilir.

Bulanıklaştırma, merkezi ortalama ters bulanıklaştırma ve üçgensel üyelik fonksiyonları kullanılan Sugeno'nun sonuçlandırma formülasyonu fuzzy corrective controller'ı [Wang, L.-X.,][Lee, C.C] dizayn etmekte kullanılır. Bu formülasyon kullanılarak, S_i girdisi alınır, $U_{c_i}^f$ kesin çıktısı aşağıdaki gibi elde edilir

$$U_{c_i}^f = T_i \left(\sum_{k=1}^{11} \mu_{i_k}(S_i) b_{i_k} \right) / \sum_{k=1}^{11} \mu_{i_k}(S_i) \quad (5.43)$$

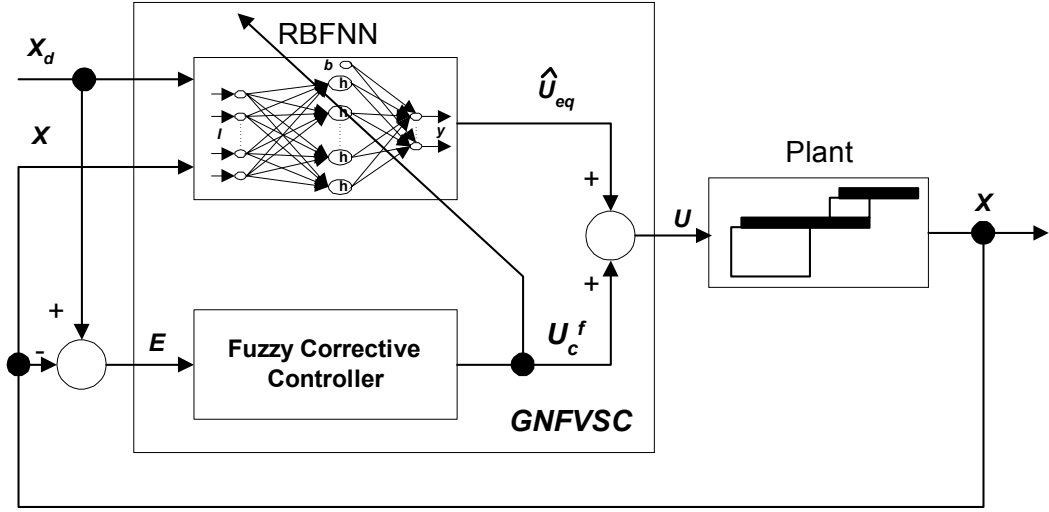


ŞEKİL 5.11. Bulanık küme üyelik fonksiyonları

$\mu_{i_k}(S_i)$, k^{th} kuralının önceki fuzzy bölümünde S_i 'nin ilişki fonksiyonudur. b_{i_k} , k^{th} kuralının izleyen(sonraki) fuzzy bölümünün kitle merkezidir ve T_i ölçülen değerdir.

Equivalent Control Düşüncesi equivalent control ve (5.43)'de gösterilen fuzzy corrective control değeri Gaussian fonksiyonu temelli uyarlanmış RBFNN fikrinin dizaynı için başlangıç noktasıdır [Palaz, H., Sarioglu, K., Akbas E., and Ertugrul, M.,]. Equivalent control equation system matrixlerini $F(X)$ ve B 'leri içerir. Bunun yanısıra, sistem dinamikleri hakkında güçlü bir bilgi birikimine sahip olmaksızın equivalent control'un kesin değerlerini hesaplamak çok zordur Bir diğer ifade ile, system parametreleri ve system dinamiklerinin (veya ters dinamiklerinin) bütün bilgisi equivalent control'un saptanması için gereklidir. Aslında bu pratikte oldukça nadirdir. Bu problemin çözümü için equivalent control [Palaz, H., Sarioglu, K., Akbas E., and Ertugrul, M.,] düşüncesine göre RBFNN kullanılabilir.

Bir yaklaşıma göre değerlendirildiğinde , VSC'deki equivalent control terimi $\hat{U}_{eq}$, bir uyarlanmış adaptive RBFNN tarafından sınır katmanı içinde tahmin edilir. Kayan modda Fuzzy corrective control terimi U_c^f (5.43) de olduğu gibi de hesaplanır ve control sinyal formu için RBFNN çıktısı ile toplanır. Fuzzy corrective control terim çıktısı RBFNN [Palaz, H. vd.] 'nın önemini(ağırlığını) güncelleştirmek için hata ölçüsü olarak kabul edilir. Tüm önerilen system controller' ı Şekil. 5.12. de verilmiştir.



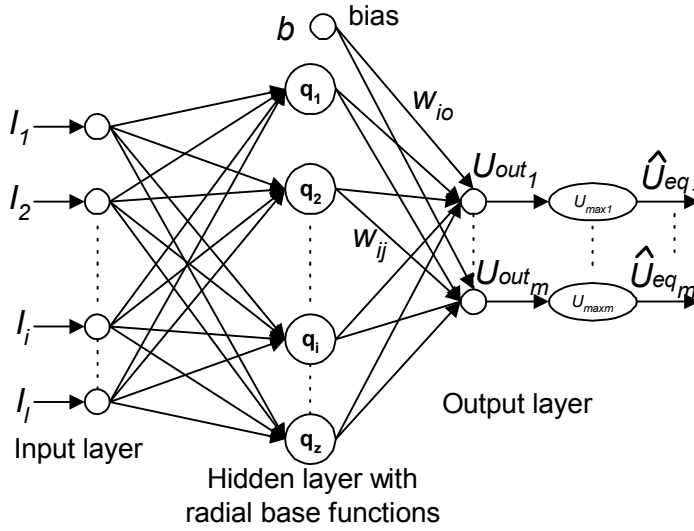
ŞEKİL 5.12. GNFSVC yapısı

RBFNN yapısı şekil 5.13 de gösterilmiştir ve 3 katmandan oluşmaktadır. İlk katman istenen ve gerçek durum için bağlantı katmanıdır. İkinci katman sadece gizli katmandır. Girdi vektör aralığından iç vektör aralığına lineer olmayan dönüşümü sağlar. Son katman çıktı katmanıdır. Lineer biçimde çıktı vektör aralığını iç vektör aralığına dönüştürür. Gizli katmanın Neuronları gerçek temel fonksiyondur. Bu fonksiyonların tipleri teoride önemli değildir ve genellikle Gaussian fonksiyonları aşağıdaki gibi kullanılır,

$$q_j(I) = e^{-\frac{1}{2} \frac{\|I - c_j\|^2}{\lambda_j^2}} \quad \lambda_j > 0 \quad j = 1, 2, \dots, z \quad (5.44)$$

I çok boyutlu bir girdi vektörü , c_j aynı boyutlu RBF merkez vektörü I gibi , λ_j RBF 'nin genişliği , z RBF 'nin sayısı , ve $q_j(.)$ is the i^{th} RBF orijindeki tek maksimumdur . Çıktı katmanındaki Neuronlar lineer fonksiyonları hesaplar.

$$f_i(I) = U_{out_i} = w_{i0}b + \sum_{j=1}^z w_{ij}q_j(I) \quad i = 1,..m \quad (5.45)$$



ŞEKİL 5.13. RBFNN yapısı

Index i çıktı neuronu i' , 'yi gösterir. Index j tüm gizli katman neuronları sayesinde çalışır., b bir önyargı girdisidir. w_{ij} gizli neuron j 'den çıktı neuron'u i' 'ye bir bağlantı ağırlığıdır. U_{out_i} RBFNN [Haykin S.][Bishop C.M] çıktısıdır i^{th}

Lineer olmayan sistemlerin kontrolü için yüksek performanslı GNFBVSC elde edilir. Gerçek bir RBFNN mimarisi tanımlamak için gereklidir. Ağ mimarisi her bir katmandaki node sayısı ve fonksiyon merkezinin konumuna göre ifade edilir. Girdi ve çıktı katmanındaki node sayısının tespiti equivalent control denkleminde seçilir. equivalent control düşüncesinde, tüm istenen ve gerçek değerler kullanılır. Bu nedenle, girdi katmanındaki nodeların sayısı istenen ve gerçek durumun sayısıdır. Robot (platform) için çıktı katmanındaki nodeların sayısı kontrol girdilerinin sayısıdır . Mesela robot kontrol uygulamasında, çıktı katmanındaki nodeların sayısı robotun harekete geçiricilerinin sayısı tarafından saptanır. .

Gizli katman nodelarının sayısının seçimi ve merkezlerin konumu GNFBVSC'daki RBFNN 'in performansını derin etkiler yapacaktır. Gizli katmandaki neuronların sayısı seçilmelidir. RBFNN, robotun ters dinamikleri için haritalama kapasitesine sahiptir.

RBFNN'nin tüm girdileri $[-1,1]$ aralığında normallenir ve ağ için uygulanır. The Gaussian fonksiyonları en iyi donuşümü elde etmek için $[-1,1]$ girdi sinyallerinin çalıştığı aralıkta, düzenli bir şekilde dağıtılır (sınıflandırılır). Böylece

RBFNN' nin çıktıları $[-1,1]$ aralığında sınırlandırılmış olur. Tahmin edilen equivalent control sinyali elde etmek için, RBFNN'nin normalleştirilmiş çıktısı, uygulamadan önce maksimum kontrol değeri tarafından çoklanır. RBF'nin merkezi ve diğer bütün serberst seçilen öğrenme parametreleri supervised öğrenmeye tabii tutulur; bir diğer anlamda, the RBF ağı en genelleştirilmiş halini alır. Böyle bir öğrenme processinin en dogal adayı gradient-descent procedurudur [Haykin S]. Genelleştirilmiş RBF ağı (centerlar, çıkış katmanındaki ağırlıklar ve RBF fonksiyonunun kendisi güncellendikten sonra multilayer perceptrons performansından daha iyi bir sonuç elde edilir .[Sanner, R.M., and Slotine, J.-J.E][Park, J. and Sandberg, J.W].

RBFNN'nin girdileri I normalleştirilmiş istenen ve gerçek durumlardan oluşmaktadır.

$$I = [(X_d)^T \quad X^T]^T \quad (5.46)$$

RBFNN'nin çıktısı tahmin edilen equivalent kontrol değeridir. Ve aşağıdaki gibi hesaplanır,

$$\begin{aligned} \hat{U}_{eq_i} &= f_i(I) U_{\max_i} \\ &= \left[w_{i0} b + \sum_{j=1}^z w_{ij} q_j(I) \right] U_{\max_i} \quad i=1..m \end{aligned} \quad (5.47)$$

Daha sonra , robota uygulanacak GNFVSC 'tasarımı elde edilebilir olmalıdır;

$$U_i(t) = \hat{U}_{eq_i} + U_{c_i}^f = \left[w_{i0} b + \sum_{j=1}^z w_{ij} q_j(I) \right] U_{\max_i} + T_i \left(\frac{\sum_{k=1}^{11} \mu_{i_k}(S_i) b_{i_k}}{\sum_{k=1}^{11} \mu_{i_k}(S_i)} \right) \quad i = \quad (5.48)$$

Sınırlı katmandaki RBFNN yasası

Sınırlı katmanda RBFNN prosedürünün geliştirilmesindeki ilk adım (S_b) , değer fonksiyonunun anlık değerini belirtmektir.. Değer fonksiyonu k th aşağıdaki gibi seçilir,

$$J(k) = \frac{1}{2} \sum_{j=1}^m (U_{out_j}^d - U_{out_j})^2 = \frac{1}{2} \sum_{j=1}^m \left(\frac{U_{eq_j} - \hat{U}_{eq_j}}{U_{\max_j}} \right)^2 \quad \text{subject to } -S_b < S_j < S_b \quad (5.49)$$

(5.49)'deki denklemde , $(U_{eq_j} - \hat{U}_{eq_j})$ terimi hesaplanamaz, çünkü equivalent control'un istenen değeri (U_{eq_j}) bilinmiyor. Kayan mod durumundaki bir değişiklik bu problemi çözebilir [Ertugrul, M.]. Fark aşağıdaki gibi yazılabilir,

$$U_{eq_j} - \hat{U}_{eq_j} \equiv \Delta U_{eq_j} \quad (5.50)$$

(3.11) 'de verilen VSC denklemi aşağıdaki gibi yeniden yazılabilir.

$$U_j = U_{eq_j} + U_{c_j}^* \quad (5.51)$$

$U_{c_j}^*$ düzeltici bir kontroldür ve equivalent control'un istenen değeri kullanıldığı zaman, çeşitli durum controllerı tarafından hesaplanır. (5.48) ve (5.49) kullanılarak , U_j aşağıdaki gibi hesaplanabilir,

$$\begin{aligned} U_j = U_{eq_j} + U_{c_j}^* &= (\hat{U}_{eq_j} + \Delta U_{eq_j}) + U_{c_j}^* \\ &= \hat{U}_{eq_j} + U_{c_j}^f \end{aligned} \quad (5.52)$$

$U_{c_j}^f = U_{c_j}^* + \Delta U_{eq_j}$ equivalent kontrolün istenen değeri kullanıldığı zaman (5.52) 'deki gibi çözülebilir. Fuzzy corrective control, hesaplanan ve istenen equivalent control arasındaki fark ve sınırlı katmandan (veya kayan yüzeyden) sapmaların her ikisinde hesaplar.

Kayan bir hareket olduğu zaman veya S_j sınırlı katmanda ise , $U_{c_j}^*$ değeri sıfırla bir noktada birleşiyorsa böylece $U_{c_j}^f$ ve ΔU_{eq_j} da bir noktada birleşir. Bu beklenen bir sonuçtur, çünkü fuzzy corrective control sadece sınırlı katmanda bir hareket gerçekleştiği zaman hesaplanan ve beklenen değer arasındaki farkı karşılar. Sonuç olarak , S 'nın sınırlı katmanda kalan küçük değeri için, fark aşağıdaki gibi hesaplanabilir,

$$\lim_{S_j \rightarrow 0} (U_{eq_j} - \hat{U}_{eq_j}) = U_{c_j}^f \quad (5.53)$$

(4.8)'deki denklem aşağıdaki gibi yazılabilir,

$$J(k) = \frac{1}{2} \sum_{j=1}^m \left(\frac{U_{eq_j} - \hat{U}_{eq_j}}{U_{\max_j}} \right)^2 = \frac{1}{2} \sum_{j=1}^m \left(\frac{U_{c_j}^f}{U_{\max_j}} \right)^2 \quad \text{subject to } -S_b < S_j < S_b \quad (5.54)$$

RBFNN 'nin tüm boş parametrelerinin optimizasyonu için denetlenmiş bilgi süreci temelli bir eğim prosedürü kullanılır. Bu bilgi on-line bir ilerlemedir. Sınırlı katmanda J minimizasyonunda olduğu gibi (S_b) , boş parametreleri bulmak için

gerekli parametreler w_{ij} , c_j and λ_j . Minimizasyon prosedürü aşağıdaki gibi verilir [Haykin S][Bishop, C.M., 1991].

