

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**İTÜ TRIGA MARK-II REAKTÖRÜNÜN
YAPAY SİNİR AĞIYLA KONTROLÜ**

YÜKSEK LİSANS TEZİ

Melih İNAL

Ana Bilim Dalı : Elektronik ve Bilgisayar Eğitimi

HAZİRAN 1996

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

İTÜ TRIGA MARK-II REAKTÖRÜNÜN
YAPAY SİNİR AĞIYLA KONTROLÜ

YÜKSEK LİSANS TEZİ

Melih İNAL

Tezin Enstitüye Verildiği Tarih : 07 Haziran 1996

Tezin Savunulduğu Tarih : 17 Temmuz 1996

Tez Danışmanı

Yrd.Doç.Dr.

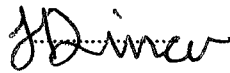
Kadir ERKAN



Üye

Prof.Dr.

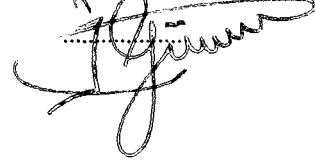
Hasan DİNÇER



Üye

Prof.Dr.

İrfan GÜNEY



HAZİRAN 1996

İTÜ TRIGA MARK-II REAKTÖRÜNÜN YAPAY SİNİR AĞI KONTROLÜ

M. Melih İNAL

Anahtar Kelimeler: Yapay Sinir Ağları, Reaktör Kontrolü

Özet: Bu çalışmada, İTÜ TRIGA Mark-II reaktörünün kontrolü için lineer olmayan yapay sinir ağı kontrolörü önerilmiştir. Geriyansıtma ağ modelinin kullanıldığı kontrolör tasarımı, sistem çıkışı veren parametrelerin bulunması için ayrıca lineer yapay sinir ağı kontrolörü tasarlanmıştır. Lineer kontrolörün, önceden hazırlanmış reaktör simülasyon programında "CANER programı" (Erkan 1994) gerekli düzenlemeler yapılarak çalıştırılması sonucu; 250 Kw'a kadar istenen güç düzeyi, başlangıç kararlı hal gücü 5 Kw, örnekleme aralığı 0.05 sn ve inceleme aralığı 0-200 sn. için 1000 eğitim verisi kullanılarak, lineer olmayan yapay sinir ağı kontrolörü; ST3.PAS programında eğitilmiştir.

Kontrol sistemi, parametreleri bilinmeyen ikinci dereceden lineer model kullanılarak tasarlanmıştır. Gerçekte lineer olmayan reaktörün, lineer modellerle yeterince iyi temsil edilebilmesi için kontrol parametreleri gerçek zamanda yenilenmektedir. Yenileme işlemi, ALC ağ modelinde en küçük kareler yöntemiyle parametre kestirimi yapılmaktadır.

Zaman gecikmelerine karşı lineer olmayan reaktörü yeterince iyi temsil edecek yörüngenin seçimi güçtür. Zaman gecikmesi nedeniyle yörünge ile fiziksel sistem çıkışı arasındaki hata artabilir. Kontrol sistemi, bir adım sonra hatayı giderecek şekilde kontrol sinyali üretirse aşım neden olur. Çalışmamızda, hatanın belirli bir değerin üzerine çıkması yörünge kaydırılarak engellenmiştir. Böylece, kontrolöre daha fazla zaman tanıyarak zaman gecikmelerinin kontrol sistemi üzerindeki istenmeyen etkisi giderilmiştir.

Çalışmamızda yörünge kübik fonksiyon olarak seçilmiştir. Lineer kontrolör, reaktör gücünün verilen yörüngeyi izlemesini sağlamaktadır. Lineer olmayan kontrolörün eğitiminde yörünge kullanılmadığından kontrolörün; istenen güç düzeyine yörünge verilmeden çıkması istenmiştir.

Önerilen kontrolörlerin, değişik çalışma koşulları altındaki davranışı YAVCAN kodu kullanılarak simüle edilmiştir. Simülasyon çalışmalarında; başlangıç kararlı hal değeri, çıkılan güç düzeyi, örnekleme zamanının lineer ve lineer olmayan kontrolör üzerindeki etkileri incelenmiştir. Yapılan simülasyonlarda; önerilen lineer kontrolörün, reaktör gücünü başlangıç kararlı hal değerinden istenilen güç düzeyine verilen yörüngeyi izleyerek çıkarabildiği ve kararlı hal hatasının olmadığı görülmüştür. Lineer olmayan kontrolörün, reaktör gücünü istenilen güç düzeyine çıkarabildiği fakat istenilen güç düzeyinde 1 Kw'lık salınımlar yaptığı görülmüştür.

YAVCAN kodu kullanılarak yapılan simülasyonlar, yapay sinir ağı kontrolörlerinin, İTÜ TRIGA Mark-II reaktörünün kontrolünde kullanılabileceğini göstermektedir.

NEURAL NETWORK CONTROL OF ITU TRIGA MARK-II REACTOR

M.Melih İNAL

Keywords: Artificial Neural Networks, Reactor Control.

Abstract: In this study, a control system for ITU TRIGA Mark-II Reactor is designed and tested with a reactor simulation program. Backpropagation network model is used in design of controller. Also a linear neural net controller is used for finding parameters of system output.

Second degree of linear model which parameters are unknown is used for control system designed. The control parameters of nonlinear reactor are updated on-line. In this process; ALC network with least squares method is used for parameter estimation.

A trajectory which is a cubic function is chosen for give a sensitive to system delay time. In this study, we allow the linear controller more time to deal with time delays by shifting the trajectory. Therefore, the linear controller controls the reactor power to follow the given trajectory.

Simulation results on the research reactor show us that both of the nonlinear and linear neural net controllers can be controlled the ITU TRIGA Mark-II Reactor.

ÖNSÖZ ve TEŞEKKÜR

Her alana rahatlıkla uygulanabilen yapay sinir ağlarının; günümüzde yapılan arařtırmalar dođrultusunda modern kontrol sistemleri içinde de vazgeçilmez bir alan oluşturacağını ve gelecekteki teknolojilerin içinde yapay sinir ağlarının yer alacağını düşünmekteyim. Yapılan çalışmanın kontrol alanındaki uygulamalara katkısı olmasını dilerim.

Bana bu konuda çalışma olanađı veren ve değerli vaktini bana ayırarak yardımlarını esirgemeyen danışmanım Sayın Hocam Yrd.Doç.Dr. Kadir ERKAN 'a (KOÜ.T.E.F.) teşekkürlerimi sunarım.



İÇİNDEKİLER

ÖZET.....	i
ABSTRACT.....	ii
ÖNSÖZ Ve TEŞEKKÜR.....	iii
İÇİNDEKİLER.....	iv
SİMGELER DİZİNİ Ve KISALTMALAR.....	viii
ŞEKİLLER LİSTESİ.....	ix
TABLolar LİSTESİ.....	xiii
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. YAPAY SİNİR AĞLARI.....	6
2.1. Y.S.A.'nın Uygulama Alanları.....	8
2.2. Y.S.A. 'nın Yapısı ve Etkinlik Şekilleri.....	9
2.2.1. Sinaps etkinliği.....	9
2.2.2. Y.S.A. etkinliği.....	10
2.3. Yapay Sinir Ağının Organize Edilmesi.....	12
2.3.1. Nörodinamik.....	12
2.3.2. Yapı "Mimari".....	13
2.4. Sinir Devreleri ve Hesaplamaları.....	14
2.5. Öğrenme Modları ve Kuralları.....	16
2.5.1. Öğrenme modları.....	16
2.5.1.1. Eğitici (Supervised) öğrenme modu.....	17
2.5.1.2. Eğitici (Unsupervised) öğrenme modu.....	17
2.5.2. Öğrenme Kuralları.....	18
2.5.2.1. En küçük kareler yöntemi KY (Least-Mean-Square "LMS") veya DELTA Öğrenme Kuralı.....	18
2.5.2.1.1. Ağırlıkların Hesaplanması.....	18
2.5.2.1.2. En dik eğim (Steepest Descent) metoduyla ağırlıkların (W^* 'nin) bulunması.....	19

2.5.2.2. Hebb öğrenme kuralı.....	23
2.5.2.3. Gradient eğimi öğrenme kuralı	23
2.5.2.4. Kohonen öğrenme kuralı.....	23
2.5.2.5. Geri yansıtma öğrenme kuralı.....	23
2.5.2.6. Grossberg öğrenme kuralı	24
2.5.2.7. Kuvvet güçlendirme teorisi-KGT(Drive-Reinforcement Theory-DRT)	24
2.6. Yapay Sinir Ağlarının Paralel Mimarideki Yeri	25
2.6.1. Sinirbilgisayarları (Neurocomputers)	25
2.6.2. Genel amaçlı sinirbilgisayarları	25
2.7. Yapay Sinir Ağı Modelleri.....	27
2.7.1. Algılama (Perceptron) Modeli	27
2.7.2. Adaline ve Adaptif Lineer Birleştirici.....	27
2.7.3. Madaline.....	28
2.7.4. Bir kutu içindeki zeka durumu-KZD (Brain-State in a Box "BSB")	28
2.7.5. Hopfield Modeli.....	29
2.7.5.1. Ayrık Hopfield Modeli	30
2.7.5.2. Sürekli Hopfield Modeli.....	31
2.7.6. Boltzman Makinesi Modeli.....	34
2.7.6.1. Benzetilmiş Tavlama (Simulated Annealing).....	34
2.7.7. Tersyansıtma Ağ Modeli-TAM(Counterpropagation Network Model-CPN)	36
2.7.8. Kendi Kendini Organize Eden Ağ Modeli (Self-Organizing Maps- SOM)	37
2.7.9. Adaptif Rezonans Teorisi-ART	40
2.7.10. Uzay ve Zaman Koordinatlarıyla Patern Sınıflandırılması-UZKPS (Spatiotemporal Pattern Classification-STP).....	41
2.7.10.1. Uzay ve Zaman Koordinat Mimarisi Ağları-UZKMA (Architectures of Spatiotemoral Network-STN).....	42
2.7.10.2. STN Uygulamaları	43
2.7.11. Geriyansıtma Ağı (Backpropagation).....	44

BÖLÜM 3. İTÜ TRİGA MARK-II REAKTÖRÜNÜN KONTROLÜ	45
3.1 Analog Kontrolör	45
3.2 Lineer Yapay Sinir Ağı Kontrolörü.....	47
3.2.1. Sistemin Ayrık Zaman Modeli	47
3.2.2. Genelleştirilmiş Minimum Varyans Kontrol	49
3.2.2.1. Yörünge.....	50
3.2.2.2. Kontrol Sinyalinin Hesaplanması	52
3.2.3. Lineer Yapay Sinir Ağı Kontrolörü-Adaptif Lineer Birleştirici (ALC).....	59
3.3. Lineer Olmayan Y.S.A. Kontrolörü	60
BÖLÜM 4. GERİYANSITMA AĞ MODELİ.....	64
4.1 Genelleştirilmiş DELTA Kuralı.....	64
4.1.1 Çıkış Katmanı Ağırlıklarının Yenilenmesi.....	66
4.1.2. Saklı Katman Ağırlıklarının Yenilenmesi.....	69
4.2. Adaptif Parametreler.....	71
4.2.1 Minimum Değeri Bulmak İçin Yapılabilecek Diğer İşlemler.....	73
BÖLÜM 5. SONUÇ Ve ÖNERİLER.....	77
5.1. Sistem Çıkışı ALC ile bulunan simülatör için elde edilen sonuçlar.....	77
5.1.1. Başlangıç Gücü ve İstenen Güç Etkisinin İncelenmesi.....	86
5.1.1.1. Başlangıç Gücünün Etkisinin İncelenmesi.....	86
5.1.1.2. İstenen Gücün Etkisinin İncelenmesi.....	90
5.1.1.3. Örnekleme Aralığının Etkisinin İncelenmesi.....	97
5.2.Yapay Sinir Ağı Kontrolörünün Simülasyonla Elde Edilen Sonuçları	104
5.2.1. Başlangıç Gücü, İstenen Güç ve Örnekleme Aralığının Etkisinin İncelenmesi.....	109
5.2.1.1. Başlangıç Gücünün Etkisinin İncelenmesi.....	109
5.2.1.2. İstenen Gücün Etkisinin İncelenmesi.....	114
5.2.1.3. Örnekleme Aralığının Etkisinin İncelenmesi.....	119

KAYNAKLAR	124
EK-A: PROGRAM LİSTELERİ	127
ÖZGEÇMİŞ.....	138



SİMGELER DİZİNİ ve KISALTMALAR

A, B, C, D, E, G	: Polinom
z^{-1}	: Geri kaydırma operatörü
$\phi(t)$: Ölçüm vektörü
$\theta(t)$: Parametre vektörü
$U(t)$: Kontrol sinyali
$Y(t)$: Çıkış sinyali
j	: Maliyet indeksi
$\xi(t)$: Gürültü
k	: Gecikme katsayısı
$\gamma(t)$: Yardımcı çıkış
$F(t)$: Yörünge
Q, R, P	: Ağırlık polinomları
τ	: Peryot
T	: Örnekleme aralığı
$\alpha, \beta, \mu, \lambda$: Ağırlık katsayıları
f_0, f_1, f_2, f_3	: Yörünge katsayıları
X	: Giriş vektörü
ε	: hata ifadesi
y	: çıkış vektörü
d	: istenen çıkış değeri
W	: Ağırlık vektörü
ξ	: Ağırlık vektörlerinin fonksiyonu
$\Delta\omega(t)$: t. zaman adımındaki ağırlıkların değişimi
∇	: Gradient operatörü
μ	: “adım”; ağırlık vektörünün minimum hataya yakınsama hızı
$\langle \varepsilon \rangle$: Hata ifadesinin beklenen değeri
E	: Hopfield modeli enerji fonksiyonu

ŞEKİLLER DİZİNİ

Şekil 2.1. Biyolojik Sinir Hücresinin Şematik Yapısı	7
Şekil 2.2. İşlem birimi.....	10
Şekil 2.3. Ağırlık vektörleri ile gösterilmiş işlem birimi.....	11
Şekil 2.4. Çeşitli transfer fonksiyonları.....	13
Şekil 2.5. Merkezi sinir sistemindeki sinir devreleri.	15
Şekil 2.6. McCulloch-Pitts ağının basit örnekleri.....	16
Şekil 2.7. En dik eğim metodu.....	20
Şekil 2.8. Şekil 2.7 'deki ağırlık yüzeyinin değişimi	20
Şekil 2.9. Ağırlık vektörünün tahmini yörüngesi	22
Şekil 2.10. Algılama modeli yapısı.	27
Şekil 2.11. Adaline yapısı.	28
Şekil 2.12. Madaline yapısı.	29
Şekil 2.13. İki boyutta İYBH enerji değişimi.....	30
Şekil 2.14. Sürekli Hopfield modelinin hafıza devresi.....	33
Şekil 2.15 Sürekli Hopfield modelinin çıkış fonksiyonları grafiği.....	33
Şekil 2.16. Bir fonksiyonun global minimum noktasının bulunması.	35
Şekil 2.17. TAM donanımı.	36
Şekil 2.18. Farklı biçimde organize edilmiş TAM ağı.	37
Şekil 2.19. Bu şekiller SOM ağında eğitimin nasıl yapıldığını anlatmaktadır.....	39
Şekil 2.20. Ağırlık koordinatlarına göre lokasyonların çizilmesi.....	40
Şekil 2.21. ART sistemi.....	41
Şekil 2.22. Sesin dalga şekli.....	42
Şekil 2.23. Şekil 2.22'deki dalga şeklinden yapılmış spektogram.....	42
Şekil 2.24. Konuşma bilgisinden elde edilmiş güç spektrasının üretilmesi.....	43
Şekil 3.1. Analog kontrol sistemi	46

Şekil 3.2. Reaktörün basitleştirilmiş şeması	48
Şekil 3.3. Sistemin ayrık zaman modeli.....	49
Şekil 3.4. Lineer Kontrolörün blok diyagramı	50
Şekil 3.5. Yörünge	50
Şekil 3.6. Kontrol sistemi.....	57
Şekil 3.7. ALC kullanılarak sistem çıkışının kullanılması.	59
Şekil 3.8. Amaçlanan Sınır Ağı Kontrolörü.....	62
Şekil 3.9. Klasik İşlem Birimi Yapısı.....	62
Şekil 3.10. Tasarlanan İşlem Birimi Yapısı.....	63
Şekil 4.1. Üç katmanlı Geriyansıtma Ağı.....	65
Şekil 4.2. Ağırlık uzayında varsayıma dayalı bir yüzey.....	67
Şekil 4.3. Ağırlık uzayında varsayılan bir hata yüzeyi.	72
Şekil 5.1. İstenen güç düzeyi 250Kw olan reaktör simülasyonuna ilişkin grafikler....	78
Şekil 5.2. İstenen güç düzeyi 100Kw olan reaktör simülasyonuna ilişkin grafikler....	82
Şekil 5.3. Farklı güç düzeyleri için reaktör simülasyonuna ilişkin grafikler.....	78
Şekil 5.4. Farklı başlangıç güçleri için güç değişiminin gösterildiği reaktör simülasyonuna ilişkin grafikler.....	86
Şekil 5.5. Farklı başlangıç güçleri için gücün yörüngeden sapmasının gösterildiği reaktör simülasyonuna ilişkin grafikler.....	87
Şekil 5.6. Farklı başlangıç güçleri için çubuk pozisyonunun değişiminin gösterildiği reaktör simülasyonuna ilişkin grafikler.....	88
Şekil 5.7. Farklı başlangıç güçleri için kontrol sinyalinin gösterildiği reaktör simülasyonuna ilişkin grafikler.....	89
Şekil 5.8. Farklı başlangıç güçleri için ağırlık değişimlerinin gösterildiği reaktör simülasyonuna ilişkin grafikler.....	90
Şekil 5.9. İstenen gücün etkisinin incelenmesine ilişkin grafikler.....	92
Şekil 5.10. Gücün yörüngeden sapmasına ilişkin grafikler.....	93
Şekil 5.11. Çubuk pozisyonlarına ilişkin grafikler.....	94

Şekil 5.12. Kontrol sinyallerinin incelenmesine ilişkin grafikler.....	95
Şekil 5.13. Ağırlık değişimlerine ilişkin grafikler.....	96
Şekil 5.14.Örnekleme aralığının etkisinin incelenmesine ilişkin güç değişimini gösteren grafikler.....	98
Şekil 5.15.Örnekleme aralığının etkisinin incelenmesine ilişkin gücün yörüngeden sapmasını gösteren grafikler.....	99
Şekil 5.16.Örnekleme aralığının etkisinin incelenmesine ilişkin çubuk pozisyonunu gösteren grafikler.....	100
Şekil 5.17.Örnekleme aralığının etkisinin incelenmesine ilişkin kontrol sinyalini gösteren grafikler.....	101
Şekil 5.18.Örnekleme aralığının etkisinin incelenmesine ilişkin ağırlık değişimlerini gösteren grafikler.....	102
Şekil 5.19.Lineer olmayan kontrolörün 250 Kw'lık çalışma noktasına ilişkin grafikler.....	105
Şekil 5.20.Lineer olmayan kontrolörün 100 Kw'lık çalışma noktasına ilişkin grafikler.....	107
Şekil 5.21. Başlangıç gücünün etkisinin incelenmesine ilişkin güç değişimi grafikler.....	110
Şekil 5.22. Lineer olmayan kontrolörün 250 Kw'daki farklı başlangıç güçlerine ilişkin güç hatalarına gösteren grafikler.....	111
Şekil 5.23. Lineer olmayan kontrolörün 250 Kw'daki farklı başlangıç güçlerine ilişkin çubuk pozisyonlarını gösteren grafikler.....	112
Şekil 5.24. Lineer olmayan kontrolörün 250 Kw'daki farklı başlangıç güçlerine ilişkin kontrol sinyallerini gösteren grafikler.....	113
Şekil 5.25. Lineer olmayan kontrolörün istenen gücün etkisinin incelenmesine ilişkin güç değişimlerini gösteren grafikler.....	114
Şekil 5.26. Lineer olmayan kontrolörün istenen gücün etkisinin incelenmesine ilişkin güç hatalarını gösteren grafikler.....	115
Şekil 5.27. Lineer olmayan kontrolörün istenen gücün etkisinin incelenmesine ilişkin çubuk pozisyonlarını gösteren grafikler.....	116
Şekil 5.28. Lineer olmayan kontrolörün istenen gücün etkisinin incelenmesine ilişkin kontrol sinyallerini gösteren grafikler.....	117

Şekil 5.29. Lineer olmayan kontrolörün örnekleme aralığının etkisinin incelenmesine ilişkin güç değişimlerini gösteren grafikler.....	119
Şekil 5.30. Lineer olmayan kontrolörün örnekleme aralığının etkisinin incelenmesine ilişkin güç hatalarını gösteren grafikler.....	121
Şekil 5.31. Lineer olmayan kontrolörün örnekleme aralığının etkisinin incelenmesine ilişkin çubuk pozisyonlarını gösteren grafikler.....	122
Şekil 5.29. Lineer olmayan kontrolörün örnekleme aralığının etkisinin incelenmesine ilişkin kontrol sinyallerini gösteren grafikler	123



TABLolar DİZİNİ

Tablo 2.1. İlk Sınır Bilgisayarları.....	26
Tablo 5.1. Simülasyon Başlangıç Verileri.....	77
Tablo 5.2. Simülasyon Başlangıç Verileri.....	83
Tablo 5.3. Simülasyon Başlangıç Verileri.....	104



BÖLÜM 1. GİRİŞ

Bir kontrol sistemindeki, doğrusallık, zaman gecikmesi, kesim, zamana bağlı değişen parametreler gibi sistem karakteristiklerinin doğru olarak sunulmasına, geleneksel kontrol yaklaşımları çözüm getirememektedir. Bu nedenle, modern kontrol sistemlerinin daha yaygın olarak kullanılmasına ve bu teorilerin geliştirilmesine yönelik çalışmaların artmasına neden olmuştur.

Günümüzde, modern kontrol sistemlerinin uygulandığı bir çok alan olmasına rağmen, nükleer reaktörlerde kullanımları ancak üniversitelerin araştırma reaktörleri ile sınırlı kalmıştır. Nükleer reaktörler, lineer olmayan, çalışma koşullarına bağlı olarak fiziksel özellikleri değişen ve bazı fiziksel değerleri kesin olarak bilinmeyen sistemler olduğu için, sistemdeki değişimlere cevap verecek doğru matematiksel modelin kurulması güçleşmektedir.

Modern kontrol sistemlerinde önemli bir yeri olan yapay sinir ağları sistemin matematiksel modeline fazla bağlı kalmaksızın, sistemdeki değişimleri öğrenerek, karar verme prensibine dayalı bir yöntem (Öztemel 1995) olduğundan nükleer reaktörler gibi lineer olmayan sistemlerin kontrolünde kullanılması mümkündür.

Yapısı gereği paralel işlem yapabilme özelliğine sahip olan yapay sinir ağları, diğer kontrol yöntemlerine göre daha kısa zamanda işlemlerin sonucunu gerçekleştirir. Ayrıca, sinir ağı bir çok girişi aynı anda işleme sokarak, bu girişlere uygun çıkışlar üretebilme özelliğinden dolayı çok değişkenli sistemlerin kontrolünde rahatlıkla kullanılabilir (Hunt 1992).

Bu amaçla, çalışmamızda İTÜ TRIGA Mark-II Reaktörü için lineer olmayan yapay sinir ağı kontrolörü tasarlanarak bilgisayarla simülasyonu yapılmıştır.

İTÜ TRIGA Mark-II Reaktörünün kontrolüne ilişkin değişik çalışmalar gerçekleştirilmiştir. Bu çalışmalar:

1. Reaktörün en kısa zamanda gücünü istenilen seviyeye çıkarılmasını sağlayacak “muhafazakar” bir kontrol yöntemi önerilmiştir. Bu çalışmada, MIT reaktör laboratuvarında geliştirilen “reaktivite sınırlama yaklaşımı” kavramından hareketle bir yöntem önerilmektedir. Güç değişimi sırasında reaktörün periyodunu sonsuz yapmak için gereken reaktivite hızı her an hesaplanmakta ve bu hızın bir güç kontrol kriteri olarak kullanılabilmesi ifade edilmektedir. Bu çalışmada bir kontrol problemi ele alınmamış, sadece kontrol için ön koşullar oluşturulmuştur (Güngördü 1989).

2. Reaktörün kararlı hal çalışmasında, istenen güç düzeyinde meydana gelen küçük sınımların kontrol edilmesini amaçlamıştır. Bu sınımların PID kontrolör ile çözülebileceği simülasyon aracılığı ile gösterilmiştir. PID kontrolörün istenen güç düzeyi civarında maksimum ± 6 Kw’lık sapmaları kontrol edebildiği görülmüştür. Sonuç olarak, reaktörün arzu edilen güç düzeyine tutmak için PID kontrolör yönteminin kullanılabilmesi önerilmiştir. Gerçek bir kontrol problemi ele alınmamış, belli koşullar altında kontrol edilebilirlik ortaya konmuştur (Can 1991).

3. Bu çalışmada TRIGA Mark-II reaktörleri için optimal kontrol yöntemi önerilmiştir. Optimal kontrolör, ana ve yardımcı kontrolör olmak üzere iki kısımdan oluşmaktadır. Ana kontrolörde Pontragyn’ın maksimum prensibi kullanılmış ve Lagrange çarpanlarıyla tanımlanmıştır. Yardımcı kontrolör olarak PID kontrol kullanılmıştır. Gücün kübik fonksiyon şeklinde seçilmiş bir yörüngeyi takip ederek istenen güce çıkması amaçlanmıştır. Kontrolör tasarımı için KONTCAN kontrol kodu yazılmıştır (Can 1992).

4. Bu çalışmada reaktör için bir kontrol tasarımı gerçekleştirilmiştir. Yörünge üç parçalı olarak seçilmiştir. Reaktör 1 Kw’tın altındaki güçlerde çalışırken, sıcaklık geribeslemesi, sistem üzerinde etkili olmamaktadır. İTÜ TRIGA Mark-II reaktörü kural olarak düşük güçlerde, 10-500 W’da çalıştırılır, sonra yüksek güçlere çıkartılır. Düşük güçlerdeki yörünge birinci dereceden; yüksek güçlerdeki yörünge ise üçüncü

dereceden seçilmiştir. Bozucu etki altında kontrolörün davranışını gözleyebilmek için reaktöre zamanla değişen lineer bozucu etki ithal edilmiştir. Kontrolörün yüksek güç düzeylerinde büyük bozucu ithali durumunda reaktörü kontrol edemediği görülmüştür. Bu çalışmalara ilişkin sonuçlar sadece simülasyon olarak gerçekleştirilmiş, deneysel çalışmalarda nasıl bir sonuç vereceği bilinmemektedir (Can 1993).

5. Bu çalışmada İTÜ TRIGA Mark-II Reaktörünün kapalı çevrim sayısal kontrolü amaçlanmaktadır. Reaktörün kapalı çevrim sayısal kontrolü için özayarı adaptif kontrolör önerilmiştir. Bunun için hazırlanan CANER yazılımında, YAVCAN kodu kullanılarak reaktör simülasyonu yapılmıştır. Reaktörün kapalı çevrim sayısal kontrolü ilk kez bu çalışmayla gerçekleştirilmiştir.

Genelleştirilmiş minimum varyans yöntemi kullanılarak gerçekleştirilen kontrolör; parametreleri bilinmeyen ikinci dereceden lineer model kullanılarak tasarlanmıştır. Gerçekte lineer olmayan reaktörün, lineer modelle yeterince iyi temsil edilebilmesi için kontrol parametreleri gerçek zamanda yenilenmektedir. Yenileme işleminde, en küçük kareler yöntemiyle, rekürsif parametre kestirimi yapılmaktadır.

Bu çalışmada yörünge kübik fonksiyon olarak seçilmiştir. Kontrolör, reaktör gücünün verilen yörüngeyi izlemesini sağlamaktadır. Bu amaçla, kontrol çubuğu hızı kontrol değişkeni olarak seçilmiştir.

Simülasyon çalışmalarında; başlangıç kararlı hal değeri, güce çıkış periyodu, çıkılan güç düzeyi, örnekleme zamanı ve μ ağırlık katsayısının kontrolör davranışı üzerindeki etkileri incelenmiştir. Yapılmış simülasyonların, önerilen kontrolörün, reaktör gücünü başlangıç kararlı hal değerinden istenilen güç düzeyine verilen yörüngeyi izleyerek çıkarabildiği ve kararlı hal hatasının olmadığı görülmüştür.

Önerilen kontrolörün gerçek çalışma koşulları altındaki davranışını izlemek amacıyla, gerekli donanım hazırlanarak, reaktörde çeşitli deneyler yapılmıştır. Deneysel çalışmalarda, tasarlanan kontrolörün reaktördeki istenilen güç değişikliğine anında cevap vererek sistemi istenilen güç düzeyine getirdiği görülmüştür (Erkan 1994).

6. Bu çalışmada reaktör için kural tabanlı bulanık kontrolör geliştirilmiştir. Kontrolör tasarımı, reaktörün peryot, güç, çubuk konumu ve sıcaklık değişkenleri, bulanık kontrolörün giriş değişkenleri olarak seçilmiştir.

Kuaral tabanı, güç-peryot, güç-kontrol çubuğu, güç-sıcaklık değişken grupları ile ilgili 30 adet kural kullanılarak oluşturulmuştur.

Değişkenler, ikinci dereceden lineer üyelik fonksiyonları kullanılarak bulanıklaştırılmışlardır. Güç için yedi, peryot için altı, çubuk konumu için dört ve sıcaklık için bir adet üyelik fonksiyonu kullanılmıştır.

Önerilen kontrolörün değişik çalışma koşulları altındaki davranışını, YAVCAN simülasyon kodu kullanılarak simüle edilmiştir. Yapılan simülasyonlarda, bulanık kontrolörün sistemdeki değişen koşullar altında gücü, aşırı bir şekilde istenen seviyeye çıkartabildiği ve tam güçte salınımlar olmadığı görülmüştür.

Tasarlanan kontrolörün gerçek çalışma koşulları altındaki davranışını incelemek için, reaktörde deneyler yapılmıştır. Bulanık kontrolörün, reaktördeki analog kontrolöre nazaran sistemi istenen güç düzeyine daha kısa sürede getirdiği ve tam güçte salınımların daha az olduğu tesbit edilmiştir (Baba 1995).

7. Bu çalışmada, İTÜ TRIGA Mark-II Reaktör kalbi için zamanla değişen iki boyutlu bilgisayar simülasyon yazılımı, Turbo C++ 3.0'da hazırlanmıştır. Simülasyon için iki temel model kullanılmıştır. Bunlar nötronik ve termohidrolik modeldir.

Nötronik model olarak yer-zaman bağımlılığını ifade etmek için kuple nodal kinetik denklemler kullanılmıştır. Bu amaçla reaktör elli noda ayrılmıştır. Çalışma için kuplaj katsayıları hesaplanmıştır. Ayrıca modelde, yakıt sıcaklığı reaktivite geribeslemesi dikkate alınmıştır. Kalbe kontrol çubukları ile reaktivite girişi vardır.

Termohidrolik model, nodlara ait yakıt sıcaklıkları, moderatör sıcaklıkları, moderatör hızları ve yoğunluklarına ait denklemlerle ifade edilmiştir. Ayrıca yakıt ve moderatöre ait parametrelerin sıcaklık bağımlı olduğu kabul edilmiştir. Moderatör hızları, momentum korunumu denklemlerinden elde edilmiştir. Modellerden elde edilen diferansiyel denklemlerin çözümü için “Semi-Implicit” ve “Implicit” yöntem kullanılmıştır (Özkan 1995).

Çalışmamızın diğer bölümlerinde sırasıyla aşağıdaki konular anlatılmıştır:

Bölüm 2’de, yapay sinir ağları ve Y.S.A. modelleri anlatılmıştır.

Bölüm 3’de İTÜ TRIGA Mark-II Reaktörünün kontrolüne ilişkin olarak, analog kontrolör, lineer yapay sinir ağı kontrolörü ve lineer olmayan yapay sinir ağı kontrolörü yapıları açıklanmaktadır.

Bölüm 4’de tasarladığımız kontrolörün kullandığı geriyanıtma yapay sinir ağı modeli anlatılmaktadır.

Bölüm 5’de, kontrol sistemine ait simülasyon sonuçları incelenmiştir ve gerekli görülen yerlerde öneriler sunulmuştur.

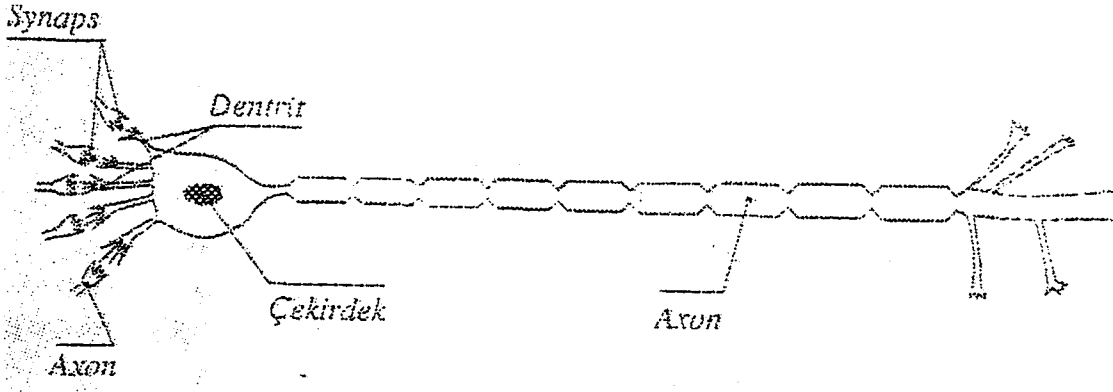
Ek-A’da yapay sinir ağının eğitimini yapan PASCAL programlama dilinde yazılmış program listesi verilmiştir.

BÖLÜM 2. YAPAY SİNİR AĞLARI

Bir sinir hücresi, sinir ağlarının en temel elemanlarından birisi olup sinir sistemi içerisindeki fonksiyon ve görevlerine göre değişik şekil ve büyüklükte olabilir. Bütün hücrelerin ortak bazı özellikleri vardır. Şekil 2.1 'de gösterildiği gibi, bir hücrenin bir ucunda bir grup liflere benzer "dentrit" adı verilen ve hücreye diğer hücrelerden veya dış dünyadan bilgiler (sinyaller) getiren bağlantı elemanları, diğer ucunda ise tek bir life benzer "axon" adı verilen ve hücrelerden diğerlerine veya dış dünyaya bilgiler (sinyaller) taşıyan bağlantı elemanı vardır. Bu axon daha sonra diğer hücrelerle birleşme sırasında dallanarak dallara ayrılmaktadır. Bu iki uçtaki bağlantı noktalarının elektrofizyolojik olarak hücrelerdeki bilgileri işlemede önemli yeri vardır. Hücrelerin birbirleri ile elektrik sinyalleri şeklinde birleştiği belirlenmiştir. Sinyaller bir hücrenin axon 'undan diğerinin dentrit 'ine gönderilir. Bir axon birden fazla dentrit ile ilişkiye girebilir: bu bağlantının yapıldığı yere sinaps (synaps) denir. Hücreler elektrik sinyalini hücre duvarlarındaki voltajı değiştirerek üretirler. Bu ise hücrenin içinde ve dışında dağılmış bulunan iyonlar aracılığıyla olur. Bu iyonlar sodyum, potasyum, kalsiyum ve klorin gibi iyonlardır. Bir hücre diğer bir hücreye elektrik enerjisini bu kimyasal iyonlar aracılığıyla transfer eder. Bazı iyonlar elektrik ve manyetik kutuplaşmaya sebep olurken, bazıları kutuplaşmadan kurtulup hücre zarını açarak iyonların hücreye geçmesini sağlar. Zaten sinyallerin bir hücreden diğerine akmasını sağlayan da bu kutuplaşmanın azalması olayıdır. Sinyaller hücrenin etkinliğini (dürtüsünü) belirler. Bir hücrenin etkinliği, hücreye gelen sinaps sayısı, sinaps 'lardaki iyonların konsantrasyonu ve sinaps 'ın sahip olduğu güç olmak üzere üç faktöre bağlıdır. Bir hücre sahip olduğu dürtü miktarınca diğer hücreleri etkiler. Bazı hücreler diğerlerinin dürtülerini pozitif yönde, bazı hücrelerde negatif yönde etkiler. İnsan beyni bu şekilde çalışan milyonlarca hücrenin bir araya gelmesinden oluşur.

Biyolojik beynin en önemli özelliklerinden birisi de öğrenme olayıdır. İnsanlar ve hayvanlar sürekli olarak buldukları çevre ile ilişkiler sonucunda bir öğrenme işlemi içerisindeyler. Öğrenilen her yeni bilgi hemen beyin fonksiyonlarını etkileyerek davranışlarda da kendini gösterir. Yapay sinir ağlarının geliştirilmesinde bu özellik esas alınır.

Yapay sinir ağıları da biyolojik sinir ağılarının bilinen temel bir kaç özelliğini (öğrenme kabiliyeti gibi) benzeterek geliştirilmiştir. Bir takım özellikler ise nörofizyolojik yaklaşımlar yerine mühendislik yaklaşımı ile geliştirilmektedir. Bir yapay sinir ağının yapısını belirleyen bazı faktörler vardır. Yapay sinir hücreleri veya mühendislik deyişiyle işlem birimleri (processing elements), sinir ağının yapısal topolojisi, ağın sahip olduğu öğrenme kural ve stratejisi bunların başında gelir. (Öztemel 1995)



Şekil 2.1. Biyolojik Sinir Hücresinin Şematik Yapısı

Y.S.A. (Yapay Sinir Ağları) değişik biçimlerde tanımlanmaktadır. Bu tanımlamalardan başlıcalarını şöyle sıralayabiliriz:

"Biyolojik modellerden esinlenmiş yeni bir işlem biçimi"

"Seviyelere ayrılarak sınıflandırılmış çok sayıda işlem yapan elemanın birleşmesinden oluşan matematiksel model"

"Bilgiyi iç girdilerin dinamik durumuna göre işleyen, birbirine bağlı işlem yapan basit elemanların oluşturduğu hesaplama sistemi".

Yapay Sinir Ağları; yaptıkları işe ve modellerine göre başka isimler de alabilir. Bunlar:

- Paralel sıralı işlem modelleri,
- Birleşimci / Birleşimsel modeller,
- Adaptif sistemler,
- Özayarlı organizasyon sistemleri,
- Sinirsel bilgi işleme,
- Sinir yapıli sistemler.

Bir bilgisayar, belli bir zamanın sadece bir diliminde bilgiyi saklayabildiğinden, çok miktardaki bilginin yüklenme hızı için sınırlar vardır. Bir çok işlemci paralel çalıştığı halde, ardışıl işlemlerin tamamlanmasını beklerken çok zaman boşa gider. Paralel işlemleri verimli ve etkin kullanabilen programlar yazmak da zordur.

Y.S.A., isminden de anlaşılacağı gibi, zihinsel işlemlerin yaklaşık modelidir. Paralel yapıları sayesinde, eş zamanlı bilgiyi işleyebilir ve çözüm üretebilirler. Ağ, bütün bu çözümleri yapabilmek için, bir öğrenme algoritmasına ihtiyaç duymaktadır. Yani ağ; çözüm üretebilmek için bir takım bilgileri öğrenmek zorundadır. Genel anlamda öğrenme; ağırlıkların yenilenmesi ile olur. Ağ amaçlanan işin örneklerini ve bu işin beklenen tahmini sonuçlarını önceden öğrenir. Bu öğrenmede programcının veya kullanıcının hiç bir teşviki ya da bazı öğeleri daha fazla öğretebilmesi söz konusu değildir. Zaten ağ kendi çağırışmalarını kendisi yaratarak öğrenecektir. Örneğin, ağ; girişlerin sık sık tekrarlanan değerlerini daha iyi öğrenecektir. Y.S.A. 'nın özelliği mekanik olmaktan çok, insan özellikleri taşıması yani öğrenmesi, unutması ve hatırlamasıdır. Ayrıca Y.S.A. diğer yöntemlerle çok zaman alan işleri daha kısa zamanda yapar.

2.1. Y.S.A.'nın Uygulama Alanları

Y.S.A. 'nın uygulandığı bir çok alan vardır. Bu alanlar ve uygulamaları:

a) Biyoloji:

- Beyni ve diğer sistemleri daha iyi anlama,
- Retina ve cochlea 'yı modelleme.

b) İş Dünyası:

- Petrol ve jeolojik yapı değişimlerinin tahmini,
- Özel durumlar için toplum eğilimlerinin tanımı,
- Veri tabanı oluşturulması,
- Havayolları yer ve ücret düzenlemesi,
- El yazısı karakterlerini tanıma.

c) Çevresel:

- Numuneleri analiz etme,
- Hava tahmini.

d) Finans:

- Kredi riski değerlendirilmesi,
- Sahte para ve evrak tanıma,
- El yazısı formlarının değerlendirilmesi,
- Eğilim yatırımları ve portföy analizi.

e) Üretim:

- Robot ve kontrol sistemlerini otomatikleştirme (görüntü ve hissedicilerle basınç, sıcaklık ve gaz kontrolü),
- Üretim işlemi kontrolü,
- Kalite kontrolü,
- Montaj hattında parça seçimi.

f) Tıp:

- Sağırılar için ses analizi,
- Semptom hastalıklarının teşhis ve tedavisi,
- Ameliyat görüntüleme,
- İlaçların yan etkilerinin analizi,
- X-Işınlarnı okuma,
- Epileptik felcin nedenlerini anlama.

g) Askeri:

- Radar sinyallerini sınıflandırma,
- Yeni ve gelişmiş silahlar yaratma,
- Keşif yapma,
- Kıt kaynakların kullanımını optimize etme,
- Hedef tanıma ve izlemedir.

Uygulamaların bir çoğu patern (örüntü) tanıma işi yapmak zorundadır. Bir örnekler serisinde paternler bulunur. Bu paternler sınıflandırılarak, örnekler serisinin bir parçasından (bir set içinden) belli bir paternin tamamı tanımlanır veya bir patern içindeki benzetilemeyen bozuk patern değerleri (bozuk pikseller) temizlenerek, patern tekrar tanımlanır.

2.2. Y.S.A. 'nın Yapısı ve Etkinlik Şekilleri

Y.S.A. 'nın çalışma şeklini anlamak için biyolojik sinir sistemi gözönünde bulundurularak, sinaps ve Y.S.A. etkinliklerinin incelenmesi gerekmektedir.

2.2.1. Sinaps etkinliği

Y.S.A. ile biyolojik sinir hücresi benzeşimi için aşağıdaki noktalar önemlidir:

- Sinir uçlarına gelen sinyallerin her birinin bir ağırlık değeri vardır. Yani sinir

uçlarına gelen bütün girişler eşit şekilde etkinleştirilmemiştir ve gelen sinyaller; ağırlık değerleri ile çarpılarak toplanmıştır.

- Elde edilen toplam, eşik değerinin üzerinde ise, sinir aktif olur, yani etkinleşir.

- Sinir sisteminde sinyallerin rahatlıkla iletimi; sinir sisteminin etkinliğine bağlıdır. Sinapslar; hücrelerdeki oksijen eksikliği, yorgunluk gibi anestetik nedenlerden olumsuz yönde etkilenebilmektedir. Bu etkiler, hücreler arasında gidip gelen sinyallerin geçişine engel olmaktadır. Bu nedenle, sinir sistemi etkinliği; sinapsları bu olumsuz etkilerden arındırarak, sinyallerin düzenli iletimini sağlamaktadır.

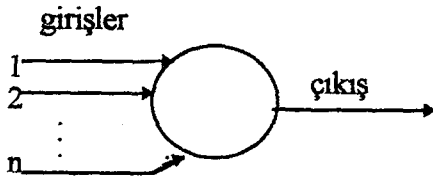
Sinir uçlarına gelen sinyallere giriş adı verilir. Her bir girişe gelen pozitif veya negatif sinyaller bir ağırlık katsayısı ile ağırlıklandırılmışlardır. Girişlerin toplamı, sinir hücresinin eşik değerine eşitse ya da büyükse, sinir hücresinin ucu aktif olur.

2.2.2. Y.S.A. etkinliği

Y.S.A.'nın etkinliği tanımlanırken, genellikle ağırlık, değişik girişlere karşı tepki ve davranışları üzerine konuşulur. Y.S.A. 'nın etkinlikleri aşağıdaki şekilde ifade edilebilir:

- davranır,
- tepki verir,
- kendi kendini organize eder,
- öğrenir,
- genelleme yapar,
- unuttur (durum değişince, eski öğrenmeler unutulur).

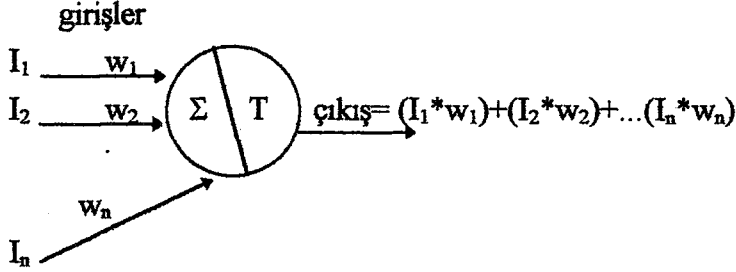
Bir birime gelen giriş sinyallerinin toplamını hesaplayıp, bu toplamı eşik seviyesiyle karşılaştırarak, çıkışın ne olacağına karar veren birime işlem birimi denir. Şekil 2.2 'de bir işlem biriminin yapısı gösterilmektedir.



Şekil 2.2. İşlem birimi

Bir sinir hücresine (neuron) gelen bir çok giriş vardır, bir işlem birimi (İ.B.) için de bir çok giriş sinyali olmalıdır. Bunların tümü İ.B.'ne aynı anda gelmelidir ve İ.B., bir çıkışa sahip olmalıdır.

İşlem birimine gelen girişlerin her birinin bir ağırlık değeri olduğu bilinmektedir. Bu ağırlıkların işlem birimine bağlantısı Şekil 2.3 'deki gibidir.



Şekil 2.3. Ağırlık vektörleri ile gösterilmiş işlem birimi

Girişler; işlem birimine bağlandıkları ağırlık değerleri ile çarpılır. Sonra iç çarpımı alınmış bu girişlerin toplamı alınır, bulunan toplam vektörel değil, skalerdir.

Geometrik olarak, iki vektörün iç çarpımı, bunların benzerliklerinin ölçümü olarak düşünülebilir. Vektörlerin yönleri aynı ise iç çarpım maksimum, eğer yönler zıt ise iç çarpım minimum olacaktır.

Şekil 2.3 'deki Σ , ağırlık değerleri ile çarpılmış girişlerin toplamını ve T de, transfer fonksiyonunu ifade etmektedir. Transfer fonksiyonu gibi başka fonksiyonların hesaba katılmasının amacı, çıkışların zamana bağlı olarak değişmelerini sağlamaktır.

Y.S.A. 'nın ilk ağırlık değerleri, bir algoritma ile ya da rastgele seçilerek düzenlenebilir. Rastgele seçilen ağırlıklar, çıkış katmanında istenen çıkış değerine erişilinceye kadar ayarlanır. Ağırlıkların ayarlanması sırasında çıkıştaki istenen değer ile o andaki gerçek değer farkı alınarak bir hata ifadesi hesaplanır. Hesaplanan hata ifadesi, diğer katmanlara, geriye doğru yansıtılır. Hatanın geriye doğru yansıtılması çok katmanlı ağlar için en popüler öğrenme algoritmasıdır. Geri yansıtma algoritmaları; öncelikle çıkış katmanına bağlı olan ağırlıkları ayarlar. Ardından, her katmandaki hataları azaltmak için, giriş katmanına doğru, geriye çalışarak her katmandaki ağırlıkları ayarlar. Bu işlem defalarca tekrarlanarak, ağırlıkların ayarlanması ile öğrenme olayı gerçekleştirilmiş olur.

Öğrenme işleminin olmadığı ağlar da bulunmaktadır. Bu tür ağlarda tüm ağırlıklar sabittir ve bu ağırlıklar her hangi bir ayarlamaya gerek duyulmadığından değişmez.

Fakat, ağırlıkları değiştirebilme sisteme adaptasyon yeteneği kazandırır, böylece farklı zamanlardaki benzer başlangıç koşullarında farklı sonuçlar oluşur.

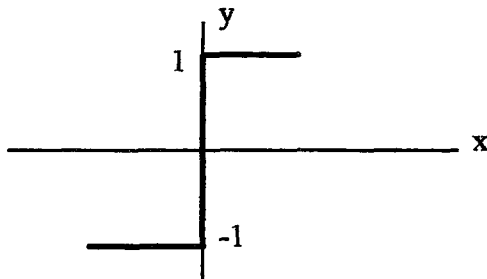
2.3. Yapay Sinir Ağının Organize Edilmesi

Y.S.A. organizasyonunda **Nörodinamik** ve **Yapı "Mimari"** olmak üzere iki sınıflandırma kullanılabilir. Bu sınıflandırmalar aşağıda ayrıntılı olarak açıklanmıştır.

2.3.1. Nörodinamik

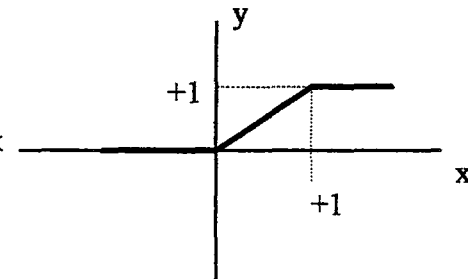
Bu sınıflandırmada, sadece yapay sinirin özellikleri tanımlanmaktadır. Yani, işlem biriminin girişleri, birime gelen ağırlıkların ayarlanması, çıkışın üretilmesi ve çıkışların transferlerinin tüm durumlarını belirlenmektedir. Bir sinir ağının giriş katmanını oluşturan girişler; diğer katmanın girişlerini oluşturacak bir tampon görevi görmekte ve ağa uygulanan girişlerin bilgisayar tarafından anlaşılabilir değerlere getirilerek, normalizasyon işlemini üstlenmektedir. Çıkış katmanı ise, ağ bilgilerini kullanıcıların anlayabileceği hale getirerek dış dünyaya sunmaktadır.

Herhangi bir anda girişler ve çıkışlar arasındaki ilişkiyi transfer fonksiyonu belirler. Ağırlıklandırılmış girişlerin toplamı; işlem biriminin çıkışını gösteren, transfer fonksiyonunun girişi haline gelir. Transfer fonksiyonunun eşiği, genellikle lineer değildir. Lineer fonksiyonda çıkış, temelde girişle orantılı olduğundan, giriş değerlerine göre çıkışın alabileceği değerler sınırlıdır. Bu nedenle, kullanışlı bir ağda, özel durumlar dışında lineer transfer fonksiyonu fazla kullanılmamaktadır.



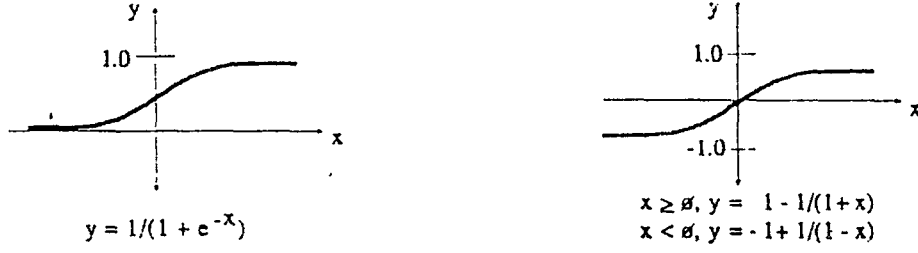
$$\begin{aligned} x < 0, & \quad y = -1. \\ x \geq 0, & \quad y = 1. \end{aligned}$$

(a) Sert limitleyici fonksiyonu



$$\begin{aligned} x < 0, & \quad y = 0, \\ 0 \leq x \leq 1, & \quad y = x, \\ x > 1, & \quad y = 1. \end{aligned}$$

(b) Rampa fonksiyonu



(c) Sigmoid fonksiyonları

Şekil 2.4. Çeşitli transfer fonksiyonları

Şekil 2.4 'de çeşitli transfer fonksiyonları gösterilmiştir. Sert limitleyici fonksiyonunda; girişler 0 ve 1 arasında (0 dahil) ise çıkış +1, girişler 0 'dan küçük ise çıkış -1 veya 0 olur. Rampa fonksiyonunda; girişler 0 ve 1 arasında değişiyor ise, çıkış girişlerin değerlerine sahip olacaktır. Girişler 0 'dan küçük ise çıkış 0, girişler 1'den büyük ise çıkış 1 olacaktır.

Bu fonksiyonların en sık kullanılanı sigmoid fonksiyonudur. Sigmoid fonksiyonunda; eğrinin eğimini ayarlamak için değişiklikler yapılabilir ve türevlerinin hesaplanması da nispeten kolaydır. Sigmoid 'in başka bir şekli, logaritmasız olanıdır. Bu temel düşünceler üzerine, sinüs fonksiyonu, hiperbolik tanjant ve diğer lineer olmayan değişimler de kullanılabilir. Bu transfer fonksiyonu, bir dinamik bölge sınırlayıcısı sağlar. Eğer girişler uygun bir bölgedeyse, girişleri ayırmak mümkün hale gelir. Böyle sınırlamalar olmazsa, giriş bilgilerinin çoğu, bölgenin dışında çıkışlar üretecektir, bu durumda da çıkış aktarımı zor olacaktır. Bu, $x < -1$ veya $x > +1$ olduğu sigmoid fonksiyonunun ikinci biçiminde gerçekleşecektir. $x = 10$ için çıkış değeri, $x = 100$ için çıkış değerine yaklaşık eşit olacaktır.

2.3.2. Yapı "Mimari"

Bu sınıflandırmada ağda kullanılacak işlem birimlerinin, katmanların sayısı ve bağlantıların şekli belirlenmektedir. Aşağıdaki her paragrafta bu özellikler sırasıyla anlatılmaktadır.

Giriş ve çıkış katmanları için gerekli birimlerin sayısı, sistemin giriş ve çıkışlarının sayısına bağlı olacaktır. Saklı katmanlarda kullanılacak birimlerin sayısı belirsizdir. Bu durumda, en iyi sonucu, deneyerek bulmak mümkündür. Bunun yanında saklı katmanlara fazladan birimler eklemek, sistemde yeni gelişmeler sağlamayabilir.

Bir Y.S.A.'nda kullanılacak katman sayısı, uygulanacağı sistemin karmaşıklığına bağlıdır. Yani, sistemin girişleri fazla ise bu girişleri ağına çok iyi anlayabilmesi için birden fazla saklı katman kullanılmalıdır. Bazı Y.S.A. modelleri; giriş patern'i (örüntü) direk olarak çıkış patern'ine yerleştirerek, sadece iki katman kullanırlar. Bu, giriş-çıkış arasında iyi bir benzerlik olduğunda yeterlidir. Fakat, giriş paterni yapısı, çıkıştaki paternden oldukça farklı ise, giriş sinyallerinin tümünü olmasa bile bir kısmını temsil eden saklı birimlere ihtiyaç vardır. Daha çok bilgiyi işleyebilme, ağıdaki katman sayısının artmasına bağlıdır. Bu çok önemli bir özelliktir.

