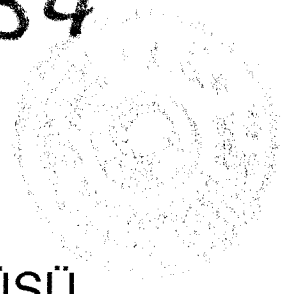


84054



T.C.

GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

OTOMATİK YÜK ARABALARI İÇİN
DENETİMCİ TASARIMI

Ertuğrul AKBAŞ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TEZ DANIŞMANI
Yrd.Doç.Dr. E. Murat ESİN

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANİZASYON MERKEZİ

GEBZE

1999

Bu tez çalışması, G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 28.1.1999 tarih ve 99/02 sayılı kararıyla oluşturulan jüri tarafından Bilgisayar Müh. Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE
(Tez Danışmanı)

: YRD. DOÇ. DR. MURAT ERİN
Murat Erın

ÜYE

: YRD. DOÇ. DR. RACIP BAŞBUC
Racip Başbuç

ÜYE

: YRD. DOÇ. DR. ERTUĞRUL KARACUMU
Ertuğrul Karacumu

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 11/02/1999 tarih ve 99/04 sayılı kararı.



ÖZET

Kendinden devinimli yük arabaları fabrika otomasyonunda ve esnek imalatla çok önemli bir yere sahiptir. Bundan dolayı otomatik yük arabası için denetimci tasarımı pek çok araştırmacının ilgisini çekmiştir. Bu çalışmalarda pek çok denetimci tasarımı ele alınmış ve onların performansları karşılaştırılmış ve sonuçlar tartışılmıştır.

Denetimci tasarlarken göz önüne alınması gereken en önemli nokta sistemin parametrelerinin zamana bağlı değişebileceğidir. Yük arabalarının taşıdığı yük değiştiğinde atalet, diğer bir deyişle dinamiği değişir. Bundan dolayı tasarlanacak olan denetiminin bu değişikliklere karşı gürbüz olması gerekir.

İlk denetimci tipi olarak, belli bir alanda gürbüz olan ters dinamik modelleme yöntemi ile denetimci tasarlandı. Bu denetimci ile birlikte bir de PD denetimci tasarlandı. Lineerleştirme yapay sinir ağı kullanılarak yapıldı ve bu metoda ters dinamik modelleme yöntemi adı verilir.

İkinci kontrol yöntemi olarak da diğer bir popüler yapay sinir ağı ile kontrol yöntemi kullanıldı. Bu metot hata geri besleme öğrenmeli yapı olarak adlandırılır. Bu online öğrenen bir yapı olup modelleme ve gürültüyü aynı zamanda da bozucuları kompanse etmek için ikinci bir denetimciye ihtiyaç duyar.

Kontrol algoritmalarını gerçekleştirebilirliği bilgisayar benzetimi ile doğrulanmıştır.

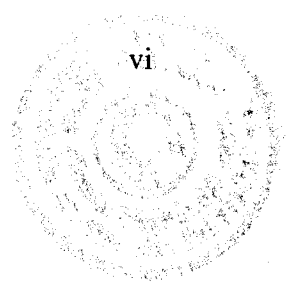
SUMMARY

Automated Guided Vehicles (AGV) have a significant importance in factory automation and flexible manufacturing. The control and guidance of an AGV has therefore attracted the attention of many researchers. In his work a number of controller designs are considered and their performances are compared and discussed.

When designing a controller the most important point is that the system parameters may change with time. With the change of the load of the vehicle, the inertia, i.e. the dynamics of the vehicle may change. Therefore, the controller should be designed to overcome any parameter changes.

The first controller considered in this work is a feedback-feedforward linearization together with a PD control algorithm. This linearization achieved by using neural network and this method is called direct inverse modeling architecture. Secondly, another popular neurocontrol algorithm is used. This method is called Feedback Error Learning Architecture. It is an online learning controller and it need a secondary controller in order to compensate the modeling errors and also noise and disturbance.

All algorithms are simulated on a digital computer.



TEŞEKKÜRLER

Bu tezin hazırlanmasında bana olan katkılarından dolayı sayın Yrd. Doç: Dr. Ragıp Başbuğ'a sonsuz teşekkürlerimi sunmak isterim.

Yrd. Doç. Dr. E. Murat Esin'e de tezle ilgili değerli görüş ve önerileri için teşekkür ederim.

Tez ile ilgili değerli görüşler sunan ve bana tezin gelişimi aşamasında değerli katkılarda bulunan Melikşah Ertuğrul' a da teşekkürü bir borç bilirim.



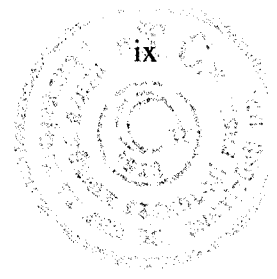
İÇİNDEKİLER

SAYFA

SUMMARY.....	v
TEŞEKKÜRLER.....	vi
İÇİNDEKİLER.....	vii
SİMGELER VE KISALTMALAR	ix
ŞEKİLLER.....	x
1. GİRİŞ	1
1.1. Otomatik Yük Arabaları.....	3
1.1.1. OYA Kullanmanın Faydaları	3
1.2. Uygulamalar	5
1.3. OYA Tipleri.....	5
2. YAPAY SİNİR AĞLARI İLE KONTROL.....	8
2.1. Yapay Sinir Ağlarına Giriş.....	10
2.2. Biyolojik Sinir Ağları	12
2.2.1. Yapay Sinir Ağının Kullanım Alanları ve Geliştirilen Uygulamalar.....	16
2.2.2. Yapay Sinir Ağının Yararları	17
2.2.3. Uygulama Çeşitleri	17
2.2.4. MICR (Manyetik Karakter Tanıma)	18
2.2.5. İşlev Belirleme	20
2.3. Temel Yapay Sinir Ağları Algoritmaları.....	22
2.3.1. Backpropagation	22
2.3.2. Radial Basis Functions Yapay Sinir Ağları	24
2.4. Yapay Sinir Ağları İle Kontrole Giriş	27
2.4.1. Ters Dinamik Modelleme Yöntemi (DIMA)	27
2.4.2. Ters Dinamik Modelleme Yapısı Tip II(DIMA II).....	28
2.4.3. Hata Geri Beslemeli Öğrenme Yöntemi (FELA).....	28
3. OTOMATİK YÜK ARABASI DİNAMİK VE KİNEMATİK DENKLEMLERİ	
30	
3.1. Giriş.....	30
3.2. Otomatik Yük Arabasının Dinamik Denklemleri.....	32
3.3. Durum Denklemleri.....	35

3.5.	Otomatik Yk Arabası Ters Kinematik Denklemleri.....	37
3.6.	Referansın Tanımlanması.....	37
4.	TERS DİNAMİK MODELLEME YÖNTEMİ.....	39
4.1.	İkincil kontrol elemanın tasarlanması:	41
4.2.	Ters Dinamik Modelleme Yöntemi Simlasyon Sonuçları.....	43
5.	HATA GERİ BESLEME ÖĞRENMELİ YAPI	54
5.1.	Hata Geri Besleme Öğrenmeli Yapı Simlasyon Sonuçları.....	56
6.	SONUÇ	62
KAYNAKLAR		65
ÖZGEÇMİŞ		68
EKLER		69





SİMGELER VE KISALTMALAR

R_a	Armatür direnci
L_a	Armatür indüktansı
$\dot{I}_a$	Armatür akımı
w_m	Motor açısal hızı
θ_m	Motorun açısal deplasmanı
T_m, T_1, T_2	Tork
J_m	Motor şaftına gelen toplam atalet momenti
J_2	Teker açısal momenti
F_{c2}	Sürtünme Katsayısı
B	Viskoz sürtünme katsayısı
$\frac{N_1}{N_2}$	İletim oranı
V	Araba ağırlık merkezi hızı
E	Toplam Hata
e	Hata

ŞEKİLLER DİZİNİ

	Sayfa
Şekil 1.1. Tipik bir OYA.....	4
Şekil 1.2. OYA nın üretim bandına entegrasyonu	6
Şekil 1.3. OYA çeşitleri	7
Şekil 2.1. Sinir hücresi	12
Şekil 2.2. Yapay sinir ağı.....	14
Şekil 2.3. Trashold fonksiyonları.....	16
Şekil 2.4. Nöron	21
Şekil 2.5. Backpropogation.....	24
Şekil 2.6. RBF yapay sinir ağı	25
Şekil 2.7. DIMA.....	27
Şekil 2.8. DIMA II.....	28
Şekil 2.9. FELA	28
Şekil 2.10. Detaylı kawato modeli	29
Şekil 3.1. OYA modeli.....	30
Şekil 3.2. Anlık hareket açısı	31
Şekil 4.1. Ters dinamik modelleme yöntemi blok diagramı	39
Şekil 4.2 Sistemin simülasyon modeli	40
Şekil 4.3. X_Y düzleminde referans	44
Şekil 4.4 X_Y düzleminde hareket	44
Şekil 4.5. Hız profili.....	45
Şekil 4.6. Hız hatası	45
Şekil 4.7. Açı profili.....	46
Şekil 4.8. Açı hatası	47
Şekil 4.9. Kontrol çıkışı	47
Şekil 4.10. X_Y düzleminde hareket	48
Şekil 4.11. X düzlemindeki hata	48
Şekil 4.12. Y düzleminde hata	49
Şekil 4.13. Hız profili.....	49
Şekil 4.14. Hız hatası	50
Şekil 4.15. Açısal hız	51
Şekil 4.16. Açı profili.....	51

Şekil 4.17. Açık hatası	52
Şekil .4.18. Kontrol çıkışı 1	52
Şekil .4.19. Kontrol çıkışı 2	53
Şekil 5.1. Kawato modeli.....	54
Şekil 5.2. Detaylı FELA yapısı	55
Şekil 5.3. X_Y düzleminde referans	57
Şekil 5.4. X_Y düzleminde hareket	57
Şekil 5.5. X hatası	58
Şekil 5.6 Y hatası	58
Şekil 5.7. Hız profili.....	58
Şekil 5.8. Hız hatası.	59
Şekil 5.9. Açık profili.....	59
Şekil 5.10. Açık hatası	60
Şekil 5.11. Kontrol 1	60
Şekil 5.12. Kontrol 2.	61

1. GİRİŞ

Günümüzde otomatik yük arabaları çok fazla önem kazanmaktadır. Bunun en önemli sebeplerinden birisi pek çok yeni teknolojinin bilgisayarlı kontrol tekniğine katkıda bulunarak otomasyonun fabrikalara yerleşmesidir. Otomatik yük arabaları *bilgisayar destekli üretimin* en önemli elemanlarından birisi olduğu için otomasyonun önemli bir parçasıdır.

Otomatik yük arabaların en önemli özellikleri:

- 1-Sürücüsüzdürler
- 2-Bir güzergah üzerinde hareket ederler. Bu güzergah yüzeyde olabileceği gibi yüzeyin içinde de olabilir.
- 3-Yük taşımak için kolayca üretim hatlarına adapte ve monte edilebilirler.

Otomatik yük arabalarının en önemli özelliği çok ağır yükleri uzaktaki bir noktaya taşıyabilmeleri ve bu esnada çok karışık bir güzergahı bilgisayar kontrolü altında takip edebilmeleridir. Burada otomatik yük arabalarının robotlara ve konveyörlere olan üstünlüğünden de bahsetmekte yarar vardır. Robotlar otomatik yük arabaları gibi hareketli değildirler buna karşılık konveyörler de otomatik yük arabaları gibi esnek değildirler.

Otomatik yük arabaları için önemli konuların başında, güzergah planlama, trafiğin düzenlenmesi, yoldaki cisimlerden korunma veya bertaraf etme, yük transferi, mekanik tasarımı ve kontrolü gelmektedir. Bu tezin amacı mekanik tasarım ve özellikle kontrol alanlarındaki bir boşluğu doldurmaktır. Çünkü yukarıda bahsedilen alanlarda pek çok çalışma yapılmış olmasına rağmen mekanik tasarım ve kontrol konularında fazla bir çalışma yoktur. Özellikle zeki kontrol yöntemlerinin otomatik yük arabalarına uygulanması konunun yapısını çok uygundur.

Kontrol yönteminin seçilmesi konunun en önemli noktalarından biridir. Çünkü araçtaki yük devamlı değişebilir ve bu da ataletle sebebiyet vermekte bunun sonucu olarak da aracın dinamiği değişebilmekte. Diğer bir değişle aracın parametreleri işletim esnasında değişmekte dolayısıyla bu değişimi göz önüne alarak bir tasarım yapmak şart olmaktadır.

Zeki kontrol yöntemleri (yapay sinir ağları ile kontrol, Bulanık mantık kontrolü, yapay sinir ağları-bulanık kontrol) ile kontrol bu şartlar altında kullanılabilecek metotlardan biridir. Bu metodun değişen şartlara karşı gürbüzlüğü kanıtlanmıştır. Doğrudan Ters Modelleme Yöntemi, DIMA, (Direct Inverse

Modelling Architecture) bu zeki kontrol modellerinden biridir. Bu metotta aracın ters dinamięi belli alıřma kořullarında bir yapay sinir aęı ile renilmeye alıřılır. Bunu yaparken aęa giriř uzayının mmkn olduęunca geniř olmasına dikkat edilirse aracın parametreleri alıřma esnasında deęiřse bile uzayın iinde kalacaęı iin aę yapıyı bařarı ile renir. Daha sonra bu yapı ile birlikte bir PD kontrolr de lineer olmayan etkileri kompanse etmesi iin kullanılarak sistemin kontrolr modellenmiř olur.

Bu alıřmada deęiřik yapay sinir aęı kontrol modelleri araca uygulanmıř ve bu modeller birbirleriyle kıyaslanmıřlardır. Bu kıyaslama aracın referansı takibi ve kontrolrn ıkıřının devamlılıęı ile olmuřtur.

Sistemin modellenmesi sırasında MATLAB/SIMULINK kullanılmıřtır. Aynı zamanda matlab script dosyaları ile yapay sinir aęı algoritmaları yazılmıřtır.



1.1. Otomatik Yk Arabaları

Otomatik yk arabasının tanımına bakmak iin Metarial Handling Institute of USA [Miller,1987]'in Otomatik yk arabaları blmne bakarsak.:

Otomatik gdm ekipmanlarıyla donatılmıř olan ve daha nceden belirlenen gzergahı izleme yeteneđi olan ve programlanabilme veya sistem tarafından ihtiya uyulan fonksiyonları icra edebilme yeteneđine sahip aralardır.

Peki otomatik yk arabaları bir robot mudur ?

Bu sorunun cevabını da Automated Guided Vehicles and Automated Manufacturing, Society /Robotics Industries Assocation [Miller,1987]' da bulabiliriz.

Robot programlanabilen ve ok fonksiyonlu bir manipulatr olup, metalleri, paraları, aletleri daha nceden programlanan řekilde tařımak iin kullanılır.

Bu iki tanımı kullanarak otomatik yk arabalarını sınırlı sayıda serbestlik dereceli birer robot sayabiliriz.

Otomatik Yk Arabaları (OYA) bilgisayar kontrol altında alıřırlar ve otomatik yn bulma ve pozisyon bulma iin programlanabilirler. Srcszdrler ve yere izilmiř bir gzergahı takip edebilirler. En nemli zelliklerinden birisi ise diđer imalat aletleriyle entegre edilebilir olmaları ve ok ađır ykleri kolayca tařıyabilir olmalarıdır. Bu zelliklerinden dolayı robotlardan ayrılırlar nk robotlar yryebilir deđildirler. Ve bundan dolayı da OYA kadar esnek deđildirler.

1.1.1. OYA Kullanmanın Faydaları

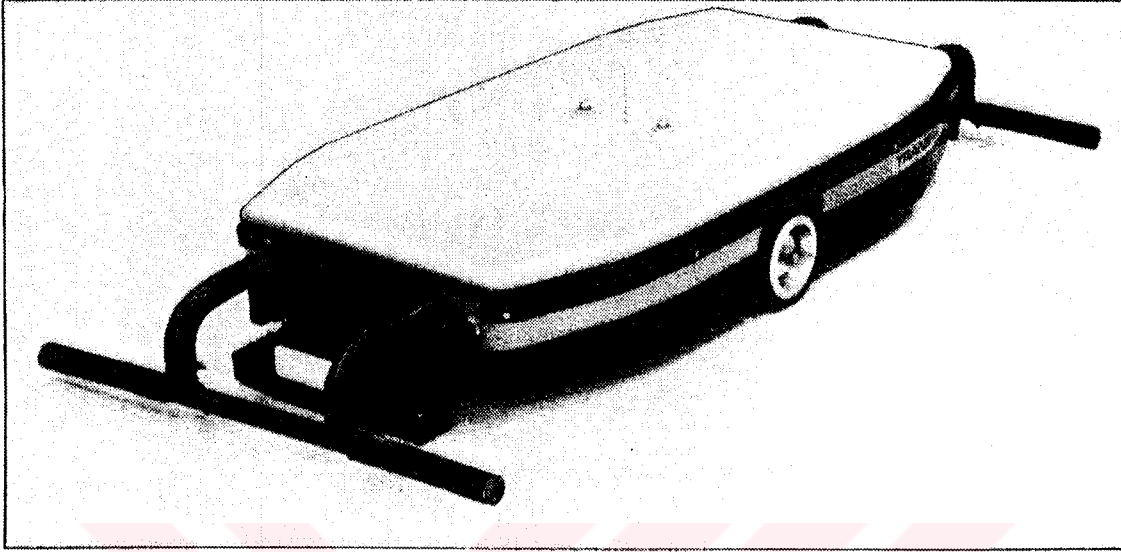
Otomatik yk arabalarının faydalarını anlamak iin yeniden The Material Handling Institue [Miller,1987] kaynaklarına bakarsak:

1-Ekonomik Faydaları :Pek ok metal tařıma durumunda Oya en ekonomik zm olarak gzkmektedir. Bundan dolayısı ki pek ok iřletmede kullanılmaktadır.

2-Diđer Sistemlerle Otomatik Entegre : Otomatik yk arabaları retimde kullanılan diđer sistemlerle otomatik olarak entegrasyona girebilmeleri iin tasarlanmıřtır.

3-Sistemin Gözlenebilirliği : Bilgisayar kontrolü altında materyallerin belirli istasyonlar arasında gidiş gelişleri ile ilgili her türlü bilgi alınabilir.

4-Emeğin Azalması\Üretkenliğin Artması : Otomatik yük arabası verimi artırır.



Şekil 1.1. Tipik bir OYA.

Otomatik yük arabasının emek /üretkenlik oranı aynı işi yapan diğer aletlere göre çok daha yüksektir.

5-Güzergah 'ın Kolayca Değiştirebilmesi : Materyalin taşınmasında oluşacak değişiklik yada herhangi bir üretim bandının genişliğinin artması durumunda OYA çok esnek olarak buna uyum sağlar.

6-Sistem Kapasitesi Arttırılabilir: Materyal taşıma ihtiyacı arttıkça OYA ' ya yeni arabalar eklenerek bu ihtiyaç giderilebilir.

7-Engelsiz Takip : OYA sistemleri yere monte edildikleri için hareketi kısıtlayıcı herhangi bir durum oluşturmazlar.

8-Sınırsız Hedef : OYA pek çok noktada yükleme ve boşaltma yapabilmek üzere programlanabilir.

9- Daha Az Ürün ve Alet Hasarı : Çalışmalar göstermiştir ki OYA daha az zarar ile çalışmaktadır. Bunun sebebi de OYA 'nın daha önceden tespit edilen bir yolu takip etmesidir.

10- Güvenilir Sistem Kapasitesi : OYA pek çok arabadan meydana geldiği için sistem kapasitesi bir arabanın bozulmasından etkilenmez.

11-Enerji Tasarrufu : OYA çalışmak için çok az bir enerjiye ihtiyaç vardır.

12-Kolay Kurulabilme : OYA profesyoneller tarafından tasarlandığı için , çok kısa zamanda kurulabilir .

13- Mevcut Mekanlara Kolay Kurulum : OYA mevcut olan binalara kolayca kurulabilir.

1.2. Uygulamalar

- 1- Uzay (Aerospace)
- 2- Otomotiv (Automotive)
- 3- Konteynır'e (Container Ports)
- 4- Elektronik (Electronics)
- 5- Hastaneler (Hospitals)
- 6- Metal Rafinerileri (Metal Refineries)
- 7- Mobil Robotlar (Mobile Robots)
- 8- Kağıt Üretimi (Paper Mills)
- 9- Çelik Üretimi (Steel Mills)
- 10-Warehousing

1.3. OYA Tipleri

- 1-Fork Trucks
- 2-Unit Load Vehicles

3-Pallet Trucks

4-Work Platforms

5-Towing Vehicles

6- Light Load Vehicles



Şekil 1.2. OYA nın üretim bandına entegrasyonu.

1-Fork Trucks :En yeni OYA tiplerinden biridir. Paletleri ile yüzeydeki vede üstüste duran yükleri alabilmektedir.