Gradient descent methodu ile Lineer aralıklar için uyarlama formülleri,

$$w_{ij}(k+1) = \begin{cases} w_{ij}(k) - \eta_1 \frac{\partial J(k)}{\partial w_{ij}(k)} & \text{if } -S_b < S_i < S_b \\ w_{ij}(k) & \text{otherwise} \end{cases} \quad i = 0, \dots, m \quad j = 0, 1, \dots, z \quad (5.55)$$

$$\frac{\partial J(k)}{\partial w_{ij}(k)} = - \left(\frac{U_{c_j}^f(k)}{U_{\max_j}} \right) q_j(I) \quad (5.56)$$

Gradient descent methodu ile RBF ‘merkezlerinin konumları için adaptasyon formulleri,

$$c_j(k+1) = \begin{cases} c_j(k) - \eta_2 \frac{\partial J(k)}{\partial c_j(k)} & \text{if } -S_b < S_i < S_b \\ c_j(k) & \text{otherwise} \end{cases} \quad j = 1, \dots, z \quad (5.57)$$

$$\frac{\partial J(k)}{\partial c_j(k)} = - \frac{1}{\lambda_j^2} \sum_{i=1}^m \left(\frac{U_{c_j}^f(k)}{U_{\max_j}} \right) w_{ij}(k) q_j(k) [I - c_j(k)] \quad (5.58)$$

Gradient descent metoduna göre standart Gaussian fonksiyonlarının genişliği için adaptasyon formülleri,

$$\lambda_j(k+1) = \begin{cases} \lambda_j(k) - \eta_3 \frac{\partial J(k)}{\partial \lambda_j(k)} & \text{if } -S_b < S_i < S_b \\ \lambda_j(k) & \text{otherwise} \end{cases} \quad j = 1, \dots, z \quad (5.59)$$

$$\frac{\partial J(k)}{\partial \lambda_j(k)} = - \sum_{i=1}^m \left(\frac{U_{c_j}^f(k)}{U_{\max_j}} \right) w_{ij}(k) q_j(I) \frac{\|I - c_j(k)\|^2}{\lambda_j^3(k)} \quad (5.60)$$

Uyarlama formülleri, η_1, η_2, η_3 bilgi oranlarını belirtir.

$$q_j(I) = e^{-\frac{1}{2} \frac{\|I - c_j\|^2}{\lambda_j^2}} \quad \lambda_j > 0 \quad j = 1, 2, \dots, z \quad (5.61)$$

I çokboyutlu bir girdi vektörü, c_j aynı boyutlu RBF merkez vektörü I gibi, λ_j RBF ‘nin genişliği, z RBF ‘nin sayısı, ve $q_j(.)$ is the i^{th} RBF orijindeki tek maksimumdur. Çıktı katmanındaki Neuronlar lineer fonksiyonları hesaplar.

$$f_i(I) = U_{out_i} = w_{i0}b + \sum_{j=1}^z w_{ij}q_j(I) \quad i = 1,..m \quad (5.62)$$

5.5. Adaptif Yapay-Bulanık Değişken Yapılı Kontrol

ANFVSC

Adaptif sistemler (Adaptive Neuro-Fuzzy Inference Systems) yapay sinir ağları ile bulanık mantığın bir kombinasyonudur. Bu hibrit karışım zeki sistemlerinin nümerik ve sözcüksel gücü daha etkili bir şekilde kullanılmasını sağlar. Bulanık mantık teorisinden de bildiğimiz gibi fuzzification ve defuzzification stratejileri ile kural tabanlarının (rule base) seçimi sonucu etkiler. Bu çalışma ANFIS yapısını birinci dereceden Sugeno modelini 9 kuralla uygulanmasıdır. Gaussian üyelik fonksiyonları (membership functions) fuzzification için kullanılan yöntemdir. İki giriş ve 1 çıkış için yapı şekil 5.14’ de gösterilmiştir. Bunun genelleştirilmiş hali olan m giriş için 1 çıkış için genel formül aşağıdadır.

IF u_1 is $U_{i,1}$ AND u_2 is $U_{i,2}$ AND ... AND u_m is $U_{i,m}$
 THEN $f_i = q_{i,1}u_1 + \dots + q_{i,m}u_m + q_{i,m+1}$

Yukarıdaki denklemde küçük harfle belirtilenler girişler büyük harfle belirtilenler bulanık kümeleri ifade etmektedir. ANFIS çıkışı liner bir fonksiyona çok yaklaşan bir çıkış olup ters bulanıklaştırma (defuzzifier) parametreleri $q_{i,j}$ ile ifade edilmiştir. Sistemdeki Gaussian üyelik fonksiyonları denklem (5.63) de gösterilmiştir.

$$\mu_{ij}(u_j) = \exp \left\{ - \left(\frac{u_j - c_{ij}}{\sigma_{ij}} \right)^2 \right\} \quad (5.63)$$

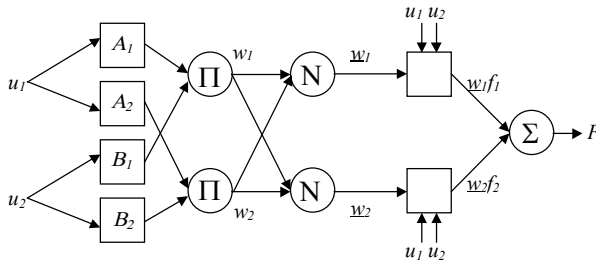
Yukarıdaki denklemde, c_{ij} ve σ_{ij} i^{th} kuralın merkezini ve j^{th} üyelik fonksiyonunun genişliğini ifade etmektedir. Üyelik fonksiyonlarının başlangıç değerleri ilgi alanımızı tamamen kaplayacak şekilde seçilmektedir. Denklem (5.64) tüm algılamayı yapan sistemin matematiksel modelidir

$$F = \frac{\sum_{i=1}^{\# Rules} f_i \prod_{j=1}^{\# Inputs} \mu_{ij}(u_j)}{\sum_{i=1}^{\# Rules} \prod_{j=1}^{\# Inputs} \mu_{ij}(u_j)} = \sum_{i=1}^{\# Rules} f_i w_{ni} \quad (5.64)$$

$$w_{ni} = \frac{\prod_{j=1}^{\# Inputs} \mu_{ij}(u_j)}{\sum_{k=1}^{\# Rules} \prod_{j=1}^{\# Inputs} \mu_{kj}(u_j)} \quad (5.65)$$

Denklem (5.64) de tetikleme vektörleri normalleştirilirse ve çıkan vektör w_n ile gösterilirse denklem (5.65) elde edilir. Uyarlanacak olan parameter seti de aşağıdaki gibi ifade edilebilir.

$$\phi = \{c_{ij}, \sigma_{ij}, q_{i,j} \mid i=1, \dots, Rules, j=1, \dots, Inputs\} \quad (5.66)$$



ŞEKİL.5.14. ANFIS Yapısı

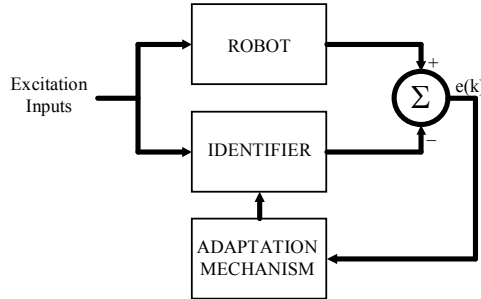
ANFIS kullanılarak yapılan kontrol uygulamaları literatürde pek çok yerde tartışılmıştır. Bunlardan en önemlilerinden bazıları [Narendra, K. S., K. Parthasarathy]. ANFIS sistemlerini sistem dinamiğini tanımda (identification) kullanımları da [Jang, J. et al] da tartışılmıştır.

Sistem Dinamiğinin Neuro Fuzzy Yapılar ile Tespit Edilmesi

Sistem dinamiğinin tespiti ve kontrolü çok boyutlu bir yüzeyin algılanması, tespiti olarak görülebilir. Bu strateji aşağıdaki gibi ifade edilirse ki burada x , x_d , ve τ sırasıyla ölçülebilir durum, istenen durum ve sisteme uygulanan girişlerdir

- a) $[\underline{x}(k), \underline{\tau}(k)] \Rightarrow [\underline{x}(k+1)]$
- b) $[\underline{x}_d(k+1) - \underline{x}(k)] \Rightarrow [\underline{\tau}(k+1)]$

Burada ilk denklemin amacı sistemi tanıyan yapıyı eğitmek ve ikinci denklemin amacı istenilen çıktıları üretebilmesi için sistemin kontrol yapısını eğitmektir. Herhangi bir sistemi tanımanın diyagramı Şekil 5.15 da gösterildiği gibi ifade edilebilir.



ŞEKİL 5.15. Sistem Tanıma modeli

Sistemdeki takip hatasına (tracking error) göre kontrolör bir dahaki seferde uygulayacağı gerekli giriş sinyalini hesaplar, bu arada eğitim işlemi de yapılır ve zeki sistemin parametreleri de bu hataya göre değiştirilir.

5.6. Test Sonuçları

Yukarıda bahsedilen yapay kontrol tekniklerini daha önce literatürde olmadığı şekilde karşılaştırmak ve bunların özellikle sanayi şartlarındaki performans parametreleri ve kolay uygulanabilirlikleri anlamında sonuç elde etmek için kurulan düzenek literatürde kullanılan [E. Akbaş ve E. Murat Esin] bir düzeneştir.

Robot sisteminin performansını karşılaştırmak için Şekil 5.16. de gösterilen eklem düzlemi (a joint space) seçilmiştir. Bu sistemin matematiksel ifadesi denklem 5.67 ve denklem 5.68 de gösterilmiştir.

$$\dot{\theta}_{d1} = \text{acos}((X^2 + Y^2 - l1^2 - l2^2) / l1l2) \times (\text{atan2}(l2 \sin(X), (l1 + l2 \cos(X)))) \quad (5.67)$$

$$\dot{\theta}_{d2} = \text{acos}((X^2 + Y^2 - l1^2 - l2^2) / l1l2) \quad (5.68)$$

Burada X ve Y kartezyen koordinat parametreleri.

$L1$ =Kol 1 uzunluğu;

$L2$ = Kol 2 uzunluğu;

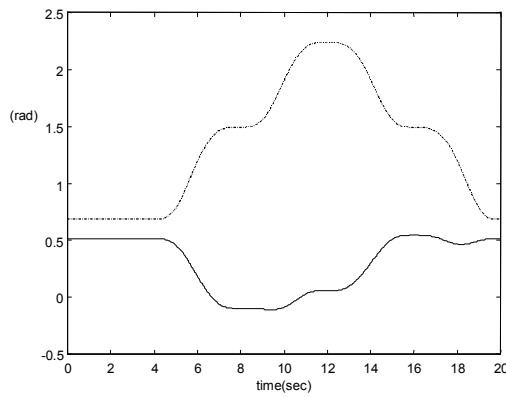
$I1$ ve $I2$ aşağıdaki gibi hesaplanır;

$I1=L1$; $I2=L2$.

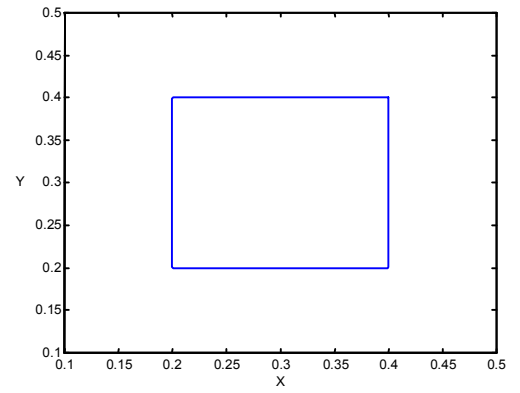
$I12$, $I22$ ve $I1I2$ aşağıdaki gibi hesaplanır;

$I12=I1 \times I1$; $I22=I2 \times I2$; $I1I2=2.0 \times I1 \times I2$.

Şekil 5.17. da referans kartezyen koordinatları



ŞEKİL 5.16. Eklem Koordinatları



ŞEKİL 5.17. Kartezyen Koordinatları

Hata hesaplama denklemi

$$\left| error_i = \sqrt{\sum_{j=1} \left(q_i(j) - q_i^d(j) \right)^2} \right| \quad (5.69)$$

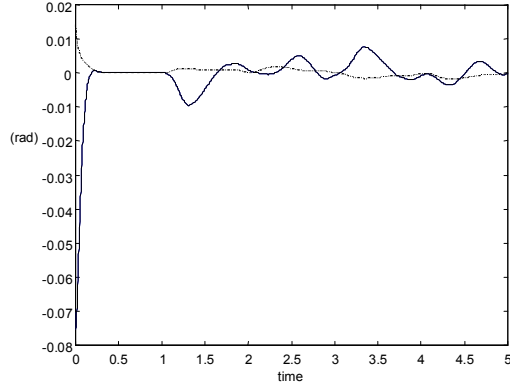
Max hata tanımı denklem 5.70 de gösterilmiştir

$$\max error_i = \max_{1 \leq j \leq k} |q_i(j) - q_i^d(j)| \quad (5.70)$$

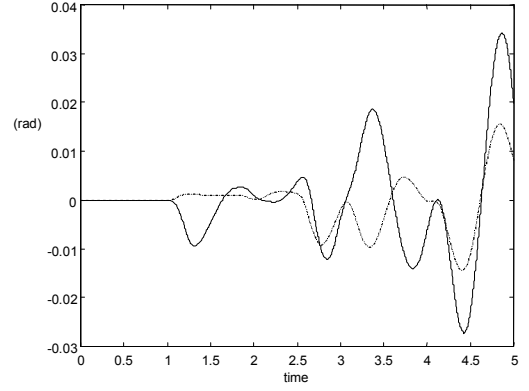
Ve kartezyen koordinatlardaki hata tanımı ki bizim elde etmek istediğimiz bu denklemi minimize etmek, denklem 5.71 de gösterilmiştir.

$$\left| error_{taskspace} \right| = \sqrt{\sum_j \left(\left\| \begin{pmatrix} x(j) - x^d(j) \\ y(j) - y^d(j) \end{pmatrix} \right\| \right)^2} \quad (5.71)$$

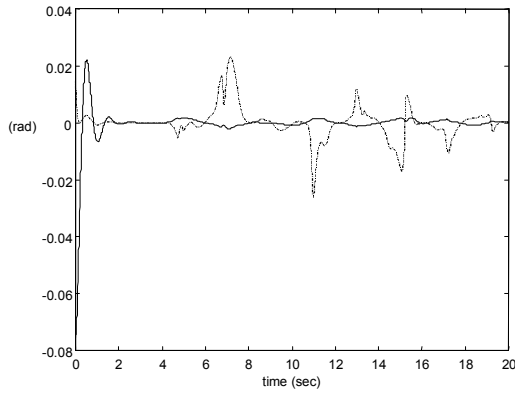
Çizelge 5.1 ve Çizelge 5.2 de belirlenene kriterlere göre yapılan karşılaştırmalar verilmiştir. Bu karşılaştırmalara yaparken elde edilen sonuçlar da Şekil 5.18 ile Şekil 5.35 ile gösterilmiş olup genel bir karşılaştırma hariç literatürde bu metodların performans karşılaştırmaları [E. Akbaş and H. Palaz][E. Akbaş and E. Murat Esin,] verilmiştir.