Eğer işlem biriminin çıkışı, bir sonraki katmandaki her birime bağlanmış ise bu birim diğer birimlere "tam olarak (fully) bağlanmıştır" denir. Yakın komşu düğüme bağlı olmak da başka bir seçenektir. Bazı ağ modellerinde (örneğin Hopfield), birimler seyrek olarak bir kaç birime bağlıdırlar, ve birimlerin ancak küçük bir kısmı bir önceki katmanın birimlerinden rastgele giriş alırlar.

Bir ağ sadece bir sonraki katmana çıktı gönderebiliyorsa, **ileribeslemeli** ağ olarak adlandırılır. Bir **geribeslemeli** ağ, çıkışlarının bir önceki katmana giriş olmasını sağlar ve **yanbeslemeli** birleşmeler, aynı katmanda bazı girişleri diğer birimlere gönderir. Bu kapalı döngülü ağlar **geridönüşümlü** ağlar olarak adlandırılırlar.

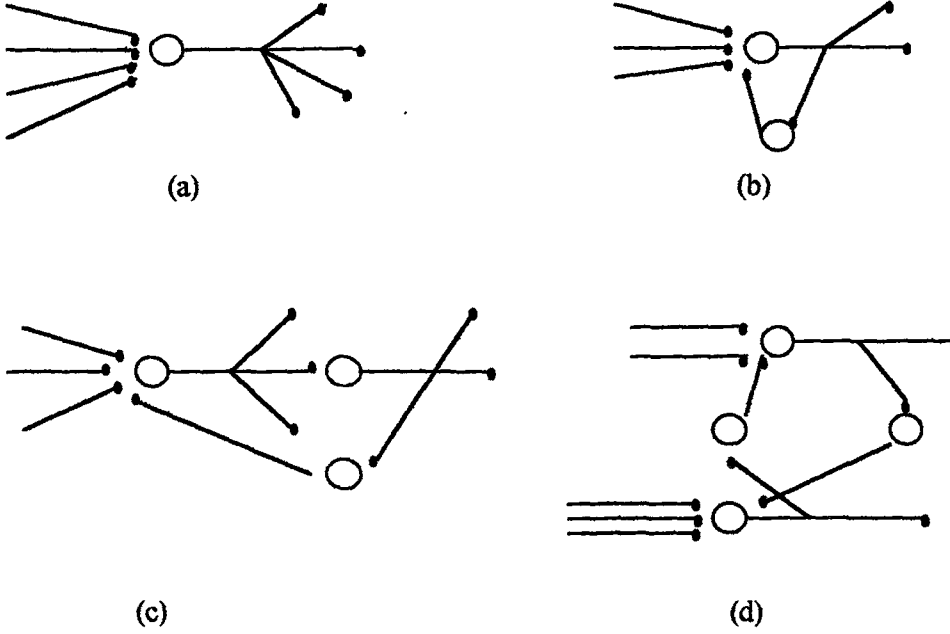
İleribeslemeli ağlar, **geribeslemeli** ağlardan hızlıdırlar, çünkü sadece bir gönderimle bir çözüm oluşturabilirler. Ayrıca, **ileribeslemeli** ağlar kararlılığı da garanti eder. Ancak, sistem kararlı olana dek, **ileribeslemeli** ağın bir çok çevrimi tekrarlama gereklidir. Bundan sonra bile sistem kararlılığa erişmeyebilir. Yakınsadığı değerler arasında tereddüt eder, veya kilitlenir.

2.4. Sinir Devreleri ve Hesaplamaları

Şekil 2.5. merkezi sinir sisteminde görülen bir kaç temel sinir devresini göstermektedir.

Şekil 2.5. 'teki şekillerin her biri merkezi sinir sistemindeki sinir devrelerinin örneklerini gösterir. (a) ve (b) yakınsama ve uzaksama kavramlarını içerirken (b), (c) ve (d) de geribeslemeli devrelere örnek olarak gösterilebilir.

Her sinir diğer tüm sinirlere impulslar gönderir ve bir çok sinirden impulslar alır (yakınsama). Şekil 2.5(b), (c) ve (d) 'deki geribeslemeli synaptic bağlantıları uyarıcı veya yasaklayıcı olabilir. Bu devreler sayesinde kontrol sistemlerinin pozitif veya negatif geribeslemeli yapılması mümkün kılınmıştır.



Şekil 2.5. Merkezi sinir sistemindeki sinir devreleri.

McCulloch - Pitts 1943 'te beş kabul üzerine kurulmuş bir teori geliştirmiştir. (Freeman and Skapura 1991).

1. Bir sinir, "1" veya sıfır "0" durumuna göre aktif veya aktif değildir.
2. Bir sinirin uyarılması, belli sayıdaki sinaps 'lerin toplamının eşik değerinden büyük olmasına (>1) ve uyarılma süresine bağlıdır.
3. Sinir sistemi içindeki önemli gecikme sinaptik gecikmedir.
4. Her hangi bir yasaklayıcı sinapse aktivitesi, o anda, sinirin uyarılmasını kesinlikle engeller.
5. Kurulu bir ağın kullandığı bağlantı yapısı, işlemler süresince aynıdır, işlemler devam ederken ağın bağlantı yapısı değiştirilemez.

1. kabul sinirlerin binari olarak yani (on veya off) aktif veya aktif olmadıklarını gösterir. $N_i(t)$, i. sinirin t zamanında yandığını yani aktif hale geçtiğini ifade eder.

$\neg N_i(t)$ notasyonu ise i. sinirin t zamanında yanmadığını gösterir.

Şekil 2.6. 'daki her şeklin lojik notasyon ifadesi, şekillerin altında verilmiştir. Küçük içi boş daireler yasaklayıcı bağlantıları, içi dolu daireler uyarıcı bağlantıları gösterir.



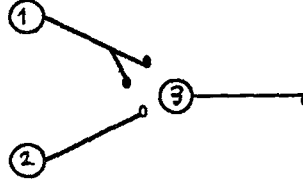
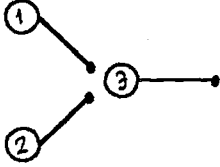
$$N_2(t) = N_1(t-1)$$

"disjonksiyon"

"2. sinir 1. sinirden sonra yanar, aktif hale geçer".

(a)

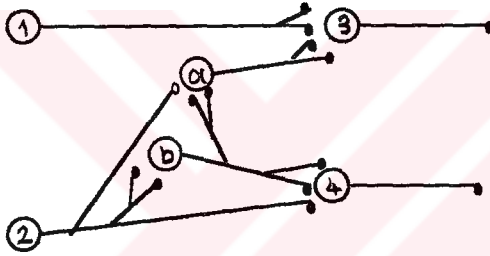
(b)



$$N_3(t) = N_1(t-1) \& N_2(t-1) \quad N_3(t) = N_1(t-1) \& \neg N_2(t-1) \text{ "Olumsuz Birleştirme"}$$

(c)

(d)



$$N_3(t) = N_1(t-1) \vee N_a(t-1) \text{ ve } N_a(t-1) = N_b(t-1) \& \neg N_2(t-2) \text{ ise}$$

$$N_3(t) = N_1(t-1) \vee [N_b(t-1) \& \neg N_2(t-2)]$$

$$N_4(t) = N_b(t-1) \& N_2(t-1) \text{ ve } N_b(t-1) = N_2(t-2) \text{ ise}$$

$$N_4(t) = N_2(t-1) \& N_2(t-2)$$

(e)

Şekil 2.6. McCulloch-Pitts ağının basit örnekleri

2.5. Öğrenme Modları ve Kuralları

2.5.1. Öğrenme modları

İki çeşit öğrenme modu vardır: **eğitici (Supervised)** ve **eğitici (Unsupervised)**. Eğitici öğrenme modu bir öğretici gerektirir. Bu öğretici; veri eğitim seti veya ağın performansını derecelleyen bir gözlemci olabilir.

2.5.1.1. Eğitici (Supervised) öğrenme modu

Bu mod içinde, ağırlıklar, ilk olarak rastgele verilerek, ağ tarafından ayarlanır, bir sonraki iterasyonda, sinir ağının gerçek çıkışı, istenen çıkışla karşılaştırılarak, ağırlıklar daha yakın bir değere ayarlanır.

Eğitici öğrenmede, sinir ağ bir sistem içinde aktif olarak kullanılmadan önce eğitilmelidir. Eğitim; ağa sunulan giriş ve çıkış verilerini içerir. Bu veriye eğitim seti adı verilmektedir. Bu set içinde, her giriş için geçerli istenen çıkış yer almaktadır. Ağın eğitim kısmı çok uzun sürebilir. Eğitim işi, verilen bir girişler dizisi için ağın istenen çıkışları üretmesine kadar sürer. Daha fazla öğrenme işlemine devam edildiğinde, ağırlıklar daha fazla değişmez. Bu tekniği kullanarak, bir ağ modeli; karar verme, bilgi ezberleme veya genelleme gibi işleri yapabilir.

2.5.1.2. Eğitici (Unsupervised) öğrenme modu

Bu öğrenme moduna bazen kendi-eğitici (self-supervised) öğrenme modu da denir. Burada, ağın ağırlıkları dışardan etkiler kullanılarak ayarlanmaz. Bunun yerine, ağın performansını kendiliğinden izlemesi söz konusudur. Ağ, giriş sinyallerinin yönüne veya düzenliliğine bakar ve ağın fonksiyonuna göre adaptasyon yapar. Ağ doğru ya da yanlış olduğu söylenmemiş olsa bile, ağ kendini nasıl organize edeceği hakkında yine de bir miktar bilgiye sahip olmalıdır.

Bir eğitici öğrenme algoritması işlem birimleri arasında işbirliği yapmalıdır. Böyle bir tasarıda, kümeler birlikte çalışarak birbirlerini uyardırmaya çalışacaktır. Eğer dışardan bir giriş, kümedeki herhangi bir düğümü etkinleştirirse, bu kümenin tümünde etkinleşme arttırılabilecektir. Tersine, bir düğümüne gelen giriş, kümedeki etkinliği düşürücü yönde olursa, bu küme üzerinde girişin yasaklayıcı (inhibitory) etkisi söz konusudur.

İşlem birimleri arasındaki yarışma (rekabet), öğrenme için temel oluşturmaktadır. Rekabet halindeki kümelerin eğitimi, belirli grupların belirli uyarılarına karşı yanıtlarını kuvvetlendirebilir, bu grupları birbirleriyle ve belirli uygun bir yanıt ile ilişkilendirebilir. Örneğin, işlem birimleri, yatay veya dikey kenarlar veya sağ ve sol kenarlar gibi çeşitli patern özellikleri arasındaki ayrımı yapmak için organize olabilirler.

2.5.2. Öğrenme Kuralları

2.5.2.1. En küçük kareler yöntemi KY (Least-Mean-Square "LMS") veya DELTA Öğrenme Kuralı

Bir işlem biriminin istenen çıkış değeri ve geçerli çıkış değeri arasındaki farkı (delta) indirmek için bu birimler arasındaki bağlantıların kuvvetliliğinin, sürekli olarak sağlanmasının fikri üzerine kurulu bir kuraldır.

Verilmiş $\{X_1, X_2, \dots, X_L\}$ gibi bir X vektörü, ağırlık seti W ile bir y çıkış değeri verecektir. Her bir giriş vektörü istenilen doğru d_k çıkış değerine sahip olacaktır. Burada, $k = 1 \dots L$ değerlerini almaktadır. Bir tek ağırlık vektörünün her bir giriş vektörü ile ilişkilendirilerek istenilen çıkış değerinin bulunması pek kolay değildir. Bu yüzden KY öğrenme yöntemi geliştirilmiştir. Bu öğrenme, Adaptif Lineer Birleştirici ALB 'ye (Adaptive Linear Combiner-ALC) sunulan girişler ve istenilen çıkışları kendi kendine adapte edebilir. ALB doğru çıkışları verinceye kadar her bir giriş-çıkış kombinasyonları için ağırlık değerlerinde küçük ayarlamalar yapılır.

2.5.2.1.1. Ağırlıkların Hesaplanması

X_k giriş vektörüne göre; istenilen çıkış değerleri d_k ile en iyi ağırlık vektörü W^* için, istenilen çıkış ve gerçek çıkış arasındaki fark minimize edilmelidir. Bu işlem her giriş vektörü için tekrarlanmalıdır. Buradaki yaklaşım, giriş vektör seti için küçük kare hatasını (mean squared error) minimize etmektir.

Eğer k . giriş vektörü için gerçek çıkış değeri y_k ise, ilgili hata ifadesi

$$\epsilon_k = d_k - y_k \quad (2.1)$$

olur. Küçük kare hatası veya beklenen hata değeri şu şekilde tanımlanır.

$$\langle \epsilon_k^2 \rangle = \frac{1}{L} \sum_{k=1}^L \epsilon_k^2 \quad (2.2)$$

Burada L eğitim setindeki giriş vektörlerinin sayısıdır. Eşitlik (2.1) kullanılarak küçük kare hata ifadesini aşağıdaki gibi genişletilebilir.

$$\langle \epsilon_k^2 \rangle = \langle (d_k - W^t \cdot X_k)^2 \rangle \quad (2.3)$$

$$= \langle d_k^2 \rangle + W^t \langle X_k \cdot X_k^t \rangle W - 2 \langle d_k X_k^t \rangle W \quad (2.4)$$

Eşitlik (2.3) 'ten (2.4) 'e geçişte, eğitim setinin istatistiksel olarak değişmediği tahmin edilmektedir. Bunun anlamı, her hangi bir beklenen değerin zamana bağlı olarak yavaşça değiştiğidir.

Giriş korelasyon matrisi denilen bir $R = \langle X_k \cdot X_k^t \rangle$ matrisi ve bir $p = \langle d_k \cdot X_k \rangle$ vektörü tanımlanmış ve $\xi = \langle \varepsilon_k^2 \rangle$ olsun. Bu ifadeleri kullanarak Eşitlik (2.4) aşağıdaki şekilde yazabilir.

$$\xi = \langle d_k^2 \rangle + W^t \cdot R W - 2 p^t \cdot W \quad (2.5)$$

Bu eşitlik ξ 'yi ağırlık vektörü W 'nin açık bir fonksiyonu gibi gösterir. Diğer bir değişle $\xi = \xi(W)$ 'dir.

Minimum küçük kare hatasına bağlı ağırlık vektörünü bulmak için:

$$\frac{\partial \xi(W)}{\partial W} = 2RW - 2p \quad (2.6)$$

$$2RW^* - 2p = 0 \Rightarrow RW^* = p \quad (2.7)$$

$$W^* = R^{-1} p \quad (2.8)$$

$$\frac{\partial \xi(W)}{\partial W}$$

ξ 'nin skaler olmasına rağmen $\frac{\partial \xi(W)}{\partial W}$ ise bir vektördür.

Eşitlik (2.6), ξ 'nin gradientidir, yani $\nabla \xi$ bir vektördür.

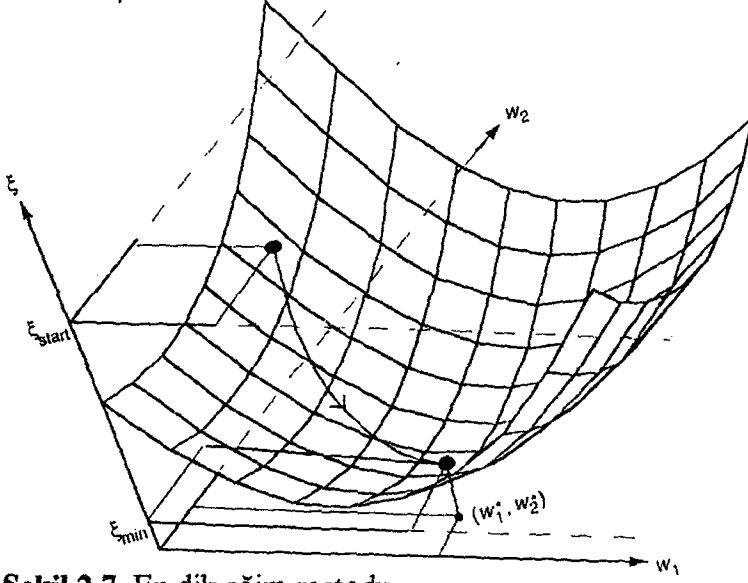
$$\nabla \xi = \left[\frac{\partial \xi}{\partial W_1}, \frac{\partial \xi}{\partial W_2}, \frac{\partial \xi}{\partial W_3}, \dots, \frac{\partial \xi}{\partial W_n} \right]^t \quad (2.9)$$

Bütün bu işlemler, fonksiyon eğimi $\xi(W)$ sıfır olan, bulduğumuz bir noktayı gösterir. Genelde, bu nokta minimum veya maksimum bir nokta olabilir.

2.5.2.1.2. En dik eğim (Steepest Descent) metoduyla ağırlıkların (W^* 'nin) bulunması

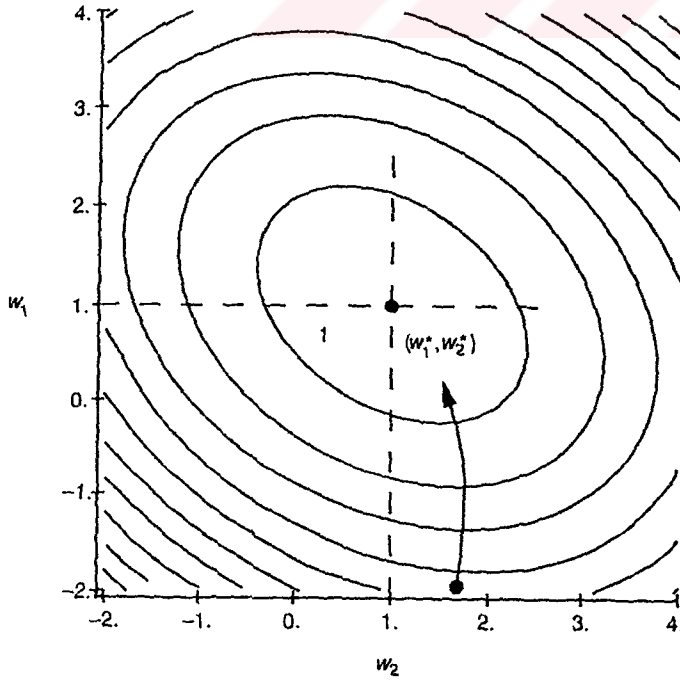
Ağırlık yüzeyinden rastgele seçilmiş bir ağırlık noktası, en dik eğim yönü aşağı doğru olacak şekilde belirlenir. Bu işlem, Şekil 2.7'de gösterildiği gibi minimuma ulaşıncaya kadar tekrarlanır. Şekil 2.7 en dik eğim metodunu göstermektedir. Ağırlık

vektörünün ilk değerinin rastgele seçilmesi ξ_{start} gibi bir hatayı doğurur. Bu metod daima aşağı yönde, çukura ulaşıncaya kadar belli adımlarla hareket etmeyi hedefler.



Şekil 2.7. En dik eğim metodu

Şekil 2.7 'de ağırlık vektörünün ilk hareket yönü direk minimum noktaya doğru olmaz. Parabolik ağırlık yüzeyinin kesiti genellikle eliptik olup, bu yüzden negatif gradient noktası, doğruca minimum noktada olmayabilir. Bu durum Şekil 2.8 'deki ağırlık yüzeyinde daha açık bir şekilde gösterilmiştir.



Şekil 2.8. Şekil 2.7 'deki ağırlık yüzeyinin değişimi

Şekil 2.8; Şekil 2.7 'deki ağırlık yüzeyinin değişimini göstermektedir. En dik eğimin yönü her noktada gösterilen değişimlere diktir ve bu yön daima minimum noktayı göstermez. Bu işlemde ağırlık vektörü değişken olduğundan, ağırlık değişimleri t zaman adımlarına göre gösterilmiştir. İlk ağırlık vektörü $W(0)$ ve t anındaki ağırlık vektörü de $W(t)$ ile gösterilmektedir. Her adımda bir sonraki ağırlık vektörü şu şekilde hesaplanmıştır.

$$W(t+1) = W(t) + \Delta W(t) \quad (2.10)$$

Burada $\Delta W(t)$; t. zaman adımındaki W 'nin değişimidir.

Yüzeydeki her noktanın en dik eğim yönüne bakılmalıdır. Bu yüzden yüzeyin gradientini hesaplayarak en dik yukarı eğim yönü bulunabilir. Sonrada gradientin negatifi alınarak en dik eğimin yönü belirlenmiş olur. Değişimin büyüklüğünü, gradienti μ gibi bir sabitle çarparak aşağıdaki gibi bir ifade ile elde etmek mümkündür.

$$W(t+1) = W(t) - \mu \nabla \xi(W(t)) \quad (2.11)$$

Her bir iterasyonda $\nabla \xi(W(t))$ değerinin belirlenmesi gerekir. Eşitlik (2.6) veya (2.9); $\nabla \xi(W(t))$ 'nin belirlenmesinde kullanılabilir. Fakat W^* 'nin analitik belirlenmesi yapılmalıdır. Bunun için de R ve P değerlerinin bilinmesi gerekir. Bunu gerçekleştirmek çok zor olduğu için, gradientin bulunması için aşağıdaki gibi bir yaklaşım yapılmıştır.

İterasyondaki her adım için, sırasıyla aşağıdaki maddeler gerçekleştirmelidir:

1. Ağ modelinin girişlerine X_k , giriş vektörü uygulanır.
2. $\varepsilon_k^2(t)$, hatanın karesi değeri belirlenir, o andaki ağırlık vektörünün kullanılmasıyla

$$\varepsilon_k^2(t) = (d_k - W^t(t) X_k)^2 \quad (2.12)$$

bulunur.

3. $\varepsilon_k^2(t)$ 'nin $\langle \varepsilon_k^2 \rangle$ kullanılmasıyla bir $\nabla \xi(t)$ yaklaşımı hesaplanır.

$$\nabla \varepsilon_k^2(t) \approx \nabla \langle \varepsilon_k^2 \rangle \quad (2.13)$$

$$\nabla \varepsilon_k^2(t) = -2\varepsilon_k(t) \cdot X_k \quad (2.14)$$

Burada Eşitlik (2.12) kullanılarak hatanın gradienti hesaplanmıştır.

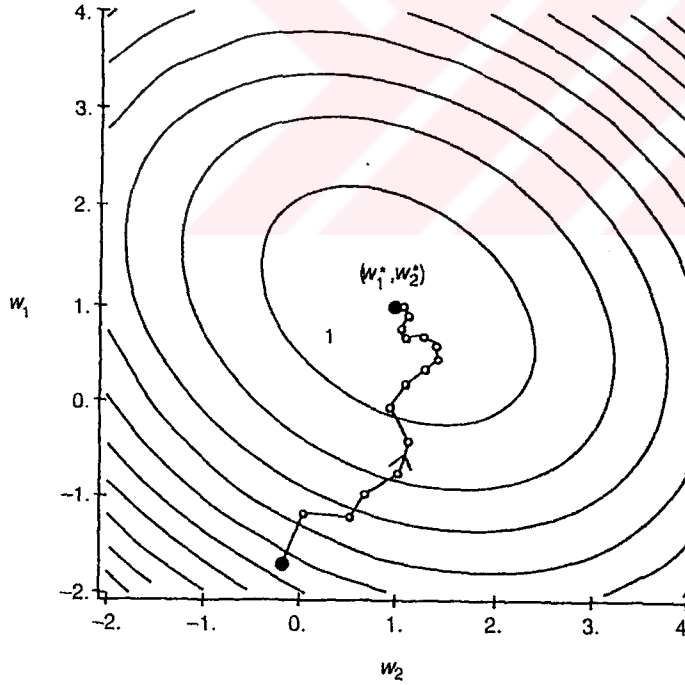
4. Eşitlik (2.14) 'teki gradient formülü yaklaşımını kullanarak Eşitlik (2.11) 'e göre ağırlık vektörü yenilenir.

$$W(t+1)=W(t)+2\mu\varepsilon_k X_k \quad (2.15)$$

5. 1 'den 4 'e kadarki adımlar diğer giriş vektörleri ile, hata kabuledilebilir bir değere indirilene kadar tekrarlanır.

Eşitlik (2.15) K.Y 'ni ifade eder. μ parametresi kararlılık ve ağırlık vektörünün minimum hataya yakınsama hızını belirtmektedir.

Şekil 2.9; Eşitlik (2.15) 'in KY algoritması ile nasıl bir yörünge oluşturacağına örnektir. Her iterasyonda ağırlık vektöründeki değişimler küçük değerli tutulmak zorundadır. Eğer değişimler çok büyük olursa, ağırlık vektörü minimum değeri asla bulamaz veya kazara yakınsayabilir. μ parametresinin fonksiyonu bu kararsızlık durumunu ortadan kaldırır.



Şekil 2.9. Ağırlık vektörünün tahmini yörüngesi

Şekil 2.9'da minimum hata için KY algoritmasının kullanıldığı, ağırlık vektörünün tahmini yörüngesi görülmektedir. Bu yörünge pürüzsüz bir eğri değildir, çünkü

gradient eşitliği her noktada yaklaşık bir değere sahip olacaktır. Minimum hata noktasına yaklaştıkça adım büyüklüklerinin küçüldüğüne dikkat edilmelidir.

2.5.2.2. Hebb öğrenme kuralı

Bir işlem birimi diğer bir işlem biriminden bir giriş alırsa ve eğer ikiside aktif ise (matematiksel olarak aynı işarete sahipse), bu iki işlem birimleri arasındaki ağırlıklar güçlendirilmelidir.

2.5.2.3. Gradient eğimi öğrenme kuralı

Bu kural gerçek çıkışla istenen çıkış arasındaki hatayı minimize etmek için bir matematiksel yaklaşıma sahiptir. Ağırlıklar, hatanın birinci türevine orantılı olarak değiştirilmektedir.

2.5.2.4. Kohonen öğrenme kuralı

Bu kural sadece eğitimcisz öğrenme ağı uygulamalarında kullanılmaktadır. Bu işlemde, bir işlem birimi en iyi öğrenme fırsatını yakalamak için rekabet eder. En büyük çıkışa sahip işlem birimi "kazanan işlem birimi" olarak anılır ve diğer işlem birimlerini yasaklayabildiği gibi uyarabilme özelliğine sahiptir. Sadece kazanan işlem biriminin çıkış vermesine izin verilmiştir ve sadece kazanan birim ile ona komşuluk eden birimlerin ağırlıklarının uyarılmasına izin verilir.

2.5.2.5. Geri yansıtma öğrenme kuralı

Hataların geri yansıtılması tekniği; Delta öğrenme kuralının geliştirilmiş halini kullanır. Bu işlem iki fazda gerçekleşmektedir. İlk faz, "ileri doğru faz" olup, giriş sunulduktan sonra ve ileriye doğru ağı yansıtılarak her birim için geçerli tüm çıkışlar istenen çıkışla karşılaştırılır, farkı veya hata ifadesi hesaplanır.

İkinci faz ise "geriye doğru faz" olup, birinci fazda hesaplanmış olan farkı şimdi de geri yönde işleme koyar. Sadece bu iki faz tamamlandığı zaman yeni girişler ağı işletilmek üzere verilir.

Her bir düğümün kendisinden önce gelen katmana, ne kadar toplam hata verdiğini anlamak için; bir düğümün ileri doğru fazda her girişe göre hatayı ne kadar arttırdığını

anlayarak, geriye doğru bu hatayı yansıtmak koşulu ile, ağırlıkları bu iki faz sonucunda uygun değere ayarlamak mümkündür.

Geri yansıtma kuralı yavaş olup, off-line eğitim gerektirir. İki nokta arasında osilasyona girebilir ve lokal minimada saplanıp kalma eğilimi gösterebilir. Lokal minima olayı; sistem, bir önceki hata değerlerinden daha düşük bir değeri bulduğunda; muhtemel en küçük hataya ulaşma yoluna gitmez. Yani lokal minima dediğimiz noktada kalır.

2.5.2.6. Grossberg öğrenme kuralı

Bu kural Hebbian öğrenme kuralı ile biyolojik unutmamanın birleşimidir. Bu kuralda, her sinir ağı içyıldız (instar) ve dışyıldızlardan (outstar) yapılmıştır. İçyıldız; bir işlem biriminin bir çok girişler almış haline denir. Dışyıldız ise bir işlem biriminin, çıkışlarını diğer bir çok işlem birimlerine göndermesine denir. Eğer toplam giriş veya çıkış küçük ise, ağırlıklardaki değişim çok küçüktür ve ağırlıklar, önemsiz bağlantılar üzerinde sifira yakın olabilecektir.

Zaman, bu öğrenme kuralında önemlidir. Eğer canlandırıcı bir giriş sistemden çıkarılırsa, belli bir zaman sonra, unutmama setlerine daha az çıkış yanıt verecektir. Sınır değeri bu öğrenmede önemli bir faktördür. Eğer sınır değeri çok büyük bir değerde ise, ağ çok küçük değişimlere cevap verir, eğer çok düşük ise, ağ; değişimleri daha çok ihmal eder. Öğrenme, gözleme işinden ayrılmalıdır. Ağ tekrar işleme başladığında, öğrenme işi bitirilmelidir. Öğrenme oranını kontrol edebilmek için, eşitliklerde bir terime ihtiyaç duyulur. "Sürekli konuşma tanıma" ve "robot koluna komutlar verme" gibi muhtemel uygulamalarda bu öğrenme kuralından yararlanılmaktadır.

2.5.2.7. Kuvvet güçlendirme teorisi-KGT (Drive-Reinforcement Theory-DRT)

Bu kuralda, işlem birimleri, belirli giriş tipleriyle koşullandırılırlar: bazı girişler uyarıcı ödüller verirken, bazıları da cezalandıran uyarıcı gibi davranır. İşlem birimleri; bazı girişlere nasıl ulaşacağını ve diğerlerinden nasıl sakınacağını ağırlıklarını ayarlama yoluyla öğrenir. Hebb kuralının tersine KGT 'de, bir girişin gücündeki değişim, giriş ve çıkıştaki değişimlerin çarpımına orantılıdır. Daha açık ifade edilirse şu an için geçerli olan çıkış; şu andaki çıkıştan çok bir önceki çıkışa bağlıdır.

2.6. Yapay Sinir Ağlarının Paralel Mimarideki Yeri

Sinir ağlarının paralel donanıma benzetilmiş uygulamaları, bilgisayar simülasyonlarından daha verimli bir yöntemdir. İşlem birimlerinin paralel yapısı ve birbirleriyle olan bağlantıları, basit bir işlemci ve iletişim linkleri ile; biri, bir diğerine birleştirilerek yapılır. Hiperküp (hypercube) buna bir örnektir.

Hiperküp üç boyutludan fazla kutu yerine, üç boyutludan fazla boşluk içinde düşünülmelidir. NCube firması, bu alanda kendi ürettiği hiperküp süper bilgisayar donanımından yararlanmıştır. 1989 'da, NCube Oracle firmasıyla birleşerek, paralel işlemi yapan bilgisayara önderlik edecek ilk ciddi iş uygulamalarını sunmuştur.

2.6.1. Sinirbilgisayarları (Neurocomputers)

Sinirbilgisayarları yeni bir bilgisayar sınıfı olup, sinir ağlarının çalıştırılması için üretilmiştir. Mimarisi, hiperküp gibi paralel makinalardan oldukça farklıdır. Aslında, bilgi işleme teknikleri, sinir ağlarıyla birleştirilmiş diferansiyel denklem sistemlerini yapmak için tasarlanmıştır. Buna ek olarak, bir sinirbilgisayarının kabiliyeti olan kendi kendini organize ettiği donanımına, uygulamalardaki belirli problemleri çözmeye, paralel mimarilerin yararlı kılınmasında bir anahtar özellik taşır.

Sinirbilgisayarları; elektronik, optik ve optoelektronik olarak sınıflandırılabilir. Optik sinir ağları; uygulamalarda, işlem birimlerinin aktivite seviyelerini, piksellerin parlaklık seviyelerini, LED 'lerin dizilişleri veya diğer sürekli emiciler belirler. Elektronik çipleri arasındaki optik bağlantılar, klasik bilgisayarlarda görülen darboğazları aşmada kolaylık sağlayacaktır. Optik işlemler daha az ısı üretir ve elektronik işlemlerden 1000 kez daha hızlı işlem yapar.

Optoelektronikler sayesinde, sinir ağı entegre devrelerinden daha karmaşık devreler tasarlanabilir. Bir optoelektronik devre, bir inç alanda 250000 bağlantı yoğunluğunu destekleyebilmektedir.

2.6.2. Genel amaçlı sinirbilgisayarları

NEC firması kişisel bir sinirbilgisayarı üretmiştir. Bu bilgisayar, sinir ağı yazılımı ve özel bir sinir ağı bordu içerir. Yazılım, kullanıcıya istediği ağ yapısını tanımlama imkanı sağlar ve tüm hesaplamaları kendisi yapar. Bord, yüksek hızda paralel işlem yapmak için NEC 'in dört tane veri akış mikroişlemcisini kullanır. Maksimum

simülasyon büyüklüğü: 82000 işlem birimi ve 246000 bağlantı ile işlem yapmak mümkündür. Maksimum hızı saniyede 146000 linktir.

Tablo 2.1: İlk Sinirbilgisayarları

Sinir-Bs Adı	Yılı	Teknolojisi	PE sayısı	Bağ. Sayısı	Hız (CUPS)
Perceptron	1957	ElektroMekanik	8	512	10^3
Ada-Madeline	1960/62	Elektronik	1/8	15/128	10^4
Cross Bar	1984	Elekt-Optik	32	10^3	10^5
Mark III	1985	Elektronik	8×10^3	4×10^5	3×10^5
Neu-EmulationProcess	1985	Elektronik	4×10^3	1.6×10^4	4.9×10^5
Optik Rezonator	1985	Optik	6.4×10^3	1.6×10^7	1.6×10^5
MARK IV	1986	Elektronik	2.5×10^5	5×10^6	5×10^6
ODYSSEY	1986	Elektronik	8×10^3	2.5×10^5	2×10^6
CROSSBAR CHIP	1986	Elektronik	256	6.4×10^4	6×10^9
OPTNOVELTYCHIP	1986	Optik	1.6×10^4	2×10^6	2×10^7
ANZA	19876	Elektronik	10^4	5.2×10^6	2.5×10^4 (1.4×10^5)
PARALLON 2	1987	Elektronik	10^4	5.2×10^4	1.5×10^4 (3×10^4)
Delta Floating-Point Processor	1987	Elektronik	10^6	10^6	2×10^6 (10^7)
ANZA PLUS	1988	Elektronik	10^6	1.5×10^6	1.5×10^6 (6×10^6)

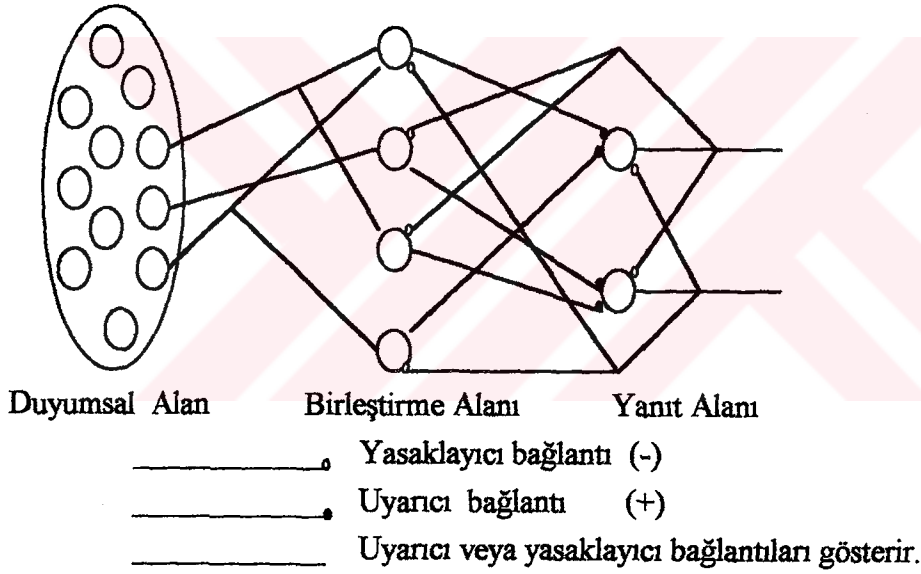
Not: Ağ modellerinin hız kolonundaki CUPS "Connection Updates Per Second" ifadesi; saniyedeki bağlantı yenileme süresini verir. Ayrıca bazı modellerin parantez içinde verilen hızları öğrenme olmaksızın modelin hızını belirtmektedir (Nelson and Illingworth 1991).

2.7. Yapay Sinir Ağı Modelleri

2.7.1. Algılama (Perceptron) Modeli

Rastgele fakat bölgesel bağlantılı üç katmanlı bir ağ modelidir. Giriş katmanı, hissedilen bilgiyi anlayıp toparlayarak onu bir sonraki katmana aktarır. Katman iki, katman üç ile karşılıklı bağlantılı bir ünedir. Çıktılar 0 ya da 1 'dir. Transfer fonksiyonu lineer olduğundan, son araştırmalar üç katmanın tek katmana inebileceğini göstermiştir.

Algılama bir çok türdeki problemi çözebilir. Başlıca uygulamalar; şekil sınıflandırma, karakter tanıma ve robot görüntü sistemlerini kapsar .



Şekil 2.10. Algılama modeli yapısı.

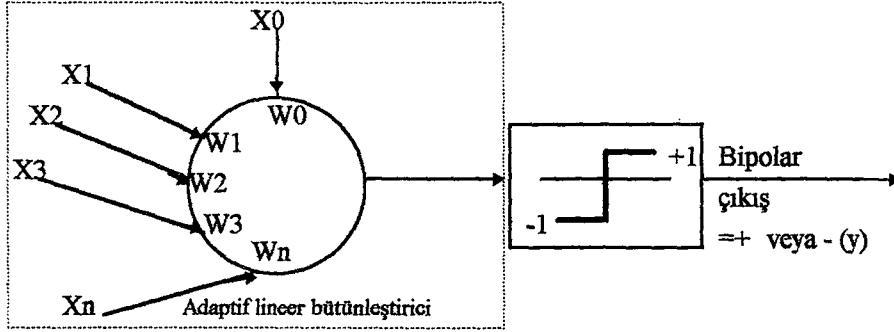
Şekil 2.10 'da gösterildiği gibi; basit bir algılamanın; bir duyumsal, bir birleştirme ve bir de yanıt alanı vardır. Birimler arasındaki çeşitli şekillerle gösterilen bağlantılar, ayrıntılı bir şekilde gösterilmemiştir.

2.7.2. Adaline ve Adaptif Lineer Birleştirici

Adaline, teknik olarak yapay sinir ağı olmayan, tek bir işlem biriminden oluşan bir araçtır. Öncelikle "ADAPtive LInear NEuron (ADALINE)" denmiş, 1960 'larda

yapay sinir ağı popülarlığını kaybedince "ADAPtive LINear Element" şeklinde adlandırılmıştır.

Şekil 2.11'de bir adaline modelinin yapısı görölmektedir. Kesik çizgili kutu içinde adaptif lineer birleştirici içeren tam bir Adaline ve bipolar bir çıkış fonksiyonu görölmektedir. Adaptif lineer birleştirici, genel bir işlem birimine benzemektedir.



Şekil 2.11. Adaline yapısı.

2.7.3. Madaline

Madaline bir çok katmandan oluşan bir donanımdır. Uygun eğitim seti ile, ağı her bir farklı sınıftaki giriş imgesi için çıkış düğümlerinde binari +1 değerini öğretmekle yükümlüdür. Böyle bir sınıflandırmaya örnek olarak dört hayvan cinsinin tanımlanmasını verebiliriz. Böyle bir ağıda, çıkış katmanındaki dört düğümden her biri tek bir cinsi ifade etmektedir. Verilen bir giriş paterni için, bir düğüm eğer giriş paterninin adı geçen düğüm tarafından temsil edilen sınıfa karşılık gelirse, o düğümün çıkışı +1 olacaktır. Geriye kalan dört düğümün çıkışları da -1 olacaktır. Eğer giriş paterni bilinen sınıflardan hiç birinin özelliklerini taşııyorsa, ağıın sonuçları belirsiz olabilir.

Şekil 2.12'de göröldüğü gibi bir çok Adaline, katmanlaştırılmış sinir ağı içinde bir araya getirilebilir.

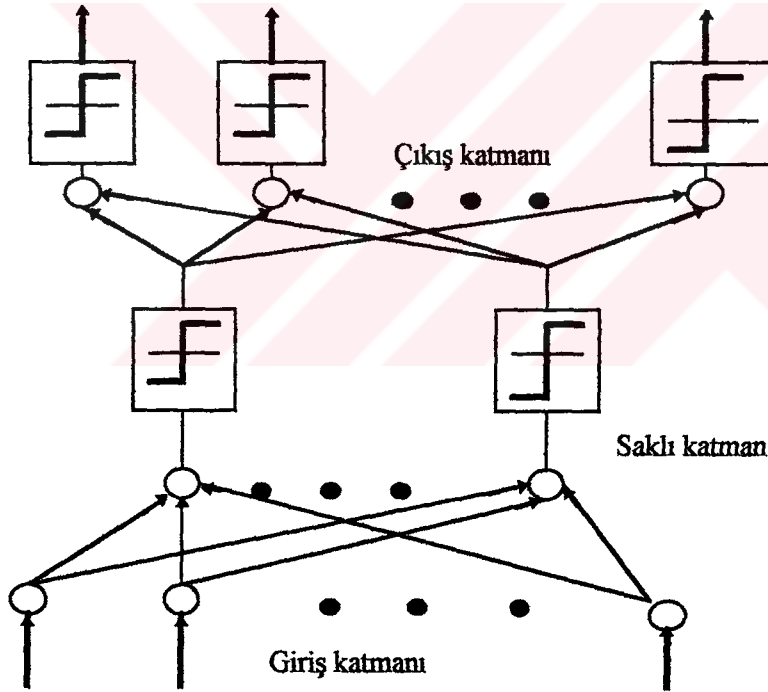
2.7.4. Bir kutu içindeki zeka durumu-KZD (Brain-State in a Box "BSB")

KZD hatayı düzelten bir geribeslemeyi kullanır. Bu otomatik birleştirici ağındaki birimler, birbirleriyle bütün olarak bağlanmamışlardır. İçinde parçalı paternlerin bulunduğu girişler vektör formundadır. Örneğin, her bir karakterin 8 bitle gösterildiği

25 ASCII karakter vektörü için giriş olarak 200 bitlik (25 X 8) vektör gerekmektedir. Daha sonra ağ, paterni tamamlamaya çalışacaktır.

KZD beyin çalışmasını ve algulamasını modeller ve otomatik olarak ilişkilendirilen hata düzeltici geribesleme sonucu hiperküpün bir köşesine yakınsama garantisi vardır. Çıkış, rampa transfer fonksiyonu veya lineer fonksiyon ile sınırları belirlenmiştir. Çıkış limit seviyesine iliştilmiştir. Bu kutunun bir köşesine yakınsamayı ve hatanın bir eğitim setine bağlı olarak minimize edilmesini garanti eder.

Uygulamalar üretilmeden daha çok tecrübe kazandırmaktadır. Örneğin, bu ağ veritabanından bilgi seçmede kullanılabilir. Bu ağ bir kuş sınıflandırma sistemiyle gösterilmiştir. Tıp teşhislerinin konulduğu sistemler de vardır. Ağ, hastalıklar ve belirtiler üzerinde eğitildikten sonra, bilinmeyen bir hastalık için belirlenen belirtiler ağa verince, ağ muhtemel tedaviyi kullanıcıya önerebilir.



Şekil 2.12. Madaline yapısı.

2.7.5. Hopfield Modeli

Yapay sinir ağlarında kullanılan modellerden biri de Hopfield modeli veya daha genel haliyle İki Yönlü Birleştirilmiş Hafıza-İYBH (Bidirectional Associative Memory - BAM) 'dır. Hopfield modeli sürekli ve ayrık çıkışlı problemler için de kullanılabilir.

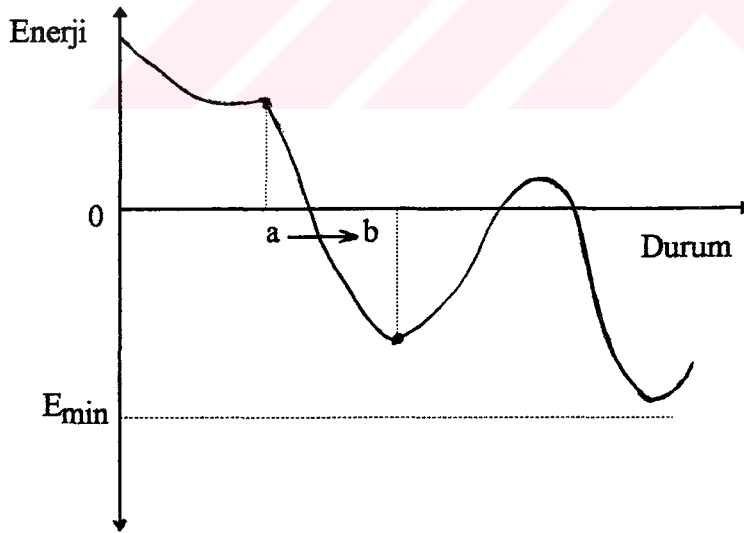
2.7.5.1. Ayrık Hopfield Modeli

Modelin temelinde bir enerji veya maliyet fonksiyonunun minimuma indirgenmesi yatar. Modelde tek katman vardır, girişteki işlem birimlerine uygulanan bir vektör zaman içerisinde bir enerji fonksiyonuna göre minimuma indirgenir. Ayrık modelin enerjisi,

$$E = -\frac{1}{2} \sum_i \sum_j v_i W_{ij} v_j - \sum_i I_i v_i \quad (2.16)$$

şeklinde olup, W_{ij} j. giriş işlem birimi ile i. çıkış işlem birimi arasındaki ağırlıktır. I_i ise, çıkıştaki i. işlem birimine ait eşik değeridir.

$V_i = g(v_i)$ ve $v_i = \sum W_{ij} v_j - I_i$ olup, $g(v)$; $v > 0$ için 1, $v = 0$ için 0 ve $v < 0$ için -1 şeklindedir. Eğer W_{ii} , yani geribesleme var ise, enerji fonksiyonu zamana göre azalan bir fonksiyon olmayabilir. Bu ise sistemin bir minimuma ulaşmasını engelleyebileceğinden, her bir işlem biriminin çıkışı değiştirilirken enerjinin azalıp azalmadığı kontrol edilmelidir. Eğer enerji azalıyor ise çıkış değiştirilmeli, azalmıyorsa çıkış değiştirilmemelidir.



Şekil 2.13. İki boyutta İYBH enerji değişimi.

Şekil 2.13. iki boyutta İYBH enerji değişimini göstermektedir. İşlem süresince, İYBH 'nin çıkışı a durumundan b durumuna geçerken, İYBH enerji değeri; başlangıç noktasından aşağıya doğru hareket ederek, minimuma yakın bir yere gelecektir. Ulaşılan bu noktanın global nokta olmasının gerekmediğine dikkat edilmelidir.

Ayrık Hopfield modelinde görüldüğü gibi öğrenme yoktur. Bir optimizasyon problemi bu modelle çözülmek istenirse, yapılması gereken şey, ilk önce bir hata, maliyet veya enerji fonksiyonu tanımlamaktır. Bu fonksiyonun minimuma indirgenmesi ile ilgilenilen problem de çözüme kavuşacaktır. Dolayısı ile, bu fonksiyon Hopfield modelinin enerji fonksiyonu olarak kabul edilip, W_{ij} katsayıları bulunur. Model çalışmaya hazırdır: girişe uygulanacak bir vektör bu enerji fonksiyonuna göre işlenir.

Girişe uygulanan vektör çıkışa yansıtılır. Giriş birimlerinin çıkış değerleri, tekrar giriş olarak alınır ve tekrar çıkışa yansıtılır. Her yansıtmanın sonunda enerji fonksiyonu kontrol edilir. Eğer enerjide değişim olmuyorsa, işlem durdurulur ve işlem birimlerinin çıkışı çözümü verir.

Ayrık giriş ve çıkışlar Hamming kökünün köşelerindeki değerlerden bulunabilir. Model; global minimuma ulaşma garantisi vermez, en çok başlangıç noktasına en yakın yerel minimuma çözümü götürür (Freeman and Skapura 1991).

2.7.5.2. Sürekli Hopfield Modeli

Ayrık Hopfield modelinde giriş ve çıkışlar Hamming kökünün köşesinde yer almaktadır. Dolayısı ile, sürekli giriş ve çıkışlar için kullanılamaz. Bu durumlar için sürekli Hopfield modeli kullanılabilir.

Sürekli Hopfield modeli içinde temelde aynı kural geçerlidir: "bir enerji fonksiyonunun minimuma indirgenmesi". Gerçek sınır hücrelerinde, bir girişin çıkışının başka bir sinirin girişine ulaşması arasında bir zaman farkı olacaktır. Bunun için kondansatörler bu modelde kullanılmalıdır.

Şekil 2.14 'de sürekli hopfield hafızası için, işlem birimi olarak kullanılan sigmoid çıkış karakteristikli yükselteçler görülmektedir. hatların kesişim noktalarındaki siyah noktalar, işlem birimleri arasındaki bağlantıyı göstermektedir. Her bağlantıda değeri $R_{ij} = 1/|T_{ij}|$ olan direnç yerleştirilmiştir. Burada T_{ij} ; ağırlık matrisini göstermektedir. Bütün dirençlerin değeri pozitif olduğuna göre, evirici yükselteçler yasaklayıcı sinyalleri temsil etmektedir. Bu yüzden, bir işlem birimi iki yükselteçten oluşmaktadır. Eğer her hangi bir birimin çıkışı, diğer birimlerden birisini etkin kılarsa, evirici olmayan yükselteçten gelen sinyal ile bağlantı kurulur, etkin kılmazsa yani yasaklayıcı bir sinyal uygulanmış ise, bağlantı evirici yükselteç üzerinden yapılır.

Her bir yükseltecin bir giriş direnci ve kondansatörü vardır. j. işlem biriminden i. işlem birimine giriş

$$\frac{v_j - u_i}{R_{ij}} = (v_j - u_i)T_{ij} \quad (2.17)$$

şeklinde olup, toplam da

$$I_{Ti} = \sum_j (v_j - u_i)T_{ij} - \frac{u_i}{\rho} + I_i \quad (2.18)$$

şeklini almaktadır. Burada $\frac{u_i}{\rho}$ kayıp akımı göstermektedir.

Dolayısıyla,

$$I_{Ti} = c \frac{\partial u_i}{\partial t} = \sum_j (v_j - u_i)T_{ij} - \frac{u_i}{\rho} + I_i \quad (2.19)$$

şeklindedir. Her işlem birimi için bu denklemler çözümlerse; $t \rightarrow \infty$ bulunan çözümler sistemin enerjisini minimuma indirmediği değerlerdir.

Burada u_i ile v_i arasında $V_i = g(\lambda u_i) = 0.5 (1 + \tan(\lambda u_i))$ ilişkisi vardır. Sistemin toplam enerjisi ise

$$E = -\frac{1}{2} \sum_i \sum_{j \neq i} T_{ij} v_i v_j + \frac{1}{\lambda} \sum_i \int_0^{v_i} g_i^{-1}(v) dv \sum_i I_i v_i \quad (2.20)$$

şeklinde olup $\lambda \rightarrow \infty$ durumunda ayrık Hopfield modeline yakınsamaktadır.

Zaman geçtikçe enerjinin azaldığını, yani enerji fonksiyonun zamana göre azalan bir fonksiyon olduğunu göstermek için $\frac{dE}{dt}$ 'nin daima negatif olduğunu göstermek gerekir.

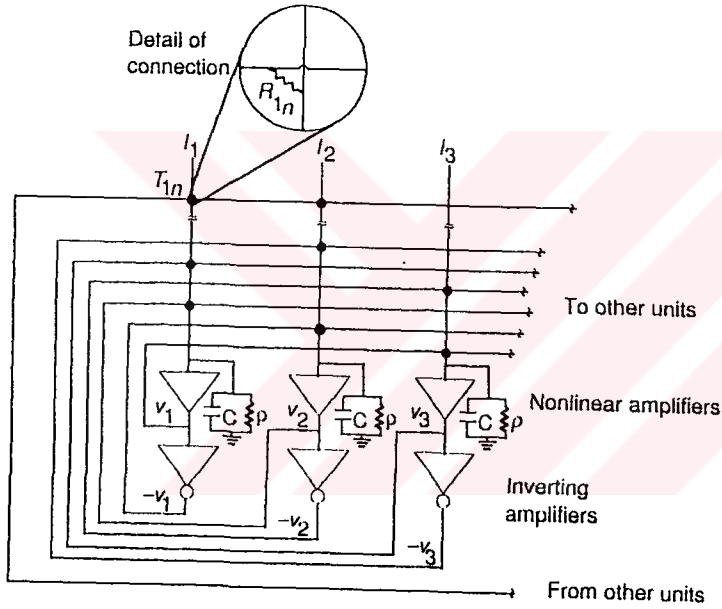
$$\frac{\partial E}{\partial t} = -\sum_i \frac{\partial v_i}{\partial t} \left(\sum_j T_{ij} v_j - \frac{u_i}{R_i} + I_i \right) = -\sum_i c \frac{\partial v_i}{\partial t} \frac{\partial u_i}{\partial t} \quad (2.21)$$

şeklinde olup, $u_i = g^{-1}(v_i)$ ve $\frac{du_i}{dt} = \frac{dg^{-1}(v_i)}{dt} = \frac{dg^{-1}(v_i)}{dv_i} \frac{dv_i}{dt}$ olduğundan dolayısıyla,

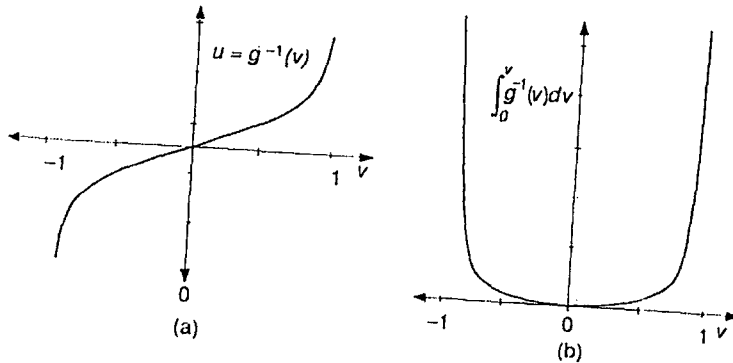
$$\frac{\partial E}{\partial t} = -\sum c \left(\frac{\partial v_i}{\partial t} \right)^2 \frac{\partial g^{-1}(v_i)}{\partial v_i} \quad (2.22)$$

$\frac{dE}{dt} < 0$ daima geçerli olup c 'nin değeri de daima pozitiftir. Ayrıca Şekil 2.15'e göre

çıkış fonksiyonu daima artan bir fonksiyon olduğundan $\frac{\partial g^{-1}(v)}{\partial v} > 0$ şeklinde olacaktır.