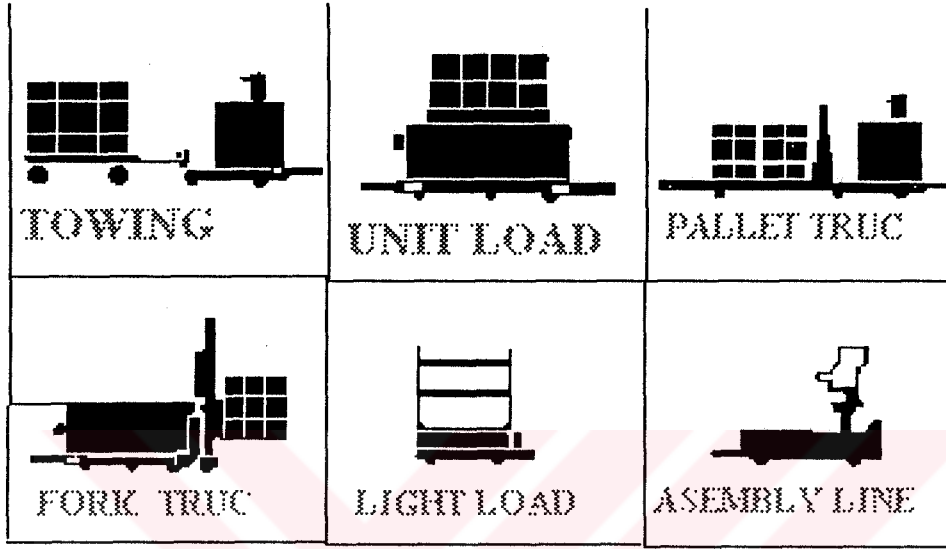
2-Unit Load Vehicles : Bir çeşit taşıyıcı eklenmiş olup birim yük taşımaya yöneltmiştir. Bu sistemdeki taşıyıcı ihtiyaca göre tasarlanabilmektedir.

3-Pallet Trucks : Paletlerden gelen yükleri taşımak için tasarlanmış olup yükleme için ayrıca bir yükleyiciye ihtiyaç duymadan taşıma yapmak için tasarlanmıştır.

4-Work Platforms : Üzerine çeşitli âletlerin de monte edilebileceği otomatik yük arabalarıdır.

5-Towing Vehicles : Şu anda kullanılan en popüler olan ve en eski OYA tipidir. Bu tip OYA 2.500 kg ila 23.000 kg arası yük taşır.

6- Light Load Vehicles : Bu tür Otomatik yük arabaları 225 kg civarında yük taşımak üzere tasarlanmıştır. Bunlar özellikle seri üretim bantları için tasarlanmış araçlardır.



Şekil 1.3. OYA çeşitleri.

2. YAPAY SİNİR AĞLARI İLE KONTROL

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

Klasik kontrol tekniklerine örnek olarak :

- 1-PID kontrol
 - 2-Model Reference Adaptive Kontrol
 - 3-Self tuning reguators
 - 4-Computed Torq
 - 5-Sliding Mode Kontrol
- verilebilir.

Diğer bir grup olan zeki kontrole ise

- 1-yapay sinir ağı 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 non-linear 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 insan oğlu 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 ağı ile kontrol konusuna değineceğiz.

Yapay sinir ağı kendi doğasından gelen özellikleri sebebiyle bir çeşit adaptif kontrol elemanı olarak görülebilirler. Bu özelliklerinin yanı sıra yapay sinir ağı fonksiyon tanıma ve fonksiyon uydurmada da kullanılan bir metottur. Yapay sinir ağının bu yukarıda bahsedilen bu özelliklerini de kullanarak yapay sinir ağının kontrol de nasıl kullanıldıklarına bir bakalım.

Yapay sinir ağıları kontrolde özellikle de Robotik uygulamalarda

1-Kinematik

2-Dinamik

3-Güzergah Tanımlama

4-Hareket Kontrolü

gibi uygulamalarda kullanılabilir.

Robot kontrolü ve dolayısıyla yapay sinir ağı ile kontrol ile ilgili ilk çalışma CMAC (Cerebeller Model Articulation Controller) ile Albus [Albus, 1975] tarafından yapılmıştır. Burada önerilen yapı ise standart yapay sinir ağı yapısından farklı olarak bir "look-up table" tablo şeklinde olup yapay sinir ağının çıkışı bu tablodaki çeşitli hücrelerin değerlerinin toplanmasıyla ortaya çıkar "

Diğer bir çalışma da ters dinamik ve ters kinematik öğretmeyle ilgi olarak yapılan çalışmadır. Bu çalışmayı Psaltis [Psaltis,1987], Guez ve Ahmed [Guez ve Ahmed,1988] yapmışlardır. Bu konulardaki diğer önemli çalışmaları Elsey [Elsey, 1988], Grossberg ve Kuperstein [Grossberg ve Kuperstein,1989], Fukuda [Fukuda,1990], G.A Bekey [Bekey,1992] yapmışlardır.

Mitsua Kawato [Kawato, 1988] " Hierarchical neural network model for voluntary movement " adlı makalesinde yani bir metot uygulamış ve bu metotta sistemde iki adet kontrolör kullanıp bunlardan bir tanesini yapay sinir ağı diğerini de PD kontrolör olarak seçmiştir.

Buradaki en önemli yenilik de modeldeki ters dinamiğin on-line olarak bir yapay sinir ağına öğretilmesi olup gayet başarılı sonuçlar vermiştir.

Daha önce de bahsedildiği gibi pek çok yapay sinir ağı ile kontrol metodu mevcuttur.

Bunları sıralarsak:

1-Ters dinamik modelleme yöntemi (Direct Inverse Modelling Architecture (DIMA))

2-Ters dinamik modelleme yöntemi tip ii (Direct Inverse Modelling Architecture (DIMA) Tip II)

3-Hata geri besleme öğrenmeli yapı (Feedbak Error Learning Architecture (FELA))

4-Yapay sinir ağı ile kayan kipli kontrol (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 yapay sinir ağının birincil kontrol elemanı olarak

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

İkinci tip ters dinamik modelleme yönteminde (DIMA II) 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 metod olan hata geri besleme öğrenmeli yapı (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 yapay sinir ağı ile kontrolde kullanılmaktadır.

Bu tezde öncelikle bir otomatik Yük arabası tasarlanacak ve daha sonra bu yük arabası için bir zeki kontrol elemanı tasarlanacaktır.

Bu tasarımlar daha sonra MATLAB/SIMULINK ortamında test edilecek ve simulasyon sonuçları sunulacaktır.

2.1. Yapay Sinir Ağlarına Giriş

Yapay Sinir Ağı konusu, ilk olarak 1943 yılında, McCulloch ve Pitts tarafından ele alınmıştır. 1943 yılı yapay sinir ağı sistemleri için bir doğuş yılı olarak görülür. McCulloch ve Pitts'in yaptıkları ilk yapay sinir ağı (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 yapay sinir ağı araştırmalarında, ilk fonksiyonel nöronu yaparak, yeni bir devir açtılar.

Bu tarihten sonra kısaca gelişmeler şöyle gerçekleşti : Bu on yılın içerisinde (1949) , Donal Hebb, bilginin, sinirler arası bağlantılarda saklanabileceğini keşfetti. Daha sonra Rosenblatt [Rosenblatt, 1960] 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.

1960 yıllarının başında ADALINE (ADaptive LINEar combiner) denen bir sistem icat edildi. Bu yeni icat Widrow-Hoff adında güçlü öğrenme kurallarına sahip idi ve Bernard Widrow-Marcian Hoff (1960-62) tarafından icat edilmişti. Bu kural kısaca şöyledir;

Desen tanıma sırasında oluşan hata payları karesel bir orantı ile hesaplanıp sisteme geri gönderiliyordu. Böylece sistem, tanıma işlemini daha doğru bir şekilde yeniden yapıyor. 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, Fukushima, Kohonen, Anderson ve Hopfield yapay sinir ağı üzerinde çalışmalar yaptılar. Bu araştırmalara rağmen yapay sinir ağı 80'li yılların sonuna kadar pek ilgi odağı olamadı. Ta ki 1982 yılına kadar...

John Hopfield [Hopfield, 1984] hafıza ilişkilerini anlamak üzere bir ağ mimarisi geliştirdi. Bu gelişme yapay sinir ağı konusundaki çalışmalara hız verdi ve bu çalışmaların ilgi odağı olmasını sağladı.

Yapay sinir ağı 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 yapay sinir ağı araştırma programlarının ortaya çıkışları hızlandı, araştırmalar kuvvetlenmeye başladı. Bunu, dünya çapında verilen konferanslar, yeni yayınlar izledi. Yapay sinir ağı ile çözülecek problemlerin sayısı ve bu daldaki uygulamaların adeti azlıktan çokluğa doğru hızla ilerledi.

1990'larda artık yapay sinir ağı için özel makineler, sistemler, mikro-çipler, yazılımlar geliştirildi, araştırma grupları arttı, kullanım alanı muazzam derecede genişledi ve bir o kadar da bu dala aktarılan maddi kaynak inanılmaz derecelere yükseldi.

Sonuçta neuro bilgisayar, uzun bir zamandır araştırma konusu ve ancak bu seviyeye geldi, fakat şunu da unutmamak lazım ki, şu an yapay sinir ağı sistemleri bebeklik çağını yaşıyor.

Yapay sinir ağıları aşağıdaki gibi tanımlanabilir:

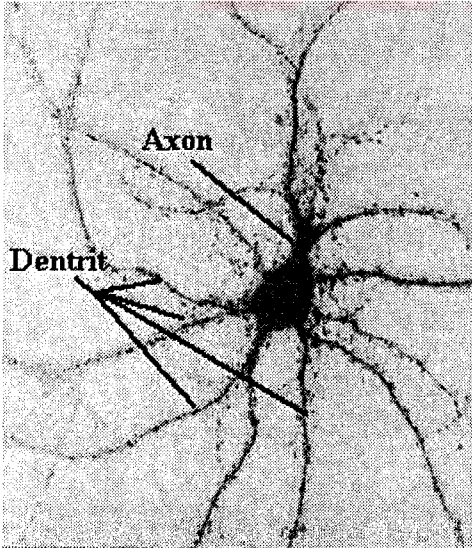
Pek çok paralel işlem yapan elemandan oluşmuş ve bu elemanların fonksiyonu da yapay sinir ağının yapısını belirlemek olan bir sistemdir.

Bu sistemde yapay sinir ağının yapısı bu elemanlar arasındaki bağlantının şekli ve kuvveti ile belirlenir.

2.2. Biyolojik Sinir Ağları

Bir sinir hücresi diğer sinir hücrelerine Axon ve Dentritlerle bağlıdır. Bilgi akışı elektrik sinyalleri olarak bu kanallardan iletilir. Sinir sistemini oluşturan ve birbirlerine Axon ve Dentritlerle 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.

Her sinir hücresinin bir axon çıkışı vardır. Daha sonra bu axondan yüzlerce bağlantı çıkabilir. Ve sinir hücrelerine giren yüzlerce dentritler vardır. Bu dentritlerin herbiri bir sinir hücresinden çıkan axona bağlı kanallardır. Kısaca sinir hücresine giren kanallara (bilgi alma) dentrit, çıkışı sağlayan kanala (bilgi verme) axon denir.



Şekil 2.1. Sinir hücresi.

Yukarıdaki şekilde bir sinir hücresi görülmektedir. Görüldüğü üzeri kalın çıkış Axon, ince girişler ise Dendritlerdir.

Sinir hücrelerinin bilgi göndermesi üç biçimde olur

- 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 Threshold değeri adı verilir ve Teta ile gösterilir. Eğer gelen sinapsların toplu bir işleminden geçirildikten sonraki sonuç değeri Threshold değerini aşabilirse, ulaşılan değer axondan sinaps olarak gönderilir. Ayrıca her bir dendritin taşıdığı enformasyonun 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 bilgi 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. Örneğin; burundaki bir sinir hücresinin kokulara karşı olan hassasiyeti, gözdeki sinir hücrelerine göre daha hassas eşik değerine sahiptir. Veya göz aşırı bir ışığa maruz kalırsa, buradaki sinir hücreleri otomatik olarak ışığa karşı olan hassasiyetlerini azaltırlar yani eşik değerlerini arttırırlar ki, gelen tepkiler büyük olsalar bile bu eşik değerinden geçmesin.

Şimdi birazda insandaki sinir sistemine kısa bir göz atıp, beynin işleyişine bakalım. Beyin ve omurilik birlikte Merkezi Sinir Sistemini oluştururlar. Beyinden ve omurilikten zuhur eden , vücudun her yerine ulaşan sık ve beyaz sinir hücreleri de peripheral (Çevresel) sinir sistemini oluşturur. Merkezi sinir sistemi iki bölümden oluşur.

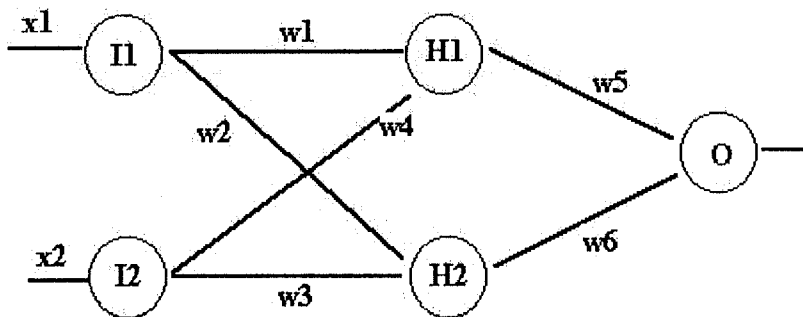
- 1) Beyin; entegre, kurumsal, düşünce, davranış kontrolü.

2) Omurilik; beyinden elen 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 organ insanlarda çok daha gelişmiştir. Bir insan beyni yaklaşık 1,4 kg ağırlığındadır.ve içerisinde milyarlarca sinir hücresi barındırır. 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 (dendritler) ne kadar karışık ve çoksa, düşünce gücünün o denli fazla olduğu görüşü yaygındır. Bu bağlar beyin çalıştıkça, 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 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 bilgisayarlarda simüle edilmeye başlandı ve duruma göre tepkileri kontrol edilmeye başlandı. Bir bakıma beynin simülasyonu yapıldı. Tabii ki milyarlarca hücreyi simüle etmek şu anki teknoloji itibari ile imkansızdır.

Peki sinir sistemi bilgisayar ortamına nasıl uyarlanır.

Bu maksatla sinir sistemi mantığı taklit eden algoritmalar kullanılır

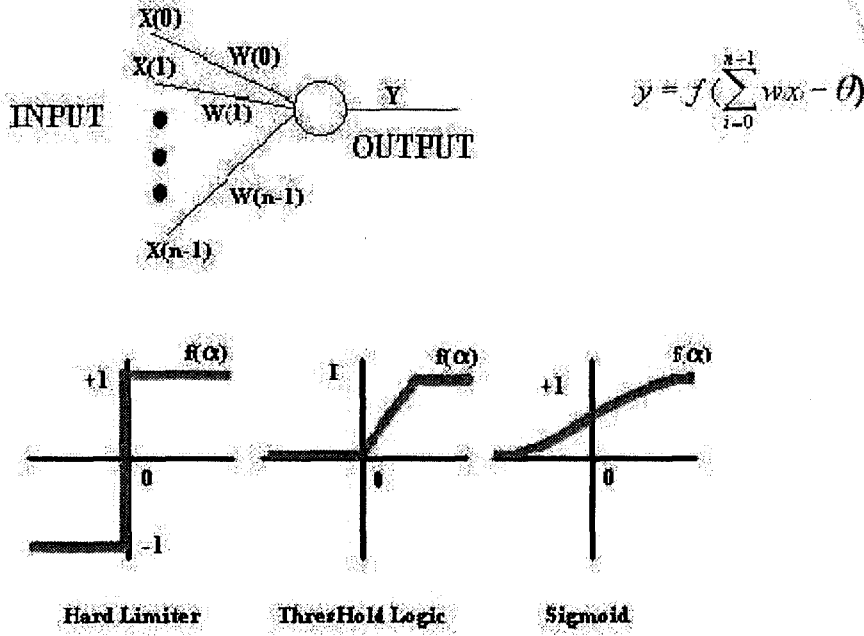


Şekil 2.2. Yapay sinir ağı.

Yukarıda verilen şekilde her bir içi boş yuvarlak, bir sinir hücresini, girdiler dentritleri, çıktılar ise aksonları temsil etmektedir. Bu şekil genel olarak kullanılan bir şablondur. Şekilde görüldüğü gibi sistem 3 tabakadan oluşmuştur. 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 daireler ise bağlantı ağırlıklarını temsil eder.





Şekil 2.3. Trashold fonksiyonları.

2.2.1. Yapay Sinir Ağının Kullanım Alanları ve Geliştirilen Uygulamalar

Günümüzde hızla gelişen bilgisayar teknolojisi ile, genelde desen tanıma ile uğraşan yapay sinir ağları artık bu işlemleri çok daha başarılı bir şekilde icra edilebilmektedir. Yapay sinir ağı genel olarak, şekil-desen tanıma, tahminde bulunma ve karakter tanıma gibi işlerde kullanılmaktadır. Bu sistem kullanılarak, desen tanımanın yanı sıra gelen bilginin (ses veya görüntü) kirliliğini, parazit ve bozukluğunu en aza indirilmesi veya tamamen düzeltilmesi işlemleri başarı ile yapılabilmektedir. Böyle bir teknolojiye iş çevrelerden parasal destek de büyüktür. Bunun başlıca sebebi, yapay sinir ağı ile insanı taklit etmenin kolaylığı ve olabilirliğidir.

2.2.2. Yapay Sinir Ağının Yararları

Yapay sinir ağı birçok açıdan kazançlı ve faydalıdır. Birincisi ; probleme kolayca adapte olabilirler, bilgi alabilirler ve öğrenebilirler. Böylece kendileri için hazırlanan datalardan, çözümleri bulabilir. Bu işlemi de genelde bilgiler arasındaki ilişkileri (bağları) kurnazca yakalayarak yaparlar. Bu özellik yapay sinir ağını diğer algoritmalarından farklı kılar çünkü, bu sistemde kurallar programcının önceliklerine göre belirlenmez ve programcıya bağlı kalınmaz. Yapay sinir ağı, bilgiler arası ilişkileri arayarak uygulama geliştirme zamanını kısaltır. Ayrıca eksik çözümler de, bu sistemle tam çözüme kavuşturulabilir.

İkincisi; genellemedir. Daha önceden sistemin eğitilmiş olduğu konularda, eksik bilgi verilerek sonuç istenirse, sistem bunu geneller veya tamamlar.

Üçüncüsü; yapay sinir ağının doğrusal olmayışıdır (nonlinear). Bu da bütün sonuçların her bir girdiyle etkileşim içerisinde olduğu manasına gelir. Doğrusal bir sistemde girdilerin biri değiştirildiğinde, çıktı bu değişmeye bağımlı olarak değişir ve bu etki sadece değiştirilen girdinin değeri ile bağıntılıdır. Çünkü girdiler seri olarak işlenir. Doğrusal olmayan sistemlerde ise çıktılar bütün girdilere bağımlıdır ve bir girdi değiştirildiğinde bu değişiklik çıktıda önemli bir değişikliğe sebep olmaz. Çünkü çıktı bütün girdilerin paralel işlenmesi ile elde edilir. Gerçek hayattaki sistemler genelde doğrusal değildir. Örneğin; ülke ekonomisindeki finansal durumlar, faiz oranları, fiyatlar, kurlar, borsa karmaşık bir etkileşim içerisinde dirler. Dolar fiyatının artışı, aldığımız ekmeğin fiyatının değişmesinde veya faiz oranlarının artmasında etkili olabilir.

Dördüncüsü ; yapay sinir ağı paralel işlemde son derece başarılıdır. Bir çok benzer ve birbirlerinden bağımsız işlemler aynı anda simüle edilerek işlenebilir. Aynı zamanda, paralel bazlı donanımlar bildiğimiz klasik mikro işlemlerden binlerce kez daha hızlıdır.

2.2.3. Uygulama Çeşitleri

Yapay sinir ağını kullanan desen tanıma algoritmaları genel olarak ikiye ayrılırlar. Bunlardan birincisi sınıflama, ikincisi de bir işlemde fikir yürütebilmedir.

Diğer kullanım amaçları ise kısaca şöyledir; veri sıkıştırma, özellik tanıma ve istatistiksel gruplandırmalardır.

Burada herbir temel özellik (istem), spesifik uygulamalar geliştirmeye neden olur. Mesela, kamera ile kombine edilmiş bir yapay sinir ağı sistemi, patates yetiştirebilir, posta kodlarını okuyabilir, parmak izlerini eşleştirebilir. Veya mikrofonla kombine edilmiş bir sistem, bir motorun sorununu anlayabilir, su altı seslerini tanımlayabilir veya titreşimi egale eden sinyal üretebilir.

Desen tanıma üzerine yapılan uygulamalara birkaç örnek verelim; Sharp Corp. Nara, Japan & Veri-Fone Inc. ve diğerleri tarafından yapılan OCR (Optical Character Recognition - Karakter Tanıma)'lardır. Özellik tanıma alanında ise; Londra NeurForeCasting merkezi tarafından yapılan finans tahmin uygulamasıdır. Bu uygulama; borsa, enflasyon, para hacmi, GSMH gibi, ekonomi ile ilgili olayları takip ederek, gelecek finans durumu hakkında tahminde bulunabilmektedir. Bir diğer uygulama ise Fujitsu LTD. & Kawasaki firmaları tarafından yapılan işlem kontrol sistemleridir.

OCR'larda karşılaşılan zorluklar vardır. Birinin el yazısı diğer kişiye göre çok daha farklı olabilir veya o anki duruma göre yazı hızlı yazılmıştır, harfler üst üste binmiştir. Kağıt yıpranmış veya ıslanmış olabilir.

Bilindiği üzere Türk alfabesinde 29 karakter vardır. Bunlara özel karakterleride eklersek ancak 100'e ulaşır bu sayı. Bir de Japon alfabesine bakalım. Genel olarak iş hayatında kullanılan karakter sayısı 3000'i geçkindir. Aynı zamanda Japon alfabesi Latin alfabesine nazaran çok daha karmaşık yapıdadır. Ufak bir ayrıntı bile harfin değişmesine sebep olabilir. Durum böyle iken, piyasada, saniyede 40 karakteri %96-99 doğruluk oranıyla tanıyan OCR'lar vardır.

2.2.4. MICR (Manyetik Karakter Tanıma)

MICR rakamları, çeklerde hesap numaralarını belirtmekte kullanılır. Manyetik okuma için dizayn edildiklerinden sabit bir font kullanılmaktadır. Bu da esneklik kaybetmesine neden olmaktadır.