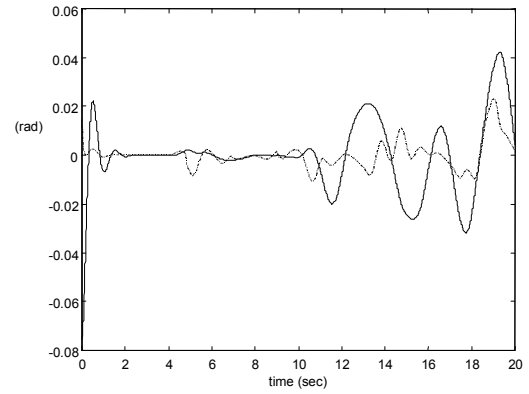
ŞEKİL 5.18. Pozisyon hataları (Neuro-SMC)



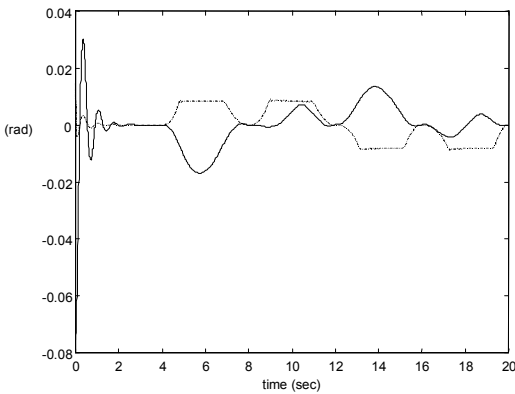
ŞEKİL 5.19. Ani yük değişimi pozisyon hataları (Neuro-SMC)



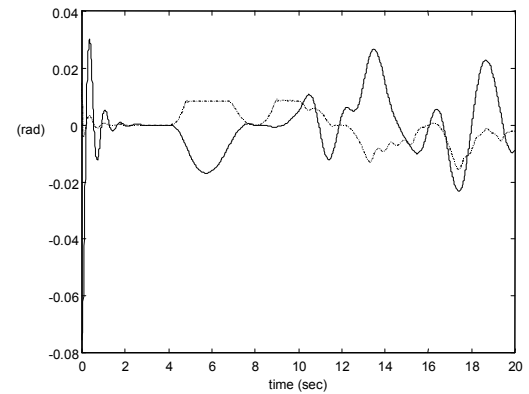
ŞEKİL 5.20. Pozisyon hataları (DIMA)



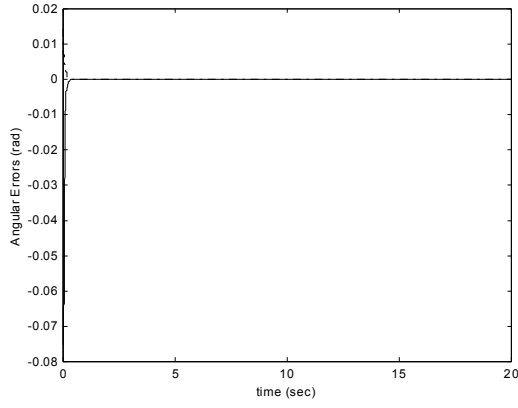
ŞEKİL 5.21. Ani yük değişimi pozisyon hataları(DIMA)



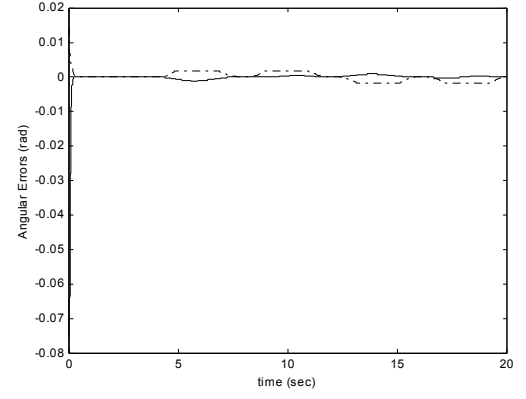
ŞEKİL 5.22. Pozisyon hataları(FELA)



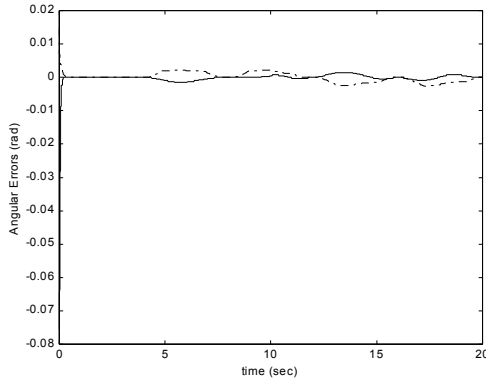
ŞEKİL 5.23. Ani yük değişimi pozisyon hataları(FELA)



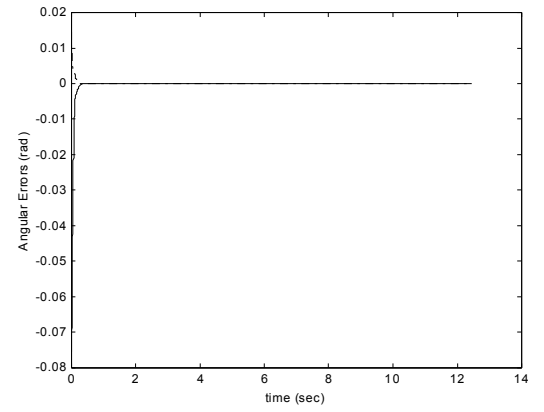
ŞEKİL 5.24. GNFSVC- Uc



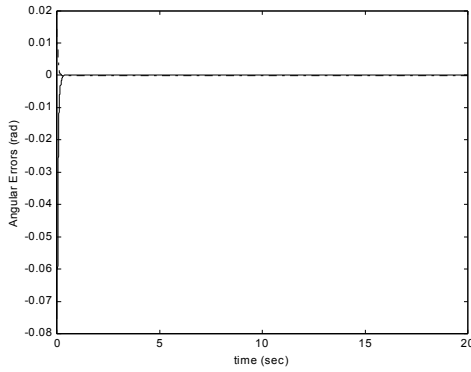
ŞEKİL 5.25. GNFSVC-Uc- not Boundry



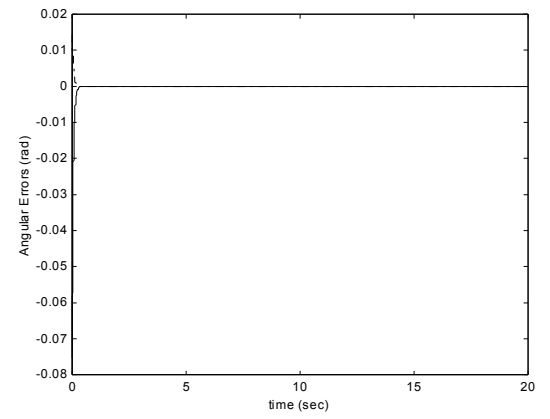
ŞEKİL 5.26. GNFSVC-Uc -Boundry



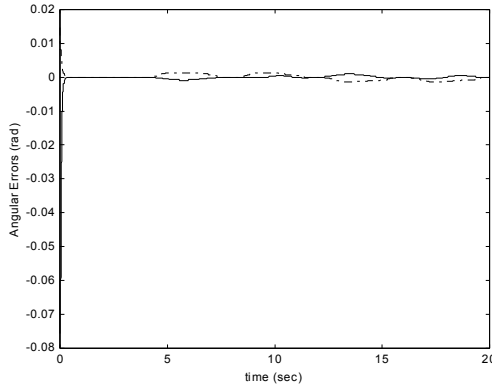
ŞEKİL 5.27. GNFSVC-Uc- Boundry-payload



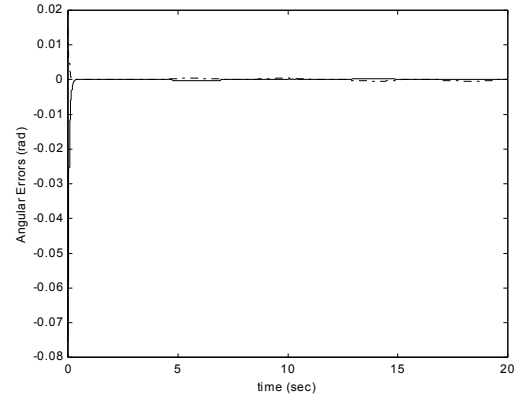
ŞEKİL 5.28. GNFSVC



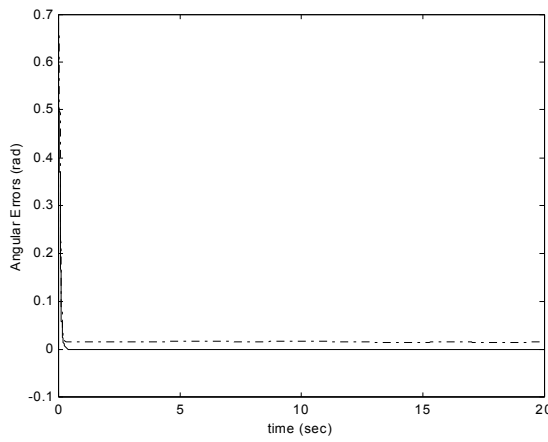
ŞEKİL 5.29. GNFSVC not boundry



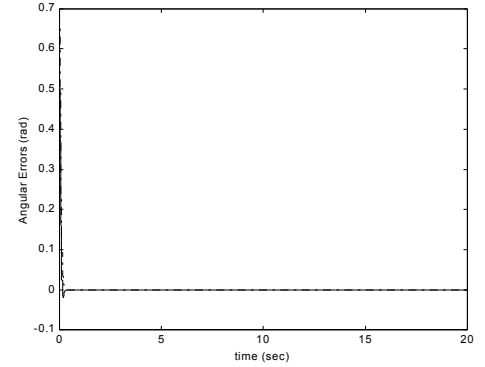
ŞEKIL 5.30. GNFSVC -not boundry - payload



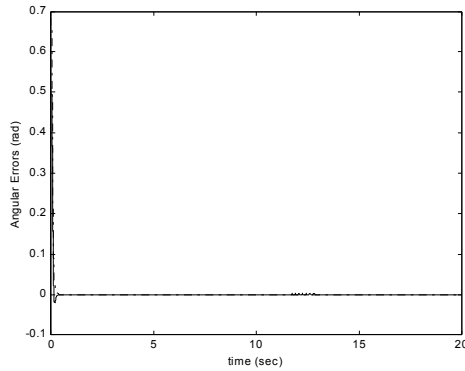
ŞEKIL 5.31. GNFSVC-payload



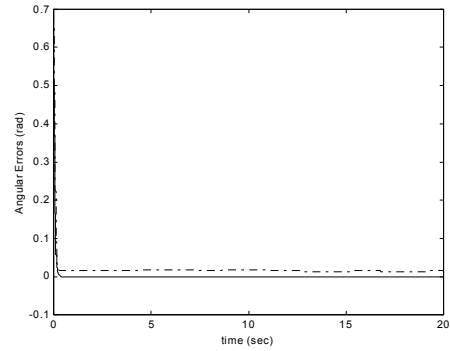
ŞEKIL 5.32. ANFIS



ŞEKIL 5.33. ANFIS Nif



ŞEKIL 5.34. ANFIS with payload



ŞEKIL 5.35. ANFIS Nif with payload

GNFSVC- Uc= GNFSVC hesaplamalı klasik kontrol

GNFSVC-Uc- not Boundry= GNFSVC de klasik kontrol parametreleri sınır değer hesaplamasız

GNFSVC-Uc- Boundry= GNFSVC de klasik kontrol parametreleri sınır değer hesaplamalı

GNFSVC-Uc- Boundry-payload= GNFSVC de klasik kontrol parametreleri sınır değer hesaplamalı ve sistemde ani yük değişimi var.

GNFVSC -not boundry -payload= GNFVSC de sınır değeri hesaplamasız ve sistemde ani yük değişimi var.

5.7 Test Yapıları

Yapılan Çalışma bölüm ikide anlatılan robot üzerinde gerçekleştirilmiş olup burada adı geçen ve bölüm üç ile bölüm 5 başında detayları verilen DIMA, FELA, Neuro-SMC, GNFVSC and ANFVSC yapay zeka metodları üzerinde yapılmıştır. Burada amaç belirlenen performans kriterleri (daha önce bahsedildi) üzerinde belirlenen sistemlerin çeşitli durumlarda ki bunlardan en önemlilerinden biri de endüstriyel ortamlarda çok geçerli olan ani yük değişimi e kolay uygulanabilirlik karşılaştırmalarının yapılmasıdır.

Buradaki bütün testlerde hata ve haza hızlı vektörleri zeki kontrolöre uygulanmış ve gelen cevaba göre gerekli kontrol sinyali uygulanmıştır. Kullanılan öğrenme algoritması gradient decent tir.

Testlerdeki RBFNN yapısında 8 girişli, 9 saklı katmanlı (hidden neurons) ve 2 lineer çıkışlı bir sistem kullanılmıştır. Bulanık kontrol için 13 üyelik fonksiyonu belirlemiş ve çalışma esnasında bu üyelik fonksiyonları güncellenmiştir.

Denklem (5.67) ve (5.68) de belirtilen ve Şekil 5.16 ve Şekil 5.17 de gösterilen referans sistemi bütün testlerde uygulanmış olup testler 0 hata ile başlatılmıştır.

GNFVSC kontrolör için yapılan testlerde metodun çeşitli özellikleri (mesela kalik kontrol veya zeki kontrol ile düzeltici kontrol sinyalinin bulunması gibi) uygulanmış ve GNFVSC kullanımına yeni yaklaşımlar geliştirilmiştir. Test sonuçları Şekil 5.24 ile Şekil 5.31 de verilmiş ve test karşılaştırmaları da çizelge 2 de yapılmıştır.