Şekil 2.14. Sürekli Hopfield modelinin hafıza devresi.



Şekil 2.15. (a) grafiği, lineer olmayan çıkış fonksiyonunun tersini göstermektedir. (b) grafiği, integrali alınmış lineer olmayan çıkış fonksiyonunu göstermektedir.

2.7.6. Boltzman Makinesi Modeli

Bu ađda, ayrıık Hopfield modelindeki bir iřlem biriminin transfer fonksiyonunun deęiřtirilmesi sonucu benzetilmiř tavlama (Simulated Annealing) iřleminin kullanımı sz konusudur.

Boltzman makinesindeki bir iřlem biriminin transfer fonksiyonu řu řekildedir

$$X_i^{\text{yeni}} = \begin{cases} -x_i^{\text{eski}} & \text{Eęer } \Delta H < 0 \\ -x_i^{\text{eski}} & \text{Eęer } \Delta H \geq 0 \text{ ve } \xi < \exp\left(\frac{-\Delta H}{T}\right) \\ x_i^{\text{eski}} & \text{dięer durumlarda} \end{cases} \quad (2.23)$$

Burada;

$$\Delta H = 4 x_i^{\text{eski}} \left[\sum_{j=1}^n W_{ij} x_j^{\text{eski}} - T_i \right] \quad (2.24)$$

řekindedir. T sıcaklık, ξ ise 0 ile 1 arasında dzensiz olasılık yoęunluęu fonksiyonunca seilen bir sabittir.

Hopfield modelinde olduęu gibi, bir zaman aralıęında sadece bir iřlem birimi yenilenir. Bu iřlem birimi o katmandaki birimlerden rastgele seilir. Yenilenen birimin durumu Eřitlik (2.23) 'deki transfer fonksiyonuna gre belirlenir. Daha sonra yeni bir birim seilerek yenileme iřlemine devam edilir.

2.7.6.1. Benzetilmiř Tavlama (Simulated Annealing)

Benzetilmiř Tavlama ynteminin temel ierięi madencilik tavlamasına benzeřiminden ıkarılmıřtır. Madencilik tavlamasında, bir metal erime noktasına kadar ısıtılır ve daha sonra yavařca oda sıcaklıęına kadar soęutulur (aslında metalin hızlıca soęumasını nlemek iin, metale su vererek yavař yavař soęutulması saęlanır). Tavlama fikri; yksek sıcaklıkta ısıl alkalanma ile yerdeęiřtirmeler ve dięer kristal yapı bozulmaları elenebilir. Yeni herhangi yerdeęiřtirme metali yavařca soęutarak sıcaklıęın dřmesiyle engellenebilir. Bu iřlemin sonunda metalin global enerji fonksiyonu bir minimum deęere ulařacaktır.

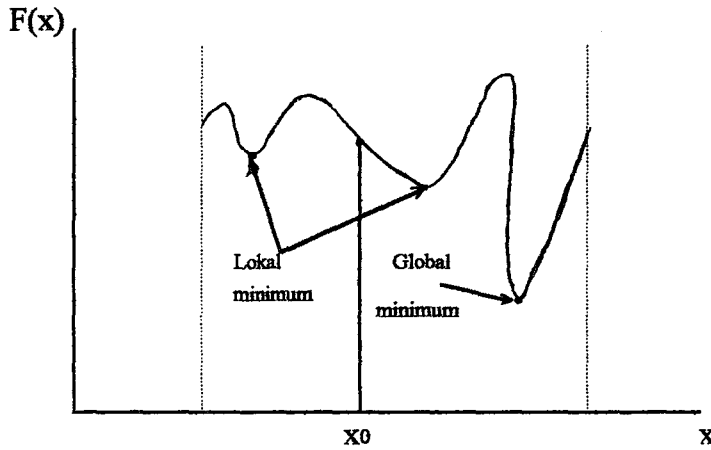
Tavlama hakkındaki bu şaşırtıcı düşünce, ısı çalkalanma istatistik işleminin yaklaşık aynı düşük enerji durumuna getirilmesine önderlik eder. Benzetilmiş tavlamanın matematiksel içeriği bu fiziksel davranışlara benzetimden türetilebilir.

Bu metodun matematiksel temel fikri, Nicholas Metropolis, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller ve Edward Teller tarafından 1953 'te bulunmuştur. Bu çalışmadaki kesin gözlem; mekaniksel moleküler etkileşimler yolu ile istatistik enerji transferi yüzünden bu işlem fiziksel sistemlerde oluşur (Hecht-Nielsen 1990). Sistemin enerjisinin global minimuma getirilmesi işlemi basit enerji değişiklik işlemi şeklinde özetlenebilir. Temel fikir, belli bir zamanda rastgele seçilmiş sistemin bir parçasının geçici olarak değişmesidir. Bu değişim sonuçları, global sistem enerjisinden düşük bir sıcaklıkta ise bu değişim kabul edilebilir ve kalmasına izin verilir. Eğer değişim sonuçları global sistem enerjisini aşıyorsa bu değişim aşağıda verilen P olasılığı ile kabul edilebilir.

$$P = \exp(-\Delta E / T) \quad (2.25)$$

Burada $\Delta E = E_{\text{yeni}} - E_{\text{eski}}$ enerjisindeki değişimdir. T ise sistem sıcaklığıdır.

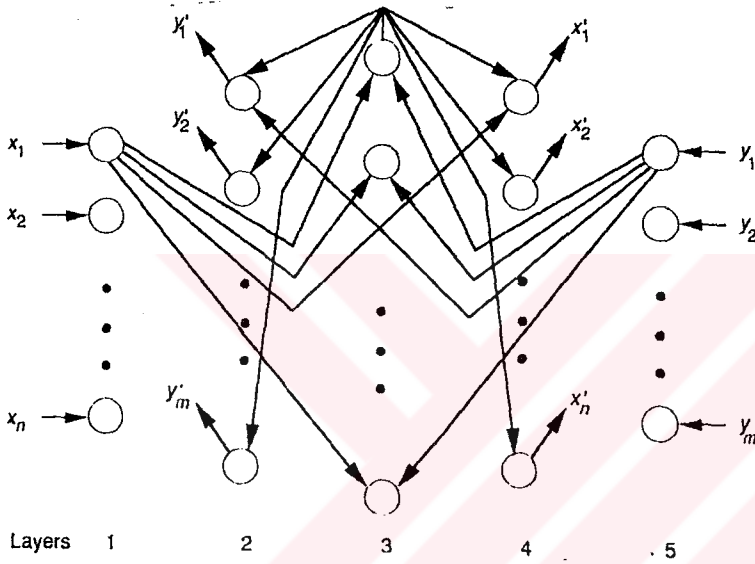
Matematiksel optimizasyondaki klasik problem bir $f(x)$ fonksiyonunun global minimum noktasının bulunmasıdır. Şekil 2.16 'da gösterildiği gibi bu problem gradient eğimi metoduyla çözülemez. Eğer gradient eğim metodu kullanılırsa, çözüm; fonksiyonun global değilde lokal minimum noktası olacaktır (Hecht-Nielsen 1990).



Şekil 2.16. Bir fonksiyonun global minimum noktasının bulunması.

2.7.7. Tersyansıtma Ağ Modeli-TAM(Counterpropagation Network Model-CPN)

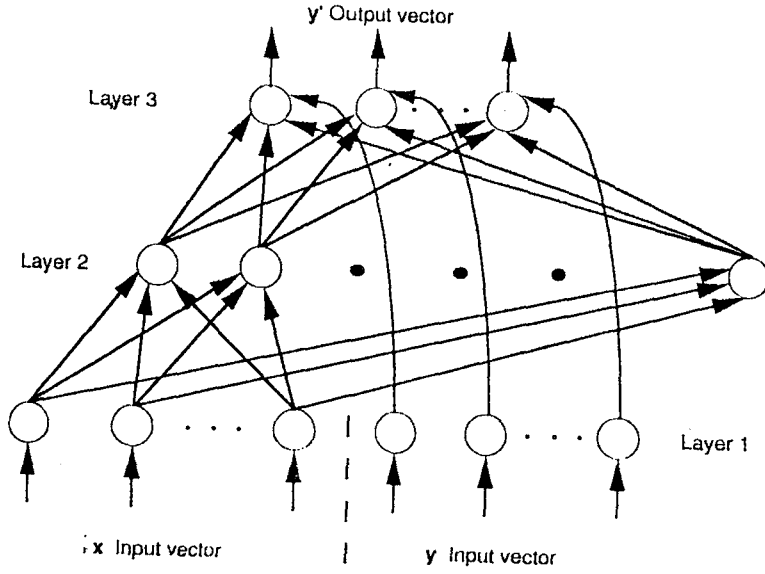
$(x_1, y_1), (x_2, y_2), \dots, (x_L, y_L)$ gibi verilmiş vektör çiftleri setini, TAM; giriş katmanında bir x vektörü ile çıkış katmanında bir y vektörü şeklinde öğrenilebilir. Eğer x ve y arasındaki ilişki ϕ gibi, sürekli bir fonksiyon ise TAM eğitim vektörünün belirlediği her hangi x değeri için $y = \phi(x)$ yaklaşımı ile öğrenecektir. Eğer ϕ 'nin tersi sunulmuşsa, x, y 'nin bir fonksiyonu olacak ve TAM; $x = \phi^{-1}(y)$ yaklaşımı ile öğrenecektir.



Şekil 2.17. TAM donanımı.

Şekil 2.17 'deki TAM donanımının 5 katmanı vardır. İki giriş katmanı (1 ve 5), bir saklı katman (3) ve iki çıkış katmanı (2 ve 5). TAM 1.ve 2. katmandaki giriş vektörlerinin ağa doğru ters yönde yansıtılmasından dolayı bu şekilde isimlendirilmiştir. Şekil 2.18 'de, TAM diagramının farklı biçimde organize edilmiş şekli görülmektedir. Ağ bu görünümü ile üç katmanlı donanım gibidir. 2. katmanındaki her birimin net-giriş değerini hesaplar ve hangi birimin en büyük net-giriş değerine sahip olduğunu belirler. Sadece en büyük net-giriş değerine sahip birim, çıkış katmanındaki birimlere bir değer gönderir. Buna göre bütün ağda bir tek öğrenme algoritması yerine, TAM her katmanda farklı öğrenme işlemlerini kullanır.

Şekil 2.18 'de ileriye taşıma TAM gösterilmektedir. Vektör çiftleri eğitim setinden 1. katmana uygulanmaktadır. Eğitim tamamlandıktan sonra, (x, ϕ) vektörlerinin 1.katmana uygulanması çıkış katmanı olan 3. katmanda y vektörüne karşılık gelen bir y' yaklaşımını doğuracaktır.



Şekil 2.18. Farklı biçimde organize edilmiş TAM ağı.

2.7.8. Kendi Kendini Organize Eden Ağ Modeli (Self-Organizing Maps-SOM)

Öğrenme işlemi süresince, pozitif geribesleme; merkezi birimden, merkezi birimin etrafındaki bir sonlu yakınlık içindeki diğer birimlere uzanacaktır. TAM 'ın tek bir çıkış veren katmanındaki birimin öğrenilmesine izin verilirken, SOM 'da öğrenme işlemi ise, çıkış veren birim etrafındaki tüm birimler, bu birimden pozitif geribesleme alır. Eğer bir komşu birimin ağırlığı giriş vektörüne ortogonal ise, bu birimin ağırlık vektörü, giriş vektörüne göre değişmeye devam edecektir.

İşlem biriminin etkinleşmesi aşağıdaki gibidir.

$$y_i = -r_1(y_i) + net_i + \sum_j z_{ij} y_j \quad (2.26)$$

$r_1(y_i)$ genel kayıp ifadesidir. $r_1(y_i) = A \cdot y_i$ şeklinde ifade edilebilir. Burada A bir sabit şeklindedir. Fakat $r_1(y_i)$; y_i fonksiyonunun daha karmaşık hali olabilir. net_i ifadesi birim i 'ye gelen giriş vektörlerinin skaler çarpımını ifade eden net giriştir.

SOM 'da öğrenme işlemi aşağıdaki eşitlikte verilmiştir.

$$W_i = \alpha(t) (x - W_i) u(y_i) \quad (2.27)$$

W_i ; i. birimin ağırlık vektörü ve x; giriş vektörüdür. $y_i > 0$ durumlarda; $u(y_i) = 1$ 'dir. $y_i > 0$ olmadıkça $u(y_i)$ sıfırdır. Sadece $u(y_i) = 1$ durumundaki birimler pozitif aktivite ile

öğrenme işleminde yer alır. $\alpha(t)$ faktörü zamana bağlı bir fonksiyon olup, öğrenmenin gelişmesi, ilerlemesi sonucu programcının isteği doğrultusunda değişmesi beklenir.

Şekil 2.19'da her işlem biriminin koordinatları (u,v) gibi iki boyutlu olarak gösterilmiştir. Bu nedenle ağırlık vektörleri de iki boyutlu olup başlangıç değerleri rastgele seçilmiştir.

Diğer yapılarda olduğu gibi, her giriş vektörü için tek bir çıkış veren işlem birimi; giriş ve ağırlık vektörü arasındaki benzerliğe dayanarak belirlenir. Bir x giriş vektörü için, tek bir çıkış veren birim;

$$\|x - W_c\| = \min_i \{\|x - W_i\|\} \quad (2.28)$$

eşitliğinden belirlenmektedir. Burada i altındisi, tek bir çıkış veren birimi göstermektedir.

Sadece tek bir çıkış veren birimin ağırlığını yenilemek yerine, bu birim etrafında bir fiziksel çevre ve komşuluk tanımlanabilir ve bu komşuluk içindeki tüm birimler ağırlık yenileme işleminde yer alır. Öğrenme ilerledikçe, komşuluk büyüklüğü bir tek birim etrafını saracak şekilde azalır. Eğer C tek çıkış veren birim ise ve N_c de komşuluk yapan birimlerin kümesi ise, ağırlık yenileme eşitliği:

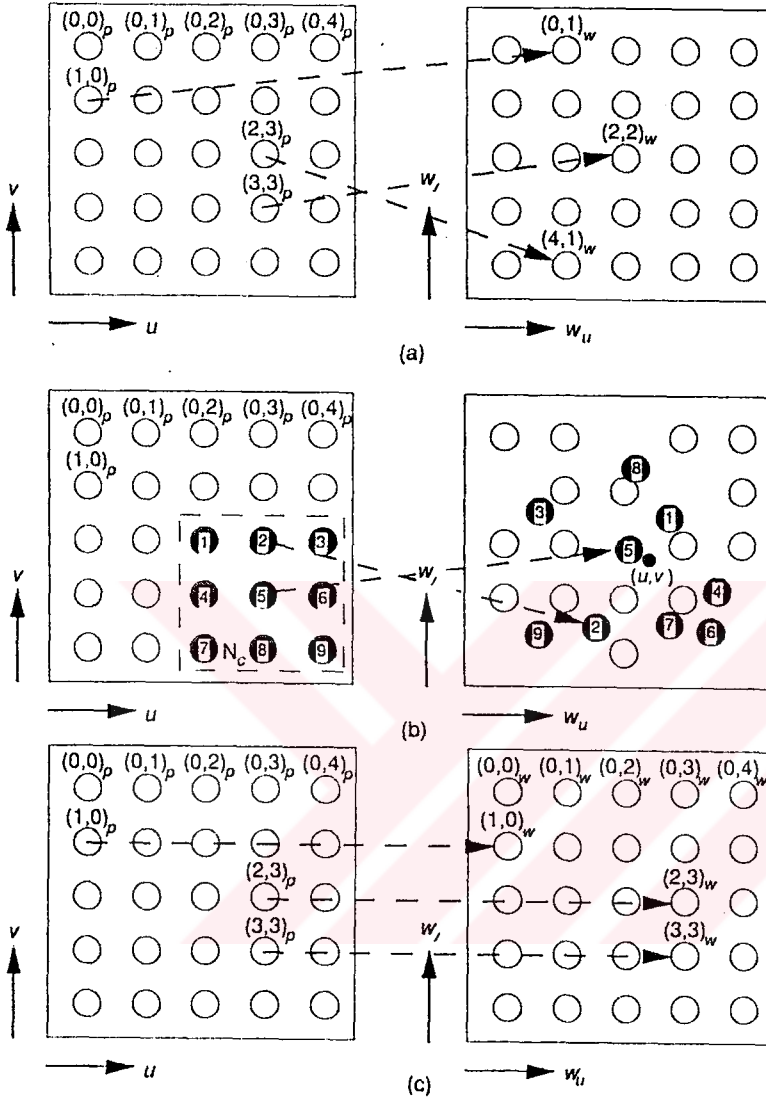
$$W_i(t+1) = \begin{cases} W_i(t) + \alpha(t) (x - W_i(t)) & i \in N_c \\ 0 & \text{Diğer durumlarda} \end{cases} \quad (2.29)$$

Her ağırlık vektörü, x giriş vektörüne doğru yavaşça yer değiştirerek yenileme işlemine katılır. Bir defada eğitim yeterli bir şekilde ilerlemişse, her birimin ağırlık vektörü, birimin fiziksel lokasyonuna yakın noktaların koordinatlarına yakınsıyacaktır.

Kohonen öğrenme işleminin dinamiklerini gösteren bir yol geliştirilmiştir. İşlem birimlerinin pozisyonlarının fiziksel lokasyonlarına göre çizilmesi yerine, lokasyonlar ağırlık koordinatlarına göre çizilebilir. Şekil 2.20 bu düşünceyi açıklamaktadır (Freeman ve Skapura 1992).

Şekil 2.20 'de her birim iki koordinat seti ile gösterilmektedir: sabit $(u,v)_p$ fiziksel-uzay koordinatları ve değişken $(W_u, W_v)_w$ ağırlık koordinatları. Çizim, ağırlık

koordinatlarındaki duruma göre çizgilerle birbirine eklenmiştir. Komşuluk birimleri $(0,0)_p$, $(0,1)_p$ ve $(1,0)_p$ şeklinde gösterildiği gibi kalın çizgilerle bağlanabilir.

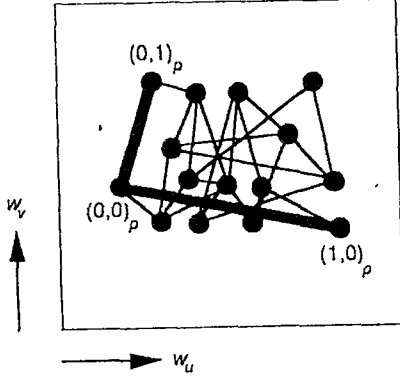


Şekil 2.19. Bu şekiller SOM ağında eğitimin nasıl yapıldığını anlatmaktadır.

(a) SOM 'daki işlem birimleri iki boyutlu dizi içinde sürekli, süreksiz olarak da $(u,v)_p$ noktalarında dizilmişlerdir. Bir iki boyutlu $(W_u, W_v)_m$ ağırlık vektörü her bir birime ayrılmıştır.

(b) Eğitim süresince, (u,v) noktası giriş vektörü olarak rastgele seçilir. Tek bir çıkış veren birim, seçilmiş nokta ile bu birimin ağırlık vektörü arasındaki öklit uzaklığı ölçülerek belirlenir ve bir N_c komşuluğu; bu birim etrafında belirlenir. Komşuluk içindeki bütün birimlerdeki ağırlık vektörleri (u,v) giriş değerine doğru yavaşça değişir. Eğitim, farklı giriş noktaları ile devam ettiği gibi, komşuluk büyüklüğü bir tek birimin etrafını çevreleyene kadar devam eder.

(c) Eğitim sonunda, her birim için ağırlık vektörü yaklaşık olarak o birimin koordinatlarına eşit olacaktır.



Şekil 2.20. Ağrılık koordinatlarına göre lokasyonların çizilmesi.

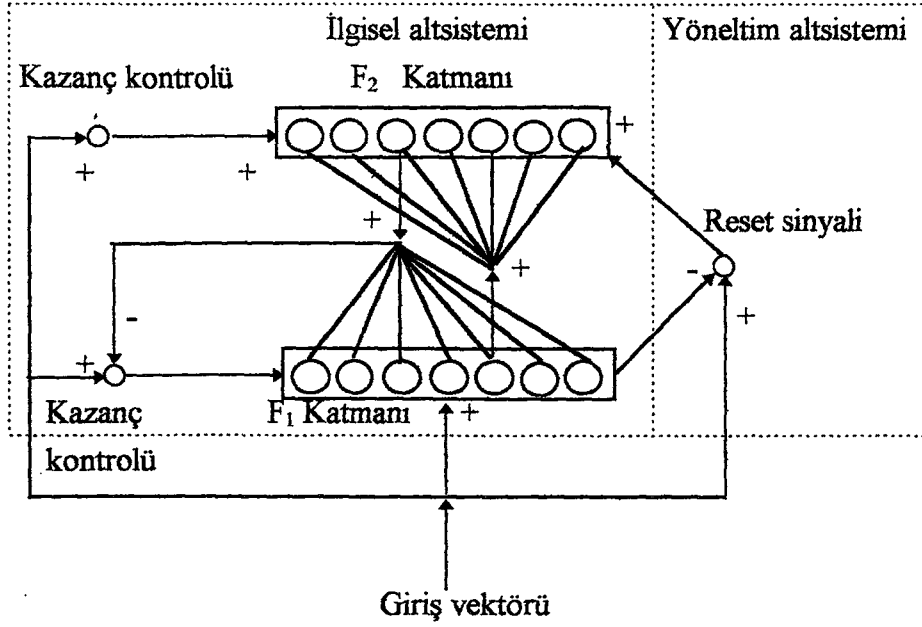
2.7.9. Adaptif Rezonans Teorisi-ART

ART, ismini ağı içinde öğrenme ve tekrar çağırma özelliğinden dolayı almaktadır. Fizikte, rezonans; uygun frekansın küçük genlikteki titreşimi, mekanik veya elektriksel sistem içinde büyük genlikte titreşimleri doğurmaktadır.

Rezonans durumuna iki yolla erişilebilir. Eğer ağ, bir giriş vektörünü öğreniyorsa, bir rezonans durumu meydana getirilir. Rezonans süresince, yenileme işlemi saklı paternin hafızasını güçlendirecektir. Eğer giriş vektörü tanıtılmışsa, ağ bu vektörün saklı paternleri arasında bir benzerlik bulmak için araştırma yapacaktır. Eğer benzerlik bulunamazsa, ağ rezonans durumuna girecektir, bunun üzerine yeni patern ilk kez saklanacaktır. Böylece, ağ önceki öğrendiği verilere karşılık verecektir, yeni veri uygulandığında öğrenmeye hazır durumdadır.

ART donanımının temel nitelikleri Şekil 2.21 'de gösterilmektedir. İki katmandaki ilgisel altsistemi belirten paternler aktivitesine Kısa Dönem Hafıza Yolları-KDHY (Short-Term Memory-STM) denir. Çünkü, sadece bir giriş vektörünün bir tek uygulamasının birleştirilmesinde kullanılır. F_1 ve F_2 arasındaki dip-yukarı (bottom-up) ve tepe-aşağı (top-down) bağlantıları ile birleştirilen ağırlıklara Uzun Dönem Hafıza Yolları-UDHY (Long-Term Memory-LTM) denir. Çünkü, uzatılmış bir period için ağın geriye kalan parçasındaki bilginin işlenmesinde kullanılır.

Şekil 2.21 'de bir ART sistemi görülmektedir. İki önemli altsistem, ilgisel (attentional) altsistem ve yöneltim (orienting) altsistemidir. F_1 ve F_2 ilgisel altsistem içindeki iki katmanın düğümlerini göstermektedir. Her bir katmandaki düğümler diğer katmandaki düğümlerin tümü ile bağlanmışlardır. Parçalar arasındaki diğer bağlantılar ok işareti ile gösterilmiştir. Artı (+) işareti uyarma (excitatory) bağlantısını; eksi (-) işareti ise yasaklayıcı (inhibitory) bağlantıyı göstermektedir.



Şekil 2.21. ART sistemi.

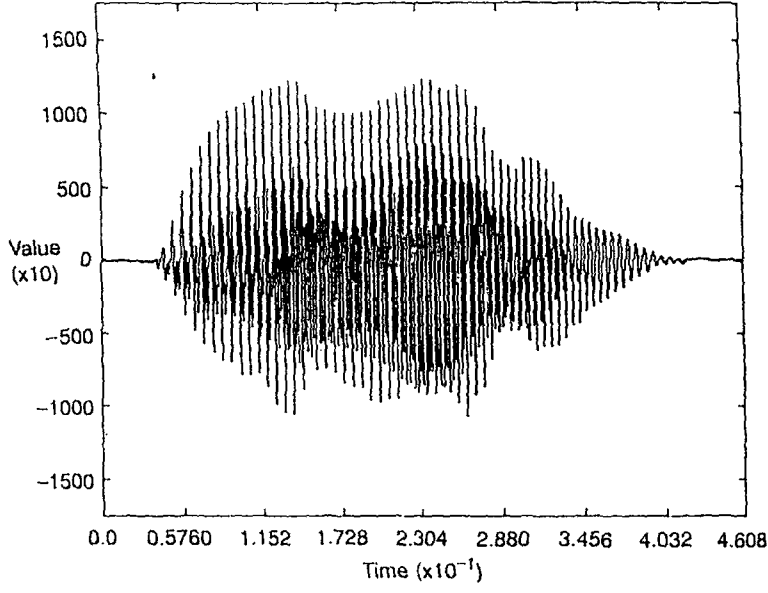
2.7.10. Uzay ve Zaman Koordinatlarıyla Patern Sınıflandırılması-UZKPS (Spatiotemporal Pattern Classification-STP)

Giriş sinyallerinin uzaysal ve zamana bağlı yönleri ile doğrudan doğruya ilgilenebilen YSA donanımları açıklanmaktadır. Bu ağlar, uzaysal patern bilgisine ek olarak uzaysal paternlerin zaman korelasyonu ile ilgili bilgiyi de kodlar.

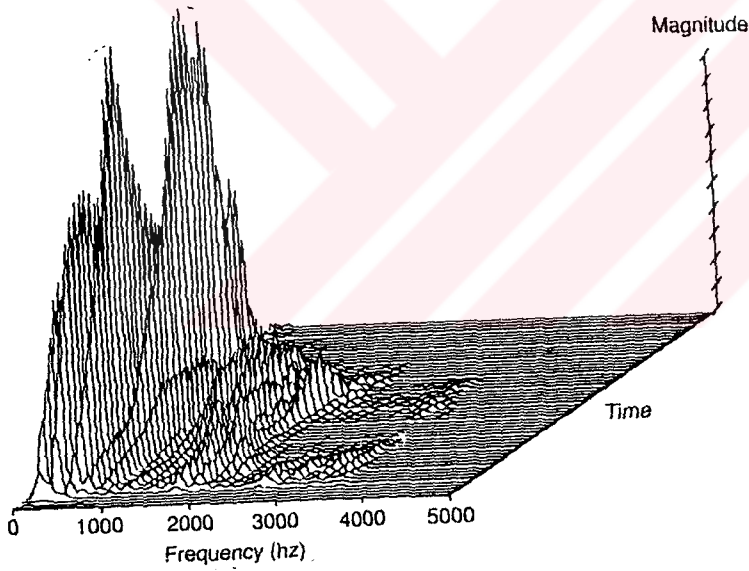
STP 'nin; patern tanımda önemli olan bir kaç uygulama alanları vardır. Bunlardan bir tanesi de STP 'nin çok kanallı ses spektrum analizörü tarafından üretilmiş, zamana bağlı güç spektrumunun olabildiği konuşma tanımadır.

Şekil 2.22 ve 2.23; konuşulmuş bir kelimededen alınmış bir STP örneğini göstermektedir. Şekil 2.22 'de gösterilen dalga şekli, Şekil 2.23 'de bir güç spektra dizisine dönüştürülmüştür. Her bir spektrum (tayf) bir uzaysal paterndir. Her bir zaman dilimindeki uzaysal paternler dizisinin bir tanesi STP 'yi oluşturmaktadır.

Şekil 2.22; çevre koşullarından arındırılmış telafuz yapan bir erkek spikerin işitsel (akustik) dalga şeklini göstermektedir. Örnekleme oranı 10 KHz 'dir.



Şekil 2.22. Sesin dalga şekli.



Şekil 2.23. Şekil 2.22'deki dalga şeklinden yapılmış spektrogram.

2.7.10.1. Uzak ve Zaman Koordinat Mimarisi Ağları-UZKMA (Architectures of Spatiotemoral Network-STN)

Şekil 2.24; konuşulmuş kelimelerden türetilmiş STP 'lerin basit bir uygulamasını göstermektedir. Her bir zaman anında, spektrum analizörünün çıkışı; bileşenleri çeşitli kanallarda güçlendirilmiş bir vektörü içerir. Örneğin $t = t_1$ anında

$$P_i(t_1) = \left(P_{i1}(t_1), P_{i2}(t_1), \dots, P_{in}(t_1) \right)^t \quad (2.30)$$

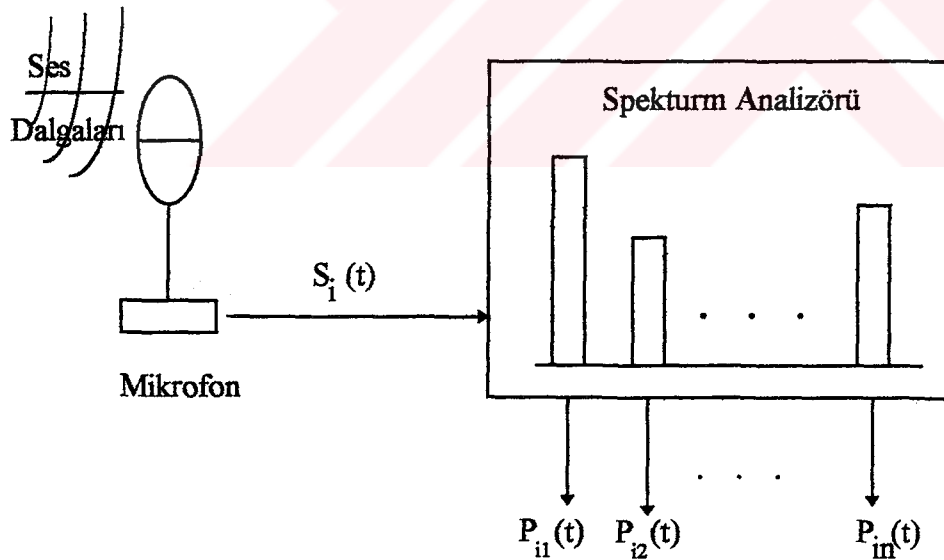
ifadesi t_1 anında i . kelime için güç spektrumdur. STP her bir kelime için $P_i(t_j)$ formunda vektör dizisi içerir. Eğer P_i ; i .kelime için STP ise

$$P_i = \{P_i(t_1), P_i(t_2), \dots, P_i(t_j)\} \quad (2.31)$$

şeklinindedir. Sesteki değişiklikler belli güç spektra normalizasyonu sayesinde açıklanabilir:

$$\|P_i(t_j)\| = 1 \quad (2.32)$$

Şekil 2.24. konuşma bilgisinden elde edilen genelleştirilmiş güç spektrasını göstermektedir. Mikrofon bir enerji dönüştürücü gibi davranır ve ses dalgalarından elektriksel $S_i(t)$ sinyalini üretir. i altındisi bir tek kelimeyi ifade eder. Spektrum analizörü her bir (n tane) frekans çubuğunun içerdiği gücü ölçer. $P_{i1}(t)$ konuşulmuş i . kelime için birinci frekans çubuğundaki gücü gösterir.



Şekil 2.24. Konuşma bilgisinden elde edilmiş güç spektrasının üretilmesi.

2.7.10.2. STN Uygulamaları

STN 'ler konuşma tanıma, radar analizi ve sonar-yankı sınıflandırması gibi alanlarda kullanılabilir. Bir sonar-yankı sınıflandırma sisteminin bir prototipi General Dynamics şirketi tarafından STN katman mimarisi kullanılarak geliştirilmiştir. Bu çalışmada,

gelen sonar sinyallerinin zaman dilimleri (slices) güç spektrasına çevrilerek uygun zaman dizisi içinde ağa sunulmuştur. Ağ; yedi ayrı tekne ile eğitildikten sonra, bu teknelerin hareketsiz sonar işaretlerinden gelen sinyallere göre, tekneleri birbirinden doğru bir şekilde ayırabilmiştir.

CPN ve STN yapıları arasındaki en büyük farklılık; STN 'deki her birimin çıkışı, daha sonra gelen birimlere giriş olur. Halbuki, gerçekte modellenmemiş CPN simülasyonundaki bağlantılar, bilgisayar sistemlerince kullanılmıştır. Benzer şekilde, STN 'deki katman üzerindeki birimler arası bağlantılar bilgisayar sistemlerinde icra edilen işlem algoritmaları tarafından hesaba katılır. Bu yüzden simülatör dizaynında bu bağlantıların hesaba katılmasına gerek yoktur.

2.7.11. Geriyansıtma Ağı (Backpropagation)

Geriyansıtma ağı; önceden tanımlanmış giriş ve çıkış örnek setlerini iki fazda "yansıtma-adapte etme" öğrenir. Ağın girişine bir patern (örüntü) uygulandıktan sonra, diğer üst katmanlara çıkış üretilinceye kadar yansıtılır. Yansıtma sonucunda elde edilen çıkış paterni daha sonra istenen çıkışla karşılaştırılır ve hata sinyali, her çıkış birimi için hesaplanır.

Hesaplanan hata sinyali daha sonra çıkış katmanından bir sonraki katmandaki birimlere transfer edilir. Bu işlem, ağdaki her birim için hata sinyali hesaplanıncaya kadar katman katman tekrarlanır. Çıkışta ilk yansıtımda elde edilen toplam hata sinyali, her düğümün ağırlık katsayıları oranlarına göre etki ettirilir. Her birim için alınan hata sinyali kadar yenilenen ağırlık katsayıları; ağın bütün eğitim paternlerinin kodlanabileceği bir duruma yakınsamasını sağlar.

Bu işlemin önemi, ağın orta katmanındaki farklı birimlerin, girişte sunulan farklı paternleri öğrenerek eğitilmesidir. Eğitimden sonra, gürültülü, eksik veya doğru bir patern sunulduğunda, orta ya da diğer adı ile saklı katmandaki birimler, eğitim sırasında öğrendiği paternler gibi doğru paternleri girişten alırsa, ancak o zaman aktif veya geçerli çıkış verir. Kısacası, saklı katmandaki birimler; eğer giriş paternleri ağın öğrendiği gibi değilse, çıkış vermeye eğilimli değildirler. Geriyansıtma ağı bölüm 4. 'de daha detaylı incelenecektir.

BÖLÜM 3. İTÜ TRİGA MARK-II REAKTÖRÜNÜN KONTROLÜ

Modern kontrol sistemleri nükleer alanda; sistem tanımlama, kontrol sistemi tasarımı, gürültü, kararlılık, yakıt yönetimi ve güç değişimi gibi problemlere uygulanmıştır. Nükleer reaktörler; karmaşık, lineer olmayan ve zaman bağımlı sistemlerdir. Isı transfer katsayısı, sürtünme katsayısı ve reaktivite geribeslemesi gibi çalışma koşullarına bağlı olarak değişen fiziksel özellikler ve kesin olarak bilinmeyen bazı fiziksel değerler doğru matematiksel modelin kurulmasını güçleştirmektedir.

Nükleer reaktörlerde performansın artırılabilmesi için matematiksel modele çok bağlı olmayan adaptif kontrol sistemleri geliştirilmiştir. Adaptif kontrolde, sistemin dinamik değişimlerine bağlı olarak kontrolör, parametreleri ayarlamaktadır. Yaygın olarak kullanılan adaptif kontrol yöntemleri; kazanç programlamalı kontrol, model referans adaptif kontrol, özayarlı (self-tuning) kontrol ve çalışmamızda incelediğimiz adaptif yapay sinir ağı kontrol yöntemleri sayılabilir (Yang and Linkens 1994, Wang vd 1994, Ahmed 1994, Lightbody and Irwin 1995).

Bu çalışmada, reaktörün lineer yapay sinir ağı kontrolü sonucundaki sistem parametre değerlerinin, lineer olmayan yapay sinir ağı kontrolörünün eğitiminde kullanılması sonucu, önceden yazılmış olan reaktör simülasyon programında reaktör, yapay sinir ağı kontrolörü ile kontrol edilmiştir.

Bu bölümde İTÜ TRİGA Mark-II reaktöründe kullanılan analog kontrol sistemi, lineer yapay sinir ağı kontrolör sistemi tanıtılacaktır.

3.1 Analog Kontrolör

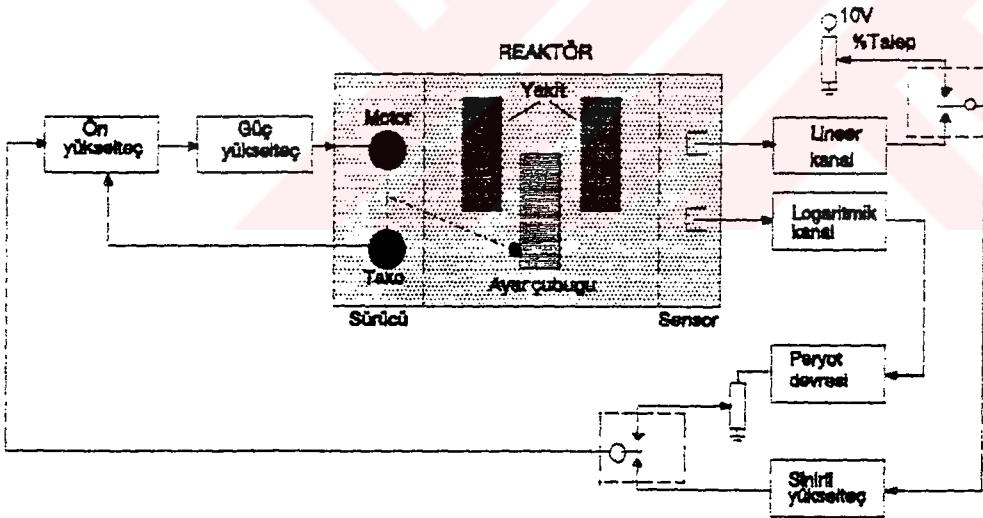
Kararlı hal modunda; reaktörün güç seviyesini değiştirme ve kararlı hal çalışması için elle kontrol (Manual) ve otomatik kontrol olmak üzere iki modda çalıştırılabilmektedir.

Reaktör gücü nötron yutucu kontrol çubukları yardımıyla ayarlanır. Kontrol çubuklarının pozisyonları elle kontrol modunda operatör tarafından, otomatik kontrol modunda ise elektro-mekanik kısıcılı analog kontrol sistemi tarafından ayarlanır.

Elle kontrol modu, reaktörü kaynak seviyesinden 100% güce çıkarmak için kullanılabilir. Güç değişimi için kontrol panelindeki yukarı / aşağı (UP / DOWN) butonlarıyla çubuk pozisyonları ayarlanır. Sayısal göstergelerden okunabilen çubuk pozisyonları alt veya üst sınıra geldiğinde yukarı / aşağı butonlarının altındaki lamba ikaz verir. Kontrol çubukları butonlarla tek tek düşürülebildiği gibi SCRAM çubuğu ile toplu olarak aynı anda düşürülebilir.

Reaktör çalışırken; çubuk pozisyonları, güç seviyesi, peryot, yakıt sıcaklığı ve akışkan sıcaklığı bilgilerinin görülmesi gereklidir. Reaktörde; 1 tane sıcaklık, 2 tane yakıt ve 3 tane güç kanalı vardır. Geniş bantlı logaritmik güç kanalından gelen nötron yoğunluğu bilgisi kontrol konsolündeki yazıcıda görüntülenir. Peryot ve yakıt sıcaklığı ise analog ölçü aletleri ile gösterilmektedir.

Otomatik kontrol modunda, reaktör gücünü ayarlamak için kontrol çubuğu pozisyonu geribeslemeli kontrol sistemi ile kontrol edilir. Kontrol çubuğunun pozisyonunu değiştiren motorun uçları yukarı / aşağı butonlarından ayrılarak servo yükselteç çıkışına bağlanmaktadır.



Şekil 3.1. Analog kontrol sistemi

Şekil 3.1 'de görülen reaktörün otomatik kontrol sistemi; sensor, lineer ve logaritmik yükselteçler, peryot, sınırlı yükselteç, önyükselteç, güç yükselteci ve çubuk sürme birimlerinden oluşmaktadır. Reaktör kalbine yerleştirilen nötron dedektörleri, güç değişimlerini algılayarak ilgili görüntüleme kanallarına gerekli sinyalleri sağlarlar. Dedektörden gelen sinyalin görüntülenmesinde logaritmik ve lineer kanallar kullanılır. Lineer güç kanalı, logaritmik güç kanalının ön yükselteciden gelen dc sinyali değerlendirerek 19 ayrı kademedeki görüntülenmesini sağlamaktadır.

İstenilen güç, %TALEP (%Demand) potansiyometresi ile ayarlanmaktadır. %TALEP potansiyometresi, REAKTÖR GÜCÜ anahtarının yüzdesi olarak kalibre edilmiştir. İstenilen güç ile lineer güç 50Hz ile çalışan kıyıcıda karşılaştırılarak birinci hata sinyali elde edilir. Birinci hata sinyali; genliği güç hatasının mutlak değerine, fazı ise, hatanın yönüne bağlı olan 50Hz kare dalgadır.

Büyük güç değişimlerinde, değişimin sabit peryotla yapılabilmesi için birinci hata sinyalinin genliğinin sınırlandırılması gerekmektedir. Analog kontrol sisteminde bu amaçla kazancı 100 olan ve çıkışı 4.7V ile sınırlandırılmış bir yükselteç kullanılmıştır. Birinci hata sinyalinin genliğinin sınırlandırılması güç değişimlerinin sabit peryotla yapılmasını sağlamaya yetmez. Güç değişimlerinin sabit peryotla yapılabilmesini sağlamak için, girişine logaritmik kanaldan nötron yoğunluğunun logaritması ile orantılı olarak gerilim gelen peryot devresi kullanılmaktadır. Peryot devresinin çıkışında, ters peryot sinyali elde edilmektedir. Ters peryot sinyali bir potansiyometre üzerinden birinci hata sinyali ile karşılaştırılmak üzere kıyıcıya gönderilmektedir.

Kıyıcıya uygulanan ters peryot sinyalinin genliğinin artması güç değişim hızını azaltır. Ters peryot sinyali ile birinci hata sinyali kıyıcıda karşılaştırılarak güç değişiminin sabit peryotla yapılmasını sağlayabilecek ikinci hata sinyali elde edilir. Reaktör gücü istenilen seviyeye erişirken peryot büyür ve sonsuza yaklaşır, hata sinyalleri ise sıfıra gider.

İkinci hata sinyali önyükselteç katında yükseltilir ve takometreden gelen hız sinyali ile toplanarak güç katına uygulanır. Önyükselteç katında elde edilen sinyal, kontrol çubuklarının servomotorunu çalıştırabilecek seviyeye güç yükselteç katında çıkarılır.

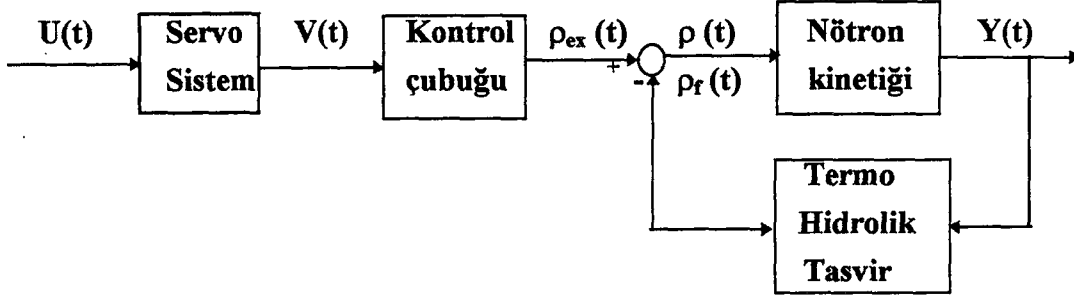
Servomotorun dönüş yönü, uygulanan sinyalin fazına bağlıdır. Motorun dönüş yönü kontrol çubuğunun çekilmesini veya indirilmesini belirler. Çubuk pozisyonu motora bağlı çok türlü potansiyometre üzerinden okunmaktadır. (ERKAN 1994)

3.2 Lineer Yapay Sinir Ağı Kontrolörü

3.2.1. Sistemin Ayrık Zaman Modeli

Sayısal kontrolü amaçlanan reaktörün basitleştirilmiş şeması Şekil 3.2'de görülmektedir.

Reaktörün açık çevrim kontrolünde, servo sistem girişine uygulanan gerileme, $U(t)$, bağlı olarak kontrol çubuğu hareket ettirilmekte ve kontrol çubuğunun pozisyonuna, $V(t)$, bağlı olarak reaktöre harici reaktivite, $\rho_{ex}(t)$, uygulanmaktadır. Harici reaktivite ve sıcaklığa bağımlı negatif geribesleme reaktivitesinin, olarak $\rho_f(t)$, toplamı net reaktiviteyi, $\rho(t)$, vermektedir. Net reaktivite, nötron kinetiğine göre reaktörün çıkış gücünü, $Y(t)$, değiştirir.



Şekil 3.2. Reaktörün basitleştirilmiş şeması

Sayısal kontrolörün gerçekleştirilebilmesi için reaktörün ayrık zaman modelinin oluşturulması gerekir. Analog / sayısal, sayısal / analog dönüşümler, yuvarlama hataları ve bozucu etkisi altındaki sistemin ayrık zaman modeli alışılmış notasyonla, aşağıdaki gibi ifade edilebilir:

$$A(z^{-1}) Y(t) = z^{-k} B(z^{-1}) U(t) + C(z^{-1}) \xi(t) \quad (3.1)$$

burada, $Y(t)$, sistem çıkışı (güç) .. $U(t)$, kontrol sinyalini (motor hızı) .. $\xi(t)$, gürültüyü .. z^{-1} , geri kaydırma operatörünü .. t , örnekleme anını .. k gecikme katsayısını .. A, B, C polinomları göstermektedir.

Eşitlik 3.1 'deki A, B ve C polinomları;

$$A(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_{na} z^{-na}$$

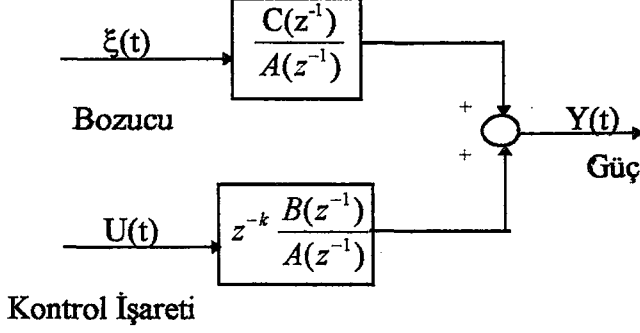
$$B(z^{-1}) = b_0 + b_1 z^{-1} + \dots + b_{nb} z^{-nb} \quad (3.2)$$

$$C(z^{-1}) = c_0 + c_1 z^{-1} + \dots + c_{nc} z^{-nc}$$

formundadır. Eşitlik 3.2'deki na, nb ve nc sırasıyla A, B ve C polinomlarının derecelerini göstermektedir. Eşitlik 3.1'i

$$Y(t) = z^{-k} \frac{B(z^{-1})}{A(z^{-1})} U(t) + \frac{C(z^{-1})}{A(z^{-1})} \xi(t) \quad (3.3)$$

şeklinde yeniden yazarsak, eşitliğin sağındaki ilk terim kontrol işaretinin, ikinci terim ise bozucunun etkisini göstermektedir. Eşitlik 3.3., blok olarak Şekil 3.2 'deki gibi ifade edilebilir.



Şekil 3.3. Sistemin ayrık zaman modeli

Reaktörü temsil eden gerçekçi model yüksek dereceden olmasına karşın, pratikte basitleştirilmiş modeller kullanılmaktadır. Çalışmamızda;

$$n_a=2, n_b=1, n_c=1, k=1 \quad (3.4)$$

olarak alınmıştır. Bundan sonraki matematiksel işlemlerde hep bu değerler kullanılacaktır.

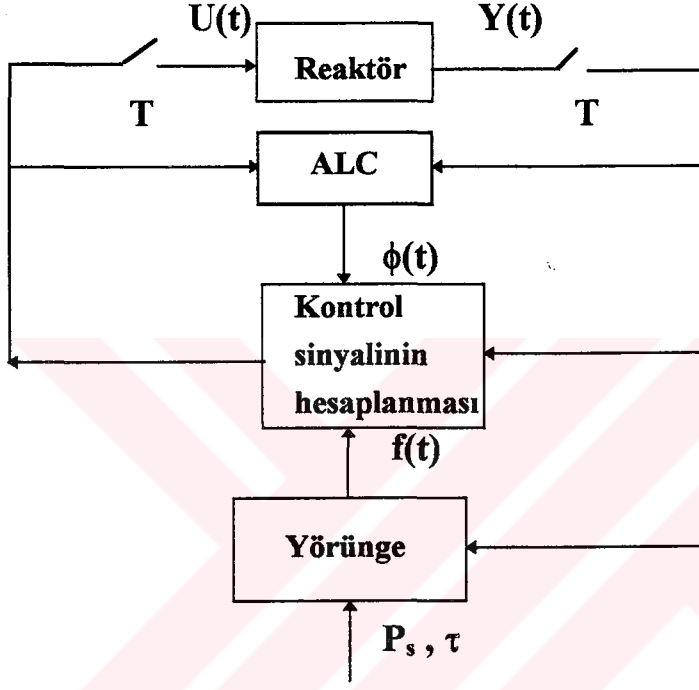
3.2.2. Genelleştirilmiş Minimum Varyans Kontrol

Sayısal kontrolörün tasarımında amaç; kontrol çubuğunun pozisyonunu değiştirerek gücü yalnız bozuculara karşın belirli noktada sabit tutmak değil, aynı zamanda istenilen güce, verilen yörüngeyi izleyerek erişmesini sağlamaktır. Çalışmamızda genelleştirilmiş minimum varyans yöntemi kullanılarak kontrolör tasarlanacaktır (Bittanti and Piroddi 1994).

Şekil 3.4 'de görüldüğü gibi kontrolör; yörünge, kontrol sinyalinin hesaplanması ve ALC (Adaptive Linear Combiner)-Adaptif Lineer Birleştirici olmak üzere üç birimden oluşmaktadır. ALC bloğunda, kontrolör parametrelerinin kestirimi yapılmaktadır (Raol 1994).

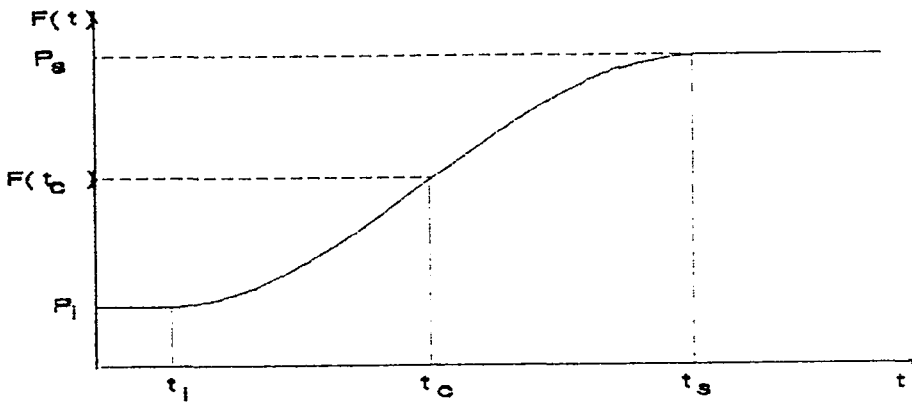
3.2.2.1. Yörünge

Yörünge biriminde, gücün, bir sonraki örneklemede alması istenen değer hesaplanır. Yörüngeye operatör tarafından istenilen çalışma güç düzeyi (ayar noktası - setpoint), P_s , girilmektedir. Reaktör gücünün o anki değeri gözönüne alınarak istenilen değere erişmesi için gereken süre ve izleyeceği yol bulunur.



Şekil 3.4. Lineer Kontrolörün blok diyagramı

Çalışmamızda Şekil 3.5’de görüldüğü gibi parabolik bir yörünge seçilmiştir.



Şekil 3.5. Yörünge

Yörünge zaman bağımlılığı aşağıdaki gibi verilmektedir:

$$F(t) = f_3(t-t_i)^3 + f_2(t-t_i)^2 + f_1(t-t_i) + f_0 \quad , \quad t_i \leq t \leq t_s$$

$$F(t) = P_i \quad , \quad t < t_i$$

$$F(t) = P_s \quad , \quad t > t_s$$
(3.5)

Burada ; f_0 , f_1 , f_2 ve f_3 polinom katsayılarını, t_i başlangıçtaki kararlı halden ayrılma noktasını, P_i , t_i anındaki reaktör gücünü, t_s , ayar noktasına erişme zamanını, P_s , ayar noktasını göstermektedir.

Eşitlik 3.5 'in türevi alınır,

$$f'(t) = 3f_3(t-t_i)^2 + 2f_2(t-t_i) + f_1 \quad , \quad t_i \leq t \leq t_s$$

$$f'(t) = 0 \quad , \quad t \leq t_i$$

$$f'(t) = 0 \quad , \quad t \geq t_s$$
(3.6)

elde edilir.

Eşitlik 3.5. ve Eşitlik 3.6'nın kullanılmasıyla,

$$f_0 = P_i \quad , \quad f_1 = 0 \quad , \quad f_2 = 3 * \frac{P_s - P_i}{(t_s - t_i)^2} \quad , \quad f_3 = -2 * \frac{P_s - P_i}{(t_s - t_i)^3}$$
(3.7)

elde edilir.

Reaktör gücünün sabit bir periyotla değişmesi istendiğinden bu şartın yörüngeye ithal edilmesi yararlı olur. Bu nedenle, gücün artmaya başladığı bölgenin orta noktasında (t_c) , periyodun istenen periyoda eşit olduğu varsayılır ve istenen periyot aşağıdaki şekilde hesaplanır:

$$\tau = f(t_c) / f'(t_c)$$
(3.8)

Eşitlik 3.5 ve 3.6 'nın Eşitlik 3.8'de kullanılması ile ayar noktasına erişme zamanı:

$$t_s = t_i + 3 * \tau * \frac{P_s - P_i}{(t_s - t_i)^3}$$
(3.9)

elde edilir.

3.2.2.2. Kontrol Sinyalinin Hesaplanması

Reaktör girişine uygulanacak optimal kontrol sinyalini hesaplamak için seçilen maliyet fonksiyonunun minimize edilmesi gerekmektedir. Bu çalışmada maliyet fonksiyonu,

$$J=E\{y(t)\} = E\left\{\alpha \left[Y(t+k)-R(z^{-1})F(t+k) \right]^2 + \beta \left[\tau \frac{1}{T} Y(t+k)-Y(t+k) \right]^2 + \mu \left[Q(z^{-1})U(t) \right]^2 \mid t \right\} \quad (3.10)$$

şeklinde tanımlanmıştır.