MICR ile hazırlanan çekleri kabul için bankalarda uzun süredir gerekli sistemler bulunmaktadır. Bu çekler sadece bankalarda değil, toptan satış yapan yerlerde de kullanılmaktadır. Ancak ilk seferde %80 doğruluk oranı olduğu için büyük kayıplara yol açmaktaydı. Ancak elle iş yapmak da çok vakit istemektedir.

Ayrıca MICR pahalı bir sistem gerektirmektedir. Dolayısıyla ucuz, süratli, küçük, kullanımı kolay ve doğru okuyan bir sistemin geliştirilmesi şart olmuştur. 1992 yılında, bir iş otomasyon firması olan VeriFone, toptancılara yönelik bir MICR okuyucu çıkardığını duyurdu. VeriFone'un belirttiğine göre, bu firmanın çek okuyucusu olan Onyx'in istenilen hedeflere ulaşmadaki püf nokta, Onyx'e yerleştirilmiş yapay sinir ağı sistemiydi. Onyx, çizilmiş, karalanmış çekleri bile %99.6 doğruluk oranı ile okuyabilmekte. Onyx'in kalbini ise 1986'da kurulan Synaptics firmasının ürettiği özel çipler oluşturuyor. Bu çip retina tipi bir optik algılayıcı ile MICR karakterlerini sınıflandırmak için geliştirilmiş olan bir yapay sinir ağı algoritmasını birleştiriyor. Yüksek yoğunluk ve hız için analog VLSI kullanan I-1000 çipi saniyede 1000 karakteri sadece 10 mW gibi az bir enerji harcayarak tanıyabiliyor. Bu küçük, ucuz ve düşük güçte çalışan CMOS dizaynı, VeriFone'un sisteminin aranan özellikleri ucuza sağlamasına yardımcı oldu.

Synaptics'in çipi, çipin retinası üzerinde görüntüden 400 pixel düşüren, LED ve lensten oluşan bir alt sistemi kullanıyor ve görüntü sürekli olarak çipe girdi olarak giriş yapıyor. Görüntüyü, çipte çalışan iki adet yapay sinir ağı girdi olarak kullanıyor. Bunlardan biri okunacak karakterin hangi düşey pozisyonda okunacağına karar veriyor. Yani eğik karakterleri düzelterek normal hale getirir. Bu belirlendikten sonra ikinci sinir ağı devreye girerek o pozisyondaki karakteri sınıflandırır. İlk sinir ağının 40 adet, ikinci sinir ağının ise 14 adet çıktı sınıfı vardır. Yani birincisi 40 düşey pozisyondan birisini, ikincisi ise 14 karakterden birisini sonuç olarak veriyor.

Burada yapay sinir ağının birçok faydası var. Örneğin, bu sistem her karakteri bütün bir resim olarak gördüğü için, çekteki kırışma, çizik v.s. etkenlerin olumsuz etkisini en aza indiriyor. İkincisi, görüntü sürekli olarak incelendiği için çekin okuyucudan hangi hızla geçtiğinin bir önemi kalmıyor. Bu da pozisyon belirleyici, merdane, motor gibi bazı donanımları gereksiz kılıyor ve sistemi basitleştiriyor. Aynı zamanda da kullanım kolaylığı sağlanıyor. Bu örnek de gösteriyor ki klasik yöntemler gün geçtikçe yerlerini yapay sinir ağı algoritmaları ile çalışan özel amaçlı çiplere bırakacaklardır.

2.2.5. İşlev Belirleme

Yapay sinir ağının diğer bir ana uygulama dalı olan işlev belirlemede sinir ağı çok kullanışlıdır çünkü gerçek dünyadaki sistemlerin bir modeli gibi çalışır. Bu model tanımladığı sistemde tahmin yapar ve kontrol eder.

Bu türden uygulamaların büyük bir potansiyeli vardır. Plastiğin kalıba dökümü, güç şarjı tahmini, resim ve seslerdeki kirlenmenin giderilmesi gibi hemen hemen bütün endüstriyel işlemlere uyarlanabilir. Ayrıca sahtekarlıkların belirlenmesi, güvenilirlik testi, hava tahmini gibi istatistiğe dayanan birçok sistem de sinir ağı uygulaması olmaya adaydır.

Önceden de belirttiğimiz gibi, sinir ağlarının kullanışlı birçok özelliği arasında adapta olabilirlik ve doğrusal olmama da vardır.

Her iki özellikler, gerçek dünyadaki işlev belirleme işlemlerine oldukça uygundur.

Örneğin;

Bir sistemin işleyişinde etkili olan parametreler bilinsin ama aralarındaki denklemler belli olmasın. Böyle bir sistemi temsil eden fonksiyonları bulmak oldukça zordur.

Ancak sinir ağlarının böyle bir sistemi temsili için ihtiyaç olan şey, sistemin hangi parametre değerlerine karşılık, ne sonuçlar ürettiğine dair örneklerden ibarettir. Bu da sinir ağlarının adapte olabilirlik özelliği sayesinde olur. Sinir ağları sistemleri teorik olarak anlamamanın değil onların deneylerdeki davranışlara göre karar vermenin kullanışlılığını kullanırlar. Onların tecrübelerden öğrenebilme özellikleri gerçek dünyada çok değerlidir. Doğrusal olmaması da çok önemlidir. Çünkü dünyadaki birçok sistem doğrusal işlemez ve bilinmeyen ilişkiler mevcuttur. Doğrusal metotlar istenen sonucu üretmediklerinden sadece matematiksel önemi vardır. Doğrusal metotlar, doğrusal olmayan problemlere sadece belli bir aralık doğru çözüm verirler. İşte burada sinir ağları sinir ağları meseleye daha geniş bir açıdan bakmayı öneriyor.

İşlev belirleyici olarak çalışan sinir ağına iki örnek; University College London'da geliştirilen finansal tahmin uygulaması, diğeri ise kalıp döküm işleminde hatalı kalıplamayı kontrol eden Fujitsu firmasının sunduğu bir kontrol uygulaması. Bunların her ikisi de geleneksel metotlardan çok daha iyi sonuç vermektedirler. Aşağıda yapay sinir ağlarında algoritmalarında kullanılan terimleri açıklayacağız.

Giriş Tabakası (Input Layer): Bilgilerin girişi için hazırlanmış pasif bir tabaka. Bilgiler bu tabaka kullanılarak ağı verilir ve bu tabakadaki düğüm sayısı giriş verisi sayısı ile eşittir.

Gizli tabaka (Hidden Layer): Ne girdi ne de çıktı tabakasına bağlı kısım. Ağı başarısını belirleyen kısım da denebilir. Bu tabakadaki düğüm sayısı tasarımcıya bağlı olmakla birlikte literatürde probleme dayalı olarak bazı öngörüler yapılmıştır.

Çıktı tabakası (Output Layer): Hazırlanan sonuçlar bu tabakadan alınır. Çıkış tabakasındaki düğüm sayısı problemdeki çıkış sayısına eşittir

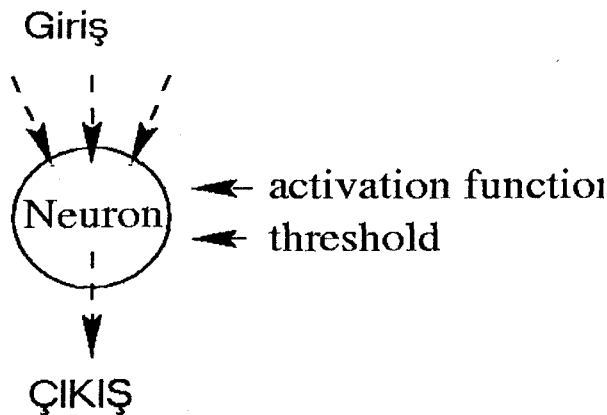
Düğüm (Node): Tek bir sinir hücresine verilen ad, bir bağlantı.

Ağırlık (Weight): Nodalar arasındaki bağların, ayarlanabilir ağırlık değerleri.

Şekil, biçim, desen tanıma (Pattern Recognition)

Geri besleme (Feedback): Bilgilerin geri gönderilip sistemin kendisini daha kusursuz hale getirmesini sağlar.

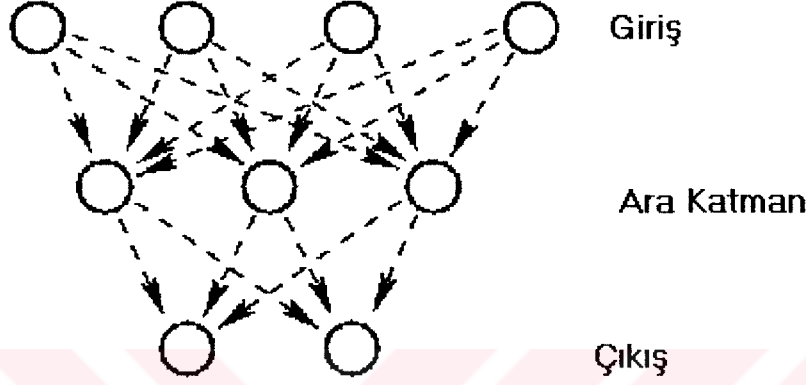
Perceptron : Analog manada bir girdinin iki veya daha fazla gruplara ayrılmasını sağlayan en küçük birim (node).



Şekil 2.4. Nöron

2.3. Temel Yapay Sinir Ağları Algoritmaları

Literatürde pek çok yapay sinir ağı yapısına rastlamak mümkündür. Bu çalışmada bunlardan en temel olan Backpropagation yapısı ve bunun en hızlı varyantı olan Levenberg -Marquart Algoritmasına ve çok yeni olan Radial Basis Functions Networks (RBF) değinilecektir.



Şekil 2.5. Çok katmanlı bir yapay sinir ağı.

2.3.1. Backpropagation

En temel yapay sinir ağı şekillerinden biri olan backpropagation' ın da en önemli özelliği sistemin hatayı Least Mean Square (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 hidden (saklı) katmanlar olmak üzere üç 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 sayı bunlardaki nöron sayısı ile çok esnek bir öğrenme modelinin kurulabilir olmasıdır.

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 düzeltereklerini bilmemeleriydi. Bu soruya cevap Rumelhart [Rumelhart, 1986], Hinton ve diğerleri tarafından geldi.

Bu çözümün ana fikri ise ara katmandaki birimlerdeki hata çıkış katmanındaki hataların geriye doğru yayılımı ile bulunabileceği idi.

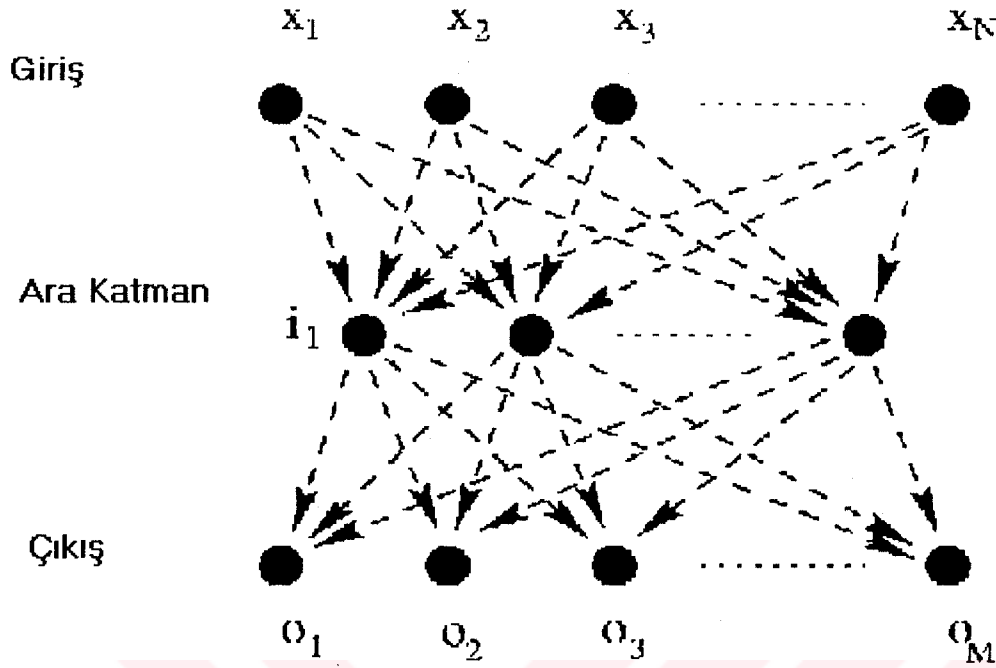
Buna göre birimler arasındaki ağırlık(şiddet) iyileştirmesi aşağıdaki hata formülüne göre olmaktadır.

$$E = \frac{1}{2} \sum_k \delta_k^2 \quad (2.1)$$

Buna göre de

Ağırlık iyileştirme=Öğrenme katsayısı x Bölgesel gradyant x giriş sinyali (2.2)

İşte bu yüzden (2.2) formülüne backpropagation denmiştir.



Şekil 2.5. Backpropagation.

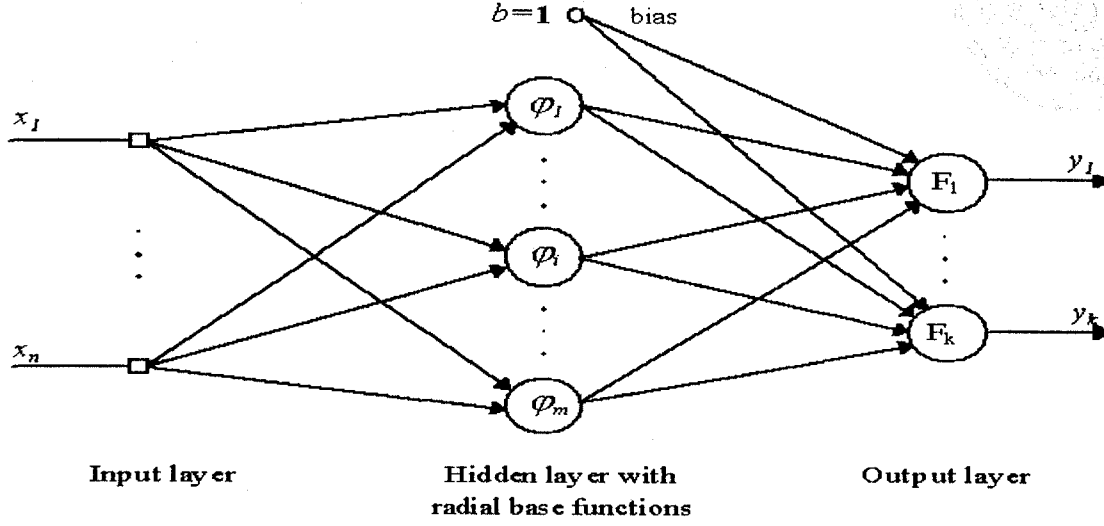
Bu algoritmanın pek çok değişik varyantı mevcuttur. Yapılan çalışmalar Levenberg-Marquart algoritmasının en hızlı olduğunu göstermiştir.

2.3.2. Radial Basis Functions Yapay Sinir Ağları

RBF öteden beri fonksiyon atamada kullanıla gelen ve pek çok başarılı sonuçlar elde edilen bir metoddur. Dolayısıyla bu metodun yapay sinir ağı ile kullanılması gayet iyi sonuçlar doğurmaktadır.

RBF yapay sinir ağı diğer temel yapay sinir ağı çeşitlerinden farklılık arz etmektedir bunun da en önemli sebebi genelde yapay sinir ağı sistemlerinde kullanılan aktivasyon fonksiyonu olarak kullanılan fonksiyonların aksine burada gaus fonksiyonu kullanılmıştır.

RBF yapay sinir ağı 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 yapı hemen hemen bütün yapay sinir ağlarında aynıdır. Önemli olan ara katmandaki nöron sayısı vede aktivasyon fonksiyonu olarak kullanılan gaus fonksiyonunu merkezinin neresi olacağıdır.



Şekil 2.6. RBF yapay sinir ağı.

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

RBF yapay sinir ağında 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 ortalama gruplama (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.

$$E = \sum_{i=1}^N \sum_{j=1}^m M_{ij} \| x_i - c_j \| \quad (2.3)$$

1. Adım. Fonksiyon merkezlerini ata

Başlangıçta merkezler rasgele 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

$$\text{Her grup için, } c_j = \frac{1}{m_j} \sum_{x_i \in \text{group } j} x_i$$

m_j grup j deki veri sayısı.

4. Adım. Hiçbir değişikliğe gerek kalmama kadar 2. Adıma dön**3-Kohonen yapay sinir ağı kullanılarak seçim.**

Burada Kohonen algoritması olarak bilinen yapay sinir ağı kullanılarak veriler gruplandırılırlar.

4-Minimumların maksimumu (MaxiMin Dictance) Algoritması

Diğer bir metot olan MaxiMin algoritmasında ki adımlar aşağıdaki gibidir.

1. Adım. İlk fonksiyon merkezlerini ata

Rasgele seçilen merkezler

2. Adım. Her bir veri için minimum mesafeyi hesapla

$$\text{Her bir veri } x_i, d_i = \min_j \|x_i - c_j\|$$

3. Adım. En büyük minimum mesafeli veriyi seç

$$D = d_{j^*} = \max_i \{ d_i \}$$

eğer $D > \frac{1}{2} \{D_{\text{önceki}}\}$ yada $D > D_{\min}$ ($D_{\min}$ belirtilen minimum mesafe

veri x_j yeni bir merkez olarak ata $c_j = x_j$.

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

2.4. Yapay Sinir Ağları İle Kontrole Giriş

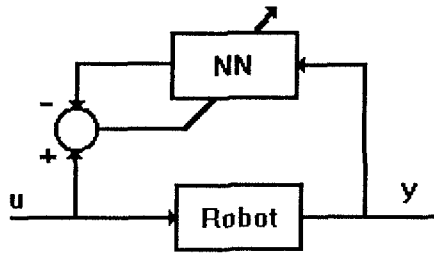
Bu konudaki ilk çalışmalar bir insanı taklitle başlar. İlk olarak widrow ve Smith [widrow ve Smith, 1964] bir insan kontrol elemanın yerine yapay sinir ağı kontrol elemanı koymayı düşünmüştür.

Bunu da ters sarkaç problemine uygulamışlardır. Aynı metodun bir kontrol elemanı tasarlamamanın zor olacağı durumlarda kullanılabilirliği de öngörülmüştür. Uygulamada kullanılan pek çok uygulama yapay sinir ağı 'yı ters dinamik olarak kullanmaktadır (Bkz . Kawato [Kawato, 1988], Orthega ve Psaltis [Psaltis, 1988]). Bu tür yaklaşımlar özellikle sistemin çok hızlı bir dinamiği olduğu zaman kullanışlıdır.

2.4.1. Ters Dinamik Modelleme Yöntemi (DIMA)

Bu metotta amaç sistemin ters dinamiğini yapay sinir ağı ile öğrenmek ve daha sonra bu ters dinamik öğrenmiş yapay sinir ağı kontrol elemanı olarak kullanmak üzere sisteme adapte etmek DIMA'nın amacıdır.

Burada öncelikle sistem ile ilgili bilgiler toplanır. Daha sonra bu bilgiler kullanılarak sistemin ters dinamiği öğretilir ve bu yapay sinir ağı sistemde birincil kontrol elemanı olarak kullanılır.

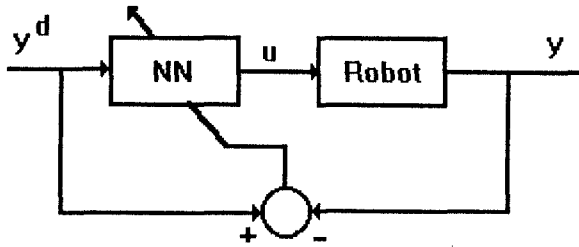


Şekil 2.7. Ters dinamik modelleme yöntemi (DIMA).

2.4.2. Ters Dinamik Modelleme Yapısı Tip II(DIMA II)

Ters Dinamik Modelleme Yapısı (Direct Inverse Modelling Architecture (DIMA II)) DIMA ile benzer bir mantıkta çalışır yalnızca fark burada sistemin tersinin öğrenilmesi on-line yani sistem çalışırken olur.

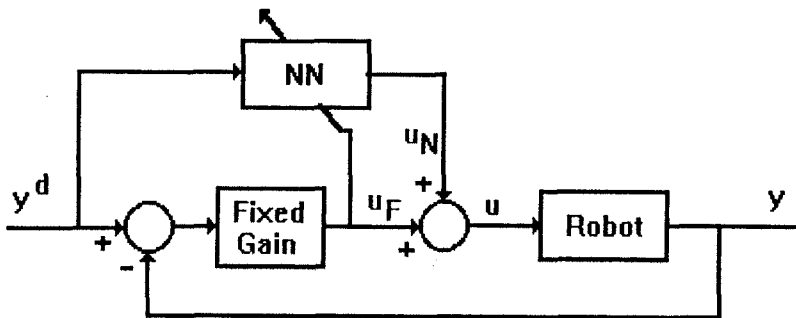
Bu metod Psaltis [Psaltis, 1987] tarafından 1988 yılında önerilmiş ve orijinal makale "specialised inverse learning " adıyla yayınlanmıştır.



Şekil 2.8. Ters dinamik modelleme yöntemi tip II (DIMA II).

2.4.3. Hata Geri Beslemeli Öğrenme Yöntemi (FELA)

Şekil 2.9 de görülen Hata Geri Beslemeli Öğrenme (Feedback Error Learning Architecture(FELA)) Kawato [Kawato, 1988] tarafından önerilmiş bir yapıdır. Bu yapıda yapay sinir ağı bir ileri beslemeli(feed-forward) kontrol elemanı olarak kullanılmış. Bunu yaparken de sistemde var olan ikincil kontrol referans olarak alınmıştır.