Diğer bir zeki kontrol sistemi olan ANFVSC de 3 üyelik fonksiyonu (her robot eklemi için pozisyon ve hız hataları için negatif, sıfır ve pozitif belirleyen) ve her üyelik fonksiyonu çalışma anında güncellenmiştir. Burada da aynı referanslar kullanılmış ve test 0 hat aile başlatılmıştır Test sonuçları Şekil 5.32 ile Şekil 5.35 de verilmiş ve test karşılaştırmaları da çizelge 2 de yapılmıştır.

Çizelge 5.1 GNFVSC ve ANFVSC metotlarının karşılaştırma tablosu.

G N F V S C		RMS ₁	RMS ₂	Peak ₁	Peak ₂
	With boundary learning.	0.0029	4.6209e-004	0.0755	0.0143
	Without boundary learning.	0.0028	5.0043e-004	0.0755	0.0143
	Payload change with boundary learning.	0.0030	5.2248e-004	0.0755	0.0143
	Payload change without boundary learning.	0.0030	9.2421e-004	0.2521	0.0145
	With classical Computation of U _c .	0.0031	4.4912e-004	0.0755	0.0144
	With classical Computation of U _c with payload.	0.0031	4.6009e-004	0.0755	0.0145
	With classical Computation of U _c without boundry learning.	0.0031	0.0012	0.0755	0.0144
	With classical Computation of U _c without boundary learning with payload.	0.0032	0.0014	0.0755	0.0145
A N F V S C					
	With Payload	0.0284	0.0409	0.5139	0.6843
	Without Payload	0.0281	0.0406	0.5138	0.6843
	NIF	0.0280	0.0377	0.5138	0.6843
	NIF with payload	0.0280	0.0377	0.5138	0.6843

Şekil 5.18 ile şekil 5.35 test sonuçlarını gösteren temel grafikleri göstermektedir. ANFVSC ile GNFVSC karşılaştırılması ani yük değişikliği altında ve yük değişikliği olmadan karşılaştırılması ve Çizilge 5.1 de gösterildiği gibi GNFVSC her durumda ANFVSC den daha iyi sonuç vermektedir.

Çizelge 5.2 DIMA, FELA ve NEURO SMC karşılaştırma tablosu

Methods	DIMA with RBF	FELA	Neuro SMC
Computation Time	20	X	X
Necessary Knowledge	X	Qualitative priori knowledge	X
Convergence Time	1200 epochs	1200	1200
$RMS(\theta_1)$	0.0068	0.0085	0.0061
$RMS(\theta_2)$	0.0022	0.0058	0.0011
$Peak(\theta_1)$	0.0755	0.0755	0.0754
$Peak(\theta_2)$	0.0145	0.0145	0.0140
After Payload Change			
$RMS(\theta_1)$	0.0150	0.0116	0.086
$RMS(\theta_2)$	0.0053	0.0059	0.0043
$Peak(\theta_1)$	0.0755	0.0755	0.0343
$Peak(\theta_2)$	0.0231	0.0157	0.0157
Training Method	Off-line	On-line	On-line
Parameters	NN structure		NN structure

Karşılaştırması yapılan diğer metodlar olan DIMA(RBF), FELA, Neuro-SMC arasındaki karşılaştırmada hem ani yük değişikliği altında, hem de ani yük değişikliği olmadan Neuro-SMC daha iyi sonuçlar vermektedir. Neuro-SMC ayrıca system ile ilgili önceden bir bilgiye ihtiyaç duymamaktadır ve öğrenme algoritması online dır.

5.8 Sonuç

Bu test çalışması literatürde pek çok uygulaması olan zeki kontrol tasarımının bazı özel ve gerekli koşullarda (mesela endüstriyel ortamdaki ani yük değişikliği) yada uygulanabilir olması için kolay tasarlanabilir olması gibi konuları bu kadar çok ve bilinen metotlar üzerinde ilk defa irdeliyor olması bakımından önemlidir. Bu metotlar literatürde pek çok kere kullanılmış olmaları sebebiyle iyi tanınan yöntemlerdir.

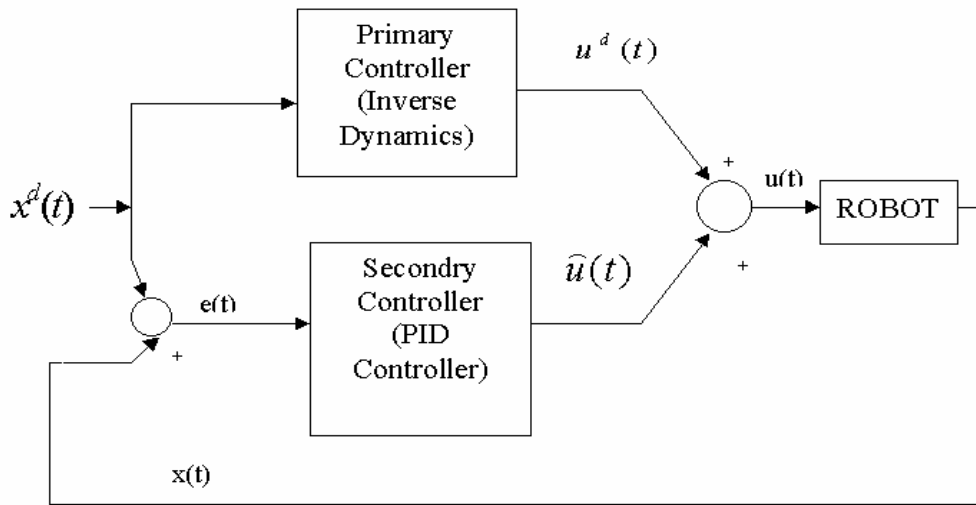
Bu çalışmadan ortaya çıkan sonuç GNFVSC ve ANFVSC hedef takibinde ani yük değişikliği varken ve yokken en iyi sonucu vermektedir Diğer taraftan DIMA - RBF ve FELA en basit tasarlanabilir metotlar olarak öne çıkmaktadırlar. Diğer önemli bir kriter olan tasarım için sistemin önceden iyi bir şekilde tanınmasına göre FELA hariç diğer bütün metotlar iyi sonuç vermektedir.

Bu çalışmanın sonucuna göre GNFVSC kontrol uygulamaları için çok iyi sonuç vermektedir. Burada seçilen karşılaştırma kriterleri pek çok uygulamada ortak kriterler olduğu düşünülürse çizelge 5.1 ve çizelge 5.2 çeşitli endüstriyel ve akademik çalışmalar için zeki kontrol sistemi tasarımı seçimi için kullanılabilir.

6. DİNAMİK BİR YÜK DEĞİŞİMİ ALTINDA LİNEER OLMAYAN BİR SİSTEMİN KONTROLÖR DİZAYNI

Robot kontrolü problemi aslında bir servo problemidir. İstenen iş robotun eklemlerinin (yada çıktısını) istenen referansı takip edebilmesini sağlamaktır. Robotun giriş sinyalleri aktuatör (actuators) ler tarafından üretilen genelleştirilmiş (generalized) kuvvetler ve robot çıktısı da konum, hız ve ivme üçlüsüdür.. Robotun kolunun gerçek konum hız ve ivme bilgileri encoders ve takamotre vasıtası ile ölçülür. İvme de bir yazılım ve bilgisayarlar vasıtası ile hesaplanabilir bir değerdir.

Robotu yönlendirmek için durum, hız ve ivme hataları yapısı için geribesleme lobları kullanılarak bu değerler ile istenen değerler karşılaştırılabilir. Bu değerlerin oluşturduğu Kontrolörden, ilk Kontrolör olarak söz edilir. Böylece lineer olmayan etkileri karşılar, ve modeldeki lineer olmayan değerleri iptal etmeye çalışır. Kullanılan matematiksel model genellikle doğru değildir ve sistem karışıklıklar için bağımlıdır. Nominal yörüngeden kaynaklanan asıl hareketin istenmeyen hataları ikincil Kontrolör olarak adlandırılan ilave Kontrolör tarafından doğrulanır.



Şekil 6.1. Birinci ve İkincil Kontrolör ile Sistem

6.1 Ters Dinamikler Kullanılarak Klasik Kontrolör Dizaynı

Linner olmayan modelde birincil Kontrolör saptamak için geribesleme-besleme İleri Lineerazisyon tekniğini kullanacağız. Sonra ikincil Kontrolör gibi bir PD Kontrolör'ü oluşturacağız ve faydalarını belirleyeceğiz.

Robot :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + f_c = \tau \quad (6.1)$$

Formülde ;

$M(q)$: (nxn) tembellik matriksi ,

$C(q, \dot{q})$: (nxn) Coriolis terimleri,

τ : (nx1) dönme momenti vektörü,

$q, \dot{q}, \ddot{q}$: (nx1) durum, hız, ivme vektörleri,

$q = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_n]^T$ ve $\theta_i, i=1, \dots, n$: eklem açıları ,

f_c : (nx1) Kolomb (Coulomb) sürtünme gücü .

(6.1)'deki model aşağıda gösterildiği şekliyle durum-aralık'da yazılabilir,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -M^{-1}(Cx_2 + f_c) \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix} u \quad (6.2)$$

veya

$$\dot{X}(t) = f(x) + B(x)U(t) \quad (6.3)$$

formülde,

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} q & \dot{q} \end{bmatrix}^T \quad \text{and} \quad u = \tau$$

Birincil kontroler açıkça belirtilmiştir. Böylece sistem denklemini geribesleme-besleme ileri linearizasyon tekniği ile linearize edilecektir. $U^d(t)$ ’i seçebiliriz $(X^d(t), U^d(t))$ gibi. (6.4) denkleminde karşılar;

$$\dot{X}^d = f(x^d) + B(X^d)U^d(t) \quad (6.4)$$

$B(x)$ ’nin tersi tembellik matriksidir, onun tersi her zaman bulunur (vardır). Ve denklem (6.5)’de $U^d(t)$ için çözülebilir. Denklemin çözümü birincil Kontrolörde açıkça belirtilir.

$$U^d(t) = [B(x)]^{-1} [\dot{X}^d(t) - f(x^d)] \quad (6.5)$$

6.1.1 İkincil Kontrolör Dizaynı

Dahili modelleme hatalarının ve rahatsızlıklarının etkilerini karşılamak için , sisteme bir ikincil Kontrolör sokulur. Nominal bir yörünge hakkında sistemin verdiği yanıtta küçük sapmaları ikincil kontrolör düzenleyecektir. Bu nedenle prespective yörünge, 6.3 denkleminde lineerleme sayesinde elde edilen temel bir modelde dizayn edilebilir.

Bir PID-kontrolör yükseltme operasyonlarını yerine getirir (P=Orantılı), (I=Integral), (D=Ayrımsal). Genelde , D-bölümü cevap hızını artırır. Kontrolörün her bir bölümü uyumlu olmak için fayda sağlar böylece istenen sistem elde edilir. Bir PID-kontrolörün faydaları eigenvalue ayrımının temelinde de saptanabilir.

Eğer $f(x) = f(x^d)$ ve $B(x) = B(x^d)$ ise , birincil Kontrolör lineer olmayan terimlerin kesin iptaline neden olur. Sonuçlandırma modeli bir ikincil Kontrolör $\hat{u}(t)$ dizaynı için spesifik lineer formda yapılabilir.

Bir PD-Kontrolör seçelim :

$$\hat{u}(t) = [B(x)]^{-1} [-K_v(\dot{x} - \dot{x}^d) - K_p(x - x^d)] \quad (6.6)$$

Denklemden belirtilmiş olan K_v , ve K_p değişmez matrikslerdir.

Şekil 6.1’de gösterildiği gibi olmalıdır. Sistem için girdi, Kontrolör’ün her ikisinin çıktılarının toplamıdır. Bu nedenle sistem için girdi aşağıdaki formda yazılabilir:

$$\hat{u}(t) = u^d(t) + \hat{u}(t) = [B(x)]^{-1} [\dot{x}^d - f(x^d) - K_v(\dot{x} - \dot{x}^d) - K_p(x - x^d)] \quad (6.7)$$

Eğer denklem 6.3’ün içine denklem 6.7 yerleştirirsek

$$\dot{x}(t) = f(x) + B(x)[B(x)]^{-1} [\dot{x}^d - f(x^d) - K_v(\dot{x} - \dot{x}^d) - K_p(x - x^d)] \quad (6.8)$$

$$\dot{x}^d - f(x^d) = -K_v(\dot{x} - \dot{x}^d) - K_p(x - x^d) \quad (6.9)$$

Aşağıdaki gibi bir hata tanımlanır,

$$e(t) = x(t) - x^d(t) \quad (6.10)$$

$$(I + K_v) \dot{e}(t) + K_p e(t) = 0 \quad (6.11)$$

Denklem 6.11 ‘ün Laplace dönüşümü olur:

$$[(I + K_v)s + K_p]E(s) = 0 \quad (6.12)$$

O, K_p ve K_v seçmek için çapraz matriks gibi olagandır. Denklem 6.12 de bir çözüm bulabilmek için matriksin determinantı $[(I + K_v)s + K_p]$ sifıra eşit olmalıdır.

$$\det \left(\begin{bmatrix} K_v^1 + 1 & 0 & 0 \\ 0 & K_v^2 + 1 & 0 \\ 0 & 0 & K_v^3 + 1 \end{bmatrix} s + \begin{bmatrix} K_p^1 & 0 & 0 \\ 0 & K_p^2 & 0 \\ 0 & 0 & K_p^3 \end{bmatrix} \right) = 0 \quad (6.13)$$

Sistemi sabit (durağan) yapmak için, tasarımcı 6.13 denklemin köklerini yerleştirmelidir, Örneğin sol kontrol düzlemi için karakteristik denklemin eigenvalues'ı. Bu yüzden dizaynır aşağıdaki gibi parametreler seçmek zorundadır:

$$\frac{K_p^1}{K_v^1} > 0 \quad \frac{K_p^2}{K_v^2 + 1} > 0 \quad \frac{K_p^3}{K_v^3 + 1} > 0$$

6.2 Zeki PD Kontrolör

6.2.1 Direkt Ters Modelleme Mimarisi (DIMA)

Karmaşık bir sistemde varyansa adapte olabilen tek kontrol aracıdır. Uyum kontrolünden fayda sağlayan bir sistem robottur, değişebilir olması veya tam doğasının bilinmemesi.(boy, kütle, kütle dağılımı, karışıklıkları). Bu bölüm RBF yapay ağ öğrenme kontrol sisteminin bir uygulamasıdır. Uyarlanmış sinyaller ile eğitilmiş kontrol ağı klasik PD kontrolör ile robotu dengede tutmakta kullanılır. İlk eğitim , ağ PD gibi diğer kontrolör olmaksızın robotu dengede tutacaktır.