Burada, $E\{\cdot\}$, t anına kadar toplanan verilere bağlı olarak $(t+k)$ anında beklenen değeri, $F(t)$, yörüngeyi, τ , periyodu, α , β , μ , ağırlık katsayılarını, R , Q ağırlık polinomlarını göstermektedir.

Eşitlik 3.10 'un sağındaki ilk terim yörüngeyle, ikinci terim periyotla, son terim ise kontrol sinyali ile ilgilidir.

Eşitlik 3.10'daki türev, "geriye fark" (backward difference) yaklaşımı gözönüne alınarak

$$\frac{1}{T} Y(t+1) = \frac{Y(t+1)-Y(t)}{T} \quad (3.11)$$

şeklinde yazılabilir. Eşitlik 3.11'de, T örnekleme aralığını temsil etmektedir.

Kayıdırma teoremi,

$$Y(t+1) = z^*Y(t)-z^*Y(0) \quad (3.12)$$

eşitliğinde $Y(0)=0$ alınırsa,

$$Y(t+1)=z^*Y(t) \quad (3.13)$$

$$Y(t)=z^{-1}*Y(t+1) \quad (3.14)$$

eşitlikleri elde edilir.

Eşitlik 3.14'ün, Eşitlik 3.11'de kullanılmasıyla türev,

$$\frac{\partial}{\partial U(t)} Y(t+1) = \left(\frac{1-z^{-1}}{T} \right) * Y(t+1) \quad (3.15)$$

şeklinde yazılabilir.

Optimum kontrol işaretini bulmak için, Eşitlik 3.10'un U'ya göre türevi alınıp sıfıra eşitlenir ve Eşitlik 3.15. kullanılırsa,

$$\begin{aligned} \frac{\partial \gamma(t)}{\partial U(t)} &= 2\alpha \left[Y(t+1|t) - R(z^{-1})F(t+1) \right] \frac{\partial Y(t+1|t)}{\partial U(t)} + \\ & 2\beta \left[\left(\frac{\tau-T}{T} - \frac{\tau}{T} z^{-1} \right) Y(t+1) \right] \frac{\tau-T}{T} \frac{\partial Y(t+1|t)}{\partial U(t)} + \\ & 2\mu Q(z^{-1})U_0(t) = 0 \end{aligned} \quad (3.16)$$

elde edilir. Burada, $U_0(t)$ optimum kontrol işaretini göstermektedir.

Eşitlik 3.1. ve 3.2. kullanılarak,

$$A(z^{-1})Y(t+1) = B(z^{-1})U(t) + C(z^{-1})\xi(t+1)$$

$$Y(t+1) + a_1 * Y(t) + a_2 * Y(t-1) = b_0 * U(t) + b_1 * U(t-1) + \xi(t+1) + c_1 * \xi(t)$$

$$Y(t+1) = -a_1 * Y(t) - a_2 * Y(t-1) + b_0 * U(t) + b_1 * U(t-1) + \xi(t+1) + c_1 * \xi(t)$$

$$\frac{\partial Y(t+1)}{\partial U(t)} = \frac{\partial Y(t+1|t)}{\partial U(t)} = b_0 \quad (3.17)$$

yazılabilir. Eşitlik 3.16'daki μ ve β değerleri,

$$\mu = \lambda b_0, \quad \beta = \left(\frac{T}{\tau - T} \right)^2 \quad (3.18)$$

seçilirse Eşitlik 3.16,

$$\left[(\alpha + 1) - \frac{\tau}{\tau - T} z^{-1} \right] Y(t+1|t) - \alpha R(z^{-1})F(t+1) + \lambda Q(z^{-1})U_0(t) = 0 \quad (3.19)$$

olarak elde edilir. Eşitlik 3.19.,

$$P(z^{-1}) = p_0 + p_1 z^{-1}, \quad p_0 = \alpha + 1, \quad p_1 = -\frac{\tau}{\tau - T} \quad (3.20)$$

tanımları kullanılarak aşağıdaki şekilde yeniden yazılabilir:

$$P(z^{-1})Y(t+1 | t) - \alpha R(z^{-1})F(t+1) + \lambda Q(z^{-1})U_0(t) = 0 \quad (3.21)$$

Eşitlik 3.4. ve 3.13., Eşitlik 3.3'de kullanıldığında

$$Y(t+1) = \frac{B(z^{-1})}{A(z^{-1})} U(t) + \frac{C(z^{-1})}{A(z^{-1})} \xi(t+1) \quad (3.22)$$

elde edilir.

Eşitlik 3.22'deki bozucu terim çarpanının seri açılımı yapılırsa,

$$\frac{C(z^{-1})}{A(z^{-1})} = \frac{1 + c_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} = 1 + (c_1 - a_1) z^{-1} + (a_1^2 - a_2 - a_1 c_1) z^{-2} + \dots \quad (3.23)$$

elde edilir.

Eşitlik 3.23, Eşitlik 3.22'de gösterildiği gibi bozucu terim ile çarpılırsa,

$$\frac{C(z^{-1})}{A(z^{-1})} * \xi(t+1) = \xi(t+1) + (c_1 - a_1) * \xi(t) + (a_1^2 - a_2 - a_1 c_1) * \xi(t-1) + \dots \quad (3.24)$$

eşitliği elde edilir.

Eşitlik 3.24.'de, bozucu terimin mevcut değerleri, $(\xi(t), \xi(t-1), \xi(t-2), \dots)$ ve bozucu terimin gelecek değeri, $(\xi(t+1))$ bulunmaktadır. Bozucu terimin gelecek değeri kestirilemediğinden, bozucunun mevcut ve gelecek değerlerini ayırmak için,

$$\frac{C(z^{-1})}{A(z^{-1})} = E(z^{-1}) + z^{-1} \frac{G(z^{-1})}{A(z^{-1})} \quad (3.25)$$

eşitliği (Diphontane equation) kullanılabilir.

Eşitlik 3.4.'de seçilen değerlere göre polinomlar:

$$\begin{aligned}
 A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} \\
 B(z^{-1}) &= b_0 + b_1 z^{-1} \\
 C(z^{-1}) &= 1 + c_1 z^{-1} \\
 E(z^{-1}) &= e_0 \\
 G(z^{-1}) &= g_0 + g_1 z^{-1}
 \end{aligned} \tag{3.26}$$

şeklinde yazılmaktadır.

Eşitlik 3.26. 'daki değerler, Eşitlik 3.25' de kullanılırsa:

$$\frac{1 + c_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} = e_0 + z^{-1} \frac{g_0 + g_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{3.27}$$

eşitliği elde edilir.

Eşitlik 3.27'den, $e_0=1$, $g_0=c_1-a_1$ ve $g_1=-a_2$ olarak bulunur. Buna göre E ve G polinomları aşağıdaki gibi yeniden yazılabilir.

$$E(z^{-1})=1, \quad G(z^{-1})=(c_1-a_1)+(-a_2)z^{-1} \tag{3.28}$$

Eşitlik 3.22; Eşitlik 3.25. ve Eşitlik 3.28. aracılığı ile yeniden aşağıdaki biçimde yazılabilir:

$$Y(t+1) = \xi(t+1) + \frac{G(z^{-1})}{A(z^{-1})} \xi(t) + \frac{B(z^{-1})}{A(z^{-1})} U(t) \tag{3.29}$$

Eşitlik 3.1 'den $\xi(t)$ aşağıdaki gibi bulunabilir.

$$\xi(t) = \frac{A(z^{-1})}{C(z^{-1})} Y(t) + z^{-1} \frac{B(z^{-1})}{C(z^{-1})} U(t) \tag{3.30}$$

Eşitlik 3.30., Eşitlik 3.29 'da kullanılırsa,

$$Y(t+1) = \xi(t+1) + \frac{G(z^{-1})}{C(z^{-1})} Y(t) + \left[\frac{B(z^{-1})}{A(z^{-1})} - z^{-1} \frac{G(z^{-1})B(z^{-1})}{A(z^{-1})C(z^{-1})} \right] U(t) \quad (3.31)$$

elde edilir.

Eşitlik 3.25. Eşitlik 3.31’de kullanılırsa $Y(t+1)$ aşağıdaki gibi bulunur:

$$Y(t+1) = \xi(t+1) + \frac{G(z^{-1})}{C(z^{-1})} Y(t) + \frac{B(z^{-1})}{C(z^{-1})} U(t) \quad (3.32)$$

Gelecekteki bozucuyu kestiremediğimiz için $\xi(t+1)$ terimi maliyet fonksiyonunun minimizasyonunda ihmal edilecektir. Eğer t anına kadar giriş ve çıkış değerleri verilirse, çıkışın bir adım sonra alacağı değer,

$$Y(t+1 | t) = \frac{G(z^{-1})}{C(z^{-1})} Y(t) + \frac{B(z^{-1})}{C(z^{-1})} U(t) \quad (3.33)$$

olarak bulunur.

Eşitlik 3.33. Eşitlik 3.21’de kullanılırsa,

$$G(z^{-1})P(z^{-1})Y(t) - \alpha C(z^{-1})R(z^{-1})F(t+1) + \left[B(z^{-1})P(z^{-1}) + \lambda C(z^{-1})Q(z^{-1}) \right] U_0(t) = 0 \quad (3.34)$$

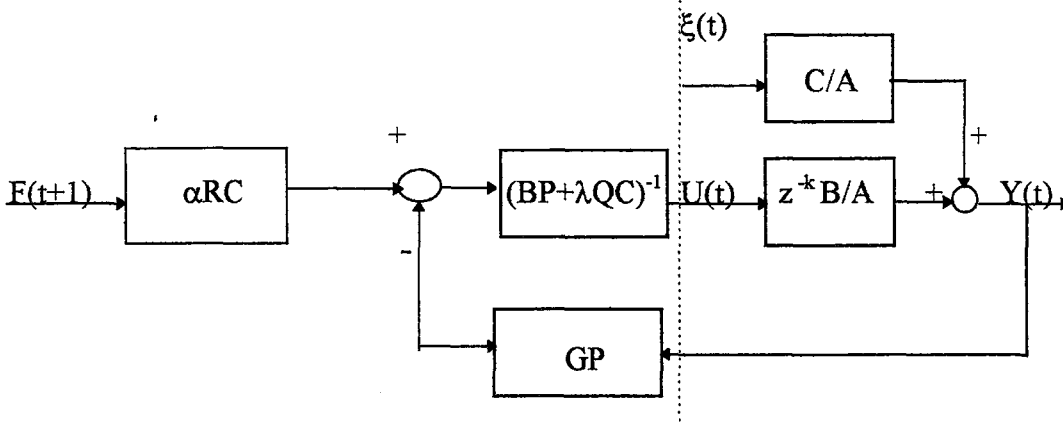
elde edilir.

Eşitlik 3.34’den kontrol işaretinin optimum değeri,

$$U_0(t) = \frac{\alpha C(z^{-1})R(z^{-1})F(t+1)}{B(z^{-1})P(z^{-1}) + \lambda C(z^{-1})Q(z^{-1})} - \frac{G(z^{-1})P(z^{-1})Y(t)}{B(z^{-1})P(z^{-1}) + \lambda C(z^{-1})Q(z^{-1})} \quad (3.35)$$

olarak bulunur.

Eşitlik 3.3 ve Eşitlik 3.35 baz alınarak kontrol sistemi Şekil 3.6 ‘daki gibi çizilebilir.



Şekil 3.6. Kontrol sistemi

Eşitlik 3.35'in Eşitlik 3.1'de kullanılmasıyla sistemin kapalı çevrim yanıtı aşağıdaki şekilde yazılabilir:

$$\begin{aligned}
 A(z^{-1})Y(t) = z^{-1}B(z^{-1}) & \left(\frac{\alpha C(z^{-1})R(z^{-1})F(t+1)}{B(z^{-1})P(z^{-1})+\lambda Q(z^{-1})C(z^{-1})} - \frac{G(z^{-1})P(z^{-1})Y(t)}{B(z^{-1})P(z^{-1})+\lambda Q(z^{-1})C(z^{-1})} \right) \\
 + C(z^{-1})\xi(t) & \left(A(z^{-1}) + z^{-1} \frac{B(z^{-1})P(z^{-1})G(z^{-1})}{B(z^{-1})P(z^{-1})+\lambda Q(z^{-1})C(z^{-1})} \right) Y(t) = \\
 z^{-1} \frac{B(z^{-1})C(z^{-1})R(z^{-1})F(t+1)}{B(z^{-1})P(z^{-1})+\lambda Q(z^{-1})C(z^{-1})} & + C(z^{-1})\xi(t) \quad (3.36)
 \end{aligned}$$

Eşitlik 3.14 'de verilen kaydırma teoremi Eşitlik 3.36 'da kullanılarak,

$$(B(z^{-1})P(z^{-1})A(z^{-1})+\lambda Q(z^{-1})A(z^{-1})C(z^{-1})+z^{-1}B(z^{-1})P(z^{-1})G(z^{-1}))Y(t)=$$

$$B(z^{-1})C(z^{-1})R(z^{-1}) F(t)+C(z^{-1})R(z^{-1})F(t)+C(z^{-1})(B(z^{-1})P(z^{-1})+\lambda Q(z^{-1})C(z^{-1}))\xi(t) \quad (3.37)$$

Eşitlik 3.25 'deki eşitlikten $z^{-1}G(z^{-1})$ bulunup Eşitlik 3.37 'de kullanılırsa sistemin kapalı çevrim cevabı aşağıdaki şekilde tekrar yazılabilir.

$$(B(z^{-1})P(z^{-1})A(z^{-1})+\lambda Q(z^{-1})A(z^{-1}))Y(t)=B(z^{-1})R(z^{-1})F(t)+$$

$$(B(z^{-1})P(z^{-1})+\lambda Q(z^{-1})C(z^{-1}))\xi(t)$$

$$Y(t) = z^{-1} \frac{\alpha B(z^{-1})}{B(z^{-1})P(z^{-1}) + \lambda Q(z^{-1})A(z^{-1})} R(z^{-1})F(t) + \frac{B(z^{-1})P(z^{-1}) + \lambda Q(z^{-1})C(z^{-1})}{B(z^{-1})P(z^{-1}) + \lambda Q(z^{-1})A(z^{-1})} \xi(t) \quad (3.38)$$

F(t) ile Y(t) arasındaki kararlı hal hatasının sıfır olması gerekir. Eşitlik 3.38 'de $Y(t) - F(t) = 0$ yazıldığı takdirde gürültü ihmal edilirse kararlı hal hatasının sıfır olması için gerekli koşul elde edilir.

$$R(1) = \frac{B(z^{-1})P(z^{-1}) + \lambda Q(z^{-1})A(z^{-1})}{\alpha B(z^{-1})} \Big|_{z=1} \quad (3.39)$$

eşitliğinin sağlanması gerekir.

Kontrol sinyalindeki ani değişimleri sınırlandırmak için Q aşağıdaki gibi seçilebilir.

$$Q(z^{-1}) = (1 - z^{-1}) \quad (3.40)$$

Eşitlik 3.39, Eşitlik 3.40 aracılığı ile yeniden aşağıdaki biçimde yazılabilir.

$$R(z^{-1}) = (p_0 + p_1) / \alpha \quad (3.41)$$

Eşitlik 3.40 ve Eşitlik 3.41'in Eşitlik 3.35'de kullanılması ile optimum kontrol sinyali aşağıdaki şekilde yazılabilir:

$$U_0(t) = \frac{(p_0 + p_1)(1 + c_1 z^{-1})F(t+1) - (g_0 p_0 + (g_0 p_1 + g_1 p_0)z^{-1} + (g_1 p_1)z^{-2})Y(t)}{g_0 p_0 + (g_0 p_1)z^{-1} + g_1 p_1 z^{-2}} \quad (3.42)$$

Eşitlik 3.14 kullanılarak Eşitlik 3.42, aşağıdaki şekilde tekrar yazılabilir:

$$U_0(t) = \left[(p_0 + p_1)(F(t+1) + c_1 F(t)) - ((g_0 p_0)Y(t) + (g_0 p_1 + g_1 p_0)Y(t-1) + (g_1 p_1)Y(t-2)) - ((b_0 p_1 + b_1 p_0 + \lambda c_1 - \lambda)U_0(t-1) + (b_1 p_1 - \lambda c_1)U_0(t-2)) \right] / (b_0 p_0 + \lambda) \quad (3.43)$$

Optimal kontrol sinyalinin Eşitlik 3.43 kullanılarak hesaplanabilmesi için G, B ve C polinomları katsayılarının Adaptif Lineer Birleştirici (ALC) Y.S.A. modeli kullanılarak bulunması mümkündür.

3.2.3. Linear Yapay Sinir Ağı Kontrolörü-Adaptif Lineer Birleştirici (ALC)

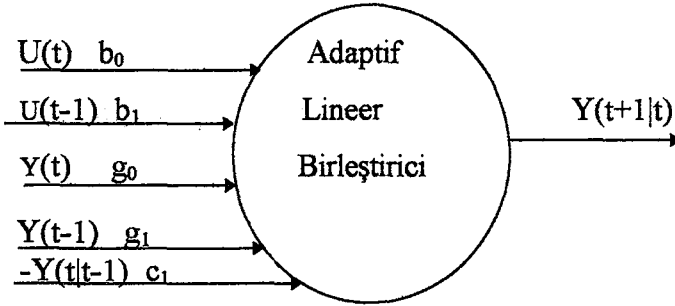
Bölüm 2.7.2. 'de açıklanan Adaptif Lineer Birleştirici - ALC kullanılarak optimum kontrol sinyalinin hesaplanmasında kullanılan Eşitlik 3.43 'deki B, C ve G polinomlarının katsayıları hesaplanmıştır. Bu katsayıların ALC ile bulunabilmesi için Eşitlik 3.33 'ü $C(z^{-1})$ ile çarparsak,

$$C(z^{-1})Y(t+1|t)=G(z^{-1})Y(t)+B(z^{-1})U(t) \quad (3.44)$$

elde edilir. Eşitlik 3.44 'ü daha açık olarak aşağıdaki şekilde yazabiliriz:

$$Y(t+1|t)=g_0Y(t)+g_1Y(t-1)+b_0U(t)+b_1U(t-1)-c_1Y(t-1) \quad (3.45)$$

Şekil 3.7. 'de görüldüğü gibi ALC; Eşitlik 3.45' e göre, t anında sistemin t+1 anındaki tahmini çıkış değerini bulmaktadır. Ağırlıkların bulunmasında Bölüm 2.5.2.1 'de anlatılan en küçük kareler yöntemi kullanılmıştır (Cichocki and Unbehauen 1994).



Şekil 3.7. ALC kullanılarak sistem çıkışının bulunması.

3.3. Lineer Olmayan Y.S.A. Kontrolörü

Eğitim seti lineer Y.S.A. kontrolünden elde edilen Lineer olmayan Y.S.A. kontrolörü; aşağıdaki tahmini ayrık-zaman fark eşitliğindeki $u(k)$ kontrol sinyalini temsil etmektedir:

$$u(k) = f[y(k), y(k-1), \dots | u(k-1), u(k-2), \dots, r(k), r(k-1), \dots | t(k), t(k-1), \dots | e_p(k), e_p(k-1), \dots | e_\gamma(k), e_\gamma(k-1), \dots] \quad (3.46)$$

Eşitlik 3.46.'daki $f[\dots]$ lineer olmayan fonksiyonu, $u(k)$ kontrol girişini (çubuk hızı), $y(k)$ ölçülen çıkışı (güç), $r(k)$ çubuk pozisyonunu, $t(k)$ yakıt sıcaklığını, $e_p(k)$ güç hatasını, $e_\gamma(k)$, periyodun tersini ve k ise örnekleme zamanını göstermektedir.

ITU TRIGA Mark-II araştırma reaktörü geniş bir güç aralığında çalıştığından, istenen güç değerine göre bir güç hatası ve gücün belli bir periyotla çıkması istendiğinden, bu periyota bağlı olarak oluşan periyodun tersi aşağıdaki eşitlikte verilmiştir:

$$e_p(k) = 1 - y(k)/P_s \quad (3.47)$$

$$e_\gamma(k) = \tau_s^{-1}(k) - \tau^{-1}(k) \quad (3.48)$$

$$\tau_s^{-1}(k) = \tau_d^{-1}, |e_p(k)| > e_1 \quad (3.49)$$

$$\tau_s^{-1}(k) = 0, |e_p(k)| \leq e_1 \quad (3.50)$$

Burada, P_s istenen güç değerini, τ_d^{-1} istenen ters periyot değerini, $\tau^{-1}(k)$ o anki ters periyot değerini, $\tau_s^{-1}(k)$ o anki sunulmuş istenen ters periyot değerini ve e_1 ise kabul edilebilir güç hatasını göstermektedir.

Eşitlik 3.46 daha basit bir şekilde ifade edilirse;

$$u(k) = f[y(k), r(k), t(k), e_p(k), e_\gamma(k)] \quad (3.51)$$

şeklinde seçilebilir.

Bu durumda, kontrol sinyali; güç, çubuk pozisyonu, yakıt sıcaklığı, güç hatası ve ters peryot hatasının o anki ölçülmüş değerlerinin bir fonksiyonu durumuna gelir.

Bu ölçülen değerleri bir ölçüm vektörü $\phi(k)$:

$$\phi(k) = f[y(k), r(k), t(k), e_p(k), e_v(k)] \quad (3.52)$$

şeklinde ifade edilebilir. Buna göre Eşitlik 3.51 aşağıdaki gibi ifade edilebilir:

$$u(k) = f[\phi(k)] \quad (3.53)$$

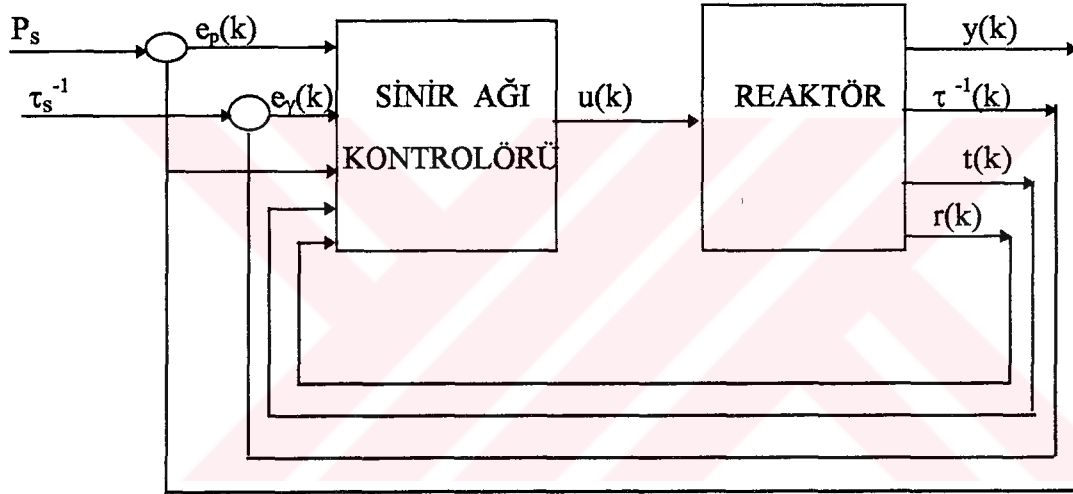
$f[\cdot]$ lineer olmayan ve bilinmeyen bir fonksiyon olduğuna göre; yapay sinir ağı kullanılarak bu fonksiyonun bir yaklaşım ifadesi bulunabilir. Lineer olmayan yapay sinir ağının kullanılması için, lineer kontrolörden üretilmiş veri; eğitim seti olarak kullanılmıştır. Eğitim verisi, simülasyonun 0.05 sn. örnekleme aralıklarıyla çalıştırılmasından elde edilmiştir. Amaçlanan sinir ağı kontrolörü Şekil 3.8'de görülmektedir.

Bölüm 4'te açıklanan geriyansıtma ağ modeli, lineer olmayan kontrolör tasarımında kullanılmıştır. Kontrolörün veri eğitim seti Bölüm 3.2'de açıklanan lineer kontrolör tarafından üretilmektedir. Burada lineer kontrolör aracılığıyla bulunan sistem çıkışına göre simülasyon 0.05 sn. örnekleme aralığında, 10 sn.'lik peryotla, 250 Kw istenen güç değerinde çalıştırılarak; lineer olmayan kontrolör için gerekli eğitim seti üretilmiştir. Eğitim seti 1000 tane veri paterninden oluşmaktadır. Her veri paterni, normalize edilmiş güç, güç hatası, çubuk pozisyonu, ters peryot, yakıt sıcaklığı ve kontrol sinyali verilerinden oluşmaktadır. Sinir ağının eğitiminde yörünge kullanılmamıştır.

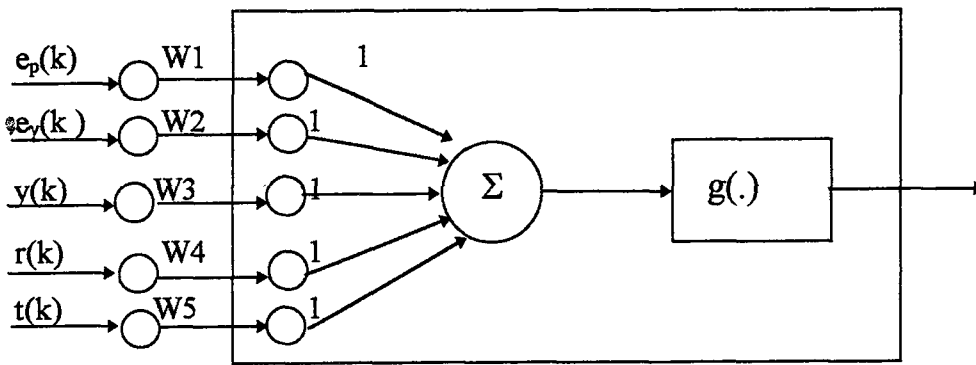
Lineer olmayan işlem birimlerinin kullanıldığı kontrolörde, reaktör istenen güce ulaştığında, yani kararlı hal durumuna geldiğinde kontrol sinyalinin sıfır olması beklenir. Kontrol sinyalinin sıfır olması için, güç ve ters peryot hatalarının sıfır olması gerekir. Bu koşulun yerine getirilmesi için Şekil 3.9'da gösterilen klasik işlem birimi

yapısı yeterli olmamaktadır. Bunun yerine Şekil 3.10'da gösterilen işlem birimi yapısı kontrolör tasarımında kullanılmıştır. Şekil 3.10'a göre işlem biriminin ters periyot hatası $e_r(k)$ ve güç hatası $e_p(k)$ girişleri sıfır olduğunda, işlem biriminin çıkışı sıfır olacaktır.

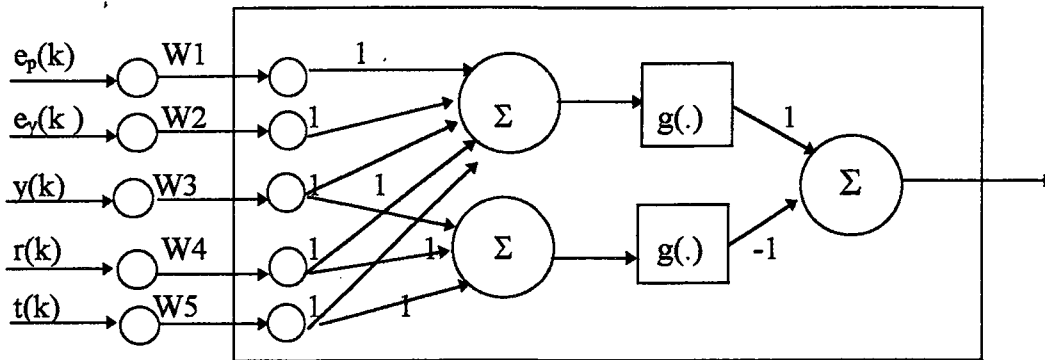
Tasarlanan kontrolör beş katmandan oluşmaktadır. Bir giriş katmanı, üç saklı katman ve bir de çıkış katmanına sahip olan ağ yapısı, saklı katmandaki işlem birimi yapısı Şekil 3.10'da gösterildiği gibi kabul edilirse, saklı katman sayısı bir olarak kabul edilebilir.



Şekil 3.8. Amaçlanan Sinir Ağı Kontrolörü



Şekil 3.9. Klasik İşlem Birimi Yapısı



Şekil 3.10. Tasarlanan İşlem Birimi Yapısı.

BÖLÜM 4. GERİYANSITMA AĞ MODELİ

Bölüm 2.7.11 'de kısaca anlatılan geri yansıtma ağının matematiksel açıklaması, öğrenme ve geri yansıtma algoritması ve ağın işlem basamakları detaylı bir şekilde sunulacaktır.

4.1 Genelleştirilmiş DELTA Kuralı

Bölüm 2.5.2.1 'de anlatılan En Küçük Kareler Yöntemi veya DELTA öğrenme kuralının genelleştirilmiş şekli geri yansıtma öğrenme kuralıdır (Huang 1990). Şekil 4.1 bir geri yansıtma ağını göstermektedir. Katmanlardaki bütün birimler birbirine bağlanmıştır. Geri besleme ve bir katmandaki birimlere bağlanmadan öteki katmandaki birimlere atlama yapılmamıştır. Burada üç katman kullanılmasına rağmen saklı katman sayısı 1 'den fazla olabilir. Şekildeki bias birimleri θ_j^h ve θ_k^o isteğe bağlı olarak kullanılabilir. Bias birimlerinin ağırlık değerleri 1 olup, diğer ağırlıklar gibi net-giriş değerine etki ederler.

$(X_1, Y_1), (X_2, Y_2), \dots, (X_p, Y_p)$ gibi bir P vektör çiftleri seti bulunmaktadır. Y ile X arasında $Y = \phi(X) : X \in \mathbb{R}^N, Y \in \mathbb{R}^M$ gibi bir bağıntı bulunmaktadır. Bir yapay sinir ağında öğrenme, bir giriş vektörü seti için uygun ağırlık setinin, ağ tarafından öğrenilerek bulunmasıdır. Burada açıklanan öğrenme tekniği; bilinen noktalar sayısına göre, bir eğrinin en iyi şekilde ifade edildiği eşitliğin bulunması problemine benzetilebilir. Böyle bir problemin çözümünde küçük kareler yöntemi yaklaşımı da kullanılabilir. Çünkü, burada yapılmak istenen şey lineer olmayan bir değişimin benzetilmesi başka bir deyişle girişlere en uygun yanıtı verecek değerlerin elde edilmesidir. Bu yüzden Şekil 4.1'deki ağ için küçük kareler yönteminin iteratif versyonu olan en dik eğim metodu uygulanacaktır.

P vektör setinin bileşeni olan $(X_p)^t$ bir giriş vektörü olarak ağın giriş katmanına uygulanmıştır. Giriş birimleri, her bir giriş değerini saklı katman birimlerine dağıtarak bir tampon görevi görmektedir. J . saklı birimine gelen toplam giriş:

$$\text{net}_{pj}^h = \sum_{i=1}^N W_{ji}^h X_{pi} + \theta_j^h \quad (4.1)$$

olacaktır. Burada W_{ji}^h , i.giriş biriminden gelen ağırlık ve θ_j^h ise bias ağırlığıdır. "h" indisi saklı katmanı ifade etmektedir. Bu birimin aktif edilebilmesi ve bir çıkış üretebilmesi için :

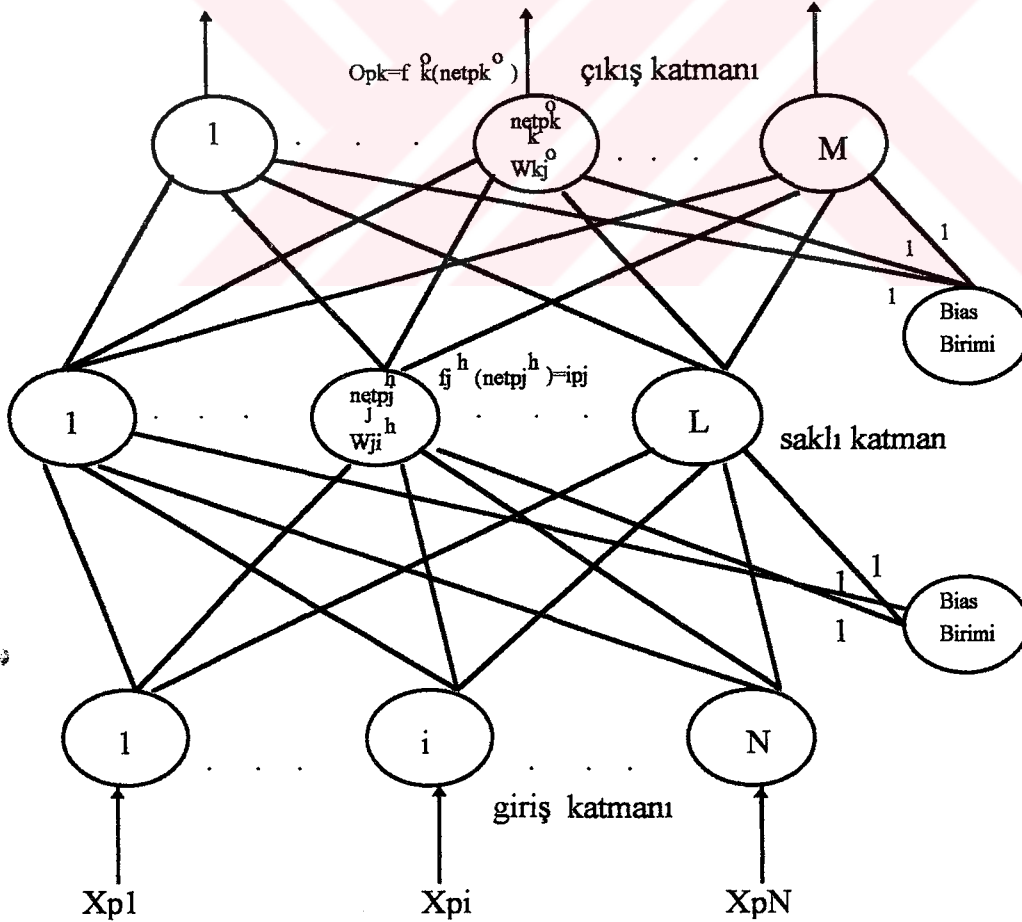
$$i_{pj} = f_j^h(\text{net}_{pj}^h) \quad (4.2)$$

olmalıdır. Çıkış düğümü için toplam giriş ve çıkış eşitliği:

$$\text{net}_{pk}^o = \sum_{j=1}^L W_{kj}^o i_{pj} + \theta_k^o \quad (4.3)$$

$$o_{pk} = f_k^o(\text{net}_{pk}^o) \quad (4.4)$$

olacaktır. Burada "o" indisi çıkış katmanını ifade etmektedir.



Şekil 4.1. Üç katmanlı Geriyansıtma Ağı.

Yapay sinir ağının eğitimi için gereken işlem sırası aşağıdaki gibidir:

1. Bir giriş vektörü ağa uygulanarak ilgili çıkış değeri hesaplanır.
2. Gerçek çıkış değeri ile istenen çıkış değeri karşılaştırılarak bir hata değeri elde edilir
3. Hatanın azaltılması için ağırlıkların hangi yönde (+veya -) olması gerektiği belirlenir.
4. Her bir ağırlığın ne kadar değiştirileceği belirlenir.
5. Ağırlık değişimleri ilgili ağırlıklara uygulanır.
6. 1'den 5'e kadar olan işlem basamakları bütün giriş vektörleri ile eğitim setindeki bütün vektörler için, bir hata yaklaşımına göre, kabul edilebilir bir değere indirgeninceye kadar tekrarlanır.

Küçük kareler yöntemi veya Delta kuralına göre; saklı birimler ve lineer çıkış birimleri olan bir ağ için ağırlık değişim kuralı:

$$W(t+i)_i = W(t)_i + 2\mu\epsilon_k X_{ki} \quad (4.5)$$

olarak verilmiştir. Burada μ pozitif bir sabittir. X_{ki} , k. eğitim vektörünün i. bileşeni ve ϵ_k ise istenen çıkış ile gerçek çıkış arasındaki farktır: $\epsilon_k = (d_k - y_k)$. Ağın ikiden fazla katmanı olduğunda veya çıkış fonksiyonları lineer olmadığında da benzeri bir eşitlik oluşacaktır.

4.1.1 Çıkış Katmanı Ağırlıklarının Yenilenmesi

Delta kuralının türetilmesinde, k. giriş vektörü için hata $\epsilon_k = (d_k - y_k)$ olup istenen çıkış değeri d_k ve gerçek çıkış değeri y_k 'dir. Şekil 4.1'deki notasyona uygun olması ve birden fazla katmanımız olduğu için hatayı $\delta_{pk} = (y_{pk} - o_{pk})$ olarak göstereceğiz. Burada "p" indisi p. eğitim vektörünü ve "k" indisi ise k. çıkış birimini göstermektedir. Bu durumda, k. birimdeki y_{pk} istenen çıkış değeri ve o_{pk} ise gerçek çıkış değeri olacaktır. Genelleştirilmiş delta kuralı tarafından minimize edilen hata bütün çıkış birimleri için:

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (4.6)$$

şeklinde olacaktır. Denklem 4.6'daki 0.5 çarpım faktörü, daha sonra hesaplanacak türevlerde uygunluk sağlaması için kullanılmıştır. Bu faktör sadece bir sabit olduğu için türevlerin alınmasında olumsuz bir durum yaratmayacaktır.

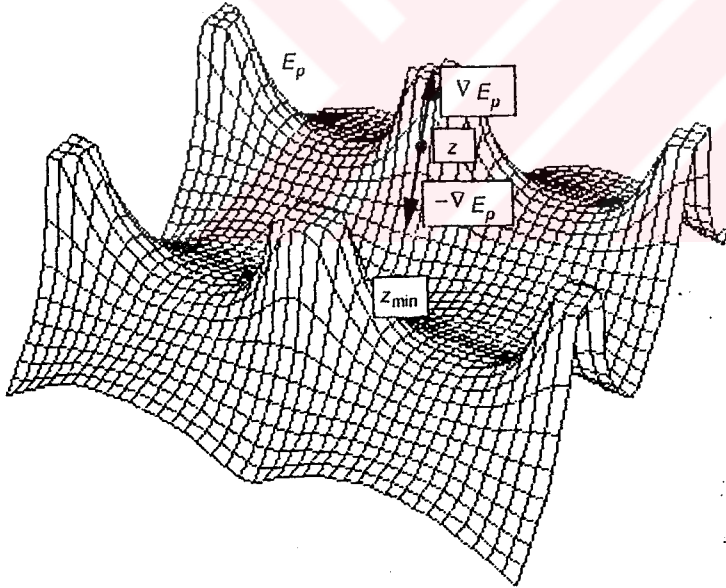
Ağırlıkların hangi yönde değişeceğini belirlemek için, W_{kj} ağırlıklarına bağlı olarak E_p 'nin negatif gradienti ∇E_p hesaplanacaktır. Bundan sonra toplam hatanın indirgenmesi için ağırlık değerleri ayarlanabilir. Şekil 4.2 'de görüldüğü gibi E_p 'yi ağırlık alanındaki bir yüzey gibi düşünebiliriz .

Eşitlik 4.6 ve δ_{pk} 'dan; E_p ve E_p 'nin W_{kj}^0 'ya göre türevi

$$E_p = \frac{1}{2} \sum_k (y_{pk} - o_{pk})^2 \quad (4.7)$$

$$\frac{\partial E_p}{\partial W_{kj}^0} = -(y_{pk} - o_{pk}) \frac{\partial f_k^o}{\partial (net_{pk}^o)} \frac{\partial (net_{pk}^o)}{\partial W_{kj}^0} \quad (4.8)$$

olacaktır.



Şekil 4.2. Ağırlık uzayında varsayımaya dayalı bir yüzey.

Şekil 4.2 'de z noktası boyunca minimuma ulaşılması için gradient ∇E_p 'nin negatifi alınmalıdır. Ağırlık değişimi; yüzeyin z noktasındaki en dik eğim yönü ile aynı olan negatif gradient yönünde olmalıdır. Başka bir deyişle E_p minimum noktaya (z_{min}) ulaşincaya kadar ağırlıklar değiştirilmelidir.

Eşitlik 4.8 deki net_{pk}^0 'daki W_{kj}^0 'ya göre değişimi yani türevi Eşitlik 4.2 'yi; i_{pj} 'yi verecektir. Bu durum formüle edilirse;

$$\frac{\partial (net_{pk}^0)}{\partial W_{kj}^0} = \left(\frac{\partial}{\partial W_{kj}^0} \sum_{j=1}^L W_{kj}^0 i_{pj} + \theta_k^0 \right) = i_{pj} \quad (4.9)$$

$$\frac{\partial f_k^0}{\partial (net_{pk}^0)} = f_k^{\prime 0} (net_{pk}^0) \quad (4.10)$$

şeklindedir.

Ayrıca $f_k^0(net_{pk}^0)$ 'nun net_{pk}^0 'ya göre değişimi $f_k^{\prime 0}(net_{pk}^0)$ 'i verecektir. Bu durum, Eşitlik 4.8, 4.9 ve 4.10 negatif gradient ifadesi için birleştirilirse;

$$-\frac{\partial E_p}{\partial W_{kj}^0} = (y_{pk} - o_{pk}) f_k^{\prime 0} (net_{pk}^0) i_{pj} \quad (4.11)$$

Ağırlık değişiminin büyüklüğüne göre ağırlıklar yenileneceğinden, ağırlık değişimini negatif gradiente oranlayarak bu değişimin ne kadar olacağı bulunabilir. Buna göre, çıkış katmanındaki ağırlıklar aşağıdaki eşitliğe göre yenilenir:

$$W_{kj}^0(t+1) = W_{kj}^0(t) + \Delta_p W_{kj}^0(t) \quad (4.12)$$

Burada $\Delta_p W_{kj}^0$ aşağıdaki değere eşittir.

$$\Delta_p W_{kj}^0(t) = \eta (y_{pk} - o_{pk}) f_k^{\prime 0} (net_{pk}^0) i_{pj} \quad (4.13)$$

Eşitlik 4.13 'deki η -eta öğrenme oranı parametresidir. Eta pozitif ve 1 'den küçük bir sayıdır. Eşitlik 4.13'deki $f_k^{\prime 0}(net_{pk}^0)$ ifadesi çıkış birimlerindeki transfer fonksiyonlarının türevini ifade etmektedir. Önceki bölümlerde incelenen transfer fonksiyonlarına göre bir birimin transfer fonksiyonu; lineer ya da sigmoid olabilir. Sigmoid transfer fonksiyonlarının lineer olmayan değişimlerin temsilde çok kullanışlı olduğu aşikardır (Jones 1990). Burada lineer çıkış için $f_k^0(net_{pk}^0) = (net_{pk}^0)$ olacaktır. Sigmoid çıkış fonksiyonu ise; $f_k^0(net_{pk}^0) = (1 + e^{-net_{pk}^0})^{-1}$ olacaktır.

Transfer fonksiyonu başka bir deyişle çıkış fonksiyonunun seçimi; çıkış verisinin nasıl seçileceğine bağlıdır. Eğer çıkışın binari olması isteniyorsa, sigmoid çıkış fonksiyonu kullanılmalıdır.

Transfer fonksiyonlarının ilk durumu için, $f_k^{o'}=1$, ikinci durumda ise $f_k^{o'} = f_k^o(1-f_k^o)=o_{pk}(1-o_{pk})$ olacaktır. Bu iki durum için ağırlık yenileme işlemleri sırası ile;

$$W_{kj}^o(t+1) = W_{kj}^o(t) + \eta (y_{pk} - o_{pk})^i_{pj} \quad (4.14)$$

$$W_{kj}^o(t+1) = W_{kj}^o(t) + \eta (y_{pk} - o_{pk})^o_{pk} (1 - o_{pk})^i_{pj} \quad (4.15)$$

olacaktır. Ağırlık değişimleri için bir büyüklük tanımlanırsa;

$$\delta_{pk}^o = (y_{pk} - o_{pk}) f_k^{o'}(net_{pk}^o) = \delta_{pk} f_k^{o'}(net_{pk}^o) \quad (4.16)$$

olacaktır. Eşitlik 4.16 'ya göre ağırlık yenileme eşitliği şu şekilde yazılabilir:

$$W_{kj}^o(t+1) = W_{kj}^o(t) + \eta \delta_{pk}^o i_{pj} \quad (4.17)$$

Gradient eğimi metodu ve küçük kareler yöntemi arasında ilişki kurarak açıklama yapmakta fayda vardır. Eğer genelleştirilmiş delta kuralı küçük kareler yöntemine benzetilirse; gerçekte hiç bir ağırlık değeri bütün paternler ağa bir kez sunulmadan değiştirilmemesi gerekir. Yalnızca bu değişiklikleri sunulan her patern için biriktirip, toplamını aldıktan sonra, ağırlıkların değeri yenilenebilir. Daha sonra bu işlem, hatanın kabul edilen düşük değerine kadar tekrarlanabilir. Bu işleme göre minimize edilen hata;

$$E = \sum_{p=1}^P E_p \quad (4.18)$$

şeklinde gösterilebilir. Burada P eğitim setindeki paternlerin sayısıdır.

Bu yöntemin bir dezavantajı; geniş bilgi saklanacağından, işlemler uzayabilecek ve bilgisayarda fazla yer kaplanmış olacaktır. Bu nedenle ağırlıkların yenilenmesi işlemi her bir eğitim paterni; ağa sunulduğunda yapılması önerilen bir yöntemdir.

4.1.2. Saklı Katman Ağırlıklarının Yenilenmesi

Saklı katman ağırlıklarının yenilenmesinde, çıkış katmanı için yapılan hesabın bir benzeri yapılacaktır. Yalnız saklı katman birimlerinin çıkışlarındaki hatanın bulunması gerekmektedir. Gerçek çıkış değeri bilinmektedir fakat birimlerin çıkışlarının ne olması gerektiğini bilmek için hiç bir yol görünmemektedir. Toplam hata, E_p 'nin bir şekilde saklı katman çıkış değerleri ile bir bağıntısı olmalıdır. Eşitlik 4.7 ifadesi tekrar hatırlanırsa:

$$E_p = \frac{1}{2} \sum_k (y_{pk} - o_{pk})^2 = \frac{1}{2} \sum_k (y_{pk} - f_k^o(\text{net}_{pk}^o))^2 = \frac{1}{2} \sum_k (y_{pk} - f_k^o(\sum_j W_{kj}^o i_{pj} + \theta_k^o))^2 \quad (4.19)$$

Eşitlik 4.1 ve 4.2 'den i_{pj} 'nin saklı katmanlardaki ağırlıklara bağlı olduğu eşitlik bilinmektedir. Aynı etki E_p 'nin gradientinin, saklı katmanların ağırlıklarına bağlı olarak hesaplanabilir.

$$\frac{\partial E_p}{\partial W_{ji}^h} = \frac{1}{2} \sum_k \frac{\partial}{\partial W_{ji}^h} (y_{pk} - o_{pk})^2 = - \sum_k (y_{pk} - o_{pk}) \frac{\partial o_{pk}}{\partial (\text{net}_{pk}^o)} \frac{\partial (\text{net}_{pk}^o)}{\partial i_{pj}} \frac{\partial i_{pj}}{\partial (\text{net}_{pj}^h)} \frac{\partial (\text{net}_{pj}^h)}{\partial W_{ji}^h} \quad (4.20)$$

Eşitlik 4.20 'deki her bir faktör bir önceki eşitliklerden hesaplanabilir. Sonuçta;

$$\frac{\partial E_p}{\partial W_{ji}^h} = - \sum_k (y_{pk} - o_{pk}) f_k^{o'}(\text{net}_{pk}^o) W_{kj}^o f_j^{h'}(\text{net}_{pj}^h) X_{pi} \quad (4.21)$$

olur. Saklı katman ağırlıkları Eşitlik 4.21 'in negatif değeri oranında yenilenir:

$$\Delta W_{ji}^h = \eta f_j^{h'}(\text{net}_{pj}^h) X_{pi} \sum_k (y_{pk} - o_{pk}) f_k^{o'}(\text{net}_{pk}^o) W_{kj}^o \quad (4.22)$$

Burada η yine öğrenme oranını göstermektedir. δ_{pk}^o 'yu kullanarak Eşitlik 4.22 tekrar yazılırsa;

$$\Delta W_{ji}^h = \eta f_j^{h'}(\text{net}_{pj}^h) X_{pi} \sum_k \delta_{pk}^o W_{kj}^o \quad (4.23)$$

Dikkat edilirse saklı katmandaki her ağırlık yenileme, çıkış katmanındaki hata ifadesi başka bir deyişle ağırlık-yenileme terimi δ_{pk}^o 'ya bağlıdır. Çıkış katmanındaki hataların saklı katmandaki birimlere uygun ağırlık değişimleri oranında geriye

yansıtılmasından dolayı, geriye yansıtma ağı adını almaktadır. Saklı katman hata ifadesinin tanımı şu şekildedir.

$$\delta_{pj}^h = f_j^{h'} (net_{pj}^h) \sum_k \delta_{pk}^o W_{kj}^o \quad (4.24)$$

Çıkış katmanı için yapılan ağırlık yenileme eşitliği saklı katman için de yazılırsa:

$$W_{ji}^h(t+1) = W_{ji}^h(t) + \eta \delta_{pj}^h X_i \quad (4.25)$$

4.2. Adaptif Parametreler

Bu bölümde, bir yapay sinir ağının eğitiminde ve işleyişinde önemli rol oynayan parametrelerin tanımı verilmiştir. Fakat bu adaptif parametrelerin ağ içinde nasıl bir rol oynadığı açıklanmalıdır. Bu düşünce ile yola çıkarak bu parametrelerin etkileri aşağıdaki gibi ifade edilebilir.

η -**eta**; öğrenme oranı parametresi olup, genellikle küçük bir sayı olmalıdır. Eta 'nın küçük bir değerde olması demek, ağın daha fazla iterasyon yapacağını gösterir. Öğrenme ilerledikçe eta 'nın değerini büyültmek mümkündür. Ağın maliyet fonksiyonu ya da hata oranı düşerken eta 'yı büyültülerek hatanın minimuma daha çabuk ulaşması yani yakınsamanın daha hızlı olması mümkün kılınmış olacaktır. Bu işlem yapılırken eğer eta çok büyük bir değer almışsa, ağın gerçek minimum değere yani global minimum değerine ulaşamayacağı da unutulmamalıdır. Bu nedenle en iyi çözüm eta 'nın sabit bir şekilde artırılmasıdır, fakat çözüm istendiği gibi gelişmiyorsa anında düşürülmesi de gereklidir. Buna göre eta değişimi:

$$\begin{aligned} & +a \quad \text{Eğer } \Delta E < 0; \\ \Delta \eta & = -bh \quad \text{Eğer } \Delta E > 0; \\ & 0 \quad \text{diğer durumlarda} \end{aligned} \quad (4.26)$$

şeklinde. Burada ΔE maliyet fonksiyonu değişimi, a ve b ise uygun sabitleri göstermektedir. Kötü bir adım eta 'yı düşürürse, iyi bir adım alınmaya kadar momentum α 'nın sifıra eşit alınmasında yarar vardır.

Yakınsama hızını arttırmak için bir başka teknik de momentum (α) 'dur. Ağırlık değişimi $\Delta_p W$ değeri hesaplanırken, bir önceki değişimi gösteren bir eklenebilir. Bu eklenen momentum ifadesi ağırlık değişiminin aynı yönde gitme eğilimini koruması

için kullanılır. Bir bakıma osilasyona uğramadan istenen noktaya yakınsamayı sağlar. Buna göre ağırlık değişim eşitliği genel olarak şu şekilde ifade edilebilir:

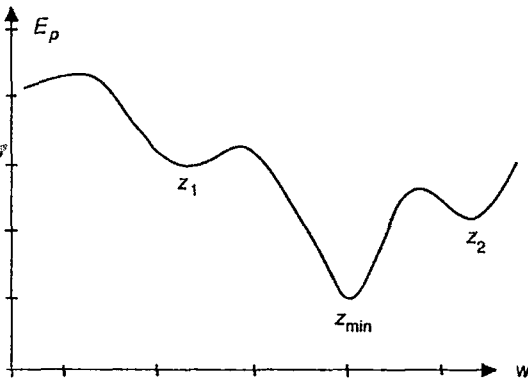
$$\Delta W_{pq}(t+1) = -\eta \frac{\partial E}{\partial W_{pq}} + \alpha \Delta W_{pq}(t) \quad (4.27)$$

Momentum parametresi α 0 ile 1 arasında olmalıdır, bu değer çoğunlukla 0.9 seçilmektedir. Eğer maliyet fonksiyonu yüzeyinin bulunduğu bölgeye ilerledikçe, her bir t anında $\partial E / \partial W_{pq}$ hemen hemen aynı değerde olacaktır. Buna göre Eşitlik 4.27 Eşitlik 4.28'deki değere yakınsayacaktır:

$$\Delta W_{pq} \approx -\frac{\eta}{1-\alpha} \frac{\partial E}{\partial W_{pq}} \quad (4.28)$$

Eşitlik 4.28'de etkin öğrenme oranı $\eta/(1-\alpha)$ 'dır. Osilasyon durumunda, ΔW_{pq} yalnızca η katsayısına $\partial E / \partial W_{pq}$ anlık dalgalanmalarında yanıt verir.

Ağırlık uzayında lokal minimuma yakınsama olasılığı fikrini Şekil 4.3. açıklamaktadır. Ağ bir kere bir minimuma (lokal veya global) ulaştığında, öğrenme durur. Eğer ağ lokal bir minimuma ulaşmışsa, hata; ağın çıkışında kabul edilmeyecek kadar büyük olabilir. Bir yapay sinir ağında; öğrenme işlemi kabul edilebilir bir değerden önce duruyorsa, saklı katmandaki işlem birimlerinin sayısı veya öğrenme parametreleri değiştirilerek problem giderilebilir. Ağ kabul edilebilir bir çözüme ulaştıncaya, çözümün global minimuma ulaştığı garanti edilemez. Eğer çözüm bir hata noktasına göre kabul edilebilir ise; bu minimum noktanın global veya lokal olmasında, hatta doğru minimuma ulaşmadan eğitim işleminin durmasında sorun yoktur.



Şekil 4.3. Ağırlık uzayında varsayılan bir hata yüzeyi.

Şekil 4.3. bir ağırlık uzayında varsayılan hata yüzeyini göstermektedir. Z_{\min} noktasına global minimum denir. Dikkat edilirse burada z_1 ve z_2 gibi diğer minimum noktalar da bulunmaktadır. Gradient eğimi metodu böyle bir global minimum noktayı ararken yanlışlıkla şekildeki z_1 veya z_2 gibi lokal minimum noktalardan birini de bulabilir.