Şekil 2.9. Hata geri besleme öğrenmeli yapı (FELA).

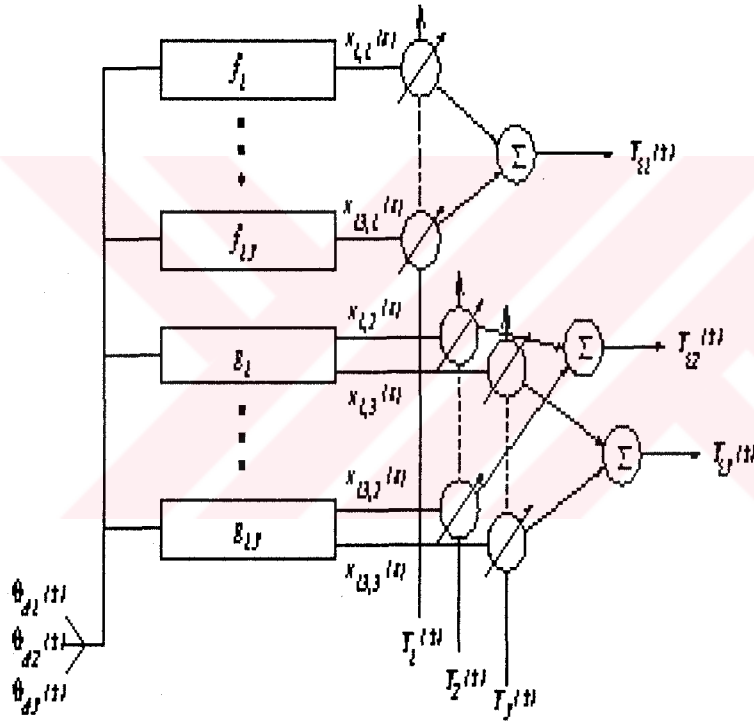
Sistemin toplam kontrol girdisi :

$$u(t) = u_F(t) + u_N(t) \quad (2.1)$$

olarak hesaplanır.

Kawato[1]' ya göre sistemin aşağıda belirtilen üstünlükleri vardır.

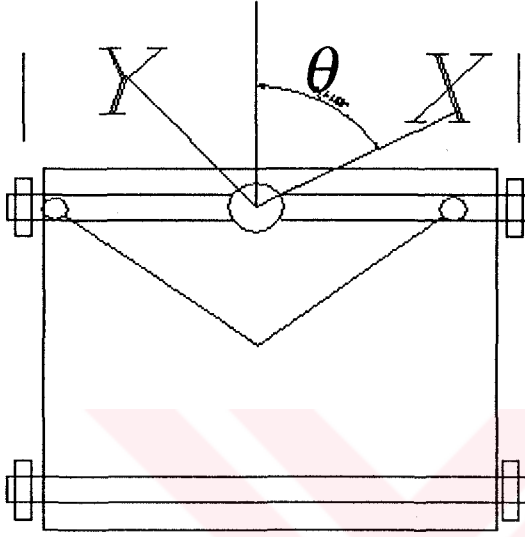
1. Öğrenme ve kontrol aynı anda yapılır
2. Hatanın geriye yayılmasına (backpropagation) gerek yok
3. Öğrenme hedefe yönelmiş (goal-directed) olduğu için daha iyi sonuç verir.



Şekil 2.10. Detaylı kawato modeli.

3. OTOMATİK YÜK ARABASI DİNAMİK VE KİNEMATİK DENKLEMLERİ

3.1. Giriş



Şekil 3.1. OYA modeli.

Sağ tekerin lineer hızı;

$$v_r = R \cdot w_r \quad (3.1)$$

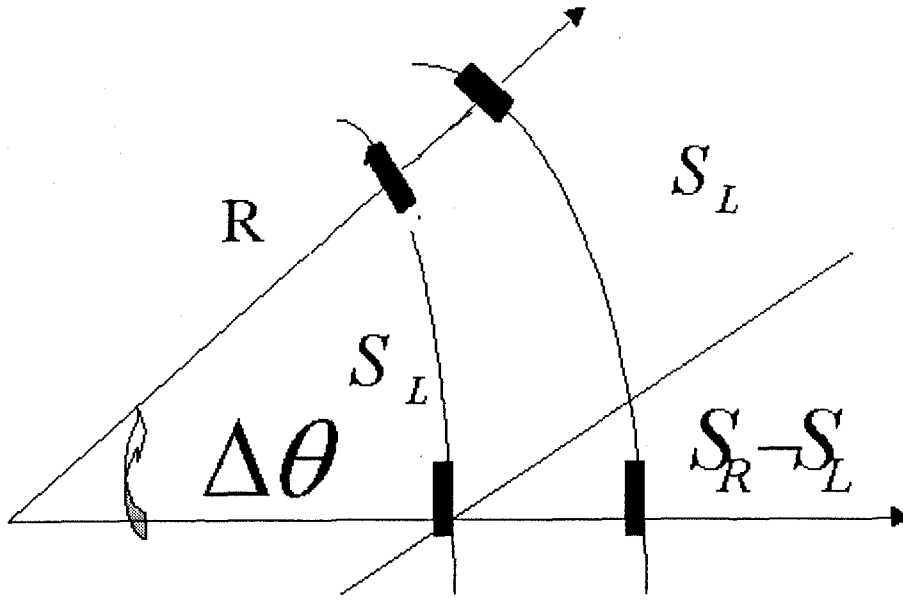
Sol tekerin lineer hızı;

$$v_L = R \cdot w_L \quad (3.2)$$

Ağırlık merkezinin hızı;

$$v = \frac{v_r + v_L}{2} = \frac{R}{2} (w_R + w_L) \quad (3.3)$$

Şimdi Hareket denklemlerini bulmak için anlık bir açı değişimini ele alalım



Şekil 3.2. Anlık hareket açısı.

Δt zaman aralığını ele alalım

$$\Delta t = t_2 - t_1 \quad (3.4)$$

Buna dayanarak yayların boyu hesaplanabilir

$$S_R = \Delta t \cdot v_R \quad (3.5)$$

$$S_L = \Delta t \cdot v_L \quad (3.6)$$

$$\Delta \theta \cdot L \cong S_R - S_L = \Delta t \cdot (v_R - v_L) = \Delta t \cdot R \cdot (w_R - w_L) \quad (3.7)$$

Her iki tarafı da Δt ile bölersek

$$\frac{\Delta \theta}{\Delta t} = \frac{R}{L} (w_R - w_L) \Rightarrow \frac{d\theta}{dt} = \frac{R}{L} (w_R - w_L) \quad (3.8)$$

$$\frac{dx}{dt} = v_x = v \cdot \cos(\theta) = \frac{R}{2} (w_R + w_L) \cos(\theta) \quad (3.9)$$

$$\frac{dy}{dt} = v_y = v \cdot \sin(\theta) = \frac{R}{2} (w_R + w_L) \sin(\theta) \quad (3.10)$$

(3.9) ve (3.10) un entegrali alınırsa x ve y pozisyonları elde edilebilir.(3.3),(3.8),(3.9) denklemleri aracın kinematik denklemlerini verir.

3.2. Otomatik Yük Arabasının Dinamik Denklemleri

Sistemin dinamik denklemlerini elde etmek için aracın tekerlerini süren motor mekanizmasından faydalanılır.

R_a = Armatür direnci

L_a = Armatür indüktansı

i_a = Armatür akımı

ω_m = Motor açısal hızı

θ_m = Motorun açısal deplasmanı

$T_m, T_1, T_2 = \text{Torque}$

J_m = Motor şaftına gelen toplam atalet momenti

J_2 = Teker açısal momenti

F_{c2} = Sürtünme Katsayısı

B = Viskoz sürtünme katsayısı

$\frac{N_1}{N_2}$ = İletim oranı

Öncelikle DC motor dinamiği ve tekerlerin dinamiği ayrı ayrı incelenebilir ve daha sonra bunlar bir araya getirilebilir.

DC motor dinamiği [23-24] öncelikle ele alınırsa aşağıdaki denklemler elde edilebilir.

$$e_a = L_a \frac{di_a}{dt} + R_a i_a + e_b \quad (3.11)$$

elektromotor kuvveti aşağıdaki gibi bulunur.

$$e_b = K_b \frac{d\theta}{dt} \quad (3.12)$$

K_f, i_f, K_1 katsayıları bir katsayı altında toplarsak

$$K_f i_f K_1 i_a = K_a i_a = T_m \quad (3.13)$$

Şimdi motor tekerinin dinamiğini ele alırsak. Motordan vites kutusuna aktarılan güce bakarsak

$$T_m(t) = J_m \frac{d^2\theta_m}{dt^2} + B_m \frac{d\theta_m}{dt} + T_1(t) \quad (3.14)$$

$T_1(t)$ aşağıdaki gibi tanımlanırsa

$$T_1(t) = \frac{N_1}{N_2} T_2(t) = \left(\frac{N_1}{N_2} \right)^2 J_2 \frac{d^2\theta_m}{dt^2} + \left(\frac{N_1}{N_2} \right)^2 B_2 \frac{d\theta_m}{dt} + \frac{N_1}{N_2} F_{c2} \frac{\dot{\theta}_2}{\left| \dot{\theta}_2 \right|} \quad (3.15)$$

Burada F_{c2} sabit terimdir.

(3.14) ve (3.15) denklemlerini birleştirirsek

$$T_m(t) = J_m \frac{d^2\theta_m}{dt^2} + B_m \frac{d\theta_m}{dt} + \left(\frac{N_1}{N_2} \right)^2 J_2 \frac{d^2\theta_m}{dt^2} + \left(\frac{N_1}{N_2} \right)^2 B_2 \frac{d\theta_m}{dt} + \frac{N_1}{N_2} F_{c2} \frac{\dot{\theta}_2}{\left| \dot{\theta}_2 \right|} \quad (3.16)$$

$\ddot{\theta}$ ve $\dot{\theta}$ katsayılarını birlikte ele alırsak yukarıdaki denklemi tekrar yazalım

$$T_m(t) = J_{eq} \frac{d^2\theta_m}{dt^2} + B_{eq} \frac{d\theta_m}{dt} + F_{eq} \text{sign}(\dot{\theta}_2) \quad (3.17)$$

burada

$$J_{eq} = J_m + \left(\frac{N_1}{N_2} \right)^2 J_2 \quad B_{eq} = B_m + \left(\frac{N_1}{N_2} \right)^2 B_2 \quad F_{eq} = \left(\frac{N_1}{N_2} \right) F_{c2} \quad (3.18)$$

$$\frac{\dot{\theta}_2}{\left| \dot{\theta}_2 \right|} = \text{sign}(\dot{\theta}_2) \text{ dir.}$$

Yeni sabitleri de aşağıdaki tanımlarsak.

$$\begin{aligned}
 \alpha_1 &= \frac{B_{eq}}{J_{eq}} \\
 \alpha_2 &= \frac{F_{ceq}}{J_{eq}} \\
 \alpha_3 &= \frac{1}{J_{eq}}
 \end{aligned}
 \tag{3.19}$$

dır

Tekerin dinamik denklemlerini aşağıdaki gibi elde edebiliriz.

$$\frac{d^2\theta_m}{dt^2} = \alpha_1 \frac{d\theta_m}{dt} + \alpha_2 \text{sign}(\dot{\theta}_m) + \alpha_3 T_m
 \tag{3.20}$$

Simülasyonda (3.19) ve (3.20) denklemleri kullanılacaktır.



3.3. Durum Denklemleri

Daha önceki bölümlerde elde edilen denklemler durum denklemlerini elde etmek için kullanılacaktır.

Kinematik denklemler.:

$$\begin{aligned} V &= \frac{R}{2}(w_R - w_L) \\ \dot{\theta} &= \frac{R}{L}(w_R - w_L) \end{aligned} \quad (3.21)$$

Ters Kinematik Denklemler.:

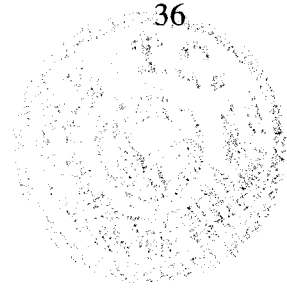
$$\begin{aligned} w_R &= \frac{1}{R}v + \frac{L}{2R}\dot{\theta} \\ w_L &= \frac{1}{R}v - \frac{L}{2R}\dot{\theta} \\ \dot{w}_R &= -\alpha_1 w_R - \alpha_2 \text{sign}(w_R) + \alpha_3 T_R \\ \dot{w}_L &= -\alpha_1 w_L - \alpha_2 \text{sign}(w_L) + \alpha_3 T_L \end{aligned} \quad (3.22)$$

v nin türevini alır ve (3.22) yi kullanırsak

$$\dot{v} = \frac{R}{2}(\dot{w}_R + \dot{w}_L) = \frac{R}{2}[-\alpha_1(w_R + w_L) - \alpha_2(\text{sign}(w_R) + \text{sign}(w_L)) + \alpha_3(T_R + T_L)] \quad (3.23)$$

Aynı şeyleri θ için yaparsak

$$\ddot{\theta} = \frac{R}{L}(\dot{w}_R - \dot{w}_L) = \frac{R}{L}[-\alpha_1(w_R - w_L) - \alpha_2(\text{sign}(w_R) - \text{sign}(w_L)) + \alpha_3(T_R - T_L)] \quad (3.24)$$



Durum uzayındaki değişkenleri aşağıdaki gibi seçersek.

$$\begin{aligned} x_1 &= v \\ x_2 &= \dot{\theta} \\ x_3 &= \theta \end{aligned} \quad (3.25)$$

$$\dot{x}_1 = -\alpha_1 x_1 - \alpha_2 \frac{R}{2} (\text{sign}(g_1(x)) + \text{sign}(g_2(x))) + \alpha_3 \frac{R}{2} (T_R + T_L) \quad (3.26)$$

$$\dot{x}_2 = -\alpha_1 x_2 - \alpha_2 \frac{R}{2} (\text{sign}(g_1(x)) - \text{sign}(g_2(x))) + \alpha_3 \frac{R}{2} (T_R - T_L) \quad (3.27)$$

$$g_1(x) = \frac{1}{2} x_1 + \frac{L}{2R} x_2 \quad (3.28)$$

$$g_2(x) = \frac{1}{2} x_1 - \frac{L}{2R} x_2 \quad (3.29)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\alpha_1 x_1 - \alpha_2 \frac{R}{2} (\text{sign}(g_1(x)) + \text{sign}(g_2(x))) \\ -\alpha_1 x_2 - \alpha_2 \frac{R}{L} (\text{sign}(g_1(x)) - \text{sign}(g_2(x))) \\ x_2 \end{bmatrix} + R\alpha_3 \begin{bmatrix} 1/2 & 1/2 \\ 1/L & -1/L \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_R \\ T_L \end{bmatrix} \quad (3.30)$$

$$x_3 = v \quad x_2 = \dot{\theta} \quad x_1 = \theta \quad (3.31)$$

3.4. Otomatik Yük Arabasının Kinematik Denklemleri

$$\begin{aligned}\frac{dx}{dt} &= v_x = v \cdot \cos(\theta) \\ \frac{dy}{dt} &= v_y = v \cdot \sin(\theta)\end{aligned}\quad (3.32)$$

3.5. Otomatik Yük Arabası Ters Kinematik Denklemleri

$$\begin{aligned}v(t) &= \sqrt{\left(\frac{dx(t)}{dt}\right)^2 + \left(\frac{dy(t)}{dt}\right)^2} \\ \theta(t) &= \arctan\left(\frac{v_y}{v_x}\right)\end{aligned}\quad (3.33)$$

3.6. Referansın Tanımlanması

Bizim referansı $\theta(t)$, $v(t)$ ile belirtmemiz durumunda aracın X_Y ekseninde nasıl hareket edeceği çok açık bir şekilde belli değildir. Gerçek hayattaki en önemli kıstas ise aracın X_Y ekseninde istenen referansı takip etmesidir. Bundan dolayıdır ki biz referansı X_Y düzleminde tanımlayıp($x'(t)$, $y'(t)$) daha sonra da $v'(t)$, $\theta'(t)$ 'ye çevirecek bir metot bulmak zorundayız. Aşağıda buna çözüm olarak önerilen metot açıklanacaktır.

Metod

- 1) x' yi t(zaman)düzleminde tanımla
- 2) $y'(t)$ yi $x'(t)$ biçiminden tanımla

örneğin:

daire için $y'(t) = \sqrt{R^2 - x'^2(t)}$

eğer bu mümkün değilse $y'(t)$ yi istenen referansı verecek şekilde ayrı olarak tanımlamak gerekir.

İstenen bir x ve y değeri verilerek gerekli olan $v'(t)$, $\theta'(t)$ değerlerinin elde edilmesi aşağıdaki gibidir.

3) $v'(t)$ nin hesaplanması

$$v'(t) = \left(\frac{dx'(t)}{dt} \right)^2 + \left(\frac{dy'(t)}{dt} \right)^2$$

4) $\theta'(t)$ nin hesaplanması

$$\theta'(t) = \arctan\left(\frac{v_y}{v_x}\right)$$

Burada

$$v_y = \frac{dy}{dt} \quad \text{ve} \quad v_x = \frac{dx}{dt} \quad \text{dir.}$$

Bu hesaplamalar yapılırken göz önünde bulundurulması gereken husus ise arctanın periyodunun $\Pi/2$ olmasıdır. Bundan dolayı arctan her quadrant için hesaplanmalıdır.

4. TERS DİNAMİK MODELLEME YÖNTEMİ

Ters dinamik modelleme yöntemi (**Direct inverse modelling architecture**) lineer olmayan sistemlerin kontrolünde kullanılan bir metottur. Bu metotta iki adet kontrol elemanı mevcut olup Bunlar

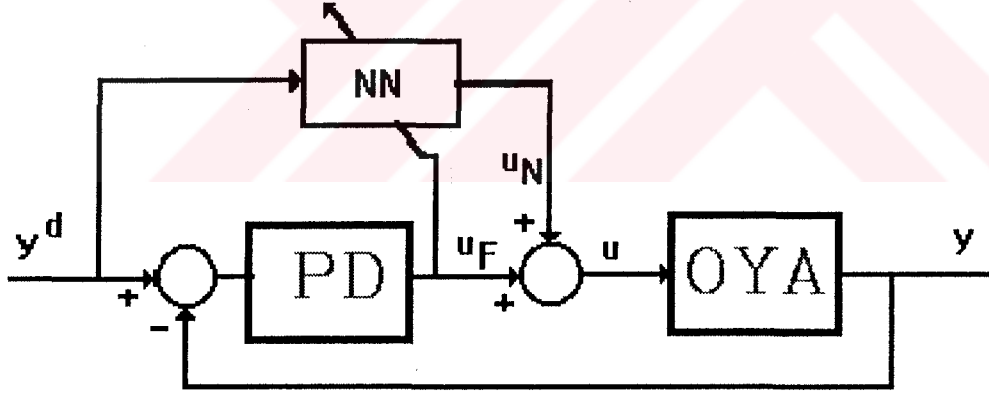
1-Birincil Kontrol elemanı (yapay sinir ağı)

2-İkincil kontrol elemanı

Birincil kontrol elemanı olarak yapay sinir ağı kullanılır. Bu tür bir yapay sinir ağı giriş ile çıkış arasında veri yetersizliği yada uydurulacak fonksiyonun analitik olarak bilinmediği durumlarda bir haritalandırma (mapping) yapar. Bu tür bir uygulama için kaç adet ara katmana sahip kaç nöron kullanılarak oluşturulan bir yapının lazım olduğu analitik olarak tespit edilemez

İkinci kontrol elemanı bir PD kontrolüdür. Bu kontrolün amacı ise sistemde oluşan modelleme hatalarını ve gürültüyü kompanse etmektir.

Bu metodun en önemli özelliklerinden birisi giriş/çıkış datasına ihtiyaç duyması ve sistemin ters dinamiğinin öğretilerek olmasıdır. Gerekli olan veri OYA'nın çeşitli bölgelerde çalıştırılması ile elde edilebilir.



Şekil 4.1. Ters dinamik modelleme yöntemi blok diyagramı.

4.1. İkincil kontrol elemanın tasarlanması:

Eğer $f(x) = f(x^d)$ ise ve $B(x) = B(x^d)$, birincil kontrol elemanı tamamen lineer olmayan terimlerin giderilmesini sağlar. Bunun sonucunda ortaya çıkan ise lineer bir sistem olur. Bu ise bize lineer bir durumda bir PD tasarlıyormuş gibi bir çalışma imkanı verir.

Eğer kontrol işaretine $\hat{u}(t)$ dersek.

PD Kontrol:

$$\hat{u}(t) = [B(x)]^{-1} [-K_v(\dot{x} - \dot{x}^d) - K_p(x - x^d)] \quad (4.1)$$

olur.

Burada K_v , ve K_p belirlenmesi gereken sabitler matrisleridir.

Şekil 4.1 den de görüldüğü gibi sisteme olan giriş bütün kontrol elemanlarından gelen çıkışın bir toplamıdır. Bundan dolayı sisteme olan giriş aşağıdaki gibi olur:

$$\hat{u}(t) = u^d(t) + \hat{u}(t) = [B(x)]^{-1} [\dot{x}^d - f(x^d) - K_v(\dot{x} - \dot{x}^d) - K_p(x - x^d)] \quad (4.2)$$

eğer (4.1)'i (4.2) 'te yerine koyarsak

$$\dot{x}(t) = f(x) + B(x)[B(x)]^{-1} [\dot{x}^d - f(x^d) - K_v(\dot{x} - \dot{x}^d) - K_p(x - x^d)] \quad (4.3)$$

$$\dot{x}^d - f(x^d) = -K_v(\dot{x} - \dot{x}^d) - K_p(x - x^d) \quad (4.4)$$

Hatayı da aşağıdaki gibi tanımlayalım,

$$e(t) = x(t) - x^d(t) \quad (4.5)$$

$$(I + K_v)s e(t) + K_p e(t) = 0 \quad (4.6)$$

(4.6) 'nın Laplace Dönüşümü

$$[(I + K_v)s + K_p]E(s) = 0 \quad (4.7)$$

K_p ve K_v 'nin diagonal matris seçilmeleri genel bir yöntemdir. (4.7) denkleminde bir çözüm bulabilmek için, $[(I + K_v)s + K_p]$ matrisinin determinantı sıfıra eşitlenirse.