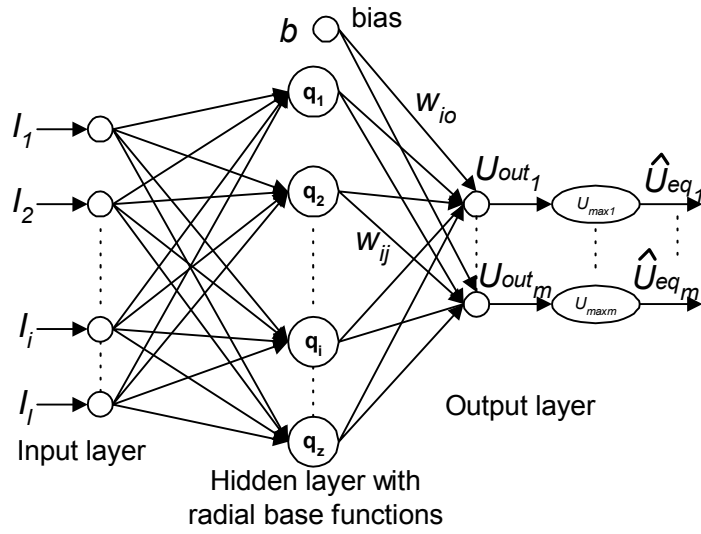
RBFFNN yapısı Şekil 6.2 'de gösterilmiştir ve 3 katmandan oluşmaktadır.İlk katman istenen ve gerçek durum için bağlantı katmanıdır.İkinci katman sadece gizli katmandır.Girdi vektör aralığından iç vektör aralığına linner olmayan dönüşümü sağlar.Son katman çıktı katmanıdır lineer tarzda bir çıktıyı iç vektör arlığına dönüştürür.Gizlik katmandaki düğümler gerçek taban fonksiyonlarıdır. Bu fonksiyonların tipi teoride önemsizdir ve genellikle aşağıdaki gibi Gaussian fonksiyonları kullanılır,

$$q_j(I) = e^{-\frac{1}{2} \frac{\|I - c_j\|^2}{\lambda_j^2}} \quad \lambda_j > 0 \quad j = 1, 2, \dots, z \quad (6.14)$$

Denklemde I bir çokluboyut girdi vektörüdür, c_j aynı boyutlu RBF'in merkezinin bir vektörüdür I gibi , λ_j of RBF'nin genişliğidir, z RBF'in sayısıdır, ve $q_j(\cdot)$ ith

RBF orijindeki tek maksimumdur. Çıktı katmanındaki Neuronlar lineer fonksiyon hesaplar.

$$f_i(I) = U_{out_i} = w_{i0}b + \sum_{j=1}^z w_{ij}q_j(I) \quad i = 1,..m \quad (6.15)$$



Şekil. 6.2 RBFNN Yapısı

Eğitim için hazırlanan ağ yapısı :

Düşünülen (önerilen) yapay zeka mimarisi

katman: 3 (1 gizli)

Düğüm sayısı: 6-5-2

Eğitim methodu: RBF

6.2.2 Yapay Kayan Modda Kontrol

Lineer olmayan süreçlerin kontrolü için, çeşitli yapıdaki sistem teorilerinin uygulanmasında , sistem kayan modda ve herhangi bir modelleme kararsızlıklarından ve /veya rahatsızlıklarından etkilenmeyecekse

Kayan Yüzey:

Genelde verilen sistemler için, kayan yüzey , S , (mx1) aşağıda verildiği gibi seçilir,

$$S(X, t) = G(X^d(t) - X(t)) = \phi(t) - S_a(X) \quad (6.16)$$

Denklemden,

$$\phi(t) = GX^d(t), \quad S_a(X) = GX(t) \quad (6.17)$$

zaman ve durum bağımlı bölümdür. Ve de X^d istenen durum vektörünü gösterir ve G (mxn) kayan yüzeyin eğimli matriksidir. Genellikle, seçilen G matriksi kayan yüzey fonksiyonu gibi olur,

$$S_i = \left(\frac{d}{dt} + \lambda_i \right)^{k_i-1} e_i \quad (6.18)$$

Denklemden e_i , x_i ($e_i = x_i^d - x_i$) için bir hatadır. Ve de λ_i 's pozitif değişmez gibi seçilir. Bu yüzden S_i sıfıra eşitken e_i sıfıra gider.

Kayan kipili kontrolde'de amaç kayan yüzeyde sistem durumunu güçlendirmektir. Kayan yüzeydeki durumlardan biri, matrix G tarafından dikte edilmiş bir hata dinamiği ile sıfır için sistem hatalarının bir noktada birleşmesidir.

Kayan mod kontrolörlerinin uygulamasında ana problem; eşdeğer kontrolün hesaplanabilir olması için gerekli olan sistem parametreleri ve sistem dinamiklerinin (veya ters dinamiklerin) tüm bilgisinin karşılaştırılmasıdır. Bu aslında pratikte çok nadiren olur. Geri beslemeli yapay zeka [Hung J. Y]'de eşdeğer kontrolün hesaplamasında önerilir. Güncellenmiş ağırlığı, kayan moda ilave kontrol teriminin sıfıra gitmesi gibidir. Gradient decent metodu ağırlık adaptasyonunda kullanılır.

Kontrolör:

Bu bölümde tanımlanan metod Lyapunov fonksiyonunun seçimi temellidir.

Farzedelim ki S 'nin V gibi scalar bir fonksiyonu var, ilk düzendeki gibi türetilmiş olarak devam ediyor;

$V(S)$ pozitif belirlidir,

$\frac{dV(S)}{dt}$ negatif belirlidir,

$$V(S) \rightarrow \infty \text{ as } \|S\| \rightarrow \infty$$

(6.19)

Sonra $S=0$ 'da denge ayrıntılı olarak [Utkin, V.I-1981] uygun.

İlk olarak , bir Lyapunov fonksiyonu aşağıda verildiği gibi seçilir,

$$V(S) = \frac{S^T S}{2} \quad (6.20)$$

D ($m \times m$) pozitif belirli çapraz kazanç matrisidir. Signum fonksiyonu anlamına gelen $\text{Sign}(S)$ S 'nin her bir elementi için eklenir,

Denklemden,

$$\frac{dS}{dt} = \frac{d\phi}{dt} - \frac{\partial}{\partial X} \frac{S_a}{dt} \frac{dX}{dt} = \frac{d\phi}{dt} - G(F(X) + BU) \quad (6.21)$$

Ve de X^d istenen (referans) durum vektörünü gösterir ve kayan yüzeyin eşdeğer kontrolünün G ($m \times n$) eğim matrisi ;

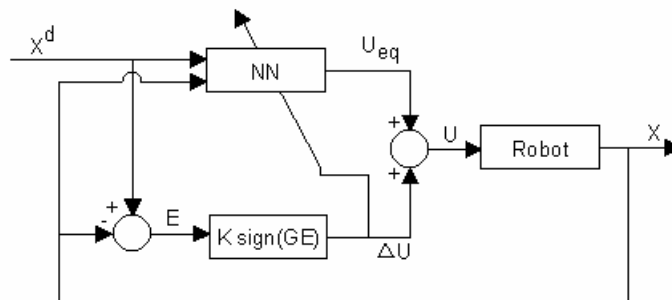
$$U_{eq}(t) = -(GB)^{-1} \left(GF(X) - \frac{d\phi(t)}{dt} \right) \quad (6.22)$$

Ve ilave kontrol terimi ,

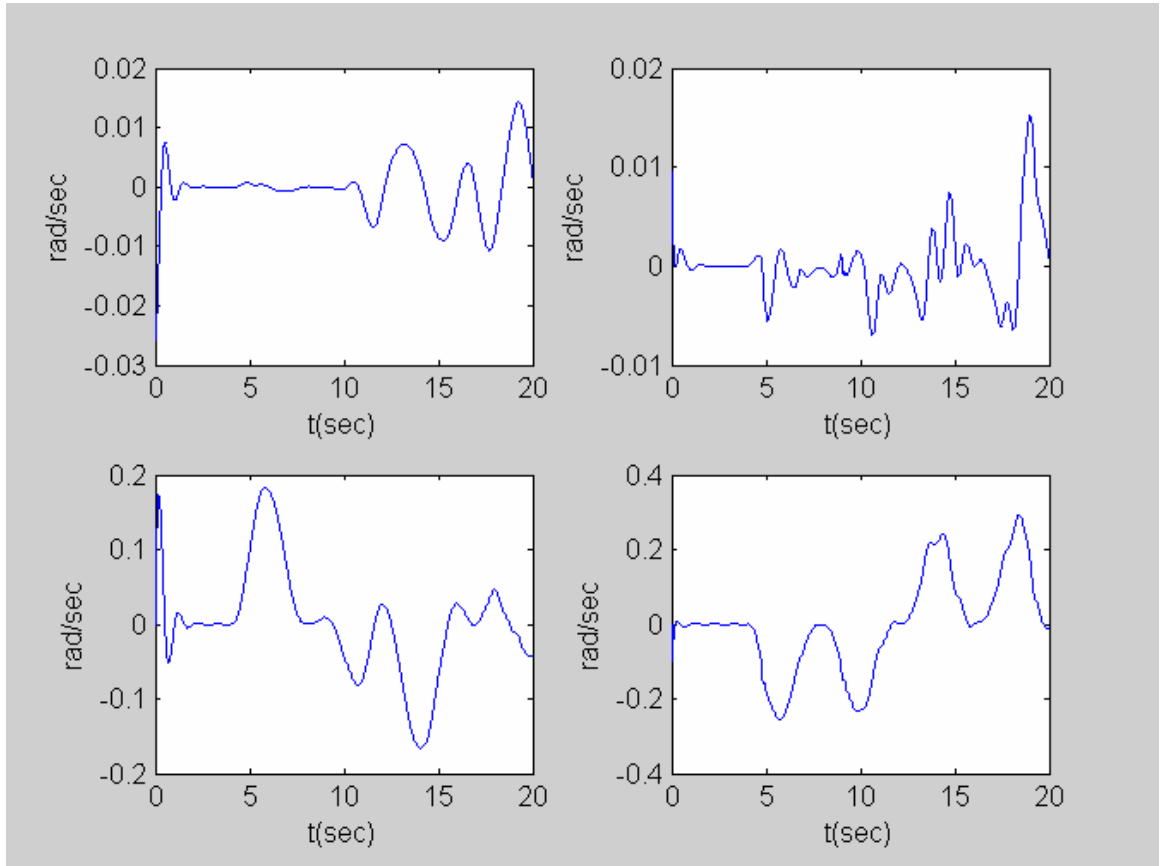
$$\Delta U(t) = (GB)^{-1} D \text{sign}(S) = K \text{sign}(S) \quad (6.23)$$

Toplam kontrol sinyali;

$$U(t) = U_{eq}(t) + \Delta U(t) \quad (6.24)$$



Şekil 6.3. Yapay Kayan Modda Kontrol



Şekil 6.4. İstenen ve Ölçülen açısal hız değerleri

6.3 PD ve Yapay-PD Kontrolörlerin Karşılaştırılması

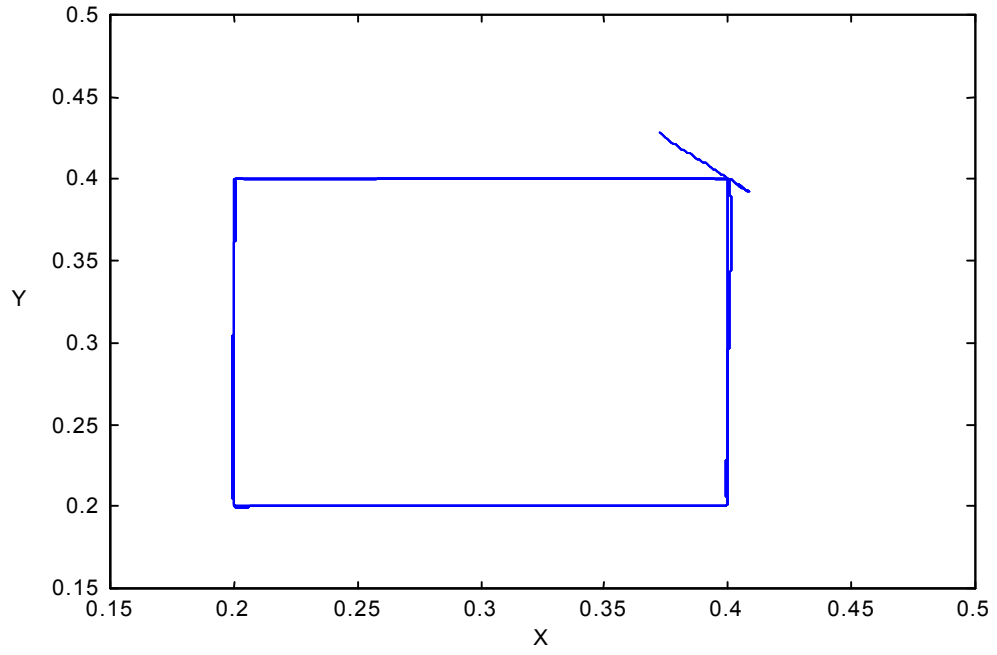
Önceki bölümde niçin bir Yapay- PD kontrolörünü dizayn ettiğimizin nedenini anlattık. Bir kontrolörün en önemli özelliklerden birisi uyarlanabilir olmasıdır. Bizim amacımız uyarlanabilir bir kontrolör dizayn etmektir. Daha önce yük değişim ile simüle ettik ve Neuro-PD ile klasik PD kontrolörden daha iyi bir sonuç elde ettik. Simülasyon için tüm kontrol parametreleri ilk bölüm parametreleri ile aynıdır. Simülasyon sıfırcı saniyede 0 yükü başlar, 10. saniyeye kadar devam eder ve 40 kg'da PD ve Neuro-PD kontrolörlerin her ikisi içinde ek bir yük uygulanır ve sonuçlar elde edilir.

Diğer simülasyon 0. saniyede sıfır yükle başlar ve 20. saniyeye kadar devam eder ve sonra 40 kg' da PD ve Neuro-PD kontrolörler için ek bir yük uygulanır ve sonuçlar elde edilir.