4.2.1 Minimum Değeri Bulmak İçin Yapılabilecek Diğer İşlemler

Ağırlık uzayında bir X vektörü bütün ağırlıkları temsil etmektedir. Buna göre X_0 noktasında $E(x)$ maliyet fonksiyonu yazılırsa:

$$E(x) = E_0 + (x - x_0) \cdot \nabla E(x_0) + 1/2 (x - x_0) \cdot H(x - x_0) + \dots \quad (4.29)$$

Burada H Hessian matrisinin ikinci türevidir. Hessian matrisi Eşitlik 4.30'daki gibi ifade edilebilir.

$$H_{ij} = \frac{\partial^2 E}{\partial X_i \partial X_j} \quad (4.30)$$

X_0 noktasında Eşitlik 4.29'un gradienti türetilirse;

$$\nabla E(x) = \nabla E(x_0) + H(x - x_0) + \dots \quad (4.31)$$

şeklinde olur. Eşitlik 4.31'i sıfıra eşitleyerek ve yüksek dereceli terimleri ihmal ederek, $E(x)$ 'in minimum değerini $\nabla E(x) = 0$ 'da bulmak mümkündür.

$$\nabla E(x_0) + H(x - x_0) = 0 \quad (4.32)$$

şeklinde olacaktır veya

$$x = x_0 - H^{-1} \nabla E(x_0) \quad (4.33)$$

şeklinde olur.

Eşitlik 4.33; iterasyonda tahmin edilen x değeri yerine x_0 değeri olarak kullanılabilir. Bu işleme Newton Metodu denir ve bir boyutlu ifadelerde aranan bir kuraldır. $E'(x) = 0$ kökünü bulmak için Eşitlik 4.34 kullanılır;

$$x = x_0 - \frac{E'(x_0)}{E''(x_0)} \quad (4.34)$$

Fakat bu metod hesaplamalarda çok büyük zorluklar yaratmaktadır. Çünkü, n ağırlık değeri için $n \times n$ H matrisi her iterasyon için hazırlanmalıdır. Bu da her iterasyon için n^3 adım demektir. Newton metodunda ikinci dereceden türevlerin hesaplanması da gerekmektedir. Fakat bu durum geri yansıtma ağında elverişli değildir. Ayrıca bu metod, yeterli bir minimum noktaya ulaşmadıkça sayısal olarak kararsız bir yapı gösterir. Bu yüzden pratik bir teknik değildir, fakat diğer yaklaşımlara iyi bir referans olmaktadır.

Bir çok pratik minimize metodları çizgi araştırmalarında seçilen bir yön boyunca yalnızca birinci dereceden türev bilgilerini kullanmaktadırlar. X_0 noktasından başlayıp d yönündeki bir çizgi (hat) araştırmasında;

$$X = X_0 + \lambda d \quad (4.35)$$

gibi bir başka noktaya ulaşabiliriz. Burada λ ; $E(X)$ 'i minimize etmek için seçilmiştir. Bu bir boyutlu minimizasyonun bir çok yolla yapılması mümkündür. Bir basit yaklaşım da; sabit η adımıyla E 'nin yeteri kadar düşürülmesidir (yani belli adımlardan sonra değişmemesidir).

Diğer bir yaklaşım çizgi (hat) araştırması kullanılarak d 'nin $d = -\nabla E(X_0)$ seçilmesi ile en dik eğim (steepest descent) metodudur.

En dik eğim metodu diğer metodlara göre sıradan bir geri yansıtma sisteminde genellikle daha hızlı yakınsar. Bu metodda ard arda gelen adımlar birbirlerine diktirler.

$$0 = \frac{\partial}{\partial \lambda} E(x_0 + \lambda d^{\text{eski}}) = d^{\text{eski}} \cdot \nabla E^{\text{yeni}} \quad (4.36)$$

Eşitlik 4.36 'da çizgi (hat) minimizasyonundan sonra ∇E^{yeni} , yeni gradient yönü, deski eski d yönüne diktir. Daha iyi bir yaklaşım ise gradient yönü ile bir önceki araştırma yönü arasında ortak (uzlaşıcı) bir yeni araştırma yönünü kullanmaktadır. Uygun β için d^{yeni} değeri Eşitlik 4.37 'deki gibidir.

$$d^{\text{yeni}} = -\nabla E^{\text{yeni}} + \beta d^{\text{eski}} \quad (4.37)$$

Eşitlik 4.37, gradient metodu temelli olup bir β değeri seçilmesiyle yeni bir araştırma yönü az da olsa minimizasyon için bir önceki değerinden uzaklaşmaktadır. Yeni araştırma yönü dyeni, bir önceki yön boyunca gradient bileşenini değiştirmemelidir. Böylece birinci dereceden λ ;

$$d^{eski} \cdot \nabla E(x_0 + \lambda d^{yeni}) = 0 \quad (4.38)$$

olacaktır. Veya Eşitlik 4.31 ve 4.36 'yı kullanarak

$$d^{eski} \cdot H \cdot d^{yeni} = 0 \quad (4.39)$$

şeklinde yazılabilir.

Pratik bir çözüm için β 'yı Eşitlik 4.40'da nasıl seçeceğimiz Eşitlik 4.39'u desteklemesi için önemlidir. Polak-Ribiere Kuralına göre β ;

$$\beta = \frac{(\nabla E^{yeni} - \nabla E^{eski}) \cdot \nabla E^{yeni}}{(\nabla E^{yeni})^2} \quad (4.40)$$

şeklinindedir. $\nabla E^{eski} \cdot \nabla E^{yeni}$ ifadesi çizgi (hat) minimizasyonu tam olarak yapılmışsa sıfırdır. Fakat sadece yaklaşık olarak hesaplandığında yardımcı bir rol oynamaktadır.

Bir başka teknik de ard arda çizgi (hat) minimizasyonlarının kullanılması olup yarı (veya hemen hemen) Newton veya değişken metrik metodu diye adlandırılır. Bu metod farklı bir yaklaşımla n adım sonucu ikinci dereceli minimuma ulaşır. Bu fikir Eşitlik 4.33'teki Newton kuralında Hessian tersinin (H^{-1}) kullanılmasıyla ortaya çıkar. Bu durum Eşitlik 4.41 ve 4.42'de açıklanmaktadır.

$$x^{yeni} = x^{eski} - \lambda G^{eski} \nabla E^{eski} \quad (4.41)$$

$$G^{yeni} = G^{eski} - F(G^{eski}, x^{yeni} - x^{eski}, \nabla E^{yeni} - \nabla E^{eski}) \quad (4.42)$$

Eşitlik 4.42'de G , H^{-1} matrisine bir yaklaşımdır ve Matris F ; eşitlikte gösterilen üç terimden daha karmaşık bir fonksiyondur. Eşitlik 4.41'de Newton kuralının, Hessian tersi değerine bir yaklaşım oluşturan G ve çizgi (hat) minimizasyon sonucunda seçilen λ gibi ekstra bir faktör kullanılmıştır. Eşitlik 4.42; gerçek Hessian matrisinin ikinci türevini değerlendirmeksizin bir yaklaşım yaratmaktadır.

Yarı-Newton tekniği genellikle Conjugate (çekmek) gradienti tekniğiyle aynı hızda karşılaştırmalı sonuçlar üretir. Bütün bu çizgi (hat) araştırma metodları: en dik eğim, conjugate gradient ve yarı-Newton sadece birinci türev bilgisini kullanır. Bir sinir ağı problemindeki türevler geriyanıtma dirac (δ) 'larıyla yeterli bir şekilde hesaplanabilir. Buna göre geriyanıtma kuralı için gradient hesabını değil, sadece gradient eğimi kuralı değiştirilmiştir:

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} \quad (4.43)$$

Bazı yazarlar ikinci türevlerin de kullanılmasını Tam Newton Kuralı (Eşitlik 4.33) yaklaşımı ile önermişlerdir. Bu yüzden matrisin tersi gerekmemektedir. En basit yaklaşımda; Hessian matrisinin off-diagonal elementleri göz ardı edilmiştir. Bu yaklaşıma takma-Newton kuralı denir (Nordita, 1991).

$$\Delta W_{ij} = -\frac{\partial E}{\partial W_{ij}} / \frac{\partial^2 E}{\partial W_{ij}^2} \quad (4.44)$$

Eşitlik (4.44) Newton Kuralını her ağırlık için ayrı ayrı yerine getirmektedir. Alternatif olarak ikinci türevler sonlu-fark yöntemiyle hesaplanabilir: W_{ij} 'deki küçük değişimden üretilen $\partial E/W_{ij}$ 'deki değişimin bulunmasıdır.

Eşitlik (4.44) 'de görüldüğü gibi bu yöntem biraz tehlikelidir. Çünkü paydanın negatif veya sıfıra yakın olması muhtelif olarak adımın yanlış yönde gitmesine neden olur. Bu problemler μ gibi küçük bir pozitif sabitin kullanılmasıyla giderilebilir. Bu durum Eşitlik 4.45' de gösterilmiştir (Nordita, 1991).

$$\Delta W_{ij} = -\frac{\partial E}{\partial W_{ij}} / \left(\left| \frac{\partial^2 E}{\partial W_{ij}^2} \right| + \mu \right) \quad (4.45)$$

BÖLÜM 5. SONUÇ ve ÖNERİLER

Bu bölümde, ITÜ TRİGA Mark-II reaktörünün kontrolü için önerilen yapay sınır ağı kontrolünün çalışmasını izlemek için yapılan simülasyon sonuçları verilecektir.

5.1. Sistem Çıkışı ALC ile bulunan Simülatör için Elde Edilen Sonuçlar

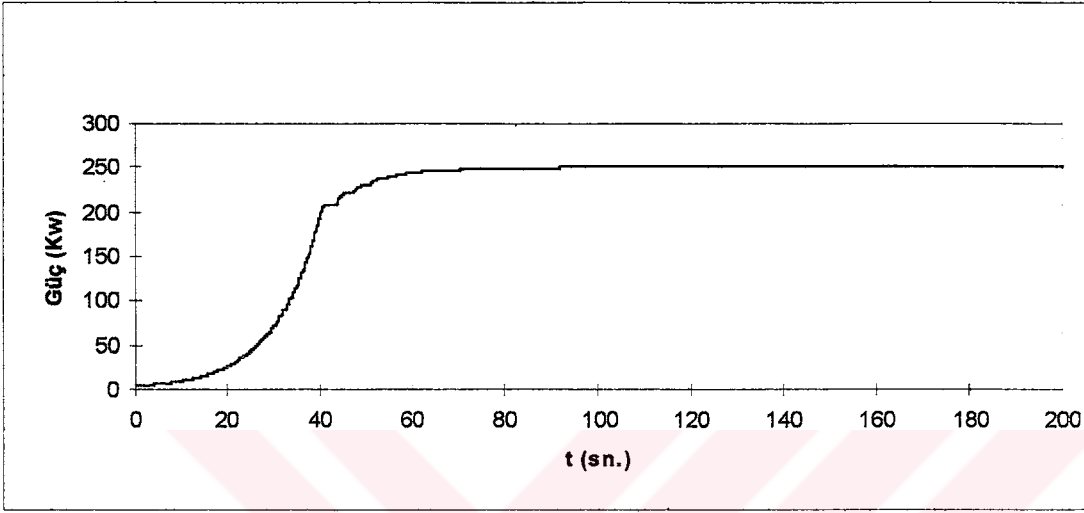
Çalışmamızda Tablo 5.1 'de verilen değerler kullanılarak elde edilen sonuçlar Şekil 5.1 'de görülmektedir.

Tablo 5.1 Simülasyon Başlangıç verileri:

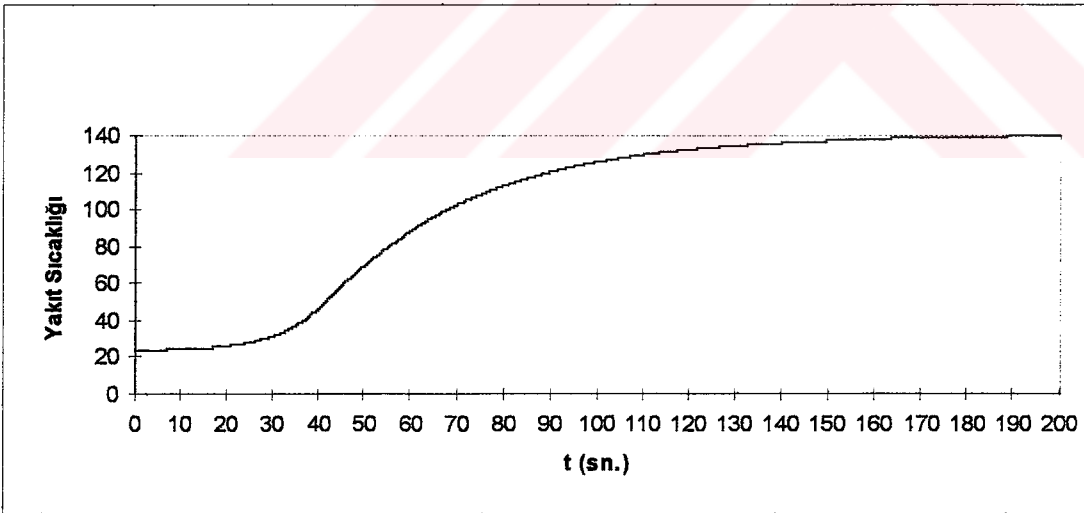
Başlangıç gücü: P_0	5 Kw
İstenen Peryot: τ	10 sn.
İstenen güç: P_s	250 Kw
Örnekleme aralığı: T	0.05 sn.
İnceleme aralığı: T_{max}	0-200 sn.

Reaktörün çıkış gücünün başlangıç kararlı hal değeri olan 5 Kw'dan 250 Kw'a çıkması istenmiştir. Bu amaçla program çalıştırılarak, kontrol çubuğu Şekil 5.1c 'de görüldüğü gibi başlangıçta maksimum hız ile çekilmektedir. Kontrol çubuğunun maksimum hızla çekilmesi sonucunda, gücün artarak yörüngeyi 6. sn.'de 100 Watt geçmesine neden olmaktadır. Bu sırada, kontrol çubuğu bir miktar aşağı indirilerek güç hatasının giderilmesi sağlanır. Daha sonra, yörüngedeki artışa bağlı olarak kontrol çubuğu düşük hızda çekilerek güç 50 sn.'de 250 Kw'a çıkarılmaktadır. Reaktör gücündeki artış yakıt sıcaklığının artmasına neden olmaktadır. Fakat sıcaklık güce göre daha geç artmaya başlar. Bu gecikme, maksimum değerlerin farklı yerlerde olmasına neden olur. Sıcaklık artışına bağlı olarak oluşan negatif reaktivite ise, gücü azaltıcı yönde etkilemektedir. Sıcaklık nedeniyle güçte meydana gelen düşüş kontrol çubuğunun çekilmesine devam edilerek, güç değerinin aynı seviyede korunmasını sağlar. Bu işlem sıcaklık artışı sona erene kadar devam ederek şekil 5.1d 'de

görüldüğü gibi kontrol sinyali sıfıra doğru inmeye başlayacaktır. Şekil 51.e'de görüldüğü gibi güç hatası (gücün yörüngeden sapması) sıfır olduğunda kontrol sinyali sıfır olarak çubuğun pozisyonunu sabit tutar. Bu sırada periyot sonsuz değerdedir. Şekil 5.1f 'de görüldüğü gibi ALC ağırlıkları 15. sn.'den sonra daha az değişmektedir.



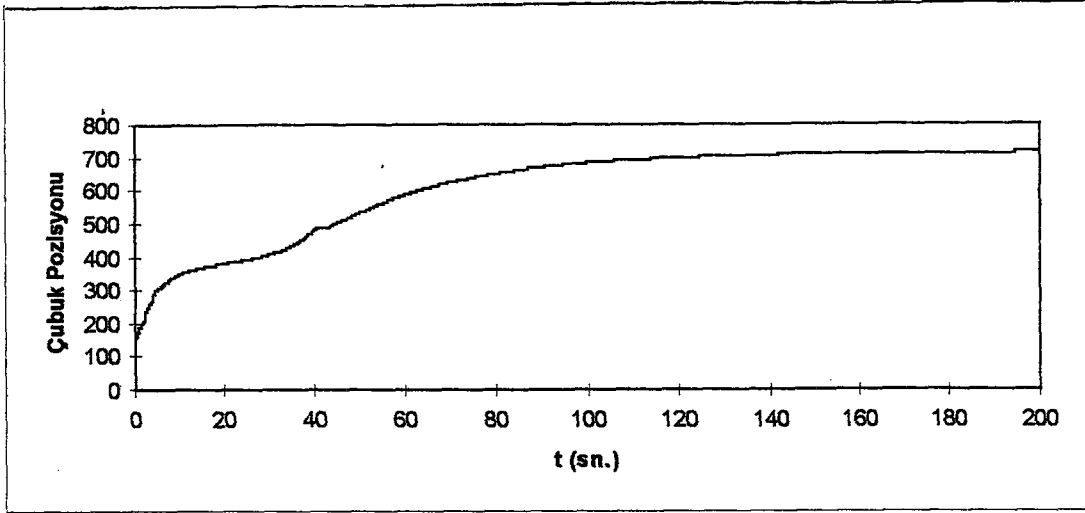
(a)



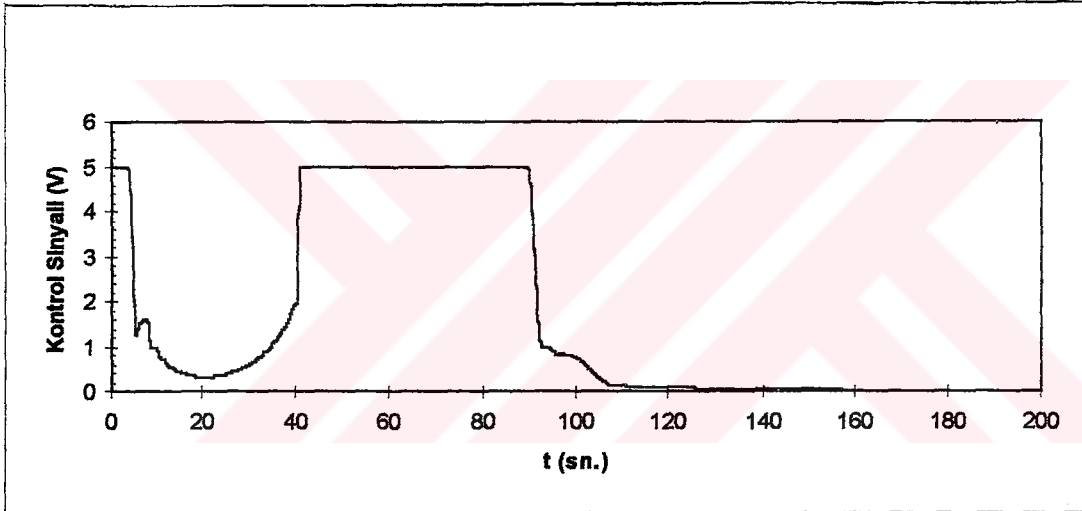
(b)

Şekil 5.1 a) Güç Değişimi

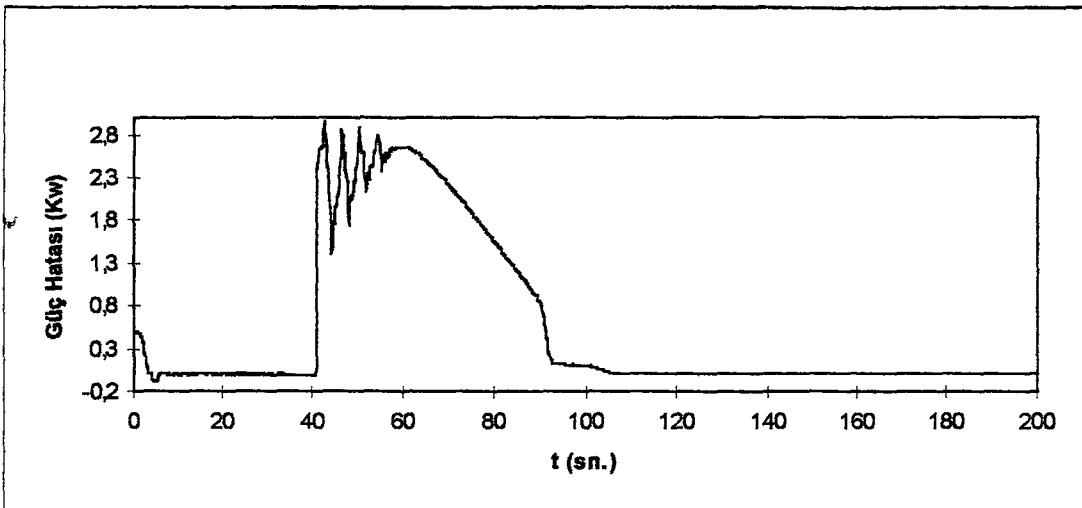
b) Yakıt Sıcaklığı (°C)



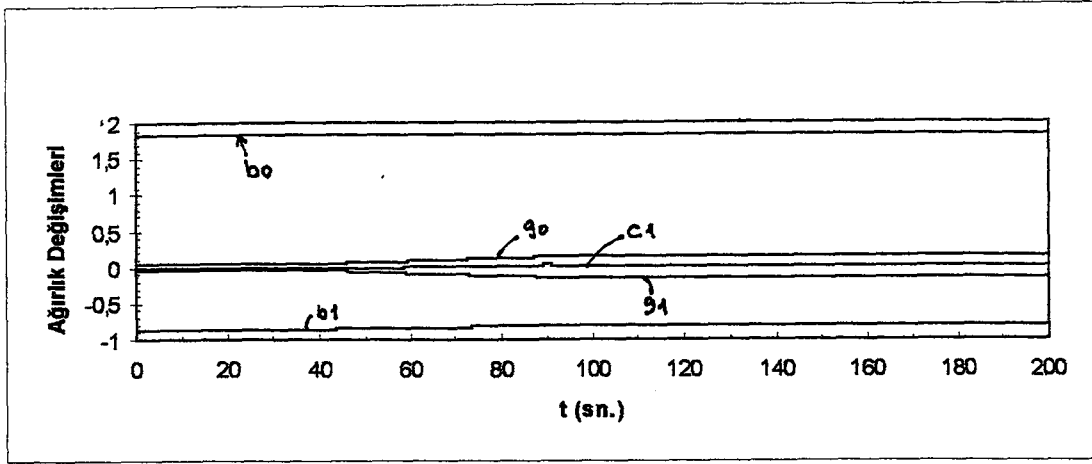
(c)



(d)



(e)



(f)

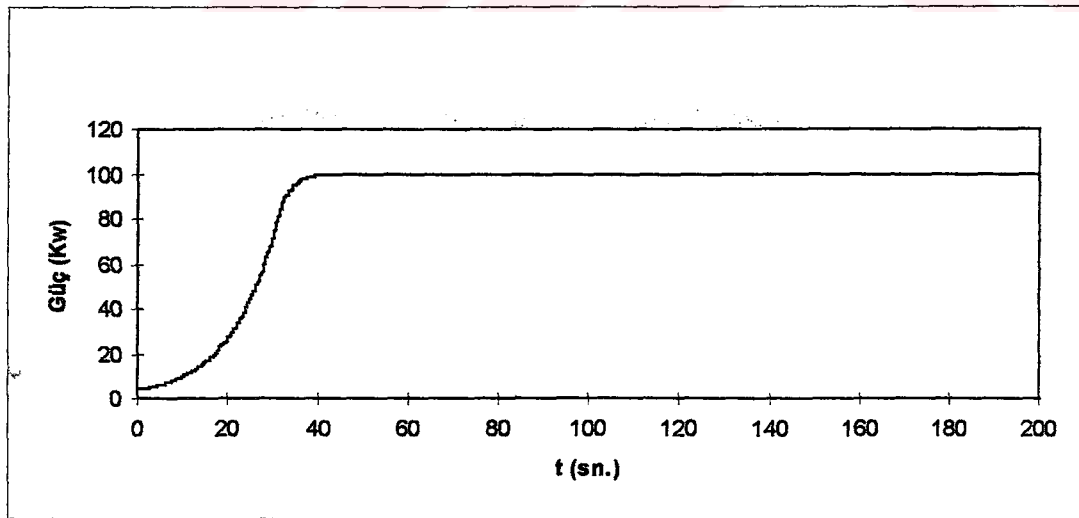
Şekil 5.1 c) Çubuk Pozisyonu

d) Gücün Yörüngeden Sapması

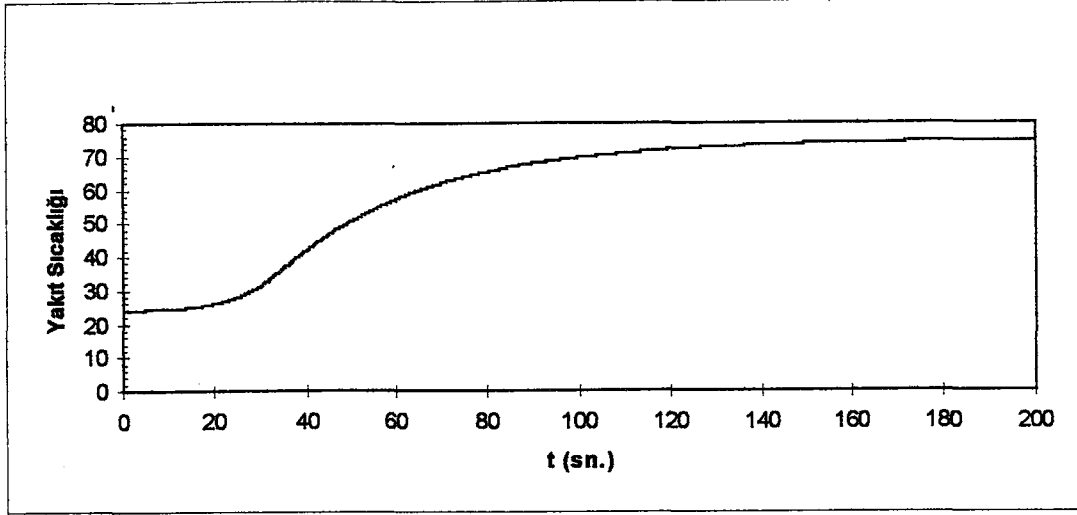
e) Kontrol Sinyali

f) Ağırlık Değişimleri

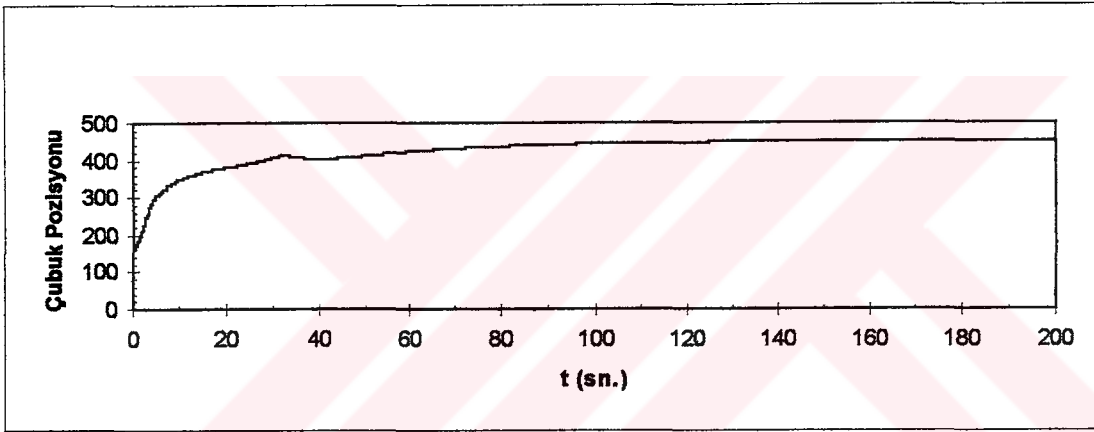
İkinci olarak reaktörün çıkış gücünün 5 Kw'dan 100 Kw'a çıkması istenmiştir. Bu amaçla program çalıştırılarak farklı maksimum çıkış gücüne karşılık sistemin cevabı gözlenmiştir. Bu çalışmaya ilişkin çizimler Şekil 5.2'de verilmiştir. Sonuçlar Şekil 5.1 'dekilerin benzeridir.



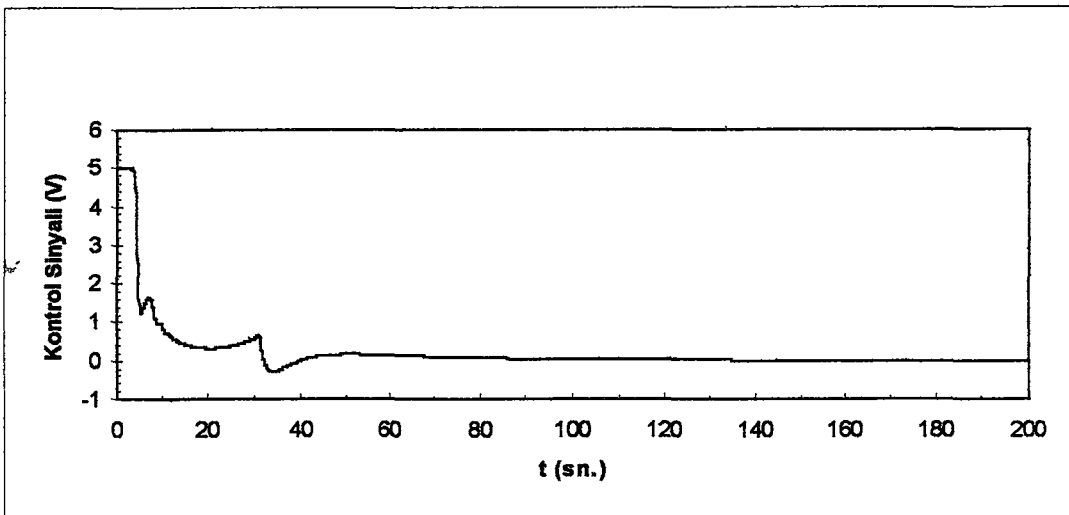
(a)



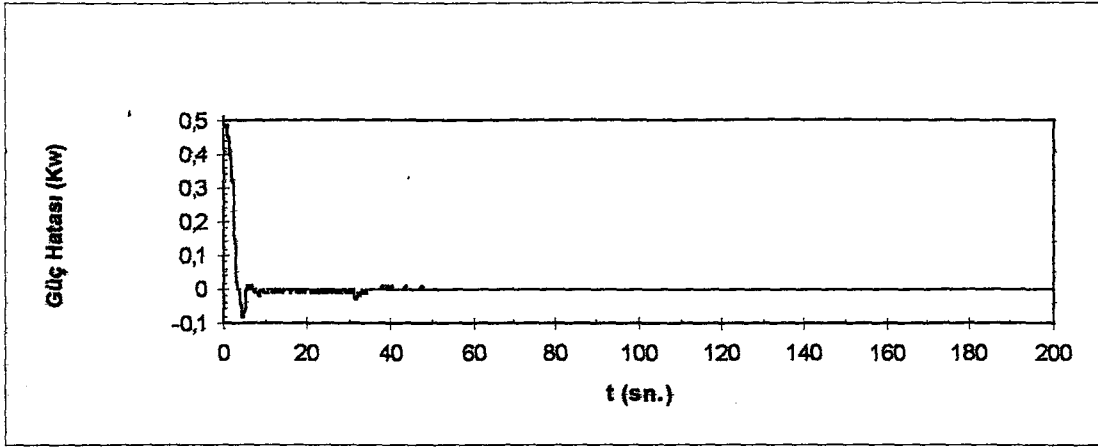
(b)



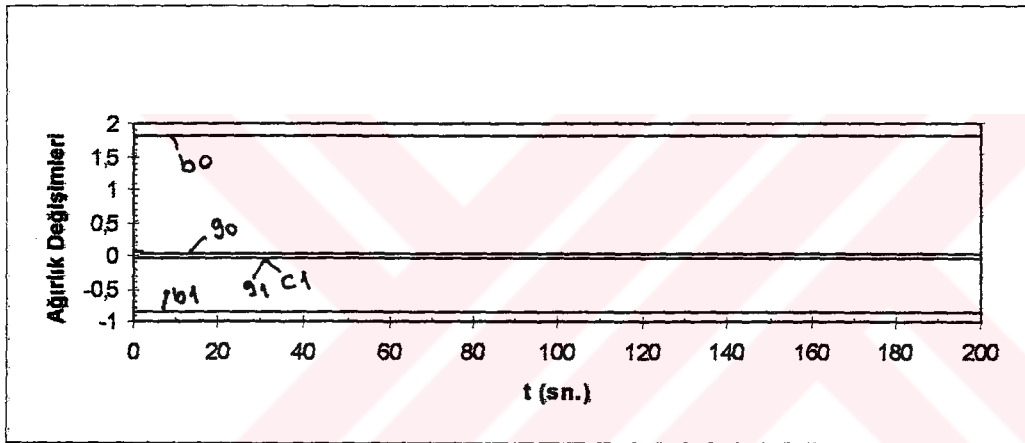
(c)



(d)



(e)



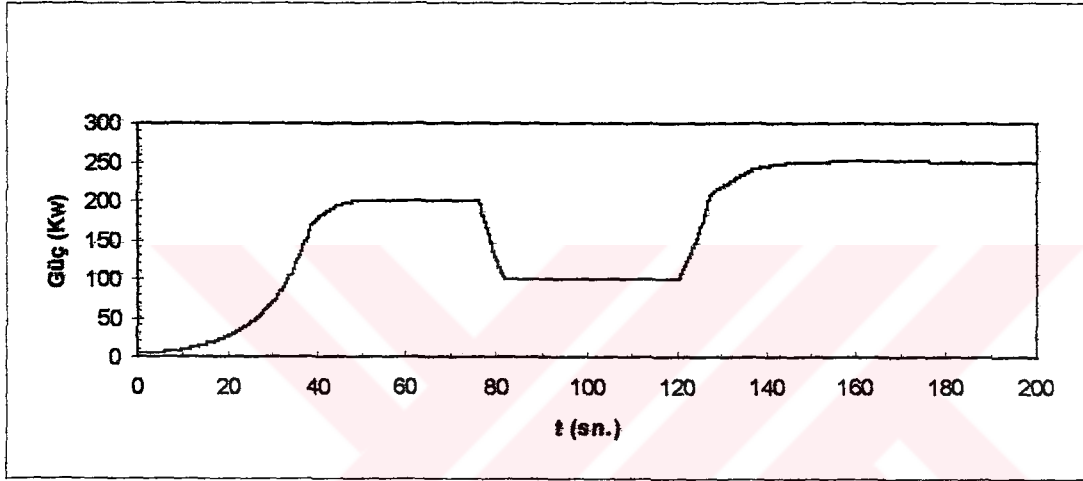
(f)

- Şekil 5.2 a) Güç Değişimi
 b) Yakıt Sıcaklığı
 c) Çubuk Pozisyonu
 d) Kontrol sinyali
 e) Gücün Yörüngeden Sapması
 f) Ağırlık Değişimleri.

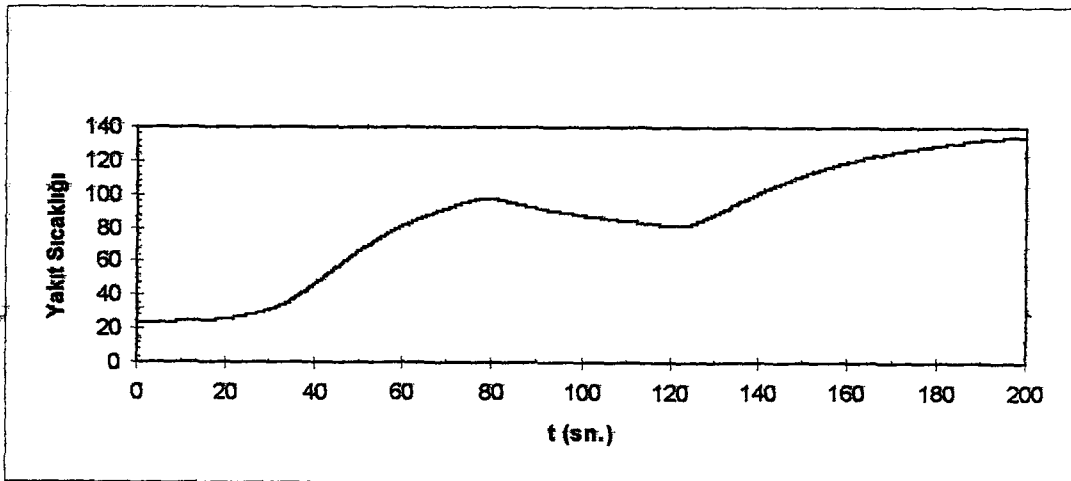
ALC ile bulunan sistem çıkışının, ayar noktasında yapılacak değişmeye karşı davranışını izlemek için Tablo 5.2 'deki değerler kullanılarak yapılan simülasyonda güç 200 Kw'a çıktıktan sonra 100 Kw'a indirilmiş ve daha sonra 250 Kw'a çıkarılmıştır. Şekil 5.3 'de bu çalışmaya ilişkin çizimler verilmiştir.

Tablo 5.2. Simülasyon başlangıç verileri

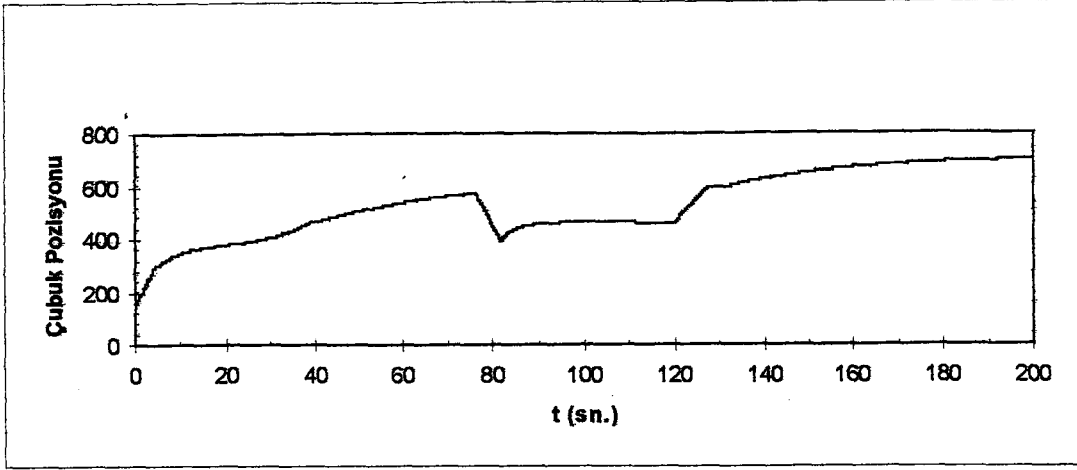
Başlangıç gücü: P_0	5 Kw
İstenen Peryot : τ	10 sn.
İstenen Güç: P_s	200 Kw-100 Kw-250 Kw
Örnekleme Aralığı: T	0.05 sn.
İnceleme Aralığı : T_{max}	0-200 sn.



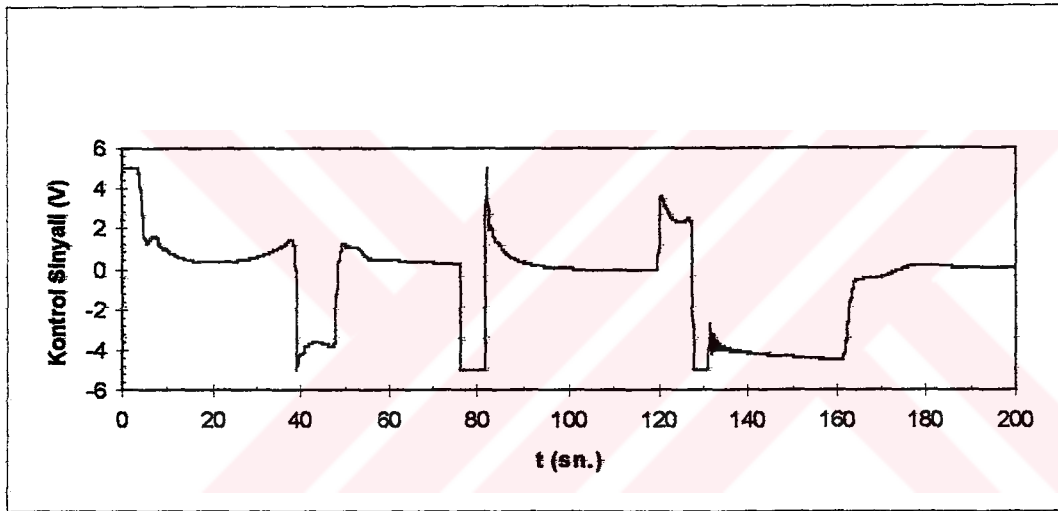
(a)



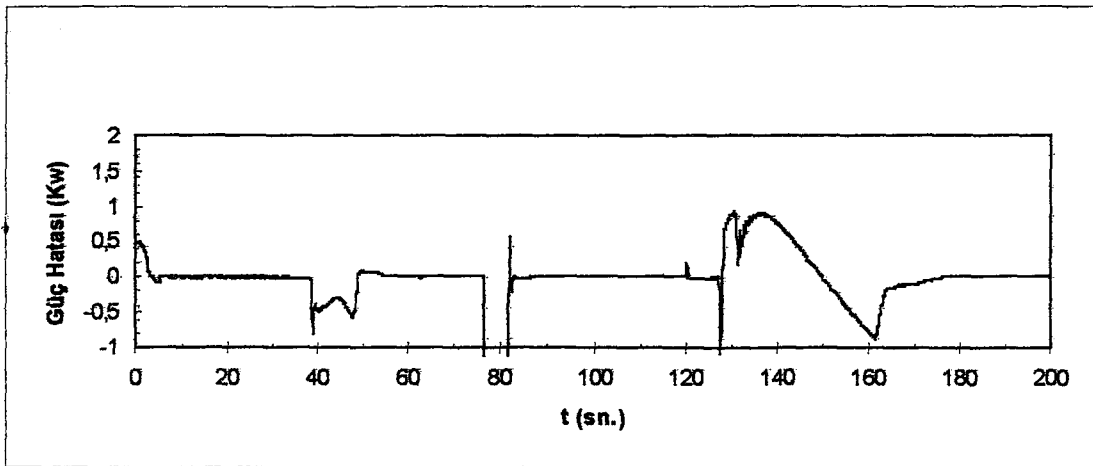
(b)



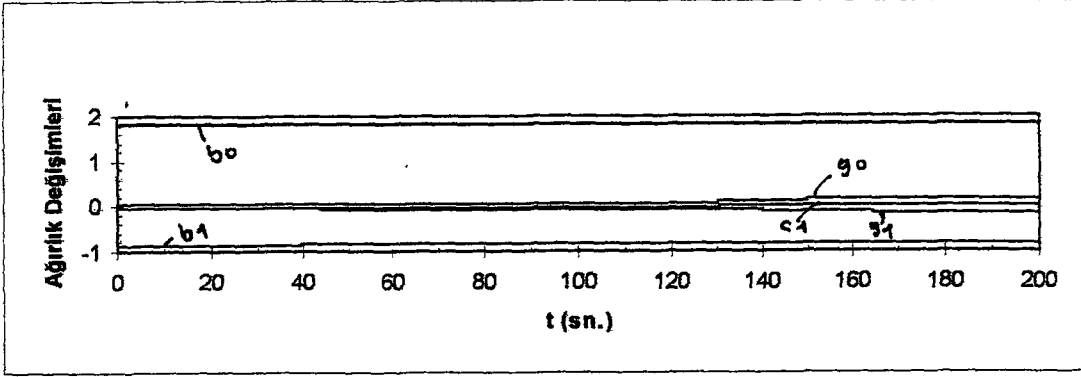
(c)



(d)



(e)



(f)

- Şekil 5.3. a) Güç Değişimi
 b) Yakıt Sıcaklığı (°C)
 c) Çubuk Pozisyonu
 d) Kontrol sinyali
 e) Gücün Yörüngeden Sapması
 f) Ağırlık Değişimleri.

Şekil 5.3 'de istenen gücün 200 Kw'a çıkışı Şekil 2.5'deki çizimlerin benzeridir. İstenen gücün 80. sn.'de 200 Kw'dan 100 Kw'a değiştirilmesi ile kontrol çubuğu maksimum hızla aşağı indirilmeye başlanmıştır. Çubuğun aşağı indirilmesi ile azalan güç Şekil 5.3b'de görüldüğü gibi yakıt sıcaklığının azalmasına yol açar.

Sistemdeki bu değişiklikler sonucu Şekil 5.3f 'de görüldüğü gibi ağırlıklar 40. sn.'den itibaren bir miktar değişmektedir. Bu değişim sonucu güç Şekil 5.3a'da görüldüğü gibi 80. sn.'den sonra 100 Kw'da sabit tutulmuştur.

Gücün 130. saniyede 250 Kw'a çıkarılması için istenen güç 250 Kw olarak girilmiştir. Şekil 5.3d'de görüldüğü gibi kontrol sinyali güç değişimini sağlayacak şekilde artmıştır. Kontrol sinyali, ayar çubuğunu Şekil 5.2c'deki gibi hareket ettirerek gücün daha önceden tanımlanan yörüngeyi izleyerek 250 Kw'a çıkmasını sağlamıştır. Şekil 5.3e'de gücün yörüngeden ne kadar saptığı gösterilmiştir. Güç 250 Kw'a çıktıktan sonra Şekil 5.2b'de verilen yakıt sıcaklığındaki artışa paralel olarak çubuk düşük hızda çekilmeye devam edilmiştir.

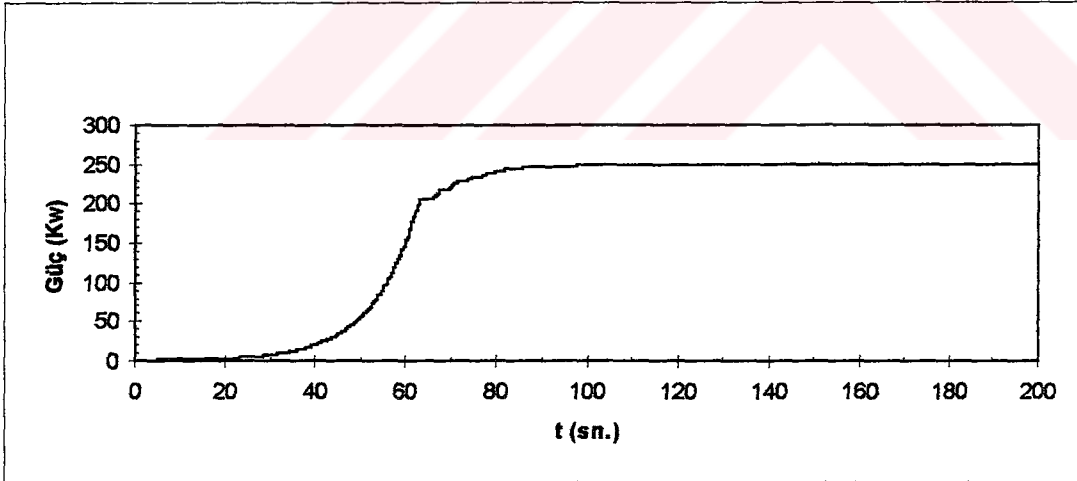
5.1.1. Başlangıç Gücü ve İstenen Güç Etkisinin İncelenmesi

Farklı başlangıç gücü ve istenen gücün ALC ile sistem çıkışını üreten reaktör üzerindeki etkileri incelemek için yapılan simülasyonlar aşağıda verilmiştir. Burada istenen güce çıkış periyodu her zaman 10 saniye alınmıştır.

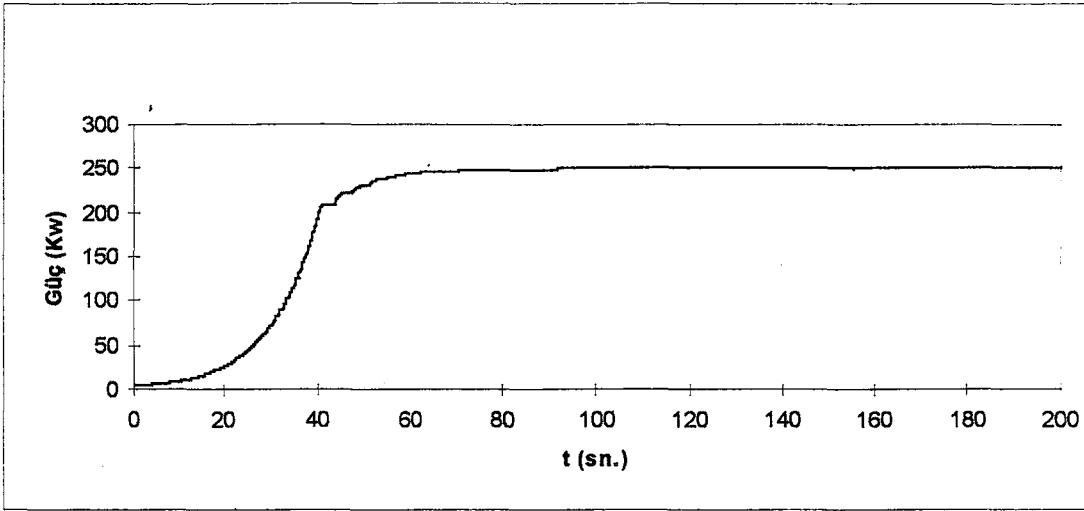
5.1.1.1 Başlangıç Gücünün Etkisinin İncelenmesi

Simülâtörün ALC kullanılarak değişik başlangıç güçleri altında davranışını izlemek için yapılan simülasyonlara ilişkin çizimler Şekil 5.4, Şekil 5.5, Şekil 5.6, Şekil 5.7 ve Şekil 5.8'de verilmiştir. Simülasyon, Tablo 5.1'deki değerler baz alınarak yapılmıştır. Başlangıç gücü ise 1 Kw ve 5 Kw olarak seçilmiştir.

Şekil 5.4'de verilen güç cevapları incelendiğinde; reaktör gücünün istenilen güç değerine çıkabildiği görülmektedir. Simülâtörün 100 Watt başlangıç gücü için iyi sonuç vermediği gözlenmiştir. Güç değişimi sabit periyotla yapıldığı için başlangıç gücüne bağlı olarak istenilen güç düzeyine çıkış süresi değişmektedir.

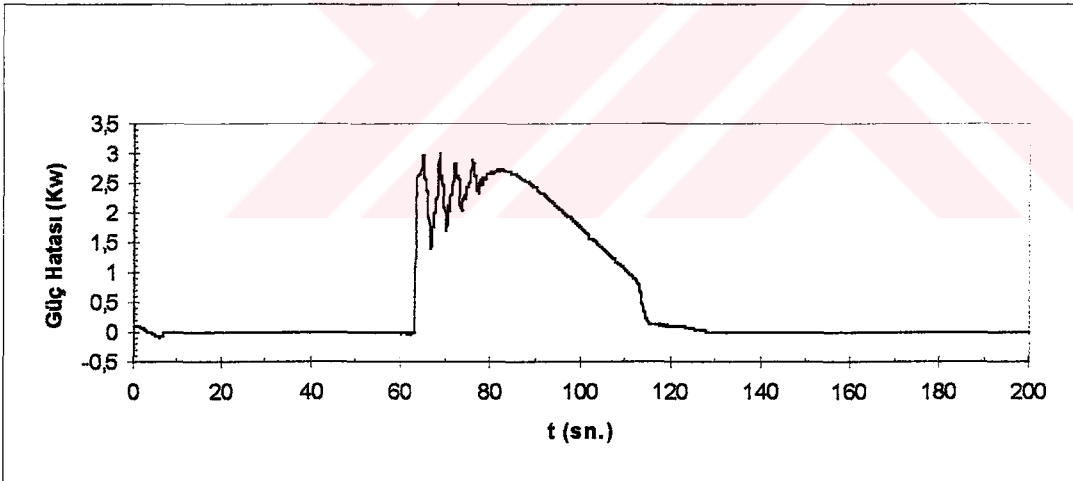


Şekil 5.4 Güç Değişimi a) P₀ = 1 Kw



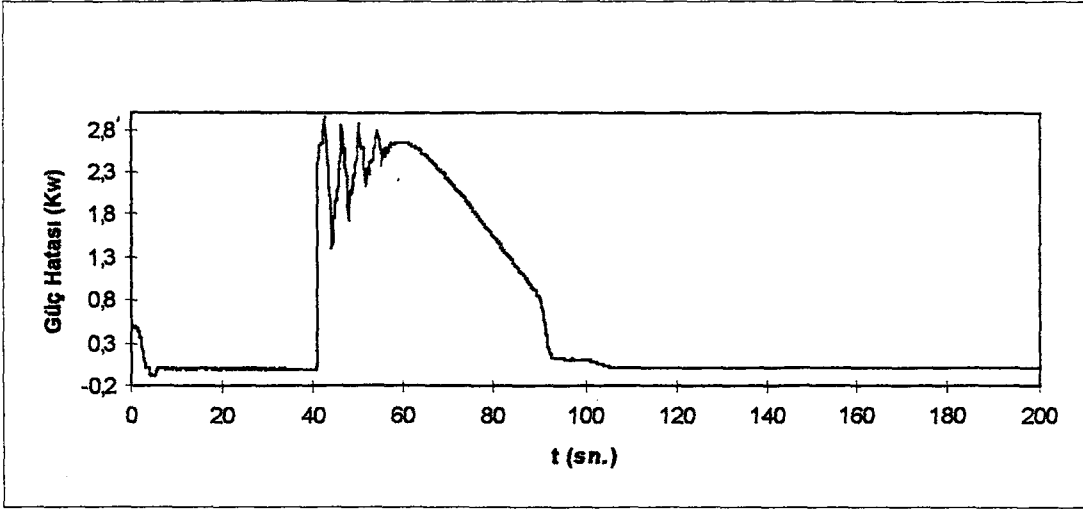
Şekil 5.4 Güç Değişimi b) $P_0 = 5 \text{ Kw}$

Şekil 5.5 'de gücün yörüngeden sapma miktarları verilmiştir. Şekil incelendiğinde kararlı hal hatasının sıfır olduğu görülmektedir. Yörüngeden en fazla sapma miktarları; başlangıç gücü 1 Kw için 3 Kw, başlangıç gücü 5 Kw için 2.8 Kw olmaktadır. Bu sapmalar istenilen güç düzeyine çıkarken olmaktadır.