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

sonuç ise karakteristik denklem olup aşağıdaki gibidir.

$$[(I + K_v^1)s + K_p^1][(I + K_v^2)s + K_p^2][(I + K_v^3)s + K_p^3] = 0 \quad (4.8)$$

Bu sistemi karalı bir hale getirmek için (4.8) denkleminin kökleri sol tarafta olmalıdır

Bu amaçla parametreler aşağıdaki gibi seçilmiştir.

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

$$K_p^1 = 100 \quad K_p^2 = 100 \quad K_p^3 = 100$$

$$K_v^1 = 0.9 \quad K_v^2 = 0.9 \quad K_v^3 = 0.9$$

4.2. Ters Dinamik Modelleme Yöntemi Simülasyon Sonuçları

Aşağıdaki Kodlar Matlab Simulink ortamında yazılmıştır.

Önerilen yapay sinir ağı Yapısı

Katman Sayısı :3 (1 Ara)

Nöron sayısı:6-4-2

yapay sinir ağı Yapısı: RBF Networks

Simülasyon Parametreleri

alfa1=0.5

alfa2=0.2

alfa3=0.5 1/kgm²

R=0.26 m

L=1.0 m

x1initial=0

x2initial=0

x3initial=1

t0=4.0 ; y0=0.4 *10; x0=0.4 *10; m=2.0*y0/t0;

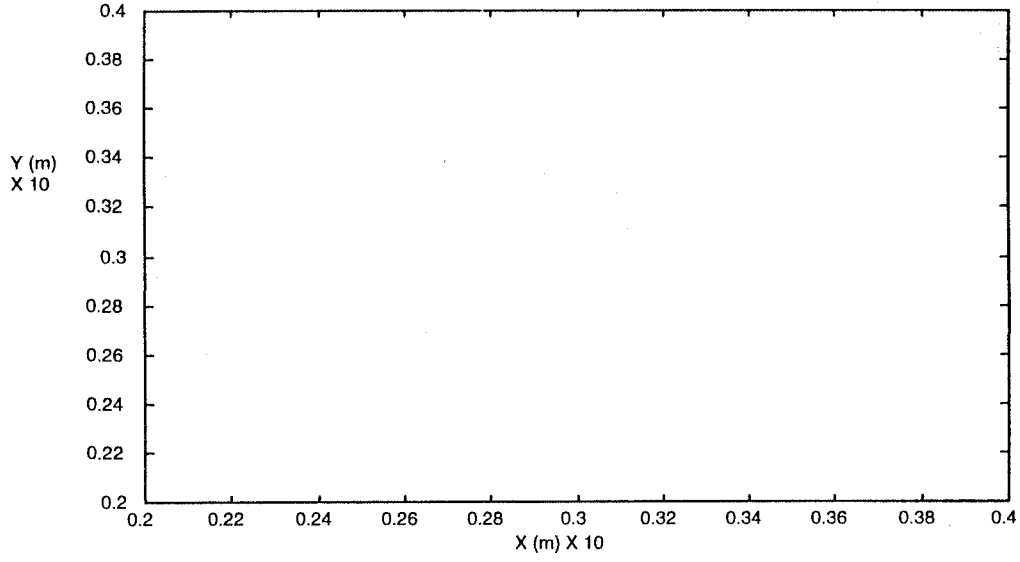
ddx=0.2; ddy=0.2;

tstop=20.0*t0; dT=0.005; Tau=1.0/40.0;

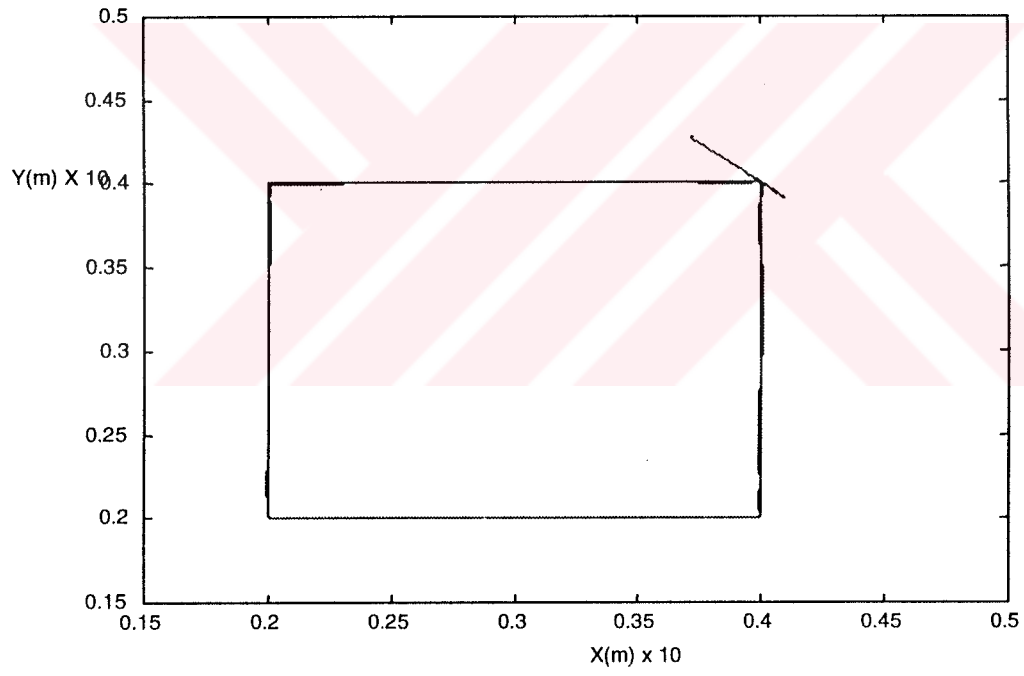
Kontrolör parametreleri

$K_p^1 = 100$ $K_p^2 = 100$ $K_p^3 = 100$

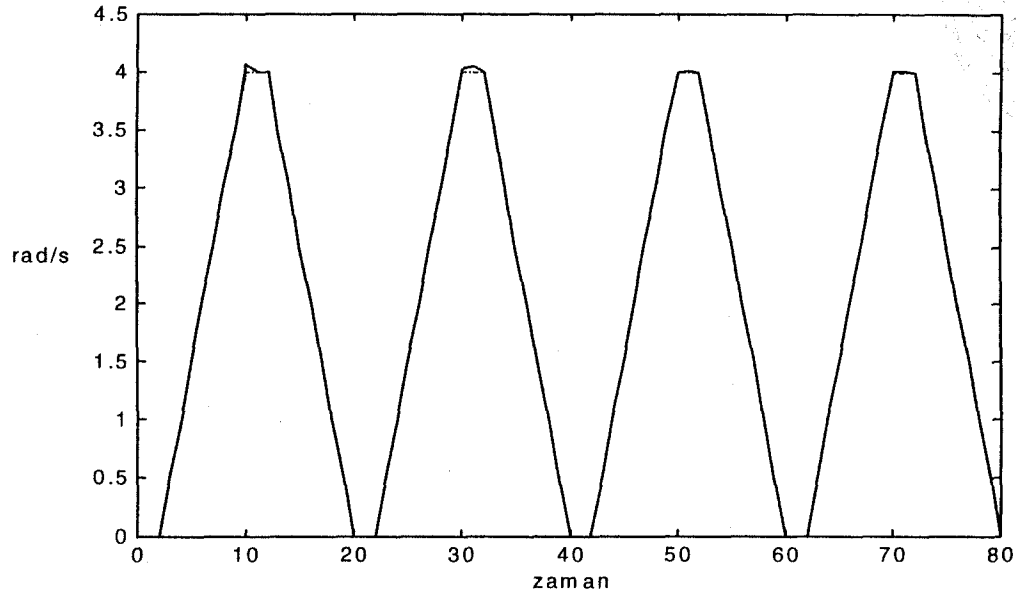
$K_v^1 = 0.9$ $K_v^2 = 0.9$ $K_v^3 = 0.9$



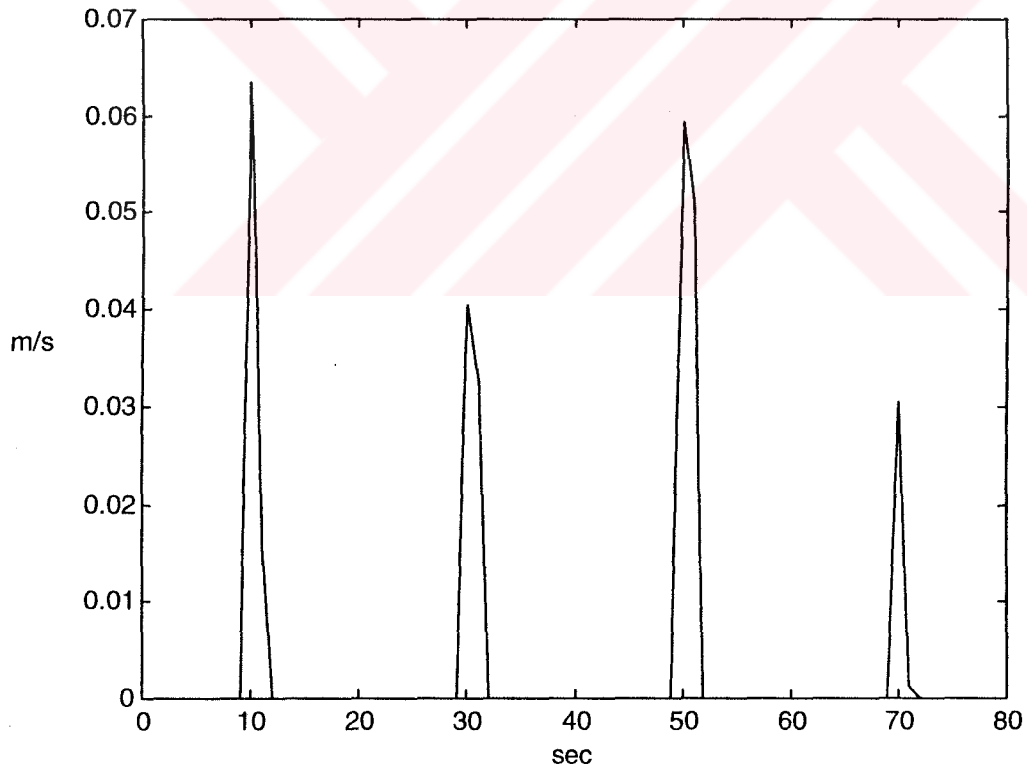
Şekil 4.3. X_Y düzleminde referans.



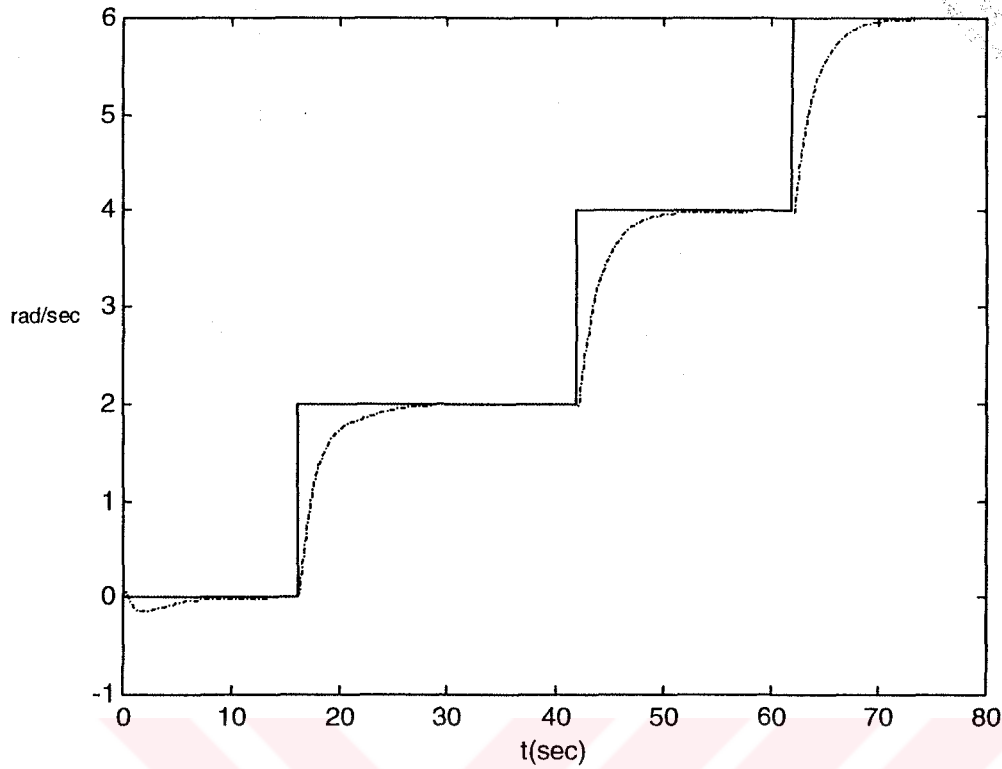
Şekil 4.4 X_Y düzleminde hareket.
(DIMA)



Şekil 4.5. Hız profili.



Şekil 4.6. Hız hatası.

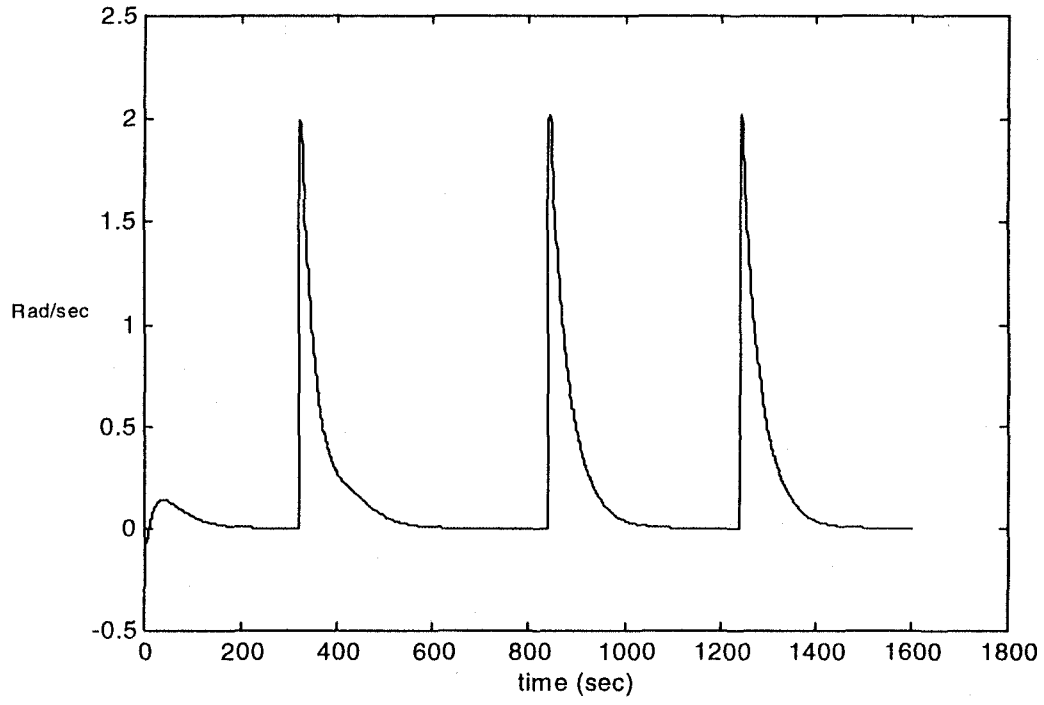


Şekil 4.7. Açı profili.

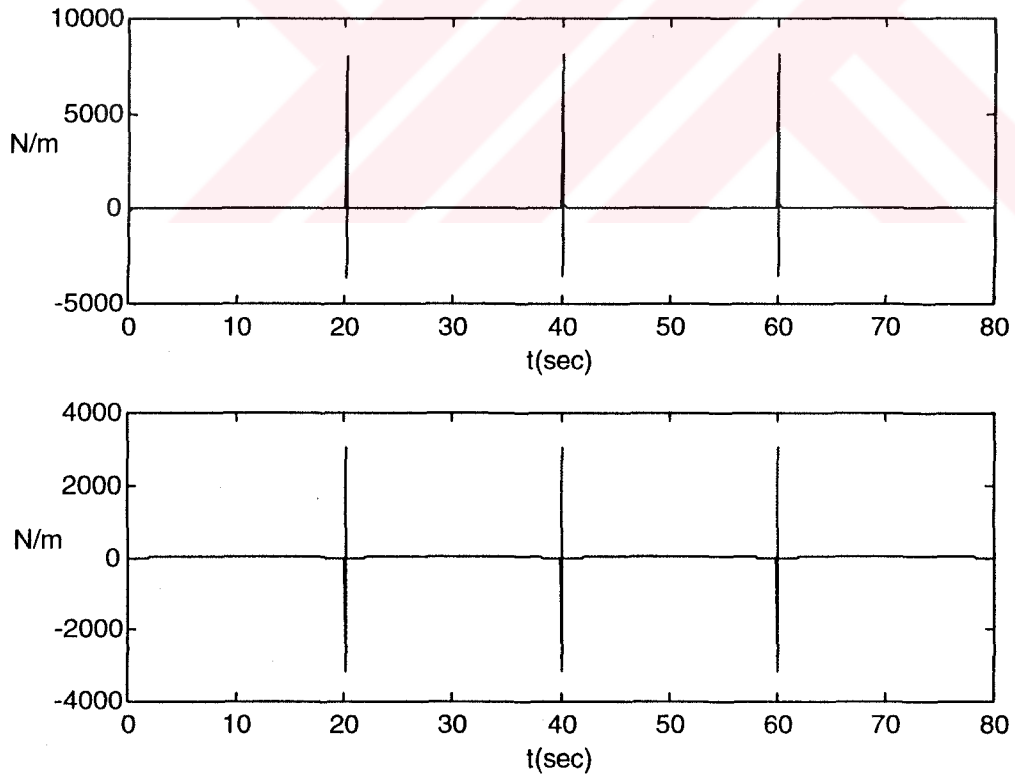
Simülasyon sonuçları göstermiştir ki hız ve açı aynı anda kontrol edilmek istenmesine rağmen sistemin hata oluşturduğu noktalar bu iki denetimin aynı anda yapıldığı yerlerdir. Şeki 4.5 ve şekil 4.6 bize göstermektedir ki sistemin hız denetimi açısından maksimum hatası hız değeri maksimuma ulaşınca olmakta ve una karşılık diğer bölgelerde araç referansı sıfır hata ile takip etmektedir

Açı denetimi bakımından sistem ele alındığı zaman açı profilinin referansı takibi şekil 4.7 de görülmektedir. Bu referansa karşılık gelen hatalar da şekil 4.8 de görülebilir

Sistemi kontrol etmek için uygulanan kuvvetler ise şekil 4.9 da görülebilir.



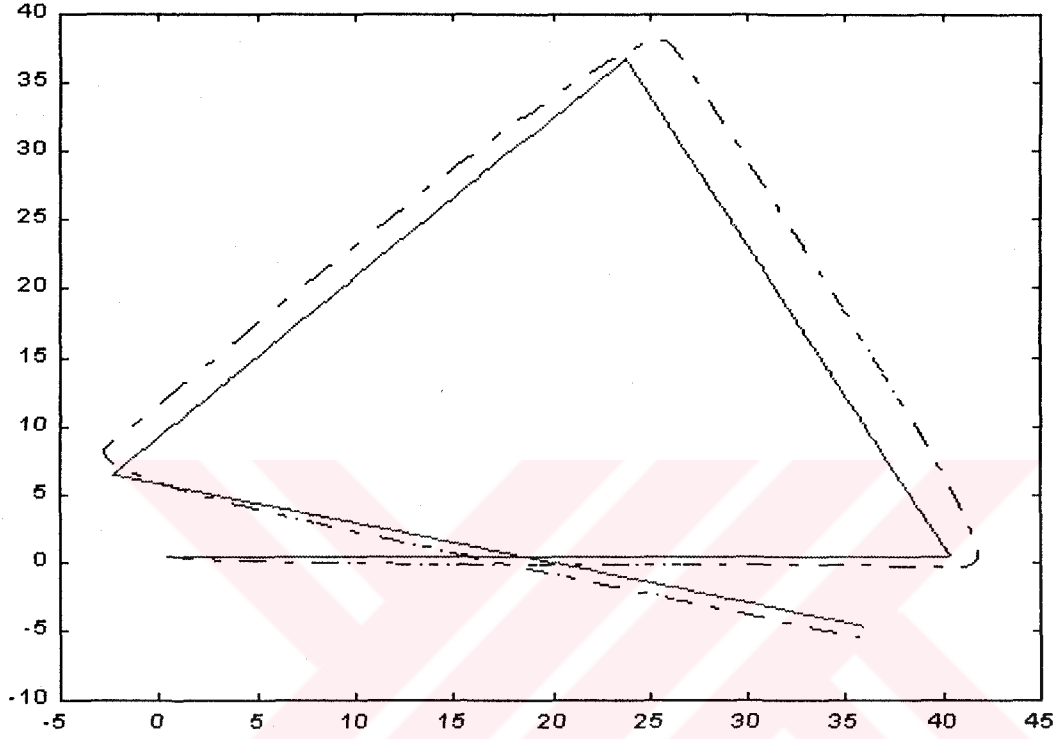
Şekil 4.8. Açık hatası.