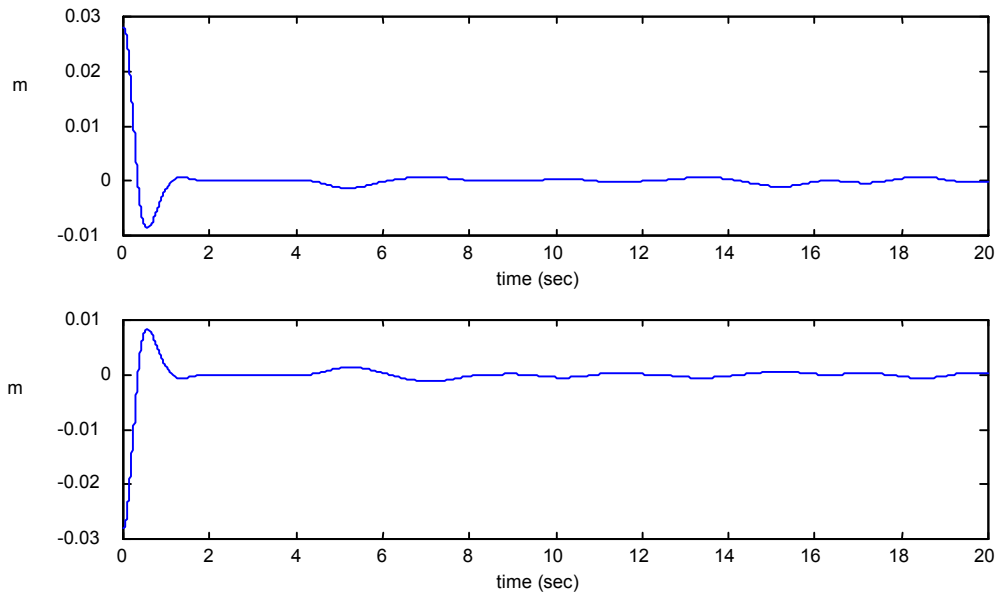
Çizelge 6.1 DIMA, PID ve NEURO SMC karşılaştırma tablosu

Methodlar	PID	DIMA Feed-forward ağ	DIMARBF	NEUROSMC
$RMS(\theta_1)$	0.0049	0.0068	0.0068	0.0023
$RMS(\theta_2)$	0.0017	0.0057	0.0022	0.0014
$Peak(\theta_1)$	0.0755	0.0755	0.0755	0.0755
$Peak(\theta_2)$	0.0145	0.0229	0.0145	0.0096
Sonraki Yük Değişimi				
$RMS(\theta_1)$	0.0147	0.0155	0.0150	0.0051
$RMS(\theta_2)$	0.0066	0.0077	0.0053	0.0035
$Peak(\theta_1)$	0.0755	0.0755	0.0755	0.0755
$Peak(\theta_2)$	0.0182	0.0229	0.0231	0.0153

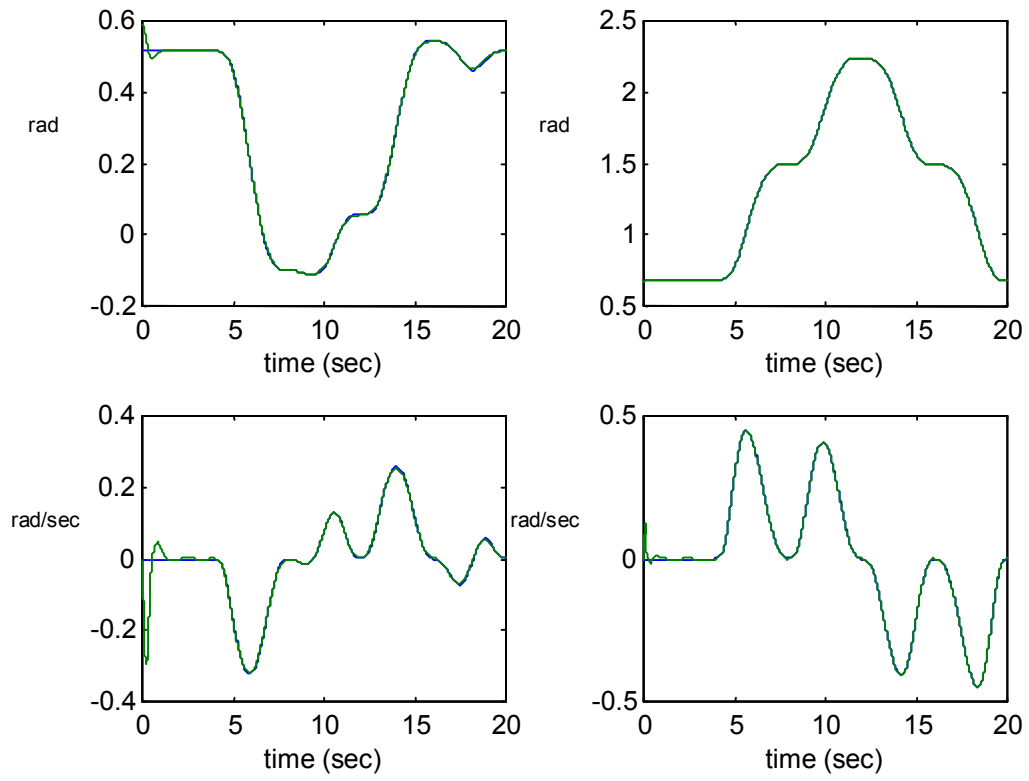
Çizelge 6.1 de PID ile DIMA (ff), DIMA (RBF) ve Neuro-SMC karşılaştırması yapılmıştır. Belirlenen kıstaslara göre Neuro-SMC daha iyi sonuçlar vermektedir. Özellikle ani yük değişikliği altında NEUROSMC daha iyi sonuçlar vermektedir.



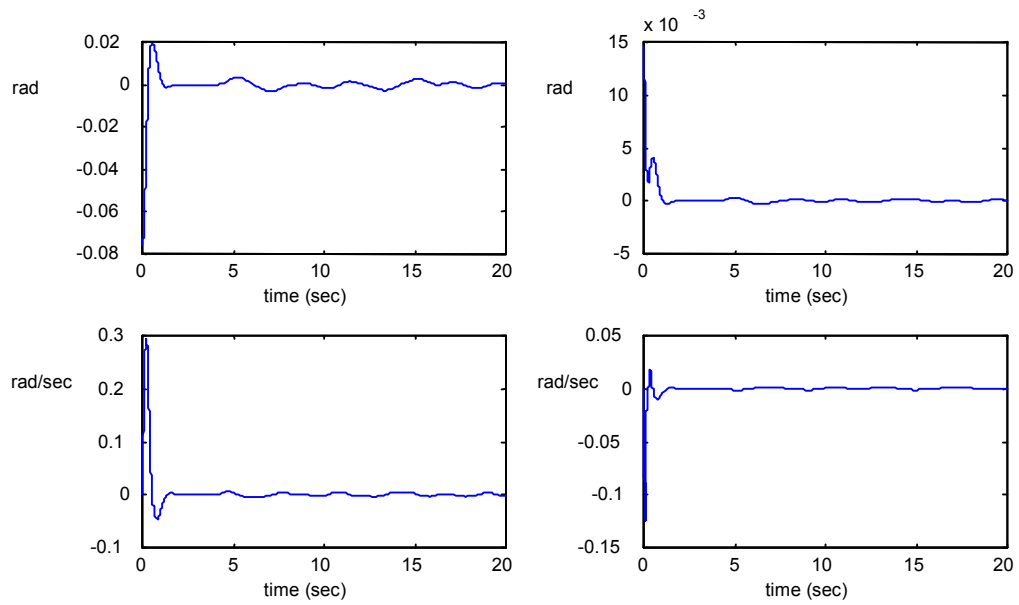
Şekil 6.5. X-Y koordinatlarındaki hareket.
(Neuro-PD controller)



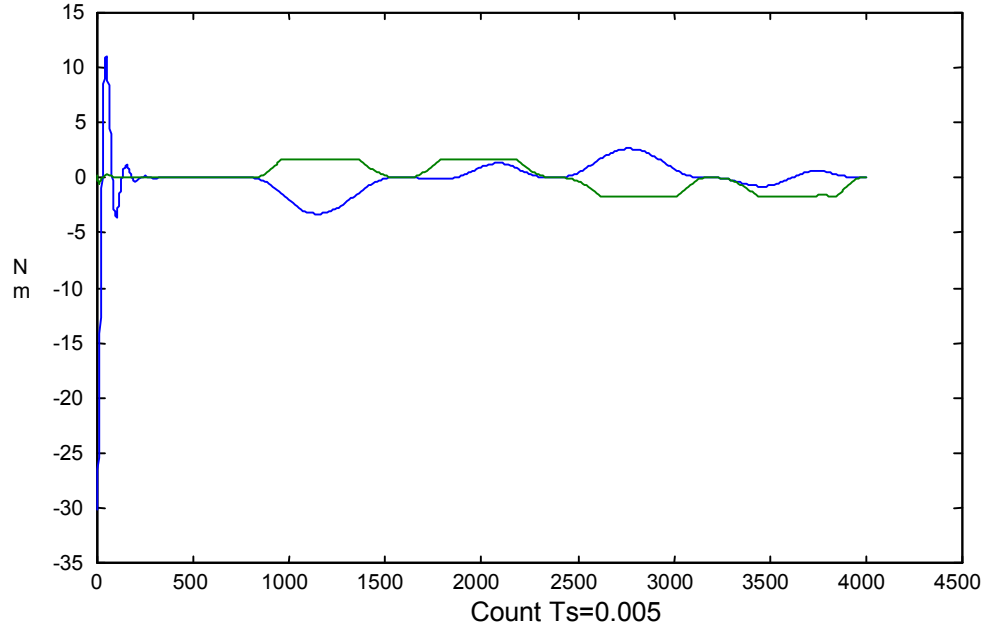
Şekil 6.6 X-Y koordinat hataları.
(Neuro-PD controller)



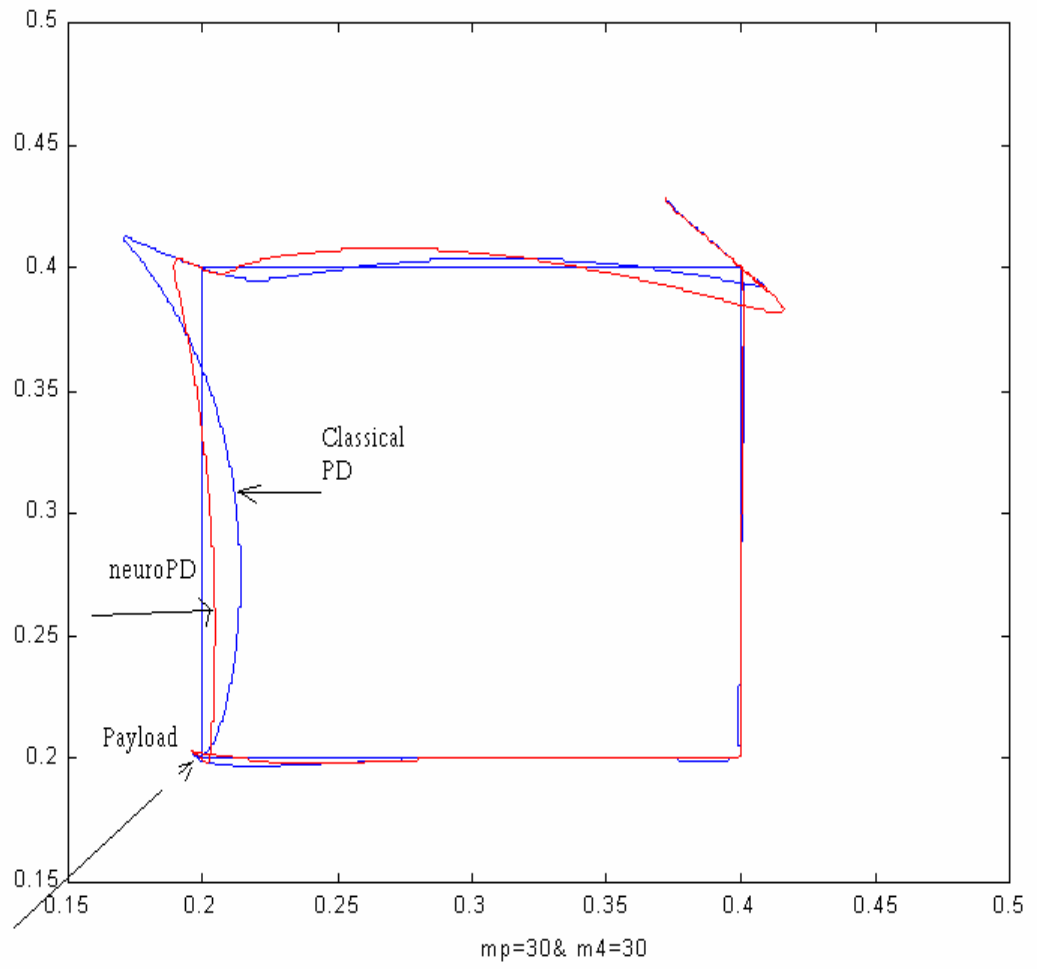
Şekil 6.7. Ölçülen ve istenen açılar ve açısal hızlar.
(Neuro-PD Kontrolör)



Şekil 6.8. Açık ve Açısal hata.
(Neuro-PD Kontrolör)

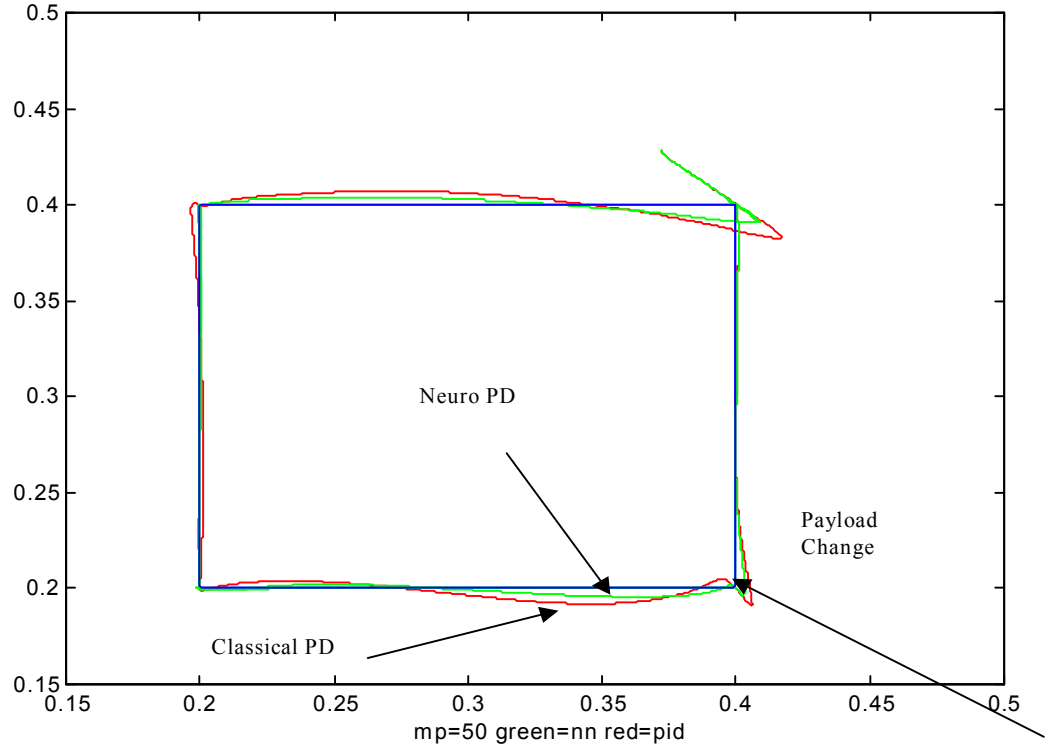


Şekil 6.9. Robota uygulanacak olan kontrol kuvveti.
(Neuro-PD Kontrolör)