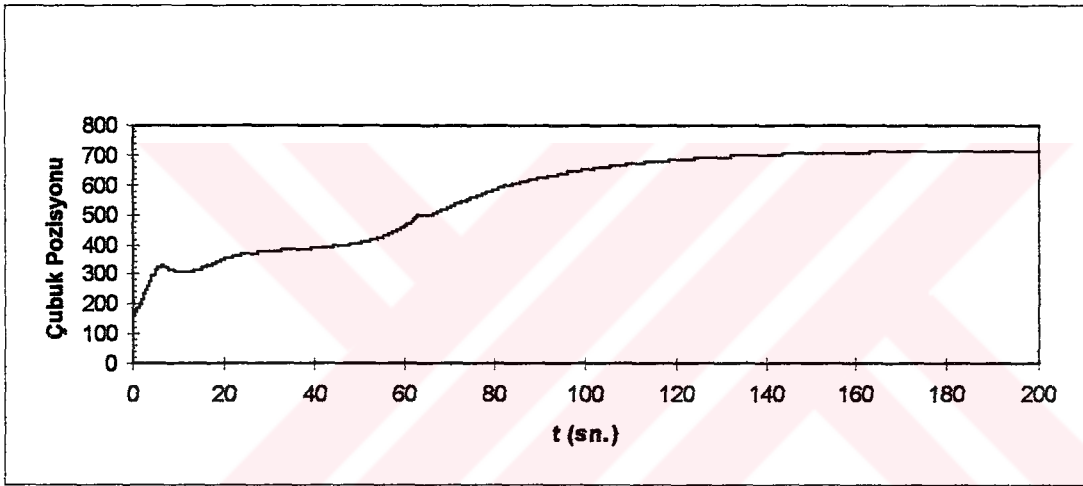


Şekil 5.5 Gücün Yörüngeden sapması a) $P_0 = 1 \text{ Kw}$

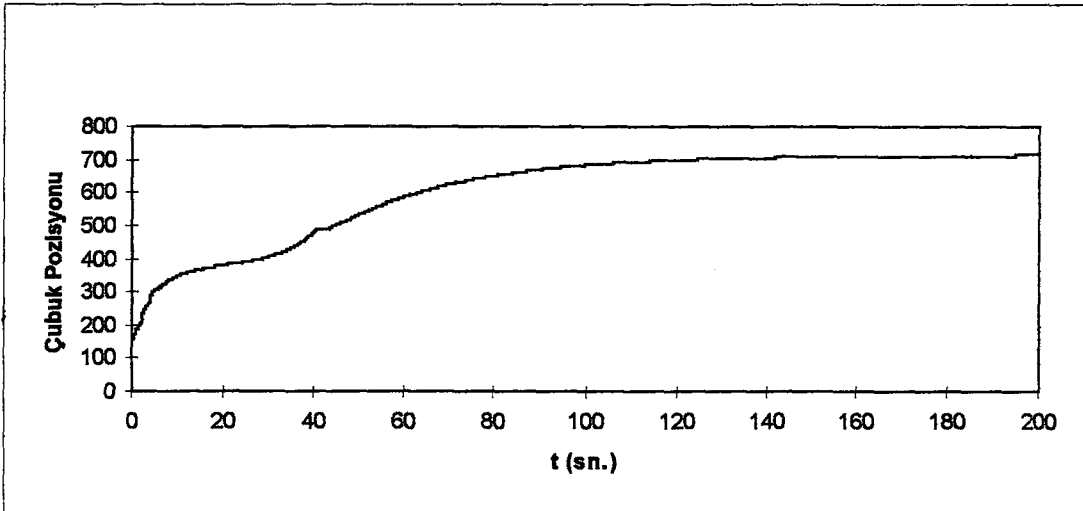
Şekil 5.6 'da farklı başlangıç güçleri için çubuk pozisyonunun değişimi ile ilgili grafikler verilmiştir. Kontrol çubuğu Şekil 5.6a ve b'de benzer şekilde hareket etmektedir. Başlangıçta hızlı çekilen çubuk bir süre sonra bir miktar aşağı indirilmekte (Şekil 5.6b'de çok az miktarda inmektedir) ve daha sonra sıcaklık etkisi nedeniyle oluşan negatif geribeslemeyi dengeleyecek şekilde yukarı çekilmektedir. Sıcaklık değişimi son bulduğunda ise çubuk pozisyonu 700 civarında sabit kalmaktadır.



Şekil 5.5 Gücün Yörüngeden Sapması b) $P_0 = 5 \text{ Kw}$

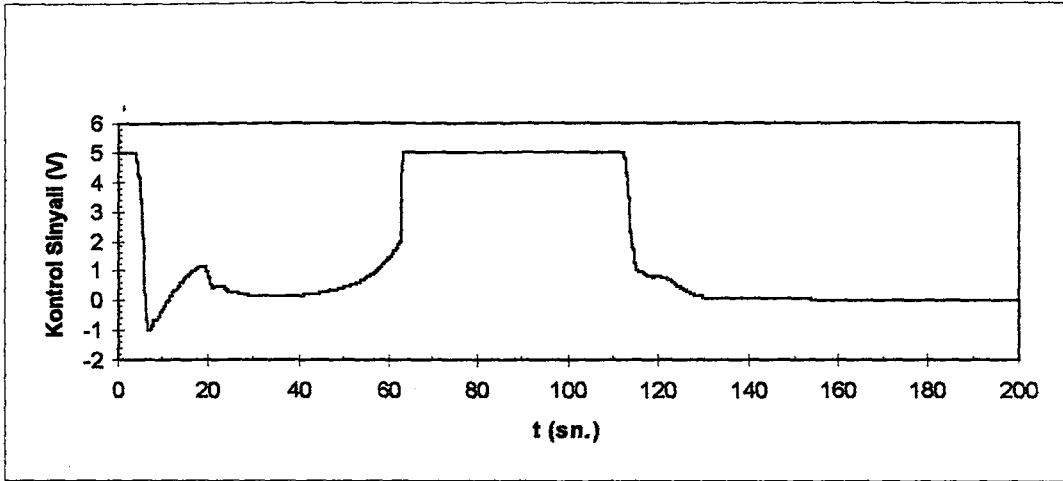


(a)

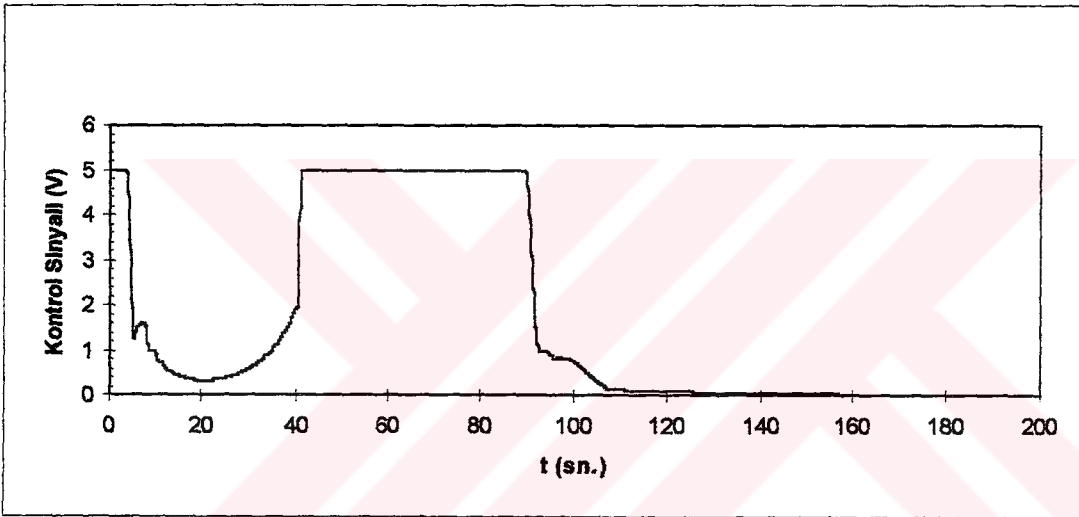


(b)

Şekil 5.6 Çubuk Pozisyonu a) $P_0 = 1 \text{ Kw}$ b) $P_0 = 5 \text{ Kw}$



(a)



(b)

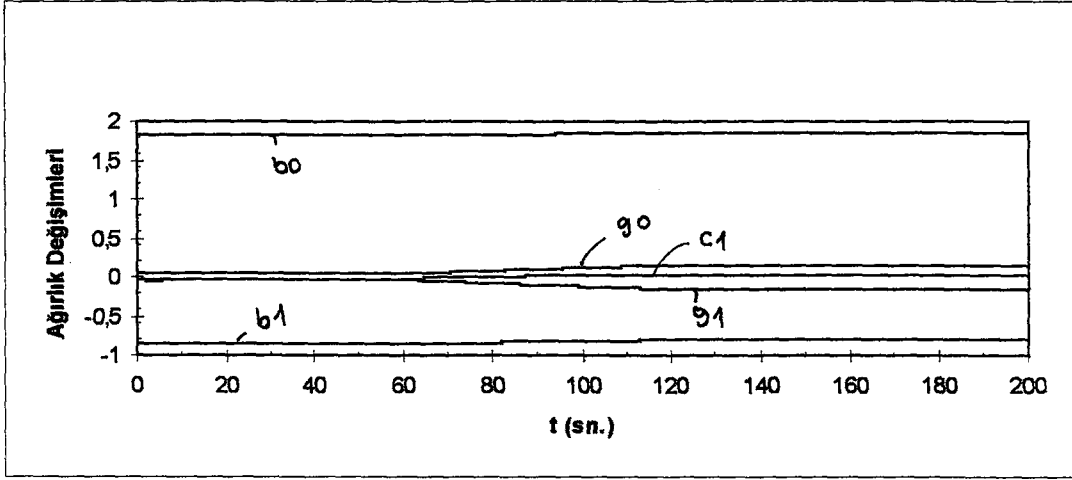
Şekil 5.7 Kontrol Sinyali

a) $P_0 = 1 \text{ Kw}$ b) $P_0 = 5 \text{ Kw}$

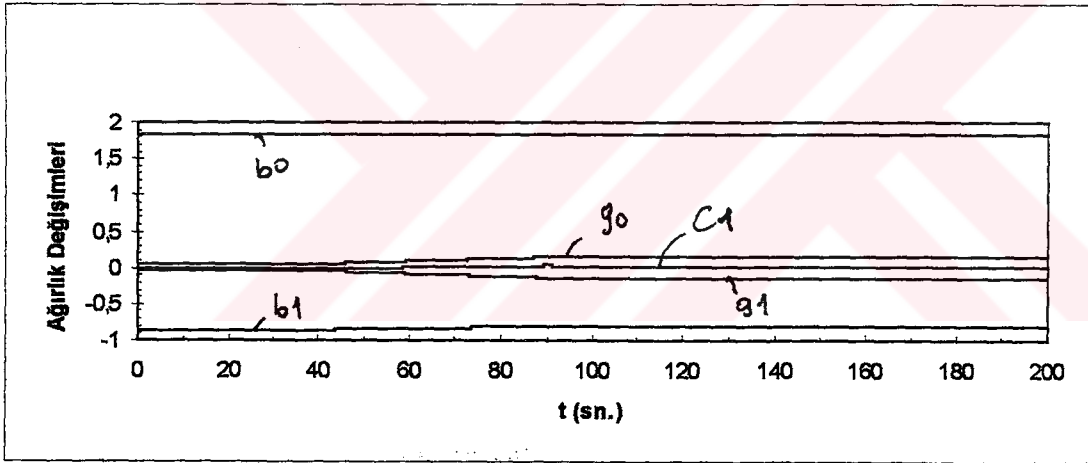
Şekil 5.7'de kontrol çubuğunu hareket ettirmek için simülatör tarafından üretilen kontrol sinyalleri görülmektedir. Şekillerde de görüldüğü gibi sıcaklık değişimi sona erdiğinde kontrol sinyali sıfır olarak çubuğun son pozisyonunda kalmasını sağlamaktadır.

Şekil 5.8 'de ALC ile sistem çıkışının bulunduğu ağırlık değişimleri ile ilgili grafikler görülmektedir. İstenilen güç düzeyine çıkarken ağırlıkların bir miktar değiştiği görülmektedir.

Sonuç olarak, simülatöre sistem çıkışı üreten, bir yapay sinir ağı modeli olan ALC; değişik başlangıç güçleri altında istenilen güç düzeyine verilen yörüngeyi izleyerek çıkabildiği, kararlı hal hatasının ise olmadığı görülmüştür.



(a)



(b)

Şekil 5.8 Ağırlık Değişimleri

a) $P_0 = 1 \text{ Kw}$

b) $P_0 = 5 \text{ Kw}$

5.1.1.2 İstenen Gücün Etkisinin İncelenmesi

ALC ile sistem çıkışı üretilmiş simülatörün değişik güç seviyelerine çıkarken göstereceği davranışı izlemek için yapılan simülasyonlara ilişkin çizimler Şekil 5.8, Şekil 5.9, Şekil 5.10, Şekil 5.11, Şekil 5.12 ve Şekil 5.13 'de verilmiştir. Simülasyon,

Tablo 5.1'deki deęerler baz alınarak yapılmıřtır. İstenilen gc ise 50 Kw, 100 Kw ve 250 Kw olarak seilmiřtir. Yalnızca 50 Kw iin simlatr 100 saniyeye kadar alıřtırılmıřtır. nk 100 saniyeden sonra da sistem parametrelerinde deęiřiklik olmamıřtır.

řekil 5.9'da verilen gc deęiřimleri incelendięinde sistem ıkıřı ALC ile bulunan simlatrn reaktr gcn istenilen dzeye ıkarabildięi grlmektedir. Gc deęiřimi sabit peryotla (10 sn.) yapıldıęı iin istenilen gce baęlı olarak gce ıkıř sresi deęiřmektedir.

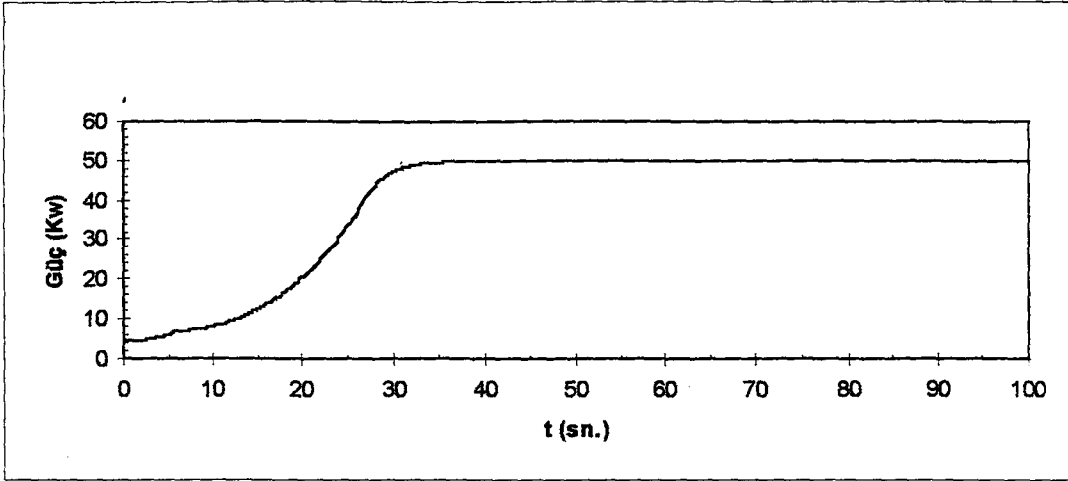
řekil 5.10 'da gcn yrngeden sapma miktarları verilmiřtir. řekil 5.10'daki incelendięinde kararlı hal hatasının sıfır olduęu grlmektedir. Yrngeden en fazla sapma miktarı; 50 Kw iin 0.5 Kw, 100 Kw iin 0.4 Kw ve 250 Kw iin 3 Kw olmaktadır. Bu sapmalar istenilen gc dzeyine ıkarken olmaktadır.

řekil 5.11 'de 50 Kw, 100 Kw ve 250 Kw gc dzeyine ıkmada kontrol ubuęunun pozisyon deęiřimi ile ilgili grafikler verilmiřtir. Kontrol ubuęu řekil 5.11a, b ve c'de bařlangıta benzer řekilde hareket etmektedir. Hızlı ekilen ubuk bir sre sonra bir miktar ařaęı indirilmektedir. Daha sonra sıcaklıęın etkisi ile ubuk ekilmeye devam etmekte ve istenilen gce eriřildikten sonra, sıcaklıęın etkisi de giderek azaldıęından ubuk ok az hareket etmektedir.

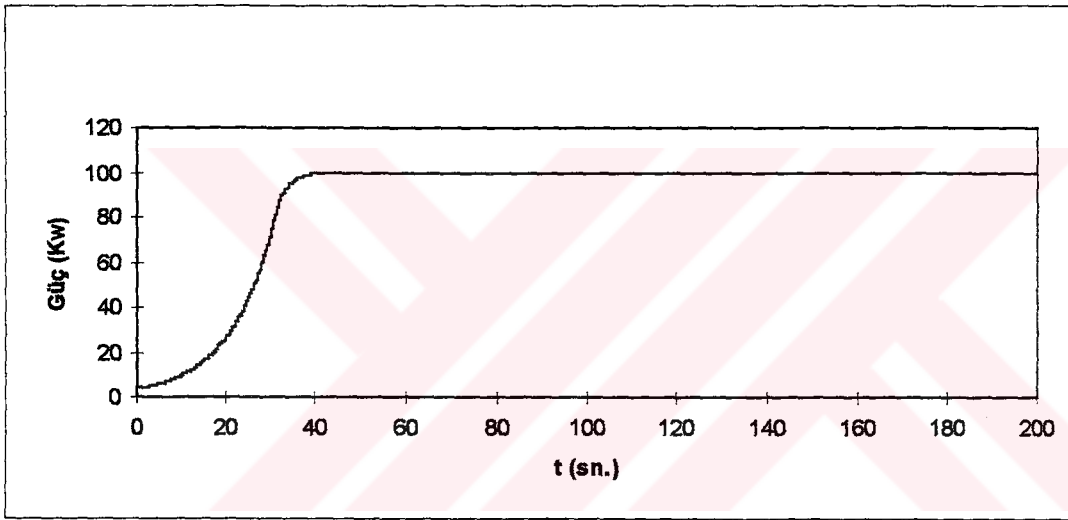
řekil 5.12'de kontrol ubuęunu hareket ettirmek iin retilen kontrol sinyalleri grlmektedir. Grafiklerde grldę gibi sıcaklık etkisinin az olduęu 50 kw ve 100 Kw'da kontrol sinyali daha nce sıfıra giderken (řekil 5.12a ve b), sıcaklık etkisinin fazla olduęu 250 Kw'da kontrol sinyali daha sonra sıfır olarak ubuęun son pozisyonunda kalması saęlanmaktadır.

řekil 5.13'de sistem ıkıřını belirleyen parametrelere iliřkin aęırlıkların deęiřimi grlmektedir. Sistem parametrelerindeki deęiřim sona erdięinde aęırlıklar sabit kalmaktadır.

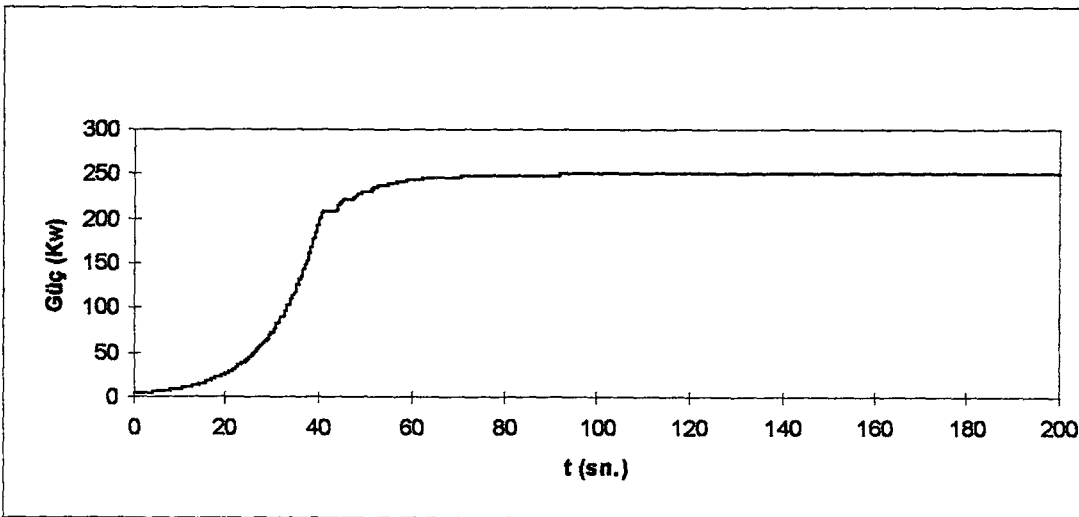
Sonuç olarak, sistem ıkıřı ALC ile bulunan simlatrn deęiřik gc dzeylerine verilen yrngeyi izleyerek ıkabildięi, kararlı hal hatasının olmadığı grlmřtir.



(a)

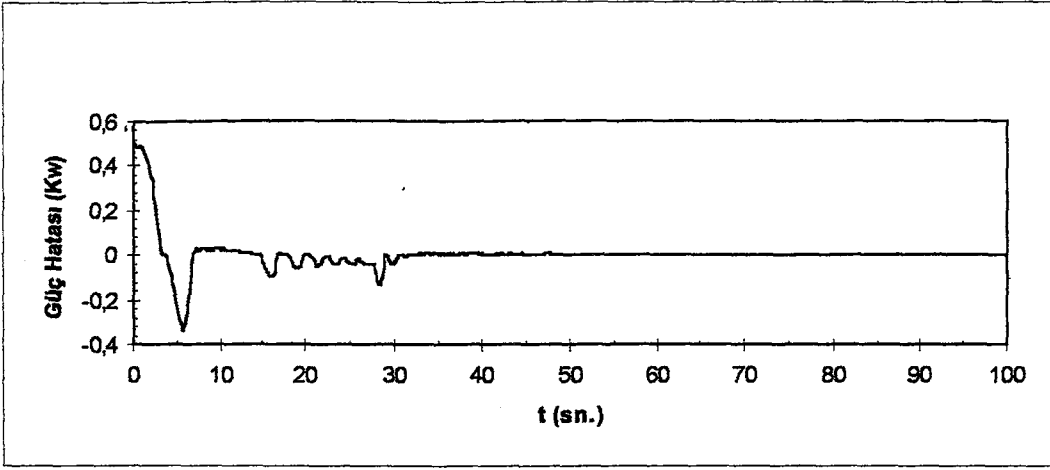


(b)

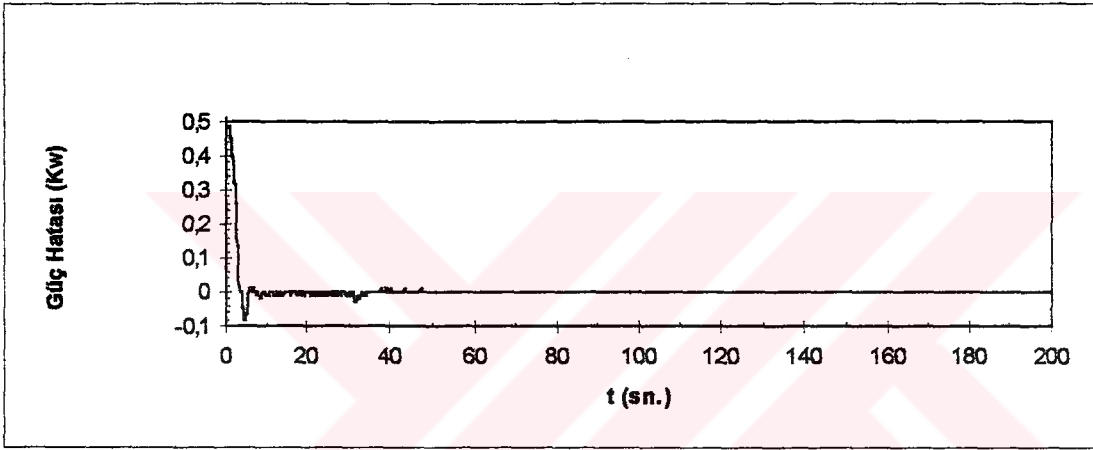


(c)

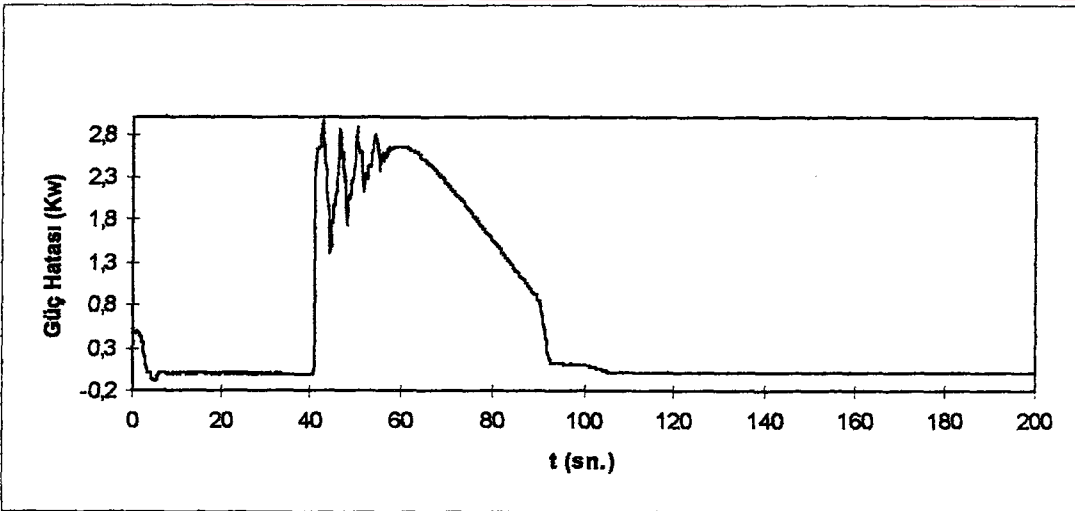
Şekil 5.9 Güç Değişimi a) $P_{\max} = 50 \text{ Kw}$, b) $P_{\max} = 100 \text{ Kw}$ ve c) $P_{\max} = 250 \text{ Kw}$



(a)



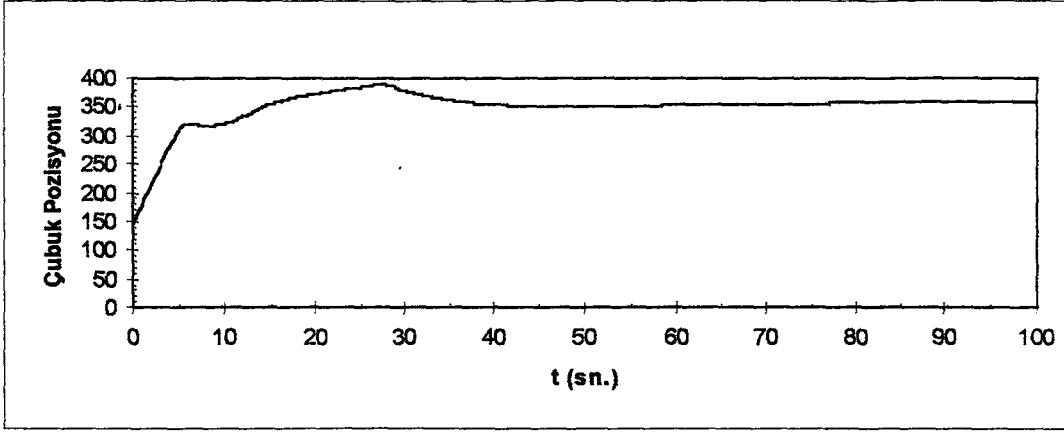
(b)



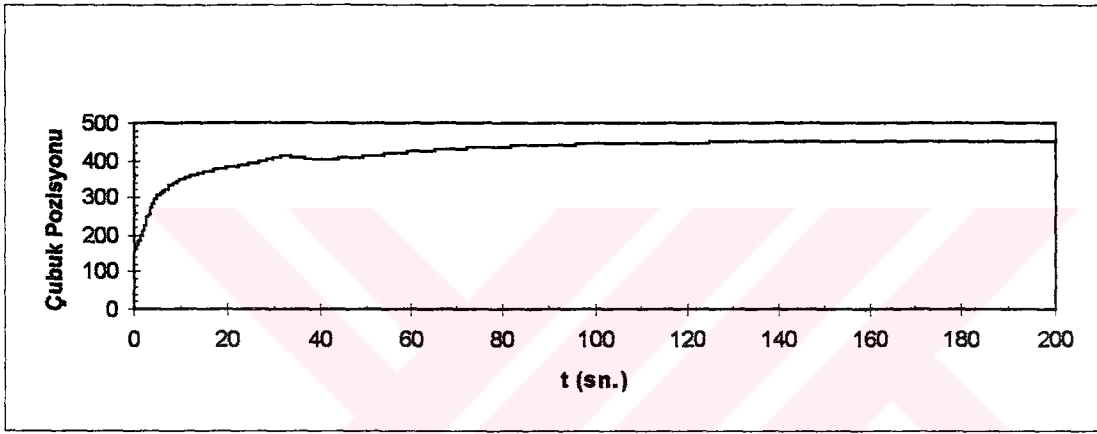
(c)

Şekil 5.10 Gücün Yörüngeden Sapması

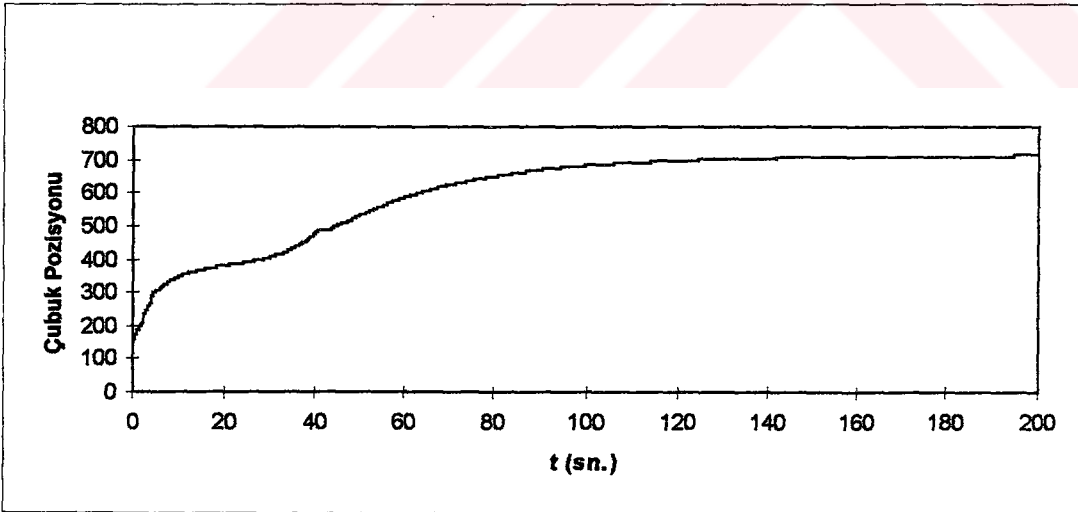
a) $P_{\max} = 50 \text{ Kw}$, b) $P_{\max} = 100 \text{ Kw}$, c) $P_{\max} = 250 \text{ Kw}$



(a)



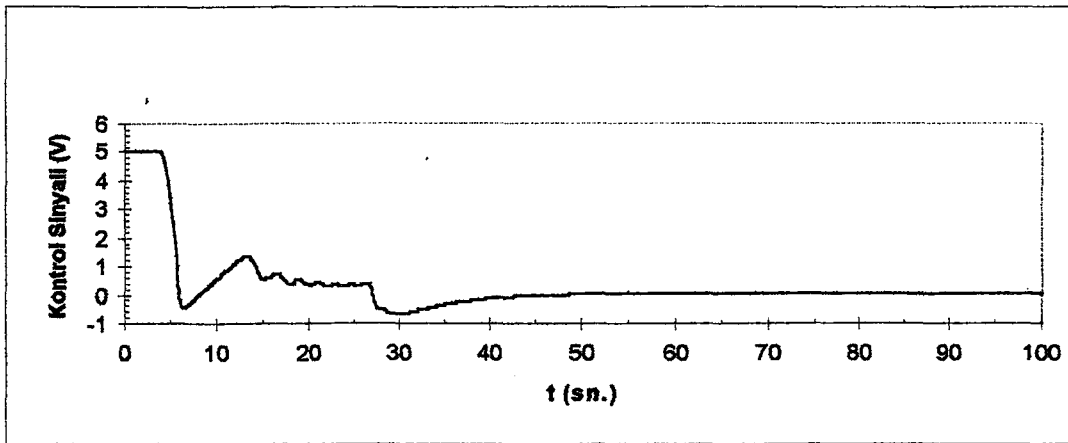
(b)



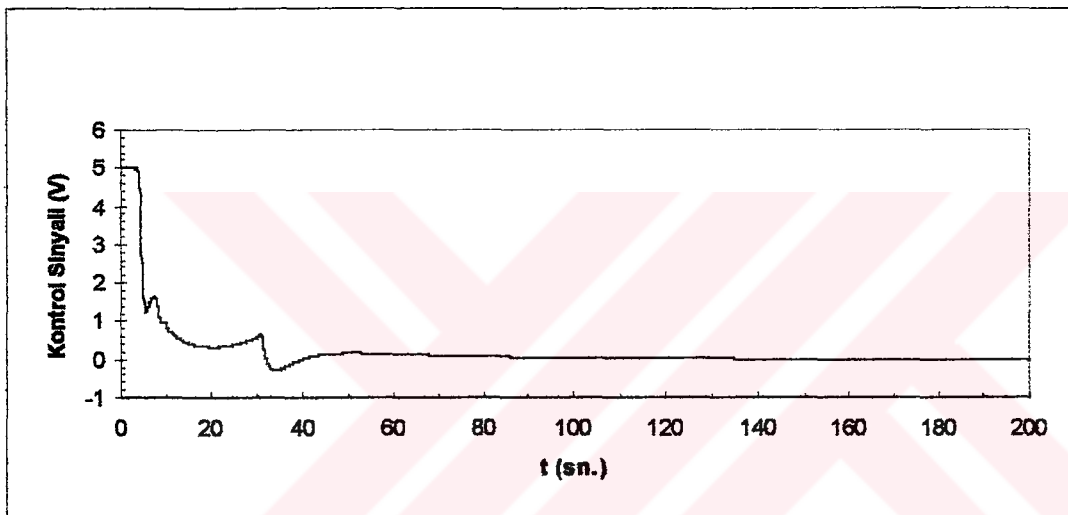
(c)

Şekil 5.11 Çubuk Pozisyonları

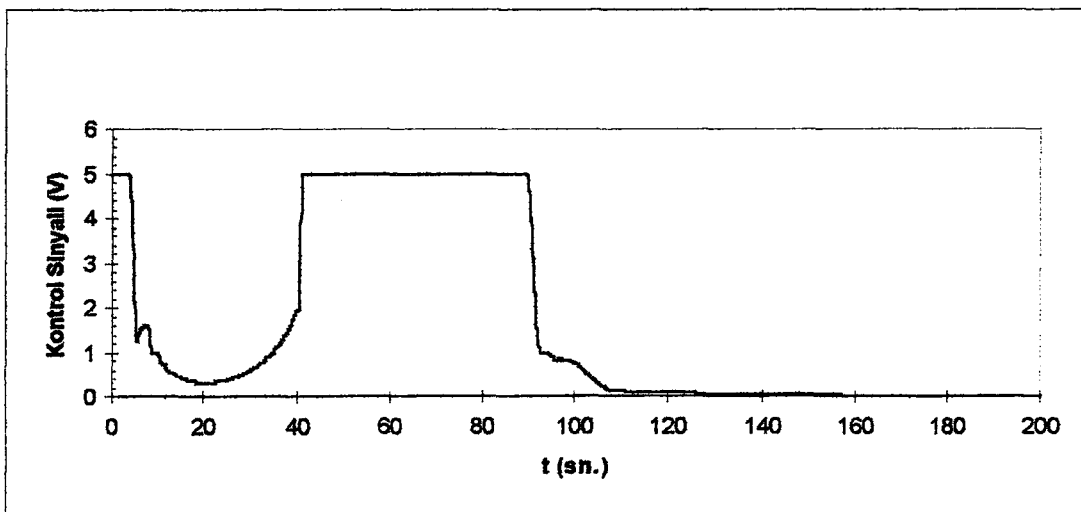
a) $P_{\max} = 50 \text{ Kw}$ b) $P_{\max} = 100 \text{ Kw}$ c) $P_{\max} = 250 \text{ Kw}$



(a)



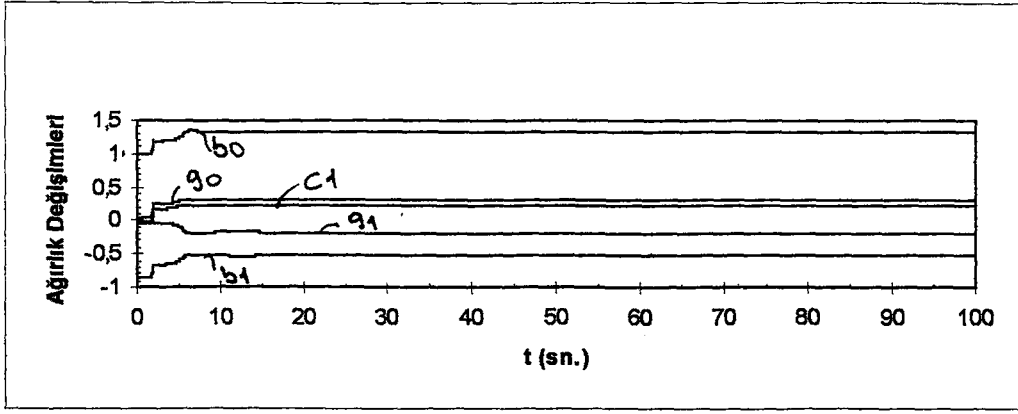
(b)



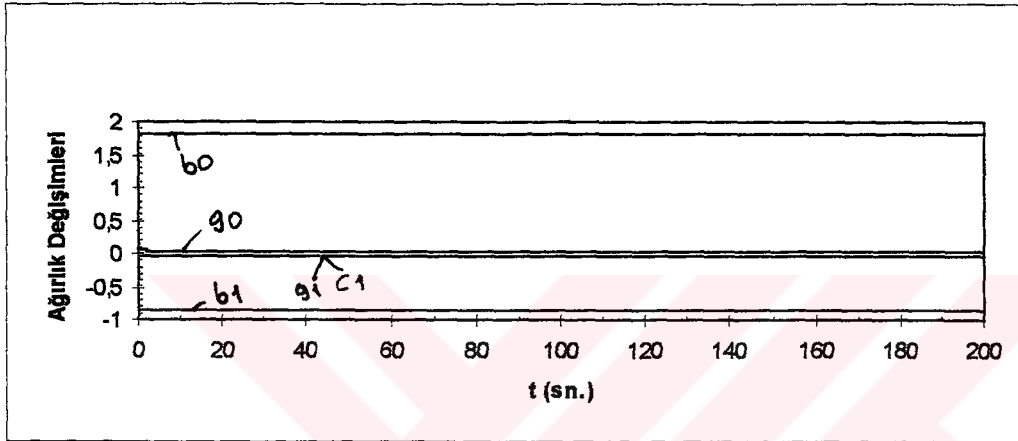
(c)

Şekil 5.12 Kontrol Sinyali

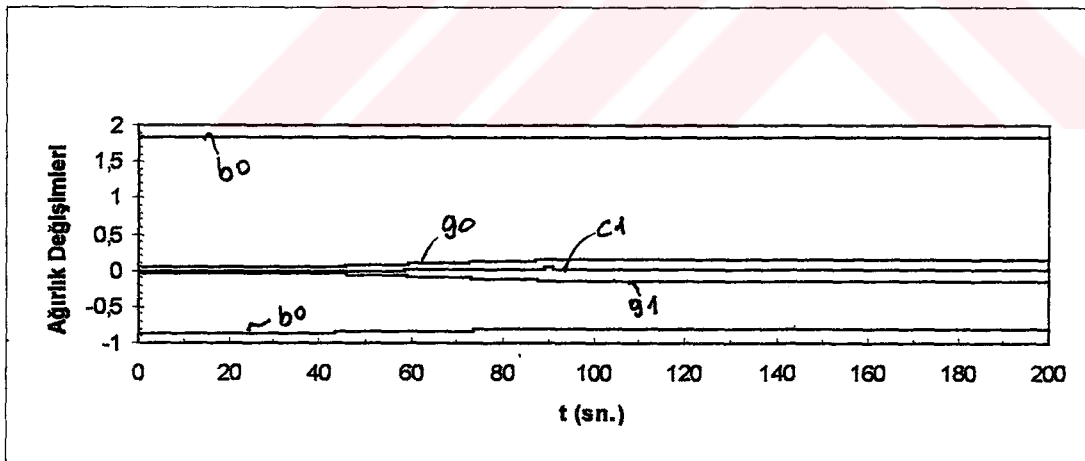
a) $P_{\max} = 50 \text{ Kw}$, b) $P_{\max} = 100 \text{ Kw}$, c) $P_{\max} = 250 \text{ Kw}$



(a)



(b)



(c)

Şekil 5.13 Ağırlık Değişimleri

a) $P_{\max} = 50 \text{ Kw}$, b) $P_{\max} = 100 \text{ Kw}$, c) $P_{\max} = 250 \text{ Kw}$

5.1.1.3 Örnekleme Aralığının Etkisinin İncelenmesi

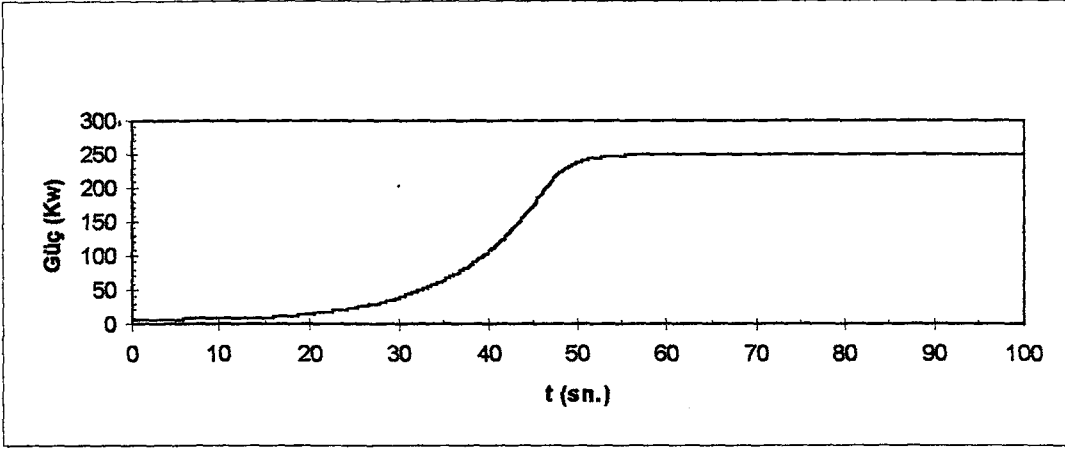
Sistem çıkışı ALC ile bulunan reaktör simülatörünün değişik örnekleme aralıklarındaki davranışını izlemek için yapılan simülasyonlara ilişkin grafikler Şekil 5.14, 15, 16, 17 ve 18’de verilmiştir. Örnekleme aralığı ise 0.2 sn., 0.4 sn. ve 0.8 sn. olarak seçilmiştir. Yalnız simülatör 0.2 sn. örnekleme aralığı için inceleme aralığı 100 sn. seçilmiştir.

Şekil 5.14 ‘de verilen güç değişimleri incelendiğinde; simülatörün istenilen güç düzeyine ulaştığı görülmektedir. Fakat, örnekleme aralığı arttıkça güç düzeyinde meydana gelen bozulmalar artmaktadır. 0.2 ve 0.4 saniye örnekleme aralığında herhangi bir bozulma görülmezken, örnekleme aralığı 0.8 alındığında, 40 Kw civarında bir bozulma meydana gelmiştir. Örnekleme aralığı 0.8 alındığında güç 250 Kw’a çıktığında, güçte meydana gelen salınımları önlemek için bir PID kontrolör tasarlanabilir.

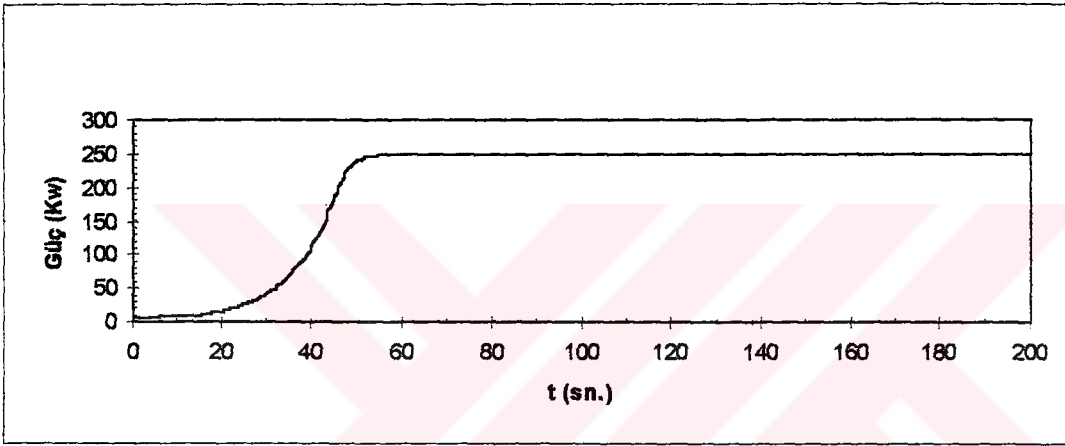
Şekil 5.15’de gücün yörüngeden sapma miktarları verilmiştir. Grafiklerde de görüldüğü gibi 0.2 ve 0.4 sn.’lik örnekleme aralıklarında kararlı hal hatası sıfırdır. Fakat 0.8 sn. örnekleme aralığında 50 Kw’lık bir hata söz konusudur. Burada örnekleme aralığı büyüdüğünden, gücün yörüngeden sapması artmaktadır.

Şekil 5.16’da çubuk pozisyonuna ilişkin grafikler verilmiştir. Kontrol çubuğu Şekil 5.16a, b ve c’de başlangıçta benzer şekilde hareket etmektedir. Hızlı çekilen çubuk bir miktar aşağı indirildikten sonra daha yavaş çekilmektedir. İstenilen güç düzeyine çıkarken, örnekleme aralığına bağlı olarak çubuk pozisyonunda bazı bozulmalar meydana gelmektedir.

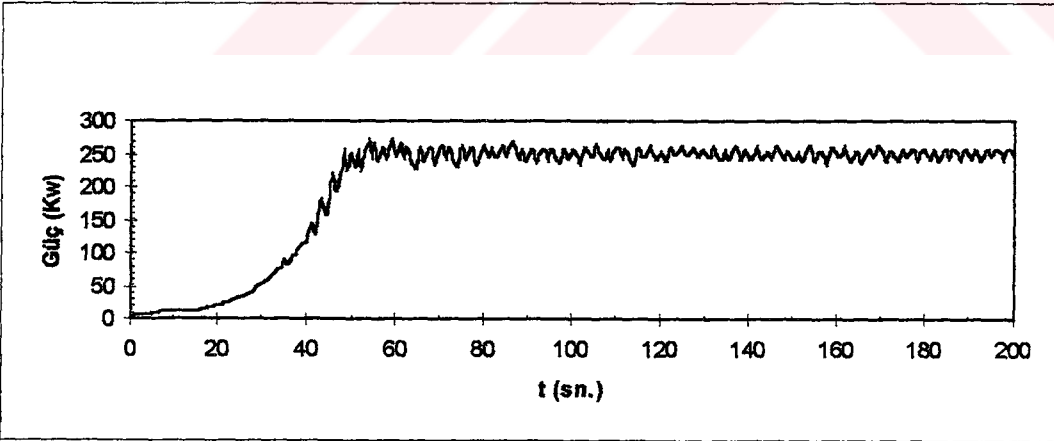
Şekil 5.17’de Kontrol sinyallerine ilişkin grafikler görülmektedir. Örnekleme aralığının 0.8 sn. gibi büyük seçilmesi kontrol sinyalinin uzun süre salınım yapmasına neden olmaktadır.



(a)



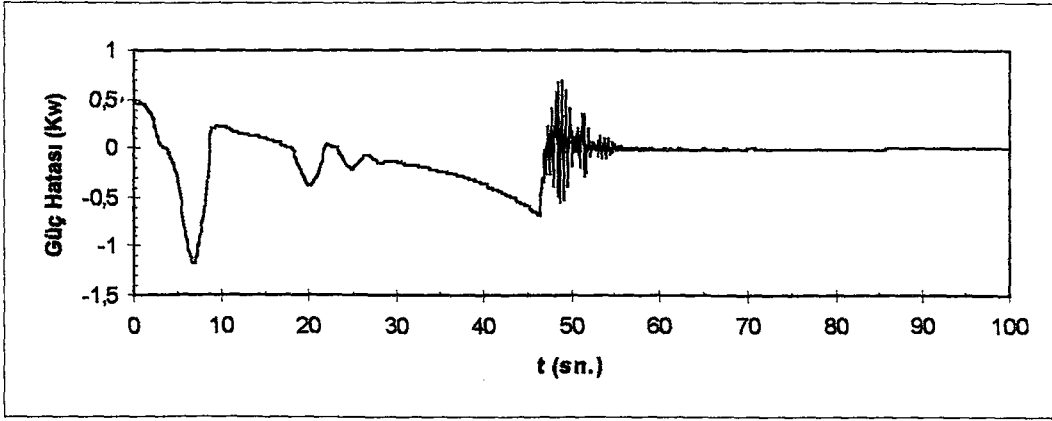
(b)



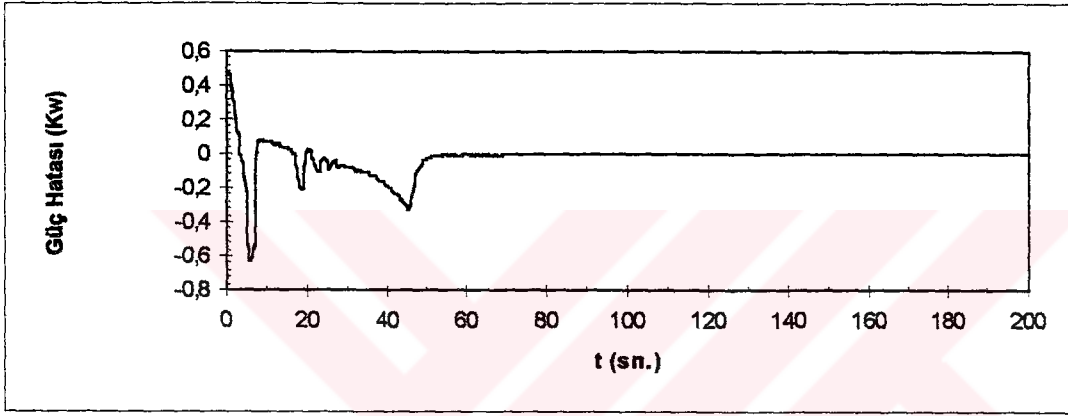
(c)

Şekil 5.14 Güç Değişimi

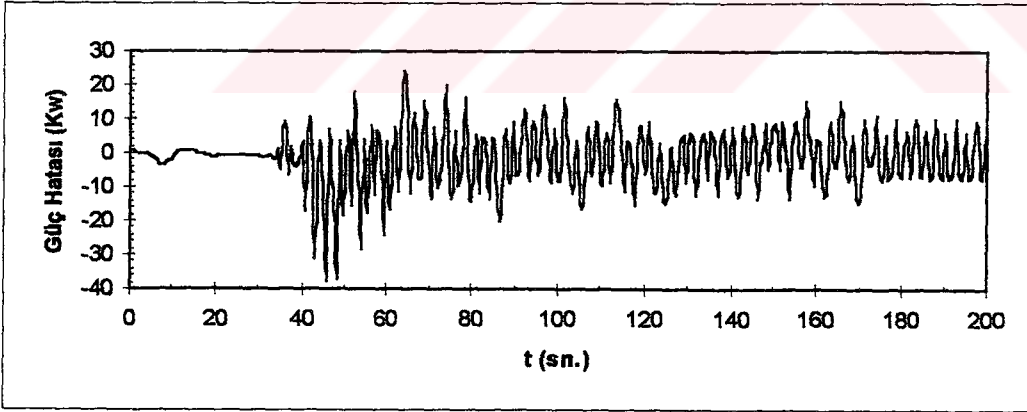
a) $T=0.2$ sn.b) $T=0.4$ sn.c) $T=0.8$ sn.



(a)



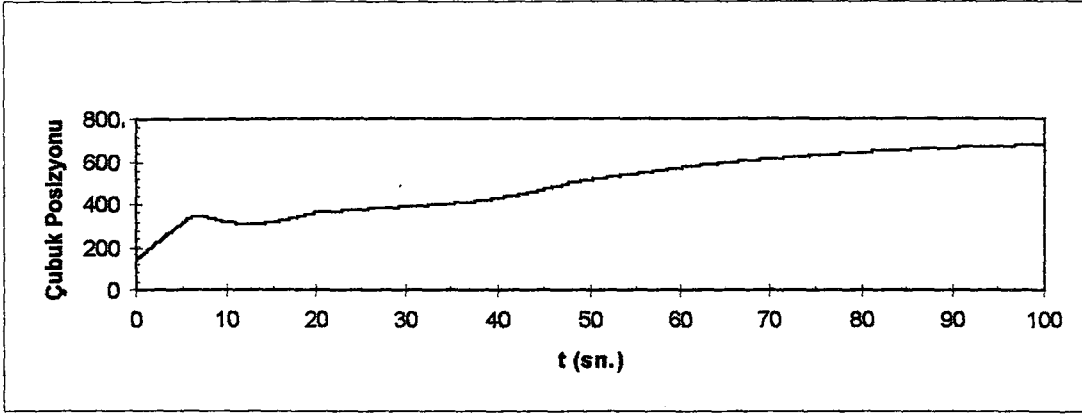
(b)



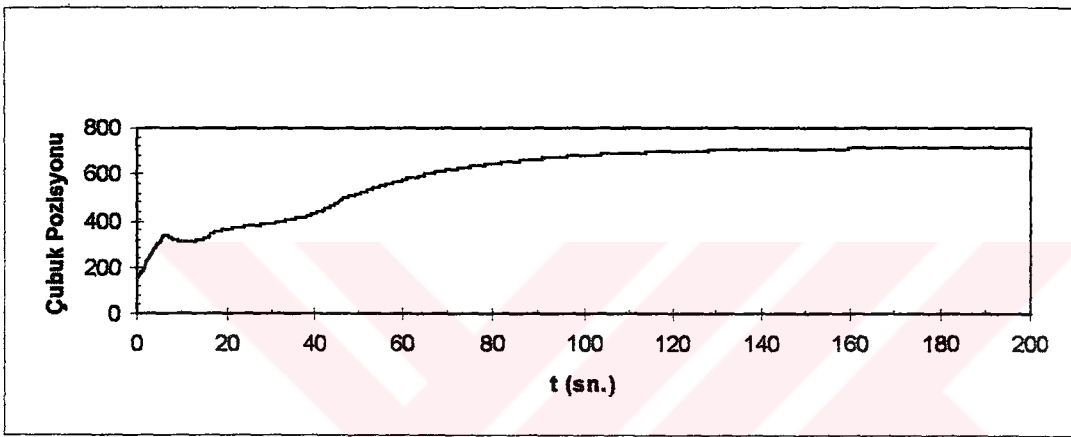
(c)

Şekil 5.15 Gücün Yörüngeden Sapması

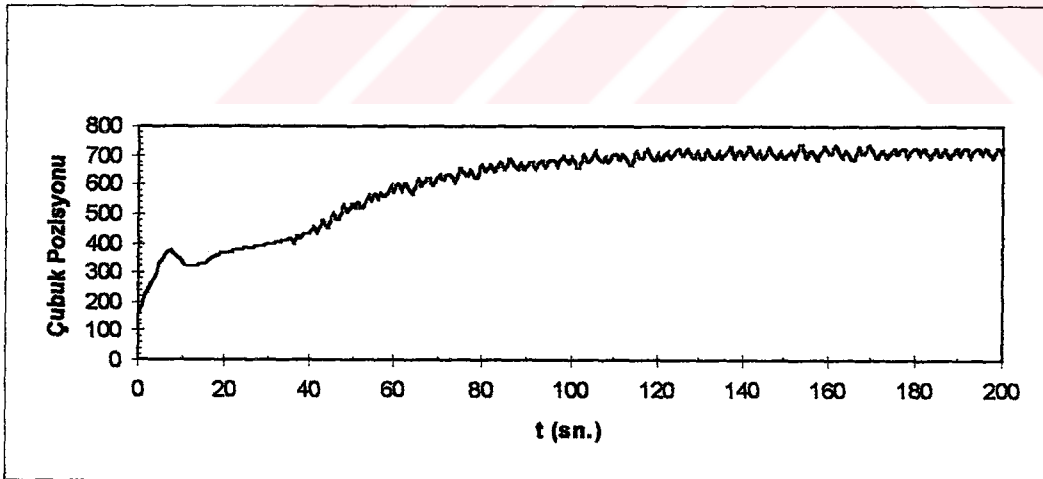
a) $T=0.2$ sn.b) $T=0.4$ sn.c) $T=0.8$ sn.