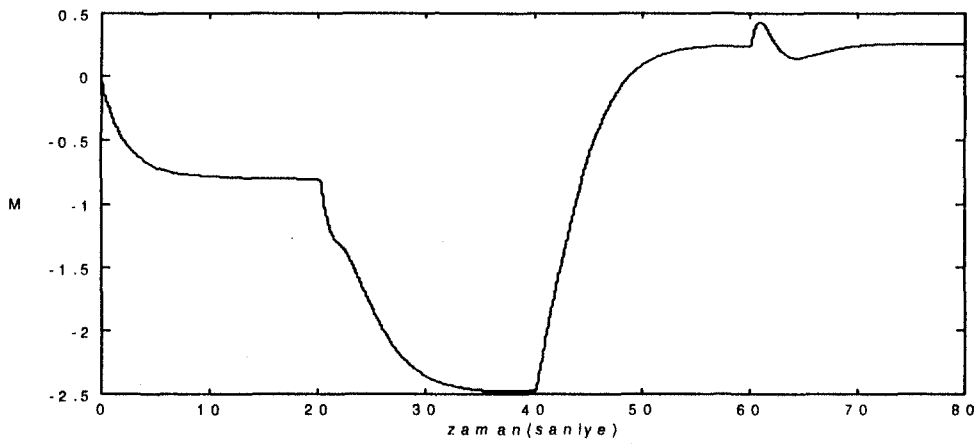
Şekil 4.9. Kontrol çıkışı.

Daha kompleks bir referans verilirse sistemin nasıl davranacağını test etmek için zor olan referanslardan biri olan üçgen referans verelim.

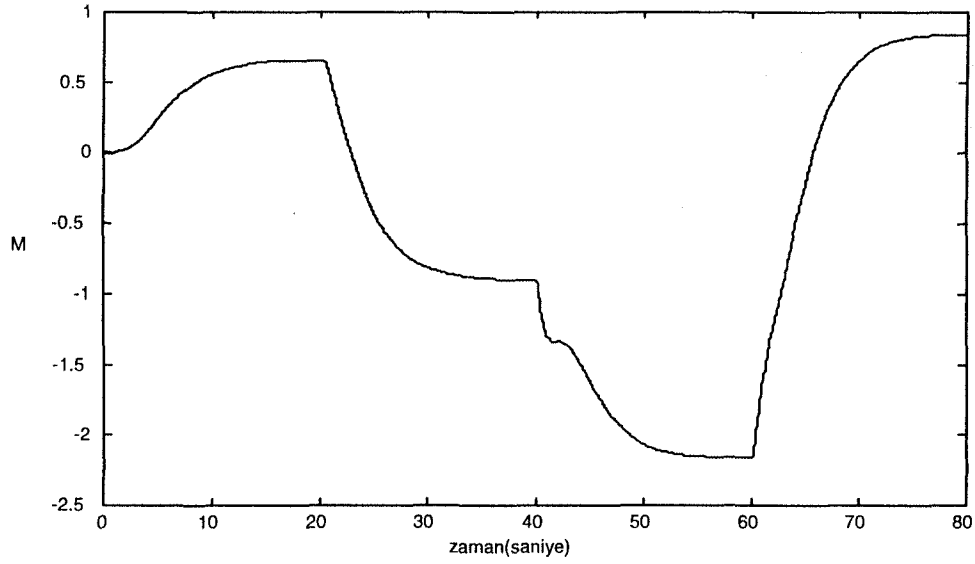
Bu seçilen referansın zorluğu sistemin çalışması esnasında hem hızın hem de açının aynı anda kontrol edilmeye çalışılıyor olmasıdır. Bundan olayı bu tür bir referansın izlenilmesi hız ve açının ayrı ayrı kontrol edildiği referansa göre biraz farklıdır ve daha zordur.



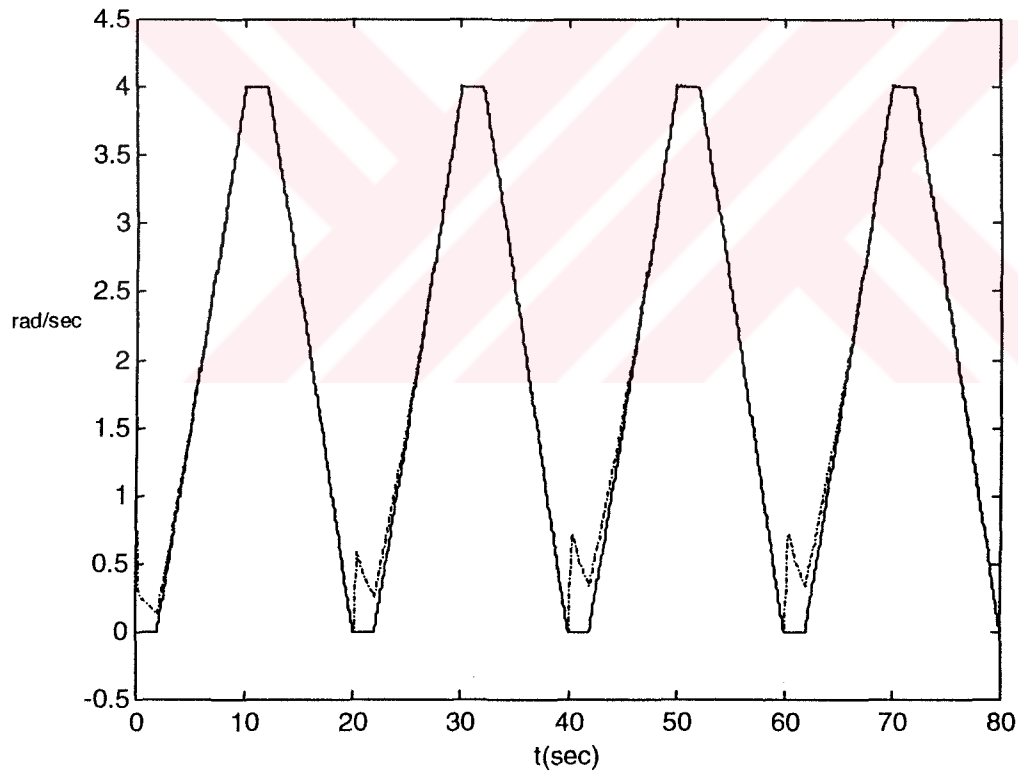
Şekil 4.10. X_Y düzleminde hareket.



Şekil 4.11. X düzlemindeki hata.

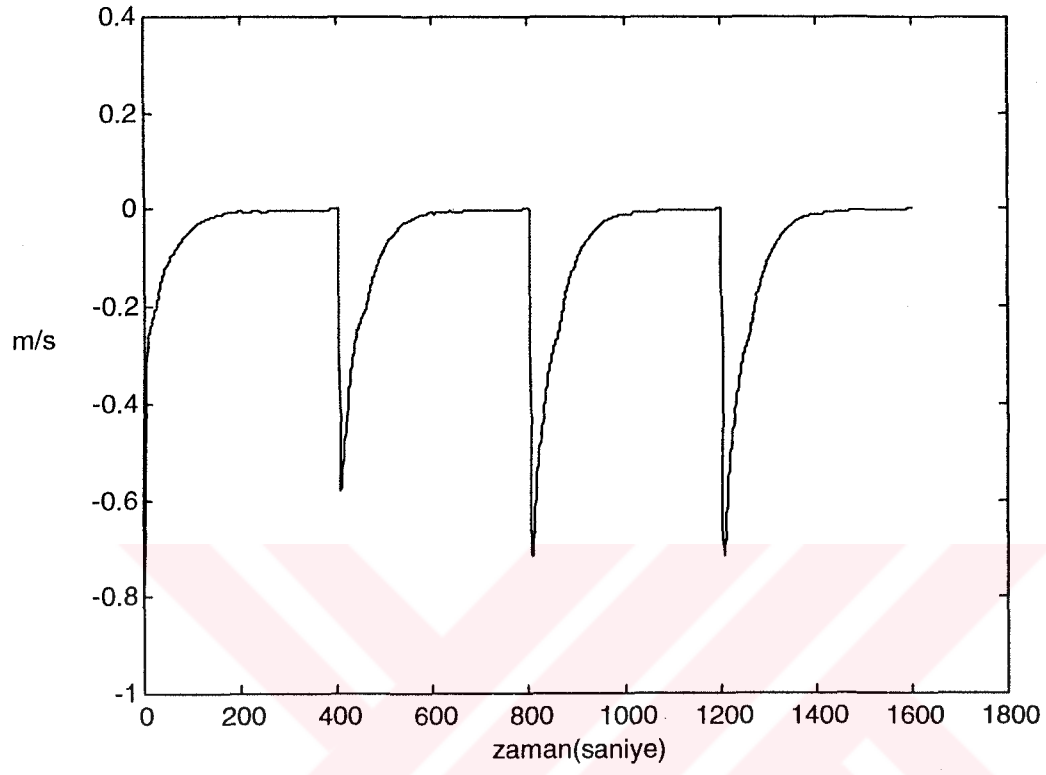


Şekil 4.12. Y düzleminde hata.

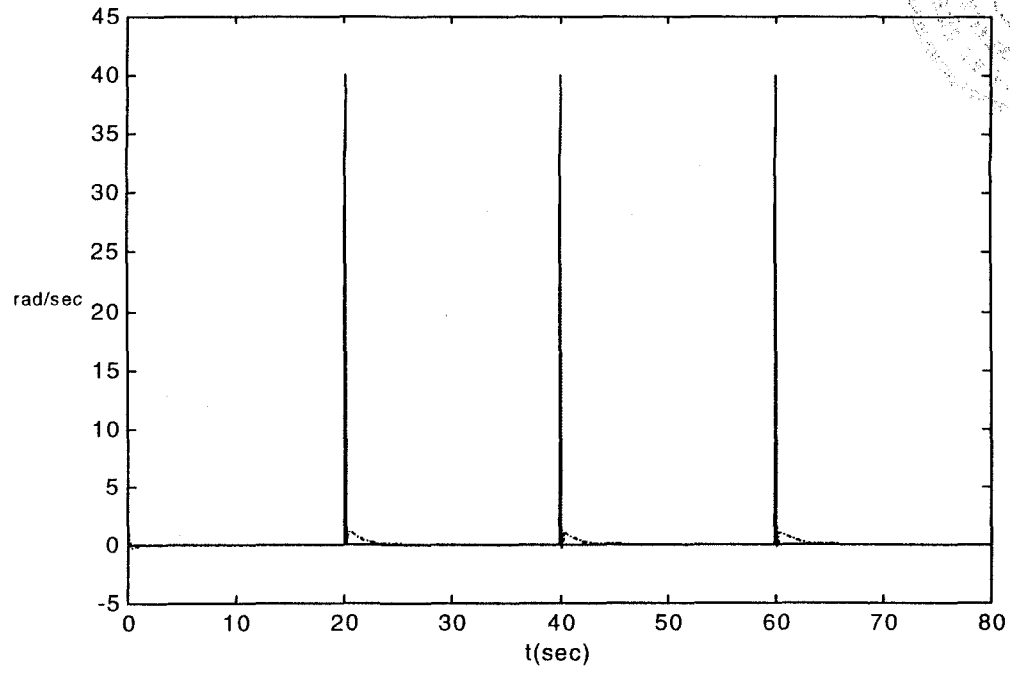


Şekil 4.13. Hız profili.

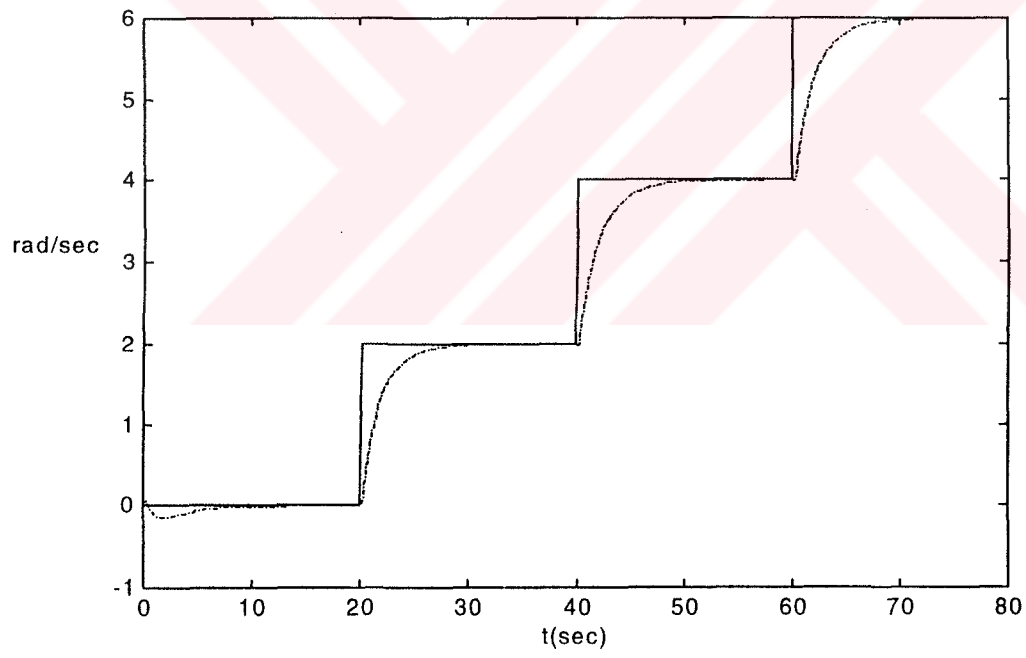
Yukarıdaki şekilden de görüldüğü gibi sistemde hız kontrolü hız sıfıra inerken ve sıfırdan yükselirken referanstan ayrılmakta ve bunun sonucu olarak ta şekil 4.10 da görülen x_y düzleminde referanstan sapmalar oluşmaktadır.



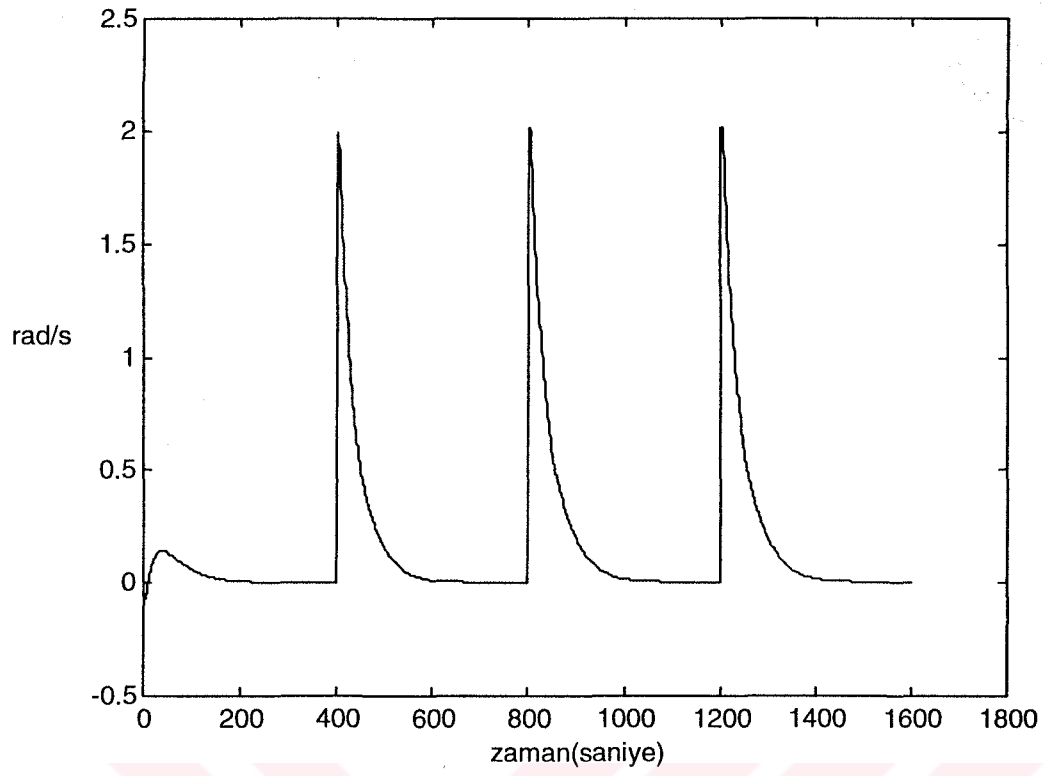
Şekil 4.14. Hız hatası.



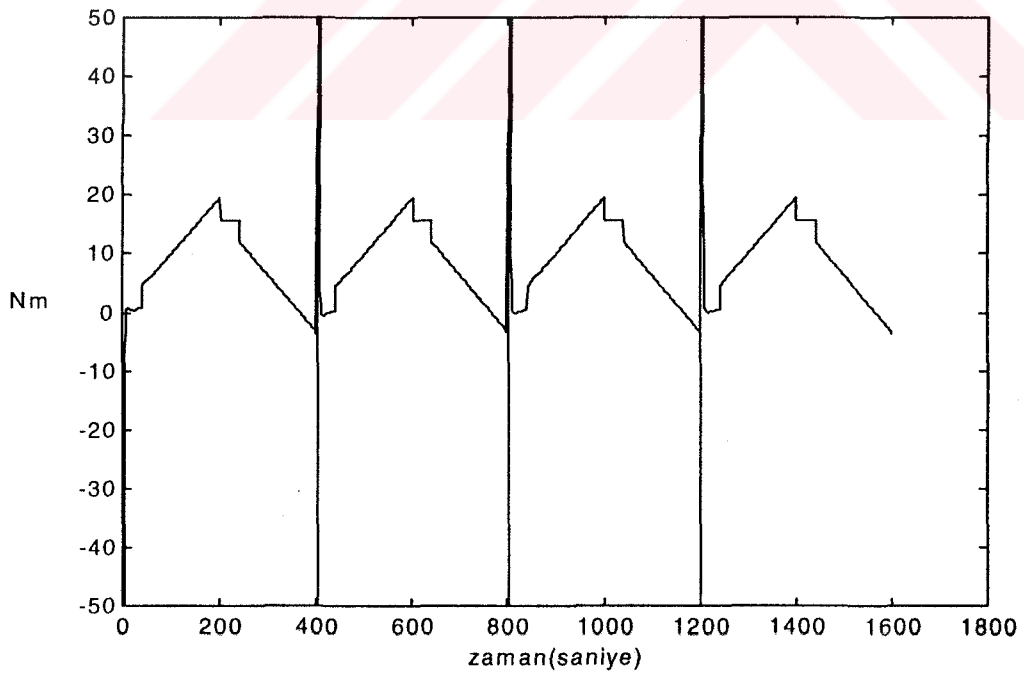
Şekil 4.15. Açısal hız.



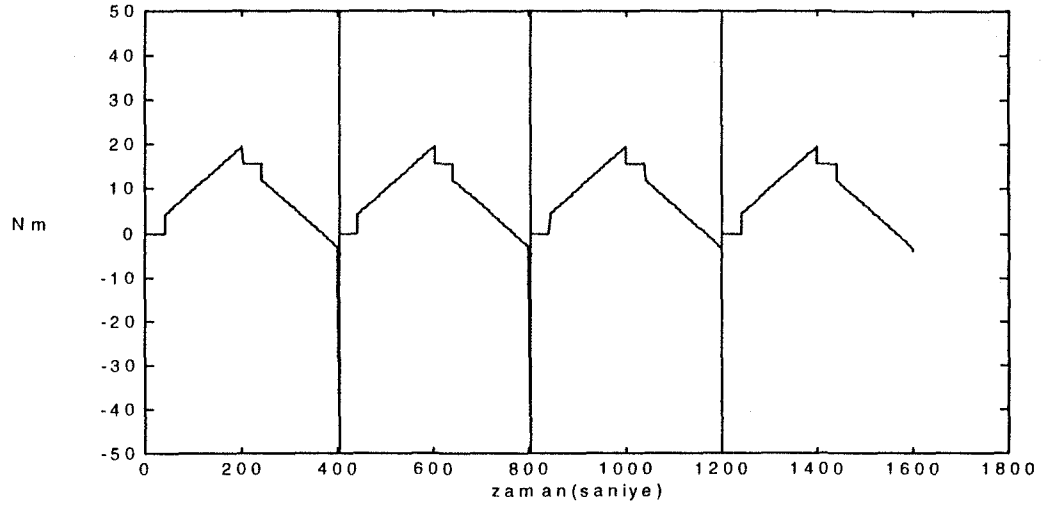
Şekil 4.16. Açı profili.



Şekil 4.17. Açık hatası.



Şekil .4.18. Kontrol çıkışı 1.



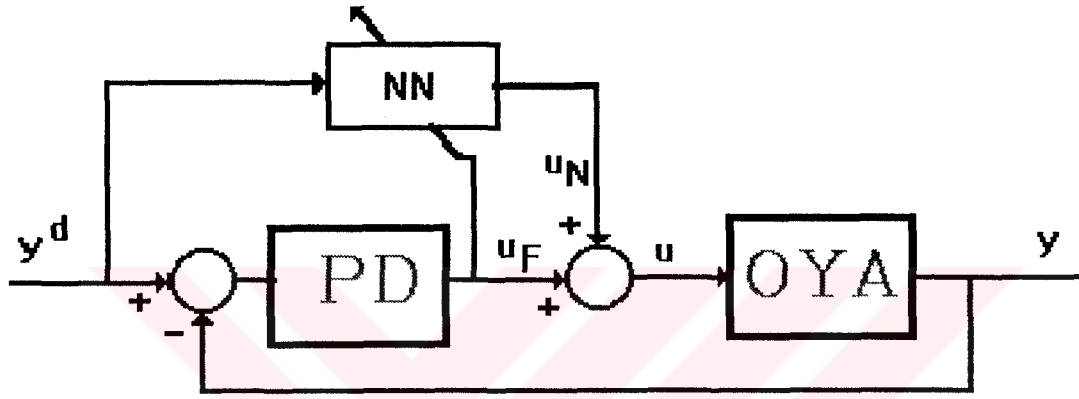
Şekil .4.19. Kontrol çıkışı 2.