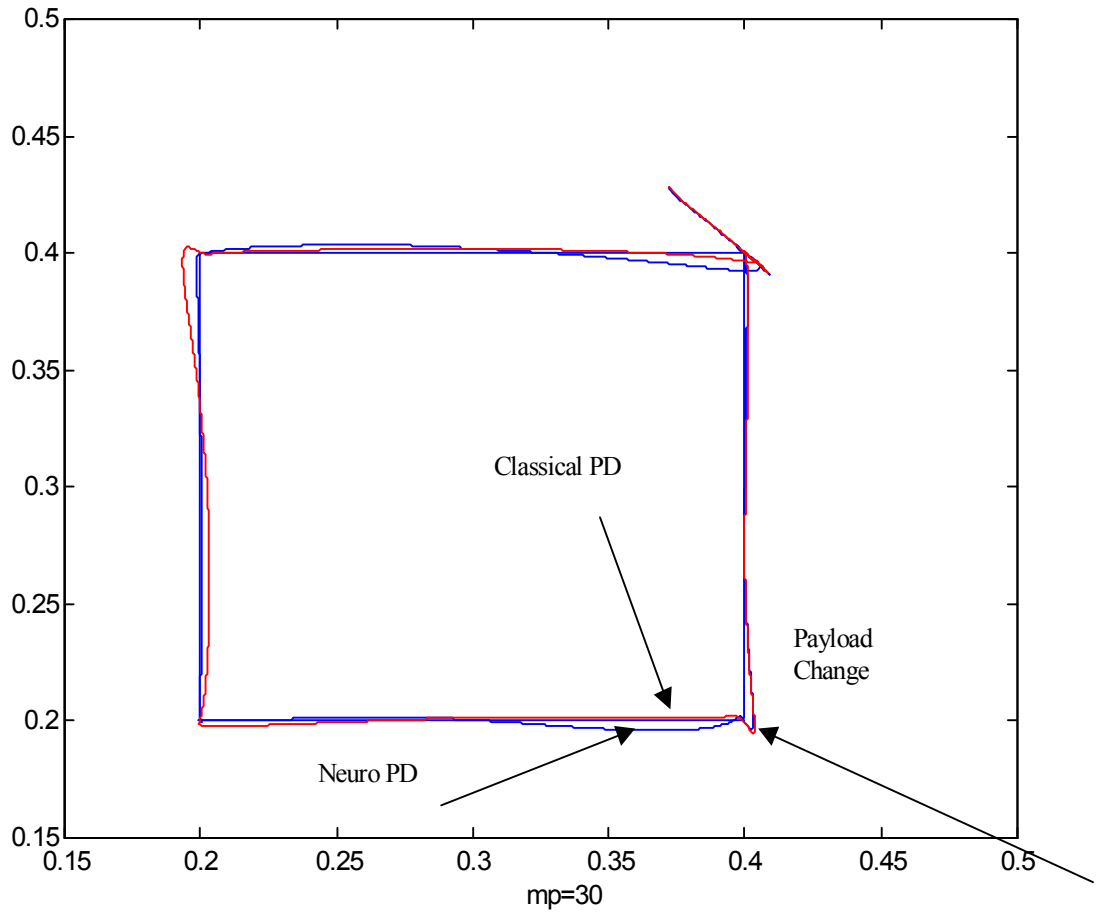
Şekil 6.10. PD ve Neuro-PD'in Karşılaştırılması

Tasarlanan Neuro PD yöntemi ani yük değişikliği altında takip etmesi gereken X-Y koordinatlarından daha az uzaklaşarak daha kararlı olduğu görülmüştür.



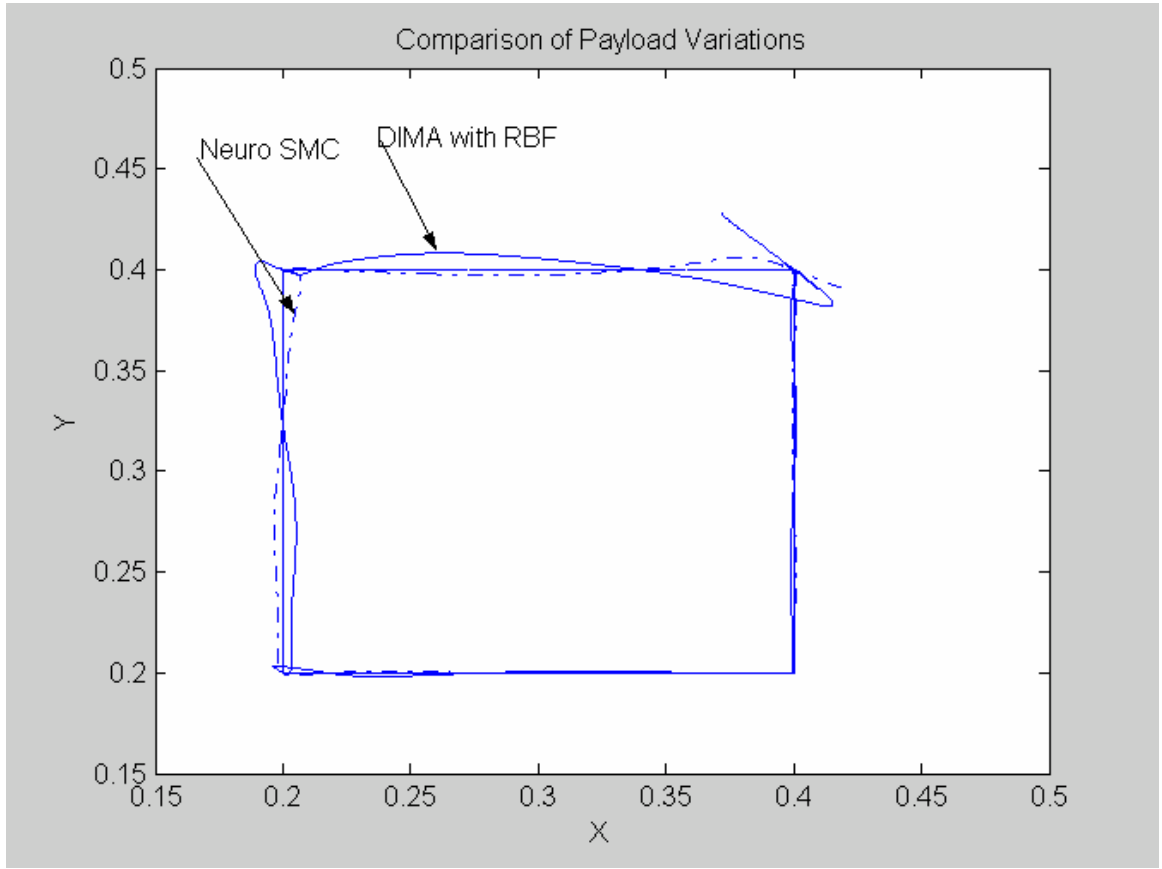
Şekil 6.11. PD ve Neuro-PD'nin Karşılaştırılması

Tasarlanan Neuro PD yöntemi ani yük değişikliği altında takip etmesi gereken X-Y koordinatlarından 10. saniyede uygulanan 50 kg lık ani yük eklenmesi durumunda daha az uzaklaşarak daha kararlı olduğu görülmüştür.



Şekil 6.12. PD ve Neuro-PD'nin Karşılaştırılması

Tasarlanan Neuro PD yöntemi ani yük değişikliği altında takip etmesi gereken X-Y koordinatlarından 10. saniyede uygulanan 30 kg lık ani yük eklenmesi durumunda daha az uzaklaşarak daha kararlı olduğu görülmüştür.



Şekil 6.13. Neuro SMC ve RBF DIMA'nin Karşılaştırılması

Şekil 6.13 Tasarlanan Neuro –SMC ile DIMA (RBF) yöntemleri ani yük değişikliği altında takip etmesi gereken X-Y koordinatlarından 20. saniyede uygulanan 50 kg lık ani yük eklenmesi durumunda daha az uzaklaşarak daha kararlı olduğu görülmüştür.

7.SONUÇ VE ÖNERİLER

Bu çalışmada endüstriyel ortamlarda ani yük değişikliğine dayalı bir control algırtması tasarlanmıştır.Bu tasarıma ulaşmak için literatürde daha önce yapılmış çalışmalar analiz edilmiş ve istenilen çalışma şartlarında performas kriterleri ve bunlara göre yapılan çalışmaların avantajlı tarafları tesbit edilmiştir.

Bu çalışmada analiz edilen yöntemler:

- 1-Direct Ters Kontrol (Direct inverse modeling architecture with radial basis functions (DIMA)),
- 2-Geri Beslemeli-Hata Öğrenmeli Kontrol (Feedback error learning architecture(FELA)),
- 3-Kayan Kipli Kontrol (Neuro-sliding mode control),
- 4-Gaussian Yapay Bulanık Kayan Kipli Kontrol (Gaussian neuro fuzzy variable structure control)
- 5-Adaptive Yapay Bulanık Çıkarım Sistemi (Adaptive neuro fuzzy inference system(ANFIS)).
- 6- Kayan Kipli Denetim (Neuro- Sliding Mode Architecture (NESMA))

Bu yöntemlerin analizi ve performans karşılaştırması için öncelikle bu parametreler tesbit edilip daha sonra bu parametreler üzerinde bir karşılaşma sonucuna gidilmiştir.

Karşılaştırmalar RMS ve Peak olarak gösterilen ve eklem düzlemindeki hataların toplamı ve verilen referans yörüngeyi takip ederken yöründegen sapmayı ölçen peak değerleriyle kıyaslanmıştır.Bu hesaplar denklem 7.1, denklem 7.2 ve denklem 7.3 de gösterilmiştir.

$$|error_i| = \sqrt{\sum_{j=1}^n (q_i(j) - q_i^d(j))^2} \quad (7.1)$$

Max hata tanımı denklem 7.2 de gösterilmiştir

$$\max error_i = \max_{1 \leq j \leq k} |q_i(j) - q_i^d(j)| \quad (7.2)$$

Ve kartezyen koordinatlardaki hata tanımı ki bizim elde etmek istediğimiz bu denklemi minimize etmek, denklem 7.3 de gösterilmiştir.

$$|error_{taskspace}| = \sqrt{\sum_j \left(\left\| \begin{pmatrix} x(j) - x^d(j) \\ y(j) - y^d(j) \end{pmatrix} \right\|^2 \right)} \quad (7.3)$$

Karşılaştırmalar yapılırken sadece yukarıdaki parametreler değil aynı zamanda

1-(Hesap Yönetim)Computation

2-(Tasarımdan önceki gerekli bilgi) Necessary Knowledge

3-(Yakınsama zamanı) Convergence Time

4-Öğrenme Algoritması

gibi parametreler de göz önüne alınıp bunlar da karşılaştırma tablolarında gösterilmiştir.

Bu test çalışması literatürde pek çok uygulaması olan zeki kontrol tasarımın bazı özel ve gerekli koşullarda (mesela endüstriyel ortamdaki ani yük değişikliği) yada uygulanabilir olması için kolay tasarlanabilir olması gibi konuları bu kadar çok ve bilinen metotlar üzerinde ilk defa irdeliyor olması bakımından önemlidir. Bu metotlar literatürde pek çok kere kullanılmış olmaları sebebiyle iyi tanınan yöntemlerdir.

Bu çalışmadan ortaya çıkan sonuç GNFVSC ve ANFVSC hedef takibinde ani yük değişikliği varken ve yokken en iyi sonucu vermektedir Diğer taraftan DIMA - RBF ve FELA en basit tasarlanabilir metotlar olarak öne çıkmaktadırlar. Diğer önemli bir kriter olan tasarım için sistemin önceden iyi bir şekilde tanınmasına göre FELA hariç diğer bütün metotlar iyi sonuç vermektedir.

Burada seçilen karşılaştırma kriterleri pek çok uygulamada ortak kriterler olduğu düşünülürse çizelge 5.1 ve çizelge 5.2 çeşitli endüstriyel ve akademik çalışmalar için zeki kontrol sistemi tasarımı seçimi için kullanılabilir

Bu karşılaştırmalar ışığında ani yük değişikliklerine karşı kararlı bir zeki kontrol sistemi tasarlanmıştır.

Bir kontrolörün en önemli özelliklerden birisi uyarlanabilir olmasıdır. Bu çalışmada tasarlanan Neuro-PD uyarlanabilir bir kontrolördür. Testlerde Neuro-PD ile klasik PD kontrolörden daha iyi bir sonuç elde ettik .

Çizelge 6.1 de tasarlanan Neuro-PD yönteminin diğer sistemler ile karşılaştırma analizi gösterilmiştir.Bu verilere göre tasarlanan sistem hem ani yük değişiklikleri altında, hem de değişiklik olmadan en iyi performansı vermektedir.Ayrıca yöntem tasarlanması kolay ve öğrenmesi online olması nedeniyle avantajlı bir algoritmadır.Şekil 6.10,6.11 ve 6.12 de gösterilen sonuçlar denklem 7.13 ün grafiksel gösterimi olup elde edilen sonuçların ne derece iyi olduğunu gösteren şekiller olması bakımından önemlidir.Bu şekiller denklem 7.13 ün görsel hali olması bakımıyla tasarlanan kontrolör performans ölçümünün rakamlar kullanmadan da yapılmasına ve yukarıda varılan ve tasarlanan yeni kontrol tekniğinin konuya uzak araştırmacılar tarafından da algılanmasına imkan vermektedir.

Bu konu ile ilgili sonraki çalışmalarda robot serbestlik derecesi değiştirilerek test ve deneyler yapılabilir. Ayrıca tasarlanan kontrol algoritmasının endüstriyel ortamdaki performansı analiz edilip bulunan sonuçlar kıyaslanabilir.