(a)



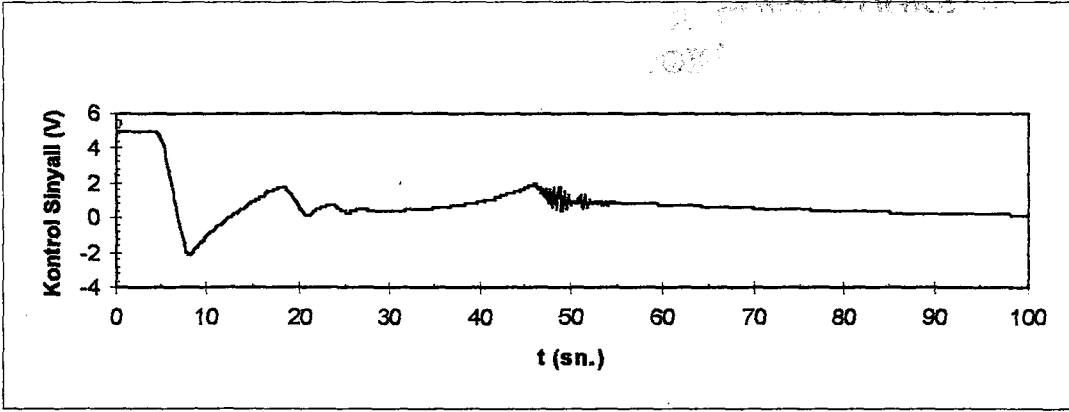
(b)



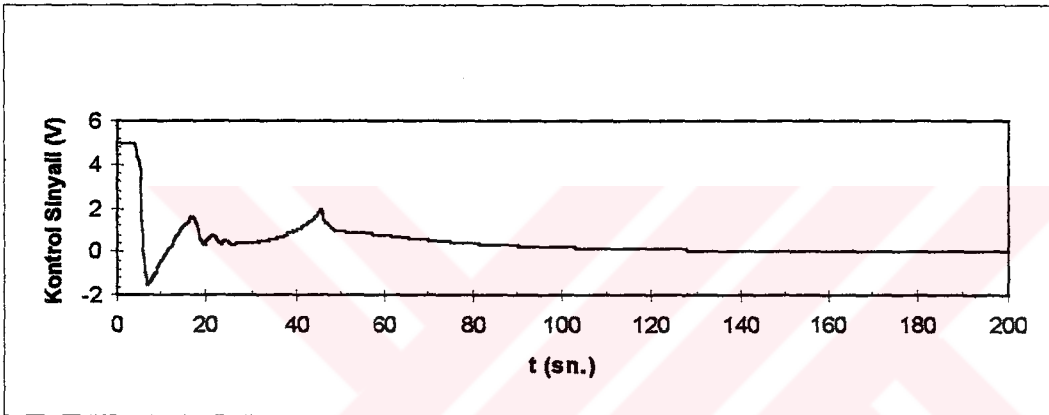
(c)

Şekil 5.16 Çubuk Pozisyonu

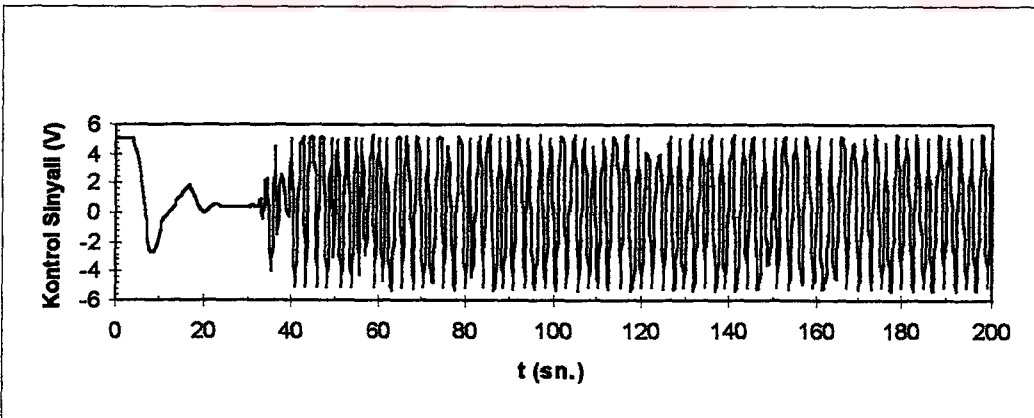
a) $T=0.2$ sn.b) $T=0.4$ sn.c) $T=0.8$ sn.



(a)



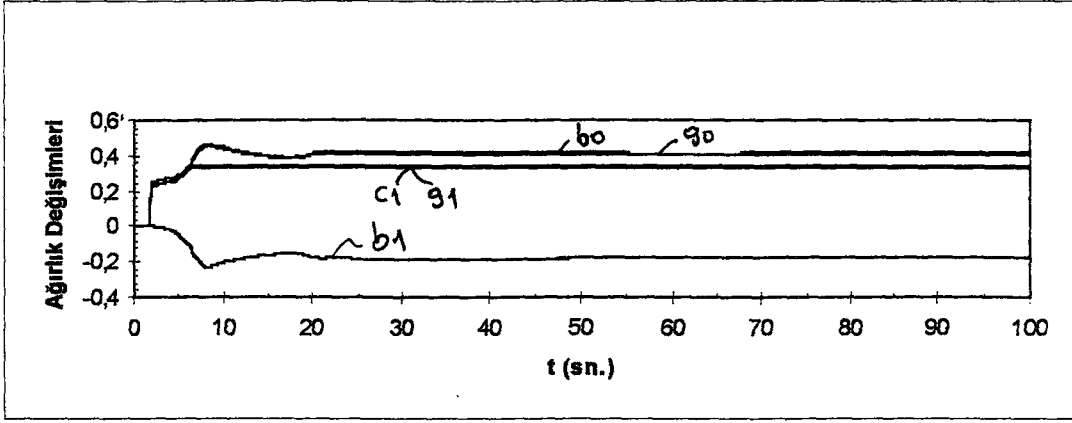
(b)



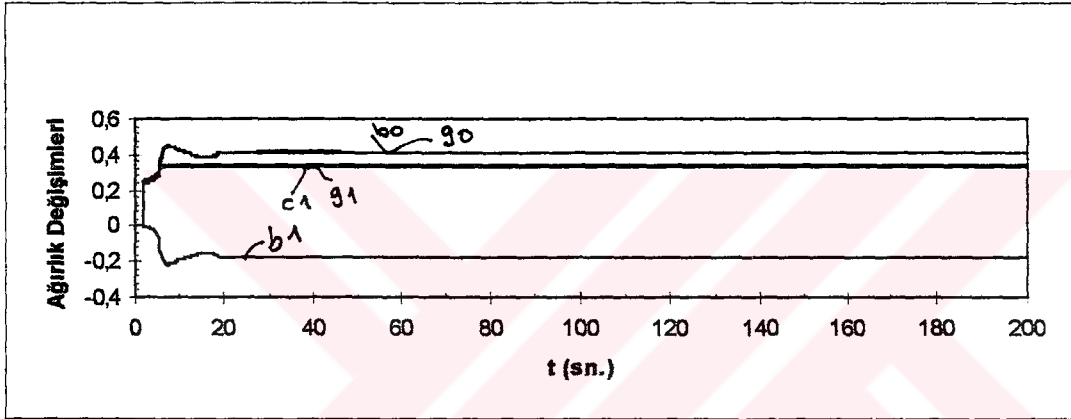
(c)

Şekil 5.17 Kontrol Sinyali

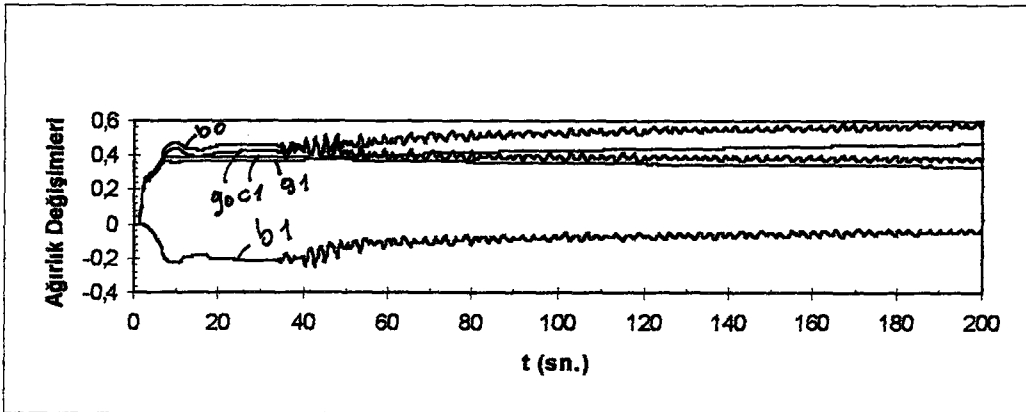
a) $T=0.2$ sn.b) $T=0.4$ sn.c) $T=0.8$ sn.



(a)



(b)



(c)

Şekil 5.18 Ağırlık Değişimleri

a) $T=0.2$ sn.b) $T=0.4$ sn.c) $T=0.8$ sn.

Şekil 5.18’de simülátöre sistem çıkışını veren parametrelerin ağırlık grafikleri görölmektedir. Grafiklerde görüldüğü gibi örnekleme aralığı arttıkça parametrelerin kararlı hal değerlerine erişme süresi artmaktadır. Parametreler ölçülen değerlere göre ALC tarafından ağırlıkları değiştirildiği için örnekleme aralığının artması sistemde meydana gelen deęişimlere göre ağırlıklarını ayarlaması güçleşmektedir. Hatta 0.8 sn. örnekleme aralığı için bazı parametreler kararlı hal değerlerine ulaşmamaktadır.

Sonuç olarak, simülátör büyük örnekleme aralıklarında istenen güç düzeyine verilen yörüngeyi izleyerek çıkamamaktadır. Bu nedenle örnekleme aralığı 0.8 saniyeden küçük seçilmelidir.



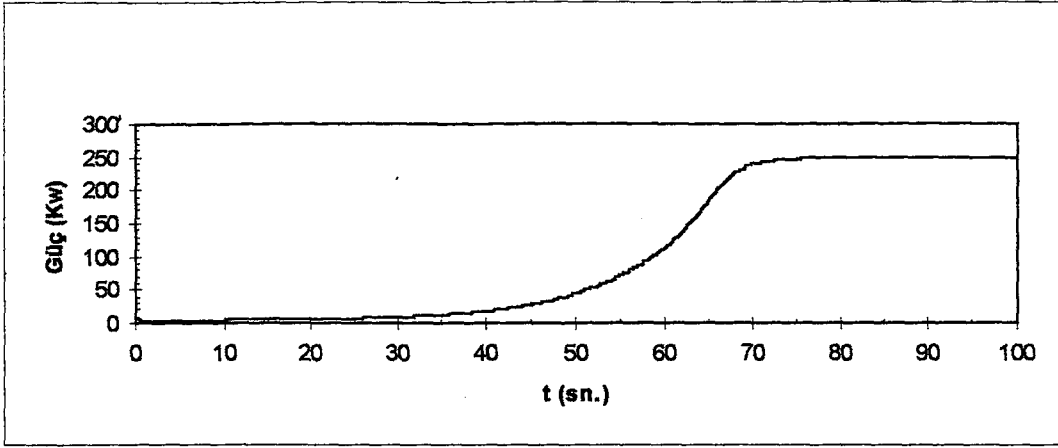
5.2 Lineer Olmayan Yapay Sinir Ağı Kontrolörünün Simülasyonla Elde Edilen Sonuçları

YSA kontrolörü simülatörde Tablo 5.3’de verilen değerler kullanılarak simüle edilmiştir. Elde edilen sonuçlar Şekil 5.19’da görülmektedir.

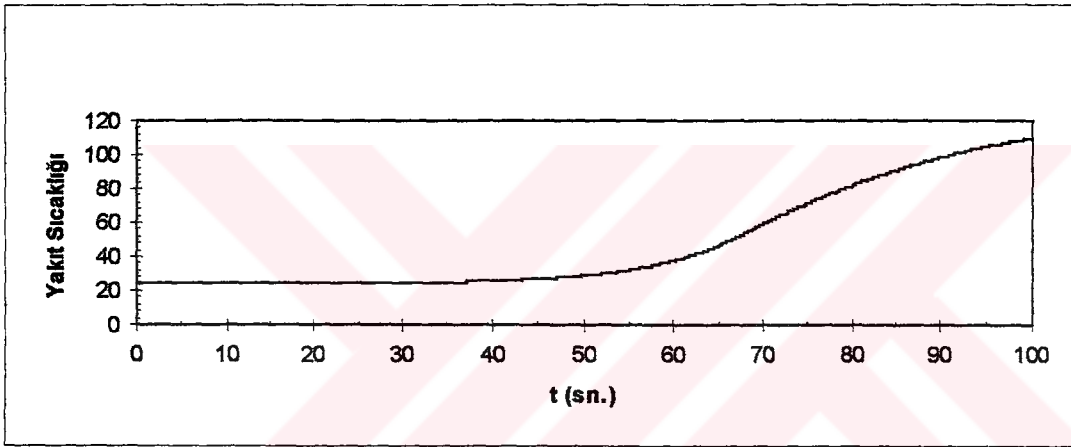
Tablo 5.3 Simülasyon başlangıç verileri

Başlangıç gücü : P_0	5 Kw
İstenen Peryot: τ	10 sn.
İstenen Güç : P_s	250 Kw
Örnekleme aralığı : T	0.05 sn.
İnceleme aralığı : T_{max}	0-100 sn.

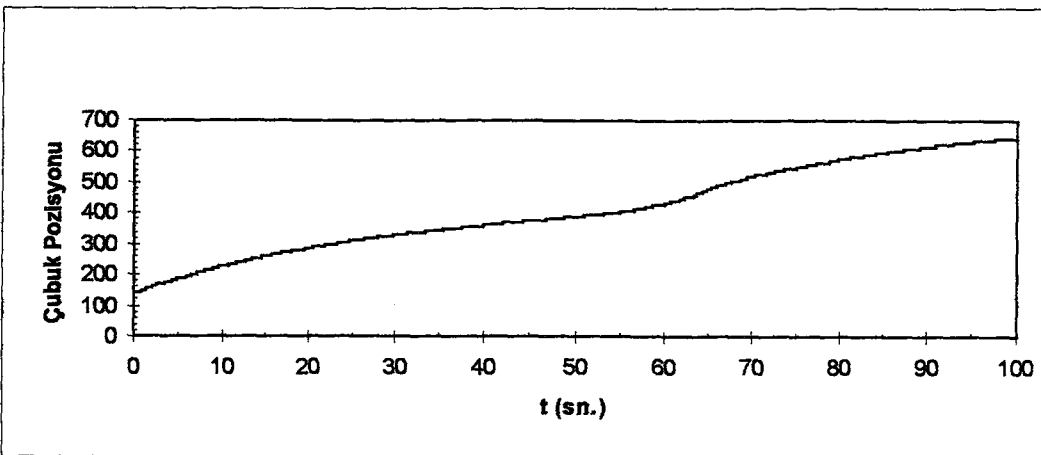
Reaktörün çıkış gücünün başlangıç kararlı hal değeri olan 5 Kw’dan 250 Kw’a çıkması istenmiştir. Bu çalışmada lineer olmayan YSA kontrolörünün eğitiminde istenen yörünge değeri verilmeden eğitilmiştir. Bu amaçla program çalıştırılmış, Şekil 5.19c’de görüldüğü gibi kontrol çubuğu çekilmeye başlanmıştır. Yakıt sıcaklığı 35 saniyeye kadar sabit bir değerde seyrederken (Şekil 5.19b), kontrol çubuğu daha yavaş çekilirken, 35. sn.’den sonra yakıt sıcaklığındaki artış nedeniyle biraz daha hızlı çekilmektedir. Şekil 5.19a’da görüldüğü gibi güç 250 Kw’a ulaştığında, Şekil 5.19d’de görüldüğü gibi 250 Kw civarında, yakıt sıcaklığı artmaya devam ettiğinden maksimum 1 Kw’lık sınımlar yapmaktadır. Şekil 5.19e’de görüldüğü gibi kontrolör tarafından üretilen kontrol sinyali gücün 250 Kw’a ulaştıktan sonra, yakıt sıcaklığındaki artışın sürmesine bağlı olarak kontrol çubuğunun çekilmesini sağlamaktadır.



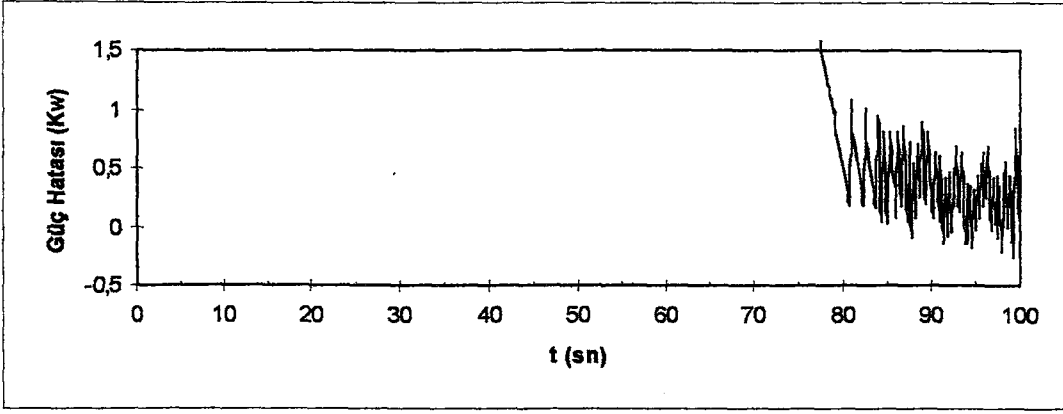
(a)



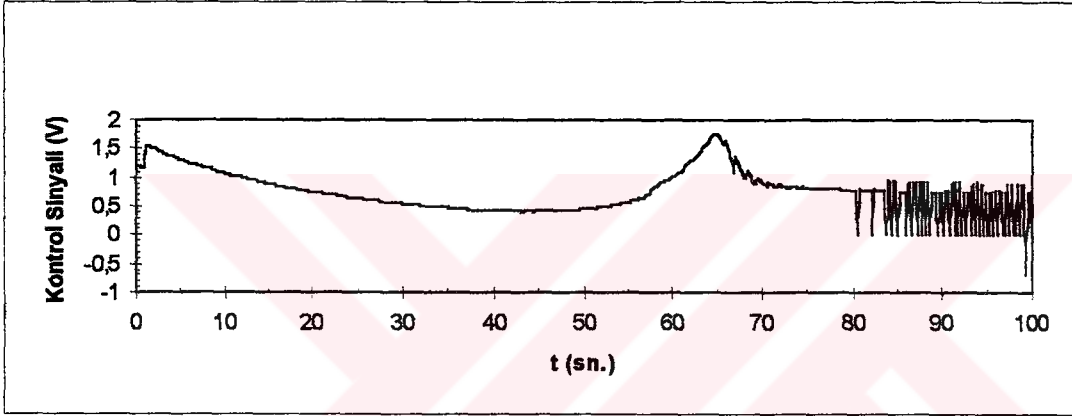
(b)



(c)



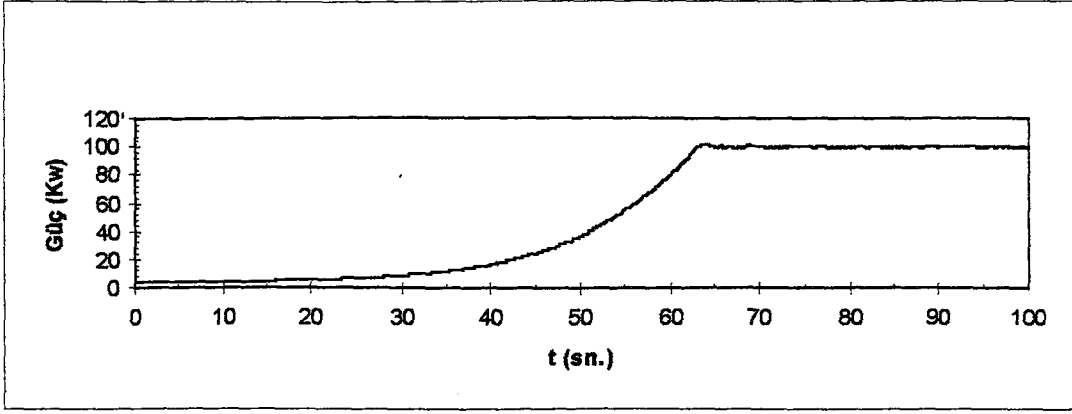
(d)



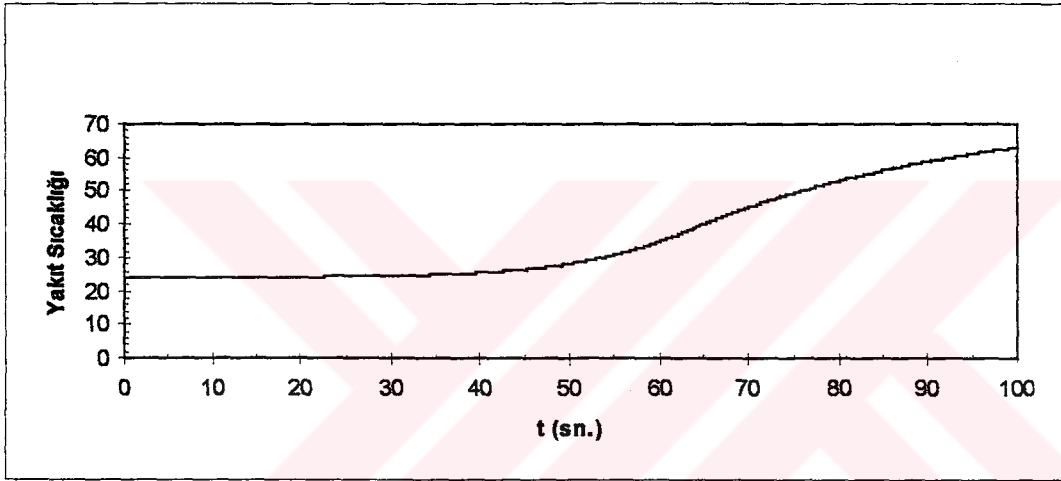
(e)

- Şekil 5.19 a) Güç Değişimi
 b) Yakıt Sıcaklığı (°C)
 c) Çubuk Pozisyonu
 d) Gücün Yörüngeden Sapması
 e) Kontrol Sinyali.

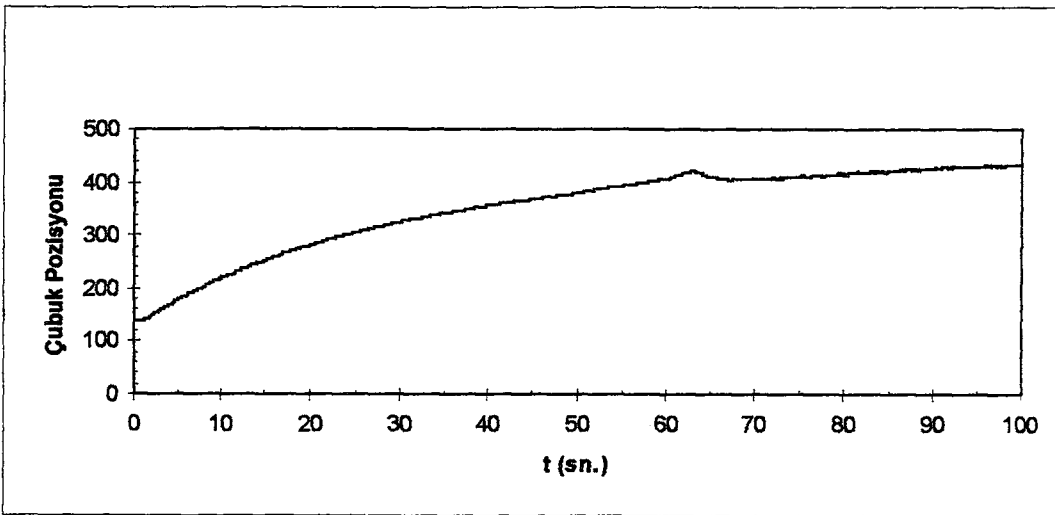
İkinci bir çalışmada ise, reaktör çıkış gücünün başlangıç kararlı hal değeri olan 5 Kw'dan 100 Kw'a çıkması istenmiştir. Bu amaçla çalıştırılan programa ait grafikler Şekil 5.20'de verilmiştir. Simülasyonda, Tablo 5.3'deki değerler kullanılmıştır. Şekil 5.20'deki grafikler Şekil 5.19'un benzeridir.



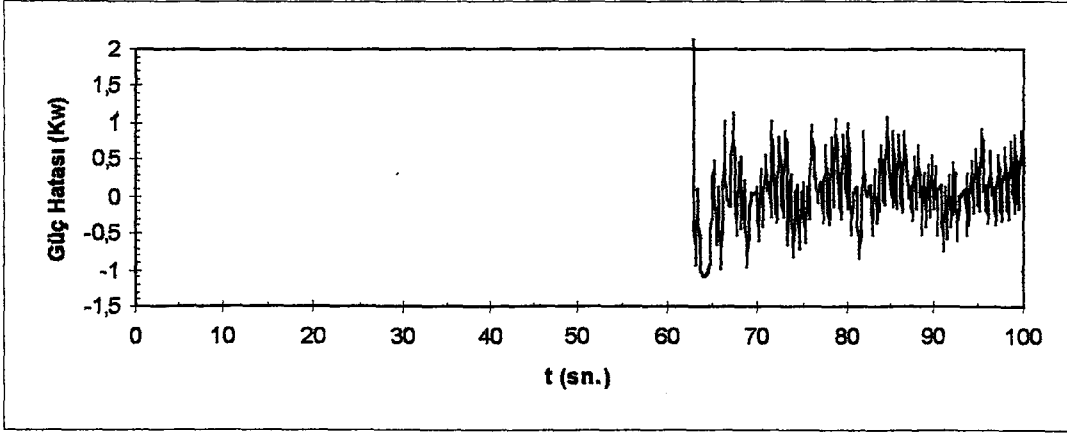
(a)



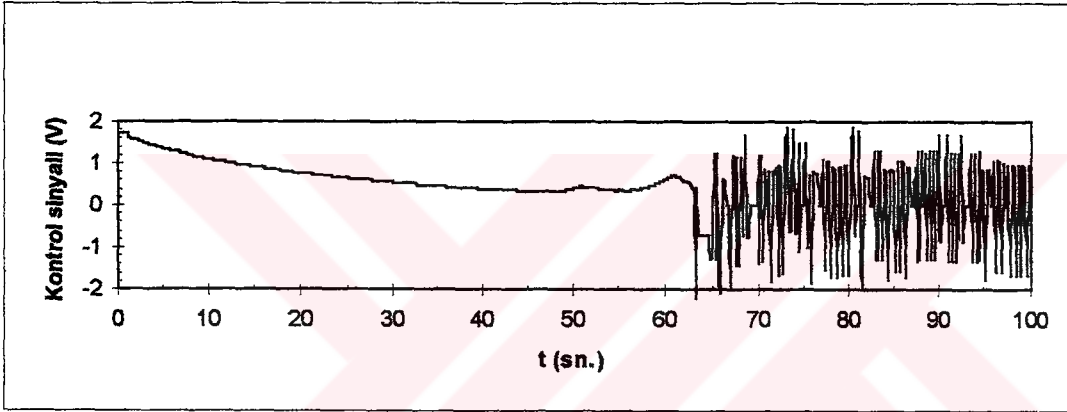
(b)



(c)



(d)



(e)

- Şekil 5.20 a) Güç Değişimi
b) Yakıt Sıcaklığı (°C)
c) Çubuk Pozisyonu
d) Gücün Yörüngeden Sapması
e) Kontrol Sinyali.

5.2.1 Başlangıç Gücü, İstenen Güç ve Örnekleme Aralığının Etkisinin İncelenmesi

Başlangıç gücü, istenen güç ve örnekleme aralığının etkisini incelemek için yapılan simülasyonla aşağıda verilmiştir.

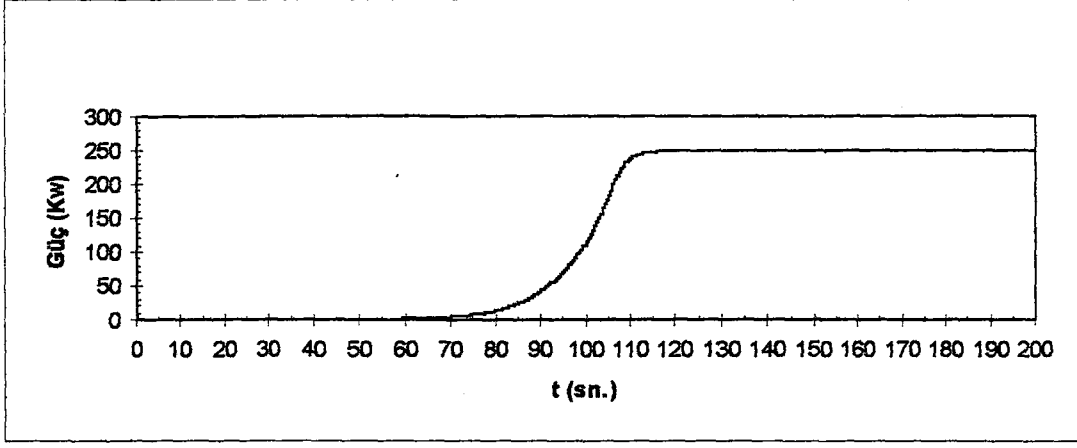
5.2.1.1 Başlangıç Gücünün Etkisinin İncelenmesi

Tasarlanan Y.S.A kontrolörün değişik başlangıç güçleri altında davranışını incelemek için yapılan simülasyonlara ilişkin grafikler Şekil 5.21, Şekil 5.22, Şekil 5.23 ve Şekil 5.24'de verilmiştir. Simülasyon, Tablo 5.3'deki değerler baz alınarak yapılmıştır. Yalnız 100 W ve 1 Kw için inceleme aralığı 200 saniye, başlangıç gücü ise 100 W, 1 Kw ve 5 Kw olarak seçilmiştir.

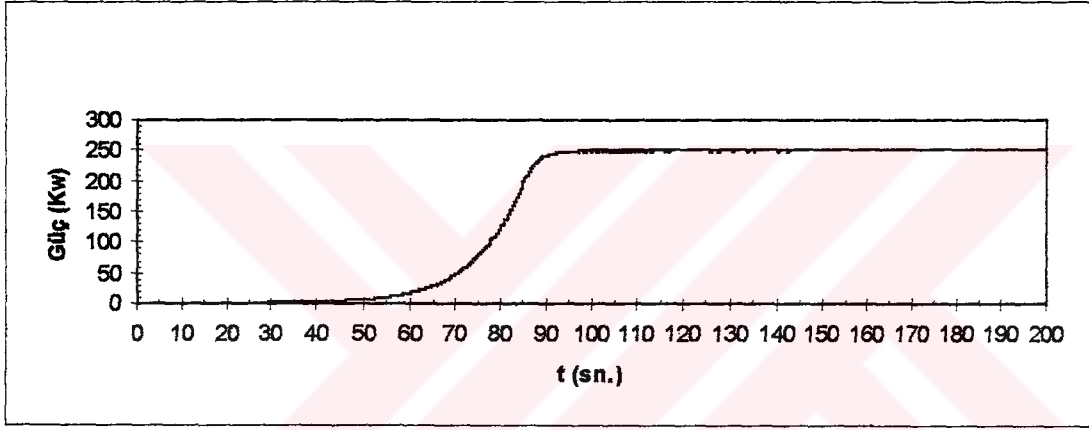
Şekil 5.21'de verilen güç değişimleri incelendiğinde; kontrolörün reaktör gücünü istenilen düzeye çıkarabildiği görülmektedir. Fakat güç 250 Kw'a çıktıktan sonra, yakıt sıcaklığının artması durmadığından çubuk hala çekilmeye devam etmektedir (Şekil 5.23). Bu yüzden, reaktör istenilen güç değeri olan 250 Kw civarında salınımlar yapmaktadır. Oluşan bu güç hatası Şekil 5.22'de görüldüğü gibi maksimum 1 Kw civarındadır. Güç değişimi sabit peryotla (10 sn.) yapıldığı için başlangıç gücüne bağlı olarak istenilen güç düzeyine çıkış süresi değişmektedir.

Şekil 5.24'de görüldüğü gibi kontrol çubuğunu hareket ettirmek için kontrolör tarafından üretilen kontrol sinyali; istenen güç düzeyine ulaşıldıktan sonra yakıt sıcaklığındaki artışın devam etmesinden dolayı -0.5 V. ile 1 V. arasında salınım yapmaktadır.

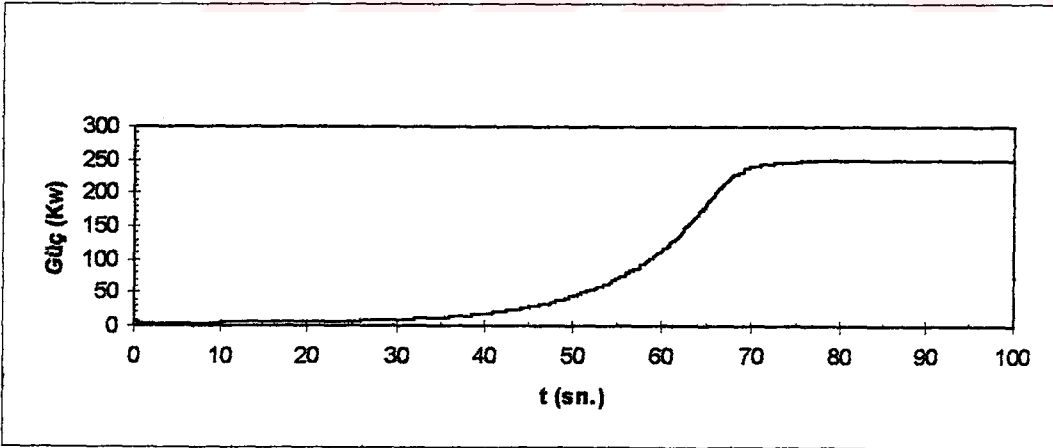
Sonuç olarak, tasarlanan kontrolörün değişik başlangıç güçlerinde, istenilen güç düzeyine yörünge kullanmadan çıkabilmekte, fakat 1 Kw civarında bir hata oluşmaktadır. Aslında kontrolör reaktörü kontrol etmeye devam ettiğinde kararlı hal hatası 400 sn. sonunda 0.1 Kw'a kadar inmektedir. Oluşan bu kararlı hal hatasını ortadan kaldırmak için reaktör istenilen güç değerine ulaştıktan sonra bir PID kontrolör ile bu kararlı hal hatası ortadan kaldırılabilir.



(a)



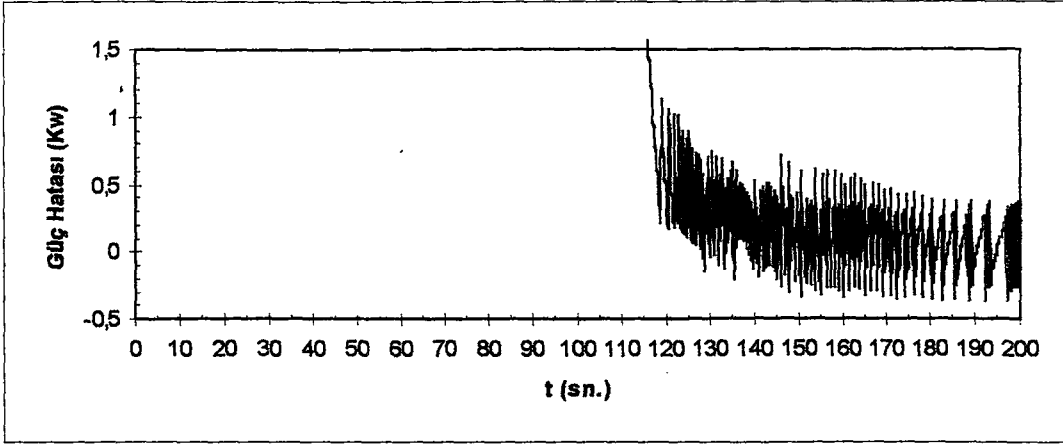
(b)



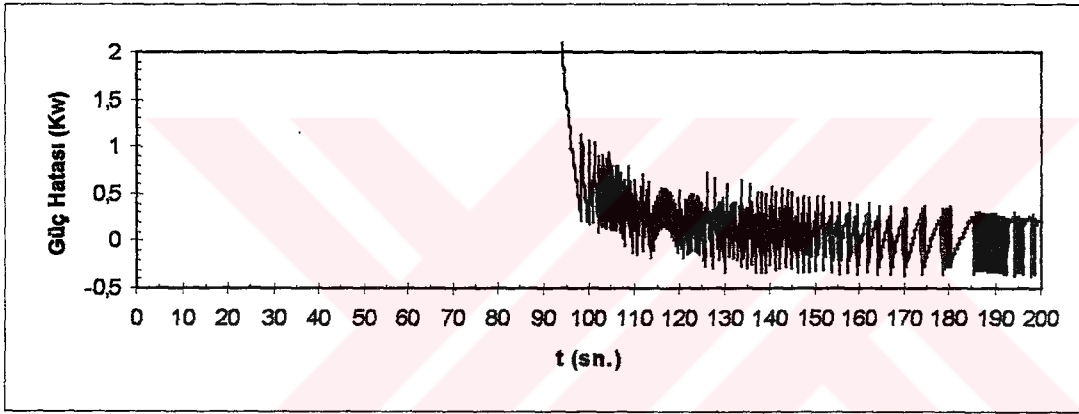
(c)

Şekil 5.21 Güç Değişimi

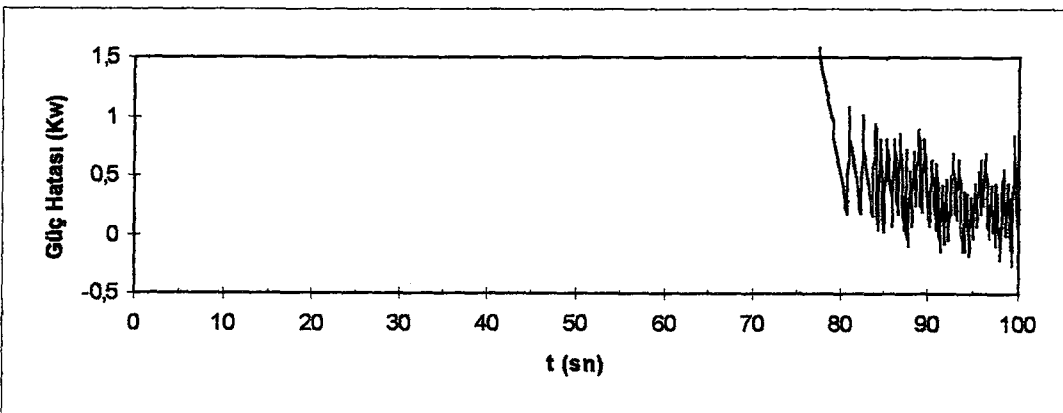
a) $P_0 = 100 \text{ W}$ b) $P_0 = 1 \text{ Kw}$ c) $P_0 = 5 \text{ Kw}$



(a)



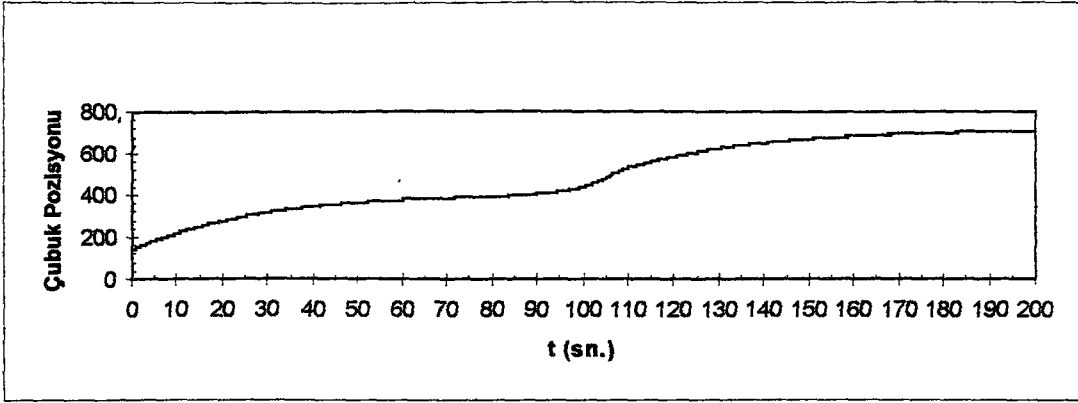
(b)



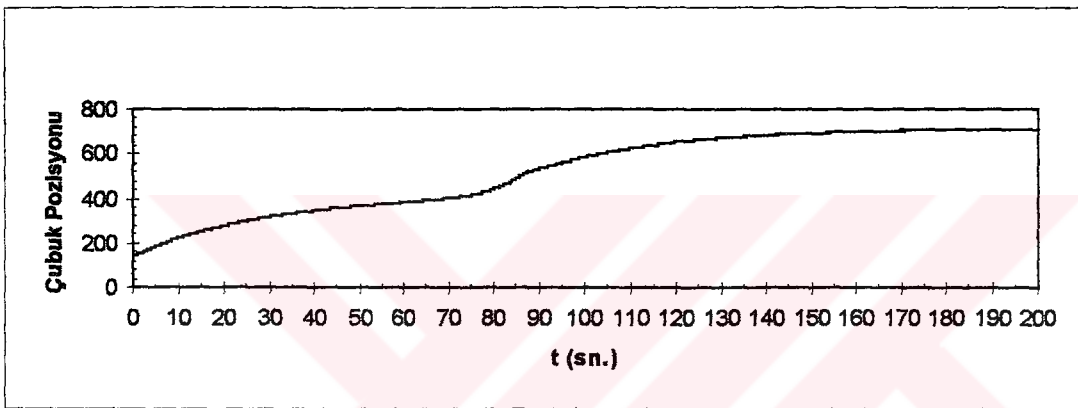
(c)

Şekil 5.22 250 Kw'daki Güç Hataları

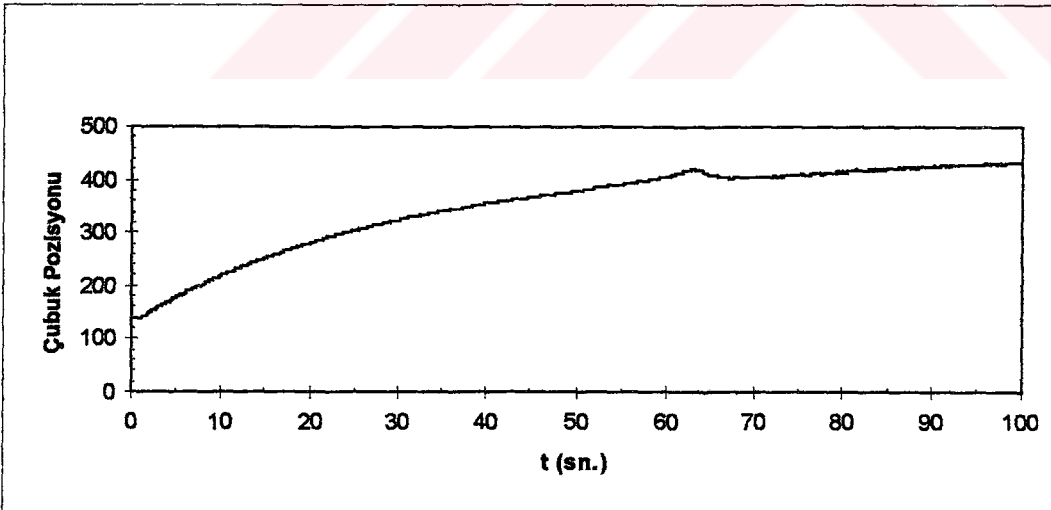
a) $P_0 = 100 \text{ W}$ b) $P_0 = 1 \text{ Kw}$ c) $P_0 = 5 \text{ Kw}$



(a)



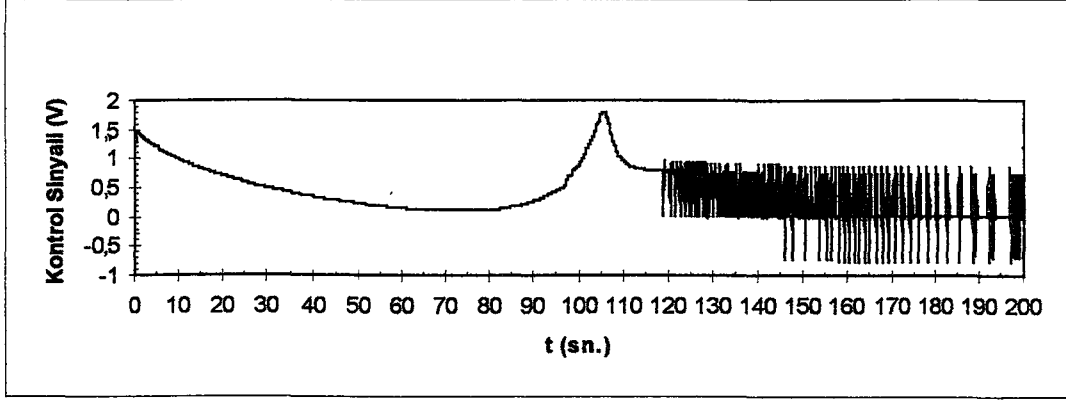
(b)



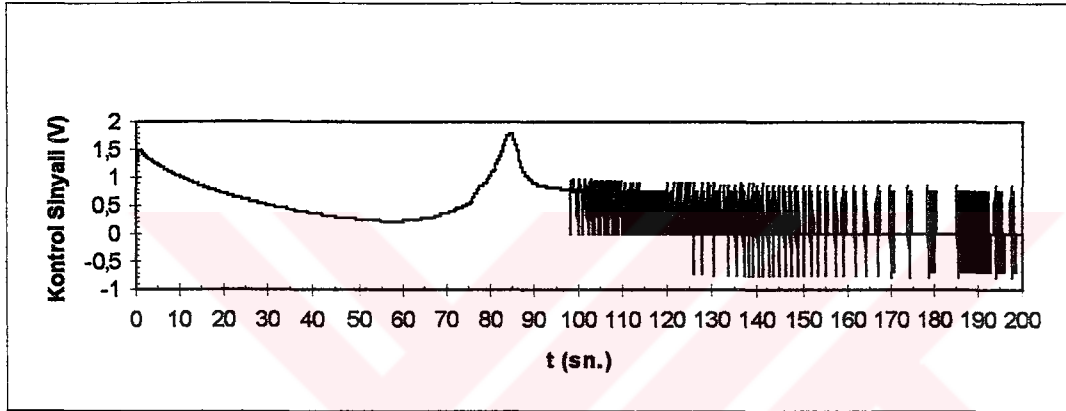
(c)

Şekil 5.23 Çubuk Pozisyonları

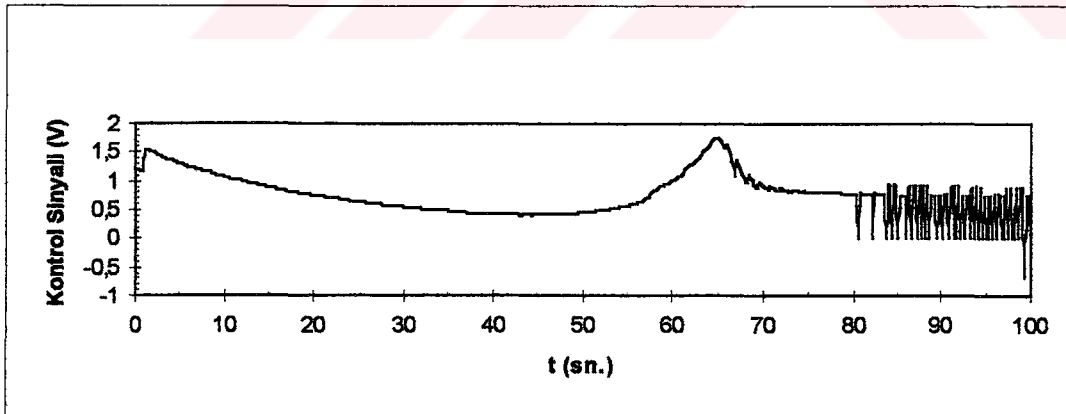
a) $P_0 = 100 \text{ W}$ b) $P_0 = 1 \text{ Kw}$ c) $P_0 = 5 \text{ Kw}$



(a)



(b)



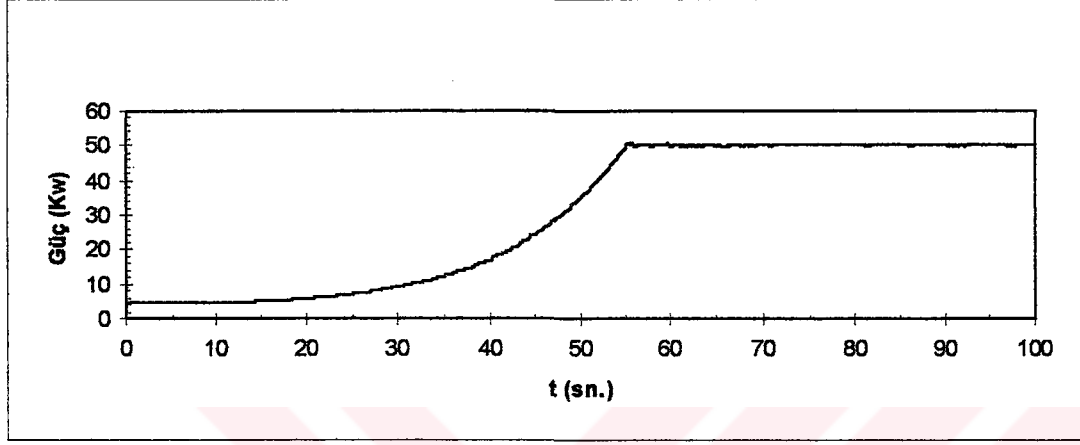
(c)

Şekil 5.24 Kontrol Sinyalleri

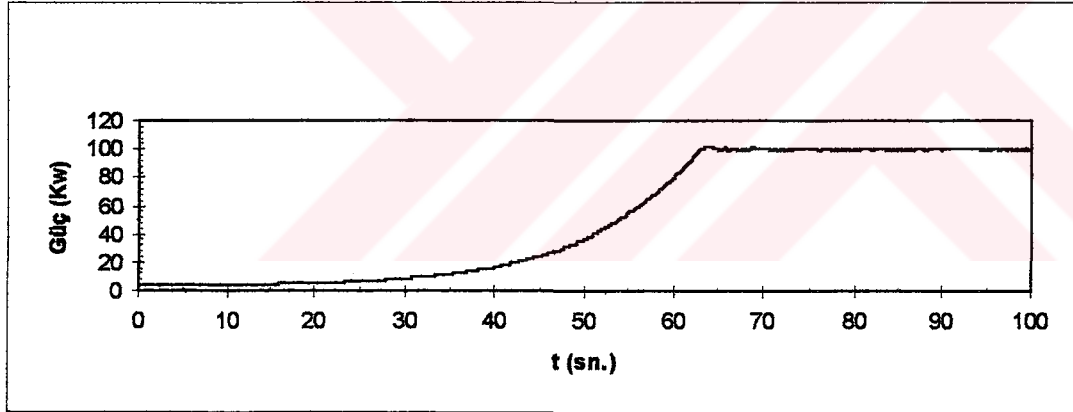
a) $P_0 = 100 \text{ W}$ b) $P_0 = 1 \text{ Kw}$ c) $P_0 = 5 \text{ Kw}$

5.2.1.2 İstenen Gücün Etkisinin İncelenmesi

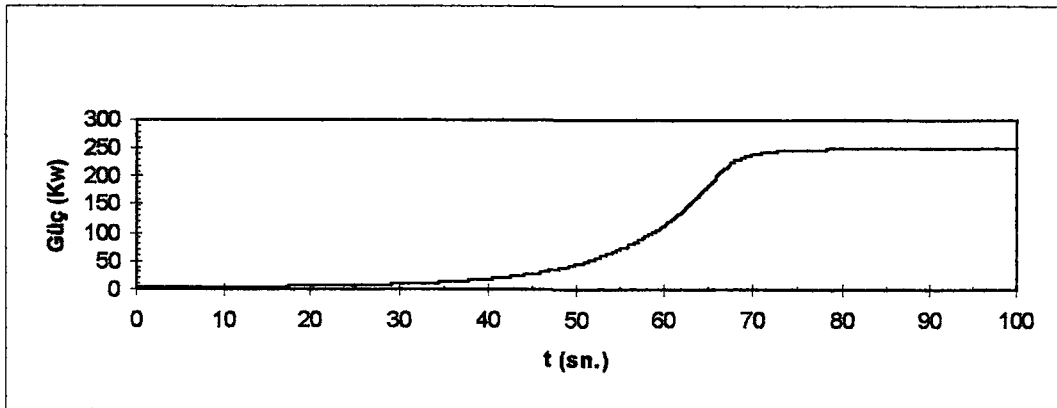
Tasarlanan yapay sinir ağı kontrolörünün değişik güç seviyelerine çıkarken göstereceği davranışı izlemek için yapılan simülasyonlara ilişkin grafikler Şekil 5.25 , Şekil 5.26, Şekil 5.27 ve Şekil 5.28’de gösterilmektedir. Simülasyon, Tablo 5.3’deki değerler baz alınarak yapılmıştır. İstenilen güç ise 50 Kw, 100 Kw ve 250 Kw olarak seçilmiştir.