5. HATA GERİ BESLEME ÖĞRENMELİ YAPI

Hata geri besleme öğrenmeli yapı (**Feedback error learning architecture**) PD kontrol elemanını öğrenmeye çalışan bir modeldir.

Kawato [Kawato, 1988] et al. Robot kolunu hareket ettirmek için basit bir yapay sinir ağı yapısı önerdi. Bu yapıda yapay sinir ağı tork öğrenmeye çalışıyor

Bu çalışma insan sisteminin nasıl bir kontrol uyguladığına bakarak onu taklit ederek oluşturulmuştur Bu sistem için önerilen yapı şekil 5.1 de gösterilmektedir



Şekil 5.1. Kawato modeli.

Biz bir kontrol mühendisi olarak sistem ile ilgili bir ön bilgiye ihtiyaç duymaktayız.

Bu durumda bizim sistemimiz

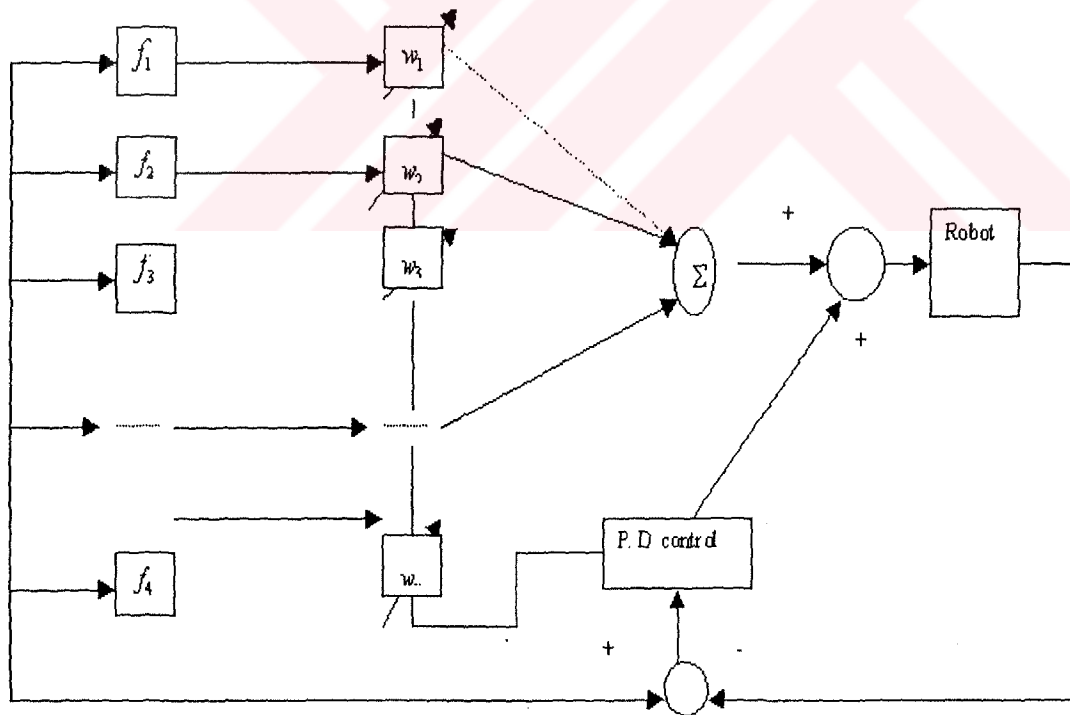
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -\alpha_1 x_1 - \alpha_2 \frac{R}{2} (\text{sign}(g_1(x)) + \text{sign}(g_2(x))) \\ -\alpha_1 x_2 - \alpha_2 \frac{R}{L} (\text{sign}(g_1(x)) - \text{sign}(g_2(x))) \\ x_2 \end{bmatrix} + R\alpha_3 \begin{bmatrix} 1/2 & 1/2 \\ 1/L & -1/L \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_R \\ T_L \end{bmatrix} \quad (6.1)$$

$$x_3 = v$$

$$x_2 = \theta$$

$$x_1 = \theta$$

Yukarıdaki sistem bilgilerini kullanarak Kawatonun önerdiği lineer olmayan terimleri oluşturan fonksiyonları tesbit etmek için (5.1) no'lu denklemi kullanabiliriz. Bu denkleme göre sistemde lineer olmayan kısmı oluşturan fonksiyonlar trigonometrik fonksiyonlardır.



Şekil 5.2. Detaylı FELA yapısı.

5.1. Hata Geri Besleme Öğrenmeli Yapı Simülasyon Sonuçları

Bu simülasyon matlab/simulink ortamında yapılmıştır ve bu kodlar matlab ortamında yazılmıştır.

Önerilen yapay sinir ağı Yapısı

Katman Sayısı :3 (1 Ara)

Nöron sayısı:6-15-2

yapay sinir ağı Yapısı: Backpropagation

Simülasyon Parametreleri

$\alpha_1=0.5$

$\alpha_2=0.2$

$\alpha_3=0.5 \text{ 1/kgm}^2$

$R=0.26 \text{ m}$

$L=1.0 \text{ m}$

$x_{1\text{initial}}=0$

$x_{2\text{initial}}=0$

$x_{3\text{initial}}=1$

$t_0=4.0$; $y_0=0.4 * 10$; $x_0=0.4 * 10$; $m=2.0*y_0/t_0$;

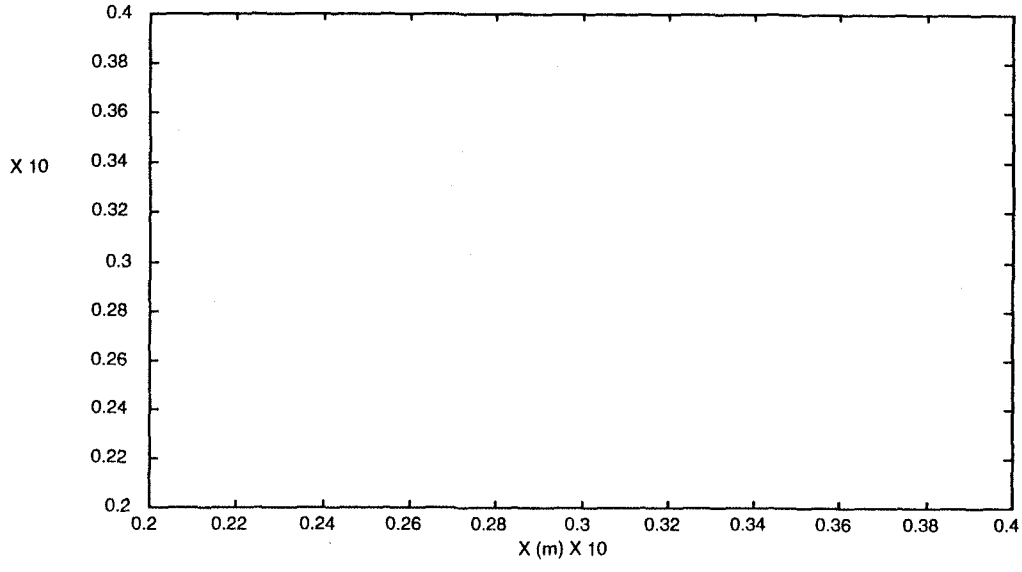
$ddx=0.2$; $ddy=0.2$;

$t_{\text{stop}}=20.0*t_0$; $dT=0.005$; $\text{Tau}=1.0/40.0$;

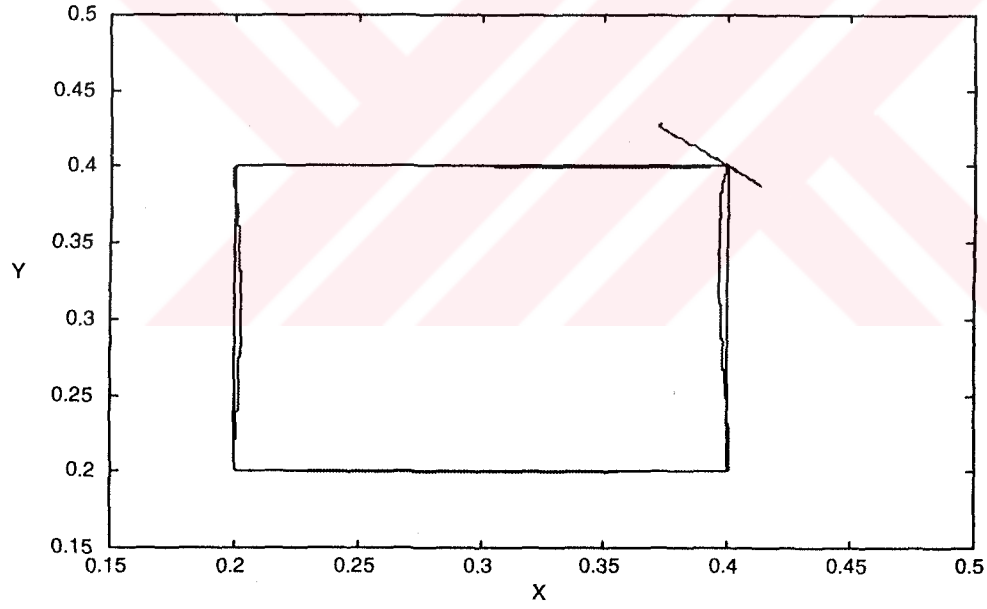
Kontrolör parametreleri

$K_p^1=100$ $K_p^2=100$ $K_p^3=100$

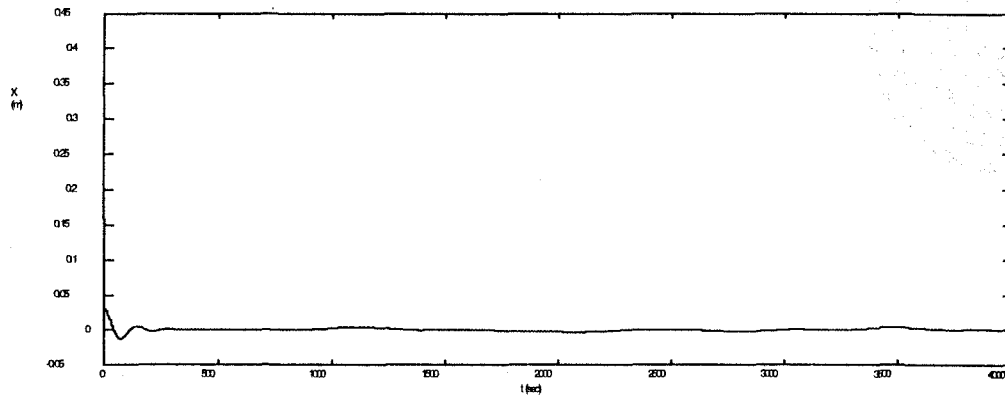
$K_v^1=0.9$ $K_v^2=0.9$ $K_v^3=0.9$



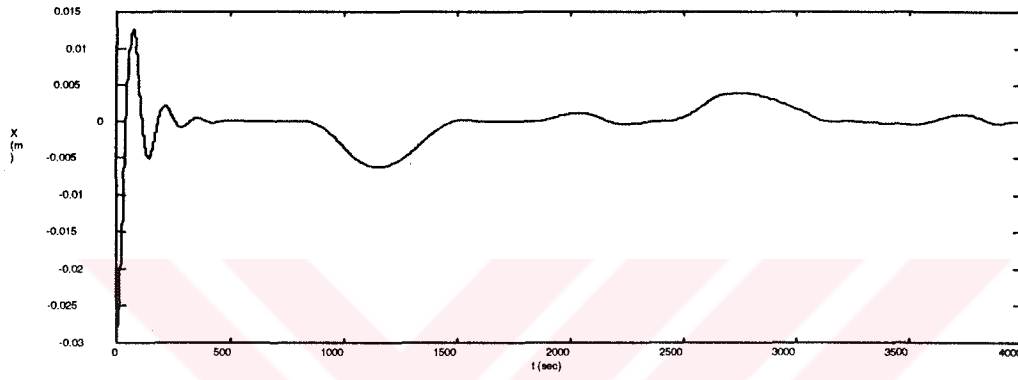
Şekil 5.3. X_Y düzleminde referans.



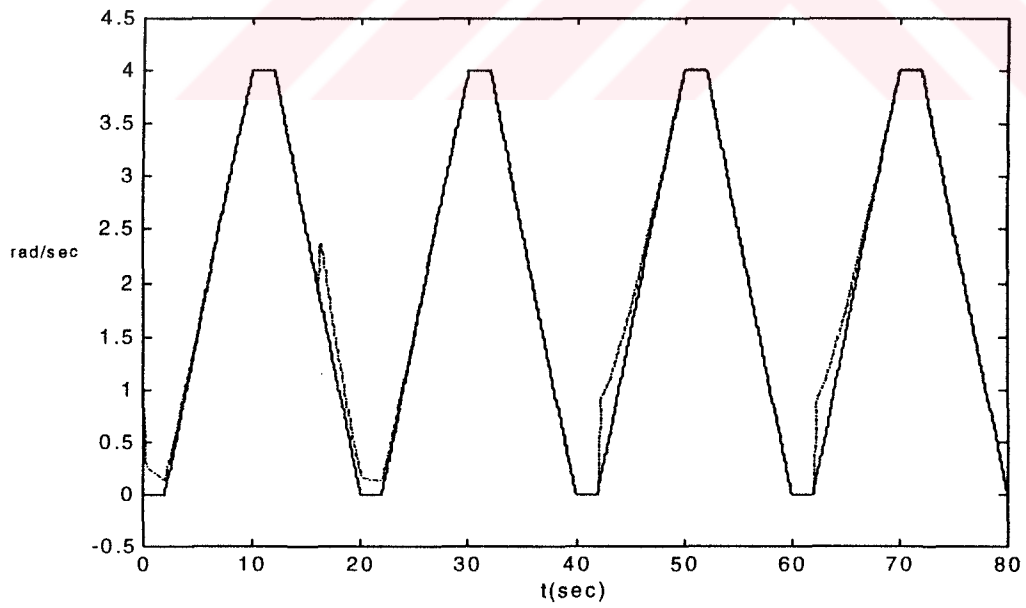
Şekil 5.4. X_Y düzleminde hareket.



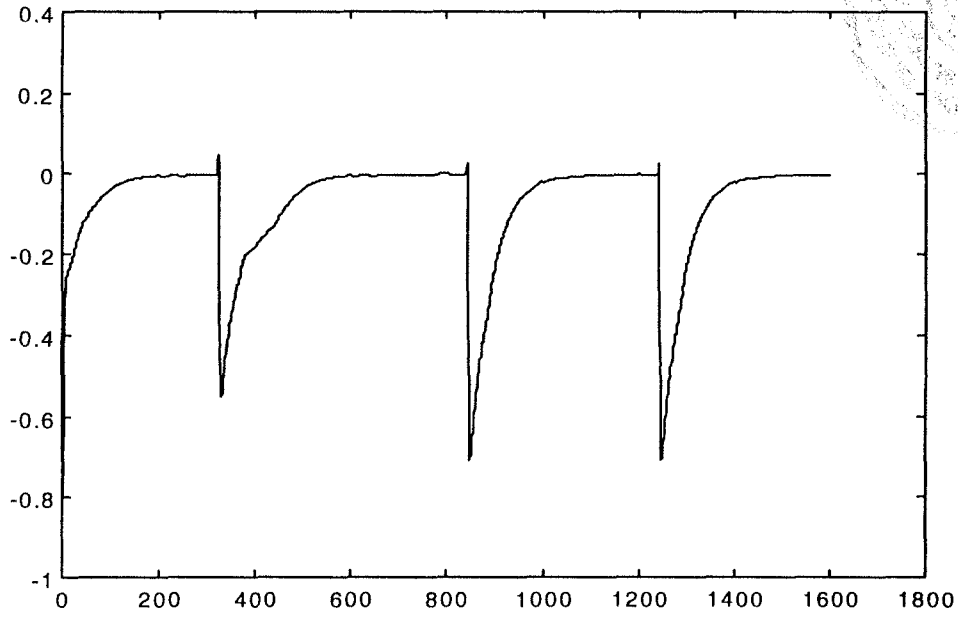
Şekil 5.5. X hatası.



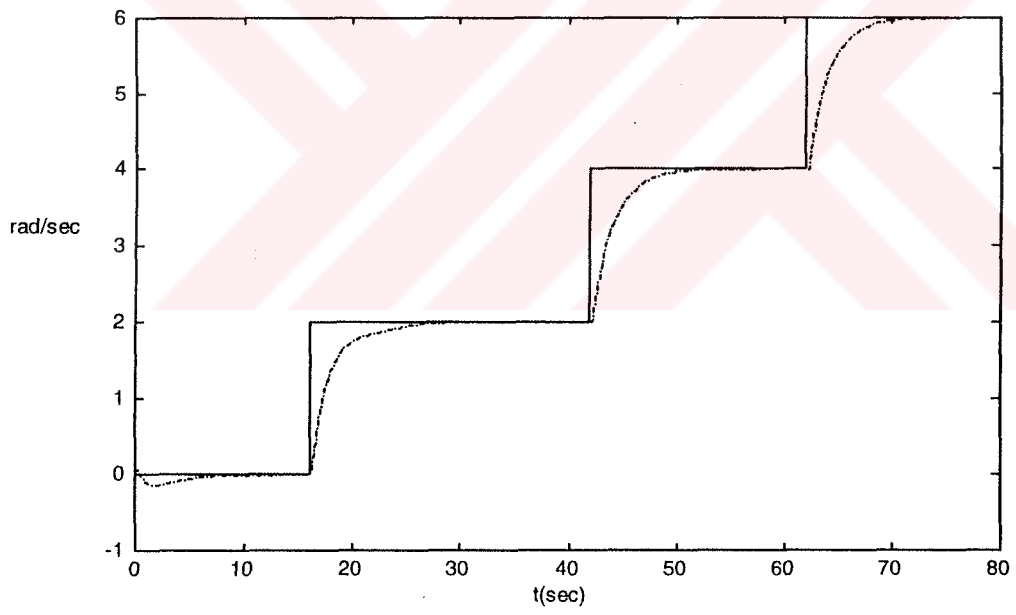
Şekil 5.6 Y hatası.



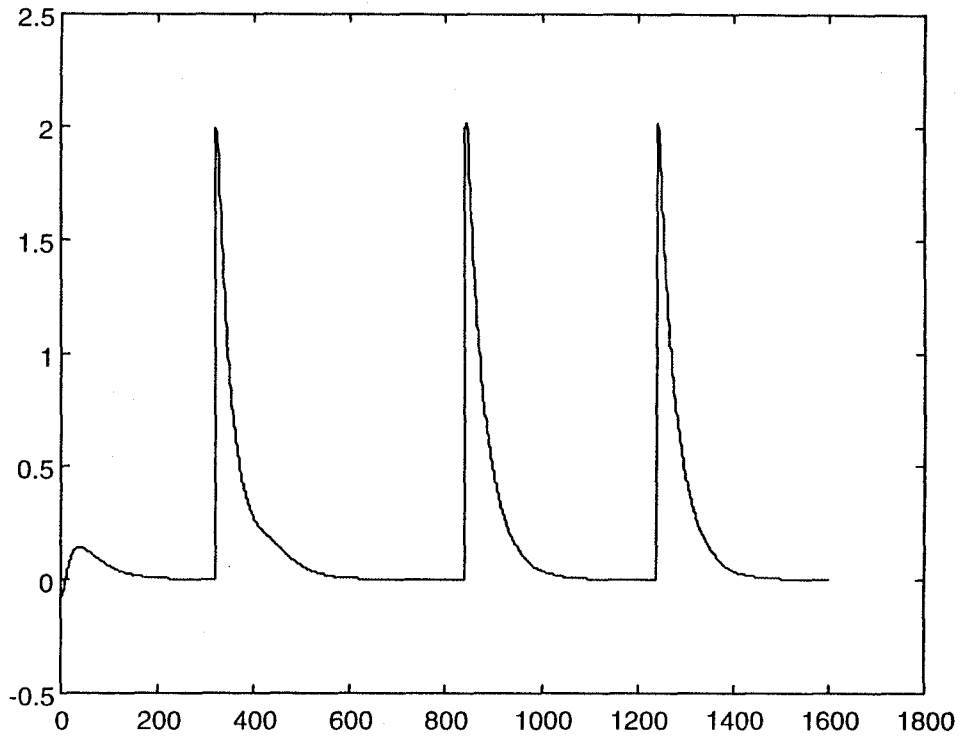
Şekil 5.7. Hız profili.



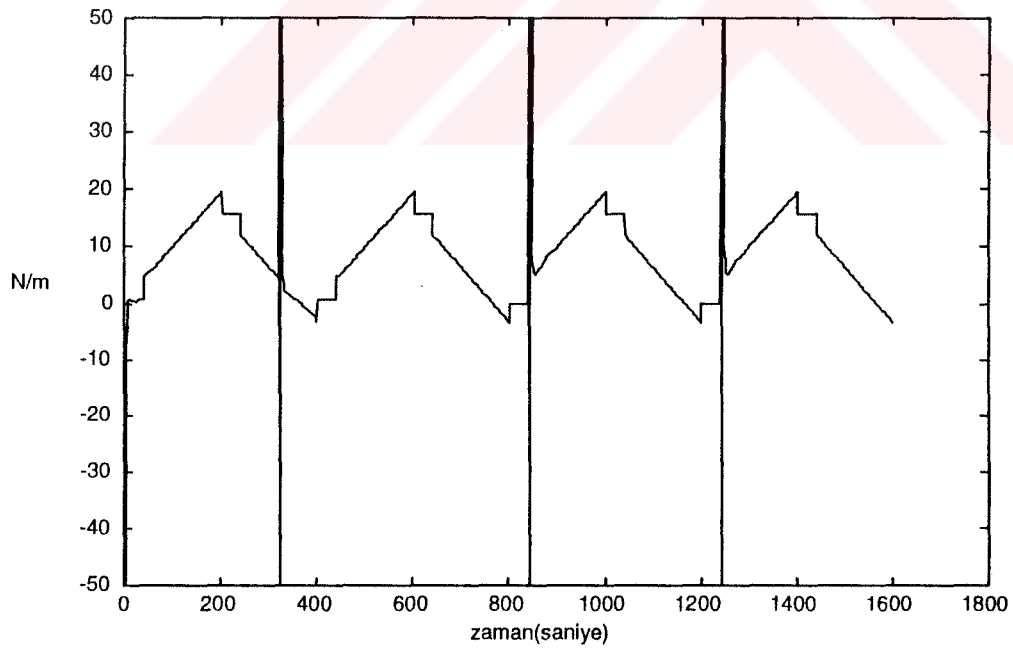
Şekil 5.8. Hız hatası.