8. KAYNAKLAR

- [1] Fu, K.S., GONZALES, R.C., Lee C.S.G., Robotics Control, Sensing, Vision and Intelligence, McGraw Hill, 1987.
- [2] BRADY, M., HOLLERBACH, J.M., et al., Robot Motion: Planning and Control, The MIT Pres, London, 1983.
- [3] KOREN, Y., Robotics for Engineers, McGraw Hill, 1987.
- [4] Craig, I.J., Introduction to Robotics: Mechanics and Control, Addison-Wesley Pub. Comp., 1989.
- [5] Coiffet, P., Robot Technology: Modelling&Control, Vol:1, Prentice-Hall Inc., 1981.
- [6] Nof, S.Y., Handbook of Industrial Robotics, John Wiley&Sons, 1985.
- [7] Asada, H., Slotine, J.J.E., Robot Analysis and Control, John Wiley&Sons, 1986.
- [8] Koivo, A.J., Fundamentals for Control of Robotic Manipulators, John Wiley&Sons, 1989.
- [9] Critchlow, A.J., Introductions to Robotics, MacMillan Publishing Company, New York, 1985.
- [10] Ranky, P.G., Ho, C.Y., Robot Modelling Control and Applications with Software, IFT Pub. Ltd., 1985.
- [11] Spong J.J., Vidyasagar, M., Robot Dynamics and Control, John Willy&Sons, 1989.
- [12] Pillay, P., Krishnan, R., Application Characteristics of Permanent Magnet Synchronous and Brushless DC Motors for Servo Drives, IEEE Transactions on Industry Applications, Vol. 27, No. 5, pp. 986-996, September 1991.
- [13] Liu, T., Liu, C., A Multiprocessor-Based Fully Digital Control Architecture for Permanent Magnet Synchronous Motor Drives, IEEE Transactions on Power Electronics, Vol. 5, No. 4, pp. 413-422, October 1990.
- [14] Sen, P.C., Electric Motor Drives and Control Past, Present and Future, IEEE Transactions on Industrial Electronics, Vol. 37, No. 6, pp. 562-575, December 1990.

- [15] Sebestian, T., Gorden, R.S. and Rahman, M.A., Modelling of Permanent Magnet Synchronous Motor, IEEE Transactions on Magnetism, Vol. MAG-22, No. 5, pp. 1069-1071, September 1986.
- [16] Paul, R.P., Robot Manipulators: Mathematics, Programming and Control, The MIT Pres, 1981.
- [17] Lee, C.S.G., Robot Arm Kinematics, Dynamics and Control, IEEE Computer Soc. December, 1982.
- [18] Nagy, F.N., Siegler, A., Engineering Foundations of Robotics, Prentice-Hall Inc., 1987.
- [19] Craig, J.J., Adaptive Control of Mechanical Manipulators, Addison-Wesley Pub. Company., 1988.
- [20] Lind, H., A Distributed Implementation of Dynamic Control on a Direct-Drive Robot System, Ph.D. Thesis, Damek Research Group, Royal Institute of Tech., KTH, Stockholm, Sweden, 1993.
- [21] Stoten, D.P., Model Reference Adaptive Control of Manipulators, Research Studies Press Ltd., UK, 1990.
- [22] Kushal, K.C., Sneckenberger, J.E., Implementation of a PC-Based Adaptive Control Scheme for an Industrial Robot, Proceedings of IEEE/RSJ Int. Conf.on Intelligent Robots and Systems IROS'92, California, USA, July 7-10, 1992.
- [23]. J. Albus, "A new approach to manipulator control: the cerebellar model articulation controller(CMAC)", J. Dyn. Syst. Meas. And Control, pp220-227, 1975
- [24]. Hung J. Y., "Variable Structure Control: A survey", IEEE Transaction on Industrial Electronics, Vol.40, no.1, February, 1993.
- [25]. Utkin, V.I., "Sliding Modes in Control Optimization", Springer-Verlag, 1981.
- [26]. Slotine, J.J., Sastry, S.S., "Tracking Control of nonlinear systems using sliding Surfaces with applications to robot manipulators", International Journal of Control, Vol 38. pp.465-492, 1983.
- [27]. Ertuğrul, M., Kaynak O., Şabanoviç A., "A Comparison of Various VSS Techniques on the Control of Automated Guided vehicles", IEEE International Symposium on Industrial Engineering, ISIE'95 Greece, 1995.

- [28]. Astrom K. J. and Wittenmark B., "Adaptive Control" Addison-Wesley Publishing 1989.
- [29]. Kawato M., Uno Y., Isobe M., Suzuki R. A., "Hierarchical model for voluntary movement and with application to robotics", IEEE Control System Magazine, pp.8-16, April 1988
- [30] Lee, C.C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II, IEEE Trans. on System, Man and Cyber. Vol. 20, No. 2, 1990.
- [31] Zhu, H.A., Teo, C.L., An Enhanced Scheme for Model-Based Control of Robot Manipulators, Int. Journal of Control, Vol. 56, No.6, pp. 1243-1261, 1992.
- [32] Su, C.Y., Stepanenko, Y., Adaptive Control of a Class of Nonlinear Systems with Fuzzy Logic, IEEE Trans. on Fuzzy Systems, Vol. 2, No. 4, pp. 285-294, November, 1994.
- [33] Çela, A., Hamam, Y., A Fuzzy Robust Controller for Trajectory of Robotic Systems, Proceedings of IEEE/RSJ Int. Conf.on Intelligent Robots and Systems IROS'92, California,USA, July 7-10, 1992.
- [34] Lee, C.C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part I, IEEE Trans. on System, Man and Cyber. Vol. 20, No. 2, 1990.
- [35] Lee, C.C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller-Part II, IEEE Trans. on System, Man and Cyber. Vol. 20, No. 2, 1990.
- [36] E. Akbaş, H. Palaz, "Comparison of Recent Intelligent Control Approaches on a Direct Drive Manipulator", The Ninth Turkish Symposium on Artificial Intelligence and Neural Networks, June 21-23, 2000, Izmir, Turkey.
- [37] E. Akbaş, E. Murat Esin, "Performance Evaluation of Intelligent Controllers " The Tenth Turkish Symposium on Artificial Intelligence and Neural Networks, June 21-21-22, 2001, Eastern Mediterranean University, Gazi Magosa, North Cyprus
- [38] Utkin, V.I., Sliding Modes in Control and Optimization, Springer-Verlag, Berlin, 1992.
- [39] Habibi, S.R., Richards, R.J., Sliding Mode Control an Electrically Powered Industrial Robot, IEE Proceedings-D, Vol. No. 2, March, 1992.
- [40] Utkin, V., Variable Structure Systems with Sliding Modes, IEEE

Transaction on Automatic Control, Vol. AC-22, No. 2, 1977.

- [36] Sabanovic, A., Ohnishi, K., Sliding Mode Control of Robotic Manipulators, Proceeding IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 1992.
- [42] Ertuğrul, M., Sliding Mode Control of Automated Guided Vehicles Master Thesis, Boğaziçi University, 1994.
- [43] Ostojic, M., Sabanovic, A., Ohnishi, K., Recursive Control of Robotic Manipulators, JAPAN/USA Symposium on Flexible Automation Vol. 1, ASME, 1992.
- [44] Sabanovic, A., et al., Chattering Free Sliding Mode Control of Robotic Manipulators, Tübitak MAM, Robotic Dep., 1994.
- [45] Narendra, K. S., K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, pp. 4-27, March 1990
- [46] Jang, J.-S. R., C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, PTR Prentice
- [47] E. Akbaş, E. Murat Esin, "A Simulational Comparison of Intelligent Control Algorithms ", *3rd International Symposium on Intelligent Manufacturing Systems*, August, 30-31 2001, Sakarya, Turkey
- [48] Palaz, H., Sarioglu, K., Akbas E., and Ertugrul, M., 1999. Gaussian Neuro Sliding Mode Controller, *Proc. International Conference on Recent Advances in Mechatronics*, ICRAM'99, Istanbul, 68-73, May.
- [49] Wang, L.-X., 1997. A Course in Fuzzy Systems And Control, Prentice-Hall, New Jersey.
- [50] Lee, C.C., 1990. Fuzzy logic in control systems: Fuzzy logic controllers-Parts I and II, *IEEE Trans. Syst., Man, Cybern.*, 20, 404-435, Mar./Apr.
- [51] Haykin S., 1994. A Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Co., Inc.
- [52] Bishop C.M., 1995. Neural Networks for Pattern Recognition, Oxford University Press, Oxford

- [53] Sanner, R.M., and Slotine, J.-J.E., 1992. Gaussian Networks for Direct Adaptive Control, *IEEE Trans. on Neural Networks*, 3, no. 6, 837-863.
- [54] Park, J. and Sandberg, J.W., 1991. Universal Approximation using Radial Basis Function networks, *Neural Computations*, 3, 246-257
- [55] Ertugrul, M., 1999. Neuro-Sliding mode control of Robotic manipulators, *Ph.D. thesis*, Boğaziçi University, Turkey, June.
- [56] Bishop, C.M., 1991. Improving the generalization properties of Radial Basis Function Networks, *Neural Computations*, 3, no. 4, 579-588.

ÖZGEÇMİŞ

EĞİTİM:

1999-2005:Doktora

1996 - 1999: Yüksek Lisans, Bilgisayar Mühendisliği, Gebze Yüksek teknoloji Enstitüsü, Kocaeli,

1992 - 1996 :Lisans, Uçuk Mühendisliği, İstanbul Teknik Üniversitesi (İ.T.Ü.).

1985 - 1992: Kadıköy Anadolu Lisesi.

İŞ DENEYİMLERİ:

2004-... : CBR Software , Kocaeli, Türkiye, Ürün Yöneticisi

2003-2004 : CBR Software , Kocaeli, Türkiye, Proje Yöneticisi

2000-2003: Türkiye Bilimsel ve Teknik Araştırma Kurumu –Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü , Kocaeli, Türkiye ,Kripto Sistemleri Yönetim Grubu, Uzman

1999-2000: Türkiye Bilimsel ve Teknik Araştırma Kurumu –Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü. Araştırmacı.

1998-1999: Türkiye Bilimsel ve Teknik Araştırma Kurumu -Marmara Araştırma Merkezi , Yazılım/Simülasyon Grubu . Araştırmacı.

1997-1998: Türkiye Bilimsel ve Teknik Araştırma Kurumu -Marmara Araştırma Merkezi, Yazılım, Yapay Zeka Grubu . Araştırmacı.

ARAŞTIRMA ALANLAR:

- Zeki Sistemler
- Robotik
- Network Yönetimi.
- Problem Yönetimi.
- Network Keşfi.
- Dağıtık Sistemler Yazılım Mimarisi (CORBA, Mobile Agents, Java and Web Technologies, Network Management).
- Nesne İlişkili Teknolojiler.
- Network Güvenliği.

DERGİLER:

E. Akbas, E. Murat Esin, "A Simulational Comparison of Intelligent Control Algorithms ", The Journal of Robotics and Autonomous Systems, 235-244, Vol 49/3-4,2004

ULUSLARARASI KONFERANSLAR:

E. Akbas, E. Murat Esin, "Designing Stable Intelligent Systems For Payload Variations", Yapay Zeka ve Sinir Ağları 30. Türkiye Sempozyumu ,ss 537-546, 09-011 Haziran 2004, İzmir, Türkiye

E. Akbas, E. Murat Esin, "A Comparative Study of Soft Computing Methodologies in Identification of Nonlinear Systems" Yapay Zeka ve Sinir Ağları 30. Türkiye Sempozyumu ,ss 537-546, 09-011 Haziran 2004, İzmir, Türkiye

E. Akbas, E. Murat Esin, "Intelligent Stabilization of Direct Drive Manipulators "IEEE Region 8 EUROCON 2003, The International Conference on Computer as a Tool,21-24 Eylül 2003, Ljubljana, Slovenia.

E. Akbas, E. Murat Esin, "SEAMLESS INTEGRATION OF NETWORK MANAGEMEN PROTOCOL WITH DISTRIBUTED CONTROL", The IJISIP Proceedings (ISSN: 1304-2386), Volume: 1, Number :1, July 2003, International XII. Turkish Symposium of Artificial Intelligence and Neural Networks (TAINN' 2003).

E. Akbas, "System Independent And Distributed Fault Management System", The Eighth IEEE Symposium on Computers and Communications,30 June-3 July 2003, Antalya, Turkey.

E. Akbas, "Web Based Management: A Novel Architecture ", The Sixteenth International Symposium on Computer and Information Sciences (ISCIS XVI), October, 29-31, 2001, Antalya, Turkey

E. Akbas, E. Murat Esin, "A Simulational Comparison of Intelligent Control Algorithms ", 3rd International Symposium on Intelligent Manufacturing Systems, August, 30-31 2001, Sakarya, Turkey

E. Akbas, E. Murat Esin, "Performance Evaluation of Intelligent Controllers " The Ninth Turkish Symposium on Artificial Intelligence and Neural Networks, June 21-21-22, 2001, Eastern Mediterranean University, Gazi Magosa, North Cyprus.

E. Akbas, C. Ersoy, "Multi Constarint Path Finder Algorithm", The Fifteenth International Symposium on Computer and Information Sciences (ISCIS XV), October, 11-13, 2000, ISTANBUL, Turkey .

E. Akbas, H. Palaz, "Comparison of Recent Intelligent Control Approaches on a Direct Drive Manipulator", The Ninth Turkish Symposium on Artificial Intelligence and Neural Networks, June 21-23, 2000, Izmir, Turkey.

H. Gümüşkaya, T. Dursun, E. Akbas, O. Davulcu, "A CORBA and Java Mobile Agent Based Network Management Architectures", International Symposium on Computer and Information Sciences XIV (ISCIS'14), pp. 975-982, October 18-20, 1999, Kusadasi, Izmir.

H. Gümüşkaya, E. Akbas, T. Dursun, O. Davulcu, "Utilization of CORBA, Mobile Agents, Java, and Web Technologies for Network Management", Regional Conference on Military Communication and Information Systems '99, October 6-8, 1999, Zegrze, Poland.

Hasan Palaz, Kemal Sarioglu, Ertugrul Akbas and Meliksah Ertugrul, "Gaussian Neuro Sliding Mode Controller", Proc. Of the 2nd International Conference on Recent Advances in Mechatronics, pp 68-74, May 24-26, 1999, Istanbul, Turkey.

Ertugrul Akbas, Meliksah Ertugrul, Okyay Kaynak and Yagmur Denizhan, "Neural Network Based Robot Controllers", Proc. Of the 2nd International Conference on Recent Advances in Mechatronics, pp 251-257, May 24-26, 1999, Istanbul, Turkey.