(a)

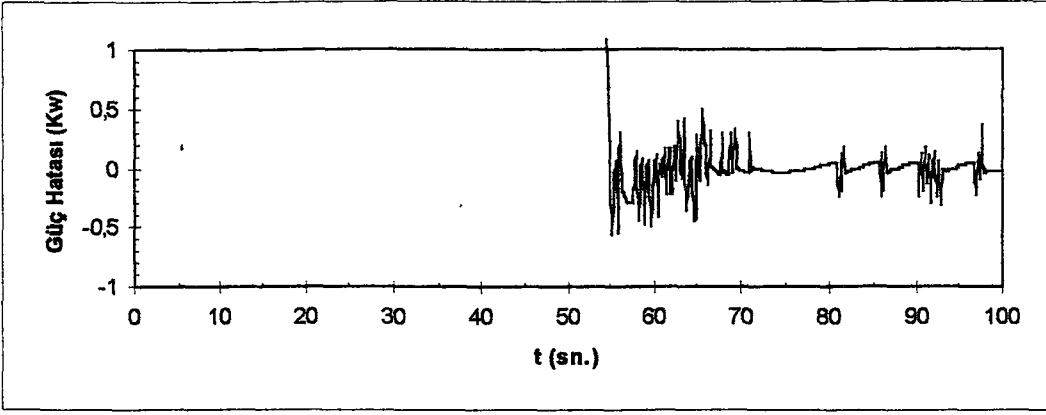


(b)

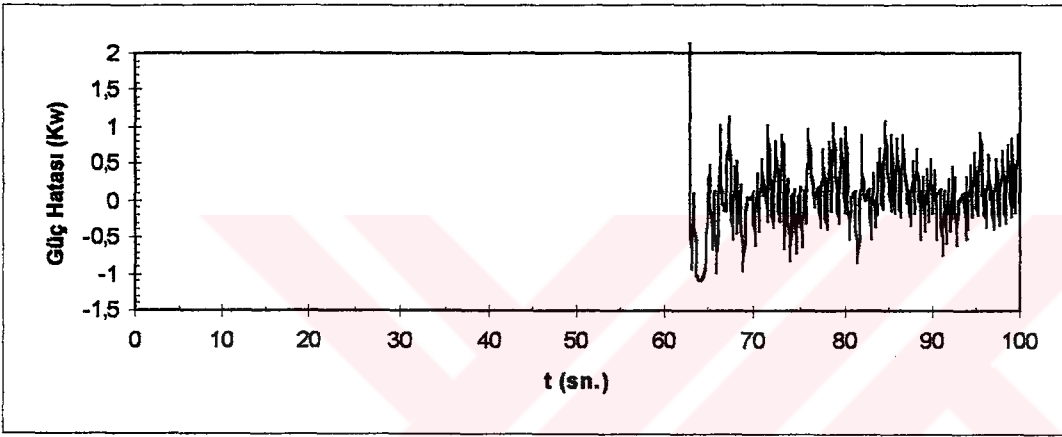


(c)

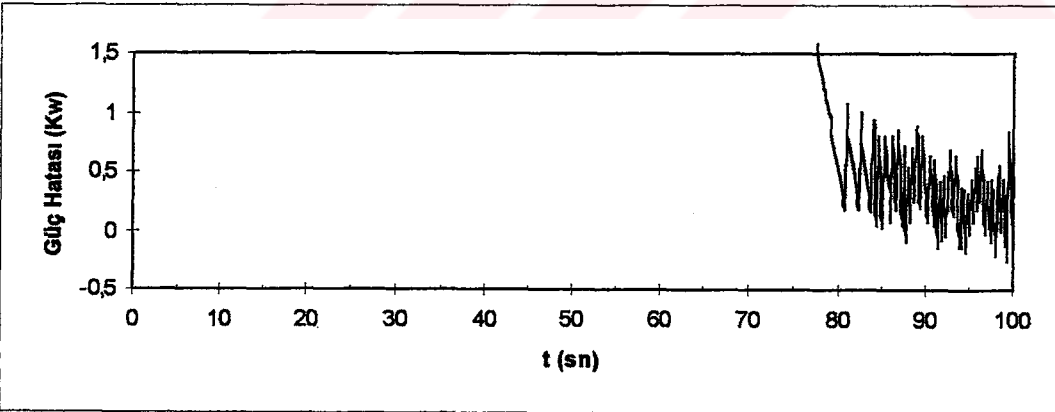
Şekil 5.25. Güç Değişimi a) $P_{\max} = 50$ Kw, b) $P_{\max} = 100$ Kw, c) $P_{\max} = 250$ Kw



(a)



(b)



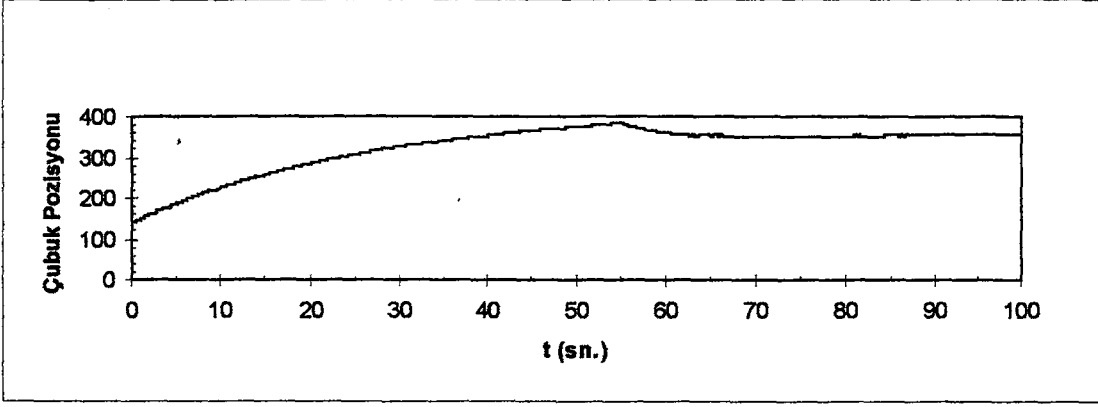
(c)

Şekil 5.26. İstenen Güç Seviyesinde Oluşan Güç Hatası

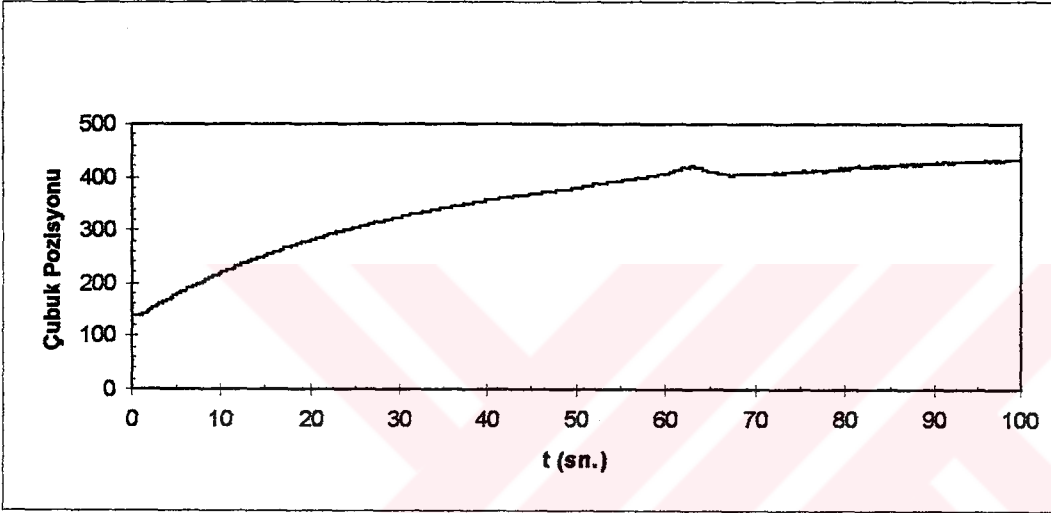
a) $P_{max} = 50 \text{ Kw}$

b) $P_{max} = 100 \text{ Kw}$

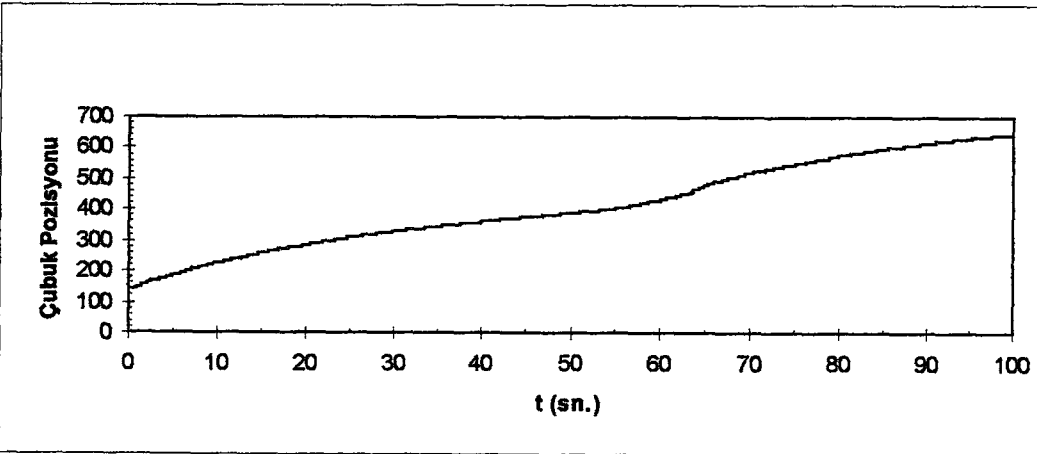
c) $P_{max} = 250 \text{ Kw}$



(a)



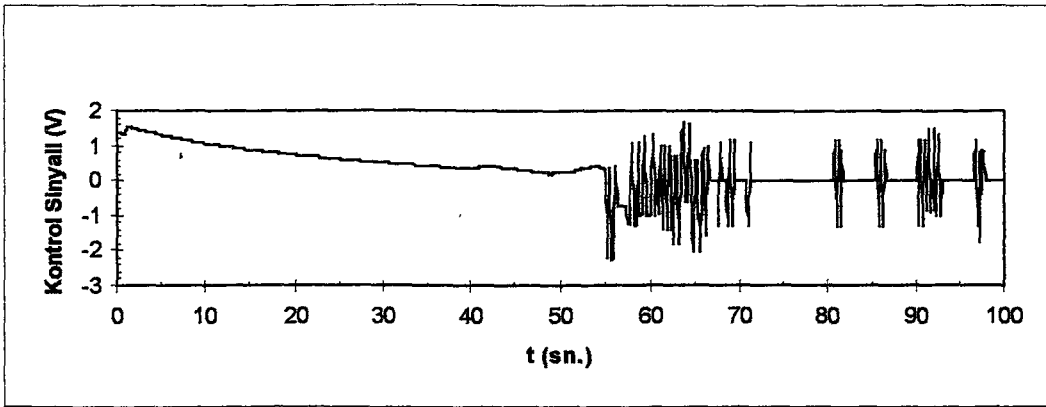
(b)



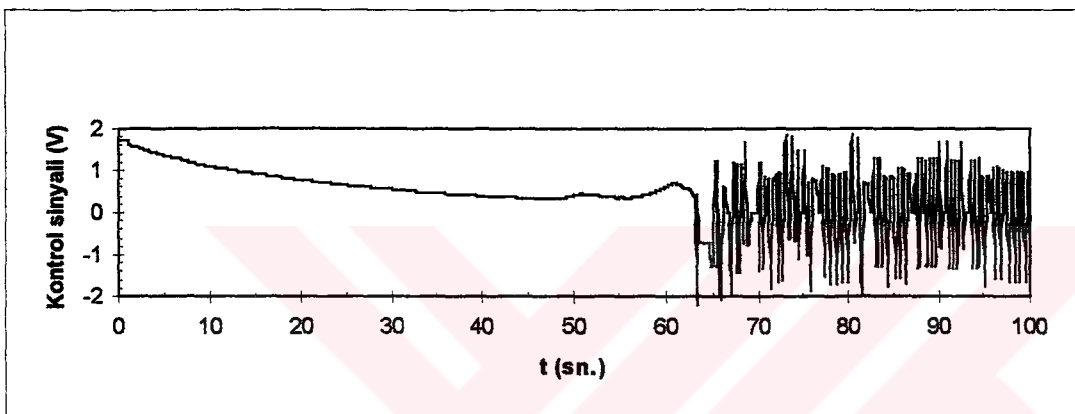
(c)

Şekil 5.27. Çubuk Pozisyonu

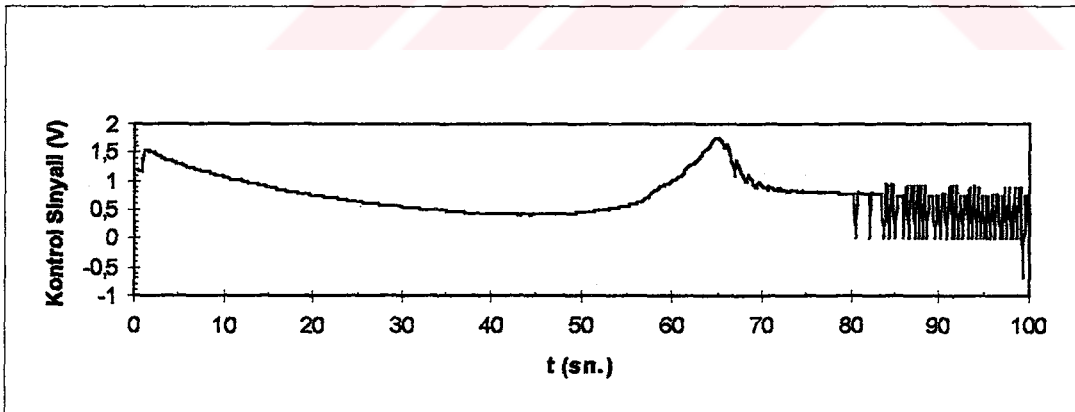
a) $P_{max} = 50 \text{ Kw}$ b) $P_{max} = 100 \text{ Kw}$ c) $P_{max} = 250 \text{ Kw}$



(a)



(b)



(c)

Şekil 5.28. Kontrol Sinyali

a) $P_{\max} = 50 \text{ Kw}$

b) $P_{\max} = 100 \text{ Kw}$

c) $P_{\max} = 250 \text{ Kw}$

Şekil 5.25 'de görüldüğü gibi güç değişimleri incelendiğinde; kontrolörün reaktörü istenilen güç düzeyine çıkardığı, fakat reaktör 50 Kw istenen güç seviyesine geldiğinde 0.5 Kw, 100 ve 250 Kw istenen güç seviyesine geldiğinde 1 Kw civarında salınım yapmaktadır. Bunun nedeni ise reaktörde yakıt sıcaklığının hala artarak, çubuğun çekilmeye devam etmesinden kaynaklanmaktadır. Güç değişimi sabit periyotla yapıldığı için istenilen güce bağlı olarak güce çıkış süresi değişmektedir. İstenilen güç 50 Kw alındığında bu süre 55 sn. iken 100 Kw alındığında 65 sn. ve 250 Kw alındığında ise 80 sn. olmaktadır.

Şekil 5.26'da reaktörün istenen güç değerine çıktıktan sonra oluşan güç hatası görülmektedir. Oluşan bu güç hatası reaktör kontrolör tarafında simüle edilmeye devam ettiğinde, 400 sn. sonunda 200 Watt'lık bir hatanın oluştuğu görülmüştür.

Şekil 5.27'de 50 Kw, 100 Kw ve 250 Kw güç düzeyine çıkmada kontrol çubuğunun pozisyonu ile ilgili grafikler görülmektedir. Kontrol çubuğu Şekil 5.27a, b ve c'de başlangıçta benzer şekilde hareket etmektedir. Hızlı çekilen çubuk reaktör istenilen güç seviyesine ulaştıktan sonra yakıt sıcaklığının daha yavaş artmasına bağlı olarak kontrol çubuğu da daha yavaş çekilmektedir. 250 Kw seviyesinde sıcaklığın etkisi daha fazla olduğundan, kontrol çubuğu bu güç seviyesinde 50 Kw ve 100 Kw güç seviyelerine nazaran daha hızlı çekilmektedir. Bunun nedeni ise sıcaklık etkisi ile oluşan negatif geribeslemeyi, kontrol çubuğunun dengelemeye çalışmasındandır.

Şekil 5.28'de ise kontrol çubuğunu hareket ettirmek için kontrolör tarafından üretilen kontrol sinyalleri görülmektedir. Reaktörün istenilen güç değerine ulaştıktan sonra sıcaklığın artmaya devam etmesinden dolayı kontrol sinyali maksimum +2 ve -2 volt arasında salınımlar yapmaktadır.

Sonuç olarak, tasarlanan kontrolörün değişik güç düzeylerine çıkabildiği, fakat bu düzeylerde maksimum 1 Kw'lık kararlı hal hatasının oluştuğu görülmüştür. Bu hata, kontrolörün reaktörü kontrol etmeye devam ettiği sürece 400. saniyede 200 Watt'a düştüğü deneylerde görülmüştür. Reaktörün istenilen güç düzeyine çıktıktan sonra oluşan kararlı hal hatası bir PID kontrolör ile giderilebilir. Can (1991) tarafından İTÜ TRIGA Mark-II Reaktörü için bir PID kontrolör önermiştir. Çalışmamızda reaktör

gücünde meydana gelen salınımların giderilmesinde yardımcı kontrolör olarak Can (1991) tarafından önerilen kontrolör kullanılabilir.

5.2.1.3 Örnekleme Aralığının Etkisinin İncelenmesi

Y.S.A. kontrolörünün değişik örnekleme aralıklarındaki davranışını izlemek için yapılan simülasyonlara ilişkin grafikler Şekil 5.29, Şekil 5.30, Şekil 5.31 ve Şekil 5.32'de verilmiştir. Simülasyon, Tablo 5.3'deki değerler kullanılarak yapılmıştır. Yalnız inceleme aralığı 0-200 sn. seçilmiştir. Örnekleme aralığı ise 0.2 sn., 0.4 sn. ve 0.8 sn. olarak seçilmiştir.

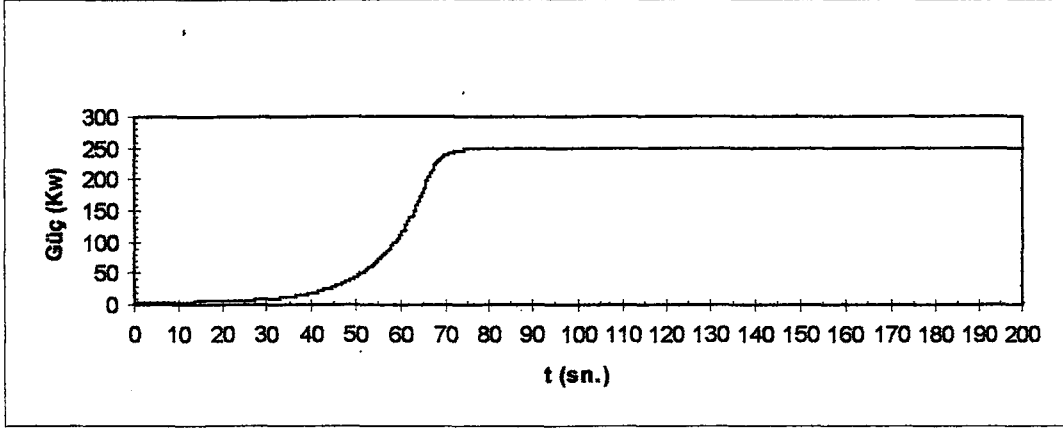
Şekil 5.29 'da verilen güç değişimleri incelendiğinde; kontrolörün reaktör gücünü istenilen düzeye çıkarabildiği, fakat istenen güç seviyesine çıktuktan sonra, güç; hem sıcaklığın artmaya devam etmesine, hem de örnekleme aralığının büyümesine bağlı olarak 0.2 sn. örnekleme aralığında maksimum 1Kw , 0.4 sn.'de maksimum 2 Kw ve 0.8 sn.'de maksimum 3.5 Kw civarında salınımlar yapmaktadır (Şekil 5.30).

Şekil 5.31'de kontrol çubuğunun pozisyonu ile ilgili grafikler verilmiştir. Kontrol çubuğu Şekil 5.31a, b ve c'de benzer şekilde hareket etmektedir. Hızlı çekilen çubuk gücün istenilen düzeye geldiğinde, daha yavaş çekilmeye devam etmektedir ve örnekleme aralığına bağlı olarak çubuk profilinde bazı bozulmalar meydana gelmektedir.

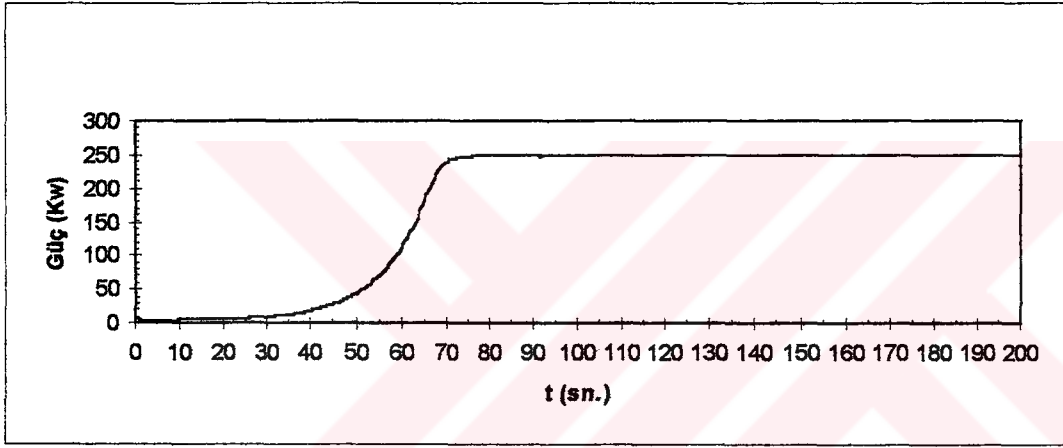
Şekil 5.32'de ise kontrol çubuğunu hareket ettirmek için kontrolör tarafından üretilen kontrol sinyalleri görülmektedir. Grafiklerde görüldüğü gibi, istenilen güç düzeyine erişen reaktör; sıcaklık artışının devam etmesinden dolayı, kontrol sinyali kontrol çubuğunu hareket ettirmek için maksimum -1 ile +1 V. civarında salınımlar yapmaktadır.

Sonuç olarak, tasarlanan Y.S.A. kontrolörü değişik örnekleme aralıklarında reaktörü kontrol etmektedir. Yalnız sıcaklığın etkisinden, istenen güç düzeyine ulaşan reaktör gücü salınımlar yapmaktadır. Reaktör kontrol edilmeye devam edilirse; 400. saniye sonunda bu salınımlar 200 Watt'a kadar düşmektedir. Bu salınımları önlemek için

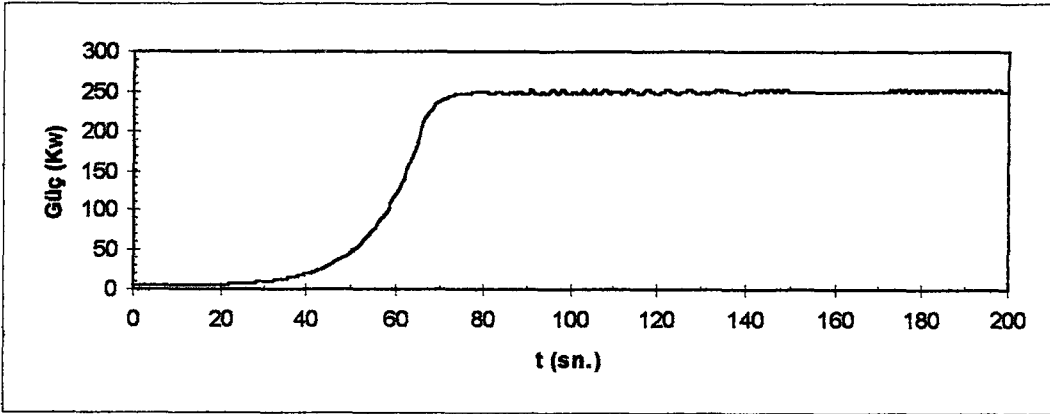
reaktörün istenilen güç düzeyine çıktıktan sonra PID kontrolör kullanılarak bu salınımlar engellenebilir.



(a)



(b)



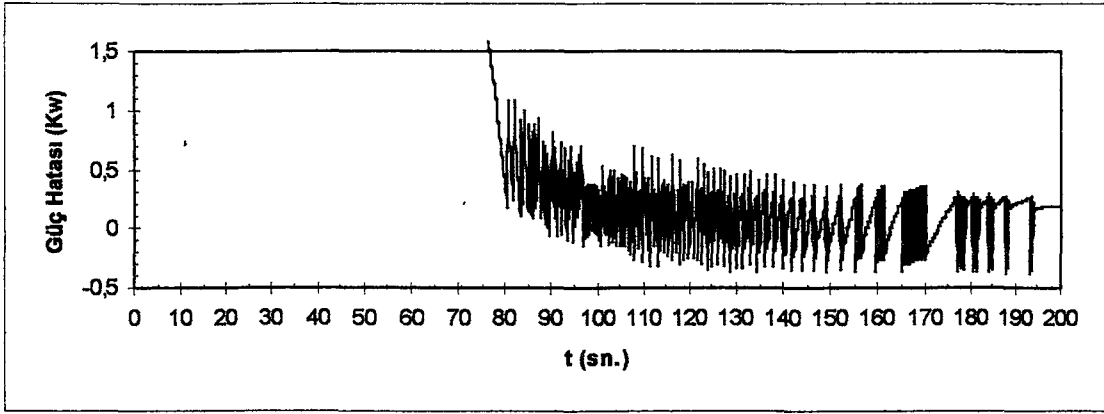
(c)

Şekil 5.29. Güç Değişimleri

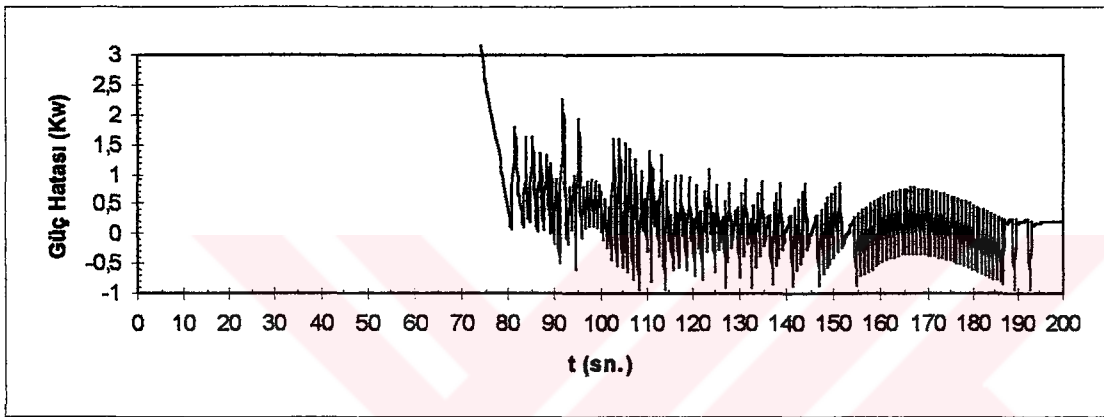
a) $T = 0.2$ sn.

b) $T = 0.4$ sn.

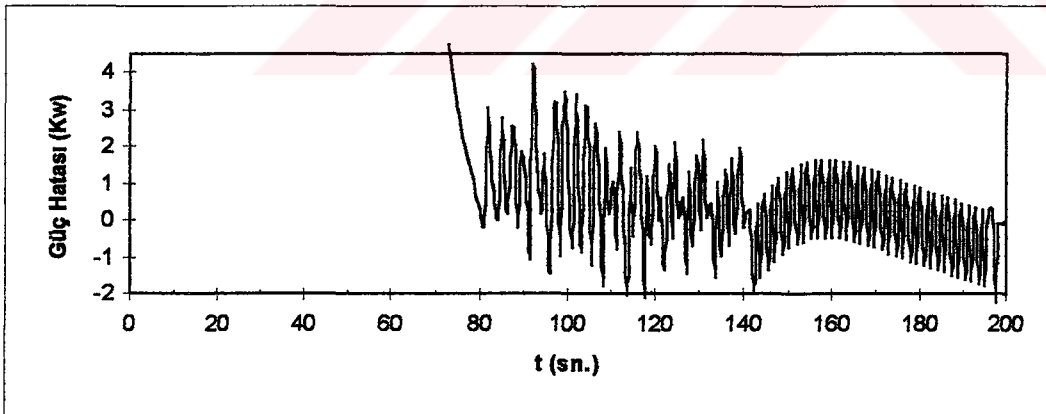
c) $T = 0.8$ sn.



(a)



(b)



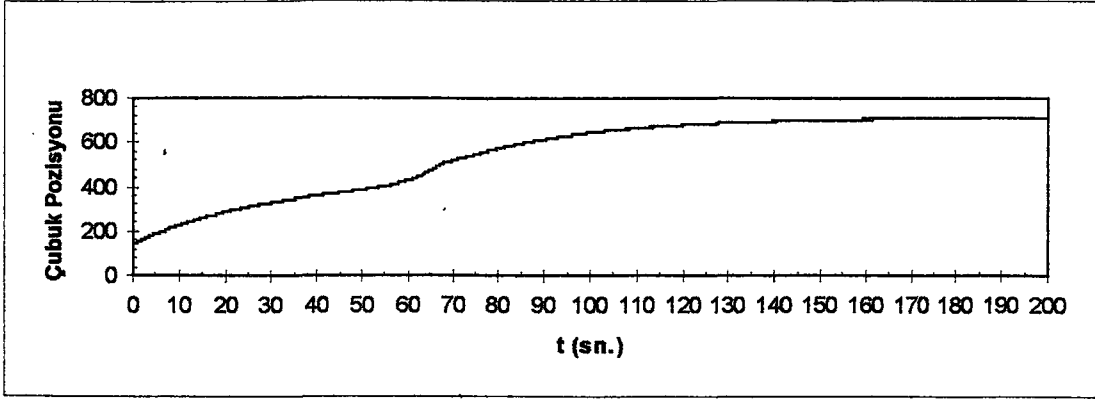
(c)

Şekil 5.30. İstenen Güç Düzeyine Ulaştıktan Sonra Oluşan Güç Hataları

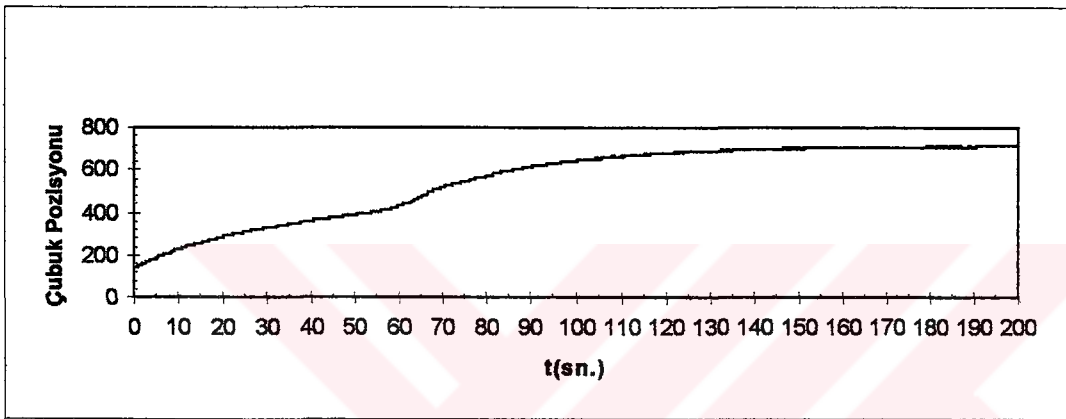
a) $T = 0.2$ sn.

b) $T = 0.4$ sn.

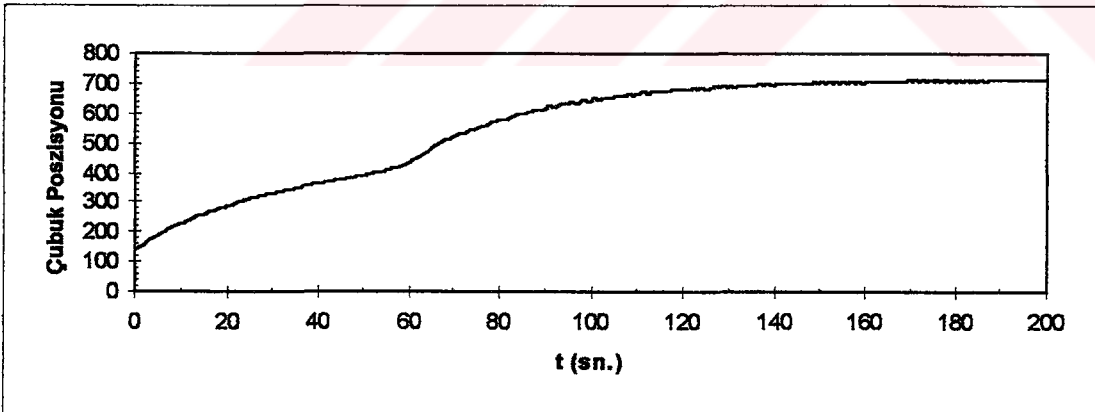
c) $T = 0.8$ sn.



(a)



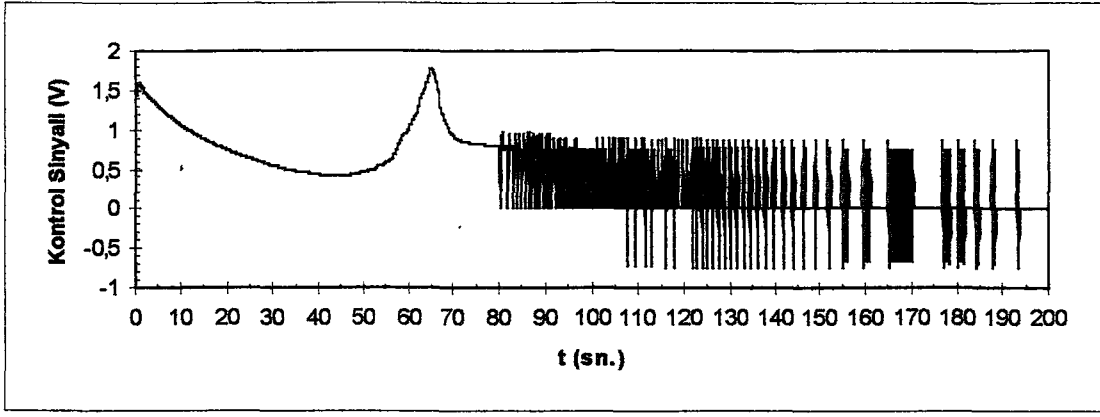
(b)



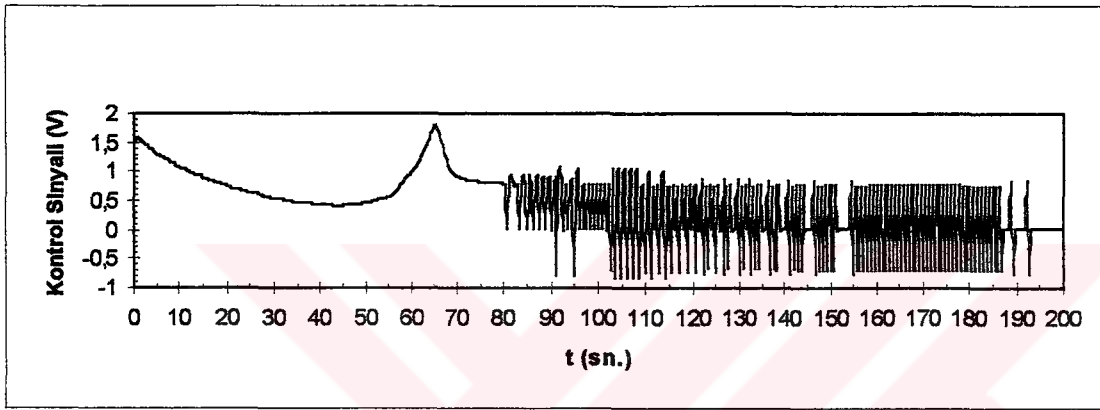
(c)

Şekil 5.31. Çubuk Pozisyonları

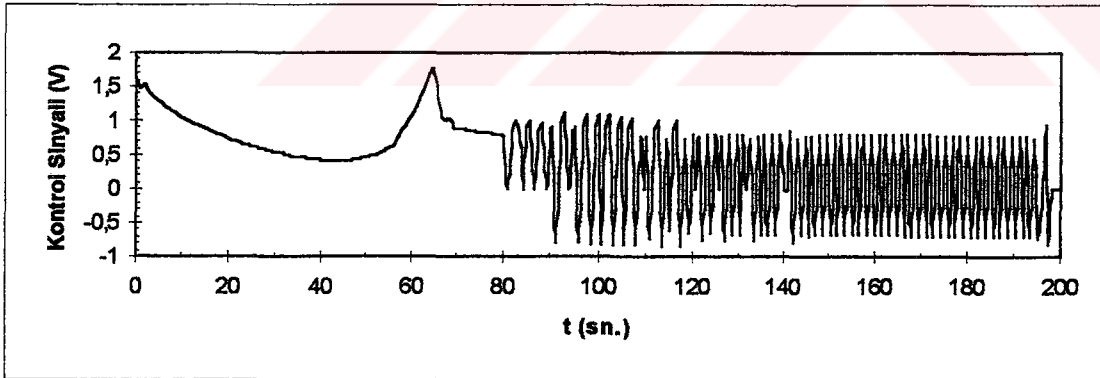
a) $T = 0.2$ sn.b) $T = 0.4$ sn.c) $T = 0.8$ sn.



(a)



(b)



(c)

Şekil 5.32. Kontrol Sinyalleri

a) $T = 0.2$ sn.b) $T = 0.4$ sn.c) $T = 0.8$ sn.

KAYNAKLAR

- 1- Ahmed, M. S. and Tasadduq, I. A., 1994. Neural-net controller for nonlinear plants: design approach through linearisation. IEE Proc.-Control Theory Appl., Vol.141, No.5, 315-322.
- 2- Baba, F., 1995. İTÜ TRIGA Mark-II Reaktörünün Bulanık Kontrolü. Marmara Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.
- 3- Bittanti, S. and Piroddi, L., 1994. GMV technique for nonlinear control with neural networks. IEE Proc.-Control Theory Appl., Vol.141, No.2, 57-69.
- 4- Can, B. ve Baba, F., 1991. İTÜ TRIGA Mark-II Reaktörünün Kontrolüne ilişkin yeni bir öneri. V. Ulusal Nükleer Bilimler Kongresi, E.Ü., N.B.E., Bildiri özetleri, s.40 İzmir.
- 5- Can, B., 1992. The Optimal Control of İTÜ TRIGA Mark-II Reactor. Twelfth European TRIGA User's Conference, NRI Bucuresti-Pitesti Romania.
- 6- Can, B., 1993. İTÜ TRIGA Mark-II reaktörü için bir kontrolör tasarımı. Elektrik Müh. 5.Ulusal Kongresi, KTÜ, Bildiri Kitabı, Cilt. 1, s.147-151, Trabzon.
- 7- Cichocki, A. and Unbehauen, R., 1994. Simplified Neural Networks for Solving Linear Least Squares and Total Least Squares Problems in Real Time. IEEE Transactions on neural networks. Vol.5, No.6, pp. 910-923.
- 8- Erkan, K., 1994. İTÜ TRIGA Mark-II Reaktörünün Özyarlı Adaptif Kontrolü. M.Ü. Fen Bilimleri Enst., Doktora Tezi, İstanbul.
- 9- Freeman, J. A. and Skapura, D. M., 1992. Neural Networks Algorithms, Applications and Programming Techniques. Addison-Wesley Publishing Co., ISBN 0-201-51376-5, U.S.A.

- 10- Güngördü, E. vd, 1989. İTÜ TRIGA Mark-II Reaktörü için bir lineer olmayan kontrol stratejisi. III. Ulusal Nükleer Bilimler Kongresi, İTÜ N.E.E. Bildiri Kitabı, Cilt 1, s.400-405, İstanbul.
- 11- Hecht, R. and Nielsen, 1990. Neurocomputing. HNC, Inc. and University of California, Addison-Wesley Publishing Co., Sandiago.
- 12- Huang, S. C. and Huang, Y.F., 1990. Learning Algorithms for Perceptrons Using Back-Propagation with Selective Updates. IEEE Control Systems Magazine, pp. 56-61.
- 13- Hunt, K.J. vd, 1992. Neural Networks for Control System- A Survey. Automatica Vol.28, No.6, pp. 1083-1112.
- 14- Jones, L. K., 1990. Constructive Approximation for Neural Networks by Sigmoidal Functions. IEEE Proc., Vol. 78, No. 10, pp. 1586-1589.
- 15- Lightbody, G. and Irwin, G. W., 1995. Direct neural model reference adaptive control. IEE-Proc.-Control Theory Appl., Vol.142, No.1, pp. 31-43.
- 16- Nelson, M. M. and Illingworth, W. T., 1991. A Practical Guide to Neural Nets. Addison-Wesley Publishing Co., ISBN 0-201-52376-0 Massachusetts.
- 17- Nordita, J. H., 1991. Introduction to the Theory of Neural Computation. Duke University and the Santa Fe Institute Lecture Notes Addison-Wesley Publishing Co.
- 18- Özkan, İ. , 1995. İTÜ TRIGA Mark-II Reaktörünün iki boyutlu bilgisayar simülasyonu. İTÜ Nükleer Enerji Enstitüsü, Yüksek Lisans Tezi, İstanbul.
- 19- Öztemel, E., 1995. Bilgisayarlarda Öğrenme ve Yapay Sinir Ağları. F.Technologies dergisi, s. 58-60.

- 20- Raol, J. R., 1994. Neural network based parameter estimation of unstable aerospace dynamic systems. IEE Proc.-Control Theory Appl., Vol. 141, No.6, pp. 385-3883.
- 21- Wang, H.vd., 1994. Neural network modelling of unknown nonlinear systems subject to immeasurable disturbances IEE Proc.-Control Theory Appl., Vol. 141, No.4, pp.216-222.
- 22- Yang, Y.Y. and Linkens, D.A., 1994. Adaptive neural-network-based approach for the control of continuously stirred tank reactor. IEE Proc.-Control Theory Appl., Vol.141, No.5, pp.341-349.



EK A: PROGRAM LİSTELERİ

Bu ekte; önerilen lineer olmayan kontrolün eğitiminin yapıldığı, PASCAL programlama dilinde hazırlanan program listeleri verilmektedir.

* * *

```
program ST3;
  uses net3, crt;
  type
  refTip=record
    guc,cubuk,sicaklik,periyot,iguc,iperiyot,kontrolU:real;
  end;
```

```
refdatadizi=array[1..1000] of refTip;
```

```
var
pref,ep, wi,w:real;
si,sk,smax:integer;
dosya,refdosya:text;
ppr, toper,topmin:real;
yt,yte1,yte2,ute2,ute1,ut:real;
bar:array[1..8] of integer;
kk,kn, ornek,i,j:integer;
  net:netTip;
  xp:dizitip;
  sref:integer;
  target:dizitip;
netdos:file of netTip;
d:array[1..1] of ^refdatadizi;
s:array[1..2000] of integer;
pr:real;
function fdot(x:real;tr:integer):real;
begin
  case tr of
    1: fdot:=1;
    2: if x>=0 then fdot:=sqr(1-x) else fdot:=sqr(1+x);
    3: fdot:=x*(1-x);
    4: if x>0 then fdot:=1 else fdot:=0;
    5: if (x>=-1) and (x<=1) then fdot:=1 else fdot:=0;
  end;
end;
```

```
function fx(y:real;tr:integer):real;
begin
  case tr of
    1: fx:=y;
    2: if y>=0 then fx:=1-1/(1+y) else fx:=-1+1/(1-y);
    3: fx:=1/(1+exp(-y));
```

```

4: if y>0 then fx:=y else fx:=0;
5: begin
    if (y>-1)and(y<1) then fx:=y else
    if y<=-1 then fx:= -1 else fx:=1;
    end;
end;
end;

```

```

procedure set_input(x:dizitip;var y:layertip);
    var
        i:integer;
    begin
        for i:=1 to y.length do
            y.output[i]:= x[i];
        end;
end;

```

```

procedure propogate_layer(lower:layertip;var upper:layertip);
    var
        i,j:integer;
        sum:real;
    begin
        for j:=1 to upper.length do
            begin
                sum:=0;
                for i:=1 to lower.length do
                    if upper.wmask[j,i]<>0 then sum:= sum+lower.output[i]*upper.weight[j,i];
                    upper.output[j]:=fx(sum,upper.trans[j]);
                end;
            end;
        end;
end;

```

```

procedure propagate_forward;
    var
        i:integer;
    begin
        for i:=1 to (net.length-1) do
            propogate_layer(net.layer[i],net.layer[i+1]);
        end;
end;

```

```

procedure weightata;
var i:integer;
begin
with net.layer[2] do
for i:=0 to 3 do
begin
weight[i*2+1,3]:=weight[i*2+2,3];
weight[i*2+1,4]:=weight[i*2+2,4];
weight[i*2+1,5]:=weight[i*2+2,5];
end;
end;

```

```
end;
```

```
procedure init_weight;
```

```
var
  i,j,k:integer;
begin
  for k:=2 to net.length do
    for j:=1 to net.layer[k].length do
      for i:=1 to net.layer[k-1].length do
        if net.layer[k].wmask[j,i]=1 then
          begin
            net.layer[k].delta[j,i]:=0;
            net.layer[k].weight[j,i]:=random;
            if random<0.5 then
              net.layer[k].weight[j,i]:=- net.layer[k].weight[j,i];
          end;
        end;
      end;
    end;
  end;
```

```
procedure yaz_weight;
```

```
var
  i,j,k:integer;
begin
  writeln('WEIGHTS');
  for k:=2 to net.length do
    if (k=2) or (k=5) then
      for j:=1 to net.layer[k].length do
        begin
          for i:=1 to net.layer[k-1].length do
            if net.layer[k].wmask[j,i]=1 then
              write(net.layer[k].weight[j,i]:16:3);
          end;
        end;
      end;
    end;
  end;
```

```
procedure yaz_outs;
```

```
var
  i,j,k:integer;
begin
  writeln('OUTPUTS');
  for k:=1 to net.length do
    begin
      write('layer ',k:2,'>');
      for j:=1 to net.layer[k].length do
        write(net.layer[k].output[j]:8:3,' ');
      writeln;
    end;
  end;
```



```

procedure compute_output_error;
var
  i:integer;
begin
  with net.layer[net.length] do
    begin
      for i:=1 to length do
        error[i]:=fdot(output[i],trans[i])*(target[i]-output[i])
      end;
    end;
end;

procedure backpropagate_error(var receiver:layertip;sender:layertip;kl:integer);
var
  i,j:integer;
begin
  for i:=1 to receiver.length do
    begin
      receiver.error[i]:=0;
      for j:=1 to sender.length do
        if sender.wmask[j,i]<>0 then receiver.error[i]:=receiver.error[i]+sender.error[j]
          *sender.weight[j,i];
        receiver.error[i]:=receiver.error[i]*fdot(receiver.output[i],receiver.trans[i]);
      end;
    end;
end;

procedure propagate_error;
var
  i:integer;
begin
  compute_output_error;
  for i:=net.length downto 3 do
    backpropagate_error(net.layer[i-1],net.layer[i],i)
  end;
end;

procedure adjust_weight;
var
  i,j,k:integer;
  del:real;
begin
  for i:=2 to net.length do
    begin
      for j:=1 to net.layer[i].length do
        begin
          for k:=1 to net.layer[i-1].length do
            begin
              if net.layer[i].wmask[j,k]>0 then
                begin
                  del:=net.layer[i-1].output[k]*net.eta *net.layer[i].error[j]+net.alfa
                    *net.layer[i].delta[j,k];
                  net.layer[i].weight[j,k]:=net.layer[i].weight[j,k]+del;
                end;
            end;
          end;
        end;
      end;
    end;
end;

```

```

net.layer[i].delta[j,k]:=del;
end;
end; {k}
end; {j}
end; {i}
end;

```

```

procedure okurefdata;
var
  dk:integer;
  ii, i:integer;x:real;
  dosya:string[20];
begin
new(d[1]);
for dk:=1 to 1 do
  begin
  case dk of
  1:dosya:='\turbo\cy.dat';
  end;
Assign (refdosya, dosya);
Reset(refdosya);
for i:=1 to 1000 do
  begin
  if i>100 then ii:=i-100;
  with d[dk]^ii do
  readln(refdosya,x,guc,cubuk,sicaklik,periyot,iperiyot,iguc,kontrolU);
  end;
close(refdosya);
end;
end;

```

```

procedure normrefdat;
var
  i,k:integer;
begin
for k:=1 to 1 do
for i:=1 to 900 do
begin
with d[k]^i do
begin
guc:=guc/300;
cubuk:=cubuk/1000;
sicaklik:=sicaklik/300;
iguc:=iguc/300;
kontrolU:=kontrolU/5;
end;
end;
end;
end;

```

```

procedure mask_oku;
var k:integer;
begin
assign(refdosya, '\turbo\melih\mask.dat');
reset(refdosya);
readln(refdosya, net.length);
for i:=1 to net.length do read(refdosya, net.layer[i].length);

for i:=1 to net.layer[net.length-1].length do
    read(refdosya, net.layer[net.length].wmask[1,i]);

for k:=net.length-1 downto 2 do
for i:=1 to net.layer[k].length do
    for j:=1 to net.layer[k-1].length do
        read(refdosya, net.layer[k].wmask[i,j]);

for i:=net.length downto 2 do
    for j:=1 to net.layer[i].length do
        read(refdosya, net.layer[i].trans[j]);

for k:=net.length-1 downto 3 do
for i:=1 to net.layer[k].length do
    for j:=1 to net.layer[k-1].length do
        read(refdosya, net.layer[k].weight[i,j]);

close(refdosya);
for i:=1 to net.length do write(net.layer[i].length:3);
writeln;
for i:=1 to net.layer[4].length do write(net.layer[5].wmask[1,i]:3);
writeln;
for k:=net.length-1 downto 2 do
for i:=1 to net.layer[k].length do
    begin
    for j:=1 to net.layer[k-1].length do
        write(net.layer[k].wmask[i,j]:3);
    writeln;
    end;

for i:=net.length downto 2 do
    begin
    for j:=1 to net.layer[i].length do
        write(net.layer[i].trans[j]:3);
    writeln;
    end;

for k:=net.length-1 downto 3 do
for i:=1 to net.layer[k].length do
    begin
    for j:=1 to net.layer[k-1].length do

```

```

        write(net.layer[k].weight[i,j]:3:0);
    writeln;
    end;
end;
begin clrscr;
    mask_oku;
    init_weight;
    net.alfa:=0.5; { momentum }
    net.eta:=0.2; { adim   }
    okurefdata;
    normrefdat;
    clrscr;
    topmin:=9999;
    pref:=250/300;
    assign (dosya,'train.dat');
    rewrite(dosya);
    for kn:=1 to 2000 do
    begin
        toper:=0;
        smax:=400;
        for i:=1 to smax do s[i]:=i;
        repeat
            si:=s[random(smax)+1];
            if si=1 then inc(si);
            sref:=si;
            kk:=((si-1) div 1000) +1;si:=si-(kk-1)*1000;
            with d[kk]^si do
            begin
                ep:=1-guc/pref;
                pr:=ep*10;
                if pr> 0.1 then pr:= 0.1;
                if pr<-0.1 then pr:=-0.1;
                xp[1]:=ep;
                xp[2]:=(pr-1/peryt);
                xp[3]:=sicaklik;
                xp[4]:=guc;
                xp[5]:=cubuk;
                target[1]:=kontrolu;
            end;
            set_input(xp,net.layer[1]);
            propagate_forward;
            propagate_error;
            adjust_weight;
            toper:=toper+sqr((target[1]-net.layer[net.length].output[1]));
            for i:=sref to smax-1 do s[i]:=s[i+1];
            dec(smax);
        until smax=0;
        writeln(kn:5,sqrt(toper):10:4,sqrt(topmin):10:4);
        writeln(kn,sqrt(toper):10:4);
    end;
end;

```

```
if toper < topmin then
  begin
    topmin := toper;
    assign(netdos, 'ms10.dat');
    rewrite(netdos);
    write(netdos, net);
    close(netdos);
  end;
dispose(d[1]);
end.
```



```

unit net3;
{*****
  Eğitim İçin Gerekli Olan Ağ Yapısının Yazılımı
  *****)}
interface
uses crt;
const
  MAXLEN=25;
  MAXLAYER=6;
  lineer=1;
  sigmoid=2;
type
  dizitip=array[1..MAXLEN] of real;
  diziint=array[1..MAXLEN] of integer;
  matristip=array[1..MAXLEN,1..MAXLEN]of real;
  matristipi=array[1..MAXLEN,1..MAXLEN]of integer;
  layertip=record
    length:integer;
    output:dizitip;
    weight:matristip;
    wmask:matristipi;
    error:dizitip;
    delta:matristip;
    trans:diziint;
  end;
  nettip=record
    length:integer;
    layer:array[1..MAXLAYER]of layertip;
    alfa,eta:real;
  end;
var
  datdos:string[20];
  k,ornek,i,j:integer;
  net:nettip;
  xp:dizitip;
  netdos:file of nettip;
  son:real;
procedure okuweight;
function netum(guc,cubuk,sicaklik,periyot,iguc:real):real;

implementation

function netum(guc,cubuk,sicaklik,periyot,iguc:real):real;
function fx(y:real;tr:integer):real;
begin
  son:= y;
  case tr of
    1: fx:=y;
    2: if y>=0 then fx:=1-1/(1+y) else fx:= -1+1/(1-y);
  end;
end;

```



```
        write(net.layer[k].weight[j,i]:10:5,');
        writeln;
    end;
end;

begin
xp[1]:=guc;
xp[2]:=cubuk ;
xp[3]:=sicaklik;
xp[4]:=periyot;
xp[5]:=iguc;

set_input(xp,net.layer[1]);
propagate_forward;
netum:=net.layer[net.length].output[1];
end;

procedure okuweight;
begin
    assign(netdos,datdos);
    reset(netdos);
    read(netdos,net);
    close(netdos);
end;
end.
```


ÖZGEÇMİŞ

1971 yılında Mardin’de doğdu. İlk, orta öğrenimini Mardin’de tamamladı. Lise öğrenimini Ankara Gazi Teknik Lisesi Bilgisayar İşletim Teknisyenliği Bölümü’nde yaptı. 1988 yılında girdiği Gaziantep Üniversitesi’nde bir yıl İngilizce hazırlık okuduktan sonra 1989 yılında ÖSYM sınavı ile girdiği Marmara Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Teknolojisi Bölümü’nden 1993 yılında mezun oldu. Ekim 1994 yılında girdiği, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Elektronik-Bilgisayar Eğitimi Anabilim Dalı’nda Yüksek Lisans yapmaktadır.

1993 yılından beri Kocaeli Üniversitesi Teknik Eğitim Fakültesi, Elektronik-Bilgisayar Eğitimi Bölümü’nde Araştırma Görevlisi olarak görev yapmaktadır.