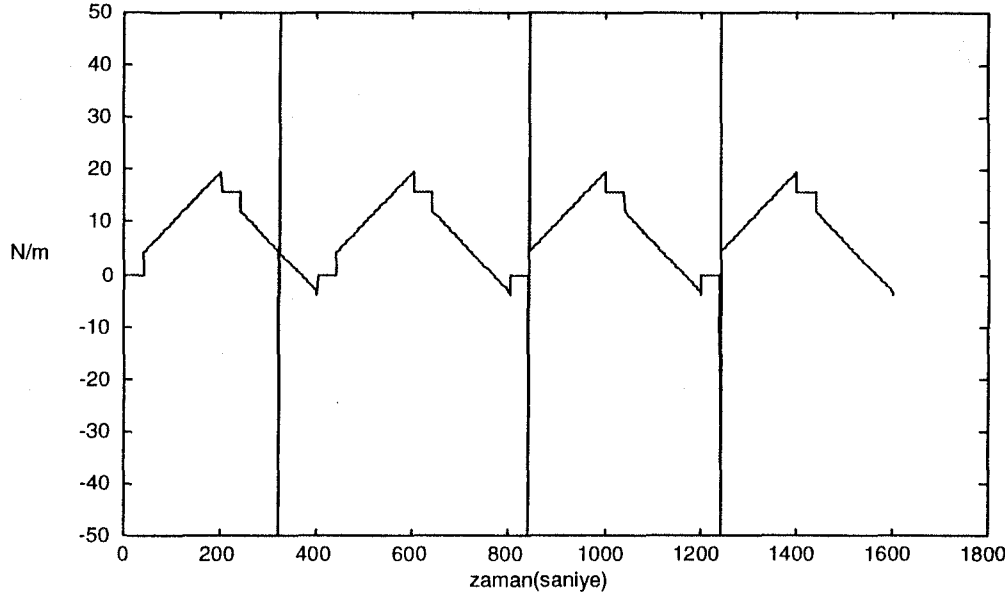
Şekil 5.9. Açı profili.



Şekil 5.10. Açı hatası.



Şekil 5.11. Kontrol 1.



Şekil 5.12. Kontrol 2.

Şekil 5.7 ve 5.8 bize bu yöntemi kullanırken oluşan hız profilini inceleme imkanı vermektedir. Bu şekillere bakarsak hız profili diğer metoda göre referansı daha kötü izlemesine rağmen bu sistem online öğrenme ve denetim yaptığı yani denetime ve kontrolü aynı anda yaptığı ve sistem dinamiği ile ilgili ön bir bilgiye ihtiyaç duymadığı için tercih edilen bir yöntemdi. Şekil 5.11 ve 5.12 den görülebileceği gibi bu yöntemin araca uyguladığı kontrol kuvveti de ters dinamik modelleme yöntemi kadar iyi değildir. Buna rağmen sistemin ters dinamiğini öğretecek yeterli verinin elde olmaması durumunda kullanılabilecek ve iyi sonuçlar veren bir metoddur.

6. SONUÇ

Bu çalışmada otomatik yük arabasının yapay sinir ağları ile kontrolü incelendi ve simülasyon sonuçları verildi.

Bu tezde yapılmak istenen çalışma otomatik yük arabaları gibi dinamiği kolayca dış etkenlerden dolayı değişebilen sistemler için bir denetleyici tasarlamaktır. Otomatik yük arabasının dinamiği çalışma şartlarına bağlı olarak değişir çünkü arabanın taşıyacağı yük birkaç yüz kilogramdan 35 ton 'a karar değişebilir. Bunun sonucunda dinamiğe doğrudan etki eden bu parametre sistemin dinamiğini de oldukça değiştirir. Bu tür bir problemi çözmek için literatürde kullanılan adaptif kontrol metodlarından biri de yapay sinir ağları ile kontroldür.

Bu çalışmada kullanılan ilk metot Ters Dinamik Modelleme Yöntemi (Direct Inverse Modeling Architecture(DIMA)) dir. Bu yöntem kontrol edilmek istenen sistemle ilgili bir ön bilgi şartı gerektirmediği için seçilmiştir. Tezde kullanılan metotlardan ikincisi ise Kawato tarafından ilk defa önerilen ve daha sonra pek çok araştırmacı tarafından çeşitli problemlere uyarlanan Hata Geri Beslemeli Öğrenme Yöntemi(Feedback Error Learning Architecture(FELA)) dir. Bu yöntemde ise sistem ile ilgili olarak bir ön bilgiye ihtiyaç vardır. Tezde seçilen bu iki metodun birbirinden ayrıldığı nokta birisinin sistem dinamiği ile ilgili herhangi bir ön bilgi gerektirmemesi diğerinin ise ön bilgiye ihtiyaç duymasıdır.

Yapılan ilk çalışmada kullanılan Ters Dinamik Modelleme Yöntemi (DIMA) olup. Bu yöntemi kullanabilmek için sistem *off-line* olarak öğretildi. Bu metodu kullanabilmek için kontrol edilecek olan sistemin durumlarına (state) ulaşabilmek yani onları ölçebilmek ve de sistemin giriş çıkış verilerini sağlıklı bir şekilde toplayabilmek gerekmektedir. Bunun sonucunda yapay sinir ağına sistemin ters dinamiği öğretilir ve sistem sanki lineer bir sistemmiş gibi kullanılır. Bu metot göstermiştir ki eğer sistem yeterince iyi modellenirse iyi sonuçlar alınabilir.

Kullanılan ikinci metot olan Hata Geri Beslemeli Öğrenme Yöntemi(FELA) daha önce de sözü edildiği gibi sistem ile ilgili ön bir bilgiye ihtiyaç duyar. Bu metot *on-line* öğrenen bir yapı olduğundan dolayı pek çok çalışmada tercih edilmiştir vede diğer metotta oluşan elde edilen dataların çalışılacak bölgede olmaması problemini çözer.

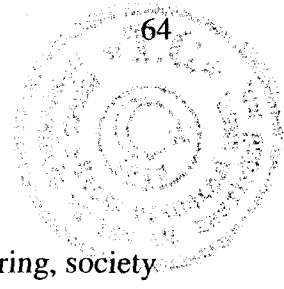
Kullanılan ikinci metotta elde edilen sonuçlar her ne kadar ilki kadar iyi olmasa da yukarıda bahsi geçen ve pek çok kere karşımıza çıkan problemi etkin bir biçimde

özmesi ve kolay uygulanabilmesi dolayısıyla ve en önemlisi de sonuçların kabul edilebilir olmasından dolayı tercih edilen bir metottur.

Bu alışmada bahsi geçen iki metot birbirinden kontrol stratejileri açısından farklıdır. Ve her metot sıka karşılaşılan bir problemi özmek için tasarlanmıştır. İki farklı metot seçiminin temelinde de bu problemlere değinmek ve bunlara özüm üretmektir.

Bu alışmanın diğerk bir amacı da yeni bir yöntem olan yapay sinir ağıları ile kontrol konusuna bir katkı ve bu yöntemin uygulanabilirliğini göstermektir.





1- Miller, R.K., Automated Guided Vehicles and Automated Manufacturing, society of Manufacturing Publications Development Marketing division, 1987.

2- Albus J., "A new approach to manipulator control: the cerebellar model articulation controller(CMAC)", J. Dyn. Syst. Meas. And Control, pp220-227, 1975.

3- Psaltis D., A. Sideris, A. Yamamura, "Neural Controllers ", Proc. IEEE Int. Conference on Neural Networks, 1987.

4- Guez A., Z. Ahmad, "Solution to inverse problem in robotics by neural networks", Proc. Int. Conference on Neural Networks, 1988.

5- Elsey R., "A learning architecture for control based on back-propagation neural-networks", IEEE conf. On Neural Networks, Vol.2, pp.584-587, 1988.

6- Grossberg S. and M.Kuperstein, " Neural Dynamic of Adaptive sensory-motor Control", Elmsford, NY: Pergamon Press, 1989.

7- Fukuda T. et al., "Neural Servo controller; Adaptation and Learning", Proc. IEEE Intl. Workshop Advanced Motion contr. Pp.107-115, 1990.

8- Bekey G. A., "Robotics and Neural Networks ", in Neural Networks for signal Processing, B. Kasko, Printice all, 1992.

9- 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.

10- Rosenblatt, F., Perceptron simulation experiments, In: Proceeding of the IRE, 1960.

11- Hopfield J.J., " Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons," Proc. Natl. Acad. Sci. USA, Vol. 81, 3088-3092, 1984.

12- Rumulhart, D.E. and J.L. McClelland. Parallel Distributed Processing. Vol. 1. MIT press. Cambridge, 1986.

13- Widrow, B. and F.W. Smith. Pattern-recognizing control systems. In: Proceeding of Computer and Information Sciences, 1964.

14- Psaltis, D., A. Sideris and A. A. Yamamura. A multilayered neural network controller. IEEE Control Systems Magazine 8, 17--21, 1988.

15- Antsaklis R.J., "Defining Intelligent Control," Report of the Task Force on Intelligent Control, , Chair, IEEE Control Systems Magazine, pp. 4-5 & 58-66, June 1994.

16- Widrow, B. and F.W. Smith, Patternrecognizing control systems. In: Proceeding of Computer and Information Sciences, 1964.

17- Hunt, K.J., D. Sbarbaro, R. Zbikowski and P.J. Gawthrop, Neural networks for control systemsa survey. Automatica 28(6), 1083--1112, 1992.

18- Hunt, K. J. and. A. Johansen, Adaptive local controller networks., <http://members.xoom.com/~hunt>, 1996.

19- Steck, J.E., K Rokhsaz and S.P. Shue, Linear and neural network feedback for flight control decoupling. IEEE Control Systems Magazine 16(4), 22--30., 1996.

20- Gomi, Hiroaki and Mitsuo Kawato, Neural network for a closedloop system using feedbackerrorcontrol. Neural Networks pp. 933--946., 1993.

21- Widrow, B. and S. D. Stearns, Adaptive Signal Processing. Prentice-Hall. Englewood Cliffs., 1985.

22- Sanner, R. M. and D. L. Akin, Neuromorphic pitch attitude regulation of an underwater telerobot. IEEE Control Systems Magazine 10, 62--67., 1990.

23- Ogata, K., Modern Control Engineering , Printice Hall, 1970.

24- Kuo, B.C., Automatic Control Engineering, Printice Hall, 1970.





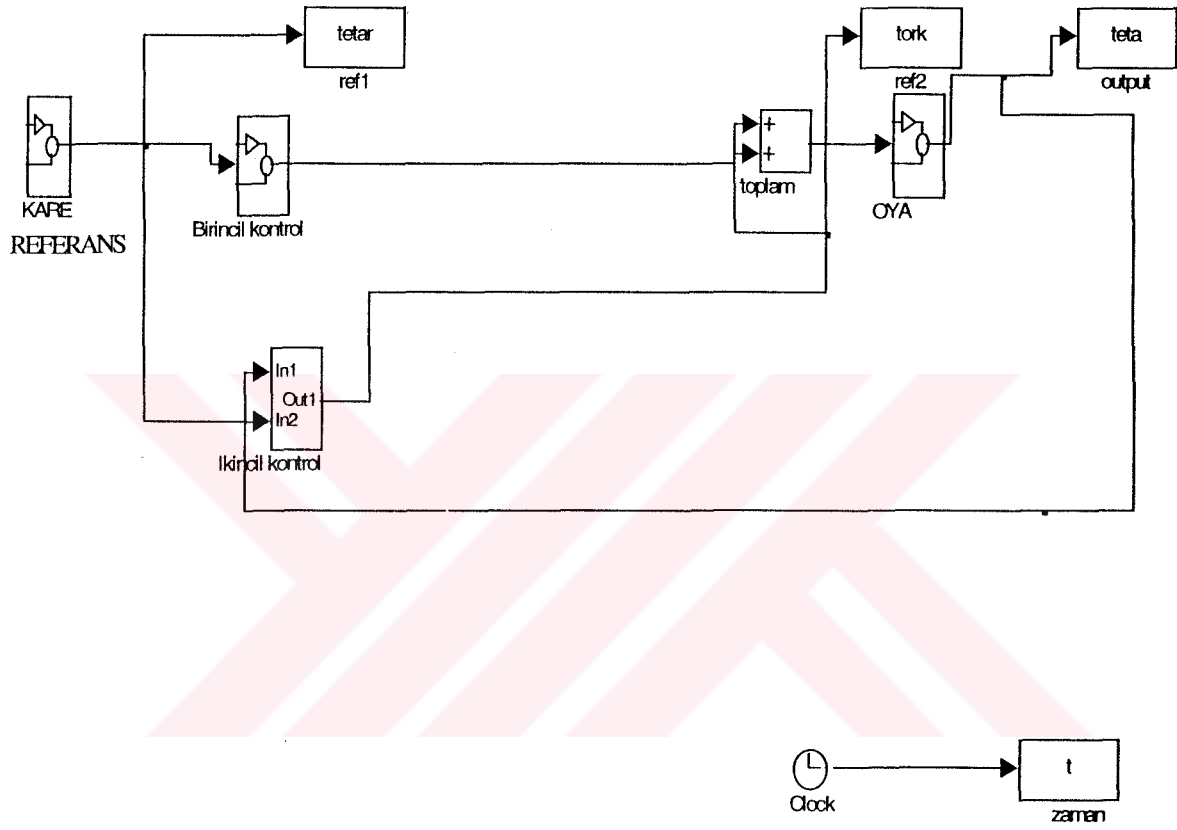
ÖZGEÇMİŞ

01-10-1973 Kastamonu doğumlu olup 1992 Kadıköy Anadolu Lisesi mezunudur. Yüksek öğrenimini İ.T.Ü. Uçak ve Uzay Bilimleri Fakültesi Uçak müh. bölümünü 1996 yılında C++ ile uçak simülasyonu konusunda tamamlamıştır. Halen TÜBİTAK Marmara Araştırma Merkezinde Araştırmacı olarak çalışmaktadır.

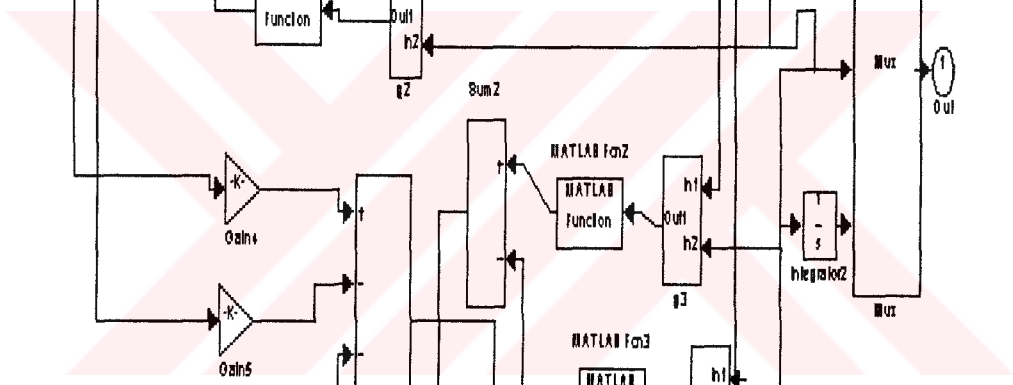


EK A

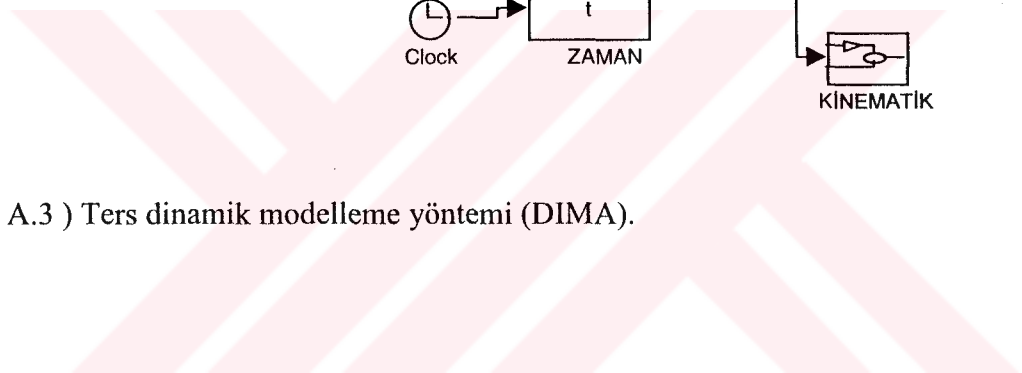
MATLAB –SIMULINK KAYNAK KODLARI



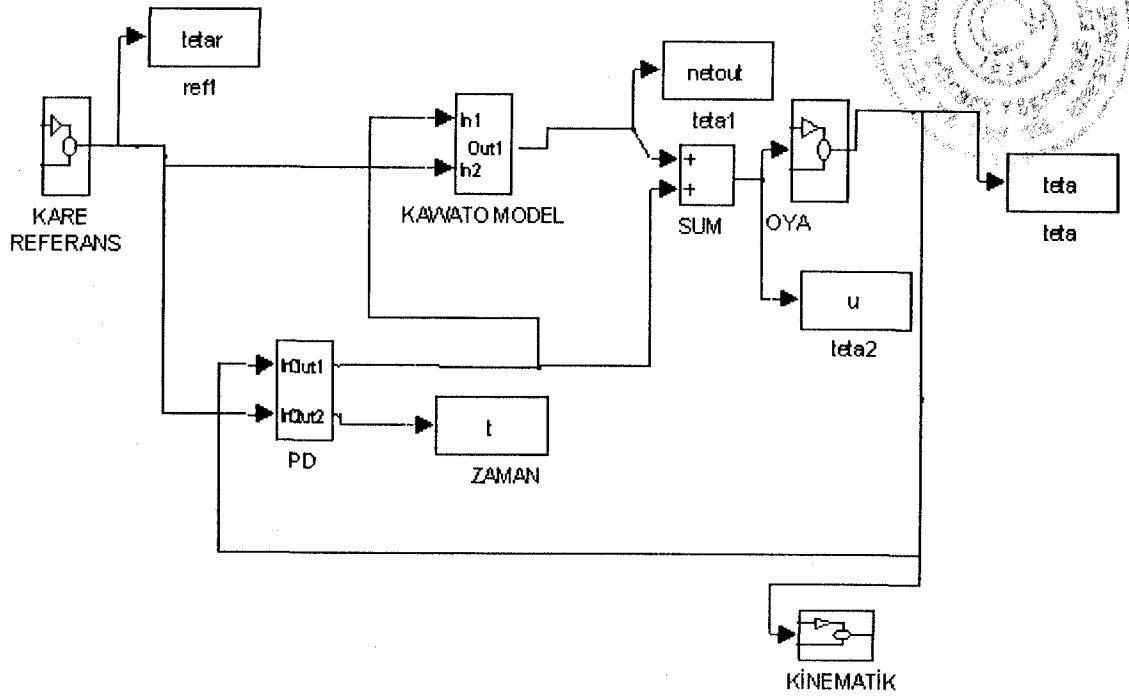
Şekil A.1) Birincil ve ikincil kontrol.



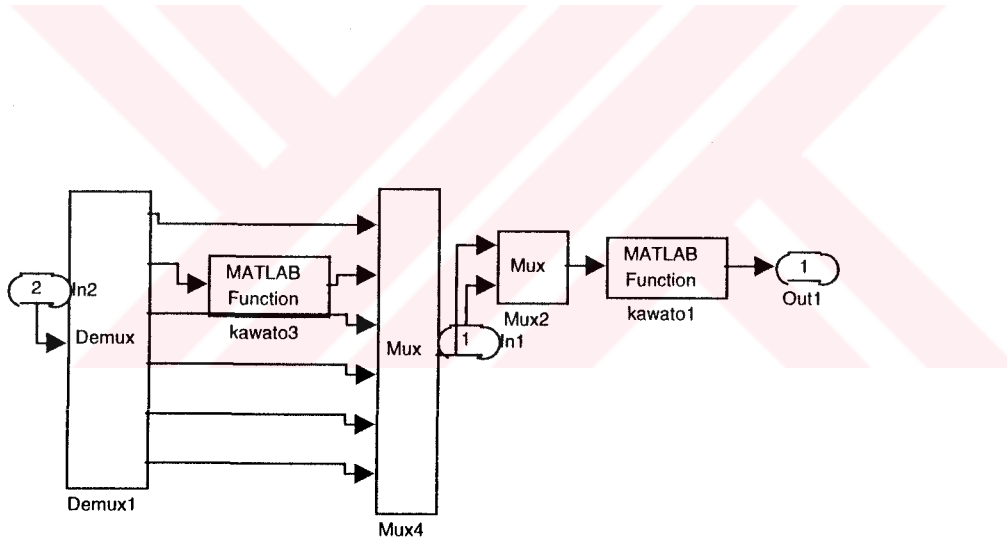
Şekil A.2) OYA.



Şekil A.3) Ters dinamik modelleme yöntemi (DIMA).



Şekil A.4)Kawato yöntemi (FELA).



Şekil A.5) Kawato model.